Anca APĂTEAN, Alexandrina ROGOZAN, Abdelaziz BENSRHAIR

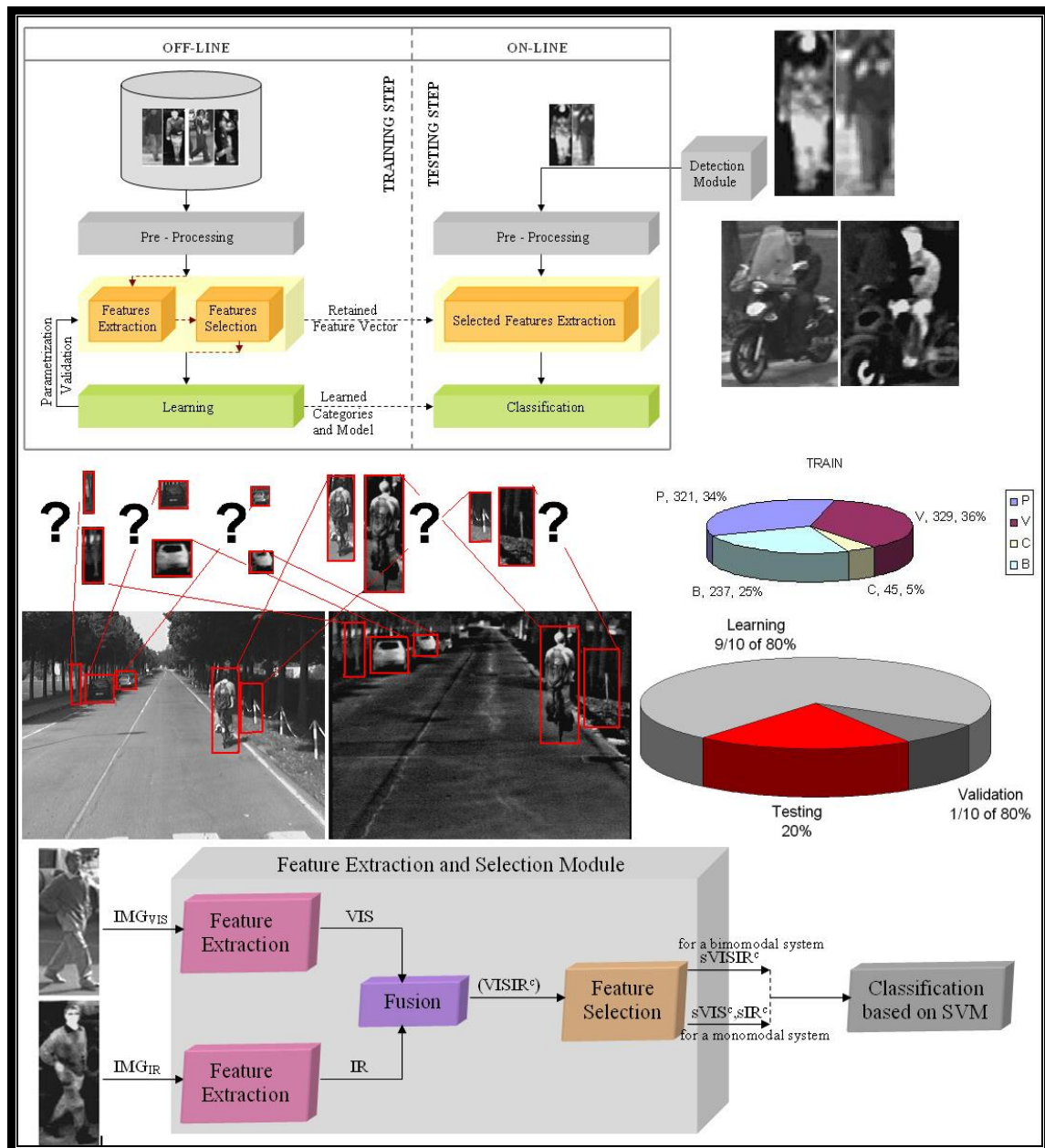# Contributions to the Information Fusion.

Application to *Obstacle Recognition* in Visible and Infrared Images

Prefață

Această carte constituie de fapt teza de doctorat a subsemnatei, publicată în forma ei originală, așa cum a fost susținută în Octombrie 2010, la INSA Rouen, Franța. Teza a fost realizată în cotutelă între Universitatea Tehnică din Cluj-Napoca(UTCN) și INSA Rouen din Franța, drepturile privind modalitatea de redactare și susținere a acesteia fiind cedate INSA, după cum a fost încheiat contractul de cotutelă. Astfel, diploma de doctor a fost eliberată de INSA, în Februarie 2011, iar cu sprijinul Ministerului Educației, Cercetării, Tineretului și Sportului această diplomă a fost recunoscută și în Romania.

Nu am reușit niciodată să adun suficiente cuvinte de laudă și mulțumire la adresa dnei Alexandrina Rogozan, care a fost pentru mine mai mult decât o „encadrante" de teză. Ea a fost atât persoana care a dat esența acestei lucrări cât și persoana care a urmărit îndeaproape toate ideile și rezultatele publicate în acest domeniu. Doamna Alexandrina este și va continua mereu să rămână modelul meu, demn de urmat, atât pe plan spiritual cât și din punct de vedere al realizărilor materializate în domeniul academic.

Domnul Prof. Abdelaziz Bensrhair a fost mai mult decât directorul tezei din partea INSA, o persoană care mereu a știut să-mi strângă mâna, să mă încurajeze să merg mai departe; a fost alături de mine și m-a sprijinit în atingerea scopului definit de subiectul tezei. Dânsul m-a coordonat împreună cu dna Alexandrina pe durata stagiilor de la INSA, au avut atât răbdarea să asiste la prezentarea sintezelor și rezultatelor obținute, cât și energia de a-mi sugera apoi direcții viitoare. Toate colaborările mele în cadrul INSA le datorez acestor două persoane minunate: sunt mai mult decât recunoscătoare pentru inițierea colaborării cu dl Alberto Broggi, o personalitate în domeniul vehicolelor inteligente.

Domnului Prof. Corneliu Rusu, nu în ultimul rând, vreau să-i mulțumesc pentru suportul acordat în UTCN; de câte ori am apelat la dânsul, am primit doar sfaturi bune și pertinente. Dânsul este părintele meu academic, de aici din Romania, fiind și cel care m-a format și sprijinit ca doctorand.

Tuturor acestor persoane nu am suficiente cuvinte să le pot transmite gratitudinea și recunoștința mea pentru sprijinul pe care mi l-au acordat și pentru răbdarea și profesionalismul de care au dat dovadă.

Autorii acestui material au fost mai mult decât implicați în obținerea lui în această formă. Perioadele de cercetare activă petrecute la INSA, din 2006 până în 2010 mi-au fost benefice, formându-mă ca cercetător și concretizându-se în acest material.

Autor principal,
Anca Apătean

Multumiri

Adresez încă o dată mulțumiri întregului colectiv de la INSA, care s-a purtat excepțional cu mine, m-au primit în sânul lor ca pe un membru veritabil. În special doresc să mulțumesc lui Laura Dioșan de la UBB și lui Filip Florea de la UPB, colegii mei români de la INSA; ei fost modelul meu de urmat îndeaproape. Ori de câte ori am avut vreo problemă, personală sau științifică, ei m-au ajutat. Sunt mai mult decât recunoscătoare pentru că i-am cunoscut!

Mare parte a rezultatelor prezentate în această teză au fost obținute datorită sprijinului financiar de la INSA, burselor acordate, dar și bursei Erasmus și grantului CNCSIS, din anii 2007-2008. Posibilitatea încheierii acordului de cotutelă nu ar fi existat fără colaborarea celor 2 universități, mulțumesc celor implicați în realizarea acestor demersuri. Mulțumesc de asemenea colegilor mei de la UTCN care au fost de acord cu plecările mele, uneori spontane la INSA, și care m-au înlocuit de câte ori a fost nevoie.

Nu în ultimul rând, vreau să mulțumesc dlui Alberto Broggi pentru amabilitatea de a interveni pe lângă laboratorul VisLab pentru obținerea bazei de imagini Tetravision.

Anca Apătean
2015

Preface

This book is actually the undersigned thesis, published in its original form, as it was presented in October 2010 at INSA Rouen, France. The thesis was carried out in co-direction between Technical University of Cluj-Napoca (TUCN) and INSA Rouen from France, the rights as concern the modality of its writing and defence being ceded to INSA, as the bilateral contract mentioned. The doctoral degree and diploma were thus released from the INSA part in February 2011 and later also recognised by the Minister of Education, Research, Youth and Sport (MECTS) from Romania.

I was never able to gather enough words of praise and gratitude to Mrs Alexandrina Rogozan; she was for me more than a "encadrante de these". She is both the person who gave the essence of this work and the person who closely tracked all the ideas and results published in this area. Mrs Alexandrina is and will always continue to remain my model, worthy to emulate, in both spiritual term and of the achievements materialized in the academic results.

Mr. Prof. Abdelaziz Bensrhair was more than the thesis director from INSA; he is the person who knew to shake my hand, to support me and encourage me to go further. He always had time to listen to me and to help me in achieving the goal defined by the thesis topic. During the internships at INSA, both Mrs Alexandrina and Mr Aziz had the patience to assist to the summaries and results presentations, but also the energy to suggest future directions. All my collaborations within INSA I owe to these two wonderful people. I am more than grateful to them for initiating the cooperation with Mr. Alberto Broggi, a personality in the field of intelligent vehicles.

To Mr. Prof. Corneliu Rusu I want to thank for the support in TUCN; whenever I contacted him, with any problem, he always gave me the best and the most relevant advice. He is my academic father, here in Romania, being the one who formed me and supported me as a graduate student.

To all these people, I have not enough words to render my appreciation and gratitude for the support they have given me and for their patience and professionalism they have shown.

The authors of this material were more than involved in its form. The active periods of research at INSA, occurred from 2006 to 2010, were beneficial to me, forming me as a researcher and being concretised in this material.

Main author,
Anca Apătean

Acknowledgement

Main author,
Anca Apătean

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA, CLUJ-NAPOCA, ROMÂNIA
FACULTY OF ELECTRONICS, TELECOMMUNICATIONS
AND INFORMATION TECHNOLOGY
and
INSTITUT NATIONAL DES SCIENCES APPLIQUEES, ROUEN, FRANCE
LABORATOIRE D'INFORMATIQUE, DE TRAITEMENT DE L'INFORMATION
ET DES SYSTEMES

Contributions to the Information Fusion. Application to Obstacle Recognition in Visible and Infrared Images

Ph.D. Student: **Anca DISCANT (épouse Apătean**)

Ph.D. Advisor: Professor **A. Bensrhair**
Institut National des Sciences Appliquées, Rouen, France

Ph.D. Advisor: Associate Professor **A. Rogozan**
Institut National des Sciences Appliquées, Rouen, France

Ph.D. Advisor: Professor **C. Rusu**
Technical University of Cluj-Napoca, Romania

2010

THÈSE DE DOCTORAT

Contributions à la fusion des informations. Application à la reconaissance des obstacle dans les images visible et infrarouge

présentée et soutenue publiquement *le vendredi 15 octobre 2010*

pour l'obtention du grade de
Docteur de l'Institut National des Sciences Appliquées de Rouen, France
et de l'Université Technique de Cluj-Napoca, România
par

Anca DISCANT (épouse Apătean)

Composition du jury :

Rapporteurs :	Fabrice MERIAUDEAU	-	Professeur des Universités, LE2I, IUT Le Creusot, France
	Vasile BUZULOIU	-	Professeur des Universités, LAPI, Université Technique de Bucuresti, Roumanie
Examinateur :	Eugen LUPU	-	Professeur des Universités, ETTI, Université Technique de Cluj-Napoca, Roumanie
Directeurs :	Corneliu RUSU	-	Professeur des Universités, ETTI, Université Technique de Cluj-Napoca, Roumanie
	Abdelaziz BENSRHAIR	-	Professeur des Universités, LITIS, INSA de Rouen, France
Encadrante :	Alexandrina ROGOZAN	-	Maître de Conférences, LITIS, INSA de Rouen, France

DEDICATION

Je dédie cette thèse à mon mari qui m'a toujours soutenue, aide et encouragée et que j'aime tant.

ACKNOWLEDGMENTS

This dissertation would have never been finished without the help of many people, to whom I would like to express my sincere gratitude.

I want to thank my advisers professor *Corneliu RUSU* and professor *Abdelaziz BENSRHAIR* for their patience and encouragement. They helped me with scientific and financial support during my Ph. D. stage.

I want to especially thank to *Alexandrina ROGOZAN*, an extraordinary person who helped me in a scientific, organisational and personal way. I wish to thank her also for the time that she dedicated to me during the last years. She always made time to answer my questions, and her advices, observations and supports were and are very valuable for me.

I want to thank also to professor *Eugen LUPU* and my colleague *Simina EMERICH* from UTCN for their understanding and trust in my ability to complete this work.

I would like to address many thanks to all *Ph.D. students and staffs from INSA* who had kindly invited me as a colleague of them, helped me and supported me when I needed.

I am very thankful to *my families (Discant and Apatean)* who sustained and encouraged me during this thesis.

I want also to thank to my friend and INSA's colleague *Laura DIOSAN*, who was an example for me and inspired me for my research.

Anca Apatean (Discant)
October 2010

ABSTRACT

The interest for the intelligent vehicle field has been increased during the last years, must probably due to an important number of road accidents. Many accidents could be avoided if a device attached to the vehicle would assist the driver with some warnings when dangerous situations are about to appear. In recent years, leading car developers have recorded significant efforts and support research works regarding the intelligent vehicle field where they propose solutions for the existing problems, especially in the vision domain. Road detection and following, pedestrian or vehicle detection, recognition and tracking, night vision, among others are examples of applications which have been developed and improved recently. Still, a lot of challenges and unsolved problems remain in the intelligent vehicle domain.

Our purpose in this thesis is to design an Obstacle Recognition system for improving the road security by directing the driver's attention towards situations which may become dangerous. Many systems still encounter problems at the detection step and since this task is still a work in progress in the frame of the LITIS laboratory (from INSA), our goal was to develop a system to continue and improve the detection task. We have focused solely on the fusion between the visible and infrared fields from the viewpoint of an Obstacle Recognition module. Our main purpose was to investigate if the combination of the visible-infrared information is efficient, especially if it is associated with an SVM (Support Vector Machine)-based classification.

The outdoor environment, the variety of obstacles appearance from the road scene (considering also the multitude of possible types of obstacles), the cluttered background and the fact that the system must cope with the moving vehicle constraints make the categorization of road obstacles a real challenge. In addition, there are some critical requirements that a driver assistance system should fulfil in order to be considered a possible solution to be implemented on board of a vehicle: the system cost should be low enough to allow to be incorporated in every series vehicle, the system has to be fast enough to detect and then recognize obstacles in real time, it has to be efficient (to detect all obstacles with very few false alarms) and robust (to be able to face different difficult environmental conditions).

To outline the system, we were looking for sensors which could provide enough information to detect obstacles (even those occluded) in any illumination or weather situation, to recognize them and to identify their position in the scene. In the intelligent vehicle domain there is no such a perfect sensor to handle all these concerned tasks, but there are systems employing one or many different sensors in order to perform obstacles detection, recognition or tracking or some combination of them. After comparing advantages and disadvantages between passive and active technologies, we chose the proper sensors for developing our Obstacle Detection and Recognition system. Due to possible interferences among active sensors, which could be critical for a large number of vehicles moving simultaneously in the same environment, we concentrate on using passive sensors, which are non-invasive, like cameras. Therefore, our proposed system employ visible spectrum and infrared spectrum cameras, which are relatively chosen to be complementary, because the system must work well even under difficult conditions, like poor illumination or bad-weather situations (such as dark, rain, fog).

The monomodal systems are adapted to a single modality, either visible or infrared and even if they provide good recognition rates on the test set, these results could be improved by the combined processing of the visible and infrared information, which means in the frame of a bimodal system. The bimodal systems could take different forms in function of the level at which the information is combined or fused. Thus, we propose three different fusion systems: at the levels of features or at the

level of SVM's kernels, or even higher, at the level of matching-scores provided by the SVM. Each one of these systems improves classification performances comparing to the monomodal systems. In order to ensure the adaptation of the system to the environmental conditions, within fusion schemes the kernels, the matching-scores and the features were weighted (with a sensor weighting coefficient) according to the relative importance of the modality sensors. This allowed for better classification performances. In the frame of the matching-scores fusion there is also the possibility to dynamically perform the adaptation of the weighting coefficient to the context.

In order to represent the obstacles' images which have to be recognized by the Obstacle Recognition system, some features have been preferred to encode this information. These features are obtained in the features extraction module and they are wavelet features, statistical features, the coefficients of some transforms, and others. Generally, the features extraction module is followed by a features selection one, in which the importance of these features is estimated and only the ones that are most relevant will be chosen to further represent the information. Different features selection methods are tested and compared in order to evaluate the pertinence of each feature (and of each family of features) in relation to our objective of obstacle classification. The pertinence of each vector constructed based on these features selection methods was first evaluated by a KNN (k Nearest Neighbours) (with the number of neighbours $k = 1$) classifier, due to the simplicity in its usage: it does not require a parameter optimization process (as the SVM does).

To increase the accuracy of the classification, but also to obtain a powerful classifier, more parametrizable for the proposed fusion schemes, the KNN one was later (after the best features selection method have been chosen on the training set and the most relevant features have been selected) replaced by a SVM classifier. Because there is not known beforehand which combination of the SVM hyper-parameters is the most appropriate for a certain classification problem, an operation of model search, performed by 10 folds cross-validation, provides the optimized kernel for the SVM to be used on each fusion schemes and on each feature vector we considered.

Finally, we tested our features extraction, features selection and the proposed fusion schemes for a 4-class problem, thus discriminating between vehicles, pedestrians, cyclists and background obstacles. The results have proven that all bimodal visible-infrared systems are better than the monomodal ones, thus the fusion is efficient and robust since it allows for improving the recognition rates. In addition, features selection scheme provides smaller vector comprising only the most relevant features for the classification process. This reduction of the feature-vector dimension besides providing higher accuracy rates, allows the reduction of the computation time which is crucial in this type of application.

Keywords: Fusion, Infrared cameras, Features extraction, Features selection, Support Vector Machine, Kernels, Matching-scores, Hyper-parameter optimization, Model search, 10 folds cross-validation.

Résumé

L'intérêt pour le domaine des véhicules intelligents a progressé au cours des dernières années, doit probablement dû à un nombre élevé d'accidents sur la route. La plupart des accidents pourraient être évités si les voitures étaient équipées avec un dispositif d'assistance du conducteur qui fournir des signaux d'avertissement quand les situations dangereuses sont sur le point d'apparaître. Récemment, les grandes entreprises constructeurs d'automobiles ont enregistré d'importants efforts et de soutien pour la recherche et la résolution de problèmes concernant le domaine des véhicules intelligents, où ils proposent des solutions pour les problèmes existants, en particulier dans le domaine de la vision. La détection et le suivi de la route, la détection des piétons ou des véhicules, leur reconnaissance et leur suivi, la vision au cours de nuit sont des exemples d'applications développées et améliorées récemment. Toutefois, il y a encore de nombreux défis et problèmes non résolus dans le domaine des véhicules intelligents.

Notre objectif dans cette thèse est de décrire un système de reconnaissance des obstacles pour améliorer la sécurité routière en dirigeant l'attention du conducteur vers des situations qui peuvent devenir dangereuses. De nombreux systèmes rencontrent encore des difficultés à l'étape de détection et, depuis cette tâche est encore un travail en cours dans le cadre du laboratoire LITIS de l'INSA, notre objectif était de développer un système pour poursuivre et améliorer la tâche de détection. Nous nous sommes concentrés uniquement sur la fusion visible et infrarouge du point de vue d'un module de reconnaissance d'obstacles. Notre but principal était de déterminer si la combinaison de l'information visible-infrarouge est efficace, surtout si elle est associée à une classification basée sur SVM (Séparateur a Vaste Marge, en anglais SVM - Support Vector Machine).

L'environnement extérieur, la variété de l'apparence des obstacles de la scène dans la route (en considérant également les nombreux types possibles d'obstacles), le fond encombré apparaissant sur ces obstacles et le fait que le système doit faire face aux contraintes de véhicule en mouvement font la catégorisation des obstacles sur la route est un véritable défi. De plus, il existe quelques exigences essentielles d'un système d'assistance au chauffeur pour être considéré comme une solution possible à mettre en oeuvre à bord d'un véhicule: le coût du système devrait être suffisamment faible pour permettre l'incorporation dans chaque véhicule de série, le système doit être suffisamment rapide pour détecter puis reconnaître les obstacles en temps réel; il doit aussi être efficace (pour détecter tous les obstacles avec très peu de fausses alarmes) et robuste (pour pouvoir faire face à différentes conditions difficile de l'environnement).

Pour le système proposé ont été développé des capteurs qui peuvent fournir suffisamment d'informations pour détecter les obstacles (même l'occlusion) dans toute situation d'illumination ou de mauvais temps, de reconnaître et d'identifier leur position sur la scène. Dans le domain des véhicules inteligentes il n'existe pas de tel capteur. Il existe des systèmes qui utilisent un ou plusieurs capteurs différents pour effectuer la détection des obstacles, leur reconnaissance ou leur suivi, ou même des combinaisons de ces fonctions. Après avoir comparé les avantages et les désavantages des technologies actives et passives, on choisi les capteurs les plus adaptés pour le système de détection et de reconnaissance des obstacles proposée. En raison de possibles interférences qui peuvent survenir entre les différents capteurs actifs, les interférences qui peuvent être critiques pour un grand nombre de véhicules qui se déplacent simultanément dans le même environnement, nous nous sommes concentrés uniquement sur l'utilisation de capteurs passifs, non invasive, à savoir les caméras. Ainsi, le système proposé utilise des caméras opérant dans le spectre visible et infrarouge, choisi relativement complémentaires, car le système doit être capable de bien travailler dans diverses conditions difficiles, tels que l'illumination pauvre ou mauvais temps (telles que l'obscurité ou nuit, du brouillard ou de pluie).

Les systèmes monomodaux sont adaptés à une condition particulière, que ce soit visible ou infrarouge, et même si elles offrent bonne taux de reconnaissance sur la base de test, ces résultats peuvent être encore améliorées par traitement de l'information combinée visible et infrarouge, ce qui signifie dans un système bimodal. Les systèmes bimodaux peut prendre différentes formes selon le niveau auquel l'information est combinée ou fusionnées. Ainsi, nous proposons trois systèmes différents de fusion: au niveau de caractéristiques, au niveau de noyaux SVM, ou même plus, au niveau de scores fournis par SVM. Chacun de ces systèmes améliore les performances par rapport aux systèmes monomodaux. Pour assurer l'adaptation de système au l'environnement, dans les schémas de fusion les noyaux, les scores et les caractéristiques ont été pondérés (avec un coefficient de pondération du capteur) en fonction de l'importance relative des capteurs de modalité. Cela permet d'obtenir des performances de classification supérieur. Dans le cas de la fusion le score permet de réaliser une adaptation dynamique du coefficient de pondération au contexte.

Pour représenter l'information sur les obstacles qui doivent être reconnus par le système de reconnaissance d'obstacles, ont préféré certains types de caractéristiques pour le codage des informations contenues dans les images des obstacles. Ces caractéristiques sont obtenues dans le module d'extraction de caractéristiques de type ondelettes, des caractéristiques statistiques, des coefficients des transformations et d'autres. En général, le module d'extraction de caractéristiques est suivi par un module de sélection qui ne retient que les caractéristiques les plus pertinentes pour représenter l'information. Différentes méthodes de sélection des caractéristiques sont testées et comparées pour évaluer la pertinence de chaque caractéristique (et chaque famille de caractéristiques) par rapport à notre objectif de classification des obstacles. La pertinence de chaque vecteur de caractéristiques construit sur ces méthodes de sélection a été évalué en premier avec un classificateur PPV (Plus Proches Voisins, en anglais KNN - k Nearest Neighbours) (avec le nombre de voisins k = 1) en raison de sa simplicité d'utilisation: il ne nécessite pas un processus d'optimisation des paramètres tel que le classificateur SVM requis.

Pour augmenter la précision de la classification, mais aussi pour obtenir un classifieur fort, avec plus des paramètres, pour les schemas de fusion proposé le classifieur PPV a ensuite (après avoir choisi la meilleure méthode de sélection des caractéristiques dans l'ensemble d'apprentissage et les plus pertinentes caractéristiques ont été sélectionnées) remplacé par un classificateur SVM. Parce qu'il n'est pas connu à l'avance quelle combinaison des hyper-paramètres de SVM est le mieux pour un problème de classification particulier, il a eu une recherche du modèle, réalisé grâce à une technique de validation croisée a 10 fois, qui fournissent le noyau optimisé pour les SVM, le noyau qui va etre utilisé pour chaque schema de fusion et chaque vecteur des caractéristiques considéré.

Enfin, nous avons testé les schemas d'extraction des caractéristiques, de leur sélection et de fusion à un problème avec 4 classes, tel que la discrimination a été faite entre les véhicules, les piétons, les cyclistes et les obstacles dans le fond. Les résultats ont démontré que tous les systèmes bimodaux visible-infrarouge sont mieux que les correspondants mono-modaux. La fusion est efficace et robuste parce qu'elle permet d'améliorer les taux de reconnaissance. De plus, la sélection des caractéristiques offre un vecteur comprenant seulement les caractéristiques les plus pertinentes pour le processus de classification. Cette réduction de la taille des vecteurs des caractéristiques en plus de produire le taux de précision élevé, peut aussi réduire le temps de calcul qui est crucial dans de telles applications.

Mot clés: Fusion, Caméras infrarouges, Extraction de caractéristiques, Sélection des caractéristiques, Séparateur a Vaste Marge, Noyau, Scores, Optimisation des hyper-paramèters, Recherche du modèle, Validation croisée a 10 fois.

Rezumat

Interesul pentru domeniul vehicolelor inteligente a crescut simţitor în ultimii ani, cel mai probabil datorită numărului mare de accidente rutiere înregistrate. Majoritatea accidentelor ar putea fi evitate dacă autovehicolul ar avea ataşat un dispozitiv de asistenţă a şoferului care să furnizeze semnale de atenţionare atunci când intervin situaţii periculoase. Recent, marile companii de autovehicule au înregistrat eforturi semnificative şi suport pentru cercetarea şi soluţionarea problemelor din acest domeniu al vehicolelor inteligente, propunând soluţii pentru problemele existente, mai ales în domeniul viziunii computerizate. Detecţia şi urmarirea drumului, detecţia pietonilor sau a vehicolelor, recunoşterea şi urmărirea acestora, viziunea pe timp de noapte sunt câteva dintre aplicaţiile dezvoltate şi îmbunătăţite recent. Cu toate acestea, încă există foarte multe provocări şi probleme nerezolvate în domeniul vehicolelor inteligente.

Scopul nostru în această teză este de a descrie un sistem de recunoaştere a obstacolelor destinat îmbunătăţirii securităţii rutiere prin direcţionarea atenţiei şoferului înspre situaţiile posibil periculoase. Multe sisteme încă întâmpină dificultăţi în pasul de detecţie şi deoarece acest pas este încă în progres în cadrul laboratorului LITIS al INSA, scopul nostru a fost de a dezvolta un sistem care să continue şi să îmbunătăţească procesul de detecţie. Ne-am concentrat doar pe partea de fuziune a câmpurilor vizibil şi infraroşu din punct de vedere al unui modul de recunoaştere a obstacolelor. Obiectivul nostru principal a fost să investigăm dacă combinarea informaţiei vizibil-infraroşu este eficientă, în special dacă este asociată cu o clasificare pe baza SVM (Support Vector Machine, în română - maşină cu suport vectorial).

Mediul exterior, multiplele posibilităţi de apariţie a obstacolelor în scena rutieră (considerând de asemenea şi multitudinea tipurilor de obstacole posibile), fondul foarte încărcat pe care pot să apară aceste obstacole şi faptul că sistemul trebuie să considere şi constrângerile de mişcare ale vehicolului determină categorizarea obstacolelor rutiere să fie o adevărată provocare. In plus, există câteva cerinţe critice pe care un sistem de asistenţă a şoferului trebuie să le îndeplinească pentru a putea fi considerat o posibilă soluţie de implementat la bordul unui vehicol: costul sistemului să fie suficient de scăzut pentru a permite încorporarea lui în orice vehicol produs în serie, sistemul să fie suficient de rapid încât să detecteze şi să recunoască obstacolele în timp real; de asemenea, el trebuie să fie eficient (să detecteze toate obstacolele şi foarte puţine alarme false) şi robust (să fie capabil să facă faţă la diferite condiţii de mediu dificile).

Pentru sistemul propus s-au căutat senzori care să poată furniza suficientă informaţie pentru a detecta obstacolele (chiar şi cele ocluzate) în orice situaţie de iluminare sau vreme, pentru a le recunoaşte şi a le identifica poziţia în scenă. În domeniul vehicolelor inteligente nu există un astfel de senzor care să poată rezolva singur toate aceste cerinţe. Există însă sisteme care folosesc unul sau mai mulţi senzori diferiţi pentru a realiza detecţia, recunoaşterea sau urmărirea obstacolelor, sau chiar combinaţii ale acestor funcţii. După compararea avantajelor şi a dezavantajelor tehnologiilor pasivă şi activă, s-au ales senzorii cei mai potriviţi pentru sistemul de detecţie şi recunoaştere a obstacolelor propus. Datorită posibilelor interferenţe care pot apare între diferiţi senzori activi, interferenţe ce pot fi critice pentru un număr ridicat de vehicole ce se mişcă simultan în acelaşi mediu, ne-am concentrat doar asupra folosirii senzorilor pasivi, neinvazivi, adică a camerelor video. Astfel, sistemul propus foloseşte camere cu funcţionare în spectrul vizibil şi infraroşu, alese relativ complementare, deoarece sistemul trebuie să poată funcţiona bine în diferite condiţii dificile, precum iluminare slabă sau vreme proastă (cum ar fi întuneric, ceaţă sau ploaie).

Sistemele monomodale sunt adaptate la o singură modalitate, fie vizibilă fie infraroşie şi chiar dacă ele furnizează rate de recunoaştere bune pe setul de test, aceste rezultate pot fi îmbunătăţite şi

mai mult prin procesarea combinată a informaţiei vizibile şi infraroşii, ceea ce înseamnă în cadrul unui sistem bimodal. Sistemele bimodale pot lua diferite forme în funcţie de nivelul la care este combinată sau fuzionată informaţia. Astfel, propunem trei sisteme de fuziune diferite: la nivelul caracteristicilor, la nivelul kernelelor SVM, sau chiar la un nivel mai ridicat, la nivelul scorurilor de potrivire furnizate de SVM. Fiecare dintre aceste sisteme îmbunătăţeşte performanţele comparativ cu sistemele monomodale. Pentru a asigura adaptarea sistemului la condiţiile de mediu, în schemele de fuziune kernelele, scorurile de potrivire şi caracteristicile au fost ponderate (cu un coeficient de ponderare al senzorului) în concordanţă cu importanţa relativă a senzorilor de modalitate. Aceasta permite obţinerea unor performanţe mai ridicate la clasificare. În cadrul fuziunii scorurilor de potrivire de asemenea există posibilitatea de a realiza în mod dinamic adaptarea coeficientului de ponderare la context.

Pentru a reprezenta informaţia despre obstacolele ce trebuie recunoscute de sistemul de recunoaştere al obstacolelor, s-au preferat câteva tipuri de caracteristici pentru codarea informaţiei existentă în imaginile obstacolelor. Aceste caracteristici sunt obţinute în modulul de extragere al caracteristicilor şi ele sunt: caracteristici wavelet, caracteristici statistice, coeficienţii unor transformate şi altele. În general, modulul de extragere al caracteristicilor este urmat de un modul de selecţie a acestora, în care este estimată importanţa lor şi doar acele caracteristici care sunt cele mai relevante vor fi alese ulterior pentru a reprezenta informaţia. Diferite metode de selecţie a caracteristicilor sunt testate şi comparate pentru a evalua pertinenţa fiecărei caracteristici (şi a fiecărei familii de caracteristici) raportat la obiectivul nostru de clasificare a obstacolelor. Pertinenţa fiecărui vector de caracteristici construit pe baza acestor metode de selecţie a fost evaluată prima dată pe baza unui clasificator KNN (k Nearest Neighbours, în română - cei mai apropiaţi k vecini) (cu numărul de vecini $k = 1$), datorită simplităţii acestuia la utilizare: el nu necesită un proces de optimizare a parametrilor aşa cum necesită clasificatorul SVM.

Pentru a creşte acurateţea clasificării, dar şi pentru a obţine un clasificator puternic, mai parametrizabil pentru schemele de fuziune propuse, clasificatorul KNN a fost ulterior (după ce a fost aleasă metoda cea mai bună de selecţie a caracteristicilor pe setul de antrenare şi cele mai relevante caracteristici au fost selectate) înlocuit cu un clasificator SVM. Deoarece nu se cunoaşte dinainte ce combinaţie de hiper-parametrii ai SVM este cea mai potrivită pentru o anumită problemă de clasificare, a fost nevoie de o operaţie de căutare a modelului, realizată printr-o tehnică de validare încrucişată prin 10 directoare, care să furnizeze kernelul optimizat pentru SVM, kernel folosit ulterior pentru fiecare schemă de fuziune şi pentru fiecare vector de caracteristici considerat.

În final, s-au testat schemele de extragere a caracteristicilor, de selecţie a acestora şi de fuziune propuse pentru o problemă cu 4 clase, adică s-a realizat discriminarea între vehicole, pietoni, ciclişti şi obstacole din fond. Rezultatele au demonstrat că toate sistemele bimodale vizibil-infraroşu sunt mai bune decât cele monomodale. Fuziunea este eficientă şi robustă deoarece permite îmbunătăţirea ratelor de recunoaştere. În plus, schema de selecţie a caracteristicilor furnizează un vector cuprinzând doar cele mai relevante caracteristici pentru procesul de clasificare. Această reducere a dimensiunii vectorului de caracteristici pe lângă faptul că produce rate de acurateţe mai ridicate, permite şi reducerea timpului de calcul care este crucial în acest tip de aplicaţii.

Cuvinte cheie: Fuziune, Cameră cu infraroşu, Extragere de caracteristici, Selecţie de caracteristici, Maşina cu suport vector, Nucleu, Scoruri, Optimizare de hiper-parametrii, Căutare de model, Validare încrucişată cu 10 directoare.

Thesis structure

The research presented in this dissertation advances the theory, the design and the implementation of the proposed Obstacle Recognition component in the frame of an entire Obstacle Detection and Recognition (ODR) system. The proposed recognition component is designed for improving the road security by discriminating between different types of obstacles from the road and it is based on the fused information provided by visible spectrum and infrared spectrum cameras. The present work contains 169 bibliographical references and it is structured in five chapters as follows.

The **first chapter** is intended to give a motivation for why the ODR task is an important area to be investigated, and how the work done in this thesis can contribute to an ODR system. It also introduces the basic information necessary to understand the main characteristics and problems of the ODR task. The fundamental requirements for developing an affordable-price, real-time, efficient and robust system to be deployed on board of the vehicle are presented, followed by the specific characteristics of the ODR systems from the intelligent vehicle domain. Finally, our proposed solution and how we intend to solve all the specified requirements for the ODR system are introduced.

Chapter 2 is dedicated to the systems (and the sensors they employed) from the intelligent vehicle field which addressed a problem similar to our, therefore it is a state of the art. Different types of sensors are investigated and their advantages and drawbacks are presented in the frame of most cited systems developed in the intelligent vehicle domain. The main types of sensors were examined, but we concentrated especially on the information each type of sensor could provide. Some sensors may have many advantages, but also some strong limitations, which make them to be not-so-properly for the implementation of an ODR system. The chapter is mainly focused on comparing advantages and disadvantages between the passive and active technologies and choosing the best solution for developing an ODR system. Considering the high price and the interference problems, we chose not to employ any active technology for the proposed system. In this chapter we motivated our choice to use only cameras, so passive sensors operating in a non-invasive way and which in addition are also cheaper than their counterparts, the active sensors.

In the next three chapters, our proposed system is presented.

In **Chapter 3** the baseline Obstacle Recognition component is presented, in the frame of an entire ODR system. The problems addressed here are intended to make a detailed presentation of the functioning mode and of the components that form this base system. The Obstacle Recognition component is more emphasized, and the following are also presented: the image database on which the proposed schemes have been experimented, the measures by which the performances of these schemes have been evaluated, but also how the feature vector that will characterize each instance within the system was composed. Basic notions about the classifier used in the frame of the developed fusion schemes, which is a SVM, are also presented. The individual or monomodal visible and infrared systems are also illustrated, together with a first set of experiments realized with these simple systems.

Chapter 4 is structured in two main parts, the first one is presenting a motivation for why the step of features selection is needed and the main possibilities to accomplish this task are given. Different features selection algorithms are presented, tested and evaluated in order to compute the most pertinent feature vector to encode the information from the image database. Our method to perform the features selection is described and the last part is presenting the experiments we realised in order to perform the selection of features by the mentioned methods. Possible improvements are studied and implemented in order to choose the best feature vector to encode the information provided by the visible and infrared cameras. Once obtained, this feature vector could improve the accuracy of the system, but also it could decrease the processing time needed for the system in the Obstacle Recognition stage.

In **Chapter 5**, three different fusion schemes are presented and evaluated having the main purpose the improvement of the recognition accuracy, but also the possibility to adapt the system to different context situations. Fusion is performed at different levels, low or high (by combining features, respective matching scores), but also at an intermediate level: fusion at the kernel level, which is the solution we propose for our final system. In this last chapter the monomodal systems are also brought in discussion, but the main processing is done with bimodal systems, thus combining both visible and infrared information. They use the bimodal information at different stages, depending on the applied fusion scheme. A comparative study of individual visual and infrared obstacle recognizers versus fusion-based systems is performed and the obtained results are presented and discussed.

In the last chapter, we draw the main conclusion about the proposed fusion schemes and several potential improvements of our work are given.

Table of Contents

List of Tables

List of Figures

"Everything, it said, was against the travellers, every obstacle imposed alike by man and by nature."

Around The World In Eighty Days, Jules Verne

CHAPTER 1

The Obstacle Detection and Recognition problem

Contents

Our purpose in this thesis is to design an Obstacle Detection and Recognition (ODR) system for improving the road security by directing the driver's attention towards situations which may become dangerous. Any system which could prevent deaths in traffic should definitely be tested because it could become a part of the future car.

There are some critical requirements that a driver assistance system should fulfil in order to be considered a possible solution to be implemented on board of a vehicle: the system cost should be low enough to allow to be incorporated in every series vehicle, the system has to be fast enough to detect and then recognize obstacles in real time, it has to be efficient (to detect all obstacles with very few false alarms) and robust (to be able to face different difficult environmental conditions).

Our proposed solution for developing an ODR system is to use two types of passive sensors: VISible (VIS) and InfraRed (IR) spectrum cameras. To improve the obstacle detection and the obstacle recognition tasks, we propose a fusion between the images provided by these types of sensors. A fusion between the information provided by VIS and IR cameras would solve difficult complementary situations, which any system based only on one type of camera could not solve by its own. Different fusion schemes using information provided by visible and infrared images are proposed for road obstacle classification and all these schemes are evaluated by an Support Vector Machine (SVM) classifier.

This chapter is intended to give a motivation for why the ODR task is an important area to be investigated, and how the work done in this thesis can contribute to an ODR system. It introduces the basic information necessary to understand the main characteristics and problems of the ODR task.

Section 1.1 gives the motivation for developing an ODR system in the frame of the intelligent vehicle field. Next, the basic problems connected with the ODR system implementation are presented in section 1.2 in order to understand why a problem so awfully simple for humans, tends to be very hard to be accomplished by machines. The fundamental requirements for developing an affordable-price, real-time, efficient and robust system to be deployed on board of the vehicle are presented in section 1.3, followed by some specific characteristics of the ODR systems in section 1.4. Finally, our proposed solution and how we intend to solve all the requirements specified for the ODR system are presented in section 1.5, followed by conclusion in section 1.6.

1

1.1   Why Obstacle Detection and Recognition task?

The interest for the intelligent vehicle field has been increased during the last years, must probably due to an important number of road accidents. According to a traffic report developed by the World Health Organization (WHO) (WHO, 2004), road accidents kill more than 1.2 million people annually, and around 50 million people are injured. If we could help to save even a part of those people by using a driver-assistance system, it would worth any effort. The easier way to act in such unwanted situations is their prevention. Many accidents could be avoided if a device attached to the vehicle would assist the driver with some warnings when dangerous situations are about to appear. These occur especially in urban traffic environments, but there are a lot of other situations: accidents can appear on rural roads and not involving only pedestrians. In addition, any situation on the road which can generate a human injury is a situation that such a driver-assistance system should detect and help to be avoided.

In recent years, leading car developers such as Daimler Chrysler, Volkswagen, BMW, Honda, Renault, Valeo, among others, have recorded significant efforts and support research works regarding the intelligent vehicle field. Their main aim is the protection of all traffic participants, from both inside and outside the vehicle, but mainly their efforts are concentrated on the pedestrian's safety. Together with these companies, many research groups approached the intelligent vehicle domain, proposing solutions for the existing problems, especially in the vision domain. Road detection and following, pedestrian or vehicle detection, recognition and tracking, night vision, among others are examples of applications which have been developed and improved recently. With all these, there are still a lot of challenges and unsolved problems in the intelligent vehicle domain.

One important example of such unsolved situation could be the development of an automatic pilot (for an autonomous vehicle) - which could therefore entirely control the vehicle at some time. For that, beside the obstacle detection step, an important step is its recognition: the system should be capable of recognizing the road trajectory, but also any possible obstacle which may appear outside, near or inside the road. Also, the system should be able to imagine a proper behaviour in different cases: when detecting an obstacle outside the road, the system should decide that it should not represent any danger, unless it will change its direction toward the road area; in the case of detecting an obstacle on the specific area of the road, the system should decide whether to change its trajectory to avoid the obstacle, or to stop and let the driver (if any) to decide how to continue the remaining road; when detecting an obstacle very closed to the road area, the system should recognize its type in order to know if it is a fixed obstacle (e.g., a tree, a rock, a landmark, a container), therefore an obstacle which is obviously not dangerous because it could not change its state, or it is a mobile obstacle (e.g., a pedestrian, a cyclist, an animal, a vehicle), therefore an obstacle which any time could change its trajectory and become dangerous. With this discrimination, a lot of accidents due to the suicides or animals jumping in front of the vehicle on night could be avoided. In any of these possible situations, when detecting an obstacle, it is recommended as prevention that the vehicle reduce its speed, until no obstacle is detected near the road area. Therefore, we conclude it is important not only to detect the obstacle, but also to identify its type. This would help the system in making the appropriate decision in many different situations. Another unsolved issue of the existing intelligent vehicles is their limited or non-existent ability to ensure a precise and robust functioning to different variations of poses, sizes, types, partial occlusions of obstacles in any illumination or weather conditions (poor light conditions or bad weather situation). This is what our system is intended to realize: to detect and then to recognize different types of road obstacles, even they are occluded or they present varying positions, shapes or sizes. These tasks have to be performed in any environmental context and in real time, therefore the system has to be precise and robust.

Next, we want to see which are the main existing impediments making so difficult the ODR task through the use of computers, while people perform it without too much efforts in normal driving conditions. Still, in difficult conditions of traffic (urban or crowded roads), accidents are happening even when a human is driving the vehicle.

1.2   What makes the ODR task so difficult to fulfil?

The outdoor environment, the variety of obstacles appearance from the road scene (considering also the multitude of possible types of obstacles), the cluttered background and the fact that the system must cope with the moving vehicle constraints make the categorization of road obstacles a real challenge.

In the recognition step, if a classifier is used to discriminate between different types of objects, some new impediments could also be found: the learning-testing step, so easy to be accomplished by humans that even children perform it in a natural way, could become a very difficult task for the machines (computers). The main difficulty is in trying to correctly learn a diversity of possible shapes: varying with the viewing angle, position and size, different outcomes which describe the same object among others. Since the obstacles we are looking for almost always are found in heavy cluttered urban environments, it is making the problem of separating them from the background a non-trivial one. Furthermore, the fact that the host vehicle is moving increases the complexity of the problem because the system must also consider the ego-motion matter and the real time constraints. All the processing has to be done with very little time consumption in order to assure enough time for the driver (or the autonomous vehicle) to make a decision and react.

Additionally, the illumination or weather conditions could also affect the well functioning of the system, because some different extreme conditions, like low visibility (e.g. night, overcast sky, dense fog, rain) or high visibility (e.g. the sunlight that fall on the windshield on a hot sunny day or the headlights of an approaching car on night) must be faced too. Hence, the developed system must be robust enough to adapt to different environmental conditions (such as sun, rain or fog) but also to their dynamic changes (such as transitions between sun and shadow, or the entrance or exit from a tunnel).

All the previously mentioned ideas also guided us to outline our system; therefore, the fusion between the information provided by visible and infrared cameras (which are appropriate for day, respective night vision) will be performed in such a way to allow the dynamic adaptation of the system to different environmental contexts.

1.3   What are the main requirements for developing an efficient ODR system?

There are some critical aspects that a driver assistance system should fulfil in order to become a viable solution to be implemented on-board of a vehicle:

R1. *The system cost*: should be low enough, since it has to be incorporated in every series vehicle. Therefore, some less expensive sensors must be employed.

R2. *The real time request*: the system has to be fast enough to detect and then recognize obstacles in real time, as an obstacle may quickly appear in front of the car and degenerate in an accident and such a situation is imperative to be avoided.

R3. *The efficiency of the system*: the system should detect all the obstacles from the scene but giving as little false alarms (ghost obstacles) as possible. Therefore, we are looking for systems which are capable to detect all obstacles but at the same time giving as few false alarms as possible. Later, in the verification step, false alarms are to be recognized and removed, but with each ghost detected as being obstacle, the processing time is increased and the accuracy of the entire system may drop; practically, the warning system will warn many times for no real reason. In addition, in the recognition step, for the detected obstacles the system must correctly identify the class they belong to.

R4. ***The robustness of the system***: the system must cope with difficult environmental context in which it has to assure a well functioning, no matter what are the illumination or weather conditions. The system should recognize different types of obstacles, like pedestrians, vehicles, animals, among others but it must also be capable of recognizing their variable shape, because e.g. pedestrians may ware different outfits, accessories, bicycles or (baby-)carriages, vehicles could be of different types: tourism cars, utility vehicles, trucks, buses and so on. Therefore, different classes of obstacles may vary with their size, shape, viewing angle, they could be occluded or not. All these types of obstacles transposed on the cluttered background and combined with different possible illumination and weather situations require fast and efficient obstacle detection and classification schemes.

How we treated these requests for our driver assistance system, to be inexpensive, fast, efficient and robust? We try to answer all these questions formulated here after we review which the specific characteristics of an ODR system are. Our proposed solution is mentioned in section 1.5.

1.4   Specific characteristics of an ODR system

Almost all categorization systems developed by now in the intelligent vehicle field employ an obstacle detection step followed by a recognition or a hypothesis verification module. Very often there is also a third module in which the recognized obstacles are tracked in their trajectory until they are no longer viewed in the scene.

The most developed systems are specific for one type of object detection, either pedestrian or vehicle. These dedicated systems are looking for obstacles in the scenes using either *an active sensor* like RAdio Detection And Ranging (radar) or Light Amplification by Stimulated Emission of Radiation scanner (laser scanner) which will provide the distance to the respective object, or *a passive one* like cameras. In this latter case, there are three main directions for searching possible obstacles in the scenes:

(1) to look for areas presenting symmetries or textures specific to the aimed obstacles,

(2) to use sliding windows encoding an obstacle specific shape of different sizes over the entire image or on some areas (determined from perspective constraints) from the image, or

(3) to try to detect specific parts of the proposed obstacles: pedestrian's legs or head, the wheels, headlights or the shadow produced by a vehicle.

In the verification step, these systems generally perform an it-is/it-is-not (the obstacle they were looking for) verification. This verification could be based on a classifier or on some *a priori* information specific to the obstacle to be checked (such as dimension constraints or ratio sizes).

When the Obstacle Detection (OD) task is limited to the localization of specific patterns corresponding to obstacles, processing can be based on the analysis of a single still image, in which relevant features are searched for. There are other systems in which a more general definition of the obstacle is exploited, therefore more complex algorithmic solutions must be handled. In this latter case of systems, all types of obstacles are searched at a time, but generally a road detection is performed in a previous step and all the obstacles are detected as being "on the road". Any object that obstructs the vehicle's driving path will be identified as an obstacle. In this case, the Obstacle Detection assignment is reduced to identifying the area in which the vehicle should safely move instead of recognizing specific patterns. Because here the discrimination between obstacles is more complex, for the identification of obstacle some multi-class classifiers are needed, not as the binary ones used for the previous type of systems. Generally, in the frame of these type of systems, the road detection is performed by a monocular camera, but the localization of possible obstacles on the vehicle path is realized through the use of some active sensors or by the analysis of some more complex vision-based techniques, such as employing two cameras instead of a single one or by using video sequences of images.

1.5 Our proposed solution for an ODR system

Almost all developed Obstacle Detection algorithms from the intelligent vehicle field are concentrated on detecting a specific type of obstacle: pedestrian or vehicle. Our purpose is to extend this processing to different types of obstacles, whether they are vehicles, pedestrians, cyclists, animals, or some others. A serious accident can occur even in a vehicle collision with a fixed obstacle, made by nature, and can endanger the life of the driver. Therefore, our intention is to develop an algorithm to detect and recognize any possible obstacle to the host vehicle and to decide its type, so to identify it.

Our work aims to recognize the detected obstacles, like pedestrians, cyclist, cars by the extraction of a compact and pertinent numeric signature, followed by an efficient classification of this signature. Both, the extraction of this signature (performed in the features extraction module, as we will see in Chapter 3) and the classification techniques should be fast and good enough to assure real time performances.

Our solution for developing an ODR system which fulfil the four requirements early mentioned, is to use two types of vision sensors: visible and infrared spectrum cameras. A fusion between these types of images would solve difficult complementary situations and will improve the obstacle detection and recognition tasks. Though, the main advantage of such a system is that it is not as much expensive as a system using radar or ladar, therefore the first requirement (*R1*) is reached. In fact, even if we add a second visible camera in order to obtain a stereo-vision schematic for the visible domain, it would not affect very much the price of the entire system, because we used low-priced and low-resolution cameras (320x280 pixels per image). The second request (*R2*) we solved by using a bi-level optimization technique which aims to improve both accuracy and computation time. We used feature selection algorithms in order to decrease the number of retained features which encode the information given as input to the classifier. In this way, the extraction time but also the classification time is very much reduced, assuring thus the possibility of obtaining a real-time recognition system. The third request (*R3*) we solved by using different decision-schemes which are based on SVM classifier. The high performance would be ensured by the fusion of visible and infrared domains, not only at the feature or decision level, but also at an intermediate level, i.e. at the kernel level. The system must work well even in difficult illumination or weather situations, like dark, rain or fog. By performing the fusion between visible and infrared domains the system will be capable of treating complementary situations and will ensure also the functioning in difficult environmental context; therefore, its robustness (*R4*) will be assured.

1.6 Conclusion

We briefly presented our solution to solve the ODR task. There are many unsolved problems in the intelligent system domain and our focus was to check if a fusion between the visible and infrared information would be helpful from the obstacle recognition point of view. Some key issues, such as the robustness to vehicle's movements and drifts in the camera's calibration must also be handled in the obstacle detection problem. Many systems still encounter problems at the detection step and since this detection part is still a work in progress in the frame of the Institut National des Sciences Appliquées (INSA) laboratory, our goal was to develop a system to continue this task. We have focused solely on the fusion between the visible and infrared fields from the viewpoint of an obstacle recognition module. Next, the recognition module (and hence the fusion) will be further integrated into a complete system performing both the Obstacle Detection and the Obstacle Recognition tasks.

In the following chapter, different types of sensors are investigated and their advantages and drawbacks are presented in the frame of the most cited systems developed in the intelligent vehicle domain.

Part I

State of the Art

CHAPTER 2

Sensors and Systems in the Intelligent Vehicle field

Contents

In this chapter, the main sensor types and their characteristics are detailed. Common sensors used in the intelligent vehicle field are examined, but we concentrated especially on the information each type of sensor could provide. Some sensors may have many advantages, but also some strong limitations, which will make them to be not-so-properly for the implementation of an ODR system. Many systems developed in the intelligent vehicle field and employing one or multiple sensors are reviewed and their performances and drawbacks are illustrated.

To outline the system we were looking for sensors which could provide enough information to detect obstacles (even those occluded) in any illumination or weather situation, to recognize them and to identify their position in the scene. In the intelligent vehicle domain there is no such a perfect sensor to handle all these concerned tasks, but there are systems employing one or many different sensors in order to perform obstacles detection, recognition or tracking or some combination of them. Before presenting different systems from the intelligent vehicle domain (section 2.2), first we introduce the main types of sensors (in section 2.1) which could be employed single or combined in order to accomplish the ODR task.

The sensors employed in the systems presented in this chapter are grouped in two main categories: passive and active. The present chapter is mainly focused on comparing advantages and disadvantages between these two types of technologies and choosing the best solution for developing an obstacle detection and recognition system.

2.1   What type of sensor to choose ?

Generally, the sensors used in the obstacle detection and recognition (ODR) field can be classified according to different criteria. First, they can be classified by the perception about the environment, which means according to the type of measured information they give as output, as *proprioceptive* and *exteroceptive*. Secondary, the sensors can be classified about the *spectrum* position of the radiation they use to function and third, they can be classified as *active* or *passive* sensors and this latter classification is referring to the presence or absence of a radiation needed also in their functioning.

2.1.1   Proprioceptive vs exteroceptive sensors

The *proprioceptive* sensors are capable to measure an attribute regarding their own state, while the *exteroceptive* sensors are capable to measure an attribute of an external object present in the scene (Appin Knowledge Solutions, 2007).

The *proprioceptive sensors* perceive the position, the orientation and the speed of the object they are mounted on. They monitor the motion of the vehicle by measuring the kinetic quantity like acceleration and velocity. Vehicles use inertial sensors like: *accelerometers* to measure the acceleration from which velocity can be calculated by integration, *tilt sensors* to measure inclination, *position sensors* that indicate the actual position of the vehicle from which the velocity can be calculated by derivation, *odometers* to indicate the distance traveled by the vehicle, *speed sensors* which measure the velocity, among others.

The *exteroceptive sensors* give information about the surrounding environment and that information allows the vehicle to interact with the scene to which it belongs to. *Cameras* working in visible or infrared spectrum, *laser scanner*, *microwave radars* and *SOund Navigation And Rangings (sonars)* are used to provide a representation of the environment (e.g. some imaging features or the distance to the respective obstacles). Proximity sensors (sonar, radar, laser range finders or tactile sensors like bump sensors) are used to measure the relative distance (or the range) between the sensor and the objects from the environment. Also, to measure the distance, two stereo cameras or the projection of a pattern on the environment followed by observations on how the pattern is distorted (markers) can be used. In order to recognize and classify objects in the obstacle detection and classification context, Charged Coupled Device (CCD) or Complementary Metal Oxide Semiconductor (CMOS) visible spectrum cameras or infrared spectrum cameras can be used as vision sensors.

External bumpers and similar sensors have been tested in the intelligent vehicle domain, but such sensors will rather be used as reactive ones (e.g. by inflating an airbag) or as some additional sensors to other more powerful ones (like *radars*, *laser scanners* or *cameras*).

2.1.2   Sensors classified about the radiation position in the electromagnetic spectrum

In order to remind where the radiation these sensors employed to function is positioned in the electromagnetic spectrum, in the following, we recall how is the entire electromagnetic spectrum[1] typically divided into its bands (Hammoud, 2009a), (B.S. Shivaram, 2010). At the very short wavelengths, there are the *gamma rays* which are produced in nuclear reactions and they could actively interact with the matters' molecules and atoms they propagate through. Then, it follows the *X-rays* which have lower energies (i.e., longer wavelengths) than the gamma rays and due to their excellent penetration ability, they has been extensively exploited in medical imaging; the *ultraviolet* spectrum consists of electromagnetic waves with frequencies higher (i.e., wavelengths shorter) than those used by humans to identify the violet colour. The *visible spectrum* is the electromagnetic radiation from a particular, very narrow frequency band (i.e., wavelengths from about 390 to 750

---

[1]The main regions of the electromagnetic spectrum are specified according to their wavelengths, from the shortest towards the longest.

nm, which corresponds to the extreme colours, violet and red), also called "the optical spectrum" due to the fact that humans are able to see it in different colours. Next, just beyond the colour red in the visible spectrum, i.e. with a wavelength longer than red colour, the *infrared spectrum* follows. It represents the electromagnetic radiation with a wavelength between 0.75 and 300 micrometers. Its name means "below red" (in terms of frequencies) and it is also referring to the visible light range. *Microwaves* and *millimeter waves* represent the band with the wavelengths from 0.3 mm to 3 cm. Radar uses microwave radiation to detect the range, the speed, and other characteristics of the remote objects. Finally, at the upper bound of the electromagnetic spectrum there are the *radio waves* which are mainly used for applications like: fixed and mobile radio communication, broadcasting, radar and other navigation systems, satellite communication, computer networks and others.

2.1.3   Active vs passive sensors

In the frame of intelligent vehicle applications, to record information about the obstacles presence on a road, one of the sensors described in the following is often used. They are grouped in two main categories: **active sensors** (like radar, *laser scanner* or LAser Detection And Ranging (ladar) or Light-Imaging Detection And Ranging (lidar) and sonar) and **passive sensors** (cameras using visible or infrared spectrum radiation).

Active sensors - emit a signal and receive a distorted copy of the respective signal. They provide their own energy for illumination and the objects are irradiated with artificially generated energy sources. That energy is reflected by the target objects providing information about the surrounding environment. In the following lines, a brief description of the most used active sensors in the intelligent vehicle domain is given:

a) **radar** (Microwave Radar) - emits microwave radiation in a series of pulses from an antenna and receives that part of the energy which is reflected back from the target. The time required for the energy to travel to the target and return back to the sensor determines the distance or range to the target (distance and speed of the objects are determined from multiple emissions of this type). A two-dimensional image of the surface can be produced by recording the range and magnitude of the energy reflected from all targets from the scene. One characteristic of radar is its possibility to be used day or night because it provides its own energy source to register the scene. Therefore, radar technology can operate in different environmental conditions (e.g. rain, snow, poor visibility) without any strong limitations. Still, this characteristic could be seen as a weak one, due to the interference problems it implies when radars are mounted on many vehicles in traffic. An important advantage of radar is that it can be used for long range target detection. There are three primary types of radars based on their transmitted electromagnetic wave form, but in order to offer more information using a single system, these technologies are often integrated together (Chan & Bu, 2005): (a1) Doppler radar - transmits a continuous wave of constant frequency and the speed of the moving obstacles is determined using a frequency shifting (static objects can not be detected using only Doppler radar). (a2) Microwave radars - transmit frequency-modulated or phase-modulated signals and determine the distance to the target by the time delay of the returned signal. (a3) Ultra Wide Band radar transmits and receives an extremely short duration burst of radio frequency, being able to detect and register the object's motion with centimeter precision. An important property of radar is that it offers good accuracy in longitudinal distance measurement; this compensates the poor accuracy in the lateral distance measurement.

In the intelligent vehicle domain, two types of radars are often used: near range radar (for detecting targets up to 30 m) which emits at 24 GHz and long distance radar which is emitting in the 77 GHz for distances up to about 150 m. Radar systems offer the capability to accurately measure target range, relative velocity, and azimuth angle of one or more object(s) inside their observation area (they are often based on some vehicles information like velocity and steering). The really challenging task when using radars is to distinguish between different object classes based on the received target

signals (echoes). Most often, radar measures point targets of objects like cars, trees, traffic signs, bicycles and human beings which present some reflections on their surface. Metallic parts of vehicles present a higher reflectivity compared to humans, or trees (in terms of reflectivity trees are the poorest obstacles due to the wood state of nature) and sometimes strong reflections could minimize other poorer reflections and lead to fails in detecting all obstacles from the scene. The separation between different types of obstacles is generally done by the evaluation of typical object-specific reflection characteristics from the radar signals (e.g. the reflected power, the power variance over time, the dimensions and dynamics of the obstacle). Thus, after the signals reflected by the targets are analyzed, the next step is to generate some features from these echo signals, which should have sufficient discriminant information to characterize different targets.

b) **lidar** (ladar, laser radar or *laser scanner*) - it uses a Light Amplification by Stimulated Emission of Radiation (laser) to transmit a light pulse and a receiver to measure the reflected light. A laser scanner emits laser pulses and detects the reflected ones. It uses optical or infrared technology to detect reflections from objects and the object's distance is determined by recording the time between transmitted and received pulses; the traveled distance is calculated using the speed of light. Laser scanners are used for long range target detection (in general up to 40 meters) and they use large field of view. These types of active sensors provide precise measurement of depth and they have a high accuracy both in lateral and longitudinal direction (Bu & Chan, 2005), (Chan & Bu, 2005). Therefore, to process the data delivered by these sensors, some procedures similar to image processing are applied (like edge detection, segmentation, clustering and tracking).

The laser scanners used in the intelligent vehicle domain generally emit pulses in the infrared spectrum, therefore they work independently of daylight. The range of velocity, the typical appearance of the respective object and the object information from the past are parameters which help in the classification step.

The ODR task is generally performed based on some information: the model of the sensing devices, the model of the street the vehicle drives on, the dynamic model of the ego vehicle and the cluster containing dynamic models of all objects to be identified. Different types of obstacles like pedestrians, cyclists, cars, trucks, busses, trees, crash barriers, motorcycles, bicycles and others could be discriminated based on the information obtained after data points coming from laser scanner are clustered (or grouped) through the segmentation method into different objects. The segmentation process is started by grouping all the measures of a scan, into several clusters often called segments, according to the distances between consecutive measures. Objects classification is performed by comparing the segment parameters (e.g. left, right and closest point to the sensor, the geometrical center of gravity of all measurement points of the segment) of the current scan with the predicted parameters of known objects from the former scan(s). Partial occluded objects could be detected by the use of the laser scanner (if they have been detected in a previous scan). Also, small objects could not be very well detected due to the limited number of scanning points provided by the scans. Generally, a Kalman filter is used to predict the object state, including the calculation of the longitudinal and lateral velocity of the object. The relative velocity of the object can be combined with the motion of the vehicle in order to determine the objects' absolute velocity. The detection of moving objects could be improved by incorporating additional knowledge, like the dynamic behavior of the objects (given by the tracking algorithm).

c) **sonar** (Ultrasonic sensor)- works very similar to radar, but instead of electromagnetic microwave, sound waves are transmitted from an antenna. Ultrasonic sensors generate ultrasonic waves (short wavelength, high frequency - in general outside the audible frequency) which are reflected by the targets and by analyzing the received signal, objects are detected together with their distance and speed. Time of flight is calculated in a similar way to detect the range to object, which is up to 20 m. One important disadvantage of sonar refers to its sensibility to weather conditions changing, because the speed of sound waves varies with the temperature and the pressure of the surroundings.

The main advantages of using active sensors like radars and laser scanners are their possibility to measure distance and speed of the targets and the fact that they work well also in bad weather or poor illumination conditions. Still, other issues remain: the interference problems, the difficulties in interpreting the output signal returned by these sensors and the acquisition price which is very high compared to that of a visible spectrum camera, for example.

Passive sensors - they just receive a signal, which could be the reflected, emitted or transmitted electro-magnetic radiation (light or temperature) provided by natural energy sources. Thus, they acquire data in a nonintrusive way. They are also called "vision sensors", because they are working similarly to human eyes: they refer to the processing of data using the electromagnetic spectrum and produce an image. The energy provided by sun could be reflected (the visible case), or absorbed and then reemitted (the infrared case). Thus, the cameras working in the visible spectrum are suitable just for daytime, when the natural light is available. In contrast, the infrared cameras could work both on day or night, as long as the amount of energy is big enough to be recorded by the sensor's receiver.

a) Images captured by colour or gray scale cameras, working in the **visible spectrum**, are very rich in content and easy to be interpreted by a person. Maybe this is the reason why the most attention in research for the Obstacle Detection task was focused in this direction. The visible spectrum has the main advantage comparing with the infrared one that it has been sufficiently well studied and understood; therefore, with the advancements of the technology, the cameras working in the visible spectrum are becoming standard and cheaper. In addition, there is a diversity in algorithms and applications for obstacle detection and tracking by using images taken with visible spectrum cameras. Still, the main disadvantage is that they have limitations due to the lighting conditions and possible shadings; therefore, they are not well suited for darkness conditions.

b) In the last decade, a variety of medical, industrial, military, and remote-sensing applications have employed the **infrared spectrum**. The main directions these applications are straightened will be pointed out after we mention how the infrared spectrum is structured. The infrared band is typically divided into multiple sub-bands but their separation is not very well defined and therefore, their associated boundaries could overlap in different literature sources (Hammoud, 2009a), (Arnell, 2005), (Global Security, 2010):
    - Near Wavelength InfraRed (NWIR) region - the wavelengths between 0.75-2.4 μm,
    - Short Wavelength InfraRed (SWIR) region - the wavelengths between 0.9-3 μm,
    - Medium Wavelength InfraRed (MWIR) region - the wavelengths between 3-5 μm, [2]
    - Long Wavelength InfraRed (LWIR) region - the wavelengths between 8-14 μm,
    - Very Long Wavelength InfraRed (VLWIR) region - the wavelengths between 14-300 μm.

Two main technologies are used for night-vision surveillance applications and these are image enhancement and thermal imaging, corresponding to the so called "reflective" and "thermal" bands (Hammoud, 2009a). The key difference between imaging in reflective and thermal infrared bands is that the first mentioned retains the information reflected by objects, while the latter records temperature emitted by objects (thermal energy is emitted from objects as heat, it is not reflected by them as it is the case of light).

The basic concept behind image enhancement is to amplify the visible light in order to enhance visibility. Cameras working in the NWIR or SWIR regions, record the reflected energy from illuminated objects on a scene; they register a similar content like visible spectrum cameras due to the fact that their wavelengths are very closed to the visible one. Cameras working in the range of 0.75-5 μm are very often used for night-vision application, due to their capability to enhance the perception of the scene when strong darkness exists. On night there is no sufficient visible light to see

---

[2]The 5-8 μm band is rarely used for imaging due to the fact that it is blocked by spectral absorption of the atmosphere, so there are very few or none cameras using this range (FLIR Technical Note, 2008).

the obstacles inside the observation area because there is no natural light to illuminate the scene like during daytime. Therefore, some sort of artificial illumination sources (like infrared lasers, filtered incandescent lamps, Light Emitting Diode (LED) type illuminators) should be applied in order to increase the contrast of some possible objects from the scene when imaging in NWIR or SWIR regions. Because these night-vision devices need some infrared illuminators to highlight the scene at infrared wavelengths, they are often called "active infrared sensors".

Thermal imaging on the other side refers to the process of capturing the heat from the scene, and transforming it into an image that can be viewed with thermal infrared cameras. Thermal energy is the upper part of the electromagnetic spectrum in terms of wavelengths. Both MWIR and LWIR are good candidates for applications involving thermal radiation. Cameras working in thermal IR band are also called "passive infrared sensors" because they need no artificial illumination sources to function. Therefore, they capture infrared radiation given by objects as heat.In addition, a thermal imaging camera can function optimally no matter what the surrounding lighting conditions are.

Cameras operating in the near or short IR domain involve the conversion of ambient light photons into electrons which are then amplified by a chemical and electrical process and then converted back into visible light (American Technologies Network (ATN) Corporation, 2010). In this manner, they deliver much more details of the registered scene (they preserve almost the same details as visible spectrum cameras on daytime) than offer their thermal counterparts. On the other side, the passive infrared sensors do not send signals as active sensors do, they just wait until the infrared energy from an object is received by the detector and then they measured it (Global Security, 2010). In addition, as we traverse from shorter to longer wavelengths, the radiation become less susceptible to ambient illumination or hot objects like sun, bulbs, fire, and so on. Most often, this type of infrared sensor is used in the obstacle detection field, in order to avoid any possible infrared radiation (no active illumination source to be used). Therefore, in the following, when we refer to infrared cameras, we denote those infrared cameras working in the thermal band.

There are two types of thermal imaging cameras: *un-cooled* and *cryogenically cooled*, but mostly the un-cooled ones are used in different applications. The un-cooled camera functions without an additional cooling unit attached, which besides producing a much clearer image (with much details), it is much more sensitive to temperature variations and also it is much more expensive than a similar camera without the cooling unit attached.

The IR spectrum is proper for object detection and tracking because all heated objects emit IR radiation that can be registered with an IR thermal camera. Since some classes of objects like pedestrians and vehicles have a specific IR signature: pedestrian's head, body and legs and vehicle's wheel and engine, the object identification could be made based on the received energy. But there are a lot of heats radiators distinct from human or vehicles, so IR sensors can be used for obstacle detection in general. The higher a body temperature, the more radiation (heat) is emitted.

The important advantage of the infrared spectrum is its ability to measure the temperature. Because infrared sensors are independent from the light source (being passive sensors) they can register the same or almost the same image even it is day or night. They can produce a clear high-contrast image of the objects from the scene even in total darkness or very strong illumination conditions.

The distance (the range) one can see with a thermal imaging camera is highly dependent of different variables. First of all, the most important ones are the cameras' functioning parameters, like the waveband in which the camera operates, if the camera is equipped with a cooled unit or not and the properties of the target, such as size (i.e., the larger the object the easier it is to see), temperature difference between the concerned object and the background (i.e., objects are better detected in winter than in summer). Second, the atmospherical conditions are also of great importance: one can always see further on a clear night with objects illuminated by the moon, than if it is cloudy and overcast, or it is dense fog or rain.

According to the International Civil Aviation Organization (ICAO), fog can be classified in four categories I, II, IIIa and IIIc denoting the visual range of 1220, 610, 305 and 92 meters. Different studies (Beier & Gemperlein, 2004), (FLIR Technical Note, 2008) showed that some types of fog are not critical for some thermal cameras: i.e., only LWIR cameras are superior to the visible ones in conditions of fog of both types I and II, while MWIR cameras could operate well only for a fog of type I. Still, extreme atmospherical conditions, like dense fog or rain are not very well handled even with LWIR band cameras.

The main issues of the infrared imaging are the sensitivity of the IR sensors to weather conditions (i.e., in dense rain or dense fog) and the fact that the IR domain is not as well understood as the visible one; thus, very few approaches have been developed to process IR images. As improvements are made to the IR technology, these camera devices became cheaper and practical for night-vision applications. They can easy "see" objects in night, light fog or rain while visible cameras could not face very well such scenarios. The IR cameras price is decreasing with the advancements of the IR technology, thus they are becoming an attractive complement for the VIS cameras, not only for night-vision, but also for daytime functioning, in different environmental conditions, but not the extremes ones in which neither technology can handle very well.

Different cameras configurations

An inconvenient of cameras comparing to active sensors is that they can not provide directly (as radar or laser scanner does) either the distance of the object in the scene nor its velocity. In order to obtain the distance information, different cameras configurations could be employed:

- The systems relying on *monocular vision* generally provide the objects distance based on the calibration information of the moving vehicle and they often assume the road is flat. Thus, they are not viable because this hypothesis is not always verified, especially due to the vertical vibrations of the host vehicle, which could not be neglected.

- An often used possibility is that instead of employing a monocular camera configuration, to exploit two cameras of the same type; in this manner, stereo visible or infrared images could be acquired. *Stereo vision* is an effective technique to extract 3D information from 2D images, information that can be applicable to visual guidance. Two major issues are inherently involved in most of the conventional stereo vision methods: depth search and cameras calibration. The depth search (also called stereo matching) is a process to find corresponding points between two related images (a pair of images). This would require an increase of the computational cost. The camera calibration on the other hand is a procedure to precisely determine camera parameters including the camera 3D position and orientation with a calibration target whose 3D shape is known. Stereo images help in establishing the region of interest (ROI) in an image, so stereo technique can be seen as an obstacle detection or segmentation method. A vision system with multiple perspectives, which means a stereo-vision system, will provide depth information.

- As active sensors provide the temporal information by scanning the area in front of the vehicle at different time intervals, in the same manner, cameras could be employed to acquire images at different moments in time in order to yield also a temporal information. By using sequences of images (video) instead of recording a single static image, *a video image or sequences of images* (single camera systems used at different time instances) will provide information about the changing appearance of the object, i.e. the motion information. Thus, the video camera is a system that can yield very rich information about the scene. Although a video camera can obtain much information about the environment compared with radar or laser scanner, the image sequences can not be used for anything directly without further interpretation. From

radar and laser scanner, motion information is extracted directly, while from video camera some image processing operations are required. From all these systems, using motion information the moving objects could be extracted, together with some other important parameters, such as distance between the camera and the respective objects and objects' speed and direction.

Based on both depth and motion information, the processing of the whole image can be replaced by the analysis of some specific ROIs only, in which the aimed objects are more likely to be found. Using a perspective translation in space respective in time, the resulted depth or motion information can be used to detect distance or time to collision of the object. If the longitudinal position of an object is known (or estimated), the search space can be drastically reduced. This will increase the detection rates and the speed of the processing system, because having the distance to an object and knowing the motion parameters, the relative velocity of the respective object can be calculated. In addition, knowing the lateral position of an object, a tracker can be initialized to follow the objects on their trajectory.

Although at this point we have mentioned the main advantages and disadvantages of all the available sensors and our proposed solution could be anticipated based on this information, in the following we present the most mentioned systems from the intelligent vehicle domain, together with their main performances and issues.

2.2   What type of system is better ?

As we earlier describe, each type of sensor has its advantages and limitations. Generally, the developed systems tried to exploit their complementarity in order to assure an improved reliability.

There are a lot of research groups working in this area, therefore different Obstacle Detection or/and Obstacle Recognition systems have been considered. First, in order to explicitly present systems which accomplished the obstacles detection, the obstacles recognition or better, both mentioned tasks, we first concentrated on different systems employing both types of technologies, active and passive (in section 2.2.1). After presenting these powerful but very expensive systems, we focused on some less expensive ones, i.e. systems employing one single technology: active (section 2.2.2) or passive (section 2.2.3). Different systems could be followed here, i.e. systems employing a single type of active sensor (section 2.2.2.1), a combination of different active sensors (section 2.2.2.2), a single type of passive sensors (in 2.2.3.1) or a combination of different passive sensors (section 2.2.3.2). For each type of system, we tried to verify how they fulfil the four main requirements stated in Chapter 1 at section 1.3. All these systems mentioned in the section 2.2 are grouped at the end of the section in two tables (systems employing active sensors and systems employing only passive sensors) in order to be better identified as belonging to one or another category of systems.

2.2.1   Systems combining active and passive sensors

There are a lot of systems using a combination between passive and active technologies, therefore using a vision sensor combined with a distance one. They may use active sensors as radar, lidar (or ladar) or laser scanner in order to perform or improve the detection step. This choice is to be expected when considering that active sensors are distance providers and they could properly function in poor illumination or bad weather conditions. Generally, for the active-passive fusion systems, in the detection step the obstacle's position is estimated by the active sensor and in the recognition step these positions are marked on the image provided by the passive sensor. However, there are systems exploiting a fusion of both passive and active sensors in order to improve the detection step. After the obstacle is segmented, possible area containing the obstacle and called bounding box (BB) is found. Then, it is verified and processed in the recognition step, where the false alarms are discarded and for

each recognized obstacle its type is also determined.

In the following, systems using a combination between active sensors and cameras (functioning in the VIS or in the IR spectrum) are presented. First, we review some systems employing radars and cameras or laser scanners and cameras, and in the last part of this section some very powerful systems employing multiple types of active and passive sensors, which we called "all fused" are presented.

2.2.1.1 Systems combining radars and passive sensors

Radar measurements not always coincide with the center axes of the host car as it is assumed (because radar could detect some metallic parts of the same obstacle or some other obstacles) and in order to increase the reliability of the system, often a fusion between radar and vision is chosen. Radar offers good accuracy in longitudinal distance measurement, but poor accuracy in the lateral one and vision has opposite properties. By combining vision with radar, good accuracies in both positional measurements will be provided. Both radar and monocular vision measures 2D information from the 3D scene: radar measures velocity or distance and radar cross section (RCS), while camera retains the objects' width and height. The information coming from radar and vision complement each other: whereas radar is able to tell at what distance it points echoes, but without providing the direction, vision can give the direction in which a relevant event is detected, but do not provide the distance at which it occurs. By fusing radar and vision the accuracy of the system is expected to be increased.

In the following, systems employing combinations of **radar(s) and visible spectrum camera(s)** (monocular or stereo) are presented.

The radar employed by Handmann et al. (Handmann *et al.*, 1998) is capable of detecting up to three objects in front of the car and also it has the ability to track them. Besides the information provided by a monocular CCD visible spectrum camera and a radar sensor, Handmann employed some additional features (like feedback over the time, local variance and vehicle shadow) in order to perform the sensor fusion process for vehicles detection. A feature vector was composed by different information and then it was given as input to a binary classifier to decide the membership of each pixel to a relevant segment or to the background.

In the approach of the researchers from Daimler Chrysler (Gern *et al.*, 2000) the run of the curve is estimated by a Kalman filter technique from the position of the leading vehicles, which have been detected by an Adaptive Cruise Control (ACC) radar. The approach consists of three steps: first, obstacles are detected by radar, then they are located in the image and finally they are tracked in the image. Their vision-radar fused system is intended for highways road following, since highways are built under the constraint of slowly changing curvatures. The run of the curve and vehicle position parameters were determined from the relation between a point on a marking and its image point, which was estimated assuming the pinhole-camera model and knowing the camera parameters, like focal length, tilt angle and height-over-ground. Gern et al. employed a vision system to estimate the road, but also to perform a visual symmetry detection for correcting the typically bad lateral accuracy of radar targets; the longitudinal distance was provided by radar. The vision system was composed by one monocular camera in a previous version of the implementation (Franke, 1992) and by two stereo cameras in the current version. Gern et al. do not use all the information provided by the vision camera (i.e., vehicles on the lateral lanes), like the systems presented in (Handmann *et al.*, 1998) and (Steux *et al.*, 2002) do. Under good weather conditions, the system proposed in (Gern *et al.*, 2000) analyzes up to 150 search windows at a range of sight of 50 m to 70 m. The monocular system runs at a cycle time of about 5.5 ms, while the binocular takes about l0 ms time for every cycle. Being addressed to highways, their system allows driving comfortable autonomously with a speed up to 160 km/h if the markings were well visible. Under good weather conditions, the range of sight was about 50 m while under bad visibility, the range of sight was decreased to about 10-14 mand the distance to the leading cars was about 60-80 meters.

A pedestrian detection system from a moving camera is proposed in (Milch & Behrens, 2001). Milch and Behrens used fusion of a radar system (consisting of two individual 24 GHz radar sensors with slightly overlapping field of view) and a monocular vision system, together with some vehicle information like velocity and steering. The fusion is based on a two-step approach for object detection: first, a list of potential targets (i.e., hypotheses for the presence of pedestrians) was generated using radar; for every object distance, angle and RCS information was extracted. Then, a filtering operation using specific constraints about speed, RCS and size was performed. In the second step, the remained pedestrian candidates were verified in the image-processing module, where a flexible 2D prior model of the pedestrian shape (trained from manually extracted pedestrian-instances with a frontal view and a side view of the human body provided by video sequences) was used.

In (Steux et al., 2002) another fusion scheme of radar and a colour camera for vehicle detection is presented, radar informing the vision system about the position of the targets, so that the ROIs in images could be brought in attention. Unlike the most approaches found in literature, where the vision module directly yields some obstacle-like information, in their Advanced Functions for Environment Detection (FADE) system, four independent vision-based detection modules (using shadow, rear lights, lines and symmetry) provided low-level information on possible targets (e.g., the position of the left shadow of the target) and the fusion module decided the real position of the target. A causal structure (belief network) for each target at each step of the fusion process provided a set of hypothesis regarding the position of the target in the ground plane.

Kato et al. (Kato et al., 2002) proposed an obstacle detection method based on the fusion of information provided by radar (a Fujitsu-Ten mechanical-scan millimeter-wave radar, with a detection range of 100 m and a horizontally scanning angle of 14 degree) and a video camera (an XC-7500 camera with a sampling resolution of the image of 640x480 pixels and the frame rate of 30 frames/s). The regions corresponding to objects detected by radar were transposed in the image, by assuming a virtual vertical plane at the distance measured by the radar. Based on this assumption, changes of motion in the image were estimated by a motion stereo technique instead of estimating the distance from them. The feature points, whose motion was easy to track, were selected and tracked frame by frame in the image sequence. Their system was able to detect obstacles (vehicles but also pedestrians) up to 50 m on an urban road.

In (Sole et al., 2004) radar and vision sensors were used for a high-level fusion: a list of radar targets and a list of vision detected vehicles were provided for the fusion process. A radar target that matches a vision target was considered a valid target. The validation strategy was based on a high-level sensor modality approach which assumed that each sensor had a capacity to form independent target acquisition: matched targets were automatically validated. On the other side, the validation process for unmatched targets was divided into a number of steps combining motion analysis with shape and texture analysis to classify it as vehicle (moving or stationary), motorcycles, pedestrians (moving longitudinally or laterally, or stationary). The matching process was based on objects parameters like range, angular position, range-rate and considered such information across multiple images. Based on the radar reflections, targets were validates as "Solid" or "Ghost" and for Solid targets the system performed also object classification in vehicle (moving or stationary) or motorcycles. Their system could also find pedestrians (moving longitudinally or laterally, or stationary) by detecting non-reflecting or weak reflecting radar targets.

In (Kawasaki & Kiencke, 2004) a vehicle detection and tracking method using fusion (at the sensor level) of a millimeter wave (MMW) radar (working at 77 GHz) and a vision video camera, by a causal model, i.e. a Bayesian Network, is presented. The inputs to the Bayesian Network were different target specific characteristics such as lateral position estimated by symmetry detection, the width estimated by vertical edges computation, the center position or width estimated from a shadow detection or a tail lamp detection, the center position of the object estimated from the tracking function, environmental brightness from the blue sky estimation, the lateral and longitudinal position,

and longitudinal relative velocity of the object provided by radar and the predicted object position and width. Each detection algorithm outputed a relevance measure: e.g., the symmetry algorithm delivered a measure of "how symmetrical the image was." Kumon et al. (Kumon *et al.*, 2005) continued the work of Kawasaki and Kiencke and added a rear projection area detection function. Target vehicle image was clipped out from the whole image using object position data provided by radar in order to reduce the computational time for the image processing module.

A system that besides the radar and monocular information integrated also some temporal data provided by the video camera is proposed in (Alefs *et al.*, 2005). Their system combined the output of a 24 GHz radar (detecting obstacles up to 20 m), single images provided by a monocular camera and the image sequence data available from the same camera. The radar detection module used condensation tracking which provided a model based approach for contour tracking. In the vision system, vehicles were detected by scaled symmetry detection. For the vehicle detection, the three modules were fused by sharing sets of hypotheses. The internal state of the system was described by two variables: a probabilistic function, describing the occurrence of an object as function of the distance (the range data were fused with 2D-position data from single images) and a set of hypotheses consisting of deterministic parameters for the position and motion for vehicle candidates (spatial object coordinates were fused with data from the image sequence). The result was a kinematic description of the obstacle, including coordinates for the position, size and the object's velocity. Detected vehicles were tracked using Lucas-Kanade template matching, resulting in additional hypotheses and including vehicles beyond the range of the radar sensor. Their results show 96% reduction of radar phantoms by fusion between range sensing and vehicle detection modules, and a 63% increase of correct detections by fusing vehicle detection and tracking modules.

In (Schweiger *et al.*, 2005) a particle filter implementation for fusing different sensor characteristics coming from an ACC radar and a monocular visible spectrum camera is presented. Under the flat world assumption, the particles were initially equally distributed in the state space. Using three features (i.e. radar information, symmetry detection and tail lamp detection), the particle filter was focused on the leading obstacle (in a distance of up to 50 m) within 3 to 5 frames. Even their system employs one VIS camera and one radar, it was designed for in front vehicle detection on night-time.

In a more recent paper (Serfling *et al.*, 2008), Serfling, Schweiger and Ritter contributes to a road course estimation system also designed for night driving, like the system from (Schweiger *et al.*, 2005), by adding a digital map and considering also a particle-filter fusion scheme between radar and visible spectrum camera. To adapt the VIS camera for night driving, they considered information from the right road border only (gradient and orientation), which generally it is assumed to be more homogeneously illuminated than the left one. Their system was provided with Global Positioning System (GPS) on a digital map (asserting a course road estimation at 120 meters), a spline based representation was calculated using the transmitted map shape points and then the GPS position was refined in value and time using inertial vehicle data. Next, the current position of the vehicle was matched to the nearest spline points and these spline points were transformed into each sensor plane (camera and radar). Finally, the particle filter was applied to correct the transformed splines in position and orientation. By using a fusion between radar and camera, the performance of the system was increased with 25% compared to the system using only the vision module.

Bombini et al. (Bombini *et al.*, 2006) proposed a vehicle detection system fusing data from a 77 GHz radar and a vision system (a gray-scale visible spectrum camera). Radar data were used to locate areas of interest on images and for each of these ROIs a vertical symmetry has been computed. Their algorithm analyzes images on a frame-by-frame basis without any temporal correlation. To benefit from both the radar precision on distance measurement and the vision refinement ability, radar was used to render distance while vision provided position and width of the obstacle. In a latter paper, that of Alessandretti et al. (Alessandretti *et al.*, 2007), the system has been improved and tested with two different configurations having two different types of radars. The first configuration used a radar for long range detection (frequency at 77 GHz) which could detect multiple non-vehicle objects and for

eliminating these false alarms, guard rail detection and a method to manage overlapping areas were employed. The second configuration used two radars with a frequency of 24 GHz which could detect with a precision of 0.24 m only obstacles relying at 40 m in front of the vehicle. Radar computed the relative speed and the position of the object. The authors assert their system provided good results on rural roads and on highways.

The system presented in (Richter *et al.*, 2008) fuses radar, image and ego vehicle odometry data to provide a list of objects, which were tracked in the ego vehicle coordinate system. The parameters of the used sensors were: the gray scale camera has a resolution of $640x480$ pixels, a horizontal opening angle of 22.5 degrees and an update rate of 30 Hz, while the long range 77 GHz radar has an opening angle of 15 degrees and an update rate of 10 Hz. Richter et al. described a multi-sensor approach for vehicle tracking, by fusing at high level radar observations and the results of a contour-based image processing algorithm (which were used not only to verify radar observations, but also to estimate additional properties of the targets, i.e. width and lateral position). The states of the tracked vehicles were estimated by a multi-object Unscented Kalman Filter. The vehicle image detection algorithm presented was based on contour chains, which were used to find U-shape like forms, which are typical for almost every type of vehicle. If a U-shape was confirmed, a rectangle image observation was created and incorporated into the filter. Their system was able to fuse data from radar and image sensor to estimate the position, direction and width of objects in front of the ego vehicle.

Because IR images are almost like visible spectrum images regarding the lateral measurements, IR cameras provide good resolution images (thermal map) of the road scenario. Thus, a fusion between radar and IR camera(s) will lead to good results for the OD problem. The main advantage of using an IR camera instead of a VIS spectrum one is that the segmentation problem is simplified. Some papers in which the sensors fusion by **IR camera(s) and radar(s)** is suggested are next presented.

The proposed integrated driver assistance system (part of EUCLIDE project) from (Andreone *et al.*, 2002), merge the functionality of radar and IR camera to support the driver in difficult situations. After the implementation of the proposed system, (Polychronopoulos *et al.*, 2004) described the employed sensors: an 8-14 μm far infrared camera with a cooling unit, a 77 GHz Celsius microwave radar and inertial sensors. The road borders tracking was performed by a Kalman filter approach, which was used to estimate the parameters of the model that described the road geometry taking into account data directly from the radar, the obstacle tracking module and the inertial sensors. In situations of partly or complete objects occlusion, the tracking was stabilized by the combination of radar and IR information. When no detections were available and neither obstacle was tracked, the proposed system could estimate the curvature of the clothoid based on the host vehicle tracking.

In (LeGuilloux *et al.*, 2002) data coming from radar and infrared camera were merged in order to help the driver in detecting obstacles in the frame of the PAROTO system. The objects they were looking for in infrared images (vehicles and pedestrians) present regional maxima (the wheels and exhausts for the motorized vehicles or the entire human body for pedestrians) with the following properties: strong contrasts with their neighbourhoods, thin frontiers and fair uniformity. Assuming a very simple model for an object (mean height, width and length), the measurements allow the computation of the object dimensions in the image at any position. By combining regional maxima and edge detection, a simple filter (based on intrinsic region properties like internal contrast, entropy, proportions, size) rejects erratic false detections. The remaining objects were vehicles, pedestrians and traffic signs. LeGuilloux et al. study the motion because of two purposes: the first one was to separate objects in "fixed in the scene" (like traffic signs) and "moving in the scene" (like vehicles). Their second purpose was to decide if an object was dangerous or not, therefore to know if its trajectory was conflicting with that of the host vehicle or not. In a paper continuing this work (Blanc *et al.*, 2004), the sensors employed for the PAROTO system implementation were: a 77 GHz radar which detected obstacles up to a distance of 150 m along the axis of the vehicle, and an infrared camera which saw vehicles even coming sideways, up to a distance of 75 m. The radar and infrared observations were fed into a constant velocity Kalman filter for the tracking operation, which was

also based on radar-IR camera fusion. The infrared processing results were obstacle image positions. In order to recover 3D positions from 2D image positions, they have assumed that the road belongs to a plane and a backprojection was performed. Since backprojection operation is sensitive to the changes of pitch, they also performed motion analysis in order to refine the pitch estimate and correct 3D positions accordingly. Infrared tracks were less accurate in distance than the radar ones, and errors on infrared measures increased with the distance; still, direction estimates were more accurate in infrared. The error area of the dual mode track (the track obtained with the fused radar-infrared data) was far smaller than its single sensor counterparts; therefore, once again fusion demonstrated its benefit.

A sensor system based on an array of passive infrared sensors is presented in (Linzmeier *et al.*, 2005b) having the ability to detect pedestrians without illuminating the environment. These thermopiles were used as sensors detecting objects located within their field of view and presenting a temperature different than that of the background. An important advantage of these sensors is that they do not need the high resolution that is offered by the infrared camera to detected pedestrians, and their reasonable price compared to other sensor systems. Signal interpretation is the main part of these thermopile sensors, involving computation of detection probabilities for individual sensors, which represent an estimate of whether an observed signal change is due to a pedestrian or not. For this purpose, all available sensorial information (like signal-voltage, gradient, velocity, steering angle and ambient temperature) was interpreted as a pattern and the probabilities were determined through a classification task. The detection probabilities computed for individual sensors were then fused using the Dempster-Shafer theory, to provide a single probability of pedestrian detection. The limitations of the system are highlighted in scenarios where pedestrians were in front of objects with similar temperature (e.g. cars exposed to sunlight or house walls on hot days).

The same author, Linzmeier et al. proposed in (Linzmeier *et al.*, 2005a) a system based on the aforementioned thermopile sensor array but combined with two short range radars. Radars, integrated in the front bumper of the test vehicle, were able to observe and track multiple targets in the ROI, but they have difficulties to distinguish between pedestrians and other objects. Even the thermopile system by itself was able to detect pedestrians and locate their position, the reliability of the detection depended on the object background contrast. The target lists of the radars were independent from each other and the maximum detection range was dependent on the target texture and it was approximately 30 meters. Radar provided the thermopile system with useful position information about objects within the ROI. By means of this information, small object background contrasts causing weak probabilities of pedestrian presence were solved. In this paper of Linzmeier et al., two architectures for the fusion of thermopiles and radars are described. The first fusion approach is at a high level and combines processed data from the thermopile and radar system. The output signal was interpreted whether there was a potential pedestrian within the field of view or not, and assigned a probability to every sensor for pedestrian or no pedestrian. These probabilities were then fused (using Dempster-Shafer combination rule) to one single probability for the respective target. The object attributes, like position, dimensions, probability of detection and uncertainty of the decision were inputs to the fusion module as well as the information coming from the radar system (i.e., position information, Doppler-velocity and angle information of each target). The second fusion approach is based on a central level fusion architecture based on an Occupancy Grid. Measurement data from all sensors was mapped into that grid and then processed by means of probabilistic techniques. Unlike the previous fusion approach, in this second case, the fusion was not based on existing objects provided from both sensor systems, but the fusion processor generated objects based on the measured data. The complementary data from thermopiles and radar enabled an accurate detection whether it was a warm object (i.e a pedestrian) or not.

In the following, we mention how the systems previously exemplified as using radar(s) and camera(s) fulfill the requirements specified in section 1.3.

How are the requirements fulfilled?

The system cost (R1): the fact that these systems employ an active sensor like radar makes that the price of such a system, even when combined with visible spectrum camera(s), to be dictated by the active technology; in addition, interference problems are present because of radar too. Combining radar with passive infrared spectrum camera(s), the cost of the system will be even higher due to the camera(s). The real time request (R2) is fulfilled by these systems, even image processing sometimes could lead to long processes, time consuming. If fast image processing algorithms are used and they are not applied on each frame provided by the vision system, but just from time to time (like most systems do), then we can talk about real time at camera(s) too. The efficiency of the system (R3) is very much improved by the use of camera(s) (visible or infrared spectrum one(s)) through the multitude of information they produce on the objects from the scene, especially when considering the object recognition task. All mentioned systems which use a combination of radar(s) and VIS spectrum camera(s), generally detect only vehicles and they are adapted to this specific shape in the image processing module (in which they performed obstacle detection or recognition based on symmetries, tail lamp, ratio constraints, and so on). These algorithms specific to vehicle shape could be adjusted to the pedestrian shape, from the radar side, by the detection of obstacles presenting poorer reflections (at least 5% reflectivity) and by the generalization of the aimed shape in such a way to include even some other types of objects. All these algorithms suited for detecting vehicles are adapted to some specific functioning time: day - for symmetry computation, or night - for tail lamp detection. Those objects which could not be detected or recognized by the radar itself, they could be detected or recognized with image processing techniques, therefore by using a system fusing radar and camera(s). By this fusion, radar's poor lateral measurement is compensated by the information provided by camera(s). Robustness (R4) is the strongest advantage of a system employing an active sensor like radar. During daytime, the visible spectrum camera(s) could help (or assist) radar(s), while on night-time the infrared camera(s) could bring in its benefits by the night vision technique. Therefore, using cameras in a specific situation, performances provided by radar(s) could be improved even in difficult conditions.

2.2.1.2   Systems combining laser scanners and passive sensors

The excellent range accuracy and fine angular resolution make laser scanners suitable for applications in which a high resolution image of surrounding is required. However, since they are optical sensors, different weather conditions like fog or snow will limit their detection range.

In (Laneurit et al., 2003) an approach able to localize vehicle and obstacles on the road is presented. Data fusion from proprioceptive and exteroceptive sensors like odometer, a wheel angle sensor, GPS, **lidar** (a Laser Mirror Scanner LMS-Z210-60 which deliver 3D images for obstacle detection and tracking), **and** a monocular **visible spectrum camera** as well as the knowledge of the road map allow this localization. In addition, a flat world assumption was made and the estimation and updating of the state vector were achieved by Kalman filters. The approach used a two-parts-detection-algorithm: first the segmentation of the 3D image in regions (by a region growing algorithm) and second the recognition of the obstacle (particularly road vehicles) among these regions were performed. The characteristics (width, height and distance) of an obstacle were compared to the model of a car. If the parameters of a region were close to those of the model (different obstacle models could be used, corresponding to cars, trucks or pedestrians), the respective region was declared as an obstacle. Furthermore, the calibration between camera and laser scanner allowed filtering obstacles in order to retain only those that were on the road. Knowing the environment map, a road membership analysis module was able to locate more accurately obstacles on the road. Three steps were needed to update the vehicle localization: determine obstacle position in world reference from laser scanner data, deduce estimated obstacles positions knowing their road membership and finally deduce vehicle position taking into account estimated obstacle positions to be fed into the Kalman filter. Their system is based on a multi level data fusion process: first, it was able to locate the host vehicle using

cooperation between odometer, GPS, wheel angle sensor and road tracker based on vision and then, after obstacles were detected, data association was done for tracking by the use of lidar.

For the obstacle detection system presented in (Labayrade *et al.*, 2005) and (Perrollaz *et al.*, 2006) data fusion is performed between stereovision and laser scanner.

Labayrade et al. presented in (Labayrade *et al.*, 2005) the obstacles detection system from the ARCOS project, which was based on fusion between stereovision and laser scanner. The stereovision algorithm used the "v-disparity" transform (Labayrade *et al.*, 2002) to perform robust and generic obstacles detection. Laser points were clustered together to build targets using Mahalanobis-like distance. A tracking algorithm implementing belief theory was used in order to perform tracking for each sensor. Instead of performing the tracking in the top view coordinates system for targets coming from both laser scanner and stereovision sensor, as in (Labayrade *et al.*, 2002), (when far targets were not detected by the stereovision module because it was not accurate enough), in the system presented in (Labayrade *et al.*, 2005) the tracking process was realised directly in the image coordinates system, and performed back-projection after the tracking process. Once the tracking was done for both sensors, the tracks were fused together using cartesian distance, width and orientation as criteria in the coordinates system. The final position, width, and relative velocity were the ones coming from laser scanner. Thus, the stereovision was used to increase the certainty about the existence of the tracks, which were confirmed when their certainty was above a threshold. The employed CCD cameras are 8 bits grey-scale, SonyTM 8500C with a frame rate of 25 Hz and the laser scanner is a SickTM model, having as output a set of 200 laser points, provided at each 26 ms. The range of the system is up to 40 m. For differentiating between different targets, Labayrade et al. used some median values for the dimensions of different targets, like pedestrian, a box, vehicles and cyclist.

In the approach presented by Perollaz et al. in (Perrollaz *et al.*, 2006), the obstacle (i.e., pedestrian and vehicle) detection and tracking tasks were performed with a laser scanner and a stereo vision system mounted on the experimental vehicle from LIVIC laboratory. The stereovision was used to confirm the detections provided by laser scanner and consisted in 4 major steps: determination of ROIs in the stereoscopic images, application of a numerical zoom to maximize the detection range, computation of a local disparity map in the ROIs and an evaluation of this disparity map to confirm the existence of an obstacle. The stereoscopic sensor was composed of two 8 bits gray-scale SonyTM 8500C cameras with images grabbed every 40 ms. The laser sensor was a SickTM scanner which measured 201 points every 26 ms, with a scanning angular field of view of 100 degree.

Scheunert et al. (Scheunert *et al.*, 2004) and Fardi et al. (Fardi *et al.*, 2005) developed a multi sensor system that used an **infrared camera with a laser scanner** and ego motion sensors. To combine the information from these sensors, a Kalman filter based data fusion was used.

A multi sensor system consisting of an uncooled far infrared camera and a laser scanning device was used for the detection and localization of pedestrians in (Scheunert *et al.*, 2004). The presented system is described as a two level signal processing system, from which the first level performs the detection operation and the second one the fusion task. Obstacles detection was performed independently for each sensor and the outcome was a list of detected objects at every measuring time for each sensor. Within the fusion level, the detection lists of the individual sensors were combined and common object tracks were generated. Every physical object in the frame of the laser scanner was represented by a pair of steps (the moving legs) in the signals, then the detection hypotheses were computed by a combination of steps in the range and finally they were verified about their reflectivity. On the other side, objects representing pedestrians in the image plane of the infrared sensor, were identified by two features: the high brightness due to the relative high human temperature (pixel based feature) and the orientation which was situated in the vertical range (region based feature). A threshold method has been used to produce a binary image, followed by a grouping process for adjacent pixels, providing a set of regions of different sizes and shapes. In the final step, only those regions with a predefined area size were considered as hypothesis. The fusion of the data streams

generated from individual sensors at the same time was realized using a dynamic movement model of the observed objects in connection with estimation techniques by a Kalman filter approach.

In (Fardi *et al.*, 2005) a multi sensor system consisting of a far infrared camera, a laser scanning device and ego motion sensors is presented. To handle the combination of the information provided by these sensors, a Kalman filter based data fusion was used. On the basis of the knowledge about the precise object position (delivered by the laser scanner and the tracking algorithm) and an assumed human height as well as the coordinate transformation, the ROI was created in the image plane for each measurement in the host vehicle coordinate. To get a precise description of the shape of the objects in the ROIs, a contour based segmentation method was used. They have applied the active contour method introduced in (Kass *et al.*, 1987). Afterwards, the extracted contour was transformed using the Fourier transform and in order to decide if the object represented a pedestrian or not, the Fourier descriptors were computed and compared with a reference sets using the euclidian distance. Inspired by the work of Curio et al. (Curio *et al.*, 2000), the optical flow was estimated using the oriented smoothness. In the next step, time series were generated for features which were computed using the zero and first order moments of the optical flow images. Finally, the Fourier transform was applied to obtain the significant cycle of the walking pedestrian.

How are the requirements fulfilled?

In the following, we mention how these systems using laser scanner(s) and camera(s) obey the four requirements from section 1.3. The system cost (R1): the fact that these systems employ an active sensor like laser scanner, makes that the price of such a system, even when combined with VIS camera(s), to be dictated by the active technology and to be even higher than in the case of systems employing a radar instead of a laser scanner; in addition, interference problems are also present because of active technology. Combining laser scanner(s) with passive IR camera(s), the cost of the system will be increased once more due to the camera(s). The real time request (R2) is fulfilled by these systems too, like in the case of systems using radar(s) and camera(s). Laserscanners are able to deliver long range images with high angular resolution and the measurements of these sensors are extremely accurate and precise. Even successful research has been made on detection of pedestrians with laser scanners, this type of sensor is however not the ideal choice due to the fact that even it offers excellent resolution, generally it is used only for moving (i.e. walking) humans. This is due to the additional information about the speed of the pedestrian, which could be inserted in the system in order to improve the laser points' clusterization process. Even laser scanner(s) provide good measurements, in both longitudinal and lateral directions, the efficiency of the system (R3) could be very much improved by the use of camera(s) (visible or infrared spectrum one(s)) through the multitude of information they produce on the objects from the scene. All the mentioned systems which use a combination of laser scanner(s) and visible spectrum camera(s), are generally dedicated to pedestrian detection and recognition and they are adapted to this specific shape in the image processing module (in which they performed obstacle detection or recognition based on legs detection, human gait detection or geometric features corresponding to humans). Vehicles and other objects (generally large-scale objects) presenting typical characteristics could also be detected by the adaptation of the features to that specific type of object. Those objects which could not be detected or recognized by the laser scanner itself, they could be detected or recognized with image processing techniques; therefore, in a system fusing laser scanner(s) and camera(s), laser scanners' measurement is compensated by the information provided by camera(s). Robustness (R4) is also the strongest advantage of a system employing an active sensor like laser scanner. In the same manner, as in the case of systems employing radar instead of laser scanner, during daytime, the VIS camera(s) could assist the active sensor, while on night-time the IR camera(s) could bring in its benefits by the night vision technique.

2.2.1.3   Systems combining several types of active and passive sensors

The new generation of intelligent vehicles senses the environment by a combination of different sensors: generally, three different types of sensors, or even more; we called these types of systems **"all fused"**. Where a single radar may not be enough to cover the whole interested area, a radar network comprised of several radars with different beam designs are used.

SAVE-U was a high performance sensor platform for the active protection of Vulnerable Road Users (VRU) such as pedestrians and cyclists (Meinecke *et al.*, 2003), (Marchal *et al.*, 2003), (Tons *et al.*, 2004). The central objective of SAVE-U was to develop a pre-impact sensor platform using three different technologies of sensors: a radar network composed of 5 single beam 24 GHz sensors distributed in front of the car and covering the full car width, an image system composed of a passive infrared camera and a visible spectrum colour camera. For increasing the reliability of the object detection task, the output of each sensor was fused in a low and high level data fusion technique. At the detection level, the radar system and the vision system performed each the own detection procedure: each radar sensor detects reflection points and a fusion algorithm combines all these reflection points, while for the vision system the detections in colour and infrared domain were fused. At the low-level fusion, radar information (radial range and speed, angle indication, rough classification) were sent to the video detection stage to check if something could be found in the areas of images corresponding to radar indications. This procedure was executed in addition to the pure video method of detection in order to achieve better results. Once the low-level fusion was done (at the pixel level since areas in sensors were put in correspondence), ROIs were sent to the classification stage if they represented potential pedestrians or cyclists, and finally, the classification stage indicated whether ROI contained the aimed obstacle or not. Executed in parallel, high-level fusion consolidated obstacles detection and object tracking to get the final object characteristics (like position, speed and signature). The multitarget tracking algorithm solved possible object association and correspondence problems.

In March 2004, the whole world was stimulated by the "Grand Challenge" organized by Defense Advanced Research Projects Agency (DARPA), when fifteen fully autonomous vehicles attempted to independently navigate approximately 400 km in desert within no more than 10 hours, all competing for a 1 million cash prize (DARPA Grand Challenge, 2004-2005). The tasks mandatory accomplished by a vehicle to be accepted in this competition were: no human intervention (i.e. no driver), no remote-control, just pure computer-processing and navigation resources (they could use GPS systems). Although, even the best vehicle (i.e. the one developed by Red Team from Carnegie Mellon University) made only seven miles, it was a very big step towards building autonomous vehicles in the future. The second competition of the DARPA Grand Challenge was held in October, 2005 when five vehicles successfully completed the race and won the competition. The first place was won by "Stanley" from Stanford Racing Team Stanford University, Palo Alto, California (Stanford Racing Team, Stanford University, 2005), the second one and the third one by "Sandstorm" and "H1ghlander" from Red Team Carnegie Mellon University, Pittsburgh, Pennsylvania (Red Team, Carnegie Mellon University, 2005), the fourth one by "Kat-5" from Team Gray, the Gray Insurance Company, Metairie, Louisiana (Gray Team, The Gray Insurance Company, 2005), and the fifth one by "TerraMax" from Team TerraMax Oshkosh Truck Corporation, Oshkosh, Wisconsin (TerraMax Team, Oshkosh Truck Corporation, 2005). To navigate, "Stanley" (Thrun *et al.*, 2006) used 5x SICK AG lidar units to build a 3-D map of the environment. An internal guidance system utilizing gyroscopes, accelerometers and odometers monitored the orientation of the vehicle and also served to supplement the GPS system. Additional guidance data was provided by a colour video camera used to observe driving conditions out to 80 m. For long-range detection of large obstacles, "Stanley" also employed 2x 24 GHz radar sensors, covering the frontal area up to 200 m. The sensors used by "Sandstorm" and "H1ghlander" (RedTeam, 2005) included 3x lidar laser-ranging units, one Long Range lidar, a radar unit, and a pair of cameras for stereo vision. Sandstorm also has a GPS and an inertial navigation system for determining geographical position. "Kat-5" (Trepagnier *et al.*, 2006) uses 2x Sick LMS 291 ladar devices providing the autonomous vehicle with environmental sensing,

together with the INS (Inertial Navigation System) or GPS module. Kat-5's primary electrical system, used to run the computers and drive-by-wire system, was powered by the standard electrical system of the vehicle while the 24-volt system, used to power the ladar sensors, was powered by six solar panels on the roof platform of the vehicle. "TerraMax" (TerraMaxTeam, 2005) was equipped with 4x SICK LMS 221 ladars providing the distance and the location information of the obstacles around the vehicle, a vision system consisting of 6 CCD digital colour cameras (two pairs were used to provide forward and rear looking stereovision information and the two single cameras sensed the terrain in front and behind the truck), 2x Eaton-Vorad radars for providing 150 m range target tracking, and 12 ultrasonic sensors mounted around the vehicle for short range sensing. Two GPS units provided the information for route and mapping purpose.

The third competition of the DARPA Grand Challenge, also known as the "Urban Challenge" (DARPA Grand Challenge, 2007), took place in November, 2007 and it involved a 96 km urban area course, to be completed in less than 6 hours. The most important rules included: the vehicles had to obey all traffic regulations while negotiating with other traffic and obstacles and merging into traffic, and they also had to be entirely autonomous, using only the information they detected with their sensors and navigation signals such as GPS. Also, the vehicles had to operate in rain and fog, with GPS blocked. From a number of 53 teams registered for this competition, DARPA qualified only 11 teams in the final race, due to safety reasons (real humans and DARPA officials scoring robot performance were expected to be situated near the robots in the traffic scene). The 2 million winner was the vehicle called "Boss" (Tartan Racing team, Carnegie Mellon University, 2007) of the Tartan Racing team, a collaboration between Carnegie Mellon University and General Motors Corporation. Coming in the second place and earning the 1 million prize was the Stanford Racing Team with their "Junior" vehicle (Stanford Racing team, Stanford University, 2007). The third place was won by the team Victor Tango from Virginia Tech winning the 500,000 prize with "Odin" (Victor Tango team, Virginia Tech, 2007). MIT (Cambridge, Massachusetts) with "Talos" (MIT team, 2007a) placed 4th, and "Little Ben" (Ben Franklin Racing team, University of Pennsylvania/Lehigh University, 2007) and "Skynet" (Cornell team, Cornell University, 2007) from University of Pennsylvania/Lehigh University and Cornell University also completed the course. "Boss", the Tartan Racing vehicle, for sensing the environment it used a combination of sensors, like: SICK LMS 291-S05/S14 lidar, Velodyne HDL-64 lidar, Continental ISF 172 lidar, IBEO Alasca XT lidar, Ma/Com radar, Continental ARS 300 radar, a vision system and GPS positioning. The Stanford Racing vehicle, "Junior", estimated its location, orientation and velocity by Applanix POS LV 420 Navigation system (three GPS antennae) and some odometers. For external sensing, "Junior" employed a Velodyne HD lidar laser range finder, an omni-directional Ladybug camera which comprised six CMOS video cameras and two long-range radar sensors, complementing thus the laser data. A coordinated pair of IBEO Alasca XT fusion laser rangefinders, a single IBEO Alasca A0 unit with a range of 80 meters were used to detect vehicles behind "Odin", the vehicle of the Victor Tango team, and navigate in reverse. In addition, two imaging source colour monocular cameras were used to supplement the IBEO classification software, while two SICK LMS 291 ladars detected sharp changes in the level road. Other two side-mounted SICK LMS 291 single plane rangefinders were used as simple "bumpers" to cover the side blind spots of the vehicle and ensured 360-degree coverage. For "Talos", the MIT vehicle, the perception system design incorporated many inexpensive lidars, radars, and cameras: twelve SICK LMS291 lidars, twelve Delphi ACC3 automotive radars, and ten Point Gray Firefly MV cameras. "Little Ben" was equiped with a variety of 2D and 3D ladars (a Velodyne HD ladar and a set of forward and rear facing SICK 2D LMS-291 ladar) as well as VIS stereo cameras. For navigation, GPS was used. In the frame of the "Skynet" vehicle, a suite of sensors, like LMS-291 lidars, IBEO(s) ALASCA and Delphi radars, a MobilEye VIS vision system (a mono camera and a pair of stereo cameras), GPS, and inertial sensors were employed.

While the 2004 and 2005 events were more physically challenging for the vehicles, the robots operated in isolation and did not encounter other vehicles on their course. Other than previous autonomous vehicle efforts that focused on structured situations such as highway driving with little interaction between the vehicles, this competition operated in a more cluttered urban environment

and required the cars to perform sophisticated interactions with each other, such as maintaining precedence at a 4-way stop intersection.

How are the requirements fulfilled?

In the following, we mention how these powerful systems using all types of technologies, passive and active together, obey the four requirements from section 1.3. The system cost (R1): the fact that these systems employ multiple active and passive sensors makes that the price of such a system, even its performances are good, to not allow the implementation on some commercial vehicles even in the near future when it is expected that the technology will evolve and the production cost will decrease. In addition, interference problems are still present because of active technology. The real time request (R2) is fulfilled by these systems, as their use in DARPA competition has proved. By using a multitude of different types of active and passive sensors, the efficiency of the system (R3) is the best from all the systems presented in this chapter. Vehicles, pedestrians and other objects have been detected by these systems, and in addition their automatic navigation on the road was performed solely by the use of sensors (there was no driver). Robustness (R4) is also very much increased in the frame of these systems; they could be able to navigate, detect and recognize obstacles on daytime, or even on night-time and in difficult conditions.

After presenting very powerful systems using both passive and active sensors and saw which their strengths but also their disadvantages are, we look toward some less expensive systems, so those employing only one type of technology: either active or passive. Next, we detail systems presented in literature which addressed the intelligent vehicle domain by the use of only active sensors.

2.2.2 Systems using only active sensors

There are many systems using just a single type of sensor in order to solve the ODR task, but we believe their performances are not very remarkable compared to those of systems which may employ a combination of two or many sensors and perform the same task. Some systems using a single type of active sensor, like radar or laser scanner are first presented; then, systems which choose to combine different types of active technology in the frame of one single system are mentioned.

2.2.2.1 Systems using a single type of active sensor

In the following, we present the most important systems from the literature which approached the intelligent vehicle domain by the use of a single type of active sensor like radar or laser scanner.

Radar

The main advantages of radar sensors are their possibility to detect obstacles and to provide their distance and velocity. Generally, a network of radar sensors is used to get the full coverage area in front of the vehicle, but there are also systems using a single powerful radar sensor.

An 76.5 GHz radar with the image resolution 32x115 pixels and a scan rate of 5 Hz was used in (Meis & Schneider, 2003). The object segmentation has been performed by clustering colour connected components in the preprocessed radar data, which previously were filtered according to some intensity noise floor. The obtained clusters leaded to estimation of objects' position and dimensions (width and length) and provide their relative velocity on longitudinal and lateral axes, together with their alignment. By using this type of radar, the detection of objects up to a distance of 120 meters was possible. The authors have performed also road course detection and estimation.

In (Gavrila *et al.*, 2001) Gavrila et al. describe three sensor technologies: radar, laser scanner and vision, all intended to be used in the Preventive Safety for Unprotected Road User (PROTECTOR)

project for pedestrian detection from a moving vehicle. They showed that in the case of radars, the power spectral density (PSD) values for a pedestrian and a vehicle differ both in signal amplitude (vehicle's PSD is 40 times higher) and in shape (the vehicle signal is larger in width and more structured) and this information conduct them to the conclusion that some other possible obstacles could also be recognized based on their received specific signals provided by radar.

To discriminate between different types of obstacles, Kruse et al. (Kruse *et al.*, 2004) developed a target recognition technique based on two different 24 GHz radar systems: (1) first, they used a sensor network composed from four 24 GHz High Range Resolution (HRR) pulse radars, capable of detecting targets up to 20 m and (2) as a second system they considered a single Universal Multimode Range Resolution (UMRR) radar of 24 GHz, able to detect targets up to 60 m. After a target has been detected by one of these two systems, the echo signal of each target was analyzed and described by a set of 10 to 15 different features enclosed in a feature vector. The discrimination between some possible obstacles like car, cyclist, pedestrian, tree, traffic sign and group of persons has been performed by the use of this feature vector extracted from the radar echoes. The calculation of different features was depending on the physical properties of the object classes, like: RCS, geometrical properties and velocity. As classification scheme, a polynomial classifier was used to evaluate the features influence in the decision process where about 2,000 samples were used per object class.

In the frame of systems employing active sensors, the obstacles could also be tracked: an $\alpha\beta$ tracker was used in (Meis & Schneider, 2003) together with the known relative radial velocity for predicting the objects' velocity in longitudinal direction.

Laser scanner

Laser range finder or laser scanner is another possible viable solution for obstacle detection.

The Kalman filter is independently applied to every hypotheses of detected obstacle also for tracking reasons in (Ewald & Willhoeft, 2000), (Fuerstenberg *et al.*, 2002), (Fuerstenberg & Dietmayer, 2004), (Mendes *et al.*, 2004).

In order to improve the pedestrian recognition task, there are authors using also the pedestrian typical movement or gait to differentiate between e.g. a tree and a pedestrian (Fuerstenberg & Dietmayer, 2004), (Mendes *et al.*, 2004). Legs detection have been performed by observing a sequence of scans, followed by the searching for alternation of one and two small objects separated by a distance less than 50 cm, which were assumed to represent the legs (Mendes *et al.*, 2004).

Ewald et al. (Ewald & Willhoeft, 2000) have used a Ladar Digital A AF which is a laser emitting pulses in the near infrared spectrum. Its scanning range covers areas in front of the car up to 150 m for targets such as traffic signs, but other objects having at least 5% reflectivity (almost all possible targets) could be detected in a reduced range: up to 40 m, with a scan frequency of 10 Hz. The objects' size, position, but also their velocity and acceleration (obtained from two successive scans) are available for all the detected obstacles after 300-500 ms from the moment of their detection. The laser scanner they used is also able to track obstacles using a Kalman filter.

In the paper of Gavrila, (Gavrila *et al.*, 2001), obstacles in a range up to 40 m could be detected by the IBEO laser scanner which covers the area in front of the vehicle, having a field of view of 180 degree, a scan frequency of 20 Hz and a precise measurement of depth ($\pm$5 cm).

In (Fuerstenberg *et al.*, 2002) the typical appearance of objects, the history of the tracked object with respect to previous classification results and the estimation of its absolute velocity were used to classify target objects. The approach from (Fuerstenberg & Dietmayer, 2004) uses as additional information the objects reflectivity, measured also by the laser scanner. In (Fuerstenberg *et al.*, 2002)

LadarDigital MultiLayer (LD ML) laser scanner was used with a range up to 50 m, a field of view up to 270 degree, variable scan area and frequency: 5 to 40 Hz (but generally used at 10 Hz). Later, the same author in (Fuerstenberg & Dietmayer, 2004) employed an ALASCA laser scanner having an horizontal field of view up to 240 degree and a 10 Hz scan frequency. Some processing time are given in this latter paper: the pedestrian is classified 0.3 s after its detection in a distance of 66 m, which means approximately 6 s before the pedestrian and the test vehicle reach each other.

The experimental system used in (Mendes *et al.*, 2004) was a four wheel-drive electrical vehicle called "Robucar", equipped with a LMS200-Sick laser scanner which was setup with an angular range of 180 degree, length range of 8 m and a scan rate up to 37.5 Hz, and was developed to be integrated in a Cybercar vehicle.

Sadou et al. (Sadou *et al.*, 2004) presented an obstacle detection algorithm that considered the possibility of occlusions along the navigation path in order to reduce the search for obstacles to only the essential regions. The obstacle detection algorithm has been implemented on-board of a mobile robot. The sensors used are a fiber-optic gyrometer, a dual axis inclinometer and the laser range finder (SICK OPTICS). Given the platform tilt angle, as measured by the inclinometer, the system could adjust the inclination of the sensor to distinguish an obstacle from a hill or a ditch on the trajectory of the vehicle.

Several different sensors were selected for further experimental test and review in (Bu & Chan, 2005). The sensors include an IBEO laser scanner, an Eaton VORAD EVT-300 radar (Doppler radar), a microwave radar MS SEDCO SmartWalk 1800 and an IRIS people counter (infrared based sensor), among others.

How are the requirements fulfilled?

How the systems previously exemplified as using a single type of active sensor fulfil the requirements specified early in section 1.3? The first requirement was about the system cost (R1): the fact that these systems employ active sensors makes that this first requirement to not be met due to the high cost of these technologies; even using a single type of active sensor, the cost of the system does not allow its implementation on some comercial vehicles. In addition, when several vehicles are equipped with such sensors, the possibility of appearing interference problems is high. The real time request (R2) which states that the system has to be fast enough to detect and then recognize obstacles in real time is fulfilled by these systems, both radars and laser scanners, and distances at which obstacles are detected in urban streets are up to 40 m. The efficiency of the system (R3) is partially fulfilled by the systems employing radars because radar is not very efficient in detecting non-metallic obstacles and not-moving obstacles. From the recognition point of view, comparing radar and laser scanner, radar is lossing this time too because its discriminant power in the lateral measurements is much smaller than the laser one. Still, both types of sensors (radar and laser scanner) often fail to detect small objects or occluded ones. Robustness (R4) states that the system must cope with difficult context in which it has to assure a well functioning, no matter what are the illumination or weather conditions. This is the strongest advantage of active sensors, their possibility to assure a well functioning also in difficult conditions of weather or illumination.

2.2.2.2 Systems combining several active sensors

In order to exploit the advantages of multiple active technologies, the sensors' information could be fused in a sensor fusion process. In this way, some complementary regions could be covered by different active sensors working together for the same purpose: the road obstacle detection.

Radar and laser scanner have features which complement each other. Laserscanner have a larger field of view than the radar, and its side and longitudinal resolution are better than the radar ones; therefore,

the recognition obstacle degree is higher for laser scanner. On the other side, radar is insensitive to atmospheric changes. To conclude, radar offers good accuracy in longitudinal distance measurement, but poor accuracy in the lateral one and because laser scanner offers a good accuracy also in the lateral one, some systems may combine radar and laser scanner to increase the reliability of the entire system.

An example of such a system is the one developed by Mobus et al. (Mobus & Kolbe, 2004) which combines a 77 GHz ACC radar (having a limited lateral range of ± 3.4 degrees, but a longitudinal one of 150 meters) with an infrared laser sensor (capable of scanning an area of about 20 degrees up to 80 meters). Radar and laser scanner are fused by a Kalman filter probabilistic data association at the sensor level. By using this configuration, complementary regions added up to a more complete range ahead the car, increasing the performances of the entire detection and tracking system.

A data fusion module developed and integrated in two different systems is presented in (Blanc *et al.*, 2004). The first one, is the PAROTO system which combines infrared imagery with radar to perform obstacle avoidance (this system is presented at the radar-infrared camera fusion, in the following section, 2.2.1). Blanc et al. have also developed the VELAC system, which combines radar and lidar tracking to improve the environment interpretation. The 3D-Laser Mirror Scanner LMS-Z210-60 is a surface imaging system based upon accurate distance measurement. The radar used is working at 77 GHz and it is aimed for long range target detection (up to 150 m). For the obstacle detection task, a two-step algorithm is used: first the segmentation of the 3D image in regions and second the recognition of the obstacle (particularly road vehicles) among these regions is performed. The characteristics provided by the laser scanner are then compared to a car model. After detecting different obstacles, the system is able to track them in consecutives frames using a Kalman filter.

How are the requirements fulfilled?

After we saw how the systems using a single type of active sensor succeed or fail in fulfilling the four requirements, we want to see if their combination in systems using both radars and laser scanners bring in some benefits. The requirement about the system cost (R1) is in fact even worst than in the previous case, because combining multiple active sensors makes the cost to be even higher; the level of interferences will be increased too. The real time request (R2) is not affected in the negative way by this combination of sensors, but it is possible that more objects to be faster detected and recognized by these fused systems. The efficiency of the system (R3) could be improved by the use of both radars and laser scanners due to the high accuracy provided by the laser scanner in the lateral measurement. Robustness (R4) is still the strongest advantage of active sensors, and their possibility to assure a well functioning also in difficult conditions is making their combination even more powerful in such situations.

As we mentioned at the begining of this section, all reviewed systems using an active technology are summarised in table 2.1. They are grouped in different categories, based on the sensors they have employed to accomplish the ODR task. There are systems using a single type of active sensor, or a combination of several active sensors or even a combination between passive and active sensors. Therefore, all the systems exemplified in this chapter and which use an active sensor would be found in this table, together with the specific type of sensor(s) they employed in their functioning.

From those presented so far, the sensors we preffered for the implementation of our ODR system can be foreseen. The major disadvantages of active sensors, to lead to possible interferences when these sensors are mounted on multiple vehicles in a cluttered traffic, and their acquisition price (especially that of laser scanners) comparing to that of vision cameras, convinced us that a system formed only by passive sensors is the best solution for our objective. Even the main problem with vision systems is the detection task, fortunately, we have several possibilities to accomplish it solely based on vision. These will be highlighted in the next section, dedicated only to passive sensors.

Table 2.1: Systems using one or multiple types of active sensors

		TYPES OF ACTIVE SENSORS	
	References	RADAR	LASSER SCANNER
RADAR	(Gavrila et al., 2001)	→ 24GHz;	
	(Meis & Schneider, 2003)	→ 76.5GHz;	
	(Kruse et al., 2004)	→ 2x24GHz HRR ;	
	(Kruse et al., 2004)	→ 1x24GHz UMRR ;	
	(Bu & Chan, 2005)	→ Eaton VORAD EVT-300;	
	(Bu & Chan, 2005)	→ MS SEDCO SmartWalk 1800;	
LASER SCANNER	(Mobus & Kolbe, 2004)	→ 77GHz + IR LS;	(Gavrila et al., 2001) → IBEO;
	(Blanc et al., 2004)	→ 77GHz + 3D-LMS-Z210-60;	(Ewald & Willhoeft, 2000) → LD A AF;
	(Pietzsch et al., 2008)	→ 24GHz + IBEO ALASCA;	(Fuerstenberg et al., 2002) → LD ML
			(Fuerstenberg & Dietmayer, 2004) → ALASCA;
			(Mendes et al., 2004) → LMS 200 SICK LS;
			(Sadou et al., 2004) → SICK OPTICS;
			(Bu & Chan, 2005) → IBEO;
VIS CAMERA	(Handmann et al., 1998)	→ 77GHz + mono;	(Laneurit et al., 2003) → 3D-LMS-Z210-60 + mono;
	(Franke, 1992), (Gern et al., 2000)	→ 77GHz + mono / stereo;	(Labayrade et al., 2005) → SICK TM + stereo;
	(Milch & Behrens, 2001)	→ 2x24GHz + mono;	(Perrollaz et al., 2006) → SICK TM + stereo;
	(Steux et al., 2002)	→ 77GHz + mono;	(Monteiro et al., 2006) → LRF + mono;
	(Kato et al., 2002)	→ 77GHz + stereo;	
	(Sole et al., 2004)	→ 77GHz + mono;	
	(Kawasaki & Kiencke, 2004), (Kumon et al., 2005)	→ 77GHz + mono;	
	(Alefs et al., 2005)	→ 24GHz + momo;	
	(Schweiger et al., 2005), (Serfling et al., 2008)	→ 77GHz + mono;	
	(Bombini et al., 2006), (Alessandretti et al., 2007)	→ 77GHz,2x24GHz + mono;	
	(Richter et al., 2008)	→ 77GHz + mono;	
IR CAMERA	(LeGuilloux et al., 2002), (Blanc et al., 2004)	→ 77GHz + mono;	(Scheunert et al., 2004), (Fardi et al., 2005) → LS + mono;
	(Andreone et al., 2002), (Polychronopoulos et al., 2004)	→ 77GHz + mono;	
	(Linzmeier et al., 2005b)	→ 2x24GHz + thermopile IR;	
ALL FUSED	(Meinecke et al., 2003), (Tons et al., 2004)	→ 5x24GHz radar + mono VIS color camera + mono IR camera (SAVE-U)	
	(Tatschke, 2006)	→ SRR–LRR radars + laser scanner + mono FIR camera;	
	DARPA05 - "Stanley" (Thrun et al., 2006)	→ 2x 24 GHz radars + 5x SICK AG lidar + a color video VIS camera	
	DARPA05 - "Sandstorm","H1ghlander"(RedTeam, 2005)	→ a radar + 3x lidars + 1 LR lidar + a pair of stereo VIS cameras;	
	DARPA05 - "Kat-5"(Trepagnier et al., 2006)	→ 2x Sick LMS 291 lidars;	
	DARPA05 - "TerraMax" (TerraMaxTeam, 2005)	→ 2x Eaton-Vorad radar + 4x SICK LMS 221 ladar + 6 CCD cameras (2 pairs stereo + 2 mono) + 12 ultrasonic;	
	DARPA07 - "Boss" (Tartan Racing, 2007)	→ Ma/Com, C.ARS300 radars + SICK LMS291-S05/S14, Velodyne HDL-64, ISF172, IBEO XT lidars + a VisionSystem;	
	DARPA07 - "Junior" (Stanford Racing team, 2007)	→ Velodyne HD lidar LRF + 6xCMOS video cameras + 2x long-range radars;	
	DARPA07 - "Odin" (Victor Tango team, 2007)	→ 4xSICK LMS 291 ladars, 2xIBEO Alasca XT Fusion LRF, an IBEO Alasca + 2x mono cameras;	
	DARPA07 - "Talos" (MIT team, 2007b)	→ 12xDelphi ACC 3 automotive radars + 12x SICK LMS 291 lidars + 10x cameras;	
	DARPA07 - "Little Ben" (Ben Franklin Racing team, 2007)	→ a Velodyne HD and a set of SICK 2D LMS-291 ladars + VIS stereo cameras;	
	DARPA07 - "Skynet" (Cornell team, 2007)	→ Delphi radars + LMS-291, IBEO ALASCA lidars + a MobilEye VIS vision system (a mono and a pair of stereo);	

All the systems from table 2.1 are using one or multiple active sensors

2.2.3   Systems using only passive sensors

Like in the case of systems employing only an active sensor in order to solve the ODR task, there is also a multitude of systems employing a single type of camera, VIS or IR to accomplish the same task. Here too, we believe their performances could be overcomed by the systems employing both types of cameras, VIS and IR by the use of fusion. Systems using a single type of passive sensor are first presented and they are followed by systems which choose to combine VIS and IR technology in the frame of one single system.

2.2.3.1   Systems using a single type of passive sensor

Generally, when the detection of obstacles from the road is the task to be solved, but mainly when image processing techniques are involved, the used criteria depends on the definition of what the obstacle is. In some systems, the detection of obstacles is limited to the localization of some specific shape corresponding to certain types of obstacles (like vehicles, pedestrians, cyclists), which is based on a search for specific patterns, such as shape, symmetry, edges, pedestrian's head or vehicle's lights. This search for patterns common to multiple obstacle classes generally lead to the determination of a BB. After potential obstacles are found, an obstacle validation process is carried out. By exploiting strong characteristics discriminating that respective obstacle from some other type of obstacles (as those belonging to background) false detections can be usually removed. When OD task is limited to the localization of specific patterns, processing can be based on the analysis of a single still image, in which relevant features are searched for. Unfortunately, the pattern-based approach is not successful when an obstacle does not match the respective model (later we will show that this model could be a static or a moving one).

A more general definition of an obstacle, which leads to more complex algorithmic solutions, identifies as an obstacle any object that obstructs the path the host vehicle is driving on. In this case, instead of recognizing specific patterns, the OD task is reduced to identifying the area in which the vehicle can safely move and anything rising out significantly from the road surface would be considered as obstacle. Due to the general applicability of this definition, the problem is using more complex techniques, like those based on the processing of two or more images, which are: the analysis of optical flow field or the processing of nonmonocular (i.e. stereo) images.

The optical flow-based technique requires the analysis of a sequence of two or more images: a two-dimensional (2-D) vector is computed, encoding the horizontal and vertical components of the velocity of each pixel. The obtained motion information can be used to compute ego-motion [3] and moving obstacles can be detected and/or tracked in the scene by analyzing the difference between the expected and real velocity fields and by removing background changes.

On the other hand, the processing of stereo images requires identifying correspondences between pixels in a pair of left and right images. In the case of optical flow-based approaches, the obstacle presence is indirectly derived from the analysis of the velocity field, while in the stereo-based approaches, this information is extracted directly from the depth information. Moreover, when both host-vehicle and obstacles have small or null speeds, the optical flow-based approach fails while the stereo one can still detect obstacles.

Many different approaches have been developed to address the processing of shape detection, pattern analysis, stereo vision, and tracking and sometimes systems even combined them to improve the ODR task. The most principal trends in research will be discussed below to give a broad view on this highly developing field. Therefore, in the following, we present which these methods aimed to Obstacle Detection (OD) and/or Obstacle Recognition (Obstacle Recognition (OR)) task are and how they were implemented in the existing systems. Different systems based solely on vision (from

---

[3]There are systems in which the ego-motion information is directly extracted from odometry.

which some have used only VIS spectrum sensors, some only IR spectrum sensors and others have combined the two technologies) are presented.

Overviews on various systems from the intelligent vehicle field performing the OD and/or OR task can be found in (Bertozzi *et al.*, 2002a), (Sun *et al.*, 2006b), (Gandhi & Trivedi, 2006), (Enzweiler & Gavrila, 2009).

Almost all systems reported in the literature follow two basic steps: 1) Hypothesis Generation (HG) where the locations of possible obstacles in an image are hypothesized and 2) Hypothesis Verification (HV) where different tests are performed to verify the presence of obstacles in the locations previously obtained. The objective of the HG step is to find possible obstacle locations in the image for further exploration. The majority of developed systems have used in the HG step one of the following three methods: 1) knowledge-based, 2) motion based, or 3) stereo-based. Knowledge-based methods employ a priori knowledge about the obstacle to hypothesize its locations in the image. Motion-based methods detect obstacles using optical flow. Stereo-based approaches generally use an v-disparity method or Inverse Perspective Mapping (IPM) to estimate the locations of obstacles in images. The hypothesized locations from the HG module constitute the input to the HV step, where different tests are generally performed to verify the correctness of the hypotheses. (Sun *et al.*, 2006b)

1) **Knowledge-based methods**

We review below some representative approaches which employ *a priori* knowledge to hypothesize obstacle locations in images. These methods are based on a specific shape of the obstacle to be detected, so from the beginning of the process, the system knows what kind of shape is looking for. This section is referring to the pedestrian and vehicle classes of objects, because they are more likely to occur on the road in front of the host vehicle; but this discussion could be generalized to some other specific types of obstacles, like animals, trees, bushes and so on. All the methods considering knowledge about a specific shape of obstacle, would fall in one of the following categories:

- **methods applicable both to pedestrians and vehicles**; two different directions can be found:

    - search for features like symmetry, horizontal or vertical edges, texture, colour, contour corresponding to the aimed obstacles
    - template matching with the obstacle's contour, provided as an image or as a set of features extracted from that obstacle image and describing its shape. This method could employ an image of the obstacle as a template and that pattern to be shifted (and sometimes rescaled) over the entire image in which obstacles are searched. Another possibility is to extract some features from the image containing the pattern and the matching with different locations from the scene image to be performed by a classifier at the feature-level.

- **methods specific only to pedestrian or only to vehicle class** (which cannot be generalized to some other types of obstacles), such as geometrical features: corners, headlamps, shadow or edges produced by vehicles or legs or head detection for pedestrian [4].

Next, we will see in detail the main characteristics of the systems which could be included in the previous mentioned categories.

---

[4]Although some systems detecting pedestrians' legs or even the specific information for their moving - human gait - fit these systems to the methods based on moving, we consider them here. Even the moving information is used, these systems are reported to a particular object type, the pedestrian shape, of which gait has been study and assigned to a specific object: the pedestrian; therefore, the pattern of moving legs is very specific to this class of objects.

a. Common specificities to the pedestrian and vehicle classes:

a1. Search for features like symmetry, horizontal or vertical edges, texture, colour, contour

When detecting vehicles or pedestrians on the road, many systems guided the search by some observations, like:
    - an image containing a vehicle inside presents strong edges on all sides where the image changes from the vehicle to the background and the contour formed by the edges, generally can be approximated by a rectangle;
    - an image enclosing a pedestrian is presenting strong vertical symmetry and high density of vertical edges.
Most systems assume that the size and position at which the obstacles appear in images follow a typical distribution.

   Symmetries:
At the University of Parma, Bertozzi et al. (Bertozzi *et al.*, 2000) and Broggi et al. (Broggi *et al.*, 2000b) have developed a real-time vision system, implemented on the ARGO vehicle, for the detection and localization of vehicles on the highway exploiting their symmetric characteristic. The perception of the environment was performed through the processing of images acquired from the stereo vision system installed on board of the vehicle, but the leading vehicle was localized and tracked using a single **monocular visible spectrum image** sequence. The authors guided the implemented vehicle detection algorithm by the following considerations: a vehicle is generally symmetric, characterized by a rectangular BB which satisfies specific aspect ratio constraints, and being on a highway, it should be placed in a specific region of the image (the assumption of a flat road was made). First, a ROI was identified on the basis of road position and perspective constraints and this area was searched for possible symmetries: a symmetry map was computed as a combined sum (the coefficients being determined experimentally) of four different symmetries (i.e., grey-level, edge, horizontal edges and vertical edges symmetries). Once the symmetry position and width have been detected, a new search was started, aimed at the detection of the two bottom corners of a rectangular BB. Finally, the top horizontal limit of the vehicle was also searched for, and the preceding vehicle was localized. Knowing the calibration of the two cameras, a pattern enclosed into the BB from the left image was searched on the right image, and a triangulation method allowed the computation of vehicle distance. The detected and localized vehicle was tracked in the image sequence by a correlation method.

In the frame of the same research group where Bertozzi belong at the University of Parma, Broggi et al. described in (Broggi *et al.*, 2004b) an improved version of the algorithm presented in (Bertozzi *et al.*, 2000). Their aim in this new version was to find horizontal lines located below an area with sufficient amount of edges. Besides this, in order to speed up computation, the new algorithm considers three different ROIs for the detection of vehicles situated at 3 different distances: far, medium and close. Compared to the algorithm from (Bertozzi *et al.*, 2000), in this paper authors used only binarized edges for the symmetry computation, droping the grey level symmetry processing (the authors assert this was very time consuming and did not provide more information compared to edges symmetry). The Sobel operator was used to find edges module and orientation, then three images were built: one with binarized Sobel modules, one with vertical edges and one with horizontal edges. The symmetry was computed for every column of the three images, on different sized BBs whose height matches the image height and with a variable width ranging from 1 to a predetermined maximum value. Also for improving the first version of the algorithm, the authors were searched for the shadow under the car in order to find the box base. In the HV step, distance and size of vehicles were computed based on image calibration. The algorithm deleted all the BBs which were too large, too small, or too far from the camera in order to decrease the number of false positives. The algorithm analyzed images on a frame by frame basis, without using the temporal correlation. Trucks were also detected since their square shape matches the algorithm assumptions. The problem of misaligned vehicles (like overtaking vehicles or vehicles coming from the opposite direction) is

asserted to be the most critical aspect of this method. The algorithm does not always detect oblique vehicles, and when it does, vehicle width may not be precise. Problems arise when a large number of vehicles overlap or in urban scenes, where road infrastructures, road signs and shadows make the scene too complex. The reported execution time was 120 ms with an Athlon XP 1.8 GHz. To conclude, computing the symmetry on the whole image is very time consuming, since the authors assert that 75% of the time was used for symmetry computation.

The pedestrian detection algorithm detailed in (Bertozzi *et al.*, 2002b) is actually an adaptation of the vehicle detection algorithm presented in (Bertozzi *et al.*, 2000). The areas considered as pedestrian candidates were rectangular BBs presenting a strong vertical symmetry and a high density of vertical edges and their projection in the image was obtained from the size and distance of the pedestrian in the 3D world combined with simple perspective considerations and the camera calibration parameters. To be validated in the HV step, the BBs were then checked against a human shape model, taking into account the contour of the object they enclosed by an Ant Colony Optimization algorithm. For the detection step, the columns of the image were considered as possible symmetry axes for BBs. Then, for each symmetry axis, different BBs were evaluated scanning a specific range of distances from the camera and a reasonable range of heights and widths for a pedestrian. A pre-attentive filter aimed at the selection of the areas with a high density of edges was applied. For each of the remaining axes the best candidate area was selected among the BBs which share that symmetry axis, while having different position and size. The selection of the best BB was based on maximizing a linear combination of two symmetry measures (i.e., on the gray-level values and on the gradient values), masked by the density of edges in the box. In the HV step, different edges were selected and connected in order to form a contour representing the shape of the pedestrian body. The process consists in adapting a deformable coarse model to the BBs. The model adjustment was done through an evolutionary approach with a number of independent pixel-sized agents (ants) acting as edge trackers. This method allows the identification of pedestrians in various poses, positions and clothing, and it is asserted to be not limited to walking people. It detected pedestrians in a range of 10 to 30 meters, but these preliminary experiments were performed mainly on synthetic images.

In (Bertozzi *et al.*, 2003b) and (Bertozzi *et al.*, 2003a), Bertozzi et al. have also considered the possibility of using **an infrared camera** for the pedestrian detection task. They utilize the fact that humans from most view points are also symmetrical in IR images and use the size and aspect ratio of a full standing human to separate pedestrians from other objects in the scene. The HG step was adapted from the visible one presented in (Bertozzi *et al.*, 2002b) and (Bertozzi *et al.*, 2003c). The algorithm was divided into the following parts: first, the localization of ROIs (attentive vision) and generation of possible candidates based on symmetry (on the input and vertical edges images) was performed; then, candidates were filtered out based on specific aspect-ratio and size constraints (the size of a pedestrian was considered to be 180 cm +/- 10% height and 60 cm +/- 10% width) and finally, candidates were validated on the basis of a match with a simple morphological model of a pedestrian. The approach does not use the temporal information, which probably would have made their system more accurate. The major problems with the system are: (1) in the presence of a complex background artefacts or objects other than pedestrians were detected, (2) the detection of pedestrians was not missed but the algorithm miscalculates the exact position or size of the BB, and (3) walking pedestrians were sometimes not detected due to aspect ratio constraints.

In (Broggi *et al.*, 2004a) the improvements of the system described in (Bertozzi *et al.*, 2003b) and (Bertozzi *et al.*, 2003a) for the detection of pedestrians in far IR images were presented. Differently-sized BBs were placed in different positions in the image and the presence of a pedestrian inside those BBs was checked for at close and far away distance. In the HV step a match with a set of 3D models encoding shape and thermal patterns of pedestrians was used to remove candidates that did not present a human shape. The 3D models represent different postures and viewing angles of the human shape and were generated from different points of view. A set of 72 configurations were chosen by combining 8 different points of view with 9 positions (one standing and 8 walking). These configurations have also considered the actual viewing angle, orientation, and height of the

camera on the test vehicle. A cross-correlation function was used for the match and the result was a percentage rating the quality of the match. The system has been tested on a 1.8 GHz Athlon XP (FSB 266 MHz) with 512 MBytes DDR 400 MHz; the time required for the whole processing was asserted at 127 ms. Some assumptions have been made: the pedestrians were not occluded, the complete shape of the pedestrian appears in the image, a number of pedestrians appear simultaneously in the image but they do not occlude each other. The authors mentioned that the localization of pedestrians was difficult in some situations such as bikers, running people, or when the BB was not precise.

Also based on symmetry, but used for vehicle detection is the system presented in (Toulminet *et al.*, 2006). It is the result of the collaboration between INSA and University of Parma and it comprised a stereo vision system for the detection and distance computation of the preceding vehicle. The 3-D vertical edges corresponding to obstacles were extracted in a first step; then, a symmetry operator investigated 4 images: the one containing the 3-D features previously computed, the grey level image and the images of horizontal and vertical edges. A match against a simplified model of a vehicle's rear shape allowed the detection of a preceding vehicle and the computation of its distance. The presence of two corners representing the bottom of the BB around the vehicle was checked by a pattern matching technique. In addition, size constraints were used to speed up the search.

Vertical/Horizontal Edges

In (Broggi *et al.*, 2000a) a method implemented on the ARGO vehicle for detecting the pedestrian shape was performed by the processing of images acquired from a vision system installed on board of the vehicle: the analysis of a **monocular visible spectrum image** delivers a first coarse detection, while a distance refinement was performed with a stereo vision technique. Pedestrians were detected through a search for objects featured by specific characteristics, like: mainly vertical edges with a strong symmetry, size and aspect ratio satisfying specific constraints, and generally placed in a specific region. First a ROI was identified based on perspective constraints and the vertical edges were extracted. After eliminating the objects belonging to the background from the vertical edges map by a logical bitwise "and" with a binary mask (obtained from the application of a positive threshold to the signed difference between the left image and a properly shifted version of the right image), the areas which present high vertical symmetry were considered. Too uniform areas were discarded by evaluating the edges' entropy, while for the remaining candidates a rectangular BB was determined by finding the object's lateral and bottom boundaries and localizing by a simple correlation function the head matched with a binary pattern representing pedestrian's head and shoulders at different sizes. Distance assessment was then performed: the evaluation deriving from the position of the BB' bottom border was refined by the stereo vision technique. Finally the pedestrian candidates were filtered: only the ones satisfying specific constraints on the size and aspect ratio and presenting non uniform composition were selected and labelled as pedestrians. Temporal correlation was taken into account in certain steps by using the results from the previous frame to correct and validate the current ones. The algorithm requires the whole pedestrian to be present in the image (even if it has proven to work also when the pedestrian is partly occluded by other pedestrians), at a distance ranging from 10 to 40 meters.

Betke et al. (Betke *et al.*, 2000) have used a **colour video camera** for driving on a highway. Their system used a combination of colour, edge and motion information to detect and track two types of vehicles on the highways: distant cars detected by edges and recognized by matching templates (described in the following) and passing cars detected by temporal differencing and tracked based on motion parameters typical for cars (described in the next subsection). The authors asserted that vehicles at the far distance, usually appearing as rectangular objects, showed very little relative motion between themselves and the host vehicle, therefore any method based only on differencing image frames failed to detect these types of vehicles. They proposed a coarse-to-fine search method looking for rectangular objects by evaluating horizontal and vertical edges in the images. The horizontal and vertical edge maps were defined by a finite-difference approximation of the brightness gradient. The coarse search checked the whole image to see if a refined search was necessary, and a refined search was activated only for small regions of the image, suggested by the coarse search. The

coarse search looked through the whole edge maps for prominent edges, such as long uninterrupted edges. Whenever such edges were found, the refined search process was started in that region. In the HV step, in order to verify that the potential object was a vehicle, an objective function was evaluated: first, the aspect ratio of the horizontal and vertical sides of the potential object was computed to check if it was between 0.7 and 1.4 (considered to be potential aspect ratios for vehicles). Then the vehicle template was correlated with the potential object marked by four corner points in the image. If the correlation yields a high value, the object was recognized as a vehicle. The system outputs the fact that a vehicle was detected, its location in the image, and its size. In addition to correlating the tracked image portion with a previously stored or cropped template, the system also checks for the portion's left-right symmetry by correlating its left and right image halves. Highly symmetric image portions with typical vehicle features indicate that a vehicle was tracked correctly. Betke et al. detected also the rear lights of the vehicles to provide additional information for the identification of the object as a vehicle (detailed at the section corresponding to vehicle lights).

Parts detection or boosting-based approaches
There are systems performing a division of the image containing the obstacle into subregions, in order to separates the broad class into smaller pieces, easier to manage. The basic idea of this division is to reduce the variability of the respective class before training. The separation is done by manually clustering the training data into a number of mutually exclusive sets, where each cluster of data represents a certain pose and illumination. The early paper we found treating this subject was one to which Papageorgiou has contributed, (Mohan *et al.*, 2001) and their pedestrian recognition algorithm was inspired by papers treating the face detection problem. Shashua et al. (Shashua *et al.*, 2004) also introduce a division of the pedestrian class into subregions, while Dellaert et al. (Dellaert, 1997) have performed a search for image rows and columns that might contain edges of a car.

Mohan et al. (Mohan *et al.*, 2001) introduce a new hierarchical classification architecture called Adaptive Combination of Classifiers, composed of distinct example-based component classifiers trained to detect different object parts (i.e., heads, legs, left and right arms). The combination classifier took the output of the component classifiers as its input and classified the entire pattern under examination as either a "person" or a "nonperson". The candidate regions were first processed by the component detectors by applying the Haar wavelet transform and then the resultant data vector was classified. The component classifiers were quadratic SVMs which were trained prior to be used in the detection process. The highest component score for each component was fed into the combination classifier which was a linear SVM. This process of classifying patterns was repeated at all locations in an image by shifting the 128x64 pixel window in all possible locations from the image. The image itself was processed at several sizes, ranging from 0.2 to 1.5 times its original size. This allowed the system to detect various sizes of people at any location in an image, but the paid price was that the extensive search made the system to function not quite in real-time.

Shashua et al. (Shashua *et al.*, 2004) have designed a pedestrian detection system based on boosted learning and the pedestrian class was divided into 9 sub-regions and 4 combinations of regions. First, candidate regions were generated by filtering out areas that did not contain enough texture. Then, a single frame classification algorithm was applied on the ROI and finally, a multi-frame approval technique was used to verify the content of the surviving candidates. Dynamic patterns such as gait and inward motion were used. To represent the objects, orientation histograms (stated to be shift-invariant) have been used as features and they were extracted from each sub-region. The extracted features were each considered as weak learners and they were boosted together with an AdaBoost algorithm.

Arnell (Arnell, 2005) have the idea to divide the candidate window into regions, by the use of layers: each area in the image was included in many of the subregions; 10 subregions were first used and each subregion was further subdivided into 4 parts; in that way, 40 subregions resulted. The principal gradient direction of each of these 40 subregions was used as feature representation. The shape descriptor Arnell have used is based on the horizontal and vertical component of the gradient image.

These features have been introduced by Oren et al. in (Oren *et al.*, 1997), and were also used by Grubb (Grubb *et al.*, 2004). Arnell have improved the shift invariance by processing the gradient image components in bins. To classify a feature vector as being provided by a pedestrian, the output decisions were encoded as 1's and 0's and combined with the feature vector extracted from the candidate window into a longer feature vector which was fed into the second stage classifier. They have evaluated a SVM as a second stage classifier, but more promising results they expected by the combination of the classifiers through boosting.

Dellaert et al. from Carnegie Mellon University in the frame of NavLab project, presented in (Dellaert, 1997) a Candidate Selection and Search (CANSS) algorithm, which was able to detect in real time vehicles on highways based on contours. The algorithm uses a Hough Transform on the image gradient and a classifier initialized by training. The classifier, based on a kernel regression method was used to candidate image rows and columns that might contain edges of a car. A combinatorial search was then performed for the BB most probably (in the Bayes sense) generated by a car. In (Dellaert & Thorpe, 1997), Dellaert et al. proposed an approach for vehicle tracking, in which the 3D position and motion of a vehicle were estimated by tracking a 2D BB in the video stream, based on the CANSS model-based vehicle detection system. One year later, in (Dellaert *et al.*, 1998) Dellaert et al. integrated the 3D vehicle tracking approach with a lane following module, called Rapidly Adapting Lateral Position Handler (RALPH), obtaining a hybrid vision system that tracks both road and vehicles with higher accuracy than each of the two systems taken individually. Based on a flat world assumption, RALPH takes into account information about the yaw or lateral offset of the ego-vehicle, providing a curvature estimate. The information provided by the two systems separately was combined using an extended Kalman filter.

a2. Template matching with the obstacle's contour

This type of approach is very much employed when only monocular image is available for the detection step. Systems that can detect pedestrians in static images by employing a template matching process are described in (Papageorgiou & Poggio, 1999) and (Gavrila, 2000).

Papageorgiou et al. have presented versions of a pedestrian detection system in a number of publications (Oren *et al.*, 1997), (Papageorgiou *et al.*, 1998), (Papageorgiou & Poggio, 1999), (Papageorgiou & Poggio, 2000), (Mohan *et al.*, 2001). They used a brute force method to detect pedestrians: a search window was shifted over all possible locations in a given image. The method, as a first approach, was computationally expensive, not quite in real-time because it required 20 minutes per examined frame. In a second approach, after removing the use of colour information, reducing the amount of wavelet features (to only 29 features) and adding stereo to locate ROIs, their system operates at 10 Hz (it was implemented on Daimler Chrysler Urban Traffic Assistant (UTA)). The approach was based on shape because wavelets were used for feature representation. The features encoded local intensity differences vertically, horizontally and diagonally. The wavelets extracted important features of a pedestrians using, e.g., intensity and colour. The method of Papageorgiou et al. do not use the temporal information, and it was expected that adding such information to yield better performance (with the inconvenience of slowing down the system). The faster version of the system, the one operating at 10 Hz, was integrated in the obstacle detection, recognition, and tracking system UTA (Franke *et al.*, 1999), which was able to recognize traffic signs, traffic lights and walking pedestrians, but also the lane, zebra crossings and stop lines. It used stereo vision to detect and segment obstacles and provided an estimate of the distance to each obstacle. This information was employed as a focus of attention mechanism for the people detection system developed by Papageorgiou et al. Using the knowledge of the location and approximate size of the obstacle allowed them to target the people detection system to process relatively small regions for just a few sizes of people, instead of the entire image for all scales of people. Papageorgiou and Poggio pioneered the use of Haar-wavelet features in combination with SVM and their pedestrian recognition approach was subsequently adapted by Elzein et al. (Elzein *et al.*, 2003) and others. Grubb et al. (Grubb *et al.*, 2004) also inspired by the approach of Papageorgiou et al., but instead of using wavelets as

features, they employed vertical and horizontal Sobel edge detectors and they have replaced the SVM classifier with two SVMs, one for front-rear pose and the other for side poses of pedestrians.

Gavrila et al. (Gavrila, 2000), (Gavrila & Giebel, 2002) working with the UTA system (Franke *et al.*, 1999), have developed a pedestrian detection system based on template matching between models of pedestrian and an image transformed by Chamfer distance. The matching was made using a technique called the distance transform (detailed in (Gavrila, 2000)), which was applied to an edge detected version of the image. In the first step, contour features were used in a hierarchical template matching approach to find candidate solutions by a combined coarse-to-fine approach in shape and parameter space. Templates were build-up in a hierarchy with general templates near the root and more specific templates at leaf level. The hierarchical clustering was automatically built off-line from the available models of pedestrians (Gavrila & Giebel, 2001) and by its use the process of mapping was speed up. Then, follows a verification step where the positively matched candidates were run through a Radial Basis Function (RBF) network, based on an intensity feature representation. Tracking also was added in later versions of this system (Gavrila & Giebel, 2002): they used an alpha-beta tracker (which is a simplified Kalman filter) to estimate the object state parameters. The pedestrian module was at that time a recent addition to UTA. If the pedestrian module was used separately (the flat world assumption was made), the system ran at approximately 1 Hz on a dual-Pentium 450 MHz with MMX, but in the alternate mode of operation, the stereo-module in UTA was used to provide a ROI for the Chamfer System and enabled a processing speed of about 3 Hz. The main drawbacks of the system are: it does not detect pedestrians in low contrast, partial or complete occlusion and it generates false alarms in areas of the image where scene presents strong texture.

Fleischer et al. (Fleischer *et al.*, 2002) reported a model-based approach which uses 3D object models and *a priori* knowledge about typical positions of traffic signs and vehicles on the road in order to detect and track such objects within image sequences. The method was integrated into Driver Assistance using Realtime Vision for INnercity areas (DARVIN), a machine-vision-based system equipped with stereo CCD cameras, used to track lane boundaries and lamp posts located next to road borders. The geometric models of traffic signs and vehicles were projected into the image plane of the camera (the pinhole model of the internally and externally calibrated cameras rigidly attached to the testvehicle was used for this projection) in order to detect image features. Based on a position estimate of the host vehicle with respect to the scene coordinate system (resulting from a GPS-based initialization step or a previous road tracking step) the model line segments of the road on which the host vehicle was driving were projected into the image planes of the divergent binocular camera setup. These projected model lines were then matched to some extracted edge elements (obtained from the greyvalue images using a gradient based method) that were likely to correspond to the image features of road borders and lane markings. Expected edge features of an object were defined by an object model that comprises bordering line segments of a set of polygonal patches used to approximate the 3D shape of a vehicle or the flat shape of traffic signs. If the coverage of a projected traffic sign or vehicle model by extracted edges exceeds a specified threshold, the estimated position resulting from the detection step and the object model were used to start a tracking process based on Kalman filtering.

b. Characteristics specific only to pedestrian or vehicle class:

Vehicle Lights
In (Betke *et al.*, 2000), rear-light detection was performed for vehicle recognition purposes: the algorithm searched for bright spots (i.e. it was looking for a pair of bright pixel regions in the tracking window that exceeded a certain threshold) in image regions that were most likely to contain rear lights. To find the centroid of each light, the algorithm exploited the symmetry of the rear lights with respect to the vertical axis of the tracking window, which means that the algorithm found either rear lights (if turned on) or rear break lights (when used). In order to reduce the search time, only the red component of each image frame was analyzed. Using additional information like the rear lights of a tracked object could be very helpful in particular situations, such as reduced visibility driving (e.g., in a tunnel, at night, in snowy conditions).

Legs detection or human gait detection

Methods based on gait recognition show a higher robustness, but they require the analysis of multiple frames; generally, they are applied only to pedestrians crossing the street in the path of the vehicle, where the alternating movement of legs is more obvious. Generally, the human motion is exploited in the recognition phase of the obstacles, but there are also authors employing human gait as a method of pedestrian detection.

The visual interpretation of biological motion has been first investigated by Hoffman and Flinchbaugh, in (Hoffman & Flinchbaugh, 1982), where anatomical constraints on how the limbs of animals typically move during ambulation were exploited in order to develop an interpretation scheme based on the assumption of planar motion. The biological motion of humans has been the subject of multiple papers, like (Viola & Jones, 2001),(Viola *et al.*, 2003), (Curio *et al.*, 2000), (Cutler & Davis, 2000), (Heisele & Wohler, 1998), (Wöhler & Anlauf, 1999) which were inspired by the work of Hoffman.

The dynamic pedestrian detector built by Viola et al. in (Viola *et al.*, 2003) is an extension of the rectangle filters presented in (Viola & Jones, 2001) used for the static face detection problem. Viola et al. extended the filters to act on motion pairs: the idea they employed was to extract a mask displacement, i.e. a succession of boundaries from a sequence of images and then to analyze those masks. The features were extracted from two successive images and the information taken into account was not only the movement but also the intensity of the images; thus, the system integrated image intensity information with motion information. The differences between region averages at various scales, orientations, and aspect ratios were measured by evaluating the motion filters as well as appearance filters using an integral image. The set of filters allowed the comparison of the values of the pixels in each image and the observation of the spatial and temporal evolution of the pixels. The filters were set to correspond to possible movements of the pedestrians.

The system developed by Viola et al. works directly with images extracting short term patterns of motion, as well as appearance information, to detect all instances of potential moving pedestrians. The training process uses an AdaBoost classifier to select a subset of features, i.e. a linear combination of the selected features. The resulting classifier employed intensity and motion information in order to maximize detection rates by a cascade architecture which make the detector extremely efficient: simpler detectors (with a small number of features) were placed earlier in the cascade, while complex detectors (with a large number of features) were placed later in the cascade. The method used by Viola et al. demonstrates that it is possible to detect pedestrians with low computation time (about 0.25 seconds to detect all pedestrians in a 360 x 240 pixel image on a 2.8 GHz P4 processor) based on simple characteristics. The combination filter and cascade classifier was effective and it was asserted to solve the problem of moving pedestrian detection. In addition, the stated results were good and they promised real-time applications. However, this method can only work on fixed cameras, and two successive images must be taken under the same conditions of acquisition. This limitation does not allow us to consider an vehicle on-board embedded application. Moreover, as the author points out, this method does not solve the problem of possible occlusions and either the static pedestrians were detected.

An interesting and different piece of work is that of Curio et al. (Curio *et al.*, 2000) which combined texture, contour matching, and IPM information into a temporal dynamic activation field in which a final decision about a reliable ROI was made using an activation threshold. The initial detection of pedestrians was performed by combining three information cues: first, the local image entropy (texture information) was calculated to reduce the search space only to structured regions and secondly models were matched based on contour information. To increase the detection performance for the short distance field, beside a monocular camera, a binocular vision system was employed. Using the IPM and the camera geometry, the estimation of object scales in the image has been done. They restricted the detection of pedestrians to the lower part of their body (i.e., hip and

legs), therefore the model resembles a down-faced V form in different deformations corresponding to the various phases of a gait, which were generated by sampling the synthetic kinematic model equidistantly in time. Different phases of the walking sequence were matched to given contour features using the Hausdorff-Distance as a measure of similarity. To reduce the search space between successive frames (i.e., for prediction of the new search region), a Kalman filter with an underlying simple accelerated movement model was used.

A drawback of the method employed by Curio et al. is that the detection is delayed, since several frames were needed to establish a walking pattern. The interesting part of this work is the utilization of a walking model, which indeed represented a strong cue when detected, but the approach had to be combined with other methods, since this cue only recognize pedestrians whose legs were visible and only when moving lateral in the scene.

Periodicity of the human gait could be also recognized with traditional methods like the Fourier transform. Some systems performed a frequency analysis of the changes of candidate patterns over time and then selected those that show the frequency spectrum characteristic of human gait. As an example, Cutler and Davis (Cutler & Davis, 2000) used a short-time Fourier transform with a Hanning windowing function to analyze the signals obtained by correlation of the pattern of detected objects. The system developed by Cutler et al. (Cutler & Davis, 2000) measured periodicity directly from the tracked images and it worked even on low resolution and poor quality images, as stated by the authors. The algorithm measured periodicity and it was comprised by two parts: first, the motion was segmented and the objects were tracked in the foreground; then, each object was aligned along the temporal axis (using the object's tracking results) and the object's self-similarity as it evolved in time was computed (the tracking step). For periodic motions, the self-similarity metric was periodic and they applied time-frequency analysis to detect and characterize the periodicity. Cutler et al. stated that their method for detecting and analyzing periodic motion of humans can be used for both static and moving camera.

Another system employing the analysis of motion by the Fourier Transform is the one developed by Heisele and Wohler (Heisele & Wohler, 1998). Their system recognized pedestrians using colour images provided by a moving CCD camera. The images were segmented using a colour/position feature space into region-like image parts. In order to determine if a cluster belonged to the legs of a pedestrian a quadratic polynomial classifier checked for periodicity in the temporal shape variation of a cluster. To extract the dominant frequency, a Fast Fourier Transform with a time window of fixed size was applied to the signal. The regions were normalized in size and finally classified by a Time Delay Neural Network (TDNN) with spatio-temporal receptive fields. The input data of the TDNN were temporal sequences of gray valued image regions selected by the polynomial classifier. In the system developed by Heisele et al., pedestrians were approximately 100 pixels in height. These image qualities and resolutions are typically not found in surveillance applications, where low-resolution cameras are employed. Another drawback of their system is that because the segmentation is based on colour, accurate segmentation methods are needed to isolate the foreground and background information. A system which continues the work of Heisele et al. is that of Wohler et al., but here the detection was based on a stereo camera system. After the objects were detected, they cropped the lower half of the ROI delivered by the stereo algorithm, which contained the pedestrian's legs, and normalized it to a size of 24x24 pixels. They replaced the TDNN with an adaptable TDNN algorithm, and the time-delay parameters of the network were learned from the training examples instead of being determined by manual adaptation.

Conclusion to knowledge-based methods: From the presented algorithms applicable to both pedestrian and vehicle shape (**case a**), we believe the most promising results could be obtained from the representation of objects on the basis of a vector of features extracted from the obstacle image, and not by a matching correlation with the entire model of the obstacle due to possible occlusions. Also, we believe methods performing a division of the obstacles in sub-regions are very promising, because they do not only serve to simplify the representation, but they can also be used to detect

partially occluded pedestrians. If only the upper regions are classified as pedestrian, the respective pedestrian detected may be classified as an occluded one instead of being rejected as not-pedestrian. Due to the presence of symmetries or edges of objects, the methods employing symmetry or edges detection in the HG step, do not limit the detection to only moving objects; still, the objects do not have to present strong occlusions (maybe not more than 25% from the object to be hidden) in order to match the obstacle model. Approaches based on a specific characteristic of objects (**case b**) are generally employed when a single type of object is searched for or when the recognition of different types of objects presenting different characteristics is desired. The methods based on human gait detection or recognition are a little bit slower than some other approaches because multiple frames are needed to be processed until the system decision is provided.

Even the approaches presented by now (which employ some knowledge about the obstacle to be detected) use a single image (or a sequence of images for the gait recognition) to perform the detection, we believe some of them have limited employability. Systems employing local symmetry, corners, or texture information for HG are effective in relatively simple environments with no or little clutter; employing these methods in complex environments (e.g., when driving in dense city traffic), would introduce many false positives. Utilizing horizontal and vertical edges for HG is probably the most promising knowledge-based approach reported in the literature. An important inconvenience is that it depends on a number of parameters (e.g. the thresholds for the edge detection step, the thresholds for choosing the most important vertical and horizontal edges) that could affect system performance and robustness; a set of parameter values might work well under certain conditions, however, they might fail in other situations. If some characteristics of the obstacles presented in the **case a** could be generalized on all classes of objects aimed to be detected and recognized by the system, and in addition if a relationship between the set of correctly operating parameters and different possible situations of day and night could be identified, then these approaches represent a possible solution for our obstacle detection system.

On the other side, using specific characteristics of a certain object type (**case b**) would not be very useful for an obstacle detection module, but multiple benefits could be added to the recognition module. By identifying some characteristics of certain classes of objects, a better discrimination between different types of objects could be obtained, much improved than using a general characterization of all types of objects. Employing shadow information, vehicle lights, legs or human gait detection for HG have been exploited in a limited number of studies. An important drawback of approaches based on legs or human gait detection is their inability to correctly classify still persons as pedestrians, because they can detect only moving obstacles. On the other hand, shape-based approaches are more sensitive to false positives and thus they need a good detection phase; at least, they correctly recognize even stationary people. Under perfect weather conditions, HG using these type of information can be very successful, but in bad weather or poor illumination conditions, when road pixels could become quite dark, this method is very possible to fail.

2) **Motion-based methods**

All the cues discussed so far used spatial features to distinguish between obstacles and background. Another cue that can be employed is the relative motion obtained via the calculation of optical flow. Pixels on the images appear to be moving due to the relative motion between the sensor and the scene. The vector field of this motion is referred to as optical flow. Optical flow can provide strong information for HG. To take advantage of these observations in obstacle detection, the image is first subdivided into small subimages and an average speed is estimated in every sub-image. Sub-images with a large speed difference from the global speed estimation are labelled as possible obstacles. Most of these methods compute temporal and spatial derivatives of the intensity profiles and, therefore, they are referred to as differential techniques.

The system developed by Betke et al. (Betke *et al.*, 2000) evaluates several consecutive image frames and employs the tracking capabilities to recognize passing vehicles. Large brightness changes over

small numbers of frames were detected by differencing the current image frame from an earlier frame and checking if the sum of the absolute brightness differences exceeded a threshold in an appropriate region of the image. If large brightness changes were detected in consecutive images, a gray-scale template of a size corresponding to the hypothesized size of the passing car was created from a model image and it was correlated with the image region that was hypothesized to contain a passing car. The normalized sample correlation coefficient was used as a measure of how well the region and the template image correlate or match eachother. Generally, when multiple frames are processed, immediate recognition from only two images is very difficult and only works robustly under cooperative conditions (e.g., enough brightness contrast between vehicles and background).

Demonceaux et al. (Demonceaux & Kachi-Akkouche, 2004) propose an approach to OD based on three steps (road detection, road motion estimation and detection of road obstacles). The road motion estimation was performed using wavelets analysis of the optical flow equation. To detect the obstacles which have small speed, they modelized the road velocity by a quadratic model. Then, to achieve a robust algorithm, a fast bayesian modelization was used instead of a simple threshold between the expected and real velocity fields. Their system was able to detect all types of obstacles in the presence of shadows, occlusions and even in the case of illumination changes.

The method proposed by Elzein et al. (Elzein et al., 2003) was also based on the principle of optical flow and it was searching for image areas containing motion. Compared to the method presented by Viola et al. (Viola et al., 2003), the processing were a bit more expensive, since the goal was not to extract features, but to define a ROI. Their main focus was the calculation of different relative velocities of objects in the scene, followed by an algorithm for pattern recognition which was applied to determine the presence or absence of a pedestrian in the ROI. To detect moving objects in the video sequences, Elzein et al. used the correlation-based optical flow estimation method of which goal was to place a BB around clusters of pixels that presented significant motion. To find such a BB, they first computed the difference between successive frames and retained only those pixels that shared a relatively large difference. For each computed BB, the overcomplete Haar wavelet transform was computed and a feature vector was constructed. The method used is the same proposed by Papageorgiou et al. in (Papageorgiou & Poggio, 1999). The authors assert that the overall computation time was about 95 seconds per frame: the computation time for motion detection was 55 seconds per frame (of which roughly 35 seconds was due to computing the optical flow) and the classification required about 10 seconds for each scale checked (4 scales were used). To these values, the computation of the wavelet features was added, and the resulted time was 336 seconds.

An approach to segment moving objects from images taken with a moving camera was presented by Arnell and Petersson in (Arnell & Petersson, 2005). The segmentation algorithm was based on a different representation of optical flow: the u-disparity was used to indirectly find and mask the background flow in the image, by approximating it with a quadratic function. The Sum of Squared Differences (SSD) was used to compute the optical flow. The algorithm was stated to provide excellent results at lower speeds (under 40 km/h); it successfully segmented moving pedestrians with few false positive, which were due to poles and organic structures, such as trees. The algorithm was intended to be used as a component in a detection/classification framework and the complementary use of stereo segmentation was proposed. Occluded objects presented no problem, since no assumptions were made about the objects shape.

Conclusion to motion-based methods: Generally, motion-based methods can detect objects based on relative motion information. In the presence of shocks and vibrations, caused by mechanical instability of the camera, a high frequency noise is introduced to the intensity profile; in general, errors introduced by shocks and vibrations are small if the camera is mounted on high quality antivibrating platforms and the vehicle is moving along usual roads. Motion-based approaches use temporal information and have proved to be quite reliable if one wants only to find a moving object and not its precise velocity. Unfortunately, it does not detect standing pedestrians or any static obstacle in general and needs to analyze a sequence of a few frames before giving a response.

3) **Stereo-based methods**

Another important contribution to vision-based OD, beside the use of shape or motion information, is provided by stereo vision. It computes depth by triangulation of matched image features in a left and right camera image. Generally, stereo vision is used as a HG method, but there are systems employing it only to provide the size/scale of the objects to be checked in the verification stage (i.e., HV step).

The stereo vision system was designed according to the model of human perception, because humans have the ability to perceive the environment visually, by locating objects in space. Using a pair of cameras two types of information can be extracted: the visual information describing the object and the objects' position in the real world coordinates. The depth information is obtained by calculating the disparity: for a given pixel, the difference in position between the right and the left images are computed. By using the stereo vision technique, it is possible to define precisely the position of the objects observed in the scene. Being a method based on distance, it is possible to remove background objects by defining regions that can distinguish different objects based on their distance from the camera.

The work of Zhao and Thorpe (Zhao & Thorpe, 2000) in the pedestrian detection domain by the use of stereo has been a source of inspiration for many authors. Their system is aimed at transit buses in urban scenes and besides the use of stereo to make foreground/background separation, it is using Neural Networks (NNs) for classification. The system first separated the image into sub-areas by the stereo information, and each depth in the image got its own BB. The BBs were preprocessed by the size ratio of a human. Small areas close to each other with similar disparity values were grouped if their combined size complies with the human ratio. Large areas were searched with a window of human size; if nothing was found the respective area remained unchanged. The pre-processed regions were then fed into a three-layer feed forward NN trained with the back-propagation algorithm. Zhao and Thorpe report a high detection rate (without using any motion cue), and their system runs at 3 to 12 Hz, depending on how many pedestrians were present in the scene. Zhao and Thorpe's approach is today somewhat outdated, but their work has been a corner stone in the development of a robust pedestrian detection system.

In (Bertozzi *et al.*, 2003c) a stereo refinement method was used to improve the pedestrian detection algorithm presented in (Bertozzi *et al.*, 2002b). This stereo technique was used to refine the computed BBs: for each BB from the list generated in the HG step, starting from a rough estimation of the distance, a portion of the other image was searched for areas exhibiting a content similar to the one included in the BB by means of a correlation measure. Once that this correspondence was found, a triangulation was used to determine the distance to the vision system. As concerning the HG step, the list of the candidate BBs was obtained in the same manner as in (Bertozzi *et al.*, 2002b), by maximizing a linear combination of two symmetry measures masked by the density of edges in the box. Due to the knowledge of the system's extrinsic parameters together with a flat scene assumption, the search for possible pedestrians was limited to a reduced portion of the image. To filter out some false detected pedestrians, aspect constraints filters were used: a pedestrian was supposed to have an average height of 1.70 m with a standard deviation of 0.1 m, and a width correlated to the height by a weighted value of 0.3. A Kalman filter has been used in the final stage of the algorithm to reconstruct an interpretation of the pedestrians positions in the scene (Bertozzi *et al.*, 2004).

Grubb et al. (Grubb *et al.*, 2004) have presented a pedestrian detection system that also utilizes stereo in the HG step. The disparity map was processed with v-disparity, where height and width of objects that "stand out" in the scene were estimated. The candidates found by the mechanism were pre-filtered also based on a human ratio. On the surviving candidates, feature extraction was performed using wavelet transform. Pedestrians from the side and front/rear pose were classified

using SVMs, one for each pose, both inspired by Papageorgiou et al.(Papageorgiou & Poggio, 1999). Temporal analysis was added in the form of a path prediction and target tracking. The system of Grubb et al. operated at 23 Hz and reached detection rates of up to 83% with false positives at 0.4%.

Thermography (thermal imaging) is mainly used in military and industrial applications but the technology is reaching the public market in the form of infrared cameras mounted on-board of vehicles due to the massively reduced production costs. Evidences of this affirmation are the systems presented in what follows.

Honda has developed an intelligent night vision system (Honda, n.d.) using two far infra-red cameras which were installed in the front bumper of the vehicle. The target distance was acquired by the stereo infra-red vision system composed from two calibrated IR cameras. This system is intended to provide some visual and audio cautions (when it detects pedestrians in or approaching the vehicle's path) for the driver in order to help him during the night driving. When implementing the same system but using visible spectrum cameras, the information retained is expected to be much more reduced. An important number of accidents are happened during night, maybe due to the driver's possible fatigue, difficulties to see obstacles on time (to react and avoid them) or even difficulties to see them on night (people wearing dark clothes or not-signalized vehicles or carriages).

Another recent innovation to help drivers see better at night and in the most diverse weather conditions, is the "BMW Night Vision" system (FLIR Application Story, 2009). Due to its long range detection capability (up to 300 m for a human being, more than 800 m for a 2.3x2.3 m object), BMW Night Vision provides a time gain of about 5 seconds at a speed of 100 km/h compared to high beam headlights. This means that drivers have more time to react and can avoid accidents.

In the case of the last two systems, the IR cameras have been utilized directly for the visualization, their main purpose being to detect and highlight pedestrians close to the road and bring them to the driver attention.

Conclusion to stereo-based methods: The main inconvenient with the stereo methods is that they are computationally complex and they are sensitive to vehicle movements and possible vibrations of the cameras. Still, they can detect all types of objects, even the occluded ones, static or moving, based on their distance with respect to the system.

How are the requirements fulfilled?

In the following, we mention how the systems using a single type of passive camera (which could be based on knowledge, motion or stereo information) obey the four requirements from section 1.3. The system cost (R1): the fact that these systems employ a single type of passive sensors makes that the price of such a system, especially in the case of the visible spectrum camera to be the lowest possible from all the systems we presented. Using the infrared spectrum technology instead of the visible one, the system cost will be increased, but still it will be lower than any other system using the active technology. When the system is equipped with a stereo configuration, the system cost (especially when the IR technology is employed) will be increased once again. In addition, the interference problems are no longer a drawback of the system due to the use of passive technology. The real time request (R2) is fulfilled by almost all these systems, but possible problems could appear when stereo or motion information is used and the processing (the respective algorithms) are not optimized. By using a single type of camera, the efficiency of the system (R3) is not as good as in the previous case, for example, but still satisfactory results could be obtained. Robustness (R4) is also decreasing comparing with the previous systems due to the limited possibility of these systems to function; they could be able to provide useful information only in specific cases, i.e. the visible camera(s) on daytime, while the infrared camera on night.

As we mentioned at the begining of this section, all reviewed systems using only passive technology

are summarised in table 2.2. They are grouped about the used information and how is the hypothesis generated and verified is mentioned in the last two columns.

2.2.3.2   Systems using a combination of different passive sensors

There are situations that a system based just on the infrared information could not handle very well, for example: during a hot sunny day, it will highlight almost the entire image, so it will provide a lot of hot areas or objects (in this case even the pavement will be seen as emitting heat). Therefore, we could conclude that none of these two individual systems would perform very well in all situations. But, if they would be combined in the frame of a VIS-IR fused system, so a system having a VIS spectrum camera and an IR one, then much more complementary situations will be faced and solved.

Even the components of our system are easily anticipated in this step, in the following, we detail some characteristics of the visible and infrared cameras in order to show that these two sensors share many complementary features. Visible and infrared images differ especially because the visible spectrum camera registers the obstacle's reflected light while the infrared camera registers their heat emitted in the scene. In an IR image, bright regions correspond to heat, while in VIS images bright regions correspond to the amount of the reflected light. The complementary characteristics of VIS and IR cameras make them to be proper for the bimodal VIS-IR fusion. Many times, the use of one single sensor is not very useful for an ODR system, because neither VIS nor IR sensor provides enough information about the surroundings in any poor illumination or bad weather conditions.

In (Bertozzi *et al.*, 2006) a tetra-vision (4 cameras) system for the detection of pedestrians by the means of the simultaneous use of one far infra-red and one visible cameras stereo pairs is presented. The two stereo flows were independently processed and then the results were fused together. The main idea of the authors was to exploit both the advantages of far infra-red and visible cameras trying at the same time to benefit from the use of each system. The system has proven to be able to detect more than 95% of pedestrians up to 45 m and more than 89% up to 75 m. The system knows the dimensions and distances of every obstacle detected, so other possible processing (e.g., aspect ratio verification) based on this information are also possible.

How are the requirements fulfilled?

By combining both, VIS and IR cameras in a single system, many benefits could be added to the systems presented in the previous case. The system cost (R1) will be a little bit higher than in the previous case, but still will be below the cost of a system employing an active technology. Still, the interference issues are no longer present. The real time request (R2) is fulfilled as in the case of the previous case of systems, because parallel processing is possible. The efficiency of the system (R3) will be very much improved compared to that of the systems employing a single type of camera, because by using the complementary VIS and IR information the systems will present a higher possibility to correctly detect and then recognize obstacles from the road. Robustness (R4) is also very much increased by the use of this information because such a system will be able to work even it is day or night. Therefore, many complementary situations could be covered and handled by such a system.

2.3   Conclusion

After we have reviewed several examples of systems using solely active or passive sensors, or a combination of them (through the active-passive fusion) and notice which their performances and disadvantages are, we can draw the following conclusions:

Table 2.2: Systems using only passive sensors

HG method	Obj type	Reference	system	HG	HV
Knowledge based	VEH	Bertozzi et al., 2000	→ mono+stereo VIS	edges,4 symmetries,bottom corners	distance by stereo,aspect ratio
		Broggi et al., 2004b	→ mono VIS	edges,3 symmetries,shadow detection	distance, size,aspect ratio,head-shoulders template
		Betke et al., 2000	→ mono VIS	edges (for distant cars)	aspect ratio, template matching
		Dellaert, 1997	→ mono VIS	contour, edges of a car	Hough transform on image gradient+kernel regression
		Fleischer et al., 2002	→ mono VIS	edges, 3D object model	matching by greyvalue+gradient method
	PED	Bertozzi et al., 2002b	→ mono VIS	v.edges symmetry	size,distance,human contour by ants
		Broggi et al., 2000a	→ mono+stereo VIS	v.edges symmetry and density	size,aspect ratio,head-shoulders template
		Bertozzi et al., 2003b	→ mono IR	v.edges symmetry and density	size,aspect ratio, morphological model
		Broggi et al., 2004a	→ mono IR	BB-template matching	size,set of 3D models
		Mohan et al., 2001	→ mono VIS	boosted: 4 regions, shifting windows	haar wavelet+SVM
		Shashua et al., 2004	→ mono VIS	boosted: 13 subregions, textured areas	gait+inward motion,orientation histo+AdaBoost
		Arnell, 2005	→ mono VIS	boosted: 40 subregions	gradient direction+SVM
		Papageorgiou et al., 1998	→ mono+stereo VIS	shifting windows	29 wavelet features+SVM
		Grubb et al., 2004	→ stereo VIS	based on stereo	vert. and horiz. Sobel edge detectors+2SVMs
		Gavrila, 2000	→ mono VIS	edge,contour-template hierarchy matching	DT+Chamfer distance, intensity features+RBF
Stereo based	PED	Zhao & Thorpe, 2000	→ stereo VIS	stereo-based: disparity	aspect ratio, intensity + 3 layers FF NN
		Bertozzi et al., 2003c	→ stereo VIS	v.edges symmetry	aspect ratio, distance by stereo
		Grubb et al., 2004	→ stereo VIS	edgestereo-based: v-disparity	aspect ratio, haar wavelet + 2SVMs
Motion based	VEH	Betke et al., 2000	→ mono VIS	temporal differencing+tracking (for passing cars)	rear lights symmetry, vertical axis
	PED	Viola et al., 2003	→ mono VIS	rectangle filters of motion and appearance	integral image+AdaBoost
		Curio et al., 2000	→ stereo VIS	texture, legs contour matching	IPM info+similarity by Hausdorff distance
		Cutler & Davis, 2000	→ mono VIS	Fourier transf.+Hanning window+correlation	time-frequency analysis
		Heisele & Wohler, 1998	→ mono VIS	color/position feature space + quadratic plyn. classifier	FFT walking periodicity + TDNN
		Elzein et al., 2003	→ mono VIS	optical flow: areas with motion	haar wavelet + SVM
		Arnell & Petersson, 2005	→ mono+stereo VIS	optical flow: SSD + u-disparity	based on stereo

All the systems from table 2.2 are using only passive sensors

48

Active sensors:

1A. Active sensors are well suited for working in *difficult weather conditions or lighting*;

2A. Because they are *distance providers*, they are mainly used in the detection step (for detecting moving or sometimes even static objects). Still, almost all active sensors cannot detect occluded and small obstacles, due to the reduced number of measurement points provided by the sensor.

3A. They are not very well adapted for the obstacle *recognition task*, because objects belonging to different classes present characteristics very much alike (the active sensors information is not as rich in the lateral measurements as vision). However, as we saw in the systems listed earlier, many systems succeed in achieving discrimination between different objects detected. Radars are not as accurate as laser scanners in their measurements (especially in the lateral one), but similarly to laser scanners, they can provide the distance to the object, its speed, and could assure a well functioning even at night or in difficult weather conditions.

4A. Still, both radars and laser scanners are suffering the same drawbacks: possible *interferences and high purchase price* (although radars are not so expensive as laser scanners).

Passive sensors:

1P. On the other side, by means of image processing algorithms based on extracting the shape of pedestrians or vehicles, good results have been achieved with the vision systems from literature. However, image processing is computationally intensive and in critical situations like night, fog, or heavy rain these systems struggle with the same problems as the driver. Infrared cameras are also suitable for pedestrian detection or even some other types of obstacle detection. They offer the additional advantage of being able to detect and classify objects by their temperature even *in night or difficult weather conditions*. However, they are a little bit expensive than ordinary cameras.

2P. The main inconvenience of the cameras working in the visible spectrum is mainly related to their low power of detection, but this issue some research groups have solved it by using a stereo system, or by some other methods (optical flow, matching by correlation). A disadvantage of this solution based solely on vision is that a system that uses only cameras functioning in the visible spectrum will not be able to properly work at night, as active sensor would do. To solve this drawback, an infrared sensor is needed to assure the operation in difficult conditions and at night. The advantage of using an IR camera instead of a VIS one is that the first one is able from its specific functioning mode to provide possible areas of interest, such as blobs in the case of pedestrians or the area of engine or wheels in the case of vehicles. Compared with cameras operating in the visible spectrum, IR cameras are not that sensitive to the change of lighting conditions. The advantage of a passive infra-red sensor is the ability to detect pedestrian without illuminating the environment. Pedestrians are bright and sufficiently contrasted with respect to the background in IR images. Because there are other "non-human" objects (vehicles, motorcycle, houses) which actively radiate heat, they have a similar behaviour. However, people, vehicles and animals can be recognized thanks to their shape and aspect ratio even in this type of images.

3P. Passive sensors are often used not only for detection, but for the recognition of objects; this is due to their discrimination information, which is higher than in the active sensors case: the information from the lateral measurements is much richer in the case of passive sensors.

4P. An important advantage would be that there are *no interferences* when using these types of sensors and *either the acquisition price would be as increased as in the case of using active sensors*.

The main problem in using active sensors for the implementation of the ODR task is represented by the possible interferences among sensors of the same type, which could be critical for a large number of vehicles moving simultaneously in the same environment. Therefore, we concentrate on using passive sensors like cameras in order to develop the ODR system.

We detailed different systems from literature which addressed a problem similar to ours. Thus, vision-based systems have been detailed and we tried to highlight the most important aspects of image processing in this field. Although extremely complex and highly demanding image processing algorithms, computer vision remains a powerful tool for sensing the environment. It has been widely employed for a large number of tasks in the automotive field, thanks to the great deal of information it can deliver. Imaging beyond the visible spectrum is another powerful process by which obstacles could be detected in difficult illumination conditions. These possibilities lead us to the idea of employing visible and infrared spectrum cameras for our system.

In the remaining part of this thesis, our proposed system is presented. As we concluded here, two types of passive sensors are aimed to be combined, the visible and infrared spectrum cameras, in order to cover complementary situation which either system employing just a single type of passive camera could not handle. In Chapter 3 the baseline Obstacle Recognition System is presented, in the frame of an entire ODR system. This is the base system from which we started our processing and here the individual or monomodal VIS and IR systems are discussed. Next, in Chapter 4, some possible improvements are studied and implemented in order to choose the best feature vector to encode the information provided by the VIS and IR cameras. Once obtained, this feature vector could improve the accuracy of the system, but also it could decrease the processing time needed for the system in the obstacle recognition stage. Different features selection algorithms are tested and evaluated for the computation of this pertinent feature vector and finally, our proposed scheme is presented. In Chapter 5, three different fusion schemes are presented and evaluated having the main purpose the improvement of the recognition accuracy, but also the possibility to adapt the system to different context situations. Fusion is performed at different levels, low or high (by combining features, respective matching scores), but also at an intermediate level: fusion at the kernel level, which is the solution we propose for our final system. In this last chapter the monomodals systems are also brought in discussion, but the main processing is done with bimodal systems, thus combining both visible and infrared information, which uses the bimodal information at different stages, depending on the applied fusion scheme.

Part II

Our System

CHAPTER 3

Baseline Obstacle Recognition System

Contents

Like it was mentioned in the previous chapter, some systems may use active sensors as radar, lidar (or ladar) or laser scanner in order to perform or improve the detection of an obstacle. Considering the high price and the interference problems, we chose not to employ any active technology for the proposed ODR system. As mentioned in the previous chapter, the system we proposed is using only cameras, so passive sensors operating in a non-invasive way and which in addition are also cheaper than their counterparts, the active sensors. Still, it has to be considered that combining information from different sources contributes to forming a more complete image of an object to be detected or recognized in a road scene. Therefore, our proposed system employ passive sensors, which are relatively chosen to be complementary (visible spectrum and infrared spectrum cameras) because the system must work well even under difficult conditions, like poor illumination or bad-weather situations (such as dark, rain, fog). The high performance and robustness of the system will be assured by the use of these two types of information, but also by the classification with a powerful SVM classifier.

At the beginning of this chapter we present the proposed architecture for our entire Obstacle Detection and Recognition system (section 3.1). The problems addressed here are intended to make a detailed presentation of the functioning mode and of the components that form the ODR system. In the second section 3.2, the Obstacles Recognition component is more emphasized, and here the following are also presented: the image database on which the proposed schemes have been experimented, the measures by which the performances of these schemes have been evaluated, but also how the feature vector that will characterize/define each instance within the system was composed. Basic notions about the classifier used in the frame of the developed fusion schemes, which is an SVM, are also presented. Finally, experimental results are given in section 3.3 and the chapter's conclusion in section 3.4.

3.1   System Architecture

As previously mentioned, the purpose of this study was to determine if the combined use of the visible and infrared information is suitable in terms of a detection system, but especially from the viewpoint of an Obstacle Recognition module. From the development of the detection and recognition modules to their testing in the context of an intelligent vehicle running on the road, other challenges remain. For the intelligent vehicle to sense the course of the road, generally some exteroceptive sensors are used (like speedometers, odometers, gyrometers) in addition to the cameras. Many developed systems are making the assumption that the road is flat, but this assumption is not always valid, therefore other solutions for road detection and following, but also for calibrating the system are searched for. Most systems developed in this area, equipped with sensors and tested on the road, were produced by close collaboration between large research teams or university laboratories and car manufacturers (like Volkswagen, Daimler Chrysler, Honda, and others).

3.1.1   The obstacle detection and recognition system

The multiple difficulties (besides the financial ones) in developing such an intelligent vehicle equipped and ready to run on the road, made us to appeal to the images already registered by such a system. The system we point at was developed in collaboration between University of Parma and the U.S. Army, and it was intended for human shape localization with the infrared and visible light cameras. This system is a tetravision one and consists in an experimental vehicle equipped with two CCD cameras and two un-cooled infrared cameras working in the 7-14 μm spectrum. The image database we employed in our processing was registered by this system and it was provided by the Artificial Vision and Intelligent Systems Laboratory (VisLab) of the University of Parma. For more details about the aspects of the video acquisition module and the procedure used to calibrate the cameras, please refer to (Hammoud, 2009b) and (Bertozzi *et al.*, 2006).

Although the tetravision system provides four images, two stereo visible and two stereo infrared, we used in our processing for the Obstacle Recognition task only a pair of images: one visible and one infrared. However, in order that also the detection part be robust, we propose a system using two visible cameras, so a stereo system for the daytime vision and a single infrared camera for the night-vision. We believe it is not necessary to have also a stereo IR system, because stereo is generally used in the segmentation stage, and by the IR spectrum images nature itself they provide also a way to segment objects from the background based on thermal information. In addition, we do not support the idea of a stereo IR system because cameras operating in the IR spectrum are more expensive than the VIS ones. On the VIS domain, stereo is desired because the segmentation is hard or even impossible using monocular visible spectrum vision in the context of a cluttered background. The fact that we employ stereo for the VIS spectrum does not increase the system's costs, but stereo IR would double the cost of the entire vision system. Therefore, for the system we propose (a combination of two components: one stereo VIS and one monomodal IR), the implementation costs will not be very high, in contrast to the expected system's performances. In this thesis we consider only the recognition step since the detection one was already treated in the frame of our laboratory by (Toulminet *et al.*, 2006), (Cabani, 2007), (Li, 2010) and others and it is still a work in progress.

As shown in the previous chapter, an object categorization system implies two steps:

   1) in **the detection step**, a rectangular region of interest (ROI) also called bounding box (BB) is found and it is associated with a potential obstacle;

   2) **the recognition or verification process** follows, where the false alarms are removed and the object type is determined. Figure 3.1 is illustrating these two steps in an infrared-visible pair of images from the database we employed in our processing. These steps are widely described in subsection 3.1.4.
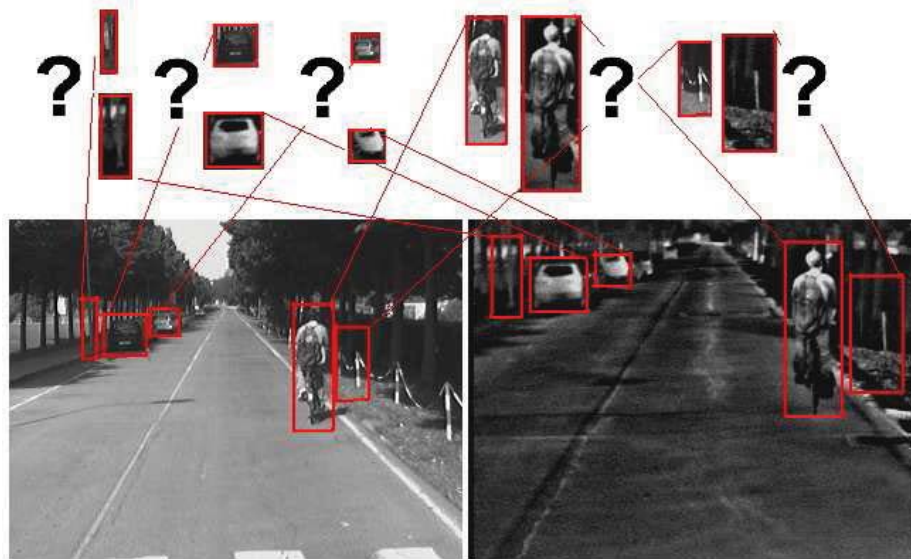
Figure 3.1: Obstacle Detection in the frame of an ODR system

Generally, the existing obstacle detection systems provide many false alarms when detecting all possible obstacles from the scene. Only in the Laboratoire d′Informatique, de Traitement de l′Information et des Systèmes (LITIS) laboratory, three VIS stereo-based obstacle detection systems (Toulminet *et al.*, 2006), (Cabani, 2007), (Li, 2010) have been developed in the last few years and they also have registered some false alarms in the detection process. Possible examples of detected BBs, including some unwanted false alarms, are presented in figure 3.1. The main component of an ODR system is the detection module, but because it has not yet reach a robust and acceptable accuracy when working alone in an autonomous system, almost all existing systems from the literature provide a second component, the recognition module. The main purpose of the recognition module is to identify the type or class of the detected obstacle, and to eliminate false alarms, so to reject them. Therefore, it helps to consider only those BBs that truly represent obstacles. Thus, the entire processing time of an ODR system will be reduced and the system will not be "strangled" with ghost obstacles which otherwise have to be taken into account when running on the road.

In the figure 3.2 is presented a scheme showing how the Obstacle Detection (OD) and Obstacle Recognition (OR) steps are arranged in the frame of an ODR system. The main purpose of the obstacle detection component is to provide the scene objects BBs, but real systems also generate some false alarms or ghost objects, as it could be seen in the figure (the BB from the left margin and the BB from the right margin). It is therefore necessary a second step, of the recognition, in which the false alarms are eliminated, and for the detected objects their class membership are provided.

For the obstacle detection task a stereo vision system has been developed in the LITIS laboratory from INSA, Rouen, France (Bensrhair *et al.*, 2002) and our efforts aim to continue this work. Our main purpose is to develop approaches to reduce the number of false alarms and to recognize the detected obstacles (like cars, pedestrians, cyclists) by the extraction of a compact and pertinent numeric signature, followed by an SVM classification. Since the OD task has been achieved in the frame of the LITIS laboratory by stereovision, we have focused on a further processing, namely the
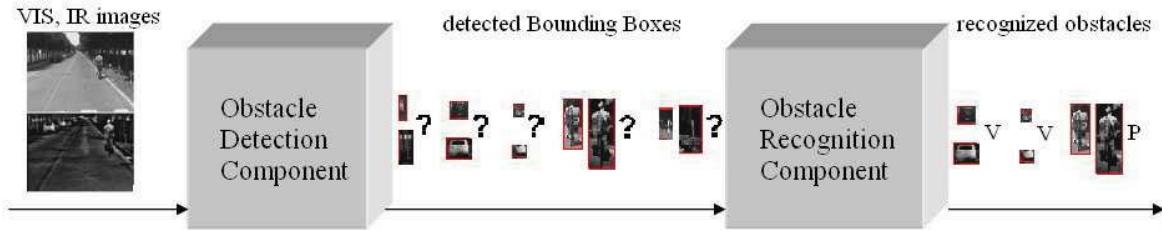
Figure 3.2: Main steps performed by an ODR system:
Obstacle Detection and Obstacle Recognition

recognition of the detected obstacles. Our main purpose was not to develop a system on the whole, but only the recognition module which was intended to be based on fusion in order to exploit the complementary information of VIS and IR cameras. Therefore, in our work we intended to verify if it worth to perform the fusion, so if a fusion will bring in benefits from the recognition point of view, besides the advantages it implies in the detection step. Most current detection systems fail to detect all obstacles in a scene, probably because they are based on only one modality, either VIS or IR. By the fusion of the information provided by two modalities, we believe that the detection results will be improved as concerns this task. Only after this study (of fusion between VIS and IR information) will be finished we expect to integrate our module into a unified system performing both Obstacle Detection and Recognition. Although the categorization of road obstacles seems to be a trivial problem, the variety of appearance of the obstacles and the cluttered background in the frame of the outdoor environment make it a true challenge.

3.1.2   How will the proposed ODR system function?

The situation in which the system is at a time, i.e. the context, could be decided by an illumination coefficient estimated by the system. So when both information are available (the stereo information from the VIS system and the information from the thermal IR sensor), the final decision could be based on the VIS or/and IR information in a more powerful or weak way, depending on the situation or the context in which the system is running at that respective moment.

For instance, in situations where the IR camera would act poorly (one hot summer day in which all objects, even the pavement, seem very hot), we enable only the stereo VIS system to provide useful information (therefore we assign more or even the whole credibility to the VIS system); in contrast, when a strong dark situation is reached, so the VIS system would not perform very well, we yield more or even all the importance to the IR sensor. In most situations (meaning those that are not extremes), we hope the fusion will demonstrate its contributions, as both systems VIS and IR will provide useful information. In these cases, the importance will be assigned in a different manner: the two systems (or to be more specific, the information provided by the two systems) will be weighted with different coefficients, experimentally determined.

Our system could be envisioned as having a set of values that specifies the system functioning mode for several possible situations, i.e. different environmental contexts. Therefore, depending on the context in which the system is intended to work (e.g. an early evening, it is still daytime but the sky is overcast and soon it will be dark), a set of operating parameters, determined *a priori*, will be selected. In order to calculate the adaptation coefficients, there should be a great database to succeed in capturing different lighting and weather conditions. Therefore, we should have images for different multiple situations: hot sunny days, almost-night situation, dense fog or strong darkness, and so on.

3.1.3   How is the context determined?

For our system, the context in which the system is functioning at a certain moment of time is determined based on the information taken from both VIS and IR images. For example, if the scene registered by the VIS camera(s) is dark (and here we suppose the camera(s)' functioning parameters were previously adapted to the lighting conditions in order that cameras provide trustful images), we know that we are dealing with a situation of night, so in this case the system will function more based on the IR camera. Unlike this situation, if all pixels from the IR scene are highlighted, it is almost sure we are in a hot-summer day situation: all objects will emit heat in a strong manner, and so the system will have to rely more on the VIS part, but still keeping into account the IR information. Therefore, from the information provided by both types of images, VIS and IR, a parameter determining the context (i.e. an illumination coefficient) will be computed and depending on its value a set of operating parameters will be chosen for a given lighting or weather situation.

Let us to suppose we have recorded images for three different situations: day, night and light fog. Therefore, we need to have more examples of images taken in these three different situations; basically, we have three different image databases (in terms of image intensity, so the content of the images is different, not their structure or their registration process). Each such database will be defined by an associated context value and each such database will have to go through a validation step and determine the best model to be used in that respective situation or that respective context.

For the Obstacle Recognition task, depending on the context, a weighting parameter between the VIS and IR information will be established. This weighting parameter between the two modalities will be determined in the system validation stage, when the search for the proper classifier model will be performed. Its determination could be done by evaluating the performances of the system when tests with different weighting values in the $[0, 1]$ domain are realized. The illumination coefficient computed in a certain lighting or weather situation is in close correlation with the weighting parameter between the modalities. For example, in a situation of night, when it is almost sure that the illumination coefficient will decide a context situation in which the processing should be based entirely or almost entirely on the IR camera, we expect that from the validation step, the weighting parameter value calculated between the two modalities to be more inclined towards the IR than towards the VIS domain. We propose a method to automatically find the adaptation or weighting parameter (between the VIS and IR information) the most suitable one for the respective environmental context.

From the viewpoint of the system adaptation to different weather and lighting conditions, one possible problem within the database we use is that there are no extreme lighting/weather situations in the scenes registered with the system and therefore our proposed solutions could not be tested under realistic conditions of night or bad weather.

Images provided by the three cameras will be correlated before the testing stage, and from the way the cameras are mounted on the host vehicle and by using markers, the correspondence between image pixels in different images can be acquired. An inherent problem is that after covering long distances, which are not always on a flat road, some problems of miscalibration of cameras could be reached, and therefore the images will be no more correlated each other. In these situations, a system involving a perfect correlation between the visible and the infrared images (as in the case of a system combining VIS and IR information at the base level, i.e. in which the fusion is performed at the lowest level, like data or pixel fusion) will be no longer trustful. For this reason, we propose in Chapter 5 different fusion schemes where fusion is performed at higher levels, like those of features, kernels or matching-scores. These proposed fusion schemes will present minor drawbacks when cameras are not perfectly correlated, so they will be not very much affected if there is no perfect correlation between the two types of cameras.

3.1.4   Problems and setup

As figure 3.1 shows, an obstacle detection and recognition system supposes the existence of two modules:

    1) the **detection module** (called also the segmentation stage because here objects are segmented from the background) - within potential obstacles are detected and assigned as "interest zones" or ROIs or BBs and

    2) the **recognition module** - in which all BBs previously determined will be labeled as belonging to one or another possible classes and the false alarms will be removed. The classifier decision depends to a large extent on the information given as input to the classifier and on the database to be learnt, but also on the functioning parameters, which have to be determined *a priori* in the validation step.

1) In the frame of our system, the **segmentation** will be realized by one of the following possibilities:

    - by using only two VIS spectrum cameras, so a stereo system - for performing the detection in daytime or hot sunny days,

    - by using only the IR spectrum camera - for the detection during night-time or in days with strong dark or dense fog,

    - all intermediate situations could be treated by performing the segmentation in parallel, for each modality; after obtaining the BBs from both VIS and IR domains, these could be fused in order to increase the detection accuracy.

Therefore, our system is designed for intermediate situations, but not leaded to extremes, where neither humans would perform well. By employing an IR camera, the proposed system is able to see beyond humans can. By the fact we propose the use of cameras functioning in two different spectrums, VIS and IR, our system will always could be seen as a bimodal one and it would be superior to a monomodal one (i.e. a system using a single type of camera).

For the detection part, as we noted above, a VIS stereo vision system has been developed in the LITIS laboratory of INSA and the detection algorithm was applied to detect both vehicles and pedestrians. In addition, also the information provided by the IR sensor could be used to perform the detection in a parallel way. So there is the possibility that at the detection level too, the VIS and the IR individual detections to be fused for obtaining better results for the entire ODR system. Instead of not detecting all obstacles from a scene, it is better to detect all obstacles and some false alarms, because these ghost objects could be eliminated in the recognition step. Therefore, we can deduce two possibilities of achieving detection:

    - detection could be done either in one modality only (the most credible for that lighting situation) and for the BBs detected in one image its correspondent in the other image will be searched for (based on the correlation information between the two types of images), or

    - detection could be performed on each modality, the results to be then fused and the BBs position in the two images could be slightly adjust, depending on the images correlation.

2) Once the system would be able to detect certain areas that are likely to contain obstacles, the next step is their **recognition**. Generally, classifier-based methods involve two steps: one of training or learning the system, in which the operating parameters of the system are also determined, and a second step, the test one, in which the system is tested with some new, unknown data, and is using the validation parameters previously obtained. To evaluate the proposed solutions, an SVM classifier was chosen, because in many applications it demonstrated efficiency, robustness and rapidity. In the training stage of the system, there is a learning module which has to be trained on a database of images, which means that the system must know beforehand some instances for each class to be detected or recognized. This step represents the training or learning phase of the system, and here the classifier will be defined together with its optimal functioning parameters. This means we have at our disposal a model that previously was capable to learn several instances of the aimed classes, for both modalities VIS and IR (correlated among themselves) and the number of instances learnt

is great enough to find examples in the model-validation step for any situation in which the vehicle can be at a given moment. Once the system was learnt, it could be used to test some other objects, unknown and this will be the testing stage.

Figure 3.3 is showing an inside view of an ODR system: training and testing stages and their modules. In this figure, if we exclude the detection module, all what remains can be framed in the recognition module.
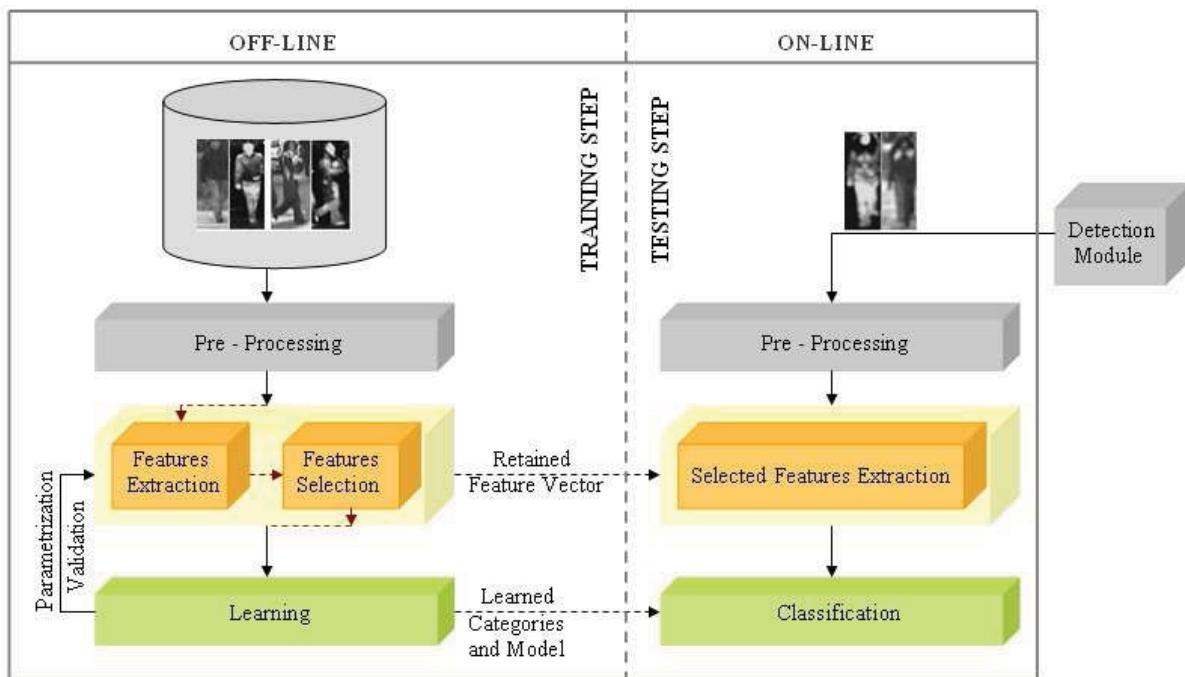
Figure 3.3: Training and testing steps in the frame of an Obstacle Recognition system

In figure 3.3 one can notice that in the frame of an Obstacle Recognition system, there are two main steps. The first one is the training step in which the database with different BBs enclosing possible obstacles (manually annotated) from the road is used. On this image database, a first module of pre-processing is noticed (within some general image processing operations are applied). The second step, the testing one comprises the same pre-processing module, but this is applied on the test image provided by the detection module, because here the system runs on-line. After the pre-processing module, in the training step it follows a features extraction and selection module which together with the last module (the learning one) has the main purpose the parameterization and validation of the system. This operation of parameterization and validation of the system consists in choosing the most pertinent features to compute a feature vector which will best characterize the data from the training database, but also in establishing the classifier which will best learn the instances of this training database. In the testing step, the feature vector used to characterize the test data will comprise the same features determined as being relevant in the training step by the parameterization and validation process. The learning module from the training step is used to learn the categories and the model of the proper classifier scheme which will be further used in the testing step to classify new test images within the classification module.

From what has been seen by now, our ODR system belongs to that category of systems based on pattern recognition and consists of three main modules:

    - sensors (visible and infrared cameras) that gather the observations to be classified or described; here it could be included also the pre-processing module,

    - a features extraction mechanism (often attended by a features selection operation) that computes a numeric or symbolic information from the observations, and

    - a classification or description scheme that does the actual job of classifying or describing observations, relying on the selected extracted features.

3.2 Obstacle Recognition component

3.2.1 Introduction

In a classical scheme of an ODR system, the sensors module is usually framed in the detection module; in the recognition stage, there is also information provided by the sensors, but here it takes the form of a training and/or testing database. Therefore, in the OR component the main parts of the module are represented by the features extraction and the learning/classification steps.

Applications based on pattern recognition aims to classify data (also called patterns) based either on *a priori* knowledge or on some statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations. Acts of pattern recognition are related with the ease with which humans recognize objects (or even sounds, letters, characters) from the real world, e.g. decide whether an obstacle from the road is a car, a motorcycle, a cyclist or a pedestrian. Not the same ease is reached when computers are aimed to perform this task. The classification scheme is usually based on the availability of a set of patterns or instances that have already been learnt and this set of patterns/instances is usually termed the training set. The resulting learning strategy is characterized as supervised learning, but learning can also be unsupervised, in the sense that the system is not provided with the labeling of patterns/instances; instead, it itself establishes the classes based on some intrinsic properties of the patterns to be classified (also called clusters in this case). How well computers succeed in recognizing patterns depend on a multitude of factors: how comprehensive is the training set (Does it cover all possible situations in which objects can appear?); How efficient is the classifier to be used (Does it succeed in learning well all the objects from the training set and then tests performed on the test set are leaded to high accuracies? What about the classification time? Is its value satisfactory from the viewpoint of a real time system?). In the frame of our system, we tried to develop an OR module to give affirmative responses to all these questions.

A wide range of algorithms can be applied for pattern recognition, from naive Bayes classifiers and neural networks (NN) to k Nearest Neighbors (kNN) or Support Vector Machines (SVM). The classification scheme usually uses some statistical (also called decision theoretic) or syntactic (also called structural) approaches. Statistical approaches are based on statistical characterizations of patterns, assuming that the patterns/instances are generated by a probabilistic system, while the syntactical (or structural) ones are based on the structural relationships between features. Given a number of "training" examples (also called samples, instances, patterns or observations) associated with desired targets (or classes), the machine learning process consists in finding the relationship between the patterns and the targets using only the training examples. This is also the case of our application: the system must learn different classes of objects (pedestrian, cyclist, vehicle) from several available examples, and then it should be able to recognize similar examples in new images, unknown. Therefore, the system goal is to predict the unknown target for new "test" examples and the resulting performance on the test data is showing how well the system is capable to generalize over new data. The training examples are used to select an optimum set of parameters.

This type of systems' learning in the image processing domain requires an additional step, namely: the extraction of some relevant information (features) from images, so that the characterization of

an image to be made with as less features as possible; this is necessary to shorten the computing time required by the system to learn and then to recognize the objects' images. A simple calculation can show us how important is this stage: an image of 128×128 pixels would have a number of 16,384 features if we choose to select as features each pixel intensity, compared with 171 features which uses our system to describe the same image. However, this operation of features extraction is a very sensitive one because it has to select a number of features great enough to assure a proper characterization of images, but also low enough to obtain a low processing time.

There are three aspects to be considered when one has to implement a recognition system based on image processing: (1) features extraction, (2) the definition of the evaluation criteria (e.g. a relevance index or the predictive power), but in our case the evaluation is based on the accuracy of the classification and the classification time and (3) the estimation of the evaluation criterion (or the assessment method) which is performed using a classifier, an SVM in our case.

From figure 3.3 one can notice that in the training stage, realized off-line, there is an image-database on which the system will be validated, that means to seek to build a model which should succeed in learning best the instances of this image-database, and then, in the test step, the same model to be used for the categorization of new images, unknown to the system. In the test stage, the used images may come also from an image-database called the test database, or they may come on-line, when the system runs on the street. In the first case, a manual annotation of images was performed, as was done for the images belonging to the image database from the training stage. In the second case, when the system runs on-line, the detection module will realize the segmentation of the objects from the road scene and will provide the BBs necessary in the next steps of the processing.

After obtaining the BBs (in both cases, the training and the test stages), it follows a pre-processing module in which generally various adjustments to the images are performed. In our case, a resize operation was carried out, to bring all the BBs to the same scale. This operation was necessary because we chose to extract features as wavelets which require that the image to be a square one.

After this pre-processing step (which in some applications can be missing) it follows a module within the feature vector characterising an object is obtained. In this module, generally two important operations are made: features extraction and features selection. The first operation extracts certain features or characteristics from the images fitting the object, so from the BBs, generally being adapted to the application and having the main purpose to retain specific characteristics to the object shape. As it could be seen in subsection 3.2.4, we have opted for the extraction of 171 features, which are general and fast to compute, in order to characterize an object in one modality (VIS or IR). The second operation (i.e. features selection) is used to reduce the size of the feature vector previously obtained, in order to reach an optimal Feature Vector (FV) in terms of accuracy and consumed time in the classification stage. By this selection of features, only the most relevant ones (for realising the discrimination between the different classes of objects) are retained and here there are several ways to accomplish this task (we will see in detail in Chapter 4). The applied features selection scheme and the most relevant features that will be retained after performing the features selection step will be decided in the training phase. In the test phase, the feature vector for the test objects will be established after the same steps and methods used in the training stage.

After forming the final FV, in the training phase all the characteristics specific to objects from the validation database will be learnt and many SVM classifier models will be constructed. A model selection is then performed, which means that from all these classifier models (obtained from different combinations of the SVM parameters, i.e. the hyper-parameters), the best model will be selected based on a 10-folds cross-validation method and having as classification criteria the accuracy and time (i.e. a bi-level optimisation). From two models that provide the same accuracy, the one providing a shorter classification time will be therefore chosen. This winner model (i.e. learned objects' categories and model) of the SVM classifier will be used next in the test step.

In order to improve the performances of the system, but also to adapt the system to various weather or illumination situations, different fusion schemes combining VIS and IR information are proposed in Chapter 5. A first fusion scheme is proposing the fusion to be performed at the features level, therefore at a low-level. This fusion would be obtained in the frame of the module which realise the features extraction and features selection operations, and for this reason, it could be performed in two possible ways: between the two modules of features extraction and features selection or after both of them. Another proposed fusion scheme is a high level one, being realised at the outputs of the VIS and IR classifiers, therefore it combines matching-scores. Two possible ways could be reached here too: a not-adaptive fusion and an adaptive fusion. The last proposed fusion scheme realizes the combination of the VIS and IR information at an intermediate level, i.e. at the SVM kernels.

In this chapter, the processing noted on figure 3.3 (image pre-processing, features extraction and features selection, learning or classification) will be realised on each individual modality, which means on the VIS and IR images separately. It is like there were two independent systems process the information in parallel. Thus, we will call them monomodal or unimodal systems.

Next, the main modules of the Obstacle Recognition (OR) component are detailed, together with their main purpose. In this section, the features extraction module (together with a widely evaluation of the proposed features that will be incorporated in a FV) and the learning scheme are detailed. Other considerations involved in the development of the recognition system will be also specified: the used image database is described (to better understand the classification problem) and the measures by which the performances of the classification process were evaluated.

3.2.2   Database we use

Most systems detailed in the literature addresses the subject of obstacles detection or classification from the viewpoint of a particular type of obstacle: pedestrian or vehicle detection/recognition is performed based on specific position, shape or features exploited by these classes of objects. Generally, these systems perform a binary classification in the recognition step: it is or it is not the object they were looking for. Still, there are documents referring to the term "obstacle" in general, and the systems described there most often were intended to detect any possible obstacle which obstruct the host vehicle path. To our knowledge there are few systems dealing with the problem of detection and classification of different types of obstacles, like pedestrian/ cyclist/ vehicle and others. This is the task we propose our OR component to solve.

In a first attempt, the developed processing schemes have been tested on two un-correlated databases: the first one was taken from the internet, but most of the images were provided from the free available Caltech database. A lot of indoor and outdoor images containing different objects could be found on this database, but we take just a few of them, mean images containing cars or pedestrians in different arbitrary poses. Details about the used database and the results obtained on it could be reached in (Apatean *et al.*, 2007) and (Apatean *et al.*, 2008c). The second database has been obtained after we have engaged in the Robin competition, where we try to test and compare the developed features extraction and selection algorithms in both types of images, visible and infrared, separately. Details about the processing but also on the image database could be reached in (Apatean *et al.*, 2008b) and (Apatean *et al.*, 2008a). Both databases were not tailored for our purpose, mainly because images from the VIS and IR databases were not correlated each other. Therefore, immediately after we obtained the third database, about we mentioned in the beginning of the subsection 3.2.1, we dropped the other two. The third, which is the last image database we employed in our processing was provided by the Artificial Vision and Intelligent Systems Laboratory (VisLab) of the University of Parma and the VIS and IR images were correlated each other. The image database has been registered with a tetravision system, but in our obstacle recognition module we employed only one pair of visible-infrared images. All the results obtained in this thesis are based on this image database.
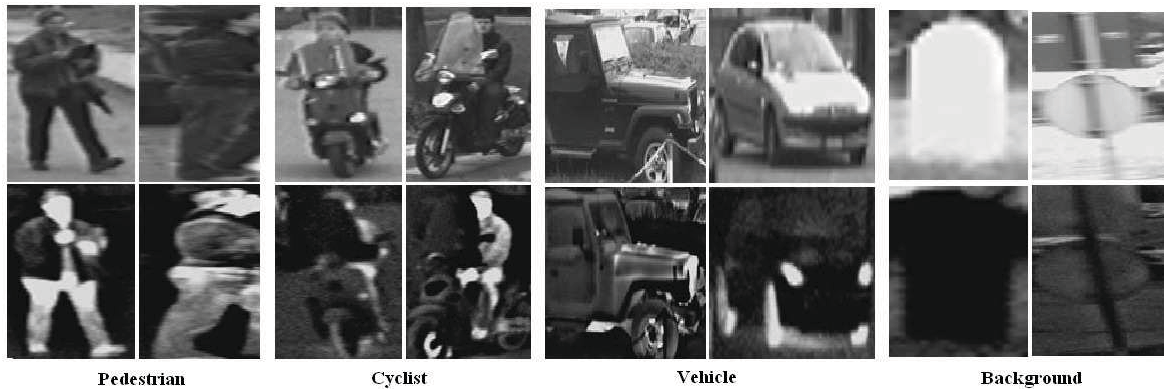
Figure 3.4: Examples of objects from the visible-infrared database

Some information about the structure of the dataset must be provided for understanding how the performance of the system was estimated. Our system performs a multi-class classification task, because three types of obstacles are aimed: pedestrians, vehicles and cyclists. To these three types of obstacles, a fourth one was added in order to anticipate the false detections provided by the obstacle detection module. This fourth type of obstacle considered in this thesis is the background class. When complex scenes are to be reached by the system, it is expected that the system provide some false alarms, like parts of some objects belonging to the background, traffic signs, barriers, or simply part of the scene-image which have no particular significance. It is inappropriate to say that there is a background object type, because in fact there are different types of objects belonging to this background class, they having different meanings; for the sake of simplicity they were grouped in a single class. Therefore, the class background was introduced precisely to anticipate these detections and to help in recognizing those types of objects when they appear as detections. Examples of objects belonging to the four classes are provided in figure 3.4.

In the provided tetravision database was noticed that the objects were almost all the time identically between the frame t and the frame $t+1$, $t+2$, and so on. A subjectivity has been introduced in the annotation process, because the human operator [1] has to select a frame t to be annotated and the next frame to be annotated could be the $t+n$ one for example. It was depending on the moment when some objects' position, pose or size was changed. Multiple objects of the same size/pose and position are for no help in the learning and classification process and this is what we tried to avoid by the manual annotation. Therefore, we tried to assure a variability of the objects from the annotated database, so the 1164 annotated objects are quite different.

Excepting the background class, all the other three classes of objects have been annotated with the following sub-classes: a first indicator F/L is showing the position of the obstacle regarding to the camera: frontal (F) or lateral (L), while a second indicator E/O/G is mentioning if the object is entirely seen in the image (E), or if it is occluded (O) or if there are multiple objects belonging to the same class very close to each other, so they will be taken as a group (G). Examples of annotation for the class pedestrian are showed in figure 3.5. Even the image database has been annotated with the first indicator F/L (therefore one can know if the object is seen from the frontal or from the lateral position considering the vision camera) this information has not been utilized in the training/classification stage. Still, it allows a detailed analysis of the obtained results.

---

[1] The tetravision image database was not previously annotated, therefore the author of this thesis have annotated it with a script developed in MATLAB.

<div align="center">P_FE   P_FG   P_FO   P_LE</div>
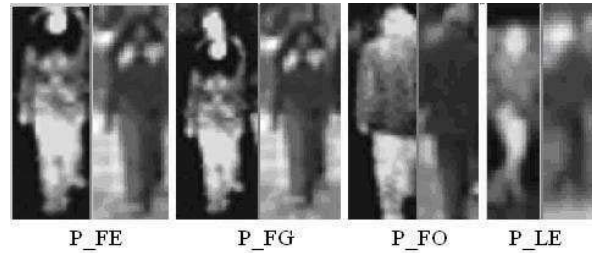
Figure 3.5: Examples of annotations for the class pedestrian

Since the image database was not annotated, we realized a manual annotation of 1164 objects, which were further randomly divided in two sets: the train set, which contains 932 objects and the test set that includes 232 objects. Objects were annotated as belonging to 8 different classes, as can be seen in figure 3.6: pedestrian entire (PE), pedestrian occluded (PO), pedestrian group (PG), vehicle entire (VE), vehicle occluded (VO), vehicle group (VG), cyclist (C) and background (B). The class cyclist (C) is also containing different poses, like occluded, group or entire, as in the case of the pedestrian and vehicle classes, but because very few cyclists have been found in the scenes we have annotated, we grouped all these in a single class of objects. From figure 3.6 one can notice the distribution of objects on these 8 classes is preserved between the train and test sets, but unfortunately between different types of objects there is not a balanced distribution of instances. For example, the class PE (pedestrian entire) is having a number of 206 instances at the train (a percent of 22% from the whole train set), but the class PG (pedestrian group) is reaching only 50 instances (5%) in the same train set. In this manner, the distribution of the objects per each class of objects pedestrian entire, pedestrian occluded, pedestrian group, vehicle entire, vehicle occluded, vehicle group, cyclist and background, briefly noted (PE, PO, PG, VE, VO, VG, C, B) will be (206, 65, 50, 133, 131, 65, 45, 237) in the training set and (51, 16, 12, 34, 33, 16, 11, 59) in the test set.
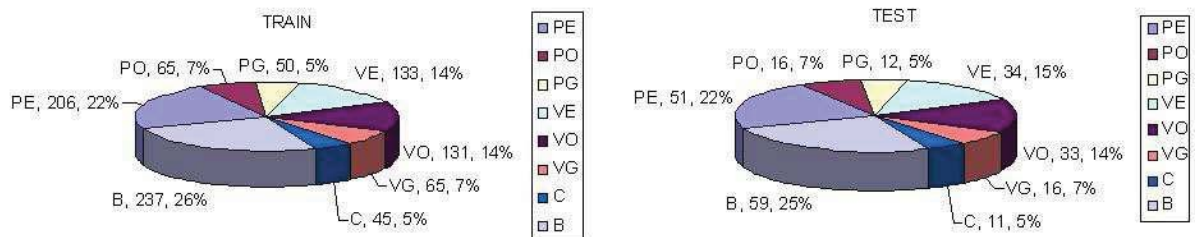
Figure 3.6: Objects distribution at train and test for the database with 8 classes

The same instances have been grouped in only 4 classes of objects (figure 3.7): pedestrian (P), vehicle (V), cyclist (C) and background (B). The distribution of the objects per each class (P, V, C, B) will be (321, 329, 45, 237) in the training set and (79, 83, 11, 59) in the test set. Thus, in both cases (the classification problem with 8 classes of objects and the classification problem with 4 classes of objects) a ratio of approximated 4 is obtained between the number of objects from the train and test sets, so we can say the instances are well balanced between the two training and testing sets. Both classification problems (with 8 and 4 classes of objects) will be addressed in this chapter and the classification results are provided in the section (3.3), where the experiments results are presented.
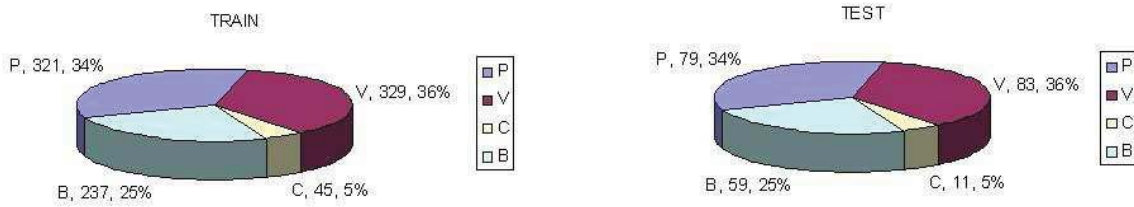
Figure 3.7: Objects distribution at train and test for the database with 4 classes

Images containing cyclists are very few: 5% from the entire database, for all the subclasses, which are cyclist entire (CE), cyclist occluded (CO) and cyclist group (CG). This means that the cyclist class will have high intra-class variability and thus it will be a class very hard to be learnt and then to be recognized by the classifier. All the other classes of objects, if we refer to the classification problem with 4 classes of objects, will have a high intra-class variability because three types of poses (entire (E), occluded (O) and group (G)) are comprised in a single class, pedestrian (P) or vehicle (V). In the frame of the class background (B), one can notice the existence of different types of elements from the road, like traffic signs, fences, barriers, and so on (i.e. signalling or infrastructure elements, but also pieces from the image which have no particular meaning). Here, in the class background, there are a lot of different images stated as "background", even they could be interpreted as obstacles (e.g. crossing barriers) because when multiple images for the same type of object will be reached in this class, the database could be refined and new classes of obstacles to be constructed in order to be individually recognised. From all these aspects, we conclude that even the database is a small one, it is also a very difficult one. Therefore, the learning and testing process will not be a trivial one. The existence of obstacles from the type occluded or group generally is due to the imperfect detection module. Ideally, if the obstacles appear entirely in the scene, the detection module will provide BBs enclosing the entire shape of those objects. However, most OD systems provide only parts of those objects in the BBs, even they are entirely seen in the scene. In a similar way, when multiple objects are close to each other, the existing OD systems provide a single BB instead of many BBs each enclosing a single obstacle. In addition, objects can appear even occluded or grouped in the scene: a pedestrian could be occluded by a tree, or many pedestrians could walk in group, very closed to each other. Therefore, we adapt our database to the obstacles types detected or provided by the detection module.

3.2.3   Performance evaluation measures

The performances of the classification are analysed by several performance measures, borrowed from the information retrieval domain. They are utilised in order to evaluate the quality of the obtained feature vectors (when a kNN was used) or the quality of the obstacle recognition process (when an SVM was utilised). These measures are detailed in the following lines.

Classification measures

In order to better explain the performance measures used in our experiments, let us consider a binary classification problem, in which the outcomes (the targets) are labelled either as positive (belonging to class y_+) or negative (belonging to class y_-) and the experiment has P positive instances and N negative instances. For the binary classification problem there are four possible outcomes formulated in a 2×2 contingency table or confusion matrix. The confusion matrix $F_{ij} = M_{ij}/m$ may be constructed for each feature and it represents the joint probability of predicting sample from class

y_i when the true class is y_j :

$$F(true, predicted) = \frac{1}{m} \begin{bmatrix} M_{++} & M_{+-} \\ M_{-+} & M_{--} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \tag{3.1}$$

where m is the number of samples (Guyon *et al.*, 2006). The outcomes and their associated terminology are:

- M_{++} is the *number of hits* in the y_+ class, or *true* ("accurate") *positives* (TP) - the outcome from a prediction is positive and the actual state or the state of nature is also positive;

- M_{--} is the number of *correct rejections* or *true negatives* (TN) - the outcome from a prediction is negative and the state of nature is also negative;

- M_{-+} is the number of *false* ("invalid") *alarms*, or *false positives* (FP) - the outcome from a prediction is positive, while the actual state of nature is negative (for example, a background object predicted as being a pedestrian, when the positive state means to detect pedestrians);

- M_{+-} is the *number of misses*, or *false negatives* (FN) - the outcome from a prediction is negative, while the actual state of nature is positive (for example, pedestrian predicted as background);

Confusion matrix has only two independent entries because each row has to sum to $F_{+j} + F_{-j} = P(y_j)$, which is the *a priori* class probability (it estimates the fraction of all samples that belong to the class y_j).

Class accuracy or conditional probability that given a sample from class y it will be really classified as class y is usually called *recall*, *sensitivity* or *true positive rate (TPR)* (also called detection rate or hit rate):

$$S_+ = \frac{F_{++}}{P(y_+)} = F(y_+|y_+) \quad or \quad TPR = \frac{TP}{P} = \frac{TP}{TP+FN}. \tag{3.2}$$

The *specificity* (or *true negative rate (TNR)*) measures the proportion of negative instances which were correctly identified as being negatives [2]:

$$S_- = \frac{F_{--}}{P(y_-)} = F(y_-|y_-) \quad or \quad TNR = \frac{TN}{N} = \frac{TN}{TN+FP}. \tag{3.3}$$

In order to better explain sensitivity and specificity, we transpose the binary classification problem to the one of recognizing pedestrians from different possible background obstacles. The diagonal elements of the conditional confusion matrix $F(y_i|y_i)$ reflect the type of errors that the predictor makes, in the following manner: sensitivity shows how well pedestrians (class $y = +$) are correctly recognized, while specificity shows how well background objects (class $y = -$) are recognized as background by the same test. An ideal predictor should achieve 100% sensitivity (i.e. predict all pedestrians as pedestrians) and 100% specificity (i.e. not predict any background object as being pedestrian). There are situations in which the cost of not recognizing a pedestrian (low sensitivity) may be much higher than the cost of detecting a background object as being a pedestrian, therefore finding a ghost object (low specificity). In our application, this choice is crucial: it is better to warn the driver for a ghost object than to ignore a pedestrian from the road. In these cases, F_{-+} type of errors (false negative) are α times more important than F_{+-} type of errors (false positive). Thus, instead of just summing the number of errors, the total misclassification cost will be calculated after the relation $E(\alpha) = \alpha F_{-+} + F_{+-}$.

---

[2]Generalization to the K-class case is obvious.

The *false positive rate* (*FPR*) is the proportion of negative instances that were erroneously reported as being positive:

$$S_{+-} = \frac{F_{+-}}{P(y_-)} = F(y_+|y_-) \quad or \quad FPR = \frac{FP}{N} = \frac{FP}{TN+FP} = 1 - TNR. \tag{3.4}$$

Positive predictive value (*PPV*) (which is also known as *Precision*) and *negative predictive value* (*NPV*) are defined as:

$$PPV = \frac{TP}{TP+FP} \quad and \quad NPV = \frac{TN}{TN+FN}. \tag{3.5}$$

Standard classifier *accuracy* (*Acc*) is obtained as a trace of the $F(y_i, y_j)$ matrix:

$$Acc = \sum_i^K F(y_i|y_i)P(y_i) \quad = \frac{TP+TN}{P+N} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right). \tag{3.6}$$

The arithmetic average of class accuracies $F(y_i|y_i)$ is called a *balanced accuracy* (*bAcc*) and it is a particularly useful evaluation measure for unbalanced datasets:

$$bAcc = \frac{1}{K}\sum_i^K F(y_i|y_i) \quad = \frac{1}{K}\sum_i^K TPR(i) \tag{3.7}$$

where K is the number of classes of objects.

Because the data used in our experiments are not balanced in classes (e.g. we have 36% vehicles and only 5% cyclists), as has been shown in the previous subsection, we chose to use in the performances calculation an arithmetic average of the class accuracies (i.e., a balanced accuracy *bAcc*), instead of a weighted one (*Acc*). Next, we present an example for the classification problem with 4 classes of objects, previously mentioned, in order to motivate the choice of *bAcc* instead of *Acc*. Let us suppose a certain classifier will provide the confusion matrix from table 3.1.

Table 3.1: An example of the confusion matrix obtained
for the classification problem with 4 classes of objects

	P	V	C	B
P	283	17	0	21
V	20	282	0	27
C	20	25	0	0
B	27	24	0	186

For this confusion matrix, the recall or TPR will be computed for each of the four classes as: $TPR = \frac{TP}{TP+FN}$. Therefore, for the class pedestrian we obtain $TPR(P) = 283/321 = 0.882$, for the class vehicle we have $TPR(V) = 282/329 = 0.857$, and so on. The value Acc will be computed after the formulae $Acc = \frac{TP+TN}{P+N}$ and it will be: $Acc = (283 + 282 + 0 + 186)/932$. In order to illustrate that it is a weighted average of class accuracies, it could be also written: $Acc = (\frac{283}{321} * 321 + \frac{282}{329} * 329 + \frac{0}{45} * 45 + \frac{186}{237} * 237)/932$. Finally, for Acc the obtained value will be: $Acc = 751/932 = 0.806$.

On the other side, if the *bAcc* will be computed after the formulae $bAcc = \frac{1}{K}\sum_i^K TPR(i)$, we will have: $bAcc = (0.882 + 0.857 + 0 + 0.785)/4 = 0.631$.

Because the class C (cyclist) has not been recognized at all, therefore there is a problem at this classification (so the classifier was not able to correctly learn the C class) we preferred *bAcc* instead of *Acc*. Unlike the value 0.806 obtained for *Acc*, the value for *bAcc* of 0.631 is showing that something has gone wrong at the classification. The best way to follow the results of a classification is to note the TPR or recall for each individual class of objects, but because we followed a single criteria for the classifier performance evaluation, we choose to employ an arithmetic average value rather than a weighted one for these individual class accuracies (or recall rates).

The confusion matrix $F(y_1, y_2)$ for the two-class problems can also be used to derive the weighted harmonic mean of recall and precision, which is called the F-measure (van Rijsbergen, 1979):

$$F_\alpha(X) \quad = \frac{1}{\frac{1}{\alpha+1}\left(\frac{\alpha}{recall} + \frac{1}{precision}\right)} \quad = \frac{(\alpha+1)recall \cdot precision}{recall + \alpha \cdot precision} \tag{3.8}$$

where $\alpha = 1$ in the case of F1-measure or simply F-measure. It is also called F-score and it can be interpreted as a weighted average of the precision and recall. An F-score reaches its best value at 1 and worst score at 0. The F-measure can be used as a single measure of performance of the test and as in the case of class accuracies, we prefer the arithmetic average F-measure of individual classes instead of the weighted one.

3.2.4   Features extraction

In the frame of our application, the image visual content represents the only available source of information. To describe the content of an image, usually some numerical measures with different ways of representing the information could be used. Next, the discussion is centred on the representation of images using their numerical signature, which means the representation via some extracted features (also called attributes). First, we present the attributes used to represent images in a digital or numeric format. These signatures could be then modified in order to reduce the size of the image space representation by a features selection procedure. If this step of selecting the most relevant features from all the features representing an image is performed, the classifier evaluation step could be less time consuming. Numerical attributes generally describe the colorimetric and/or the geometric properties of the images or of some regions of images. The choice of these attributes influences the classification results and the obstacles recognition process. Transforming the visual information (which humans observe easily in images) in some numerical values, features or attributes of low level (primitives) is not an easy thing. For computing the features, we opted for the choice of different families, which then will be combined to ensure a wide representation of the image content. Considering the relatively large size of the resulting vector, we took into account the selection of the most relevant features by the features selection operation.

Generally, different types of attributes capture different information from the images (even attributes belonging to the same family). Considering several types of attributes, there is the advantage of storing their possible complementarities. Thus, by using many types of attributes, combined with algorithms of features space reduction, the pertinence of each attribute can be evaluated in relation with the classification operation on different modalities (or even in relation to each object class). Once the most relevant attributes were selected, the system will have to extract only those attributes from the images. There are no studies indicating that a particular type of attribute is good (i.e. it succeed in capturing the most relevant information) in any object recognition system. Also, there is no research to indicate types of feature families to be used on different modalities (i.e. in the visible and the infrared domains). In order to adapt to a new base of images using a variety of attributes, the system should re-evaluate the importance of the attributes to differentiate between the new classes of objects (belonging to different modalities). Generally, to represent the image content, some intuitive, generic and low level features are used, such as colour, texture and shape.

Colour is a commonly used feature in the objects recognition domain, especially of those from nature, due to the multitude of colours that can represent different objects and therefore can assist in the segmentation or classification process. In the context of our application, where IR images are represented by different gray levels, and visible images also suffer a reduction of information due to the existence of situations like fog, night (where many white-gray-black pixels are present, so no colour) we choose to represent images on a single channel (in gray levels). Thus, the absence of colour information on most of the images used in our application, led us to believe that this feature, the colour, is not adapted for capturing the images content in our case of road obstacles. In addition, our application is aimed for the outdoor scenes, which are much more complex than the indoor ones, and objects can have a multitude of colours, shapes; therefore, we think this type of colour feature will not help much in the recognition process. Besides, by removing the three colour channels and by considering just a single one (gray levels) the time consumed by the application will be reduced with about two thirds.

Shape attributes are very useful for representing objects when some *a priori* information is known about the shape of the object. For example, there are a multitude of applications that use shape features specific to the pedestrian class (it is known that a pedestrian should have a roughly circular area representing his head; also a pedestrian must fall into certain patterns concerning the ratio height/width). These applications, which are based on shape features, are very limited from the viewpoint of the type of objects to be classified. They will be able to perform only a binary classification of the type pedestrian/non-pedestrian. Since our application is aimed for the detection and classification of several classes of obstacles from the road and not to only one type, we did not choose to use this type of shape-specific features. Our proposed method by extracting some features that characterize objects is more robust than other methods proposed in the literature, for example those based on shape (symmetry, snakes, template matching) in which all shape of the object must be included in the BB in order to be recognized by the recognition module. By the fact that we extract some features from the BBs enclosing the objects, the contain of the BB is better preserved than using some other methods, for example as those which use explicit shape features.

Since features extraction is desired to be fast for real-time constraints, the performances of the entire system depend heavily on the chosen features. We choose to extract obstacle shape independent but fast to compute features, so we have concentrated on different texture-based features. We did not select symmetries or edges because they are slower and it might not work very well for obstacles with arbitrary poses. Therefore, we retained a number of 171 texture-based features for VIS and respectively IR images, denoted in the following as VIS_{171} and IR_{171} feature vectors.

In the context of our application for road obstacle recognition by the fusion of information provided by VIS and IR images, for the image representation we choose the visual attributes described below.

- Width and height of the initial BB enclosing the object were chosen to be part of the FV because some of the applied transformations deform the image by a resize operation. Therefore, in order to preserve the initial size of the BB, we retain width and height (denoted in the following w and h). At a first sight, one might say we are cheating by considering these features, because if they would be extracted only from objects like pedestrians or vehicles, they would present a great power of discrimination. It is obvious that a vehicle will have a height approximately equal to the width, or even lower, while for a pedestrian these characteristics would be exactly the opposite. However, considering that in the image-database we used there are also cyclists and backgrounds, and objects could be occluded (so not the entire shape of the object will be comprised in the BB), or grouped (so there will be more objects belonging to the same class in a single BB), it can be said that unfortunately these **2 features** will loose from their discrimination power.

- The mean, median, mode, variance, standard deviation, skewness and kurtosis are the **7 statistic features** (denoted *stat* in our experiments) we have used. If X denote a random variable representing the gray levels of an image, the first-order histogram $P(X)$ is defined as the number of pixels with gray level X divided by the total number of pixels from the image. Let N be the number of possible gray levels. The mean value or the expected value of X is defined as $stat_1 = E(X) = \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ where $E(X) = \mu$ if X is normally distributed with parameters μ (the mean) and σ (the standard deviation). The median ($stat_2$) is the gray level from the image at which half of the rest of the gray values are below and half are above. The mode is the third $stat_3$ feature and it represents the most likely gray level from the image. The variance ($stat_4$) is a measure of the histogram width, that is, a measure of how much the gray levels differ from the mean. It is also interpreted as the second order moment on the image gray levels, and it is defined as: $stat_4 = Var(X) = E((X-\mu)^2) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$, where the standard deviation is defined as the square root of the variance: $\sigma(X) = \sqrt{Var(X)}$. The standard deviation of the zero mean (also called the L^2 norm) is the $stat_5$ feature: $stat_5 = L^2 norm(X) = \sqrt{\sum_{i=1}^{N}(x_i^2)}$. The skewness or the third order moment, which is $stat_6$ is a statistical measure of the degree of histogram asymmetry around the mean, or the degree of deviation from symmetry about the mean. Generally, it is computed with the formulae: $stat_6 = \frac{1}{N}\sum_{i=1}^{N}(\frac{x_i - \bar{x}}{\sigma})^3$. The fourth-order moment denotes the kurtosis of an image and it is the last statistical feature we have used. It is a measure of the degree of the histogram sharpness, i.e. a measure of the flatness or peakedness of a curve. It is computed by the relation: $stat_7 = \frac{1}{N}\sum_{i=1}^{N}(\frac{x_i - \bar{x}}{\sigma})^4$.

- From the Haar wavelet we obtained a number of **64 features** ($haar_1, ..., haar_{64}$) from each modality, VIS and IR. In order to obtain these features, we used two-dimensional wavelets and corresponding scaling functions obtained from one-dimensional wavelets by tensorial product. The Discrete Wavelet Transform (DWT) of a signal is calculated by passing it through a series of filters (high and low pass filters) and then downsampled. At each level, the signal is decomposed into low and high frequencies, and this decomposition halves the resolution since only half the number of samples are retained to characterize the entire signal (Mallat, 1998). The DWT leads to a decomposition of approximation coefficients at level $j+1$ in four components: the approximation components and the detail components in three orientations (horizontal, vertical, and diagonal). Due to successive downsampling by 2, the signal length must be a power of 2, or a multiple of a power of 2, and the length of the signal determines the maximum levels in which the signal can be decomposed. The algorithm retains the even indexed columns respectively rows. For the extraction of wavelet coefficients from the VIS and IR images, we used in our experiments the Daubechies family of wavelets with the order one, which is the same as Haar wavelet. In order to apply the wavelet decomposition, a resize operation was need. Because the objects' images have very small size (the size of the whole scene was 320×240 pixels, therefore the objects within these images were much lower, especially in VIS, as it can be seen from the examples given in figure 3.1), we choose to resize the VIS BB images at 16×16 pixels and the IR ones at 32×32 pixels. Thus, the wavelet decomposition was chosen to be performed at level one (for VIS BB images) or level two (for IR BB images), and finally we obtained a number of 8×8 wavelet coefficients for both types of images. Studies have been performed in (Apatean *et al.*, 2008d) regarding the wavelet family and the level of decomposition for the application of the wavelet transform. If the objects' image resolution were better, a higher level at which to apply the wavelet decomposition could be used, and thus to obtain 16×16 or 32×32 coefficients or even more.

- Next, besides features like Haar wavelet (64 coefficients), the Gabor (**32 coefficients**) wavelet (denoted $gbr_1, ..., gbr_{32}$) have been also considered, because both types of wavelets offer complementary information about the pattern to be classified and have proved good performance in other systems (Sun *et al.*, 2006a). Let $g(x, y)$ be the mother Gabor wavelet; then a dictionary of filters obtained by the dilation and rotation of this mother function results: $g_{mn} = a^{-m}G(x', y')$,

where $G(x',y')$ is the Fourier transform of the function $g(x,y)$ and the factor a^{-m} assures that the energy is not depending on m. In this relation, $a > 1$, $m,n \in N$, $m \in \{0,...,K-1\}$, $n \in \{0,...,S-1\}$, $x' = a^{-m}(x \cdot cos\theta + y \cdot sin\theta)$, $y' = a^{-m}(-x \cdot sin\theta + y \cdot cos\theta)$, $\theta = n\pi/K$, m, n are the orientation and the filter scales, and K, S are the total number of orientation and the total number of scales. In our experiments, we used four orientations $K = 4$, and four scales $S = 4$, obtaining thus a decomposition in 16 levels. The mean μ_{mn} and the standard deviation σ_{mn} of the magnitude of the Gabor transform coefficients were used, resulting in a number of $4 \times 4 \times 2 = 32$ features. These were arranged in the following manner in the feature vector: $[\mu_{00},\sigma_{00},\mu_{01},\sigma_{01},...,\mu_{33},\sigma_{33}]$. The method is described in (Manjunath & Ma, 1996) and later applied in (Florea, 2007).

- The Discrete Cosine Transform (DCT) tends to concentrate information, being intensively used for image compresion applications. The two-dimensional DCT is a separable linear transform and it is defined as: $B_{p,q} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{m,n} cos\frac{x(2m+1)p}{2M} cos\frac{x(2n+1)q}{2N}$, with $0 \le p \le M-1$, $0 \le q \le N-1$, $\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0, \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{cases}$, $\alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0, \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{cases}$, where A is the input image and B is the output image and M and respective N are the row and respectively the column size of A. If the DCT is applied to real data, the result will be also real. The first nine DCT coefficients are suggested to be used as texture features in (Ng *et al.*, 1992), (Ngo, 1998), (Ngo *et al.*, 2001), but it is also proposed to ignore the base component. Therefore, as in (Florea, 2007) we obtained a number of **8 *dct* features**.

- For our grayscale images, the co-occurrence matrix characterizes the texture of the image and the generated coefficients are often called Haralick features, after the author of (Haralick *et al.*, 1973). They are in number of 7, but in (Cocquerez & S.Philipp-Foliguet, 1995) only four of them are proposed to be used: the homogeneity, entropy, contrast and correlation, because they seemed to be the most important ones. The Gray Level Co-occurrence Matrix (GLCM) is used to explore the spatial structure of the texture and it captures the probability that some pixels appear in pairs with the same level of gray but with different orientations. Therefore, we concentrated on the 4 features proposed in (Cocquerez & S.Philipp-Foliguet, 1995) and we performed the computation in 4 different directions: $0°$, $45°$, $90°$ and $135°$ as it is proposed in (Florea, 2007). In this manner, we obtained a number of **16 features**, denoted $cooc_1,...,cooc_{16}$.

- The Run Length Encoding (RLE) method works by reducing the physical size of a repeating string of characters, i.e. sequences in which the same data value occurs in many consecutive data elements are stored as a single data value and counted. Galloway proposed the use of a run length matrix for the extraction of texture features in (Galloway, 1975). For a given image, the proposed method defines a run-length matrix as number of runs (i.e. the number of pixel segments having the same intensity) starting from each location of the original image in a predefined direction. The direction (in our case $0°$ and $90°$) and the number of gray levels (8 in our case) have to be mentioned because the value contained in the matrix's (l,n) square is equal to the number of segments of length l and gray level n . This implies that the matrix's number of columns is dynamic, being determined by the length of the longest segment. Short run emphasis, long run emphasis, gray-level distribution, run-length distribution and run percentage are the five features proposed by Galloway. Two supplementary measures (low gray-level run emphasis and high gray-level run emphasis) proposed in (Chu & Greenleaf, 1990), have been also considered. Thus, a set of 7 *rle* features obtained in one direction have been chosen. We performed the computation at $0°$, $90°$ and we obtained a number of **14 features**, denoted $rle_1,...,rle_{14}$.

- Some signal processing techniques are based on texture filtering and analyze the frequency contents in the spatial domain. Laws have suggested a set of 5 convolution masks for feature extraction based on texture in (Laws, 1980). From these 5 masks, a set of 25 two-dimensional masks have been further obtained and based on these 2D masks, 14 features called *laws* in our processing are reached. These 14 features are reported to the elements from the first

diagonal, in the following manner: the first 10 features are normalized with the first element from the diagonal, and the rest of 4 features are normalized with the remaining 4 (from the set of 5) diagonal elements. To these 14 features, the mean and the standard deviation have been applied as it is suggested in (Pratt, 2001) and (Florea, 2007), resulting thus a number of **28** *laws* **features**.

Therefore, a number of 171 features have been extracted from each modality VIS and IR. These coefficients, grouped in families of features are summarized in table 3.2. Due to the fact that the information is extracted individually from the VIS and IR images, the provided FVs are called monomodals, therefore they are extracted from a single modality, VIS or IR. These FVs can be seen as input vectors to two different and independent systems: one specialized on the VIS image and the other specialized on the IR image. In the following, the importance of these coefficients but also the individual performance of each family of features will be evaluated.

To maximize the performance of individual descriptors, new vectors have been formed as combinations of feature families. Thus, we have combined the texture descriptors in a single FV of texture (abbreviated "Text"). It summarizes all texture feature-families (*haar*, *dct*, *cooc*, *gbr*, *rle*, *laws*) and includes 162 characteristics for the VIS and IR monomodal systems case. Adding the 7 statistical moments, a new vector called (*StatText*) is obtained. If in addition, we add the 2 geometrical features *geom*, the maximum size vectors (denoted "*AllFeatures*") of 171 features will be obtained.

Table 3.2: Feature vectors (FVs) for monomodal systems

		Features		No. of att	Vector structure
		Geometric features	$geom$	**2**	$[w, h]$
		Visible features (v)			
		Statistical moments	$stat_{VIS}$	**7**	$[stat_{v1} \quad \ldots \quad stat_{v7}]$
		Haar wavelet	$haar_{VIS}$	**64**	$[haar_{v1} \quad \ldots \quad haar_{v64}]$
		Gabor wavelet	gbr_{VIS}	**32**	$[gbr_{v1} \quad \ldots \quad gbr_{v32}]$
		DCT coefficients	dct_{VIS}	**8**	$[dct_{v1} \quad \ldots \quad dct_{v8}]$
FEATURE FAMILIES	Monomodal FVs	Coocurence matrix	$cooc_{VIS}$	**16**	$[cooc_{v1} \quad \ldots \quad cooc_{v16}]$
		Run Length Encoding	rle_{VIS}	**14**	$[rle_{v1} \quad \ldots \quad rle_{v14}]$
		Laws features	$laws_{VIS}$	**28**	$[laws_{v1} \quad \ldots \quad laws_{v28}]$
		Infrared features (i)			
		Statistical moments	$stat_{IR}$	**7**	$[stat_{i1} \quad \ldots \quad stat_{i7}]$
		Haar wavelet	$haar_{IR}$	**64**	$[haar_{i1} \quad \ldots \quad haar_{i64}]$
		Gabor wavelet	gbr_{IR}	**32**	$[gbr_{i1} \quad \ldots \quad gbr_{i32}]$
		DCT coefficients	dct_{IR}	**8**	$[dct_{i1} \quad \ldots \quad dct_{i8}]$
		Coocurence matrix	$cooc_{IR}$	**16**	$[cooc_{i1} \quad \ldots \quad cooc_{i16}]$
		Run Length Encoding	rle_{IR}	**14**	$[rle_{i1} \quad \ldots \quad rle_{i14}]$
		Laws features	$laws_{IR}$	**28**	$[laws_{i1} \quad \ldots \quad laws_{i28}]$
		Texture from VIS	$Text_{VIS}$	**162**	$[haar_v, dct_v, cooc_v, gbr_v, rle_v, laws_v]$
		Texture from IR	$Text_{IR}$	**162**	$[haar_i, dct_i, cooc_i, gbr_i, rle_i, laws_i]$
COMBINATION OF FEATURES	Monomodal FVs	Statistical and Texture from VIS	$StatText_{VIS}$	**169**	$[stat_v, haar_v, dct_v, cooc_v, gbr_v, rle_v, laws_v]$
		Statistical and Texture from IR	$StatText_{IR}$	**169**	$[stat_i, haar_i, dct_i, cooc_i, gbr_i, rle_i, laws_i]$
		Geometrical and	$geom$	2	$[w, h,$
		Statistical and Texture from VIS	$StatText_{VIS}$	169	$stat_v, haar_v, dct_v, cooc_v, gbr_v, rle_v, laws_v]$
		All features from VIS	$AllFeatures_{VIS}$	**171**	$[w, h, stat_v, txt_v]$
		Geometrical and	$geom$	2	$[w, h,$
		Statistical and Texture from IR	$StatText_{IR}$	169	$stat_i, haar_i, dct_i, cooc_i, gbr_i, rle_i, laws_i]$
		All features from IR	$AllFeatures_{IR}$	**171**	$[w, h, stat_i, txt_i]$

3.2.5 Features evaluation

As we saw in the previous section, different families of features were considered to be extracted from the VIS and IR images. Different algorithms of features extraction provide different characteristics (as we already mentioned, grouped in feature-families) which can be combined in different FVs, representing the inputs into the classifier. The accuracy of the classifier depends on how well these features succeed in representing the information and it is not necessary proportional with their number (or FV dimension). Is it possible that the same features extraction algorithm applied on the VIS and on the IR domains to deliver distant results, i.e. to exist some features better suited for the VIS domain and others better suited for the IR domain. Also, their combination can bring in some improvements from the viewpoint of the recognition performance, depending on how complementary they are when representing the information. There are features extraction algorithms consuming less time than others at the extraction of these features from images. There are also families of features that can be separable (when calculating the coefficients of a family, they can be calculated individually, and do not need to be calculated all if we do not need all of them) and this will influence the extraction time of those coefficients.

To assess the performance representation of the numerical attributes, in this section we present a first experiment in which we tested, using a simple classifier kNN the representation ability of the visual content of each family of attributes. The kNN algorithm (Cover & Hart, 1967) is a method for classifying objects based on closest training examples in the feature space. The kNN classifier is a type of instance-based learning where the function is only approximated locally and it is amongst the simplest of all machine learning algorithms. The value k is a positive integer, typically small ($k = 1$ or $k = 3$). The training phase of the algorithm consists only of storing the FVs and class labels of the training samples. In the classification stage, the test sample (whose class is not known) is represented as a vector in the feature space. Distances from the new vector to all stored vectors are computed and k closest samples are selected. The test object is classified by a majority vote of its neighbours, with the object being assigned to the class most common amongst its k nearest neighbours. [3] In our processing, the case $k = 1$ was chosen, where the object is simply assigned to the class of its nearest neighbour.

In this stage of features evaluation, we opted for the use of a kNN classifier instead of the powerful SVM because the kNN is much simple. It does not need a model-selection stage, as the SVM does, because it is not having multiple parameters which have to be optimized before the usage. Still, because SVM is more parametrizable and therefore better adapted to any classification problem, it is expected that the recognition rates to be higher by the use of the SVM. In the frame of our final system we employed an SVM classifier (where a model validation stage will be performed to find the proper hyper-parameters to be used for a specific FV), but in this section, just for the sake of simplicity, we prefer the kNN. Our purpose in this section is not to optimize the classifier on each family or combination of feature-families, but to evaluate their importance.

Few questions are foreseen here:

1. Are several features better adapted for VIS and other better adapted for IR ? Or, if a family is behaving well on VIS, it will be also good on IR?

2. The number of features of one family influences the classification rate? A family with many features will provide a greater recognition rate compared to another family having less features?

3. Are the chosen features pertinent for the learning process? Or they will suffer of overfitting (will provide good results on the training set, but they would not predict very well the test data)?

---

[3] The kNN method used in our experiments is the one implemented in Weka, and we call it from MATLAB; the Euclidean distance was selected in the frame of the IBk algorithm and the number of neighbours was $k = 1$.

In order to answer these questions, we used all the vectors comprising families or combination of feature-families from the table 3.2 and we performed 2 experiments for the classification problems with 4 and 8 classes of objects previously mentioned in subsection 3.2.2. In a first experiment we considered only the training dataset (932 objects), where by the 10 folds cross-validation (denoted 10f-CV) procedure the results from tables 3.3 and 3.4 have been obtained.

In the second experiment, we concentrated on the results obtained when the system is learnt with the data from the training dataset and is tested with the data from the test set (232 objects). The classification results for the second experiment were denoted LT and they are summarized in figures 3.8(a) and 3.8(b) and figures 3.9(a) and 3.9(b). Because the final FV is chosen on the training database, the most important accuracies are the ones obtained in the 10f-CV case, but in order to illustrate that the chosen features are also pertinent for tests, we have retained also the accuracies for the LT case. The values obtained in these two experiments and illustrated in the tables and figures previously mentioned are the recognition rates of 4 classes and respective 8 classes of objects by the bAcc (balanced accuracy) criteria.

Table 3.3: Performance representation of monomodal FVs obtained using 10f-CV on the training set for the classification problem with 4 classes of objects

Input vector		Accuracy using 10f-CV		Inputs arranged by decreasing value of the bAcc for VIS		
Attributes	Number of attributes	for the 1-NN classifier		Input vector	Accuracy for the 1-NN classifier	
name	VIS, IR	VIS	IR		VIS	IR
geom	2	47.5	47.5	haar	77.0	79.6
stat	7	59.0	66.0	gbr	72.6	81.6
haar	64	77.0	79.6	laws	67.4	69.8
gbr	32	72.6	81.6	dct	65.7	75.0
dct	8	65.7	75.0	stat	59.0	66.0
cooc	16	54.2	66.1	cooc	54.2	66.1
rle	14	43.0	55.0	geom	47.5	47.5
laws	28	67.4	69.8	rle	43.0	55.0
Text	162	83.8	87.1			
StatText	169	83.5	87.9			
AllFeatures	171	83.7	88.0			

Table 3.4: Performance representation of monomodal FVs obtained using 10f-CV on the training set for the classification problem with 8 classes of objects

Input vector		Accuracy using 10f-CV		Inputs arranged by decreasing value of the bAcc for VIS		
Attributes	Number of attributes	for the 1-NN classifier		Input vector	Accuracy for the 1-NN classifier	
name	VIS, IR	VIS	IR		VIS	IR
geom	2	39.5	39.5	haar	66.2	70.7
stat	7	40.9	44.5	gbr	61.9	72.2
haar	64	66.2	70.7	dct	52.8	59.7
gbr	32	61.9	72.2	laws	51.2	47.6
dct	8	52.8	59.7	cooc	43.2	54.9
cooc	16	43.2	54.9	stat	40.9	44.5
rle	14	29.2	34.8	geom	39.5	39.5
laws	28	51.2	47.6	rle	29.2	34.8
Text	162	73.8	78.5			
StatText	169	73.3	79.4			
AllFeatures	171	73.9	79.3			

From the viewpoint of the accuracies obtained for the two problems of classification (with 4 and respectively 8 classes of objects) one can notice from all the three sets of representations [4] that the

---

[4]the best visualization could be performed on the figures with bars, 3.8(a) and 3.8(b)

accuracies obtained for the classification with 4 classes of objects is with approximately 10% higher than those obtained for the classification problem with 8 classes of objects. Therefore, the higher the number of classes from the classification, the lower the recognition rates (supposing the same dimension of the database from the validation and test stages). In our case, these low accuracies obtained for the 8-class classification problem are due to the reduced number of instances per each object class.

It can be easily seen from tables 3.3 and 3.4, but also from the figures previously mentioned, that from VIS and IR vector types, the accuracies for the VIS FVs are lower than those of the IR FVs. It is possible that this happens because the VIS image resolution is lower than its IR counterpart. For all the 10 vectors (with the exception of the *geom* vector which is the same vector for the two situations VIS and IR), the accuracies obtained with the IR FVs has overperformed the accuracies obtained with the VIS FVs.

1. Are several features better adapted for VIS and other better adapted for IR? Or, if a family is behaving well on VIS, it will be also good on IR?

   The interpretation of the tables 3.3 and 3.4 must be realized in two steps, because they are structured in two parts:

   - the first part (or the first half) of the tables, which contains the first 4 columns and where the FVs are arranged in the same order as in table 3.2, and

   - the second half of the tables, which is containing the last three columns; its content is nothing more than the first part of the table, but arranged in a decreasing order of the bAcc obtained on the VIS domain. We performed this arrangement of the vector families in order to verify if the order of the families' importance from the VIS and IR domains is the same. Therefore, we intend to verify if the families' behaviour from the VIS is maintained also on the IR domain; in this manner, we try to answer the first question formulated earlier.

   For the classification problem with 4 classes of objects, one can notice from table 3.3 that the order (in the sense of a decreasing bAcc) from the VIS domain is as follows: *haar*, *gbr*, *laws*, *dct*, *stat*, *cooc*, *geom*, *rle*, while the order from the IR domain (the last column from the table followed also by decreasing values for the accuracies) is *gbr*, *haar*, *dct*, *laws*, *cooc*, *stat*, *rle*, *geom*; therefore, comparing the two lists, we can conclude all the families from the VIS have been reversed in pairs of two to obtain the family order from the IR list.

   Unlike this inversion scheme, for the classification problem with 8 classes of objects (table 3.4), only the first families (*haar* and *gbr*) and (*laws* and *cooc*) have changed their order, all the rest of the families remaining in the same order as on the VIS, which is: *haar*, *gbr*, *dct*, *laws*, *cooc*, *stat*, *geom*, *rle*.

   The importance of the features from the VIS is not exactly the same as with the importance of the features from the IR domain, but there are features better than others. The obtained results indicate that a finer selection process has to be performed, at the features level not at their families. Therefore, in Chapter 4 we return to this idea.

   In the first parts of the two tables 3.3 and 3.4 it can be seen that for both classification problems (with 4 classes and 8 classes of objects), the higher accuracies on the 10f-CV procedure are obtained for the families combinations comprising most of the features. Thus, the best accuracies are obtained for all texture features comprised in a vector, therefore the *Text* vector, or combined with the *stat* features by the vector *StatText*, or all features grouped in a single

FV called *AllFeatures*. It can be seen that the difference between the performances obtained with these three FVs is very small. It was expected that the higher value for the accuracy to be obtained for the vector comprising all the features (*AllFeatures*) in all the four cases (VIS and IR for the problem with 4 classes and VIS and IR for the problem with 8 classes), but as it can be noticed, this is not always true: in the case of the classification problem with 4 classes of objects, the maximum accuracy is reached for the vector $Text$ in the VIS case and for the classification problem with 8 classes, the maximum is obtained for the vector $StatText$ in the IR case. In order to not remove an entire family of features from the very beginning, we choose to perform the features selection operation on the vector comprising all the features. This is done in order to consider their complementarities (we will detail this idea in the next chapter dedicated to features selection mechanism).

From the results obtained in the 10f-CV process, as can be seen in tables 3.3 and 3.4, the families order in terms of decreasing accuracies will be: for the problem with 4 classes, on the VIS: *haar*, *gbr*, *laws*, *dct*, *stat*, *cooc*, *geom*, *rle*, and on the IR we have: *gbr*, *haar*, *dct*, *laws*, *cooc*, *stat*, *rle*, *geom* ; for the classification problem with 8 classes of objects, on the VIS we have: *haar*, *gbr*, *dct*, *laws*, *cooc*, *stat*, *geom*, *rle*, and on the IR we have: *gbr*, *haar*, *dct*, *cooc*, *laws*, *stat*, *geom*, *rle*.

For the values obtained in the LT procedure (they were not given explicitly, as were those from the 10F-CV in tables 3.3 and 3.4) [5], the following order will be obtained: for the classification problem with 4 classes of objects, on the VIS we will have *gbr*, *haar*, *laws*, *dct*, *stat*, *cooc*, *geom*, *rle*, while on the IR *gbr*, *haar*, *dct*, *laws*, *cooc*, *stat*, *rle*, *geom*, and for the classification problem with 8 classes of objects, for the VIS: *haar*, *gbr*, *laws*, *dct*, *geom*, *cooc*, *stat*, *rle*, and for the IR: *haar*, *gbr*, *dct*, *cooc*, *laws*, *stat*, *rle*, *geom*.

From the analysis of the two cases (the classification problem with 4 classes and the classification problem with 8 classes of objects) we can not conclude that there is a certain order of the feature families importance, but we could perform a grouping to obtain pairs of families, and then we could say that on both 10f-CV and LT the following order is maintained: the wavelet feature pair (*haar* and *gbr*) is the first one, followed by the pair *laws-dct* and then *cooc-stat* and finally the *geom-rle* characteristics pair. There are two exceptions, on the LT for the problem of classification with 8 classes: on IR the *laws* and the *cooc* features have interchanged, and on VIS the *geom* and the *stat* have also changed their position between them.

Also, for the obtained results we could generalize that the families *haar*, *gbr*, *laws* and *dct* are better than *stat*, *cooc*, *geom* and *rle*. However, the first group of families (with the exception of *dct*) has the largest number of features, therefore the increased accuracies could be of that reason. In the following, we want to verify this affirmation but also to answer the question number 2.

2. The number of features of one family influences the classification rate? A family with many features will provide a greater recognition rate compared to another family having less features?

To better compare the results for 10f-CV and LT for all feature vectors (comprising individual families and/or combinations of families) from the table 3.2, we plotted the accuracy values in another form (figures 3.8(a) and 3.8(b)), where on the *x*-axis it is specified the name of different family/family combinations together with the proportion they represent within the maximum size vector, which is *"AllFeatures"* from the table 3.2).

---

[5]the best visualization could be performed on the figures 3.9(a) and 3.9(b)
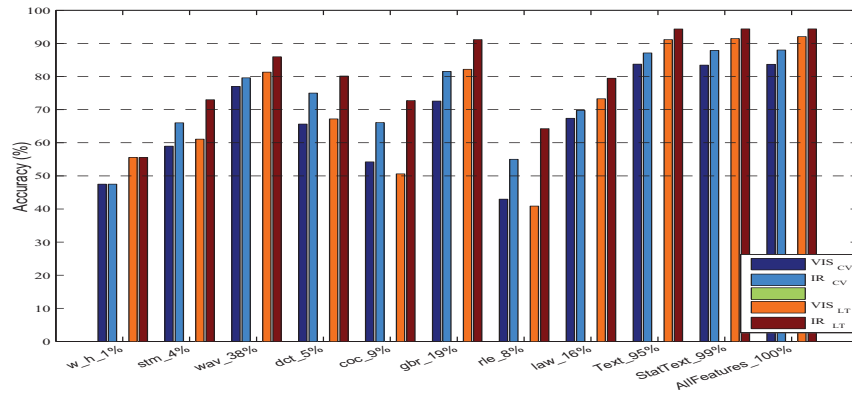
(a) Accuracy obtained for different FVs (comprising families and combinations of families of features) with 10f-CV (first 2 of each group) and LT (last 2 of each group) using 1-NN for the classification problem with 4 classes of objects
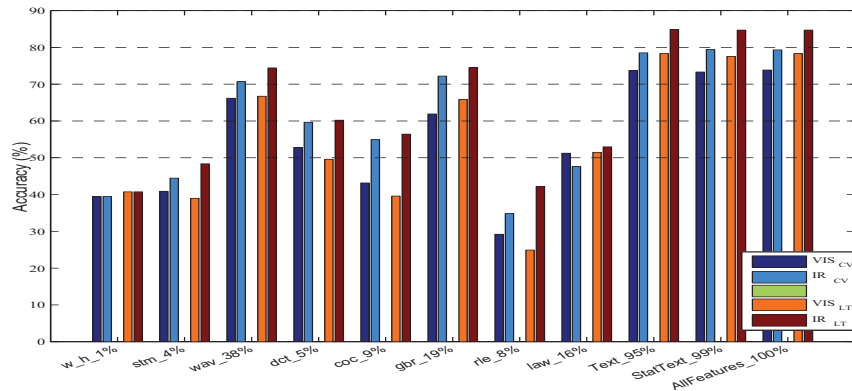
(b) Accuracy obtained for different FVs (comprising families and combinations of families of features) with 10f-CV (first 2 of each group) and LT (last 2 of each group) using 1-NN for the classification problem with 8 classes of objects

Figure 3.8: Accuracy obtained for different FVs (comprising families and combinations of families of features) using 1-NN for the classification problem with a) 4 classes and b) 8 classes of objects

Thus, each group represents a family or a family combination, and within each group the first two values are the accuracies (for VIS and IR) obtained in the cross-validation (CV) stage, and the last two are obtained for the LT stage. On the *x*-axis we have chosen to represent each feature vector from the tables 3.3 and 3.4 together with the percentage they represent in the frame of the maximum-size vector with 171 features. The results obtained at 10f-CV are the first two, represented in blue in figures 3.9(a) and 3.9(b), and the results from LT are the next two, represented in red in each group.

For example, the Haar wavelets family comprise a number of 64 coefficients (features) from a total of 171 coefficients. The percent obtained in this case is approximately 38%. Similarly, we obtain the size percentage for all families represented in this figure. We choose to represent in the results this manner in order to better show the accuracy provided by a certain family reported to the number of coefficients used to obtain that accuracy rate.
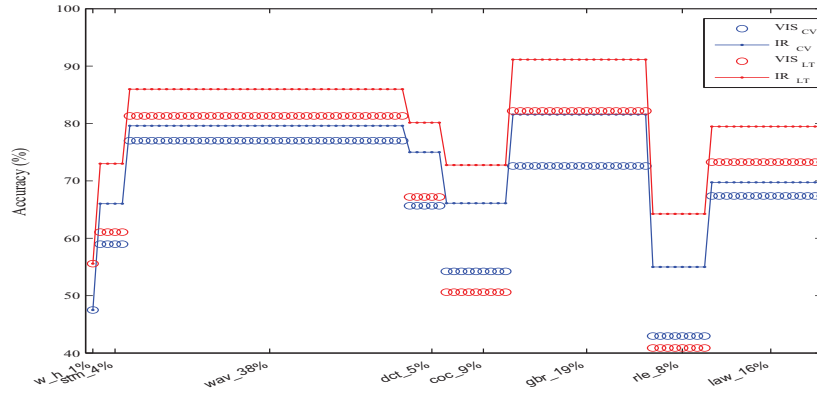
(a) Accuracy obtained for different FVs (comprising only families of features) from visible (circle) and infrared (line) domains, evaluated by 1-NN using 10f-CV (blue) and LT (red) for the classification problem with 4 classes of objects
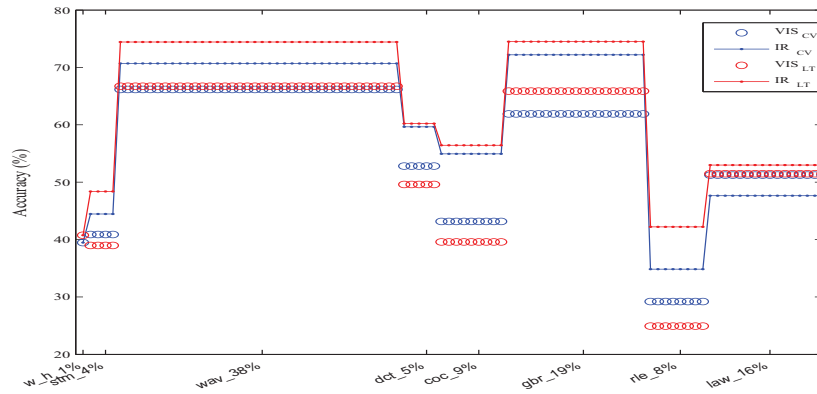
(b) Accuracy obtained for different FVs (comprising only families of features) from visible (circle) and infrared (line) domains, evaluated by 1-NN using 10f-CV (blue) and LT (red) for the classification problem with 8 classes of objects

Figure 3.9: Accuracy obtained for different FVs (comprising only families of features) using 1-NN for the classification problem with a) 4 classes and b) 8 classes of objects

From the figures 3.8(a) and 3.8(b), one can notice that although the Haar wavelet coefficients are most numerous, they are exceeded in their performance by Gabor features which are only half as concerning their number. From the viewpoint of the performance, after *gbr* and *haar* features, on the 3rd and 4th positions are *laws* and *dct* features, accounting 16%, respectively 5% of the total vector, followed by *cooc* and *stat*, and finally *rle* and *geom*. The *geom* features do not have to be ignored, because with only 1% of the features, they succeed to obtain an accuracy of about 50%-60%.

Therefore, the conclusion is that the number of features of one family does not necessary influences the classification rate, because there are families with fewer features providing a greater recognition rate than another family having more features (e.g. *gbr* and *haar*).

3. Are the chosen features pertinent for the learning process? Or they will suffer of overfitting (will provide good results on the training set, but they would not predict very well the test data)?

   The danger of "overfitting" is to find features that "explain well" the training data, but have no real relevance or no predictive power (for the test set). Generally, one can notice that the accuracies obtained in the LT stage overperformed (or are very closed to) the values obtained using the 10f-CV procedure, so our data is not presenting overfitting. Therefore, we can say we have chosen some general features, which are capable to retain the pertinent information from both VIS and IR individual domains.

By now, we have discussed methods of features extraction from the point of view of the accuracy and we saw which are the performances given by different feature-families or combinations of feature-families. Next, we want to see these issues also in terms of their extraction time. In table 3.5 we plotted the average extraction time of the FVs from the table 3.2 for one object (obtained through mediation on the training set). This extraction time, together with the time required to test/classify the vector from the test set within the system are the ones that most affect the performances of a real time system from the recognition point of view.

From table 3.5 it can be seen that generally the time for the extraction of various vectors including families or combinations of families is smaller on VIS than on the IR domain (the only exception is on *rle*). The objects' image on the IR domain (the BBs) were greater in size than those from the VIS, so in the pre-processing step, at the resize operation, the ones from the VIS have been resized to 16×16 pixels, and the ones from the IR to 32×32 pixels. Therefore, when double the size on x and y axis has turned into a total of four times number of pixels; for this reason, the time necessary to extract the features is higher in the IR case. It can also be noticed that the time required for the extraction of *geom* coefficients is zero, so if these coefficients would help from the viewpoint of the accuracy, the time would not be a criteria for their rejection.

Table 3.5: Mean extraction time for different FVs for one object

Attributes name		Number of attributes Monomodal systems VIS, IR	Mean Extraction time [msec] for 1 object	
			VIS	IR
geom		2	0.0	0.0
stat		7	1.2	2.5
Texture	haar	64	2.2	4.0
	gbr	32	9.2	17.2
	dct	8	1.9	2.1
	cooc	16	0.9	1.2
	rle	14	7.3	6.4
	laws	28	2.1	2.5
Text		162	23.6	33.5
StatText		169	24.8	36.1
AllFeatures		171	24.8	36.1

If we assume that the proper vector used to characterize the data from the image database is the one comprising all the 171 features, i.e. *"AllFeatures"* and no parallel processing is implied, then the average time required for the extraction of the features corresponding to a single test object (i.e. one BB) will be:
   - approximately 25 milliseconds if we have a monomodal VIS system,
   - about 36 milliseconds if the system is a monomodal IR one or if the system is a bimodal VIS-IR one capable to process the VIS and IR information simultaneously,

- nearly 61 milliseconds in the case we have a bimodal VIS-IR system which cannot process the VIS and IR information in a parallel way [6].

In addition to this features extraction time, there is the time required by the classifier to provide the estimated class for the respective test object. This classification time depends to a large extent by the parameters of the classifier (i.e. the SVM) chosen in the training stage of the system, but generally it cannot exceed a few tens of milliseconds for a single test object. If we consider a very pessimistic scenario, we could say that the time needed for the OR module to recognize the type of a single test object is maximum 100 milliseconds. This value is not a favourable one, considering that the proposed detection system is a stereo one (therefore a time consuming one). As it is specified in (Toulminet *et al.*, 2006) the detection time for one object is also approximately 100 milliseconds. Therefore, the detection and classification processes (which can not be paralelized because they are dependent) require for a single BB a time of almost 200 milliseconds in the worst case. At a first sight it does not seem to be much, but when considering that in a scene-image almost never appears a single BB detected, then it may become an unhappy scenario. Considering a number of 10 objects per scene (which is a quite frequently number in a cluttered environment) this amount of time become a significant one [7].

In order to decrease this large time, at least from the recognition point of view, we propose to encode the information with a feature-vector as smaller as possible. Decreasing the number of features which compose the FV, it is possible to decrease also the time in which these features are extracted from the image, if we consider that some feature-families are separable (i.e. we can compute only some features from that family, not all of them if we do not need all of them). As a consequence, the time required by the classification to provide the class for the test object will be also decreased. In the next chapter, we propose to decrease the dimension of the FV by performing the features selection operation. Thus, the features will be individually evaluated and those which will be assessed as being not-relevant, could be discarded from the final FV.

Next, when running the experiments with the SVM classifier, we have focused on the feature vector *AllFeatures*, incorporating all the 171 features, because only after the features selection process (detailed in the next chapter) we will drop some features if they will not help in the classification process, i.e. if they will be found as being not relevant.

3.2.6   Classification with SVM

Support vector machines (SVMs) are supervised kernel based learning methods, used for classification and regression. A specific characteristic of SVMs is that they map the input vectors to a higher dimensional space where a maximal separating hyper-plane is constructed.

Suppose the training data as a set of instance-label pairs (x_i, y_i), $i = 1, ..., m$ where $x_i \in R^n$ represents the input vector and $y_i \in \{-1, +1\}$ the output label associated to the corresponding item x_i. The parameter n represents the input vector dimension, where x_i corresponds to $(x_i^1, x_i^2, ..., x_i^n)$. These vectors will be mapped into a feature space using a kernel function K, which defines similarities between pairs of data, with $K(x, z) = \langle x, z \rangle$, $\forall x, z \in R^n$ (Boser *et al.*, 1992), (Vapnik, 1998), (Cristianini & Shawe-Taylor, 2000). In order to use this kernel function for an SVM classifier, one has to solve the following optimization problem:

$$\max_{a \in R^m} \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i,j=1}^{m} a_i a_j y_i y_j K(x_i, x_j) \tag{3.9}$$

---

[6] in today's systems, most processing implies pipeline to speed up the computation; therefore, this assumption in rarely met.
[7] the reported execution time was obtain on an Intel(R) Core(TM)2Duo CPU at 2.00GHz

under the constraints $\sum_{i=1}^{m} a_i y_i = 0$, $0 \le a_i \le C$, $\forall i \in 1, 2, ..., m$, where $K(x_i, x_j)$ represents the kernel function and $a \in R^m$ denotes the vector with components a_i, with a_i the Lagrange coefficients. The coefficient $C > 0$ is the penalty parameter that controls the trade off between maximizing the margin and classifying without errors. The optimal separating hyper-plane is used to classify the un-labelled input data x_k using the following decision function:

$$y_k = sign\left(\sum_{x_i \in S} a_i y_i K(x_i, x_k) + b\right) \tag{3.10}$$

where S is the set of support vector items x_i and the offset value b is calculated based on vector a and the training set.

The basic kernels used in the literature are the linear (LIN), the polynomial (POL) and the radial basis function RBF ones:

- LIN : $K_{Lin}(x_i, x_j) = x_i^T x_j$

- POL : $K_{Pol}(x_i, x_j) = (x_i^T x_j)^d, d > 0$

- RBF : $K_{RBF}(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2), \gamma > 0$

In our experiments the POL kernel has a degree $d \in \{1, 2, ..., 15\}$ (with $d = 1$ for the linear kernel) and the RBF kernel has the bandwidth γ, of the form $\gamma = q \cdot 10^t$ with $q \in \{1, 2, ..., 9\}$ and $t \in (-5, -1)$. In (Hsu *et al.*, n.d.) is suggested that the parameter C, which is the penalty parameter, to be chosen in the range $C \in \{2^{-5}, 2^{-3}, 2^{-1} ..., 2^{+15}\}$. Larger values of C might lead to linear functions with smaller margin, allowing to classify more examples correctly with strong confidence. The results found in literature indicate that these discrete spaces of parameters are the most suitable for an efficient classification. A proper choice of these parameters is crucial for SVM to achieve good classification performance. The values C and d or γ are tuneable hyper-parameters which need to be determined by the user. They are usually chosen by optimising a validation measure (such as the k-fold cross validation error) on a grid of values (e.g. uniform grid in the (C, d) or (C, γ) space).

The SVMs classification performances depend on the chosen kernel, the penalty parameter C and the parameters γ, d corresponding to that certain type of kernel. All these parameters are called hyper-parameters.

Due to the fact that we envisioned the use of a complex kernel, also called a multiple kernel in Chapter 5, in the following, when processing the information through the SVM by the use of classical kernels, we denoted those kernels as single kernels (SKs).

Because the kernel functions from the two modalities VIS and IR (SK_{VIS} and SK_{IR}) could be of different types, and could work with different hyper-parameters, for the kernel type we choose the LIN, POL and RBF cases, which have less than two parameters to optimize. Because the linear SVM is a particular case of the polynomial SVM, it can be concluded that the single kernel (SK) could be either a radial basis function or a polynomial one:

$$SK \in \{RBF, POL\}. \tag{3.11}$$

3.3 Classification Experiments and Results

Because there is not known beforehand which combination of the SVM hyper-parameters is the most appropriate for a certain classification problem, in almost all applications involving the use of an

SVM, an operation of searching for the proper combination is performed. This search is generally called parameter search or model-selection.

Generally, the kernel and hyper-parameters selection task is performed by training the classifier with different functions (acquired with different kernels and hyper-parameter values chosen from a discrete set which is fixed *a priori*) and choosing the one corresponding to the best performance measure. This choice is realised based on a procedure of k-folds cross-validation which implies an average of the results of multiple splitting.

Each data set corresponding to a multi-class classification problem (as we mentioned in section 3.2.2, we defined 2 classification problems: with 4 and respectively 8 classes of objects) was manually annotated, and further they were randomly divided into two sub-sets: training sub-set (80%) and testing sub-set (20%), but considering a ratio of almost 4 between the number of objects belonging to each class from the training and testing sets. Averaging the results of multiple splitting (or the "cross-validation" technique) performed on the training data is a commonly used technique to decrease the variance of the performance estimator. Therefore, to perform the 10 folds cross-validation (10f-CV) procedure, the learning sub-set was randomly partitioned into learning (9/10) and validation (1/10) parts. In figure 3.10 is depicted this partitioning of the dataset.
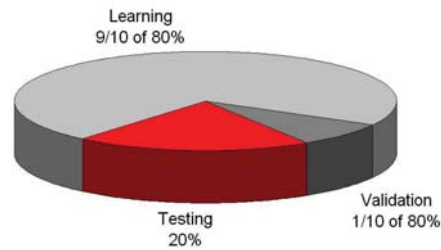
Figure 3.10: Partitioning of the dataset

Therefore, after obtaining the training part as 80% of the entire set of data, it was divided again in: a learning sub-set (used by the SVM algorithm in order to learn the model that performs the class separation) and a validation sub-set (used in order to optimise the values of the hyper parameters). The SVM model, which is learnt in this manner, classifies (or labels) the unseen examples from the test set, which is disjoint to the training one.

A new set of experiments were performed by using the SVM-based model. The C-SVM algorithm, provided by LIBSVM (Chang & Lin, 2001), with an RBF or POL kernel is actually used in this second experiment. The parameters of the SVM model (the penalty for miss-classification C and the kernel parameters) are optimized on the validation set. The cross-validation framework is utilised in order to avoid the over fitting problems. Thus, we automatically adapt the SVM classifier to the problem, actually the recognition of road obstacles.

Having in mind that it is not known beforehand which parameters for the SVM classical kernels (C and γ or d) gives the best solution for one problem, there must be done a model selection (parameter search) that could identify good C, γ or d values (for SKs). For our problem, the kernel functions SKs could be of different types (POL or RBF), and could work with different hyper-parameters. A grid search was performed for every type of kernel, with the kernel parameter and the penalty parameter C representing the values to optimize (different values among a discrete set were used). When the optimization process is ending, a winner kernel is chosen on each modality: SK^*_{VIS} and SK^*_{IR} (with the corresponding hyper-parameters ($SKtype,C,SKparameter$)) for the optimization of the monomodal systems.

In our experiments, the optimisation of the kernel type and hyper-parameters is performed by a parallel grid search method in the following ranges:

- the tuning coefficient $C \in \{2^{-5}, 2^{-3}, 2^{-1} \ldots, 2^{+15}\}$ range;
- the bandwidth of the RBF kernel $\gamma \in \{2^{-25}, 2^{-23}, \ldots, 2^{-11}\}$;
- the degree of the POL kernel is chosen among $d \in \{1, 2, ..., 7\}$.

For each combination of these two parameters, a 10-fold cross-validation is performed during the training phase, the quality of a combination being computed as the average of the accuracy rates estimated for each of the 10 divisions of the data set. Therefore, on the training set, 10-folds cross-validation has been performed in order to optimize: the hyper-parameters set (C, γ) or (C, d) and the kernel type: POL or RBF for SKs and the best combination was indicated by the best balanced accuracy rate. The result of the parameter selection process is that the classifier will be able to predict accurately the unknown data.

In table 3.6 the results obtained for the visible monomodal system and for the infrared monomodal system are provided. In these cases the input vector is the one corresponding to the respective domain: VIS_{171} or IR_{171} and here no-fusion scheme is applied. For the two datasets mentioned in section 3.2.2 (i.e. where the object instances were assigned to one of the 4 classes or 8 classes) we computed 3 different classification problems:

- the first one is considering the classification in 4 classes: pedestrian, vehicle, cyclist and background, denoted (P, V, C, B) and it is based on the inputs corresponding to the 4 classes of objects. To remind, the objects' distribution in classes is: (321, 329, 45, 237) for the training set and (79, 83, 11, 59) for the test set.

- the second classification problem is considering the classification in 8 classes of objects pedestrian entire, pedestrian occluded, pedestrian group, vehicle entire, vehicle occluded, vehicle group, cyclist and background, briefly noted (PE, PO, PG, VE, VO, VG, C, B) and starts from the data corresponding to the 8 classes, where the distributions are: (206, 65, 50, 133, 131, 65, 45, 237) in the training data set, and (51, 16, 12, 34, 33, 16, 11, 59) in the test set.

- the last classification problem starts from the data corresponding to 8 classes of objects, classifies the test data in 8 classes, but then it merges the results in 4 classes. In this manner, objects recognized as pedestrian entire, pedestrian occluded, pedestrian group (PE, PO, PG) will all be classified as belonging to class pedestrian (P) and objects recognized as vehicle entire, vehicle occluded, vehicle group (VE, VO, VG) will all be classified as belonging to class vehicle (V).

To these 3 classification problems, we will refer in the following part of this thesis, as: $ClasifPb_{4cls}$, $ClasifPb_{8cls}$ and $ClasifPb_{8cls \rightarrow 4cls}$.

The results obtained in the 10f-CV process are important for the optimization process, because on each modality the maximum balanced accuracy will be searched for on the training set. As we mentioned before, in our experiments two types of kernels are possible for the SVM SKs, and they are RBF or POL. From all the balanced accuracies obtained in the 10f-CV step for both types of kernels, the maximum one has been picked for each modality on the training set and the corresponding kernel has been called the "winner" one. For these winner SKs, the system is trained with all the data from the training set and tested with the test data, therefore the LT procedure is performed. The accuracies obtained on the training set are also presented in table 3.6, but they are mentioned only to see that data are not suffering from overfitting.

From table 3.3, we have for the classification problem with 4 classes $ClasifPb_{4cls}$ a balanced

Table 3.6: Single kernel (SK) optimization based on accuracies provided by different FVs and obtained for different classification problems

Method		Classification problem		Performance	
		Pb.	Input vector	bAcc [%]	winner SK $SK^*(SKtype, C, SKparam.)$
Train 10f-CV	4 classes		VIS_{171}	90.0	$(RBF, 2^7, 2^{-15})$
			IR_{171}	92.2	$(RBF, 2^9, 2^{-19})$
Test LT			VIS_{171}	94.7	-
			IR_{171}	94.9	-
Train 10f-CV	8 classes		VIS_{171}	75.9	$(RBF, 2^9, 2^{-19})$
			IR_{171}	81.3	$(RBF, 2^9, 2^{-19})$
Test LT			VIS_{171}	76.5	-
			IR_{171}	76.3	-
Train 10f-CV	8 cls->4cls		VIS_{171}	89.9	$(RBF, 2^3, 2^{-15})$
			IR_{171}	92.2	$(RBF, 2^7, 2^{-17})$
Test LT			VIS_{171}	93.4	-
			IR_{171}	89.3	-

accuracy of 83.7 for the vector comprising all the features from the visible domain VIS_{171} and $bAcc = 88.0$ for the vector comprising all the features from the infrared domain IR_{171}. For the classification problem with 8 classes $ClasifPb_{8cls}$ from table 3.4, we have $bAcc = 73.9$ for VIS_{171} and $bAcc = 79.3$ for IR_{171}. When comparing these values with the ones from table 3.6 corresponding to the same classification problems but rendered by the SVM classifier, it can be noticed that in all cases the SVM provides better results, at least with 2% higher than those obtained with the kNN. Therefore, our choice for the SVM is motivated by the obtained results. In addition, it has to be mentioned that the SVM is more parametrizable and thus better suited for the fusion schemes we propose.

Next, it can be noticed that when comparing the results obtained for the classification problem with 4 classes $ClasifPb_{4cls}$ with the ones acquired for the classification problem with 8 classes $ClasifPb_{8cls}$, the first set is better with at least 10% than the second one. Even when considering a classification in 8 classes and than the results further adapted to 4 classes ($ClasifPb_{8cls \to 4cls}$), the results are better than the ones obtained with 8 classes ($ClasifPb_{8cls}$) with at least 10%. When comparing the classification problems $ClasifPb_{8cls \to 4cls}$ and $ClasifPb_{4cls}$, the later one provide higher accuracies. Therefore, when analyzing these situations, we decided to perform all the processing from the next part of this thesis with a single classification problem, the one providing best results, i.e. the classification problem with 4 classes of objects: pedestrian, vehicle, cyclist and background. The other 2 classification problems are not well suited due to the small number of objects per class (too few instances per class of objects). When a greater database will be available, i.e. with more instances for each class, we will return to these classification problems because they could provide even better results than the classification problem with 4 classes due to their smaller intra-class variability.

Even the accuracies are not very distant, i.e. the ones obtained for the FV VIS_{171} are just a little bit smaller than ones provided by the FV IR_{171}, means that the objects have not always been assigned to the same classes in both modalities. In order to study this problem in a more profound manner and take into account the VIS-IR complementarity, we intend to approach the classifier outputs at an early level, i.e. at the scores level, not directly at that of classes. In this manner, we believe a fusion of this information (i.e. the matching scores) will provide better results than each of the monomodal VIS or IR system. We believe that even the fusion at some other levels, i.e. the features or kernels, could provide improved results.

As we have noticed in section 3.2.5, the computation time from the Obstacle Recognition module could be decreased by reducing the dimension of the FV. We propose to analyze this problem in the next chapter, dedicated to the features selection. Even the SVM classifier have been proven to be better than the kNN one on the FV comprising 171 features, we have to analyze if it will provide also good results on the vector obtained after performing the features selection.

3.4   Conclusion

For our system, different global texture features have been extracted from the visible and infrared images in order to compute feature vectors (FVs) for both types of data, VIS and IR. Even the number of features in our case is not very high for an object, when considering all the objects within an image, it has to be reduced to render a real-time system. Therefore, the number of features could be decreased in order to obtain a new FV containing only the most relevant features for our classification problem. This process of data reduction is needed for time reduction purposes, whether it is a features extraction time, or a system learning time, or the time needed to classify obstacles. All these values for the time will be reduced once the FV dimension is decreasing. Therefore, we envisioned the reduction of the feature vector in order to obtain the smaller time as possible in the obstacle recognition module, and thus to help the entire ODR system to perform the detection and the classification of obstacles in real time. In the next chapter (chapter 4) we will see which is the proposed solution to perform this reduction of the FV dimension, i.e. the use of some features selection procedures, without decreasing classification performances.

From the features extraction module, a numeric description, including different types of features (wavelets, statistical features, coefficients of some transforms, among others) has been obtained for each object image. We choose to extract features as rich and diverse as possible in order to have the advantage of some sort of complementarity concerning them. We do not have to ignore the possibility of some redundant information which therefore will have to be eliminated. This elimination of redundant features would be performed in a features selection step (described in the next chapter) which main purpose would be to find a set or a sub-set of features more compact and more relevant for the classification task. This step could be also seen as a reduction of the learning complexity.

The systems analyzed by now, are adapted to a single modality, therefore they are monomodal systems; even if they provide global recognition rates on the entire test set very closed on the two modalities, we will show that these results could be improved by the combined processing of the VIS and IR information, which means in the frame of a bimodal system. The bimodal systems could take different forms in function of the level at which the information is combined or fused. Thus, we propose three different fusion systems: at the levels of features or SVM's kernels, or even higher, at the level of matching-scores provided by the SVM. Each one of these systems could render improved results comparing to the monomodal systems. We intend to analyze this in what follows, in chapter 5. In addition, even the performances obtained with the monomodal systems are very good (above 90%), we aim higher ones. It has to be considered that these results have been obtained in privileged conditions: they were registered during daytime and there were not too much objects captured in a scene-image (there was not quite urban cluttered environments). As well, the detection phase has been realized by providing the BBs manually annotated. All these, will be further degraded when real conditions of functioning will be reached. This is the reason why we aim even a higher accuracy rate for the recognition module. In addition to all these motivations, the monomodal systems do not allow the adapted functioning to the context. By using a fused VIS-IR system this issue can also be solved. Therefore, we envisioned a bimodal system which dynamically adapts to new environmental situation.

CHAPTER 4
Features selection

Contents

In the previous chapter it was noted that in order to represent the obstacles' images which have to be recognized by the ODR system, some features have been preferred to represent this information. These features are obtained in the features extraction module and they can be wavelet features, statistical features, the coefficients of some transforms, or others, the more varied, the better as they can retain much complementary information. Generally, the features extraction module is followed by a features selection one, in which the importance of these features is estimated and only the ones that are most relevant will be chosen to represent the information. In this chapter different Features Selection (FS) methods are tested and compared in order to evaluate the pertinence of each feature (and of each family of features) in relation to our objective of obstacle classification.

This chapter is structured in two main parts, the first one is presenting a motivation for why this step of features selection is needed (section 4.1), and then the main possibilities to accomplish this task (section 4.2) are given. For the FS step there are multiple criteria, concentrated on two fundamental directions which differ mostly by their evaluation method: filters and wrappers. For any filter method an attribute evaluator (applied for individual features (subsection 4.2.2) or subset of features (subsection 4.2.3)) and a search method (detailed in subsection 4.2.1) should be mentioned. Our method to perform the features selection is described in section 4.3 and the last part is presenting the experiments we realised in order to perform the selection of features by the methods previously mentioned (section 4.4). The chapter is ending with conclusion in section 4.5.

4.1   Motivation for Features Selection

In many supervised learning tasks the input data are represented by a large number of features, but only a few of them are relevant for predicting the outcome or the target label. The main issue is that even the best classification algorithms cannot assure a well functioning in situations where the information is represented by a large number of weakly relevant or irrelevant features. In addition, besides the low accuracy rate, even the classification time could be very much increased compared to a situation in which a small set of relevant features is used to represent the same information. On the other hand, once a good small set of features has been chosen, even the most basic classifiers (e.g. kNN) can achieve high-level performances. Therefore, the FS process (i.e. the task of choosing a smaller subset of features which is adequate to predict the target labels) is decisive/essential for the implementation of an efficient learning/testing system.

Although the FS step is generally performed to select a subset of relevant features to further describe the same information, it can have other motivations, including:

- performance improvement, to gain in predictive accuracy, especially when redundant data are present before appling the FS step;

- data understanding, to simply understand or visualize the data or the process that generates the data.

In the literature there are many systems in which the feature construction stage does not consider the FS operation, but only the features extraction one. As it can be seen from figure 3.3, we applied the FS step after the features extraction operation and in the following we will demonstrate that the performances from both viewpoints (accuracy of the classification and processing consumed time) are improved by this FS operation.

Two main problems can appear in the stage of feature vector construction, and the first one refers to the impossibility of the algorithm to generalize over the entire data set. Even if it will provide very good results on data belonging to the learning and validation set, it would not accurately predict the test set (this stands for the overfitting problem). As we saw in the previous chapter, data obtained after the features extraction step are not overfitted, therefore we will demonstrate that also after the FS step they are still proper for the classification process. The second problem which can appear is regarding the consumption time during the entire classification process (including the tasks of features extraction, features selection, fusion and classification). From this point of view, the FS step will help in obtaining a lower processing time in the test stage. After passing through a FS module, the accuracy could be increased - if in the initial feature vector there were some features contradicted one eachother and these were rejected in the FS step, or the accuracy could be decreased if too few features have been retained and there is not enough information to characterize the data.

The features selection step is very important and it has to be done before the classification stage itself. The experiments have proved us that finding a convenient set or ensemble of features is as important as finding the best classifier to be used on that set of data. If we do not reach a compact and pertinent numeric representation of data, even the most performant classifier would not succeed to compensate the deficit. As mentioned in Chapter 3, two operations are important for a classification problem: (1) finding the proper FV and (2) finding the best classiffier to process the respective FV. These two main issues were addressed also in this chapter as follows: both choosing the proper FV (through the operation of features selection) and choosing the best classifier (on the training set) for the previously obtained FV. The results are presented in the experiments section. As in the previous chapter, the FV was constructed from the results provided by different features extraction algorithms. Next, a new FV provided by the application of the FS method was constructed. On the obtained FV, comprising only the selected features, the choice for the best SK for the SVM has been performed in the same manner as it was realised in the previous chapter for the monomodal vectors comprising all 171 features.

4.2   Methods for Features Selection

For the features selection step there are multiple classification criteria, concentrated on two fundamental directions which differ mostly by the evaluation criteria:

(a) *Filters* - these methods select a set or a subset of features by the evaluation of the general properties of data (the entire ensemble of data is filtered in order to provide the most promising subset of features), independent of the classification stage. Filters use criteria not involving any learning machine, e.g. a relevance index based on correlation coefficients or test statistics, and others.

(b) *Wrappers* - use the performance of a learning machine trained using a given feature subset. The respective learning machine is included (or wrapped) in the selection procedure as a "black box" to score subsets of features according to their predictive power.

The wrapper methods (especially when using the cross-validation method for the evaluation) are much more time consumers, therefore we choose not to use these type of feature selection methods in the frame of our system. Even if the learning step of the system is performed off-line, there are multiple loops in its optimization and where is possible, we choose to use rapid algorithms instead of slow ones.

Filters provide the cheapest approach to the evaluation of feature relevance, because they select features without optimizing the performance of a predictor and they provide a complete order of the features using a relevance index. Methods for computing ranking indices include correlation coefficients (which assess the degree of dependence of individual variables with the target), or other statistics (Chi-squared, T-test, F-test among others). Therefore, in the following, we will concentrate only on filter methods, which include more possible approaches:

 i) *Single-attribute evaluators* - the evaluation of the importance of each feature is done by considering only the individual predictive capacities of each feature. These approaches provide a ranked list of the importance of each feature assessed by the used criteria.

 ii) *Attribute subset evaluators* perform the evaluation of the importance of each subset of features.

Filter (but also wrapper) methods can make use of search strategies to explore the space of all possible feature combinations that is usually too large to be explored exhaustively. Search methods traverse the attribute space to find a good subset, while the quality is measured by the chosen attribute evaluator. Therefore, for any filter method we employed an *attribute evaluator* (applied for individual features i) or subset of features ii)) and a *search method* should be mentioned.

For the search methods working with subset evaluators ii) there are two possible selection procedures: *ascending* or *forward selection* where features are added progressively and *descending* or *backward selection* where features are discarded progressively. In a forward selection method one starts with an empty set and progressively adds features yielding to the improvement of a performance index. In a backward elimination procedure one starts with all the features and progressively eliminates the least useful ones. These selection procedures may lead to different subsets and, depending on the application and the objectives, one approach may be preferred over the other one. In the experiments we choose the forward direction, because it is the method most used in practice.

In the following, we will detail the search methods and the attributes evaluators used in the FS step.

4.2.1   Search methods

Search methods get through the attribute space to find a good subset, but the quality of the respective subset is measured by an attribute subset evaluator. The most utilised search methods are presented and briefly described in the following:

4.2.1.1   Best First

The Best First method search the features space by greedy hill climbing combined with a backtracking facility. This method explores the entire space of features, without being interrupted by any stopping criteria. It does not stop when the performance is reduced, but it has a parameter that specify how many consecutive nonimproving nodes must be encountered before the system backtracks. Therefore, a Best First search will explore the entire search space and involves limiting the number of fully expanded subsets that result in no improvement. It can search forward from the empty set of attributes, backward from the full set, or start at an intermediate point and search in both directions by considering all possible single-attribute additions and deletions (Witten & Frank, 2005).

4.2.1.2   Linear Forward

This search method takes a restricted number of k attributes into account. Fixed-set selects a fixed number k of attributes, whereas k is increased in each step when fixed-width is selected. The search uses either the initial ordering to select the top k attributes, or performs a ranking (with the same evaluator the search uses later on). The search direction can be forward, or floating forward selection (with optional backward search steps) (M. Gütlein, 2006).

4.2.1.3   Genetic Search

This method uses a simple genetic algorithm described in (Goldberg, 1989). In order to select the important features, the Genetic Search is inspired from the genomics evolution. Like the genomic characterize an individual, also some selected features from an entire set of features could characterize the data set. The algorithm makes that a population randomly initialized by some crossovers and mutations to evolve by the elimination of the individuals which minimize the weights. This expensive search is very sensitive to the parametrization, but it provides a global solution, unlike the Greedy Stepwise method which is more a local-optimal one (Witten & Frank, 2005).

There are some **other searching methods** (e.g. *Exhaustive Search*, *Random Search*) but generally, they offer poorer results and they are much time consumers. For example, *Random Search* randomly searches the space of attribute subsets. If an initial set is supplied, it searches for subsets that improve on (or equal) the starting point and have fewer (or the same number of) attributes. Otherwise, it starts from a random point and reports the best subset found (Liu & Setiono, 1996). *Exhaustive Search* is another possible method, which searches starting from the empty set, and reports the best subset found. If an initial set is supplied, it searches backward from this starting point and reports the smallest subset with a better (or equal) evaluation (Witten & Frank, 2005). Like the Best First method, there is the *Greedy Stepwise* one, which may progress forward from the empty set or backward from the full set[1]. This Greedy Stepwise method searches greedily through the space of attribute subsets, selecting an descriptor in each iteration. Unlike Best First method, it does not backtrack, but terminates as soon as adding or deleting the best remaining attribute decreases the evaluation metric. In (Guyon & Elisseeff, 2003) it is mentioned that this method is not optimal, because there are features showing a great discrimination power only in the presence of some other features. *Rank Search* - This method sorts attributes using a single-attribute evaluator and then ranks promising subsets using an attribute subset evaluator. It starts by sorting the attributes with the single-attribute evaluator and then evaluates subsets of increasing size using the subset evaluator-the best attribute, the best attribute plus the next best one, and so on. This procedure has low computational complexity: the number of times both evaluators are called is linear in the number of attributes. Using a simple single-attribute evaluator, the selection procedure is very fast. We did not use this scheme because the results are the same as in the case of Ranker, which is described next.

---

[1]Because the Greedy stepwise method has provided identic results with Best First (in some cases maybe one or two features were different, but all the rest were the same), we decided to replace it with the Linear Forward Selection, which is an extension of BestFirst.

In the experiments results section, three from the previous mentioned search methods (i.e. Best First, Linear Forward and Genetic Search) would be used in combination with an attribute subset evaluator (ii)) and the number of selected features will be determined by the combination of these two procedures.

Finally we describe the *Ranker* method, which is not a search method for attribute subsets, but a ranking scheme for individual attributes.

4.2.1.4   Ranker

It sorts attributes by their individual evaluations and must be used in conjunction with one of the single-attribute evaluators (i)) and not with an attribute subset evaluator (ii)). Ranker not only ranks attributes but also performs attribute selection by removing the lower-ranking ones. A cut-off threshold (below which attributes are discarded) or the number of attributes to be retained could be mentioned.

4.2.2   Single-attribute evaluators

Single-attribute evaluators (i) are used with the Ranker search method to generate a ranked list of features from which Ranker discards a given number. In the same manner, they can be used in the Rank Search method. In the case of single-attributes evaluators the main idea is to order the features regarding to a weight value (the feature importance) determined from a criteria. There are many criteria which could be used and they are briefly reviewed in the following.

4.2.2.1   Chi^2 (χ^2)

Chi Squared attribute evaluator computes the chi-squared statistic with respect to the target. $\chi^2(y,x)$ measures the dependence (or independence) between the attribute y and the target x. If the attribute y and the class x are independent, then $\chi^2(y,x)$ is nul. Generally, the following weight is used

$$\chi^2(y,x) = \frac{N[P(y,x)P(\neg y, \neg x) - P(y, \neg x)P(\neg y, x)]^2}{P(y)P(\neg x)P(y)P(\neg x)}, \tag{4.1}$$

where $P(y,x)$ is the probability that the attribute y would have an influence on the target x, $P(\neg y, x)$ is the probability that the absence of the attribute y have an influence on class x. N represents the number of examples (images in our case) from the training set (on which the main selection statistics are calculated). Finally, when the results are averaged above all the classes (M), the weight corresponding to the selection of attribute y will be:

$$\chi^2(y) = \sum_{k=1}^{M} P(x)\chi^2(y,x). \tag{4.2}$$

4.2.2.2   Information Gain and Information Gain Ratio

Information theory indices are most frequently used for feature evaluation: *information Gain* or "the Kullback-Leibler divergence" evaluates attributes by measuring their information gain with respect to the class or the target. This method ranks features according to the mutual information between each feature and the labels. Recall that the mutual information between two random variables X, Y is defined as $I(X,Y) = \sum_{x,y} p(x,y)log[p(x,y)/(p(x)p(y))]$. Information (entropy) contained in the *class distribution* could be written as: $H(Y) = -\sum_{i=1}^{K} P(y_i)log_2 P(y_i)$, where $P(y_i) = m_i/m$ is the fraction of samples x from class y_i, $i = 1...K$. The same formula is used to calculate information contained in the

discrete *distribution of feature X values*: $H(X) = -\sum_i P(x_i)log_2 P(x_i)$.

Information contained in the *joint distribution of classes and features*, summed over all classes, gives an estimation of the importance of the feature and it could be written:

$$H(Y,X) = -\sum_i \sum_{j=1}^{K} P(y_j, x_i)log_2 P(y_j, x_i) \tag{4.3}$$

where $P(y_j, x_i)$, $j = 1...K$ is the joint probability (density for continuous features) of finding the feature value $X = x_i$ for vectors x that belong to some class y_j and $P(x_i)$ is the probability (density) of finding vectors with feature value $X = x_i$. Low values of $H(Y,X)$ indicate that vectors from a single class dominate in some intervals, making the feature more valuable for prediction. Information is additive for the independent random variables. The difference $MI(Y,X) = H(Y) + H(X) - H(Y,X)$ may therefore be taken as "mutual information" or "information gain". Mutual information is equal to the expected value of the ratio of the joint to the product probability distribution, that is to the Kullback-Leibler divergence:

$$MI(Y,X) = -\sum_{i,j} P(y_j, x_i)log_2 \frac{P(y_j, x_i)}{P(x_i)P(y_j)} = D_{KL}(P(y_j, x_i)|P(y_j)P(x_i)) \tag{4.4}$$

where the Kullback-Leibler divergence is defined as:

$$D_{KL}((P(X)\|(P(Y)) = \sum_i P_Y(y_i)log\frac{P_Y(y_i)}{P_X(x_i)} \geq 0, \tag{4.5}$$

A feature is more important if the mutual information $MI(Y,X)$ between the target and the feature distributions is larger. Decision trees use closely related quantity called "information gain" $IG(Y,X)$. In the context of feature selection this gain is simply the difference $IG(Y,X) = H(Y) - H(Y|X)$ between information contained in the class distribution $H(Y)$, and information after the distribution of feature values is taken into account, that is the conditional information $H(Y|X)$. This is equal to $MI(Y,X)$ because $H(Y|X) = H(Y,X) - H(X)$. A standard formula for the information gain is easily obtained from the definition of conditional information:

$$\begin{aligned}IG(Y,X) \quad &= MI(Y,X) = H(Y) - H(Y|X) \\ &= H(Y) + \sum_{ij} P(y_j, x_i)log_2 P(y_j|x_i) = H(Y) - \sum_{ij} P(x_i)[-P(y_j|x_i)log_2 P(y_j|x_i)]\end{aligned} \tag{4.6}$$

where the last term is the total information in class distributions for subsets induced by the feature values x_i, weighted by the fractions $P(x_i)$ of the number of samples that have the feature value $X = x_i$.

Various modifications of the information gain have been considered in the literature on decision trees (J.R. Quinlan, 1993), aimed at avoiding bias towards the multivalued features. These modifications include information gain ratio (described in the following) and symmetrical uncertainty (described in subsection 4.2.2.5).

Gain Ratio attribute evaluator computes the worth of the attributes by measuring their gain ratio with respect to the class. Its computation consists in dividing the information gain by the associated entropy,

$$IGR(Y,X) = \frac{MI(Y,X)}{H(X)} = \frac{H(Y) - H(Y|X)}{H(X)} = \frac{H(X) + H(Y) - H(X,Y)}{H(X)}. \tag{4.7}$$

Like the information gain, the information gain ratio is a low-consumption time criteria.

4.2.2.3 ReliefF - Recursive Elimination of Features

ReliefF attribute-evaluator is an instance-based technique: it samples instances randomly and checks neighbouring instances of the same and different classes. It operates on discrete and continuous class data. Parameters specify the number of instances to sample, the number of neighbors to check, whether to weight neighbors by distance, and an exponential function that governs how rapidly weights decay with distance.

The Relief algorithm was first proposed by (Kira & Rendell, 1992), and then analyzed in (Guyon *et al.*, 2006). The algorithm holds a weight vector over all features and updates this vector according to the sample points presented. The Relief feature selection algorithm was shown to be very efficient for estimating features quality, Kira & Rendell proving that under some assumptions, the expected weight is large for relevant features and small for irrelevant ones.

Relief was initially designed to treat the binary classification problems: in each step of the algorithm's loop, one vector from the training set is considered together with its closest neighbors. The importance of each such vector is updated in the following manner:
  • if the vector has the same class as its neighbour, the weights of the features having different values in these two vectors will be decreased (the respective features does not illustrate the fact that the two vectors belong to the same class).
  • if the vector has a different class compared by its neighbour, the weights of the features having different values in these two vectors will be increased (the respective features will surely illustrate the fact that the two vectors belong to some different classes).

Relief was extended to deal with multi-class problems by (Kononenko, 1995) and the new version is called Relief-F. Instead of using the distance to the nearest point with an alternative label, ReliefF looks at the distances to the nearest instance of any alternative class and takes the average.

The Relief method is a classical example of **multivariate filter**. The so-called "multivariate" methods take into account feature dependencies and potentially achieve better results because they do not make simplifying assumptions of variable/feature independence. Most multivariate methods rank subsets of features rather than individual features. The ranking index derived from the Relief algorithm is:

$$C(j) = \frac{\sum_{i=1}^{m} \sum_{k=1}^{K} |x_{i,j} - x_{M_k(i),j}|}{\sum_{i=1}^{m} \sum_{k=1}^{K} |x_{i,j} - x_{H_k(i),j}|}. \tag{4.8}$$

The algorithm is based on the K-nearest-neighbors from the same class, and the same number of vectors from different classes. To evaluate the index, for each example x_i, the K closest examples of the same class $x_{H_k(i)}$, $k = 1...K$ (nearest hits) and the K closest examples of a different class $x_{M_k(i)}$ (nearest misses) are identified in the original feature space (all features are used to compute the closest examples). Then, in projection on feature j, the sum of the distances between the examples and their nearest misses is compared to the sum of distances to their nearest hits. In equation 4.8, the ratio of these two quantities was used to create an index independent of feature scale variations.

4.2.2.4 Significance

It evaluates the worth of an attribute by computing the Probabilistic Significance as a two-way function, like attribute-class and class-attribute association. If an attribute is significant, then there is a strong possibility that elements with complementary sets of values for this attribute will belong to complementary sets of classes, and alternatively, given that the class decisions for two sets of elements are different, it is expected that the significant attribute values for these two sets of elements should also be different (Ahmad & Dey, 2005).

4.2.2.5 Symmetrical Uncertainty

This method evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class. One possibility to discard the redundant features is to choose a subset of individual features correlated with the target, but having a low correlation between eachother. The correlation between 2 features *A* and *B* could be measured using the symmetrical uncertainty

$$SU(Y,X) = 2 * \frac{gain}{H(X) + H(Y)} = 2\frac{MI(Y,X)}{H(X) + H(Y)} = 2 * \frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)} \in [0,1] \qquad (4.9)$$

where H(X,Y) is the entropy function of *A* and *B*, and it is calculated starting from the common probabilities of each combinations of values *A* and *B*. The symmetric uncertainty always lies between 0 and 1. The symmetrical uncertainty coefficient seems to be particularly useful due to its simplicity and low bias for multi-valued features (Hall, 1999a).

The inconvenient with these methods is that the result is a list of ranked features, from which someone would have to choose the number of features to retain. The winner features could be decided by an integer number specifying the number of features to retain or by a threshold specifying the value above which the features are retain. In the next section (at the experimental results) we will see the thresholds that have been considered together with the values illustrating their efficiency (by the accuracy measure).

4.2.3 Attribute subset evaluators

Subset evaluators take a subset of attributes and return a numeric measure that guides the search. The approaches combining a feature *search method* and *a feature-subset evaluator* measures the quality of a descriptor by considering a context: the other attributes. These approaches generate some candidate solutions by the subset searching method and evaluate the performance with the evaluator, which asses a weight to the generated subset of features. Also, the evaluator allows a comparative analysis between all the generated subsets of features in order to conduct the search and finally choose the winner subset of features.

Generally, one of the two following methods are used as an attribute subset evaluator.

4.2.3.1 CFS (Correlation-based feature subset evaluator)

Correlation-based Feature subset Evaluator (CFS) subset evaluator assesses the predictive ability of each attribute individually and the degree of redundancy among them (Hall, 1999b). The weight associated to a subset of features is calculated by the correlation matrix. Generally, subsets of features having low intercorrelation but highly correlated with the class are preferred. CFS subset evaluator imposes a ranking on feature subsets in the search space of all possible feature subsets. A forward selection, which begins with no features and greedily adds one feature at a time until no possible single feature addition results in a higher evaluation was used in the frame of this thesis. The search will terminate if five consecutive fully expanded subsets show no improvement over the current best subset.

If a group of *k* features has already been selected, correlation coefficients may be used to estimate correlation between this group and the class, including inter-correlations between the features. Relevance of a group of features grows with the correlation between features and classes, and decreases when inter-correlation grows. The CFS algorithm is based on equation 4.10, calculating average correlation coefficients between features and classes and between different features. Denoting the average correlation coefficient between these features and the output variables as r_{ky} and the

average between different features as r_{kk} the group correlation coefficient measuring the relevance of the feature subset is defined :

$$J(X_k, Y) = \frac{kr_{ky}}{\sqrt{k + (k-1)r_{kk}}}.$$

(4.10)

It is usually thought that feature correlation (or anticorrelation) means feature redundancy, but this is not true: there are examples in which a perfect separation is achieved using two features while each individual feature provides a poor separation.

4.2.3.2   CSE (Consistency-based feature subset evaluator)

It evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes. The method implies the taking out of 10% from the total number of instances (the training data), run the algorithm and check the inconsistency criterion based on its selected features on the remaining 90% of the data. Then, add those patterns causing inconsistencies to the training data and re-run the algorithm. This process continues until the number of inconsistencies is below a tolerable value.

The Consistency-based feature Subset Evaluator (CSE) method evaluates the value of a subset of features, the purpose being to find the smallest subset of features which best identify the examples of a class with the same consistency as the complete set (Liu & Setiono, 1996). The inconsistency phenomenon appears when two or more vectors with the same subset of feature values are associated with different classes. The inconsistency count is equal to the number of samples with identical features, minus the number of such samples from the class to which the largest number of samples belong.   Summing over all inconsistency counts and dividing by the number of samples the inconsistency rate for a given subset is obtained. This rate is an interesting measure of feature subset quality, being monotonic (it decreases when the feature subsets increases).

A feature is relevant if it contains some information about the target. Relevance indices discussed in the previous sections treat each feature as independent (with the exception of Relief algorithm and the CFS coefficient), allowing for feature ranking. Those features that have relevance index below some threshold are filtered out as being not useful. There are limitations to individual feature ranking, because of the underlying feature independence assumptions made by "univariate" methods: features that are not individually relevant may become relevant in the context of others, and features that are individually relevant may not all be useful because of possible redundancies (Guyon *et al.*, 2006).

To conclude, by selecting a single-attribute evaluator (like ChiSquared, GainRatio and others) accompanied by the ranker method, a potentially faster but less accurate approach is reached. It evaluates the attributes individually and sort them, discarding attributes that fall below a chosen cut-off point. On the other side, by combining one attribute subset evaluator (like CFS or CSE) with one search method (like Best first, Greedy among others) attribute selection is normally done by searching the space of attribute subsets.

The ensemble attribute evaluators **(i)** provide an ordered list of features according to the importance given by the use criteria. On this list of features, generaly some threshold criteria are used, in order to select only a part of it. On the other side, the individual attribute evaluators **(ii)** select a fixed subset of features using some methods of searching and generation of ensembles of features. Both methods were selected to be tested in the following, and all the methods we employed in our processing are summarized in table 4.1.

4.3   Our proposed method for Features Selection (FS)

In this section we will see how the features selection methods described by now are generally applied and we will next present our proposed method for performing the features selection. Generally, the FS methods are applied directly, on the entire set of data from the training, but we propose also the application by a cross-validation procedure, due to the advantages it implies.

Cross-validation is generally used for model selection (e.g. find the best classifier or kernel, find the best regularization hyperparameters), but it could be also applied for features selection, since cross-validation provides an estimate of the generalization ability of models. Actually, the prediction ability of models depends both on the features used and on the complexity of the model. We have considered to use the cross-validation process in the FS stage too, and indeed the time needed for the algorithm to perform the selection process is much more increased compared to the situation in which no cross-validation process is applied. Still, the obtained benefits are both from the accuracy of the recognition and the classification time. It is essential to estimate the computational burden of algorithms for features selection problems, because the computational time is essentially conducted by the search strategy and by the evaluation criteria. The evaluation criteria may also be expensive as it may involve training a classifier or comparing every pairs of examples or features. The fact that this processing of FS performed in the cross-validation loop takes much more time than another one not using the cross-validation process is not critical for our system, because the FS operation is performed off-line, when the system is not running on the road, therefore no real-time operating is required in this stage.

In order to understand how the FS methods described in the previous section were applied in the frame of our experiments, we grouped all of them in table 4.1. There are two sets of methods, the first set is comprising the methods noted with FS1→FS6 and the second set contains the other 6 methods, denoted FS7→FS12. This last set is further divided in two groups, depending on the procedure the features selection methods has been applied[2]:

   - by a cross-validation technique - the FS method was applied on each individual fold, therefore there is a number equal to the number of folds for the application of the respective FS method; in our case a number of 100 folds have been chosen and we denote this situation 100f-CV, or

   - on the entire training set of data - when the FS method was applied only once. This situation is denoted FullTS.

The attribute selection method for the cases FS1→FS6 was applied using the full training set which means that the FS method was applied on all the data from the training set at a time. Therefore, a list of ranked features will be obtained as result for these methods. The first attribut from this list would have the greater importance, the second one from the list would have a lower importance and so forth, the importance being decreased as we go further to the end of the list.

Unlike this, for the methods FS7→FS12 if the FS method would be applied directly on the full training set, the result would be an ensemble of features (in the same manner as the two methods CFS and CSE have been previously described), provided ordered, in function of features' name. Therefore, no information about the importance of the features in the subset will be given. In addition, the number of features selected is very small. As we previously mentioned, we considered to perform the FS for the cases FS7→FS12 through the cross-validation process because of 2 reasons: (1) the result will be a list of features containing more attributes than in the case of the application of the same method on the full training set and, in addition, (2) each feature will be accompanied by a value showing in how many folders (from the total number of $k = 100$ folders) the respective feature has been selected as being relevant. Therefore, values different from zero will indicate features important or relevant in a specific degree or rank (discretized values with the step of 1 in the domain

---

[2]In both cases the same data set, i.e. the training one, is used when applying the FS method.

Table 4.1: Features Selection (FS) methods

Evaluators		(i) Single-attribute Evaluators						(ii) Subset-attribute Evaluators			
		Chi Squared	Information Gain	Ratio of the Info. Gain	ReliefF	Significance	Symmetrical Uncertainty	Correlation (CFS)		Consistency (CSE)	
Attribute Selection Mode		Full TS						Full TS	100f-CV	Full TS	100f-CV
Search Approaches	Ranker	$FS1_{TS}$	$FS2_{TS}$	$FS3_{TS}$	$FS4_{TS}$	$FS5_{TS}$	$FS6_{TS}$	×	×	×	×
	Search — Best First	×	×	×	×	×	×	$FS7_{TS}$	$FS7_{CV}$	$FS10_{TS}$	$FS10_{CV}$
	Search — Linear Forward	×	×	×	×	×	×	$FS8_{TS}$	$FS8_{CV}$	$FS11_{TS}$	$FS11_{CV}$
	Search — Genetic Search	×	×	×	×	×	×	$FS9_{TS}$	$FS9_{CV}$	$FS12_{TS}$	$FS12_{CV}$

$[0; 100]$).

By including the FS process in the cross-validation loop, the number of selected features will be determined as the number of elements of the reunion of all sets selected in each of 100 folders of the cross-validation loops. Therefore, to produce the final FV, it is taken into account all set of training data, but it is comprised in the 100 folds. The FS process may be repeated for many sub-samples of the training data and the union of the subsets of features selected may be taken as the final "stable" subset. An index of relevance of individual features can be created considering how frequently they appear in the selected subsets. This manner to apply the FS operation, by the cross-validation procedure would be very helpful (by creating this index of relevance for each feature) especially in the cases in which the FS method does not provide such information. In our situation, these are the FS methods based on Search, i.e. FS7→FS12. As in the case of methods FS1→FS6 exists such an index of relevance (provided directly by the FS method), also in the case of methods FS7→FS12 it will be available by the application of the FS method by a cross-validation procedure. Using this index of relevance, the features would be easier evaluated individually.

Because it is not known beforehand which method of FS will provide best results, instead of selecting a single FS method from the very beginning, we propose to evaluate multiple FS methods, to evaluate their results and only after this step to choose the proper one for our problem.

If two FS methods will provide two different FVs but with the same accuracy rate, the FV having a smaller number of coefficients should be chosen as the winner one. A smaller number of features could lead to the elimination of some entire families of features (especially if the reduction is a severe one) and in this case the time to compute the new FV will be smaller. Another possibility to reduce the extraction time is that when discarded the features, some of them to be separable ones. The best scenario will be that when before ending the FS process if a majority of features from a family is retained or discarded, then all the features from that family to be retained or discarded by the moment when that FS operation will be stopped.

We have at our disposal the methods provided in table 4.1, from which some of them are capable to provide an index of relevance directly (these are the methods FS1→FS6), while some of them can give such an index only indirectly, by the use of the cross-validation procedure (these methods are FS7→FS12). In the following, we propose to evaluate the methods given in table 4.1.

If in the case of the methods FS1→FS6 we chose to use thresholds, we will have at our disposal also the possibility to apply some thresholds for the methods FS7→FS12, because both types of methods will provide rank values. These thresholds will be employed to select more or less features from the ones already rendered by the FS method.

4.4   Experiments and results

From the previous chapter, we recall figure 3.3 where the feature construction module was comprising a features extraction and a features selection (FS) steps. In this chapter, we consider two types of FVs, which we called monomodal and bimodal. The monomodal FV is the one obtained on the VIS or IR modality individually, therefore it could be seen as a type of FVs dedicated to monomodal system (i.e. systems processing a single type of data). Unlike these systems, are the bimodal ones, which are capable of processing two types of information (in our case VIS and IR). The benefits are from both sides: first, the monomodal ones are not so constrained as regarded the processing time, because they can manage the data in a parallel way; still, performances are higher in the case of the bimodal systems. By referring to figure 3.3, the FS step could be performed:

    - imediately after the individual vectors have been obtained from the features extraction step, and we would have monomodal FVs to which the FS operation will be applied (they will be referred with an "i" at the power index, showing that the FS operation has been applied on VIS and IR vectors individually) or

    - after the extracted features have been combined in a bimodal FV and in this case we would have a concatenated FV (it will be denoted with an "c" at the power index, because the FS operation has been applied on the concatenated VISIR vector) on which the FS will be applied.

These individual or concatenated FVs will be introduced in this chapter and they can be reached in tables 4.3, 4.4, 4.5 and 4.6 but their specific use will be described widely in the next chapter, when we detail the proposed fusion schemes. Because these two types of FVs are needed in the fusion processings, but their purchase is very much influenced by the mode the FS scheme has been applied, we choose to describe also these bimodal FVs in this chapter.

Table 4.2: Different notations for the used FVs

Input FVs	Obtained FVs		
Application	Bimodal systems	Monomodal systems	
Without the FS step	$(VISIR)$	(VIS)	(IR)
With the FS step $(FS(VISIR))$ $(FS(VIS)FS(IR))$	$(sVISIR^c)$ $(sVISIR^i)$	$(sVIS^c)$ $(sVIS^i)$	(sIR^c) (sIR^i)

In table 4.2 it was noted (VIS) (respective (IR)) the monomodal vector containing the features extracted from the VIS image (respective from the IR one). The notation $(VISIR)$ is refering to the vector which contains both the features extracted from the VIS image and the features extracted from the IR one. When the notation $FS(VIS)$ (respective $FS(IR)$ or $FS(VISIR)$) is used, we are refering to the fact that the FS has been performed on the vector mentioned in the paranthesis. The use of the expresion $FS(VIS)FS(IR)$ is refering to the use of the FS method individually, on the VIS and IR domain. The difference between the notations $FS(VISIR)$ and $FS(VIS)FS(IR)$ is that in the first case the FS was applied on the concatenated FV $VISIR$, while in the second case the selection has been realised separately, on each individual FV of the two domains. It was denoted $(sVISIR^c)$ the bimodal vector which contains the features from both domains, VIS and IR, after the FS operation has been performed on the concatenated vector $VISIR$. By the separation of the features in monomodal vectors, the FVs denoted $(sVIS^c)$ and (sIR^c) has been obtained. In the case

the selection scheme is applied on the individual monomodal vectors, the FVs obtained could be $(sVIS^i)$ and (sIR^i), and they will be combined in the bimodal vector $(sVISIR^i)$.

For the results presented in this chapter, the selection has been realised at the level of vector *AllFeatures* coresponding to the respective modality and mentioned in table 3.2. Recall that the vector *AllFeatures* is having a dimension of 171 features in the monomodal case and 340 features for the bimodal case. If the selection has been applied on the monomodal FVs, it means the respective method has been applied on VIS and IR separately, the vectors $sVIS^i$ and sIR^i have been obtained and then, by their combination the vector $sVISIR^i$ was acquired (the VIS and IR features have been rearranged in the new FV in function of their importance). If the selection has been realised on the bimodal vector VISible and InfraRed concatenated (VISIR), then the FS method has been applied on this vector, obtaining $sVISIR^c$ and then by the identification of features belonging to the VIS or IR domains, the vectors $sVIS^c$ and sIR^c have been formed (in which the VIS and IR features are arranged by their importance).

A critical aspect of feature selection is to properly assess the quality of the selected features. This section reviews some aspects of the experimental design. The obtained FVs after the FS process will be evaluated and compared with the ones from the table 3.3. Therefore, we will consider only the classification problem with 4 classes of objects (because the results are higher than those obtained for the classification problem with 8 classes of objects).

When performing the FS by the classical methods from the literature or by the method we propose, it has to be noticed that the obtained processing time (which includes the features extraction time and the classification time) at the test stage should be decreased, as fewer features will composed the FV. Thus, besides the evaluation criteria of the accuracy, the time required to extract the features which compute the FV will be also considered.

The classifier used in this first part of the chapter, is also a kNN, with $k = 1$ for the same reason it was employed in the previous chapter: the simplicity in its usage, because it does not require a parameter optimization process (as the SVM does). In addition, the main purpose in this step was to optimize the FV, not the classifier hyper-parameters. The performance criterion of the 1-NN classifier is the balanced accuracy (bAcc) obtained on the training dataset, when a 10 folds cross-validation procedure is performed.

Next, the performances obtained by the 1-NN when using 10f-CV for the FVs obtained after the application of different FS methods will be compared. This means that first, the FS method has been applied on the data from the training set and new FVs have been obtained. Next, with the information encoded by these FVs an 1-NN classifier has been tested by a 10f-CV procedure on the same training data set.

We are interested in selecting the most pertinent FS methods (also for computing the average rank value needed for our proposed FS method) after the criteria: accuracy obtained on each modality VIS and IR has to be above a specified value for both of them in order that the respective FS method to be retained. An important question is: how we should establish the respective value which should be overcome, in order that only the best methods to be provided by this criteria? As the FS methods start from the *AllFeatures* FVs, we will compare the accuracies provided after the FS step with the accuracies obtained in the previous chapter for the vectors *AllFeatures*. If after the FS process the obtained accuracies (for both the modalities VIS and IR) will be at least as high as the ones obtained with the initial FV, the respective method will be selected.

To emphasize the best obtained accuracies, we use *italic* for the accuracy values which overcome the value 100% * Acc_{max} (where Acc_{max} is the accuracy obtained for the FV having the maximum size, i.e. the one with 171 features for the monomodal cases or the one with 340 features from the bimodal

case) obtained for the same vectors in the features extraction step.

In order to exemplify, on the columns $sVIS^i$ and $sVIS^c$ the accuracy values have been compared with the accuracy values obtained in the table 3.3, therefore with the vector comprising the maximum number of features. Where such values are reached, it means that even a FV comprise less features than the maximum-size one, it reaches a good accuracy, sometimes even better (when the value is italicized) with the help of the FS operation. Where the accuracy value is with **bold**, it means that the respective FV has provided a value overcoming 97% from Acc_{max}, thus even after the reduction of the FV dimension, the remaining features are able to provide good results.

Once we have described the input FVs, the classification problem and the evaluation criteria, we can start by presenting the obtained results. First, in order to construct the average rank value (needed by our FS method) we will evaluate the methods from the table 4.1. In the following we want to see which one provides the best results based on the criteria previously mentioned.

The methods based on the *individual selection of features* (FS1→FS6) provide only an arranged list of features, in function of the importance the criteria is asserting to each attribute, and it does not perform a real reduction of the FV as it is happening in the case of the methods which *select subsets of features* (FS7→FS12).

In the case of methods of *individual selection of features*, the reduction of the FV has to be realised by the retaining from the arranged FV of the best features, the most important ones, and by ignoring the others. For the begining, in the table 4.3 the results obtained for these FV methods (from the table 4.1) are presented. Thus, at the *individual selection of attributes* there are 6 methods FS1→FS6 in table 4.3 but they are grouped in two halves because for each method two possibilities of selecting features (denoted a) and b)) have been considered.

In each of the half it could be noticed there is a first line where on the column denoted the name of the FS method it is specified *AllFeatures* and this line is followed by the 6 methods, each comprising three values for the threshold. In fact, the 3 methods which have associated a threshold value start from the ranked list provided by the application of the respective FS method on the full training set and select features as the threshold specifies. Therefore, it should be mentioned a fourth method in each of these category of 3 methods, the one applied on the `FullTS`.

Considering that all the methods from this table use a Ranker selection method which does not select a subset of features, but it just arrange the features in the order of their importance, it could be mentioned that all 6 methods will provide the same number of features, i.e. the maximum number of features from each modality. In fact, these methods do not discard any feature, so they provide the same set of features as no FS method has been performed at all. It is not reducing the size of the feature vector, therefore the accuracy obtained with these vectors is the same as the one obtained for the vectors *AllFeatures*. This is the reason we did not specify a fourth method, but we consider a single one for all the 6 FS methods[3] at the beginning of each half from the table.

Because each of the 6 methods only provide a ranked list of features, we are expected that all the features (or at least the majority of them) to present a rank value strict positive. By the normalisation of this rank value in the domain $[0, 100]$, the selection thresholds could be chosen equally distributed in the specified domain of the rank values. Thus, in order to select only the most important attributes from the ranked list provided by the algorithms, 3 values of threshold have been fixed (the methods `FSn`$_{\text{Thr}}$, where $n \in [1; 6]$ and $Thr \in \{25, 50, 75\}$). The application of the three mentioned thresholds has been considered in two different manners:

- case a) in which the threshold values 25, 50, 75 are refering to the **percentage of attributes**

---

[3]The order in which the features are ranked by each of the 6 methods is different and it is provided by each method individually.

(denoted *nbFeat*) desired to be selected from the entire set. Therefore, for a threshold of 25, only the most important features from the ranked list will be selected, until their number is reaching a 25% from the entire feature set. The dimension of the entire feature set could be 171 or 340 (is depending if the method has been applied on the individual or concatenated FVs) and the selected features in the case of the threshold at the value of 25, a number of 43 or 85 features will be selected (first half of the table).

- case b) in which the threshold values are refering to a **rank value** (denoted *rankThr*) which has to be overcome in order that a feature to be selected. For example, the method using the threshold at the value of 75 will select a smaller number of features than the method using a threshold of 50, and this latter will select a number of features even smaller than the one employing a threshold of 25. In this last case, all the features having a rank value higher than 25 will be selected (second half of the table).

The first indicator from the table 4.3 (the one preceeded by the sign →) is showing the dimension of each vector after the application of the respective features selection method. The second indicator (the one after the sign →) is showing the balanced accuracy obtained with the 1-NN classifier when using the 10f-CV procedure applied on the training set. The data from the training set has been characterized only by the features retained after the application of the respective FS operation.

For the Ranker methods ($FS1 \rightarrow FS6$) the best results are obtained for the FS methods with a threshold value of 75 (from the first half of the table) and for the methods from the second half of the table having a threshold of 25. Still, it is important to notice that the values Acc_{max} are sometimes overcome by these FVs from which some features have been rejected, which is suggesting that there are features contradicting others and these FS algorithms have succeeded in their elimination from the final FV. Good recognition rates (above 97%* Acc_{max}) are also obtained for FVs with fewer features, even for the methods which have selected a number of almost 1/2 or 3/4 from the entire number of features. We can exemplify with the FVs from the individual selection: *FS1a_75*, *FS2a_50*, *FS2a_75*, *FS3a_50* and so on, or from the concatenated selection: *FS1a_50*, *FS1a_75*, *FS2a_75*, *FS3a_75* and others. All these FS methods provide good results, even if not all of them meet the criteria to select the FS method, criteria outlined earlier: to overcome Acc_{max} on both modalities VIS and IR. However, they give results close to those obtained with the whole set of features. For example, the method *FS4a_50* with only half features obtains all results above 97%*Acc_{max}, and it even overcome the value Acc_{max} for the vector obtained on the IR domain.

From the table 4.3 it could be noticed there are FVs which has succeeded in overcoming the value Acc_{max}, and they also meet our selection criteria, i.e. to fulfill the requirement that this should happened on both modalities in order that the respective FS method to be retained. In this manner, few FS methods have been selected: for the individual selection: *FS2a_75* (it succeed to overcome even all the 3 accuracies from the VIS, IR and VISIR domains) and *FS6a_75*, while for the concatenated selection we have: *FS2a_75*, *FS5a_75*, *FS6a_75* and *FS5b_25* (they are also with dark gray value on the table 4.3).

For the selected FS methods, in table 4.4 we showed the percentage variation for the accuracy and the dimension of the FV. Thus, almost all selected methods have a number of features of 75% on each modality in the individual selection case. The improvements from the accuracy point of view are higher on the VIS than on the IR modality (above 3.5% compared to not much above 1%). On the VISIR FV there is a smaller improvements for the case *FS2a_75* or even a negative one for *FS6a_75*. As concerning the FS methods obtained when using the concatenated selection, there is a single method (*FS5b_25*) selecting almost all features (96% from VIS, 99% from IR and 98% from VISIR) and still it does not provide very much improved accuracies. In the following, we will not consider this FS method as being selected, because neither it provides good accuracies (compared with the rest of the selected methods), nor the number of retained features is small. The other three methods are much better because with almost 75% of the initial data they give accuracies higher with 1.5% on VIS.

Table 4.3: Accuracy obtained with the features selected by the Ranker methods (FS1→FS6) using a 10f-CV procedure applied on the training set of data and using a 1-NN classifier

Features selection methods		No. of selected features → bAcc obtained with 1-NN and 10f-CV for the resulted FV containing the selected features					
FS method	threshold	$sVIS^i$	sIR^i	$sVISIR^i$	$sVIS^c$	sIR^c	$sVISIR^c$
AllFeatures	**nbFeat**	**171 → 83.7**	**171 → 88.0**	**340 → 94.4**	**171 → 83.7**	**171 → 88.0**	**340 → 94.4**
FS1a_25	≥ 25	43 → 79.9	43 → 82.5	85 → 89.0	30 → 74.7	56 → 84.5	85 → 88.0
FS1a_50	≥ 50	86 → 81.0	86 → 85.0	170 → 92.5	81 → 81.7	90 → 85.9	170 → 92.0
FS1a_75	≥ 75	128 → 86.4	128 → 87.3	255 → 94.0	121 → 84.8	135 → 87.9	255 → 91.9
FS2a_25	≥ 25	43 → 81.0	43 → 82.2	85 → 88.9	30 → 79.5	56 → 84.7	85 → 87.1
FS2a_50	≥ 50	86 → 82.8	86 → 85.4	170 → 92.5	80 → 81.9	91 → 85.9	170 → 91.4
FS2a_75	≥ 75	128 → 86.6	128 → 88.6	255 → 94.7	124 → 85.2	132 → 88.3	255 → 92.2
FS3a_25	≥ 25	43 → 81.1	43 → 83.1	85 → 91.8	35 → 82.3	50 → 84.2	85 → 91.3
FS3a_50	≥ 50	86 → 82.2	86 → 86.4	170 → 92.7	88 → 82.2	83 → 86.3	170 → 90.6
FS3a_75	≥ 75	128 → 85.0	128 → 85.5	255 → 93.9	122 → 82.8	134 → 88.4	255 → 92.8
FS4a_25	≥ 25	43 → 80.7	43 → 86.5	85 → 89.7	36 → 76.2	49 → 86.8	85 → 90.4
FS4a_50	≥ 50	86 → 82.1	86 → 89.9	170 → 92.8	69 → 81.0	101 → 86.7	170 → 93.3
FS4a_75	≥ 75	128 → 83.1	128 → 86.3	255 → 94.7	113 → 82.6	143 → 87.4	255 → 93.8
FS5a_25	≥ 25	43 → 79.4	43 → 81.7	85 → 88.9	38 → 79.0	48 → 82.6	85 → 89.5
FS5a_50	≥ 50	86 → 83.0	86 → 87.4	170 → 92.2	87 → 82.3	84 → 86.3	170 → 91.5
FS5a_75	≥ 75	128 → 85.3	128 → 87.9	255 → 93.0	124 → 85.0	132 → 88.2	255 → 92.6
FS6a_25	≥ 25	43 → 79.6	43 → 82.3	85 → 87.3	31 → 80.0	55 → 84.2	85 → 87.9
FS6a_50	≥ 50	86 → 82.7	86 → 86.1	170 → 92.6	84 → 82.3	87 → 86.9	170 → 92.4
FS6a_75	≥ 75	128 → 87.0	128 → 88.9	255 → 94.0	122 → 85.0	134 → 89.3	255 → 94.1
AllFeatures	**rankThr**	**171 → 83.7**	**171 → 88.0**	**340 → 94.4**	**171 → 83.7**	**171 → 88.0**	**340 → 94.4**
FS1b_25	≥ 25	110 → 85.0	77 → 86.3	186 → 92.0	64 → 80.5	77 → 86.2	140 → 92.3
FS1b_50	≥ 50	41 → 80.5	33 → 82.7	74 → 88.7	5 → 48.9	33 → 82.7	38 → 83.6
FS1b_75	≥ 75	11 → 64.5	9 → 75.2	20 → 81.5	0 → 0.0	9 → 75.2	9 → 75.2
FS2b_25	≥ 25	115 → 84.2	71 → 87.6	185 → 93.0	60 → 80.6	71 → 87.6	130 → 91.3
FS2b_50	≥ 50	46 → 80.8	33 → 82.7	79 → 87.6	2 → 50.1	33 → 82.7	35 → 85.0
FS2b_75	≥ 75	9 → 62.3	8 → 74.0	17 → 80.9	0 → 0.0	8 → 74.0	8 → 74.0
FS3b_25	≥ 25	127 → 85.0	113 → 86.4	239 → 93.4	104 → 83.8	113 → 86.4	216 → 91.7
FS3b_50	≥ 50	54 → 81.4	45 → 82.7	99 → 91.0	25 → 73.6	45 → 82.7	70 → 89.3
FS3b_75	≥ 75	9 → 68.5	17 → 77.3	26 → 84.5	1 → 31.6	17 → 77.3	18 → 76.9
FS4b_25	≥ 25	129 → 83.0	128 → 86.3	255 → 94.7	88 → 81.9	121 → 87.0	209 → 93.7
FS4b_50	≥ 50	46 → 81.6	31 → 84.0	77 → 88.7	12 → 65.8	22 → 83.1	34 → 86.5
FS4b_75	≥ 75	11 → 61.7	5 → 76.2	16 → 79.0	0 → 0.0	6 → 73.8	6 → 73.8
FS5b_25	≥ 25	167 → 83.6	169 → 88.7	334 → 93.8	165 → 84.0	169 → 88.7	332 → 93.3
FS5b_50	≥ 50	118 → 85.5	86 → 87.4	203 → 93.4	93 → 82.6	86 → 86.7	178 → 91.8
FS5b_75	≥ 75	34 → 77.9	26 → 82.7	60 → 88.8	5 → 48.9	26 → 82.7	31 → 84.2
FS6b_25	≥ 25	129 → 86.3	85 → 85.5	213 → 94.2	81 → 83.4	85 → 85.5	165 → 91.2
FS6b_50	≥ 50	65 → 81.3	38 → 81.9	103 → 89.3	6 → 59.4	38 → 81.9	44 → 82.6
FS6b_75	≥ 75	16 → 66.7	13 → 77.9	29 → 81.7	0 → 0.0	13 → 77.9	13 → 77.9

Table 4.4: Percentage variation for the accuracy and size of the FVs comprising the features selected by the Ranker methods (FS1→FS6) (reported to the accuracy and size corresponding to the maximum size FV - *AllFeatures*); accuracy was obtained with 1-NN through 10f-CV

Features selection methods		FV size percentage variation → Variation of the bAcc obtained with 1-NN and 10f-CV compared to the ones of the vector *AllFeatures* for the FS methods selected in table 4.3					
FS method	threshold	$sVIS^i$	sIR^i	$sVISIR^i$	$sVIS^c$	sIR^c	$sVISIR^c$
FS2a_75	≥ 75	75% → 3.53%	75% → 0.68%	75% → 0.24%	73% → 1.76%	77% → 0.31%	75% → -2.41%
FS5a_75	≥ 75	75% → 1.91%	75% → -0.17%	75% → -1.48%	73% → 1.55%	77% → 0.28%	75% → -1.96%
FS6a_75	≥ 75	75% → 3.94%	75% → 1.05%	75% → -0.48%	71% → 1.55%	78% → 1.48%	75% → -0.34%
FS5b_25	≥ 25	98% → -0.12%	99% → 0.82%	98% → -0.69%	96% → 0.39%	99% → 0.77%	98% → -1.19%

In the case of methods *selecting a subset of features* (methods `FSn_Thr` from table 4.5, where $n \in [7; 12]$) we maintained the same threshold values, but their significance will be different. Here we have 2 posibilities to apply the original method of FS. The first possibility is to apply it exactly in the same manner as in the preceeding case (on the full training set (`FullTS`), but the number of selected attributes will be very small (last line, method FSn_{TS}, where $n \in [7; 12]$), or we could apply it by a 100 folds cross-validation (`100f-CV`) procedure. In this second posibility, to the list obtained by the application on the `FullTS`, some other possible important features will be added (the results are on the first line from the table, the method FSn_{CV}, where $n \in [7; 12]$).

Table 4.5: Accuracy obtained with the features selected by the Search methods (FS7→FS12) using a 10f-CV procedure applied on the training set of data and using a 1-NN classifier

Features selection methods		No. of selected features → bAcc obtained with 1-NN and 10f-CV for the resulted FV containing the selected features					
FS method	threshold	$sVIS^i$	sIR^i	$sVISIR^i$	$sVIS^c$	sIR^c	$sVISIR^c$
AllFeatures	nbFeat	**171 → 83.7**	**171 → 88.0**	**340 → 94.4**	**171 → 83.7**	**171 → 88.0**	**340 → 94.4**
$FS7_{CV}$	> 0	**65** → *85.1*	**62** → *88.9*	**126** → *93.3*	**51** → 82.2	**53** → 88.0	**103** → 91.8
FS7_25	≥ 25	**49** → 82.6	**43** → 85.4	**91** → 91.5	**35** → 81.0	**43** → 83.2	**77** → 91.4
FS7_50	≥ 50	**44** → 81.3	**41** → 85.2	**84** → 92.0	**31** → 80.5	**41** → 83.2	**71** → 90.8
FS7_75	≥ 75	**38** → 81.8	**39** → 83.5	**76** → 90.5	**27** → 81.0	**38** → 83.9	**64** → 89.6
$FS7_{TS}$	alg	**43** → 82.0	**39** → 84.0	**81** → 92.2	**31** → 80.5	**41** → 83.2	**71** → 90.8
$FS8_{CV}$	> 0	**47** → 80.7	**57** → 84.9	**103** → 90.1	**50** → 81.2	**35** → 84.0	**84** → 88.4
FS8_25	≥ 25	**30** → 78.2	**39** → 79.8	**68** → 87.6	**32** → 78.1	**19** → 81.5	**50** → 86.9
FS8_50	≥ 50	**27** → 78.0	**37** → 81.2	**63** → 86.5	**26** → 77.6	**19** → 81.5	**44** → 85.5
FS8_75	≥ 75	**23** → 77.7	**33** → 79.3	**55** → 86.9	**22** → 77.4	**18** → 80.5	**39** → 86.3
$FS8_{TS}$	alg	**28** → 77.7	**37** → 81.2	**64** → 87.3	**27** → 77.3	**19** → 81.5	**45** → 87.8
$FS9_{CV}$	> 0	**169** → 83.6	**171** → 88.0	**338** → 94.4	**169** → 83.6	**171** → 88.0	**338** → 94.4
FS9_25	≥ 25	**143** → *84.6*	**105** → 87.1	**247** → 92.8	**119** → *83.7*	**132** → 86.9	**250** → *94.6*
FS9_50	≥ 50	**82** → *84.2*	**78** → 87.7	**159** → 91.6	**75** → *84.6*	**91** → *89.3*	**165** → 93.5
FS9_75	≥ 75	**29** → 79.4	**41** → 83.7	**69** → 89.7	**28** → 81.4	**38** → 80.9	**65** → 90.3
$FS9_{TS}$	alg	**88** → *84.1*	**77** → *88.4*	**164** → *93.0*	**78** → 82.8	**92** → 84.3	**170** → 91.0
$FS10_{CV}$	> 0	**59** → 81.9	**66** → 84.3	**123** → 89.5	**54** → 80.8	**24** → 78.6	**76** → 87.5
FS10_25	≥ 25	**11** → 77.1	**10** → 75.2	**20** → 85.5	**6** → 67.7	**5** → 76.3	**10** → 85.6
FS10_50	≥ 50	**6** → 67.3	**8** → 74.7	**14** → 81.7	**3** → 50.4	**2** → 64.6	**5** → 79.1
FS10_75	≥ 75	**5** → 65.9	**6** → 79.4	**11** → 84.0	**1** → 36.9	**2** → 64.6	**3** → 73.1
$FS10_{TS}$	alg	**8** → 71.5	**9** → 73.7	**17** → 79.9	**5** → 57.8	**4** → 73.4	**8** → 83.4
$FS11_{CV}$	> 0	**41** → 75.7	**43** → 76.9	**82** → 84.0	**41** → 75.7	**2** → 47.5	**41** → 75.7
FS11_25	≥ 25	**14** → 66.0	**17** → 67.0	**29** → 75.5	**14** → 66.0	**2** → 47.5	**14** → 66.0
FS11_50	≥ 50	**10** → 64.3	**8** → 60.4	**17** → 71.0	**10** → 64.3	**2** → 47.5	**10** → 64.3
FS11_75	≥ 75	**5** → 53.9	**3** → 48.1	**7** → 59.0	**5** → 53.9	**2** → 47.5	**5** → 53.9
$FS11_{TS}$	alg	**11** → 63.4	**12** → 65.5	**22** → 73.6	**11** → 63.4	**2** → 47.5	**11** → 63.4
$FS12_{CV}$	> 0	**23** → 77.3	**38** → 79.5	**61** → 90.1	**25** → 77.8	**29** → 80.6	**54** → 87.8
FS12_25	≥ 25	**22** → 76.3	**22** → 77.4	**44** → 86.5	**25** → 77.8	**29** → 80.6	**54** → 87.8
FS12_50	≥ 50	**22** → 76.3	**22** → 77.4	**44** → 86.5	**16** → 71.0	**22** → 76.8	**38** → 85.4
FS12_75	≥ 75	**22** → 76.3	**22** → 77.4	**44** → 86.5	**1** → 38.0	**1** → 43.3	**2** → 44.5
$FS12_{TS}$	alg	**22** → 76.3	**22** → 77.4	**44** → 86.5	**16** → 71.0	**22** → 76.8	**38** → 85.4

Table 4.6: Percentage variation for the accuracy and size of the FVs comprising the features selected by the Search methods (FS7→FS12) (reported to the accuracy and size corresponding to the maximum size FV - *AllFeatures*); accuracy was obtained with 1-NN through 10f-CV

Features selection methods		FV size percentage variation → Variation of the bAcc obtained with 1-NN and 10f-CV compared to the ones of the vector *AllFeatures* for the FS methods selected in table 4.5					
FS method	threshold	$sVIS^i$	sIR^i	$sVISIR^i$	$sVIS^c$	sIR^c	$sVISIR^c$
$FS7_{CV}$	> 0	**38%** → *1.73%*	**36%** → *1.08%*	**37%** → -1.16%	**30%** → -1.73%	**31%** → -0.06%	**30%** → -2.83%
FS9_50	≥ 50	**48%** → *0.66%*	**46%** → -0.37%	**47%** → -2.99%	**44%** → *1.11%*	**53%** → *1.53%*	**49%** → -0.98%
$FS9_{TS}$	alg	**51%** → *0.48%*	**45%** → *0.40%*	**48%** → -1.51%	**46%** → -0.99%	**54%** → -4.20%	**50%** → -3.63%

Using `100f-CV`, in the same manner as in the case of Ranker we could obtain some rank values or indexes of relevance for the features. The rank values will be obtained as being the number of folders (from a total number of 100) in which a certain feature has been chosen as being relevant (it was selected in the winner subset). The selected attributes using cross-validation are more than (and includes) the ones which should be selected by the application of the method on the all training set (the case FSn_{TS}). Due to the fact that in the cases FSn_{Thr} from the Ranker methods (table 4.3) the rank values could have discrete values in the range $[0, 100]$ with a step of 1, we have chosen the same three thresholds, equaly distributed on this interval: 25, 50, 75.

The same criteria for the selection of best FS methods was applied as in the case of the Ranker methods: the accuracy from both modalities VIS and IR should be equal or greater than the ones obtained with the FVs *AllFeatures*. In this case of Search methods, fewer FS methods have been selected by the previous mentioned criteria: two FS methods for the individual selection ($FS7_{CV}$ and $FS9_{TS}$) and one for the concatenated case: (*FS9_50*). For these selected FS methods, the percentage variation of the FV size and the obtained accuracy compared to the ones of the initial FVs *AllFeatures* are provided in table 4.6.

From this table, compared to the one corresponding to the Ranker methods (table 4.4) it can be noticed that the Search methods are able to accomplish the criteria based on which a FS method is selected with a much smaller number of features than their Ranker counterparts. The size of the FVs is reduced with at least 25% in these cases and these FVs are still able to provide accuracies which overcome the ones obtained with the vectors VIS_{171} and IR_{171}. It has to be noticed the dramatic reduction of the FV size in the case of the method $FS7_{CV}$, which with only 38% respective 36% features on VIS, respectively IR domains provides accuracies higher with at least 1% than those rendered by the vectors *AllFeatures*.

In the figure 4.1, the features' rank values for the vectors obtained when applying the FS method on individual vectors are represented on top of the figure, while the ones obtained when the FS method was applied on the concatenated vector VISIR are on bottom. In the first half (of each of the two representation) being the VIS features and in the last half the IR ones. As concerning the total number of selected methods for the individual selection, a number of 4 FS methods were retained, from which the first 2 are with the Ranker method (*FS2a_75* and *FS6a_75*), and the last two are with the Search methods ($FS7_{CV}$ and $FS9_{TS}$). On the concatenated selection, a number of 3 Ranker methods (*FS2a_75*, *FS5a_75* and *FS6a_75*) and one Search method (*FS9_50*) were selected.

It can be noticed from all the selected FS methods from figure 4.1 that in the case of the Ranker methods, they are using thresholds reported to the number of features (the case a)). Thus, this modality to retain features based on a percent relative to the total number of features is better than the one which retain features considering a threshold relative to their rank value (case b)). This latter type of Ranker method has been selected in a first step (*FS5b_25*) but it was rejected because of the great number of selected features. Also, it could be noticed that for the Search methods, both types of application have been retained: when the FS method was applied once on the full training set (one case from 3), and when the `100f-CV` method was used (2 times, one with the entire set selected - CV and one with the threshold 50).

Also, it could be observed that from all the 5 Ranker methods selected, 2 cases are for the Information Gain single-attribute evaluator, other 2 cases are for the Symmetrical Uncertainty and one is for Significance. From all the 6 possible single-attribute evaluators from table 4.1, these are the ones which provided good results for our classification problem.

For the Search methods, all the 3 methods selected are based on the correlation-based attribute subset evaluator and as a search method, for 2 cases it was a Genetic Search and for one case a Best First search.
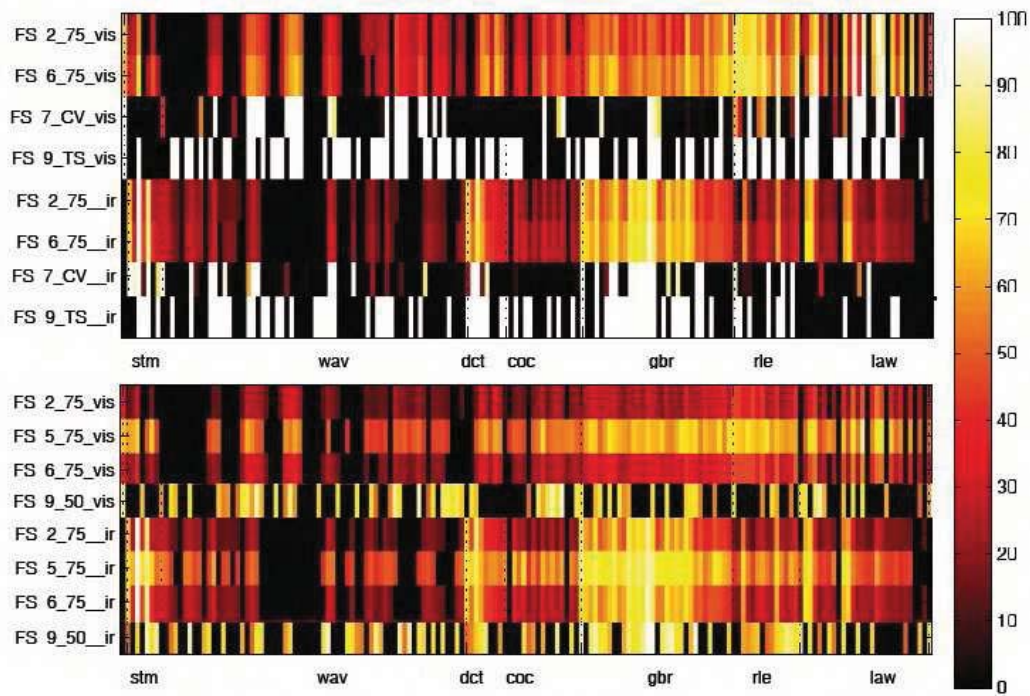
Figure 4.1: Rank scores for the retained FS methods: individual (top) and concatenated (bottom)

Another remark considering the figure 4.1 is that the features selected by the Ranker methods ($FS1 \rightarrow FS6$) are presenting more colors. This is due to the fact that there are much many features selected and they are having different rank values. For the Search methods, one can notice that the one which does not use thresholds and perform the selection of features on the full training set (i.e. the one from the individual application, $FS9_{TS}$) presents only white/black zones. This is due to the fact that here features are retained or not, i.e. they have a corresponding rank value either 100 or 0. In addition, also for the Search methods but for those performing the selection based on the cross-validation procedure (even if they use or not thresholds), the colors are also improved, as in the case of Ranker methods. For all the the selected FS methods (with the exception of $FS9_{TS}$) discretized rank values in the range [0; 100] with the step 1 exist. The features from these selected FVs will have a higher or a lower rank value, as the FS method provided for them, but there will be features having a rank value 0, and this will be happened for the ones that were discarded from these FVs.

Also, it could be observed that the features selected on the VIS domain are differing from the ones selected on the IR domain. In both cases, individual (figure from top) and concatenated (bottom) it can be observed that on the IR domain, the areas corresponding to *gbr* family are the most selected ones. Like this, on the VIS domain, such zone highlighted by the strong selection of features could be the one corresponding to the *rle* family. It could be noticed that here the first features from the *rle* family seems to be selected by the FS methods, for the individual case (figure from top), and in the case of the concatenated one (bottom), *rle* and *gbr* families present such highlighted areas.

In table 4.7 the first features from each of the selected methods from figure 4.1 are illustrated. For a better visualization of the importance of each features, it is suggested to follow figures 4.1. Here all the selected features can be found, not only the first ones.

In table 4.7 only the first few of the selected features (presented in figure 4.1) are mentioned. There is only one selected FS method which was obtained by the application of the respective FS procedure on the full training set at once and this is $FS9_{TS}$. The difference from all the rest of the selected methods comprised in this table is that the $FS9_{TS}$ is the only one which does not contain any information about the importance or relevance (i.e. the rank value) of the selected features in the retained set. Therefore, for the FS method $FS9_{TS}$ the features are provided ordered by the family and the index inside the respective family. It could be noticed that besides the first feature (wid_{i01}) all the rest are from the Haar wavelet family from the VIS domain, ordered by an increased index. For all the rest of the methods presented in this table, the features are provided ordered by their importance, which is specified by their rank value. For example, the first method selected, from the individual set, $FS2a\_75$ presents as the most relevant feature the $laws_{v17}$ and if we search this feature on the corresponding representation from the figure 4.1, it can be noticed it has the most highlighted area from all the features of the respective vector. Other features from the laws family from the VIS domain are mentioned in the table, i.e. $laws_v$ with the index 04,10,12,14,16,18,20,22 and this is also verified with multiple highlighted areas (i.e. having high rank values) for the laws family from the VIS domain on the figure 4.1.

Table 4.7: The first features selected for the retained FS methods

FS_{method}	No. of att	First features from the VISIR vectors obtained by the application of the FS method on the individual VIS and IR vectors
FS2a_75	**255**	$[laws_{v17}, stm_{i03}, laws_{v12}, laws_{v14}, gab_{i15}, dct_{i03}, stm_{i05}, rle_{v01}, laws_{v18}, gab_{i01}, gab_{i11}, rle_{v05}, rle_{v02}, laws_{v04}, rle_{v07}, gab_{i08}, gab_{i16}, rle_{v14}, rle_{v10}, laws_{v10}, gab_{i13}, gab_{i23}, gab_{i04}, gab_{i19}, gab_{v31}, rle_{v04}, dct_{i01}, gab_{i14}, laws_{i03}, laws_{v20}, wid_{i01}, rle_{v03}, stm_{i01}, gab_{i21}, gab_{i12}, gab_{v23}, laws_{v16}, \ldots]$
FS6a_75	**255**	$[laws_{v14}, stm_{i03}, laws_{v12}, laws_{v04}, laws_{v18}, gab_{i15}, laws_{v17}, stm_{i05}, rle_{v14}, rle_{v10}, rle_{v07}, gab_{i16}, rle_{v01}, gab_{i11}, rle_{v04}, rle_{v03}, laws_{v03}, dct_{i03}, rle_{v05}, rle_{v02}, gab_{i13}, laws_{v10}, gab_{i01}, dct_{i01}, gab_{i04}, gab_{i08}, laws_{v20}, gab_{i14}, gab_{i12}, laws_{i03}, gab_{v23}, stm_{i01}, gab_{v31}, gab_{i23}, stm_{v03}, laws_{i02}, gab_{i19}, \ldots]$
FS7$_{CV}$	**126**	$[wid_{i01}, haar_{v19}, haar_{v20}, haar_{v21}, haar_{v22}, haar_{v27}, haar_{v28}, haar_{v29}, haar_{v30}, haar_{v36}, haar_{v48}, haar_{v50}, haar_{v51}, haar_{v52}, haar_{v55}, haar_{v56}, haar_{v58}, haar_{v60}, gab_{v15}, gab_{v30}, rle_{v04}, rle_{v14}, laws_{v03}, laws_{v04}, laws_{v12}, laws_{v13}, laws_{v14}, laws_{v17}, laws_{v18}, haar_{i01}, haar_{i11}, haar_{i12}, haar_{i20}, \ldots]$
FS9$_{TS}$	**164**	$[wid_{i01}, haar_{v03}, haar_{v04}, haar_{v06}, haar_{v07}, haar_{v09}, haar_{v11}, haar_{v14}, haar_{v15}, haar_{v19}, haar_{v20}, haar_{v22}, haar_{v24}, haar_{v25}, haar_{v26}, haar_{v28}, haar_{v29}, haar_{v30}, haar_{v38}, haar_{v39}, haar_{v40}, haar_{v42}, haar_{v45}, haar_{v46}, haar_{v47}, haar_{v48}, haar_{v50}, haar_{v51}, haar_{v52}, haar_{v56}, haar_{v57}, haar_{v59}, haar_{v60}, \ldots]$

FS_{method}	No. of att	First features from the VISIR vectors obtained by the application of the FS method on the concatenated VISIR vector
FS2a_75	**255**	$[stm_{i03}, gab_{i15}, dct_{i03}, stm_{i05}, gab_{i01}, gab_{i11}, gab_{i08}, gab_{i16}, gab_{i13}, gab_{i23}, gab_{i04}, gab_{i19}, dct_{i01}, gab_{i14}, laws_{i03}, stm_{i01}, gab_{i21}, gab_{i12}, laws_{i02}, laws_{i11}, gab_{i07}, gab_{i24}, rle_{i10}, dct_{i04}, gab_{i22}, gab_{i20}, laws_{i10}, laws_{v17}, gab_{i03}, rle_{i03}, gab_{i06}, gab_{i09}, gab_{i18}, laws_{v12}, laws_{v14}, gab_{i05}, gab_{i29}, gab_{i26}, \ldots]$
FS5a_75	**255**	$[stm_{i03}, gab_{i15}, gab_{i01}, gab_{i16}, stm_{i05}, laws_{v14}, gab_{i11}, gab_{i04}, gab_{i13}, dct_{i03}, gab_{i08}, dct_{i01}, gab_{i14}, laws_{i11}, laws_{v12}, laws_{i03}, gab_{i10}, gab_{i02}, laws_{v18}, gab_{i12}, gab_{i07}, gab_{i24}, stm_{i01}, laws_{i02}, laws_{v17}, gab_{i06}, laws_{v04}, laws_{i10}, gab_{i19}, gab_{i23}, rle_{i10}, gab_{v32}, laws_{v03}, rle_{i03}, gab_{i17}, rle_{v04}, gab_{i20}, \ldots]$
FS6a_75	**255**	$[stm_{i03}, gab_{i15}, stm_{i05}, gab_{i16}, gab_{i11}, dct_{i03}, gab_{i13}, gab_{i01}, dct_{i01}, gab_{i04}, gab_{i08}, gab_{i14}, gab_{i12}, laws_{i03}, stm_{i01}, gab_{i23}, laws_{i02}, gab_{i19}, laws_{i11}, gab_{i21}, gab_{i07}, gab_{i24}, laws_{i10}, gab_{i20}, rle_{i10}, gab_{i10}, dct_{i04}, gab_{i22}, rle_{i03}, gab_{i03}, gab_{i06}, gab_{i09}, laws_{v14}, gab_{i18}, laws_{v12}, gab_{i17}, laws_{v04}, laws_{v18}, \ldots]$
FS9_50	**165**	$[gab_{i16}, gab_{i15}, gab_{i25}, coc_{v13}, stm_{i05}, gab_{i20}, haar_{i36}, haar_{v55}, coc_{v11}, haar_{v21}, laws_{v05}, gab_{i01}, gab_{i04}, haar_{v50}, haar_{i07}, gab_{i19}, haar_{i46}, haar_{i51}, stm_{i07}, haar_{v28}, haar_{i53}, gab_{i11}, gab_{i14}, rle_{i12}, coc_{v08}, dct_{i05}, rle_{i13}, haar_{v19}, haar_{v60}, coc_{i12}, gab_{i03}, haar_{v61}, haar_{v63}, laws_{v04}, haar_{i55}, gab_{i18}, coc_{v16}, gab_{v09}, \ldots]$

In table 4.8 and figure 4.2 the selection percentages on each family of features in function of the retained FS method are represented. At the top of each the table and the figure are the values obtained for the retained FVs after the application of the respective FS method on the individual vectors from the VIS and IR domains, while on the bottom side of each representation are the values corresponding to the application of the respective FS method on the concatenated vector VISIR.

Table 4.8: Selection-percentage on each family of features for retained FS methods
applied on the individual (top) and concatenated (bottom) FVs

FS method	No. of features selected			Selection-percentage on each family [% from the number of features per each family]							
	VIS	IR	VISIR	2wh	$7stm_v$	$64haar_v$	$8dct_v$	$16cooc_v$	$32gbr_v$	$14rle_v$	$28laws_v$
$FS2a_{75}$	**128**	128	255	50.0	85.7	71.9	75.0	87.5	100.0	100.0	75.0
$FS6a_{75}$	**128**	128	255	50.0	100.0	71.9	75.0	93.8	100.0	100.0	75.0
$FS7_{CV}$	**65**	62	126	50.0	28.6	35.9	0.0	25.0	43.8	64.3	42.9
$FS9_{TS}$	**88**	77	164	50.0	71.4	51.6	50.0	43.8	46.9	64.3	50.0

FS method	No. of features selected			Selection-percentage on each family [% from the number of features per each family]							
	VIS	**IR**	VISIR		$7stm_i$	$64haar_i$	$8dct_i$	$16cooc_i$	$32gbr_i$	$14rle_i$	$28laws_i$
$FS2a_{75}$	128	**128**	255	-	28.6	46.9	100.0	75.0	100.0	85.7	67.9
$FS6a_{75}$	128	**128**	255	-	28.6	43.8	100.0	75.0	100.0	85.7	67.9
$FS7_{CV}$	65	**62**	126	-	14.3	40.6	62.5	18.8	50.0	42.9	14.3
$FS9_{TS}$	88	**77**	164	-	14.3	46.9	75.0	43.8	65.6	50.0	14.3

FS method	No. of features selected			Selection-percentage on each family [% from the number of features per each family]							
	VIS	IR	VISIR	2wh	$7stm_v$	$64haar_v$	$8dct_v$	$16cooc_v$	$32gbr_v$	$14rle_v$	$28laws_v$
$FS2a_{75}$	**124**	132	255	50.0	71.4	51.6	75.0	81.3	100.0	100.0	71.4
$FS5a_{75}$	**124**	132	255	50.0	57.1	50.0	75.0	87.5	100.0	100.0	75.0
$FS6a_{75}$	**122**	134	255	50.0	42.9	51.6	75.0	81.3	100.0	100.0	71.4
$FS9_{50}$	**75**	91	165	50.0	85.7	43.8	25.0	50.0	43.8	35.7	39.3

FS method	No. of features selected			Selection-percentage on each family [% from the number of features per each family]							
	VIS	**IR**	VISIR		$7stm_i$	$64haar_i$	$8dct_i$	$16cooc_i$	$32gbr_i$	$14rle_i$	$28laws_i$
$FS2a_{75}$	124	**132**	255	-	57.1	56.3	100.0	87.5	100.0	100.0	82.1
$FS5a_{75}$	124	**132**	255	-	42.9	57.8	100.0	87.5	100.0	100.0	82.1
$FS6a_{75}$	122	**134**	255	-	71.4	56.3	100.0	87.5	100.0	100.0	85.7
$FS9_{50}$	75	**91**	165	-	42.9	48.4	75.0	37.5	68.8	78.6	39.3

The values specified in the table 4.8 show which is the percentage a specific family of features has been selected by a FS method. Each value from the table is showing how many features corresponding to the respective family have been selected as being relevant by the respective method on the VIS or IR domains. For example, in the case of the first FS method from the individual application, *FS2a_75* for the statistical moments family on the IR domain it is specified a value of 28.6, which means that from the number of 7 stm_i features, it was selected a number of $(28.6*7)/100 = 2$ features. In the same manner, one can notice that in the case of the first two selection methods from the individual application, all the features from the *gbr* and *rle* families have been retained in the VIS domain, and all the *stm*, *gbr* and *rle* features have been selected in the IR domain. Also, it has to be mentioned that for the third selected method from the individual application, $FS7_{CV}$, the *dct* family have been entirely discarded (i.e. no *dct* feature have been selected as being relevant) on the VIS domain, even other FS methods, i.e those from the concatenated application, selected all the features from this family.

All the values mentioned in table 4.8 could be followed on the figure 4.2, where each combination between a FS method and a family of features is represented by a color showing how relevant that family was assigned to be for the respective FS method. The lighter the color, the most relevant the entire family of features, i.e. more features have been selected as being relevant from that family.

In the previous chapter, each family of features has been evaluated by the use of a kNN classifier and it has been noticed that some families could be more relevant than others (for example wavelet families compared to geometrical features). In this case, the analysis is much more detailed, because here each feature could be evaluated by the use of the FS methods. In this way, the relevance of the features-families could be evaluated by a degree, i.e. the selection-percentage from table 4.8, or even more, we could see exactly which features are more often selected by the proposed FS methods (table 4.7).
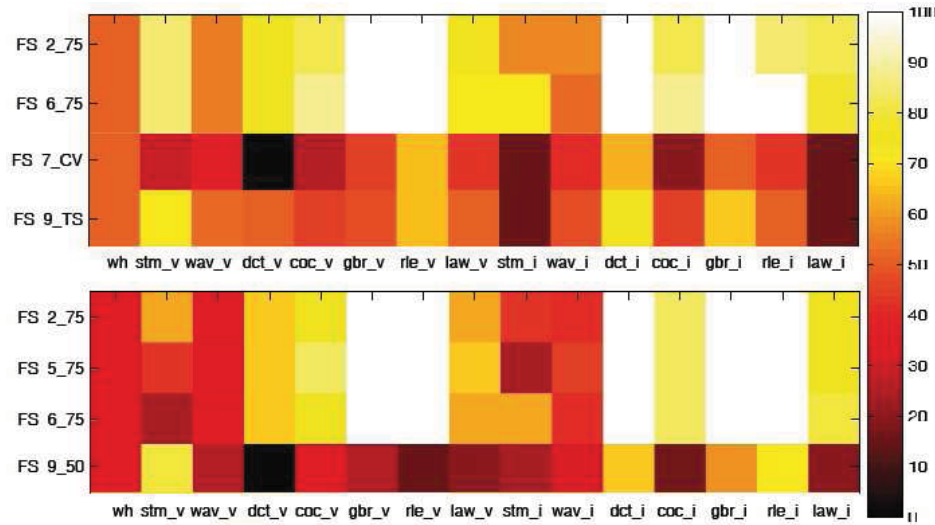
Figure 4.2: Selection percentages of all families of features: individual (top) and concatenated (bottom)

From the total number of selected FS methods for the individual case, where 2 methods were for the Ranker (*FS2a_75* and *FS6a_75*), and 2 for the Search (*FS7$_{CV}$* and *FS9$_{TS}$*) and from the ones with the concatenated selection (3 Ranker methods *FS2a_75*, *FS5a_75* and *FS6a_75* and one Search *FS9_50*), we choose to select only methods applied on the individual selection in order to take into account the most important information from both domains, separately. In the case of the individual application of the FS method, the differences between the accuracies obtained on VIS and the accuracies obtained on IR domains is smaller than the corresponding ones obtained on a concatenated application of the FS method. In addition, the accuracies obtained on the VISIR domain, thus for the bimodal vector comprising both sets of features, for both types of application (individual and concatenated) is almost the same.

Thus, from the methods obtained by the application of the FS procedure on the individual vectors, only 2 FS methods have been selected in the following in order to perform the comparation with the initial feature vector *AllFeatures*. We choose one with the best accuracies (i.e. the one which overcome both the modalities but also the VISIR domain) and one with the best number of selected features (i.e. the smallest number of features selected on each modality). Thus, *FS2a_75* from the Ranker and *FS7$_{CV}$* from the Search, both for the individual selection have been considered. The number of features of these FVs is VIS_{128} and IR_{128} and VIS_{65} and IR_{62}.

In table 4.9 balanced obtained for the FVs provided by different feature selection schemes are presented and compared with the ones obtained for the vectors AllFeatures.

By now, in this chapter, the selected FVs have been evaluated by the use of the kNN classifier with $k = 1$. This has been chosen due to the necessity of evaluating the relevance of the features, not the performance of the classifier. Once the FVs have been established, the next step could be performed and this is represented by the choice of the proper SVM classifier on these computed FVs.

Like in the previous chapter, we consider the vectors to comprise only monomodal features, i.e. either VIS or IR features will compute a FV. On these monomodal FVs, the SVM hyper-parameters should also be optimized like in the case of the initial FVs (those which have a maximum size of 171

features) denoted *AllFeatures* (i.e. VIS_{171} and IR_{171}).

For the optimization of the SVM hyperparameters, the same procedure as in the previous case will be followed: for all the hyper-parameters combinations, the maximum accuracy is searched for when the training set and the 10f-CV procedure are used. The difference, when comparing with the processing from the previous chapter, is that here the data are characterized only by the features retained as being relevant, not by all 171 features from one modality. The best bAcc will provide the winner SK, i.e. the optimized one (SK*) with the corresponding parameters: ($SKtype,C,SKparameter$).

Table 4.9: SK optimization based on accuracies provided by different FVs
obtained before or after the FS step
for the classification problem with 4 classes of objects

Method				Classification problem		Performance	
		Pb.		Input vector	bAcc [%]	winner SK $SK^*(SKtype,C,SKparam.)$	
Train	10f-CV	4 classes	*AllFeatures*	VIS_{171}	90.1	$(RBF,2^7,2^{-15})$	
				IR_{171}	92.2	$(RBF,2^9,2^{-19})$	
Test	LT			VIS_{171}	94.7	-	
				IR_{171}	94.9	-	
Train	10f-CV	4 classes	$FS7_{CV}$	VIS_{65}	91.3	$(POL,2^{-1},1)$	
				IR_{62}	90.4	$(RBF,2^9,2^{-17})$	
Test	LT			VIS_{65}	96.2	-	
				IR_{62}	95.2	-	
Train	10f-CV	4 classes	$FS2a\_75$	VIS_{128}	89.3	$(RBF,2^{11},2^{-19})$	
				IR_{128}	92.4	$(RBF,2^9,2^{-19})$	
Test	LT			VIS_{128}	95.6	-	
				IR_{128}	95.5	-	

As it was previously mentioned, two different FS methods have been chosen for the application of the FS methods (a Ranker one *FS2a_75* and a Search one $FS7_{CV}$, coresponding to those FS methods which provided the best accuracies or the best nomber of features). On each of these FVs the optimization process has been realised, in the same manner as in the case of the FVs obtained without the application of any FS methods, which were optimised in the previous chapter.

As it can be noticed when comparing table 3.6 with table 4.9 the values of the bAcc obtained for the FVs after the FS operation have been applied are better than the ones provided by the initial FVs VIS_{171} and IR_{171}. In addition, the number of the features retained by the FVs obtained after the application of the FS methods is smaller. Thus, besides the accuracy, the classification time will be also improved by the use of these vectors incorporating only the most relevant features.

Even the number of features provided by the FS method *FS2a_75* is higher than that of the method $FS7_{CV}$, the latter provide higher accuracy rates.

Another remark is that the SKs obtained after the optimization process are not the same in the two experiments. There are some winner SKs for the classification problem with 171 features and there are other winner SKs for the classification problem with the vectors obtained after applying the FS methods.

4.5   Conclusion

In this chapter different FS methods have been tested and compared in order to evaluate the pertinence of each feature (and of each family of features) in relation to our objective of obstacle classification. The experiments we developed in this chapter tried to verify which is the FS method that gives the best solution to our problem. Besides the methodology in which these FS methods are applied in literature (i.e. on the whole training set once - the case `FullTS`), we have also proposed an original manner to apply them: to repeat the respective FS method for multiple sub-samples of the training data by including it in a cross-validation loop - the case `100f-CV`. From the search methods and their evaluators described in the first part of this chapter, a number of 66 possible FS methods have been constituted. There were 36 FS methods using a Ranker searching (we have employed 6 types of single-attribute evaluators combined with 3 possibilities to retain the ranked features and 2 ways to choose the selection thresholds) and 30 methods using a Best First, Linear Forward or Genetic Search (each of this search methods have been combined with 2 possible subset attribute evaluators and for each, 5 ways to apply and retain the features have been considered). All these 66 FS methods have been tested on two types of vectors; the first one was obtained by the application of the FS method on the individual FVs and the second one by the application of the FS method on the concatenated FV.

The main difference between the two ways to apply the FS method, on the individual vectors VIS and IR or on the concatenated one VISIR, is that in the first case the most relevant features will be considered on each domain, separately and they will be further combined together. In the case of a concatenated application of the FS method, when the complementarity of VIS and IR domains is considered, the VIS domain will be disadvantaged, although the final results obtained with the bimodal vector VISIR will be as good as the ones obtained with the bimodal vector from the individual application. Moreover, if the individual application is selected, the chosen FS method could be applied in a parallel way, on VIS and IR domains. Thus, once that each set of VIS and IR features have been extracted from the corresponding images, they could be processed separately, to gain in the computation time.

The pertinence of each vector constructed based on these FS methods was first evaluated by a simple kNN classifier. From all the FS methods evaluated by the 1-NN classifier, only two of them have been chosen for further processing: a Ranker one ($FS2a\_75$ and a Search one ($FS7_{CV}$), i.e. that FS methods providing the higher accuracies on the training set considering all the 3 domains VIS, IR and VISIR and the smallest number of features selected on each modalities. The Ranker FS method $FS2a\_75$ is using Information Gain as a single-attribute evaluator), while the Search method $FS7_{CV}$ combines Best First search with a subset evaluator based on correlation. Both selected methods were obtained on individual FVs.

In order to increase the accuracy of the classification, but also to obtain a powerful classifier, the kNN was later (after the best FS methods have been chosen on the training set) replaced by the SVM. The SVM's hyper-parameters optimization has also been realised for each FV obtained with the retained FS methods. The results demonstrated that FS methods improve the recognition rates for the monomodal systems employing the vectors previously evaluated. Also, it was noticed that the use of the SVM classifier assures improved results compared to the kNN. Still, the monomodal systems used in this chapter are not proper for our ODR system. They are all dedicated to the processing of either VIS or IR information and they could not perform the adaptation to different environmental context as we intend our ODR system does. Thus, in the next chapter we propose three fusion schemes which all could accomplish this task and hopefully will even increase once again the obstacle classification accuracy obtained by now.

<div align="right">

CHAPTER 5

Fusion

</div>

Contents

From what has been seen in the previous chapters, the ODR system we propose in a first step it extracts different features by which the obstacles' images are described. In order to see which of these features are indeed relevant for the classification process, but also to decrease the processing time required by the obstacle recognition task, we opted for the application of some features selection schemes. Although the results are promising up to this point, we choose also to apply some fusion schemes in order to verify which is the additional contribution which could be added from the accuracy point of view. Still, the major objective when integrating fusion was to assure the robustness of the ODR system to function well, even in different environmental conditions. Therefore, we want to check if it is worth to perform the fusion between VIS and IR images.

The fusion process is the main idea of this thesis. We tried to address and develop different fusion schemes which combine visible and infrared information for road obstacle categorization based on SVM classification. In this thesis we want to formally compare various fusion-based solutions before the system is implemented. This would help in choosing the best solution for a given scenario. For example, we have to decide if it is better to fuse data and then to detect/recognize obstacles with the fused data (*low-level fusion*) or to detect/recognize obstacles in each image separately and then to fuse the decisions (*high level fusion*).

Three types of fusion are detailed in this chapter: a low level feature-based fusion, an intermediate level kernel-based fusion and a high level matching-score fusion. These fusion techniques are compared for a road-obstacles SVM-based classification problem. The feature-level fusion we present in section 5.2.1 yields a feature vector integrating both visual and infrared information and this vector will be used as input to an SVM classifier, while the matching-score fusion from the section 5.2.2 combines matching scores of individual obstacle recognizers in order to improve the system's final decision. The fusion at the SVM's kernel level we present in this chapter at the section 5.3 could be considered as an intermediate level of fusion because the kernels are components of the SVM classifiers and they are applied to different vectors of information (one kernel from the VIS and another kernel from the IR domain). In order to ensure the adaptation of the system to different weather and illumination conditions, features, kernels or matching-scores could be weighted (with a sensor weighting coefficient) according to the importance of the sensor in a specific environmental situation. A comparative study of individual visual and infrared obstacle recognizers versus fusion-based systems is performed in this chapter in section 5.4, where some experiment results are given and finally, we draw the conclusion in section 5.5.

5.1   Our Fusion Schemes for an OR Component

The fusion process is the main idea of this chapter. We tried to address and develop different fusion schemes which combine visible (VIS) and infrared (IR) information for a road obstacles SVM-based classification. We concentrate on VIS and IR sensors because they seem to be a good solution compared to some other technologies (like radar, laser, etc) considering the price and the lack of interference problems. We chose these two complementary vision sensors because the system must work well even under difficult conditions, like poor illumination or bad-weather situations (dark, rain, fog). The high performance and robustness of the system will be assured by the fusion of these two types of information, weighted in such a manner to allow the adaptation of the system to the environmental conditions.

A comparative study of individual visual and infrared obstacle recognizers versus fusion-based systems is performed in this chapter. We propose a framework for fusion of features, kernels and matching-scores in an obstacle categorization system based on an SVM classification scheme. In the following, the three methods we propose for performing the fusion are detailed.

Nowadays, there are a lot of sources of information that can be considered in a complex system. The fusion process combines different type of information (like sensor inputs, data or even algorithms) together in order to provide a complementary perspective and to increase the system's performances. Sanderson et al (Sanderson & Paliwal, 2002), referring to a biometric matching system, classified the fusion types in two main classes:

   - fusion prior to matching (or early fusion, which is the low-level fusion we previously mentioned) and
   - fusion after matching (or late fusion, which stands for the high-level fusion case).

Data fusion and feature fusion belong to the first category, while from the second type we have matching-score fusion and decision level fusion.

Which one of these proposed fusion schemes will be chosen for the implementation of the final system? Criteria by which we chose the final fusion scheme are the accuracy of the recognition but also the time in which this operation can be accomplished. Also, of great importance are the required conditions for the scheme to be implemented in the entire ODR system.

Even the cameras are supposed to be mounted in a fixed frame on the vehicle, when multiple hills and holes come upon the road, it is for sure that the cameras will be no more calibrated and a new calibration process would be needed (this operation is under investigation in the frame of multiple research groups and their main purpose is to perform the calibration in an online fashion).

Once the detection step has provided the BBs in the two types of images, our proposed fusion schemes could be realised, even there is not a perfect correlation between the two types of images (even from a reason or another the cameras have been easily decalibrated). For example, if the fusion is performed at the scores level, even if the two types of cameras (VIS and IR) would suffer a slight mis-calibration, this would not affect too much the outcome of the system. If we transpose the same situation in the case of a fusion at the image pixels level, this might introduce errors because the pixels of the two images would be totally decorrelated. The lower the level at which the fusion scheme is applied, the higher the sensitivity of the system to the errors introduced by the mis-calibration of cameras. Therefore, we have not proposed a fusion scheme at the pixel level, but we focused on the higher levels: the ones of features, kernels and matching-scores. So in the case of our proposed fusion schemes, the most sensitive to the mis-calibration of cameras is the one which accomplish the fusion at the features level.

Different types of systems are defined by the information which is fused in (Jain *et al.*, 2008):

(a) *A multi-sensor vision system*, used to extract diverse information about the same object. In the frame of our application, the two-dimensional texture content of a person's body could be recorded using a VIS camera, while a second vision sensor (e.g. a far-infrared Far InfraRed (FIR) camera) could be used to measure the temperature of that person's body. Besides improving the accuracy of the system, the availability of these multi-sensor data converging to a single trait can also assist the obstacle detection process.

(b) *A multi-algorithm system* can be considered either any system which uses multiple forms to represent data (e.g. different types of features) provided by the same sensor, or any system which uses multiple processing schemes which operates on the same type of information (using the same feature sets).

A system that retains different poses of an obstacle in order to correctly learn the obstacle possibilities of appearance is (c) *a multi-sample system*. Such a system could be very useful for an obstacle recognition task because the pedestrian form is very likely to change: pedestrians may use different outliers, different clothes, they can appear in different positions, occluded or not.

Therefore, it could be said that we defined a multi-sensor vision system (because it uses both visible spectrum and infrared sensors), which is actually also a multi-algorithm system due to the fact that we extracted different information by different algorithms to characterize an object (as shown in section dedicated to the feature extraction task) and which is also a multi-sample system because it uses multiple instances of the class vehicle, pedestrian and cyclist to classify a new obstacle in one of these possible classes.

From all types of fusion we mentioned by now, we chose to use only three of them (fusion at the feature level, at the kernel level and at the matching scores level) and not to consider the extreme cases: data-fusion and decision fusion.

In (Apatean *et al.*, 2009) we have used for comparation purposes a data-level fusion, performed at the pixel level. Besides these types of methods performed at the pixel level, others based on region processing (like filtering technique or the fusion ones operating in the transformation domain, such as those based on discrete wavelet transform, cosine transform, laplacian or gradient pyramid, principal component analysis or independent component analysis) are intensively used for computing a single image by fusing two or more images. The major inconvenient with these data-level fusion techniques is that a perfect correlation between the images is required before performing the fusion itself. If the images are not well correlated, this misregistration could lead to errors in the image fusion process. In the frame of our application, possible problems in correlating images provided by cameras are inherent due to the cameras' calibration issues and the fact that the vehicle is moving.

To the classical scheme of an ODR system (presented in figure 3.1), which was followed in developing our system, we added the fusion part. Both stages (training and testing) require that the information provided as input to the entry of the classifier from each stage to be encoded in the same way. This information can take one of the following forms: a) monomodal features, i.e. the VIS and IR information corresponding to possible obstacles enclosed in BBs is comprised in two different vectors, one VIS and another one IR, and thus there is one classifier for each type of data (for monomodal systems or systems including a matching-score fusion) or b) bimodal features, where the VIS and IR information will be merged in the frame of the same vector (denoted VISIR) and thus a single classifier will be used to process this type of data (for the systems including a feature-based fusion or a kernel-based fusion). The proposed fusion schemes are conducted at three possible levels: features, SVM kernels and matching scores and they are presented in what follows.

Besides the three fusion schemes we mentioned, there is another possible situation, when no fusion scheme is applied. This is valid for monomodal systems, where no fusion, at either stage

is considered. In figure 5.1 one VIS and one IR monomodal systems are presented and as it could be remarked from this figure, each system process its own monomodal vector of information. This information is extracted from each type of image (i.e. each modality) and it will be processed individually by an SVM classifier. The class to which the test object will be assigned is decided on each modality separately.
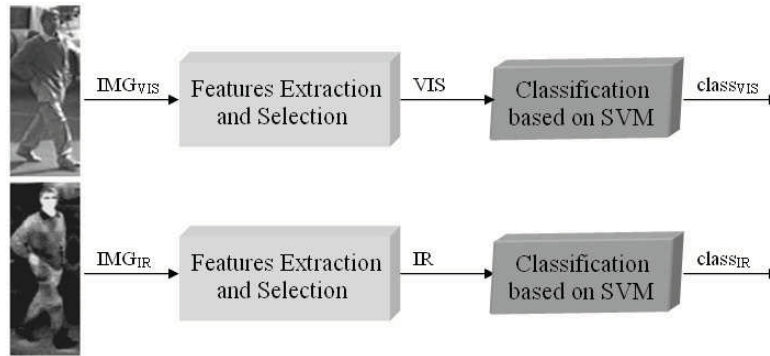

Figure 5.1: Visible and infrared monomodal systems - no fusion scheme is applied

The three fusion schemes we propose are realized at the following modules reported to the ones from figure 5.1:

- at the **Features extraction and Selection Module** the first fusion scheme we propose, i.e. fusion at the feature-level, is realized. Thus, the output of the fusion module will be a combined, i.e. fused feature vector, enclosing both VIS and IR information, which is provided as inputs to the fusion module. In this way, at the entrance of the classifier used in this fusion scheme (to evaluate the performances of the fused system), a bimodal feature vector will be provided;

- at the **Classifier Module**[1], the other two fusion schemes we propose will be realized in the following manner:

    - inside the classifier module, but at the component which compute the SVM kernel; therefore, the decision about the objects' class will be taken based on a combined kernel which process the bimodal VISIR information - this is valid for the kernel-fusion scheme we propose;

    - inside the classifier module, but after the VIS and IR information has been processed individually by the SVM kernel, therefore two kernels (inside of two different classifiers), one dedicated to the monomodal VIS vector and the other for the monomodal IR vector have to exist; the decision about the class of the object will be taken based on a combined information (i.e. a probability or a score) provided by the two SVM kernels - this is valid for the matching score fusion we propose.

The two types of vectors previously mentioned, monomodals or bimodals, could be the ones obtained immediately after the features extraction step (i.e. VIS_{171}, IR_{171} or $VISIR_{340}$) or they could be the ones obtained after also the features selection scheme has been applied (i.e. $sVIS_m$, sIR_n or $sVISIR_{m+n}$), where m is the number of features selected on the VIS domain and n is the number of features selected on the IR domain).

In order to evaluate the proposed fusion schemes, the situation of monomodal systems has also to be considered and the obtained results will be compared with the ones provided by the proposed fusion schemes at the experiments section.

---

[1]We have considered that in the frame of the SVM classifier also the decision of the class to which the test object is assigned is taken.

5.2 Low and high level fusion

In this section we treat two types of fusion: the fusion at a low level (i.e. feature fusion) and the fusion at the high level (i.e. matching score fusion). We choose to discuss them together because they have been already addressed in literature in the field of the VIS and IR images. To our knowledge, the fusion scheme applied to the level of the SVM kernels in VIS and IR images has not been addressed so far in literature, therefore it will be discussed separately in the next section. However, from both viewpoints (the data from the input of the fusion schemes and the method used to optimize the SVM model in the training stage of the system) these two fusion schemes are different.

In the case of the features-fusion, where the fusion takes place at an early stage, i.e. the data is already combined when it reaches the classifier, bimodal vectors (obtained by the combination of VIS and IR information) will be employed.

5.2.1 Feature fusion

In order to compute the bimodal (i.e. fused) feature vector $VISIR_{340}$, denoted *AllFeatures* in table 3.2, which will be the input to a bimodal system, the information extracted from the VIS and IR images is combined together by the following rule: we retained width and height from the VIS domain, followed by 169 features (64 haar wavelet *haar*, 32 *gbr*, 7 *stm*, 8 *dct*, 16 *cooc*, 14 *rle* and 28 *laws*) corresponding also to the VIS domain and we added the similar 169 features from the IR domain. In this case of feature-fusion, the feature sets extracted from the VIS and IR images has been fused in order to create a new feature set which will represent each object:

$$(x_1, ..., x_p)_{VISIR} = (x_1, ..., x_m, x_{m+1}, ..., x_p)_{VISIR} \\ = (x_1, ..., x_m)_{VIS}, (x_{m+1}, ..., x_p)_{IR} \tag{5.1}$$

where $p = 340$, $m = 171$, $n = 169$ and $m + n = p$ for the vector obtained imediately after the features extraction step and denoted *AllFeatures*. For the case of the vectors obtained after performing also the features selection step, m, n and p will have lower values and they will depend on the number of retained features from both modalities VIS and IR.

In figures 5.2 and 5.3 the module of the features extraction, which could also imply a features selection stage, it is presented and here two possible situations can be noticed:

   - if the fusion is performed imediately after the features extraction stage (figure 5.2), then from the monomodal vectors VIS and IR by concatenation (equation 5.1) the bimodal feature vector $VISIR$ will be computed. If a second step of features selection (FS) follows, then on the $VISIR$ vector the FS operation will be applied and a new vector $sVISIR^c$ will be obtained, from which the monomodal vectors $sVIS^c$ and sIR^c could be computed. These last vectors could be used when the performances of the fused FVs are to be compared with the performances of some vectors wich does not imply any type of fusion, i.e. when bimodal vectors are to be compared with the monomodal vectors;

   - if in a first step the feature selection operation is realized, and only after that the fusion is performed above the results (figure 5.3), it means that the FS operation is applied on the monomodal VIS and IR vectors and only after obtaining the selected features $sVIS^i$ and sIR^i, they will be combined in the fused bimodal vector $sVISIR^i$.

The first case, when features selection is applied on the concatenated FVs is the one from chapter 4, from the right side of the tables 4.3, 4.4, 4.5 and 4.6, and the second case, when the features selection operation is applied directly on the individual vectors, is the one from the same tables, but the left side. From the previous chapter, after analyzing the results obtained for both possibilities: to apply

the FS scheme to the individual vectors VIS and IR or to apply it to the concatenated vector VISIR, we decided to use the individual one. This is also motivated by the choice to consider the most relevant information on each domain, but also by a possible processing of the data in a parallel way. Also, from the previous chapter we have obtained two FS methods giving better results than their *AllFeatures* counterparts. Therefore, when we will refer to vectors obtained after the application of the FS procedure, we will denote those vectors $sVIS_m^i$, sIR_n^i or $sVISIR_p^i$.
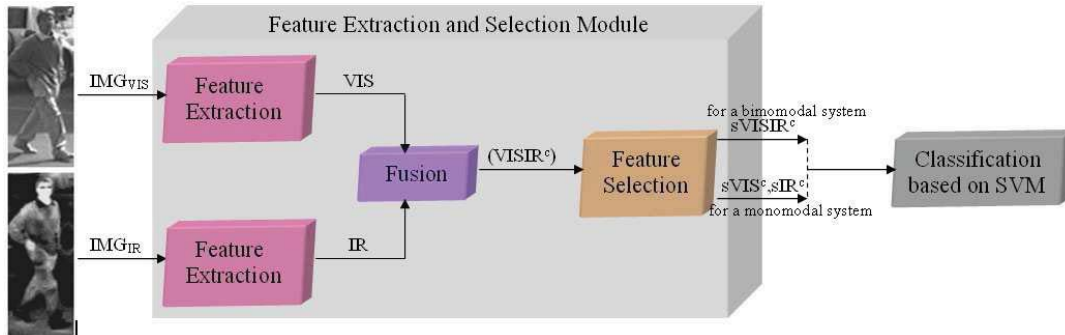
Figure 5.2: Feature-fusion before the Feature Selection step
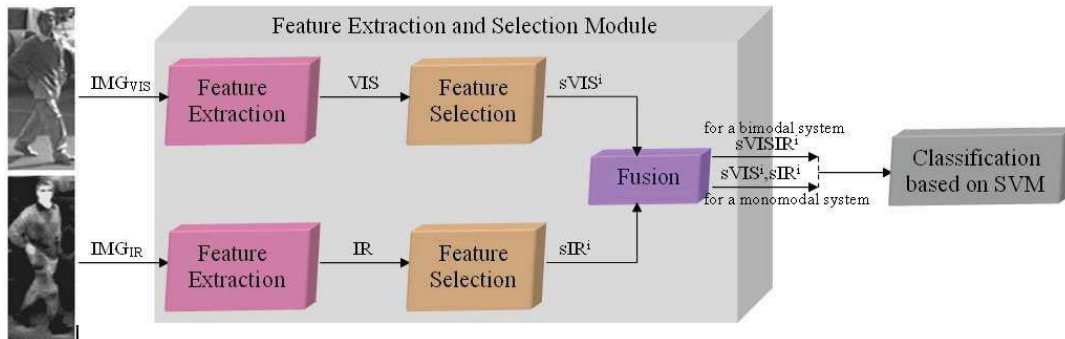

Figure 5.3: Feature-fusion after the Feature Selection step

The advantage of this fusion scheme (i.e. at the feature level) is, as we will see in the section dedicated to the experimental results, that provides very good results, i.e. good recognition rates. Still, it has a strong disadvantage: because the fusion process is realised at a low level, this fusion scheme requires a good correlation of VIS and IR data. Therefore, in cases when host vehicle is moving on a road with ditch, holes, bumps, it is possible that cameras to be moved from their initial position and thus to produce images not correlated eachother. Even if this case of fusion is not that sensitive to the images decorrelation as the data-fusion case, still some possible errors caused by the movement of the vehicle could appear.

All three types of vectors, obtained before (VIS_{171}, IR_{171} and $VISIR_{340}$) but also after ($sVIS_m^i$, sIR_n^i and $sVISIR_p^i$) the application of the FS operation will be evaluated also by an SVM with classical kernel, i.e. SK, in order to compare their performances (in section 5.4). Even the kernel type is the same (an SK) at this fusion scheme and at the monomodal systems, it is possible

that after the SVM kernel optimization process, performed on the validation set, the optimised SK obtained on the bimodal VISIR vector to be different from those obtained on VIS or IR domain. In this case of features-fusion, the model of the SVM will be determined by the kernel hyperparameters: ($SKtype_{VISIR}$,C_{VISIR},$SKparameter_{VISIR}$). To compute the bimodal FV, the features provided by the VIS and IR images will be combined in a single vector VISIR. For this new vector, the accuracy is computed by the 10f-CV procedure, and the maximum value of the accuracy obtained on the training set will determine the value of the winner (i.e. optimized) $SK*$ to be retained.

5.2.2 Matching-score fusion

Within the fusion of scores, where the fusion is performed in a later stage, i.e. after the classification process is already addressed on the two VIS and IR domains separately, the monomodal vectors will be used as inputs for the classifier, as it can be seen from figure 5.4.
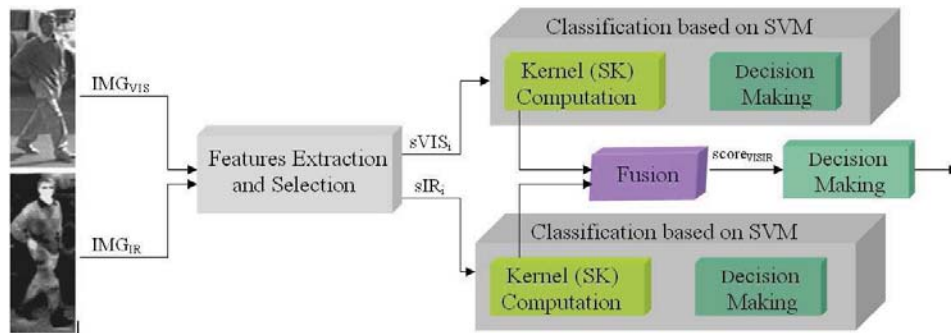

Figure 5.4: Matching score-fusion

For a matching-score fusion, multiple classifiers output a set of matching scores which represent the probabilities that one object belongs to different possible classes, based on different modalities. The matching scores generated by the classifiers from the VIS and IR modalities for a given test object s can be combined by the weighted parameter α in order to obtain a new match score which is then used to make the final decision:

$$H^*(s) = argmax_{i=1}^{N} P_{VIS}^{\alpha}(H_i, s) \cdot P_{IR}^{1-\alpha}(H_i, s), \tag{5.2}$$

where P_{VIS} is the score from the visible subsystem, P_{IR} is the score from the infrared subsystem, α is a weighting factor which varied between 0 and 1, N is the number of classes in which objects could be classified and $H^*(s)$ is the retained classification hypothesis for object s. The scores of each subsystems are in fact the output probabilities from the corresponding classifier: SK from VIS or SK from IR for the i-th class hypothesis. By using the weight defined by the system, the normalized scores from both subsystems are fused using equation (5.2) and a new, bimodal score is obtained.

In order to calculate the optimal weight, α_*, two methods have been considered here: a static Adaptive Fusion of Scores (sAFScores) and an dynamic Adaptive Fusion of Scores (dAFScores).

In the case of sAFScores, the weight α is fixed in the range $[0; 1]$. It is obtained by knowing the matching scores and the real class for each object from the validation database.

The weight for the dAFScores scheme is dynamic, being adapted correspondingly to the quality of the current input (test image) instead of using the optimum weight that is estimated from the available

training set. The weighted factor from the VIS domain, which could be different for an object s, is computed as a function of the VIS and IR scores dispersions:

$$\alpha(s) = \frac{\sigma_{VIS}(s)}{\sigma_{VIS}(s) + \sigma_{IR}(s)}, \tag{5.3}$$

where the VIS and IR scores dispersions are computed as a mean:

$$\sigma_{u \in \{VIS, IR\}}(s) = \frac{\displaystyle\sum_{i,j=1, i \neq j}^{N} |P_u(H_i, s) - P_u(H_j, s)|}{C_N^2}. \tag{5.4}$$

This second approach is more advantageous especially when the system is implemented in uncertain environment conditions or it is crossing some different extreme situations (e.g. when the vehicle passes through a tunnel). The dAFScores approach directly validates the quality of the incoming test image so as to adaptively change the weighting factor for fusion of both subsystems scores. It is important to priory check the monomodal systems because unreliable data give incorrect scores hence affect the accuracy of the total scores of the fusion systems.

SVMs are excellent tools for classification, novelty detection, and regression but they do not provide probabilities. They only provide the estimated class for the test object. From Libsvm, which is the implementation we addopt in our processing, the well known C-svc formulation has been used, and it also supports class-probabilities output. If one choose to use the probability model defined in Libsvm, the output probabilities will contain the parameters of the sigmoid fitted on the decision values. The name of the formulation is coming from the parameter C, which is the cost of constraints violation and a constant of the regularization term in the Lagrange formulation. The output of a classifier should give the possibility to provide the posterior probability in order to enable different post-processing. Standard SVMs do not provide such probabilities, but Platt (Platt, 1999) among others proposed a solution to create probabilities by training the SVM and then to train the parameters of an additional sigmoid function to map the SVM outputs into probabilities. This implementation yields probabilities of comparable quality to the regularized maximum likelihood kernel method. As H.-T. Lin, the creator of Libsvm, stated in (Lin *et al.*, 2007), the Libsvm probability output is the implementation of Platt's algorithm, described in (Platt, 1999) and it is just a monotonic transform of regular output of SVM by a sigmoid function.

The system adaptation to the context:

The main advantage of these fusion schemes is that the level at which they are computed does not require a strong correlation of VIS and IR images. Because these two types of data are processed at a high level, therefore, after the classification process began, also the time required by the system could be very much decreased by the processing of the data in a parallel way. The only problem which could appear in these types of fusion is that of contradicted scores between multiple classes of objects. For example, from the VIS modality the probability scores will state that the test object is a pedestrian with a probability of 0.9, but from the IR modality the probability scores will mentioned that the same test object is a background object with the same 0.9 probability. Which one will be more credible in this situation? Even for such situations, the adapted weighted parameter α demonstrates its main scope, i.e. to give more or less credibility to one of the two modalities, based on some information about the environmental conditions in which the system is at that moment of time. Therefore, the α parameter aims to strengthen the decision provided by one or another modality.
Each simple kernel is involved with a weight that represents its relative importance for the classification process. The kernel selection process, with the optimization of the hyper-parameters is described in what follows.

The static-adaptive fusion of scores (sAFScores) requires that in the validation stage, by knowing the matching scores and the real class for each object from the validation database, the accuracy to be computed using the SVMs from the VIS and IR domains which were previously determined as being the optimised ones on the monomodal systems. Before computing the accuracy, the matching scores are also weighted with all possible values of an weighted coefficient, α, established in the domain [0;1]. From all computed accuracies by the 10f-CV procedure, the maximum one obtained in the validation stage will be chosen and thus the optimised $\alpha*$ to be used on the test set is found. This static-adaptive fusion scheme will assure that the α parameter used to combine the matching scores for all the objects from the test database to be the same value.

The dynamic-adaptive scheme (dAFScores) implies a dynamic adaptation of the parameter α to the context. The parameter α adapts to each object from the test set, and may have different values from one test object to another. The value of α is calculated based on the scores dispersion of each test object.

5.3   MKs for kernel-fusion

The SVM formulation is based on kernel functions, that have already proven to be suited for complex classification problem from the real world in many types of applications. These kernel-methods represent the data by means of a kernel function, which defines similarities between pairs of items. Kernel methods have been intensively used in the frame of pattern classification problems also because the kernel function takes relationships that are implicit in the data and makes them explicit. Each kernel function has a specific functioning: it extracts a specific type of information from a given data set, and provides a partial description of the respective data by mapping the original data in a new hyperspace. Generally, classical kernel-based classifiers use only a single kernel (SK), but more and more applications from the real world prefer a combination of kernels in order to perform a better adaptation to the heterogeneous and multi-sensorial data. Our goal in the kernel-fusion case is to find a multiple kernel (multiple kernel (MK)) that best represents all of the information available for the two types of images.

The intermediate-level fusion scheme we propose is the one based on the SVM's kernels. Each type of information which will compute the bimodal vector given as input to the system, will be processed by the corresponding kernel. This fusion scheme takes place inside the SVM classifier, as it is presented in the figure 5.5.
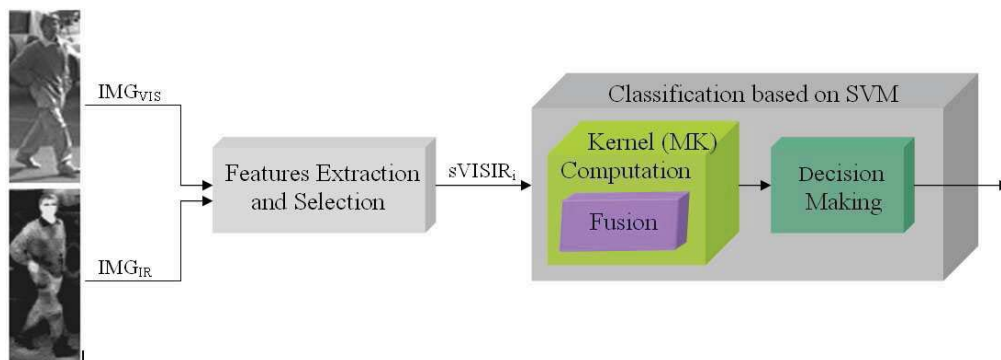

Figure 5.5: Kernel-fusion

Recent applications (Lanckriet *et al.*, 2004) based on SVMs have shown that using multiple kernels

(MK) instead of a single one can help in the interpretation of the decision function and improve the classifier performance. Given two kernels $K1$ and $K2$, inducing the embeddings $\Phi_1(x)$ and $\Phi_2(x)$, respectively, it is possible to the kernel $K = K1 + K2$, inducing the embedding $\Phi(x) = [\Phi_1(x); \Phi_2(x)]$, but even of greater interest, is to use a parameterized combinations of kernels. A common approach is to consider that the kernel function $K(x_i, x_j)$ is a linear combination of the basic kernels (Rakotomamonjy *et al.*, 2007) and this is:

$$K(x_i, x_j) = \sum_{d=1}^{D} \alpha_d K_d(x_i, x_j) \tag{5.5}$$

with $\alpha_d \geq 0$, $\sum_d \alpha_d = 1$, where D is the total number of kernels. Each basis kernel K_d may either use the full set of variables describing x_i or only a subset of these variables. The choice of weights α_d is another problem of data representation through the multiple kernel (MK) formalism.

Choosing $D = 2$ in (5.5), the obtained MK will be the sum of two independent kernels, each one corresponding to one modality VIS or IR and weighted with a value α_d representing the context. We propose for the VIS-IR kernel-fusion case a MK learned as a linear combination of two kernels:

$$
\begin{aligned}
MK(x_i, x_j) = &\alpha \cdot SK_{VIS}(x_i^{1,k}, x_j^{1,k}) \\
&+ (1-\alpha) \cdot SK_{IR}(x_i^{k+1,n}, x_j^{k+1,n})
\end{aligned}
\tag{5.6}
$$

where the single kernels SK_{VIS} and SK_{IR} could be any simple kernel with similar or different hyper-parameters. In (5.6) the SKs represent simple kernels, like those used by the classical SVMs, where a single kernel function is used for all the components of a vector which represents an object in the SVM hyperplane. These SK functions could be of different types and could have different hyper-parameters in (5.6), therefore one could chose one SK for the VIS domain and another SK for the IR domain. The value α, respective $1 - \alpha$ represents the weight assigned to the VIS kernel, and IR respectively. The values $x_i^{1,k}$, $x_j^{1,k}$ are the first k components of the feature vectors x_i, respectively x_j which retain the information from the VIS domain, while the values $x_i^{k+1,n}$, $x_j^{k+1,n}$ are the last $n - k$ components which retain the information from the IR domain. The calculated weighted sum from (5.6) represents the value of the proposed MK (which is a distance, in fact).

As we already mentioned in chapter 3 at section 3.2.6, the single kernel could be either RBF, or Polynomial: $SK \in \{RBF, POL\}$.

Our MK solution uses a linear combination of simple kernels for different types of feature vectors $VISIR$, revealing thus different combinations referred as: $RbfPol(VISIR)$, $PolRbf(VISIR)$, $Rbf_{VIS}Rbf_{IR}(VISIR)$ and $Pol_{VIS}Pol_{IR}(VISIR)$. From all these possible combinations we concluded that our MK has the following parameters: the kernel type (RBF or POL), the context adjustment value α, the penalty parameter C and the kernel specific parameters, according to the kernel type: the bandwidth γ and the order d (for the MK having different types of kernels), γ_{VIS} and γ_{IR} (for the MK having two SKs of type RBF but with different hyperparameters) and d_{VIS} and d_{IR} (for the MK with two different POL kernels). The parameters of the SK from the VIS domain will be denoted p_{VIS}, while the parameters of the SK corresponding to the IR domain will be p_{IR}. Thus, our MK is entirely described by the parameter set: $(MKtype, C, MKparameter_{VIS}, MKparameter_{IR}, \alpha)$.

The system adaptation to the context:

For the idea of the adaptation to the system context, we propose two possible solutions: the first one is to automatically determine the context and to have a battery of classifiers for each possible

context from which the proper one to be chosen. Therefore, every time the database is changing, all possible systems have to be learned again in the training step and from all those to pick up the best one. If the determined context is not found in the battery of classifiers, we propose to choose the closest classifier in terms of the context parameter value. The second possible solution is to use a weighted parameter. But in this case the inconvenient is that this solution is valid only for the case of a fusion of scores, so the scheme cannot be applied to any level we proposed the fusion.

The weighted parameter which establishes the importance of each modality in a specific context, could be determined on the training set, as in the case of feature-fusion, kernel-fusion and sAFScores schemes, or it could be adapted on the test set (as in the case of the dAFScores scheme), or even both combined (one fixed value to be determined on the test set and a dynamic value to contribute to this first one, this latter being determined on-line on the test set).

The optimal model of SVM for a given problem corresponds to the configuration that generates the best classification performance using the 10-folds cross-validation (10f-CV) technique. Each simple kernel is involved with a weight that represents its relative importance for classification. The kernel selection process, with the optimization of the hyper-parameters, but also of the weighting value α is performed in the validation stage, by using the 10f-CV procedure. The accuracies for all possible combinations of the SVM hyperparameters, also combined with the weighted coefficient α are computed and the higher value will determine the optimised set of MK's parameters: ($MKtype$, C, $MKparameter_{VIS}$, $MKparameter_{IR}$, α).

5.4   Experiments and results

In Weka there is a collection of machine learning algorithms, but there is no algorithm to treat the fusion problem. Thus, we implemented our fusion schemes, starting from a similar toolbox (libsvm) of classification integrated for MATLAB (Chang & Lin, 2001).

For these four possibilities for the system to function (one where no fusion scheme is applied and the three proposed fusion schemes), the methodology of choosing the classifier is different and depends on the system functioning scheme. For the SVM model selection task, performed in the validation stage, the methodology could be:

- if we consider the case of the **monomodal systems** VIS and IR cases, so where **no fusion** is applied, then the model selection task consists in choosing the best model, i.e. the SK for the SVM classifier on the VIS domain and on the IR domain with the corresponding hyperparameters: ($SKtype_{VIS}$,C_{VIS},$SKparameter_{VIS}$) and ($SKtype_{IR}$,C_{IR},$SKparameter_{IR}$) (where $SKparameter$ could be the order d for the POL kernel or the bandwidth γ for the RBF kernel) determined by a 10 folds cross-validation procedure. In this case, the monomodal systems are able to process a single type of information , either the one extracted from the VIS image or the one extracted from the IR image.

- if we are in the case of a **feature-fusion**, where bimodal vectors should be considered as inputs to the system, in the same manner as for the monomodal systems we have to act for the selection of the SVM model. Therefore, the model of the SVM which is also an SK, will be determined by ($SKtype_{VISIR}$,$SKparameter_{VISIR}$,C_{VISIR}). The only difference is that instead of monomodal FVs (VIS and IR) some bimodal FVs (VISIR) will be employed. To compute the bimodal FV, the features provided by the VIS and IR images will be combined in the frame of the same vector. For the bimodal vector and for each combination of the SK hyper-parameters, the accuracy is computed by the 10f-CV procedure, and the maximum value of the accuracy will determine on the training set the value of the hyper-parameters coresponding to the optimized $SK*$. Finally, for the test objects, the features from the VIS and IR domains will be combined in the same manner as in the training stage in the frame of the bimodal feature vector. Therefore, in the

case of a fetaure-fusion scheme, some bimodal vectors should be considered as inputs to the classifier and in the same manner as for the monomodal systems we have to act for the selection of the proper SVM model. Therefore, the model of the SVM will be also an SK and it will be optimized also by a 10f-CV procedure.

- if we consider the case of a **fusion of SVM kernels**, a MK will be used instead of a SK one for the SVM classifier to discriminate between different objects. In this case the information provided to the fusion system will be also a bimodal one, as int he case of the feature-fusion scheme. In the validation stage of the system, by using the 10f-CV procedure, the accuracies for all possible combinations of the SVM hyper-parameters corresponding to the MK and combined with a weighted coefficient α will be computed and the higher one will determine the optimised value of α to be used on the test set. So in this case, the searched parameters are those of a MK: ($MKtype$, C, $MKparameter_{VIS}$, $MKparameter_{IR}$, α);

- if we consider the case of a fusion of scores, in the validation stage, we act in the same manner as in the case of monomodal systems: the winner SVMs determined on the VIS and IR modality separately, with the corresponding hyperparameters will be used. The static-adaptive fusion scheme requires that in the validation stage, by knowing the matching scores and the real class for each object from the validation database, the accuracy to be computed using the SVMs from the VIS and IR domains which were previously determined as being the winners on the monomodal systems. Before computing the accuracy, the matching scores are also weighted with all possible values of an weighted coefficient, α, established in the domain [0;1] with the step of 0.1. From all computed accuracies by the 10f-CV procedure, the maximum one will be chosen and thus the optimised $\alpha*$ to be used on the test set is found. This static-adaptive fusion scheme will assure that the α parameter used to combine the matching scores for all the objects from the test database to have the same value. Unlike this scheme, the dynamic-adaptive one implies a dynamic adaptation of the parameter α to the context. The parameter α adapts to each object from the test set, and may have different values from one test object to another. The value of α is calculated based on the scores dispersion of each test object.

Because it is not known beforehand which parameters for the SVM kernels (even SKs or MKs gives the best solution for one problem, as we stated in chapter 3 at section 3.2.6) there must be done a model selection (a parameter search) that could identify appropriate hyper-parameters but also the α weighted value (for the proposed fusion schemes). When this optimization process is finished, a winner kernel, i.e. an optimised one is chosen on each modality: SK^*_{VIS} and SK^*_{IR} for the optimization of the monomodal systems and a single winner kernel SK^*_{VISIR} for the optimization of the bimodal system (or the feature-fusion case).

The optimization of the MK's parameters set ($MKtype$, C, $MKparameter_{VIS}$, $MKparameter_{IR}$, α) for the kernel-fusion case has been performed in the following manner: for each combination of kernel type, C, p_{VIS}, p_{IR} and α parameters, the accuracy is computed. The best accuracy denotes the winner multiple kernel MK^*_{VISIR} and it's hyper-parameters. Combination of kernels which process combinations of features can be revealed by the use of the MK formalism.

The score-based fusion system uses raw scores from the visible and from the infrared subsystems combined by the weighted exponential fusion rule. For the static-adaptive fusion systems, the optimum weight has been chosen by the 10f-CV optimization process. First, sAFScores the winner SKs from the monomodal systems, SK^*_{VIS} and SK^*_{IR} have been used to provide the scores for each modality; then, combining these scores with all the possible values for α, the best accuracy denotes the winner α_*. For the second case, in the dynamic-adaptive fusion dAFScores approach, the optimal α could be a different value for each object from the test set.

In table 5.1 the results of the proposed fusion schemes are presented, together with the results obtained when no fusion scheme was applied. An accuracy (or recognition rate in the frame of our

Table 5.1: SK and MK optimization for the proposed fusion schemes

System inputs			MONOMODAL SYSTEMS		BIMODAL SYSTEMS				SCORES	
			NO FUSION		FEATURES FUSION		MULTIPLE KERNEL KERNELS FUSION		SCORES FUSION	
Met.	Pb.	Input vector	Performance		Performance		Performance		Performance	
		Input	$bAcc$	winner SK	$bAcc$	winner SK	$bAcc$	winner MK	$bAcc_{sAFSc}$	$bAcc_{dAFSc}$
10f-CV	4 classes AllFeatures	VIS_{171}	90.1	$(RBF,128,2^{-15})$						
		IR_{171}	92.2	$(RBF,512,2^{-19})$	92.9	$(RBF,128,2^{-19}),\alpha_*=0.4$	94.2	$(RbfRbf,128,2^{-19},2^{-17}),\alpha_*=0.5$	93.0 $\alpha_*=0.5$	- -
		$VISIR_{340}$ -	-	-						
LT		VIS_{171}	94.7	-						
		IR_{171}	94.9	-	97.3	-	96.7	-	98.7	98.4
		$VISIR_{340}$	-	-						
10f-CV	4 classes FS_{7CV}	$sVIS_{65}$	91.3	$(POL,0.5,1)$						
		sIR_{62}	90.4	$(RBF,512,2^{-17})$	92.9	$(RBF,32,2^{-17}),\alpha_*=0.5$	94.3	$(RbfRbf,128,2^{-17},2^{-19}),\alpha_*=0.5$	92.8 $\alpha_*=0.5$	- -
		$sVISIR_{127}$ -	-	-						
LT		$sVIS_{65}$	96.2	-						
		sIR_{62}	95.2	-	97.7	-	96.9	-	98.7	98.7
		$sVISIR_{127}$	-	-						
10f-CV	4 classes $FS2a\_75$	$sVIS_{128}$	89.3	$(RBF,2048,2^{-19})$						
		sIR_{128}	92.4	$(RBF,512,2^{-19})$	92.9	$(RBF,32,2^{-17}),\alpha_*=0.4$	92.7	$(RbfRbf,256,2^{-19},2^{-17}),\alpha_*=0.6$	92.7 $\alpha_*=0.4$	- -
		$sVISIR_{256}$ -	-	-						
LT		$sVIS_{128}$	95.6	-						
		sIR_{128}	95.5	-	97.0	-	96.1	-	98.4	98.3
		$sVISIR_{256}$	-	-						

classification problem) of 94.7% was achieved for the visible monomodal system and 94.9% for the infrared monomodal system. In these cases, when no-fusion scheme was applied, the input vector was the one corresponding to the respective domain: VIS_{171} or IR_{171}.

The accuracies for the proposed fusion schemes are presented for different types of FVs, obtained imediately after the features extraction step (the first group of results, denoted *AllFeatures*) or even after the application of the FS method on the individual vectors VIS_{171} and IR_{171} (the next 2 groups of results, denoted $FS7_{CV}$ and *FS2a_75*).

Performance of the system using the feature-fusion scheme, is observed to be better than any of the monomodal systems or the system employing a MK approach, but it is worse when comparing to the fusion of scores situation. One reason for obtaining higher accuracies with the bimodal vectors is that the input vector was containing all the features from both visible and infrared domains, therefore a double dimension (and include also some complementary features) of the FV.

It could be noticed that in all situations when comparing the results obtained with the monomodal systems (where no fusion scheme was employed) with the ones using fusion, the first set is worst than each of the set provided by the proposed fusion schemes. Therefore, all the proposed fusion schemes demonstrated efficiency at the classification with the SVM.

For the MK (or kernel-fusion) case the obtained maximum accuracy is higher than the ones corresponding to the monomodal systems for all three types of FVs we used, but it is below the accuracy of the bimodal system (or of the matching-score fusion), even the MK system and the bimodal system use the same input vector. Therefore, our MK solution could be used in this case just to improve the monomodal decisions. But for the database we employed it has to be considered that there are no multiple situations for the illumination or weather. Therefore, when such a database will be available, some improved results are expected. In addition, the dimension of the database is small, therefore there is not enough data for a correct validation of this case of kernels fusion.

In both cases of scores-fusion, the obtained accuracy values are the best (compared with all the situations not using scores), even compared with the one corresponding to the bimodal vector. When using the fusion of scores, the input vector of each classifier is the one corresponding to the monomodal case (and for this reason the column "winner SK" is missing from the table in the case of scores-fusion). The scores were provided individually, for each monomodal vector VIS or IR, but in the scores-combination process the information from both types of features (i.e., the scores) was fused at the scores-level.

Another remark is that the SKs obtained after the optimization process are not the same in the experiments developed before or after the application of ths FS process. There are some winner SKs for the classification problem with 171 features and there are other winner SKs for the classification problem with the vectors obtained after applying the FS methods. Therefore, for each problem it must be performed an individual step of validation, where the hyper-parameters of the SK or MK to be determined together with the proper weighted value α for that respective context.

5.5   Conclusion

From the previous chapter, two methods of FS have been selected: one combining a Ranker search and an Information gain single-attribute evaluator (*FS2a_75*) and the other one combining a Best First search with a Correlation subset evaluator (*FS7$_{CV}$*). Both methods have provided on both modalities VIS and IR higher accuracies than their *AllFeatures* counterparts at the evaluation with kNN (with $k = 1$) and SVM classifiers. In this manner, the FS methods have proved efficiency from the accuracy point of view, as well as concerning the number of features used to encode the information about the objects from the database. Therefore, using FS methods, the computation time was significantly reduced, even it was the vector computation time (i.e. the time required to extract all the features characterizing the object image) or it was the object classification time (i.e. the time needed to classify a new test object).

In this chapter, different fusion schemes performed at different levels: low-level, high-level, but also an intermediate-level are presented. The low-level fusion is also called feature-level fusion, the high-level one is a matching-score fusion and the intermediate-level fusion we used is performed at the SVM's kernel level. All these three types of fusion were compared for our road-obstacles SVM-based classification problem. A comparative study of individual visual and infrared obstacle recognizers (i.e. monomodal systems) versus fusion-based systems (i.e. bimodal systems) was performed in this chapter. For each type of system we defined the inputs as being monomodal or bimodal feature vectors. The monomodal vectors are for the use with the monomodal systems (i.e. systems capable to process a single type of information, either VIS or IR) or for the systems employing a matching-scores fusion. The bimodal vectors are proper for the systems using a feature-fusion scheme or a kernel-based fusion. In the previous chapter, it has been mentioned that we preferred the FS method to be performed on the individual vectors, on each modality separately, in order to select the most relevant features on each specific domain. There, the evaluation of these vectors (provided by the selected FS methods) has been considered in the frame of the monomodal systems. In this chapter, the proposed fusion schemes have been evaluated by the use of the same FS methods and the results were compared with the ones previously obtained. The *AllFeatures* FVs (i.e. those comprising all the features, VIS_{171}, IR_{171} and $VISIR_{340}$) have also been evaluated by the fusion systems we propose.

From the analysis of the obtained results, it has been noticed that all the proposed fusion schemes are better than the monomodal systems, either they were evaluated with reduced FVs on which it was applied a FS operation or they were evaluated with *AllFeatures* FVs on which no FS operation was applied. Comparing all the fusion schemes we propose, best results are obtained for the matching-scores fusion schemes, followed by the ones which employ a feature-fusion scheme and finally the ones rendered by the kernel-fusion based systems. Even the kernel-based fusion scheme (which we believe is the most proper one for our problem of road-obstacle recognition) have not provided the best results from all the fusion schemes we employed, we believe that a greater and better balanced database could render much better results. In the case of the database we use for the evaluation of the fusion schemes, there are not multiple illumination or weather contexts registered and in addition the dimension of the database was very small. It contained too few instances for the classes which have to be learned by a multiple kernel. When using a MK instead of a SK one for the SVM, it is expected that the second one to learn better the instances of a small database, because there are not so many different combinations of hyper-parameters to be optimized as in the case of a MK. Therefore, when a complex database will be available, we believe the results render by the kernel-fusion scheme to be much improved when comparing with the ones provided by the other two fusion schemes.

In order to ensure the adaptation of the system to the environmental conditions, within fusion schemes the features, the kernels and respectively the matching-scores were weighted (with a sensor weighting coefficient) according to the relative importance of the modality sensors. This allowed for better classification performances.

Part III

Final Considerations

Conclusion and Final Considerations

The aim of the present PhD thesis was to investigate the fusion of the visible-infrared information in the frame of an Obstacle Recognition system.

Several fusion schemes, at different levels and employing a SVM have been developed in the present work and their application has been performed in the frame of a 4-class road obstacle classification problem. The purpose of the proposed fusion schemes is besides improving the recognition rates, the possibility to adapt the system to different environmental contexts, based on weighted schemes between the visible and infrared information provided by cameras. There are three main contributions resulting from this effort.

The first contribution is concerning the problem of representing the information enclosed in the images of the obstacles to be discriminated. As concerning the features extraction operation, we propose the use of some general and fast to compute features which could be used to compute some monomodal or bimodal vectors. Each family of features has been evaluated in order to estimate the benefit it was added to the final feature vector. We proposed the use of a combination of different families of features in order to encode the information about the obstacles to be classified.

In the frame of the features selection, we proposed a different manner to apply and evaluate the features selection methods. The selection of the most relevant features could be performed by the use of Ranker or Search methods, and for each of these we propose the use of thresholds with different meaning in order to retain the features to be comprised in the selected feature vector. The criteria used to select these features was based on the accuracy of the classification and it was considered that on each modality, visible and infrared, the resulted feature vector should overcome the results obtained with the initial vectors, i.e. which does not use such a features selection operation.

For the fusion schemes we propose the features, kernels or matching scores to be combined in order to improve the performances of the recognition system. These could also be weighted before their combination in order to assure the system adaptability to different environmental contexts.

Further work will concern the improvement of the signature used by now to encode the information about the available database with some other types of features, besides the ones we employed. For example, some edge-based features, the coefficient of other transforms (such as PCA, SURF points), features considering geometric properties such as area, or even some dynamic features (like the ones based on active contours) among others. These complementary features would contribute to better encode the information about the objects when a complex database registered in real conditions will be available.

Moreover, a new FS method adapted to this problem of ODR systems is under development. We aim the use of multiple FS methods and their results to be combined in an average rank value or an average index of relevance available for each feature. A new method to constructs an average rank value for each feature and based on a bi-level criteria (accuracy of the classification and time required to extract those features) to select only the most relevant features in order to compute the final FV. In this manner, we could control the length of the final FV and thus we could decide how many features to be comprised in the final FV. Features which could be separable within their family could be individually computed in order to gain in the computation time.

Relative to the fusion problem, some other methods to perform the fusion of the VIS-IR information could be implemented and evaluated. We also aim to address some other levels at which to perform the fusion, such as: data-level, rank level and decision level. All these schemes could be compared and could improve the results obtained by now. Data-level has already been study but because it was sensitive to the calibration problems, we chose not to consider it as a final fusion scheme. Still, if its

usage in the frame of an entire ODR system could improve the results (if its results will be considered just as a part of the final decision, therefore to be performed in parallel with some other powerful fusion schemes), it could be considered. Also, we envisioned to improve the feature-fusion scheme we proposed by the use of some weighted schemes regarding each feature or each family of feature, besides the weighted at the modality level. Thus, we propose to enhance the SVM learning with weighted features. Regarding the decision level fusion, multiple classifiers are aimed to contribute to the decision about the class of the test objects based on a majority vote for example. All the fusion methods we considered in this thesis treat the information using probabilities, while other based on possibilities, like the ones using Dempster-Shafer theory are also available. We propose to compare our fusion schemes with these ones. As further improvements, we intend to integrate these fusion schemes in an entire obstacle-detection and classification system.

We also propose as further improvements a method to determine the weighted parameter based on both sets of data, the training and the test one. In order to establish the importance of each modality in a specific context, this parameter could be determined on the training set, as a fixed value, but it also could be adapted on the test set, as a dynamic value. Thus, the contribution of both values (the fixed one but also the dynamic one) will be considered.

All these possible improvements have to be evaluated on a greater, complex and balanced database, with multiple contexts registered in order to test the proposed schemes in real conditions. This will also help the MK solution we propose, because when enough data will be available, the MK could better learn each instance. When multiple hyper-parameters are used to compute the kernel, also much many instances should be available in order to best learn the kernel with that respective database. Different types of kernels with different values for the hyper-parameters could be tested in order to enlarge the searching space and to find the best global solution for our problem.

Finally, we propose the recognition module (and hence the fusion) will be further integrated into a complete system performing the ODR task.

Bibliography

AHMAD, A., & DEY, L. 2005. A feature selection technique for classificatory analysis. *Pages 43–56 of: Pattern Recognition Letters 26(1)*. 93

ALEFS, B., SCHREIBER, D., & CLABIAN, M. 2005 (June, 6-8). Hypothesis based vehicle detection for increased simplicity in multi sensor ACC. *Pages 261–266 of: Procs. IEEE Intelligent Vehicles Symposium*. 19, 31

ALESSANDRETTI, G., BROGGI, A., & CERRI, P. 2007. Vehicle and guard rail detection using radar and vision data fusion. *Pages 95–105 of: IEEE Transactions on Intelligent Transportation Systems*. 19, 31

AMERICAN TECHNOLOGIES NETWORK (ATN) CORPORATION. 2010. *How night vision works*. `"http://www.atncorp.com/hownightvisionworks"`. 14

ANDREONE, L., TANGO, F., SCHEUNERT, U., CRAMER, H., WANIELIK, G., AMDITIS, A., & POLYCHRONOPOULOS, A. 2002 (June). A new driving supporting system, integrating an infrared camera and an anti-collision micro-wave radar: the EUCLIDE project. *Pages 519–526 of: Procs. IEEE Intelligent Vehicles Symposium*, vol. 2. 20, 31

APATEAN, A., EMERICH, S., LUPU, E., ROGOZAN, A., & BENSRHAIR, A. 2007 (September). Ruttier Obstacle Classification by use of Fractional B-spline Wavelets and Moments. *In: Proceedings of The IEEE Region 8 Eurocon 2007 Conference Computer as a Tool*. 62

APATEAN, A., ROGOZAN, A., & BENSRHAIR, A. 2008a (October, 12-15). Kernel and Feature Selection for Visible and Infrared based Obstacle Recognition. *Pages 1130–1135 of: 11th International IEEE Conference on Intelligent Transportation Systems (ITSC2008)*. 62

APATEAN, A., ROGOZAN, A., & BENSRHAIR, A. 2008b (May, 22-25). Objects recognition in visible and infrared images from the road scene. *Pages 327–332 of: IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2008)*, vol. 3. 62

APATEAN, A., EMERICH, S., LUPU, E., ROGOZAN, A., & BENSRHAIR, A. 2008c (March, 12-14). Wavelets and Moments for Obstacle Classification. *In: The Third International Symposium on Communications, Control and Signal Processing (ISSCSP2008)*. 62

APATEAN, A., ROGOZAN, A., EMERICH, S., & BENSRHAIR, A. 2008d. Wavelets as Features for Objects Recognition. *Pages 23–26 of: Acta Tehnica Napocensis - Electronics and Telecommunications*, vol. 49. ISSN1221-6542. 70

APATEAN, A., ROGOZAN, A., & BENSRHAIR, A. 2009 (July, 09-10). Information Fusion for Obstacle Recognition in Visible and Infrared Images. *In: International Symposium on Signals, Circuits and Systems (ISSCS2009)*. 113

APPIN KNOWLEDGE SOLUTIONS. 2007. *Robotics*. Infinity Science Press, Engineering Seeries. Chap. 6. 10

ARNELL, F. 2005. *Vision-based pedestrian detection system for use in smart cars*. M.Phil. thesis, Royal Institute of Technology, Stockholm, Sweden. 13, 37, 47

ARNELL, F., & PETERSSON, L. 2005 (June). Fast object segmentation from a moving camera. *Pages 136–141 of: Procs. IEEE Intelligent Vehicles Symposium*. 43, 47

BEIER, K., & GEMPERLEIN, H. 2004. Simulation of infrared detection range at fog conditions for enhanced vision systems in civil aviation. *Aerospace Science and Technology*, 63–71. 15

BEN FRANKLIN RACING TEAM. 2007. *2007 DARPA Urban Challenge: The Ben Franklin Racing Team - B156 Technical Paper*. "http://www.darpa.mil/grandchallenge/TechPapers/Ben_Franklin_Driving.pdf". 31

BEN FRANKLIN RACING TEAM, UNIVERSITY OF PENNSYLVANIA/LEHIGH UNIVERSITY. 2007. *Philadelphia, Pennsylvania*. "http://www.benfranklinracingteam.org/". 26

BENSRHAIR, A., BERTOZZI, M., BROGGI, A., FASCIOLI, A., MOUSSET, S., & TOULMINET, G. 2002 (17-21, June). Stereo vision-based feature extraction for vehicle detection. *Pages 465–470 of: Procs. IEEE Intelligent Vehicles Symposium*, vol. 2. 55

BERTOZZI, M., BROGGI, A., FASCIOLI, A., & NICHELE, S. 2000. Stereo vision-based vehicle detection. *Pages 39–44 of: Proceedings of the IEEE Intelligent Vehicles Symposium*. 34, 35, 47

BERTOZZI, M., BROGGI, A., CELLARIO, M., FASCIOLI, A., LOMBARDI, P., & PORTA, M. 2002a (July). Artificial Vision in Road Vehicles . *Pages 1258–1271 of: Proceedings of the IEEE*, vol. 90. 33

BERTOZZI, M., BROGGI, A., FASCIOLI, A., & LOMBARDI, P. 2002b (June). Vision-based Pedestrian Detection: will Ants Help? *Pages 1–7 of: Procs. IEEE Intelligent Vehicles Symposium*, vol. 1. 35, 44, 47

BERTOZZI, M., BROGGI, A., GRAF, T., GRISLERI, P., & MEINECKE, M. 2003a. IR Pedestrian Detection for Advanced Assistance Systems. *Pages 582–590 of: DAGM-Symposium*. 35

BERTOZZI, M., BROGGI, A., GRAF, T., GRISLERI, P., & MEINECKE, M. 2003b (June). Pedestrian detection in infrared images. *Pages 662–667 of: Procs. IEEE Intelligent Vehicles Symposium*. 35, 47

BERTOZZI, M., BROGGI, A., CHAPUIS, R., CHAUSSE, F., FASCIOLI, A., & TIBALDI, A. 2003c (October). Shape-based pedestrian detection and localization. *Pages 328–333 of: Procs. IEEE Conf. on Intelligent Transportation Systems*. 35, 44, 47

BERTOZZI, M., BROGGI, A., CHAPUIS, R., CHAUSSE, F., FASCIOLI, A., & TIBALDI, A. 2004 (June). Pedestrian Localization and Tracking System with Kalman Filtering. *Pages 584–589 of: Procs. IEEE Intelligent Vehicles Symposium*. 44

BERTOZZI, M., BROGGI, A., FELISA, M., VEZZONI, G., & ROSE, M. DEL. 2006. Low-level Pedestrian Detection by means of Visible and Far Infra-red Tetra-vision. *Pages 231–236 of: In Procs. IEEE Intelligent Vehicles Symposium*. 46, 54

BETKE, M., HARITAGLU, E., , & DAVIS, L. 2000. Real-Time Multiple Vehicle Detection and Tracking from a Moving Vehicle. *Pages 69–83 of: Machine Vision and Applications*, vol. 12. 36, 39, 42, 47

BLANC, C., TRASSOUDAINE, L., GUILLOUX, Y. LE, & MOREIRA, R. 2004 (June 28 - July 01). Track to track fusion method applied to road obstacle detection. *Pages 775–782 of: Procs. 7th International Conference on Information Fusion*. 20, 30, 31

BOMBINI, L., CERRI, P., MEDICI, P., & ALESSANDRETTI, G. 2006 (Mar). Radar-Vision Fusion for Vehicle Detection. *Pages 65–70 of: Procs. Intl. Workshop on Intelligent Transportation*. 19, 31

BOSER, B.E., GUYON, I.M., & VAPNIK, V.N. 1992. A training algorithm for optimal margin classifiers. *Pages 144–152 of: In proceedings of the 5th Annual ACM Workshop on COLT. ACM Press*. Pittsburgh, PA: In Haussler, D. (ed.). 80

BROGGI, A., BERTOZZI, M., FASCIOLI, A., & SECHI, M. 2000a (October). Shape-based Pedestrian Detection. *Pages 215–220 of: Procs. IEEE Intelligent Vehicles Symposium*. 36, 47

BROGGI, A., BERTOZZI, M., FASCIOLI, A., BIANCO, C.G. LO, & PIAZZI, A. 2000b (September). Visual perception of obstacles and vehicles for platooning. *Pages 164–176 of: IEEE Trans. Intelligent Transportation Systems*, vol. 1. 34

BROGGI, A., FASCIOLI, A., CARLETTI, M., GRAF, T., & MEINECKE, M. 2004a (June). A multi-resolution Approach for Infrared Vision-based Pedestrian Detection. *Pages 7–12 of: Procs. IEEE Intelligent Vehicles Symposium*. 35, 47

BROGGI, A., CERRI, PIETRO, & ANTONELLO, P.C. 2004b (June). Multi-Resolution Vehicle Detection using Artificial Vision. *Pages 310–314 of: Procs. IEEE Intelligent Vehicles Symposium*. 34, 47

B.S. SHIVARAM. 2010. *The science of imaging*. "http://galileo.phys.virginia.edu/classes/USEM/SciImg/home_files/introduction.htm". 10

BU, F., & CHAN, C.Y. 2005. Pedestrian detection in transit bus application - sensing technologies and safety solutions. *IEEE Intelligent Vehicles Symposium*, 100–105. 12, 29, 31

CABANI, Y. 2007. *Segmentation et mise en correspondance couleur. Application : étude et conception dún système de stéréovision couleur pour láide à la conduite automobile*. Ph.D. thesis, Institut National Des Sciences Appliquées INSA de Rouen, Rouen, France. 54, 55

CHAN, C.Y., & BU, F. 2005 (April 30). *Literature review of pedestrian detection technologies and sensor survey*. Tech. rept. California PATH Institute of Transportation Studies, Berkeley, CA. Mid-term report. 11, 12

CHANG, CHIH-CHUNG, & LIN, CHIH-JEN. 2001. *LIBSVM: a library for support vector machines*. 82, 121

CHU, A., & GREENLEAF, C.M. SEHGAL NAD J.F. 1990. Use of Gray Value Distribution of Run Lengths for Texture Analysis. *Pages 415–420 of: Pattern Recognition Letters*, vol. 11. 71

COCQUEREZ, J.-P., & S.PHILIPP-FOLIGUET. 1995. *Analyse dímages : filtrage et segmentation*. Masson. 71

CORNELL TEAM. 2007. *Technical Review of the DARPA Urban Challenge Vehicle*. "http://www.darpa.mil/grandchallenge/TechPapers/Team_Cornell.pdf". 31

CORNELL TEAM, CORNELL UNIVERSITY. 2007. *Ithaca, New York*. "http://www.cornellracing.com/". 26

COVER, T.M., & HART, P.E. 1967. Nearest neighbor pattern classification. *Pages 21–27 of: IEEE Transactions on Information Theory*. 73

CRISTIANINI, N., & SHAWE-TAYLOR, J. 2000. *An introduction to support vector machines*. Cambridge, UK: 1st edn. Cambridge University Press. 80

CURIO, C., EDELBRUNNER, J., KALINKE, T., TZOMAKAS, C., & VON SEELEN, W. 2000 (September). Walking Pedestrian Recognition. *Pages 155–163 of: IEEE Transactions on Intelligent Transportation Systems*, vol. 1. 24, 40, 47

CUTLER, R., & DAVIS, L.S. 2000 (August). Robust Real-Time Periodic Motion Detection, Analysis, and Applications. *Pages 781–796 of: PAMI*, vol. 22. 40, 41, 47

DARPA GRAND CHALLENGE. 2004-2005. *Autonomous Ground Vehicles*. "http://www.darpa.mil/grandchallenge04/index.htm". 25

DARPA GRAND CHALLENGE. 2007. *Urban Challenge*. "http://www.darpa.mil/grandchallenge/index.asp". 26

DELLAERT, F. 1997. *CANSS: A Candidate Selection and Search Algorithm to Initialize Car Tracking*. Tech. rept. CMU-RI-TR-97-34. Carnegie Mellon Robotics Institute. 37, 38, 47

DELLAERT, F., & THORPE, C.E. 1997. Robust car tracking using Kalman filtering and Bayesian templates. *In: Intelligent Transportation Systems*, vol. Proc. SPIE 3207. 38

DELLAERT, F., POMERLEAU, D., & THORPE, C.E. 1998. Model-Based Car Tracking Integrated with a Road-Follower. *In: IEEE International Conference on Robotics and Automation (ICRA)*. 38

DEMONCEAUX, C., & KACHI-AKKOUCHE, D. 2004. Robust obstacle detection with monocular vision based on motion analysis. *Pages 527–532 of: International Symposium on Intelligent Vehicles*. 43

ELZEIN, H., LAKSHMANAN, S., & WATTA, P. 2003 (June). A motion and shape-based pedestrian detection algorithm. *Pages 500–504 of: Procs. IEEE Intelligent Vehicles Symposium*. 38, 43, 47

ENZWEILER, M., & GAVRILA, D.M. 2009. Monocular Pedestrian Detection: Survey and Experiments. *Pages 2179–2195 of: IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31. 33

EWALD, A., & WILLHOEFT, V. 2000 (October). Laser scanners for obstacle detection in automobile applications. *Pages 682–687 of: Procs. IEEE Intelligent Vehicles Symposium*. 28, 31

FARDI, B., SCHUENERT, U., & WANIELIK, G. 2005 (June, 6-8). Shape and motion-based pedestrian detection in infrared images: a multi sensor approach. *Pages 18–23 of: Procs. IEEE Intelligent Vehicles Symposium*. 23, 24, 31

FLEISCHER, K., NAGEL, H.H., & RATH, T.M. 2002 (June). 3D-model-based-vision for innercity driving scenes. *Pages 477–482 of: IEEE Intelligent Vehicles Symposium*, vol. 2. 39, 47

FLIR APPLICATION STORY. 2009. *Bmw incorporates thermal imaging cameras in its cars*. "`www.flir.com/uploadedFiles/ApplicationStory_BMW.pdf`". 45

FLIR TECHNICAL NOTE. 2008. *Seeing through fog and rain with a thermal imaging camera*. "`www.flir.com/uploadedFiles/ENG_01_FOG.pdf`". 13, 15

FLOREA, F. 2007. *Annotation automatique dímages médicales en utilisant leur contenu visuel et les régions textuelles associées. Application dans le contexte dún catalogue de santé en ligne*. Ph.D. thesis, Institut National Des Sciences Appliquées, INSA de Rouen and Technical University of Bucharest, Romania, Rouen, France. 71, 72

FRANKE, U. 1992. Real time 3D-road modeling for autonomous vehicle guidance. *Page 277 of: Selected Papers of the 7th Scandinavian Conference on Image Analysis, World Scientific Publishing Company*. 17, 31

FRANKE, U., GAVRILA, D.M., GORZIG, S., LINDNER, F., PAETZOLD, F., & WOHLER, C. 1999. Autonomous driving goes downtown. *Pages 40–48 of: IEEE Intelligent Systems*, vol. 13. 38, 39

FUERSTENBERG, K.C., & DIETMAYER, K. 2004. Object tracking and classification for multiple active safety and comfort applications using a multilayer laser scanner. *Pages 802–807 of: Procs. IEEE International Symposium on Intelligent Vehicles*. 28, 29, 31

FUERSTENBERG, K.C., DIETMAYER, K.C.J., & WILLHOEFT, V. 2002 (July 10). *Pedestrian Recognition in Urban Traffic using a vehicle based Multilayer Laserscanner*. 28, 31

GALLOWAY, M.M. 1975. Texture analysis using grey level run lengths. *Pages 172–179 of: Comput. Graphics Image Process*, vol. 4. 71

GANDHI, T., & TRIVEDI, M.M. 2006. Pedestrian collision avoidance systems: A survey of computer vision based recent studies. *Pages 976–981 of: IEEE Conference on Intelligent Transportation Systems*. 33

GAVRILA, D. M., & GIEBEL, J. 2002. Shape-based pedestrian detection and tracking. *In: Procs. IEEE Intelligent Vehicles Symposium*. 39

GAVRILA, D.M. 2000. Detection from a moving vehicle. *Pages 37–49 of: In the European Conference on Computer Vision*, vol. 2. 38, 39, 47

GAVRILA, D.M., & GIEBEL, J. 2001. Virtual sample generation for template-based shape matching. *Pages 676–681 of: Cvpr*, vol. 1. Proc. of IEEE Conference on Computer Vision and Pattern Recognition. 39

GAVRILA, D.M., KUNERT, M., & LAGES, U. 2001. A multi-sensor approach for the protection of vulnerable traffic participants - the PROTECTOR project. *Pages 2044–2048 of: Procs. IEEE Instrumentation and Measurement Technology Conference*, vol. 3. 27, 28, 31

GERN, A., FRANKE, U., & LEVI, P. 2000 (October). Advanced lane recognition - fusing vision and radar. *Pages 45–51 of: Procs. IEEE Intelligent Vehicles Symposium*. 17, 31

GLOBAL SECURITY. 2010. *Space Based Infrared System - Infrared Tutorial.* "`http://www.globalsecurity.org/space/library/report/1998/sbirs-brochure/part02.htm`". 13, 14

GOLDBERG, D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. 90

GRAY TEAM, THE GRAY INSURANCE COMPANY. 2005. *Metairie, Louisiana.* "`http://www.graymatterinc.com/darpa-challenge.php`". 25

GRUBB, G., ZELINSKY, A., NILSSON, L., & RILBE, M. 2004. 3D vision sensing for improved pedestrian safety. *Pages 19–24 of: Procs. IEEE International Symposium on Intelligent Vehicles*. 38, 44, 47

GUYON, I., & ELISSEEFF, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, **3**, 1157–1182. 90

GUYON, I., GUNN, S., NIKRAVESH, M., & MASOUD, L.A. 2006. *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing).* Secaucus, NJ, USA: Springer-Verlag New York, Inc. 66, 93, 95

HALL, M. 1999a. Correlation-based feature selection for discrete and numeric class machine learning. *Pages 359–366 of: Proceedings of the Seventeenth International Conference on Machine Learning*. 94

HALL, M. 1999b. *Correlation-based Feature Selection for Machine Learning.* Ph.D. thesis, University of Waikato, Department of Computer Science, Hamilton, New Zealand. 94

HAMMOUD, R.I. 2009a. *Augmented Vision Perception in Infrared: Algorithms and Applied Systems.* Springer Publishing Company, Incorporated. "Multiresolution Approach for Noncontact Measurements of Arterial Pulse Using Thermal Imaging", Sergey Y. Chekmenev, Aly A. Farag, William M. Miller, Edward A. Essock, and Aruni Bhatnagar". Chap. 4, pages 105–130. 10, 13

HAMMOUD, R.I. 2009b. *Augmented Vision Perception in Infrared: Algorithms and Applied Systems.* Springer Publishing Company, Incorporated. "Multiresolution Approach for Noncontact Measurements of Arterial Pulse Using Thermal Imaging", Sergey Y. Chekmenev, Aly A. Farag, William M. Miller, Edward A. Essock, and Aruni Bhatnagar". Chap. 4, pages 371–401. 54

HANDMANN, U., LORENZ, G., SCHNITGER, T., & SEELEN, W.V. 1998 (June). Fusion of Different Sensors and Algorithms for Segmentation. *Pages 12–17 of: Procs. IEEE International Conference on Intelligent Vehicles*. 17, 31

HARALICK, R.M., SHANMUGAM, K., & DINSTEIN, I. 1973. Texture features for image classification. *Pages 610–621 of: IEEE Trans. Systems, Mans and Cybernetics*, vol. SMC-3. 71

HEISELE, B., & WOHLER, C. 1998. Motion-Based Recognition of Pedestrians. *Pages Vol II: 1325–1330 of: ICPR98*. 40, 41, 47

HOFFMAN, D.D., & FLINCHBAUGH, B.E. 1982. The interpretation of biological motion. *Pages 195–204 of: Biological Cybernetics*. 40

HONDA. "`http://world.honda.com/HDTV/IntelligentNightVision/200408/`". 45

HSU, C.-W., CHANG, C.-C., & LIN, C.-J. *A Practical Guide to Support Vector Classification*. 81

JAIN, A.K., FLYNN, P., & ROSS, A.A. 2008. *Handbook of Biometrics*. Springer. 113

J.R. QUINLAN. 1993. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann. 92

KASS, M., WITKIN, A., & TERZOPOULOS, D. 1987. Snakes, Active contour models. *Pages 259–268 of: First International Conference on Computer Vision IEEE Computer Society Press*. 24

KATO, T., NINOMIYA, Y., & MASAKI, I. 2002 (September). An Obstacle Detection Method by Fusion of Radar and Motion Stereo. *Pages 182–188 of: IEEE Transactions on Intelligent Transportation Systems*, vol. 3. 18, 31

KAWASAKI, N., & KIENCKE, U. 2004 (June). Standard platform for sensor fusion on advanced driver assistance system using Bayesian network. *Pages 250–255 of: Procs. IEEE Intelligent Vehicles Symposium*. 18, 31

KIRA, K., & RENDELL, L.A. 1992. A practical approach to feature selection. *Pages 249–256 of: Proceedings of the ninth international workshop on machine learning*. 93

KONONENKO, I. 1995. On biases in estimating multi-valued attributes. *Pages 1034–1040 of: Ijcai'95: Proceedings of the 14th international joint conference on artificial intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 93

KRUSE, F., FOLSTER, F., ARHOLDT, M., ROHLING, H., MEINECKE, M. M., & TO, T. B. 2004 (June). Target classification based on near-distance radar sensors. *Pages 722–727 of: Procs. IEEE Intelligent Vehicles Symposium*. 28, 31

KUMON, H., TAMATSU, Y., OGAWA, T., & MASAKI, I. 2005 (June, 6-8). ACC in consideration of visibility with sensor fusion technology under the concept of TACS. *Pages 447–452 of: Procs. IEEE Intelligent Vehicles Symposium*. 19, 31

LABAYRADE, R., AUBERT, D., & TAREL, J.P. 2002 (June). Real time obstacle detection in stereovision on non flat road geometry through V-disparity representation. *Pages 646–651 of: Procs. IEEE Intelligent Vehicles Symposium*, vol. 2. 23

LABAYRADE, R., ROYERE, C., & AUBERT, D. 2005 (June). A collision mitigation system using laser scanner and stereovision fusion and its assessment. *Pages 441–446 of: Procs. IEEE Intelligent Vehicles Symposium*. 23, 31

LANCKRIET, G.R.G., CRISTIANINI, N., GHAOUI, L.E., BARTLETT, P., & JORDAN, M.I. 2004. Learning the kernel matrix with semidefinite programming. *Pages 27–72 of: J. Machine Learning Research*. 119

LANEURIT, J., BLANC, C., CHAPUIS, R., & TRASSOUDAINE, L. 2003 (June). Multisensorial data fusion for global vehicle and obstacles absolute positionning. *Pages 138–143 of: Procs. IEEE Intelligent Vehicles Symposium*. 22, 31

LAWS, K.L. 1980. Rapid texture identification. *Pages 376–380 of: Proc. SPIE 238*. 71

LeGuilloux, Y., Lonnoy, J., Moreira, R., Bruyas, M.-P., Chapon, A., & Tattegrain-Veste, H. 2002 (June). PAROTO Project: The Benefit of Infrared Imagery for Obstacle Avoidance. *Pages 81–86 of: Procs. IEEE Intelligent Vehicles Symposium*, vol. 1. 20, 31

Li, Y. 2010. *Image segmentation and stereo vision matching based on declivity line: Application for vehicle detection*. Ph.D. thesis, Institut National des Sciences Appliquées, INSA de Rouen, Rouen, France. 54, 55

Lin, H.-T., Lin, C.-J., & Weng, R.C. 2007. A note on Platt's probabilistic outputs for support vector machines. *Pages 267–276 of: Machine Learning*. 118

Linzmeier, D.T., Mekhaiel, M., Skutek, M., & Dietmayer, K.C.J. 2005a. A Pedestrian Detection System based on Thermopile and Radar Sensor Data Fusion. *Pages 1272–1279 of: 7th International Conference on Information Fusion*. 21

Linzmeier, D.T., Vogt, D., Prasanna, R., Mekhaiel, M., & Dietmayer, K.C.J. 2005b (June). Probabilistic Signal Interpretation Methods For A Thermopile Pedestrian Detection System. *Pages 12–17 of: Procs. IEEE Intelligent Vehicles Symposium*. 21, 31

Liu, H., & Setiono, R. 1996. A probabilistic approach to feature selection: A filter solution. *Pages 319–327 of: Procs. of the Thirteenth International Conference on Machine Learning*. 90, 95

M. Gütlein. 2006. *Large scale attribute selection using wrappers*. Ph.D. thesis, Albert-Ludwigs-Universität Freiburg, Freiburg, Germany. 90

Mallat, S. 1998. *A wavelet tour of signal processing*. Academic Press. 70

Manjunath, B.S, & Ma, W.Y. 1996. Texture features for browsing and retrieval of image data. *Pages 837–842 of: IEEE Transactions on Pattern Analysis and Machine Intelligence (Special Issue on Digital Libraries)*, vol. 18. 71

Marchal, P., Gavrila, D., Letellier, L., Meineke, M., Morris, R., & Tons, M. 2003. Save-U: an innovative sensor platform for vulnerable road user protection. *Pages 1–14 of: World Congress on Intelligent Transportation Systems - ITS*. 25

Meinecke, M.M., Obojski, M., Tons, M., Dorfler, R., Marchal, P., Letellier, L., Gavrila, D.M., & Morris, R. 2003. Approach For Protection Of Vulnerable Road Users Using Sensor Fusion Techniques. *In: International Radar Symposium*. 25, 31

Meis, U., & Schneider, R. 2003 (June, 9-11). Radar image acquisition and interpretation for automotive applications. *Pages 328–332 of: IEEE International Symposium on Intelligent Vehicles*. 27, 28, 31

Mendes, A., Bento, L.C., & Nunes, U. 2004. Multi-target detection and tracking with a laser scanner. *Pages 796–801 of: Procs. IEEE International Symposium on Intelligent Vehicles*. 28, 29, 31

Milch, S., & Behrens, M. 2001. *Pedestrian Detection with Radar and Computer Vision*. `"http://www.smart-microwave-sensors.de/Pedestrian_Detection.pdf"`. 18, 31

MIT team. 2007a. *Cambridge, Massachusetts*. `"http://grandchallenge.mit.edu/"`. 26

MIT team. 2007b. *Technical Report-DARPA Urban Challenge*. `"http://www.darpa.mil/grandchallenge/TechPapers/MIT.pdf"`. 31

Mobus, R., & Kolbe, U. 2004. Multi-target multi-object tracking, sensor fusion of radar and infrared. *IEEE International Symposium on Intelligent Vehicles*, 732–737. 30, 31

Mohan, A., Papageorgiou, C., & Poggio, T. 2001 (April). Example-Based Object Detection in Images by Components. *Pages 349–361 of: PAMI*. 37, 38, 47

Monteiro, G., Premebida, C., Peixoto, P., & Nunes, U. 2006. Tracking and classification of dynamic obstacles using laser range finder and vision. *In: In Proc. of the IEEE, RSJ International Conference on Intelligent Robots and Systems -IROS*. 31

Ng, I., Tan, T., & Kittler, J.V. 1992. On Local Linear Transform and Gabor Filter Representation of Texture. *Pages 627–631 of: In International Conference on Pattern Recognition*. 71

Ngo, C.W. 1998. Exploiting Image Indexing Techniques in DCT Domain. *Pages 196–206 of: In APR International Workshop on Multimedia Information Analysis and Retrieval*. 71

Ngo, C.W., Pong, T.-C., & Chin, R.T. 2001. Exploiting image indexing techniques in DCT domain. *Pages 1841–1851 of: Pattern Recognition*, vol. 34. 71

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., & Poggio, T. 1997. Pedestrian detection using wavelet templates. *Page 193 of: Computer vision and pattern recognition*. IEEE Computer Society. 38

Papageorgiou, C., & Poggio, T. 1999 (October,24–28). Trainable pedestrian detection. *Pages 35–39 of: IEEE International Conference on Image Processing (ICIP-99)*. 38, 43, 45

Papageorgiou, C., & Poggio, T. 2000 (June). A Trainable System for Object Detection. *Pages 15–33 of: IJCV*, vol. 38. 38

Papageorgiou, C., Evgeniou, T., & Poggio, T. 1998. A trainable pedestrian detection system. *Pages 241–246 of: IEEE Intelligent Vehicles Symposium*. 38, 47

Perrollaz, M., Labayrade, R., Royère, C., Hautière, N., & Aubert, D. 2006 (June, 13-15). Long Range Obstacle Detection Using Laser Scanner and Stereovision. *Pages 182–187 of: Procs. IEEE Intelligent Vehicles Symposium*, vol. 2. 23, 31

Pietzsch, S., Vu, T. D., Burlet, J., Aycard, O., Hackbarth, T., Appenrodt, N., Dickmann, J., & Radig, B. 2008. Results of a Precrash Application based on Laser Scanner and Short Range Radars. *Pages 367–372 of: Procs. IEEE Intelligent Vehicles Symposium*. 31

Platt, J.C. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Pages 61–74 of: Advances in Large Margin Classifiers*. MIT Press. 118

Polychronopoulos, A., Scheunert, U., & Tango, F. 2004. Centralized data fusion for obstacle and road borders tracking in a collision warning system. *Pages 760–767 of: Procs. of the 7th International Conference on Information Fusion*. 20, 31

Pratt, W.K. 2001. *Digital image processing: Piks inside*. New York, NY, USA: John Wiley & Sons, Inc. 72

Rakotomamonjy, Alain, Bach, Francis, Canu, Stéphane, & Grandvalet, Yves. 2007. More efficiency in multiple kernel learning. *Pages 775–782 of: Icml '07: Proceedings of the 24th international conference on machine learning*. New York, NY, USA: ACM. 120

Red Team, Carnegie Mellon University. 2005. *Pittsburgh, Pennsylvania*. "http://www.cs.cmu.edu/~red/Red/". 25

RedTeam. 2005. *DARPA Grand Challenge Technical Paper*. "http://www.darpa.mil/grandchallenge04/TeamTechPapers/RedTeamFinalTP.pdf". 25, 31

Richter, E., Schubert, R., & Wanielik, G. 2008 (June, 4-6). Radar and Vision based Data Fusion-Advanced Filtering Techniques for a Multi Object Vehicle Tracking System. *Pages 120–125 of: IEEE Intelligent Vehicles Symposium*. 20, 31

Sadou, M., Polotski, V., & Cohen, P. 2004. Occlusions in obstacle detection for safe navigation. *IEEE International Symposium on Intelligent Vehicles*, 716–721. 29, 31

Sanderson, C., & Paliwal, K.K. 2002. *Information fusion and person verification using speech and face information*. Tech. rept. IDIAP-RR 02-33. IDIAP Research Report. 112

Scheunert, U., Cramer, H., Fardi, B., & Wanielik, G. 2004. Multi sensor based tracking of pedestrians: a survey of suitable movement models. *Pages 774–778 of: Procs. IEEE International Symposium on Intelligent Vehicles*. 23, 31

Schweiger, R., Neumann, H., & Ritter, W. 2005 (June, 6-8). Multiple-cue data fusion with particle filters for vehicle detection in night view automotive applications. *Pages 753–758 of: Procs. IEEE Intelligent Vehicles Symposium*. 19, 31

Serfling, M., Schweiger, R., & Ritter, W. 2008 (June, 4-6). Road course estimation in a night vision application using a digital map, a camera sensor and a prototypical imaging radar system. *Pages 810–815 of: Procs. IEEE Intelligent Vehicles Symposium*. 19, 31

Shashua, A., Gdalyahu, Y., & Hayun, G. 2004. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. *Pages 1–6 of: Procs. IEEE International Symposium on Intelligent Vehicles*. 37, 47

Sole, A., Mano, O., Stein, G.P., Kumon, H., Tamatsu, Y., & Shashua, A. 2004. Solid or not solid: Vision for radar target validation. *IEEE International Symposium on Intelligent Vehicles*, 819–8824. 18, 31

Stanford Racing team. 2007. *Stanford Robotic Vehicle Junior: Interim Report*. "http://www.darpa.mil/grandchallenge/TechPapers/Stanford.pdf". 31

Stanford Racing Team, Stanford University. 2005. *Palo Alto, California*. "http://cs.stanford.edu/group/roadrunner//old/index.html". 25

Stanford Racing team, Stanford University. 2007. *Palo Alto, California*. "http://cs.stanford.edu/group/roadrunner/". 26

Steux, B., Laurgeau, C., Salesse, L., & Wautier, D. 2002 (June). Fade: a vehicle detection and tracking system featuring monocular color vision and radar data fusion. *Pages 632–639 of: Procs. IEEE Intelligent Vehicles Symposium*, vol. 2. 17, 18, 31

Sun, Z., Bebis, G., & Miller, R. 2006a. Monocular Precrash Vehicle Detection: Features and Classifiers. *Pages 2019–2034 of: IEEE Transactions on Image Processing*, vol. 15. 70

Sun, Z., Bebis, G., & Miller, R. 2006b. On-road Vehicle Detection: A Review. *Pages 694–711 of: IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28. 33

Tartan Racing. 2007. *A Multi-Modal Approach to the DARPA Urban Challenge*. "http://www.darpa.mil/grandchallenge/TechPapers/Tartan_Racing.pdf". 31

Tartan Racing team, Carnegie Mellon University. 2007. *Pittsburgh, Pennsylvania*. "http://www.tartanracing.org/". 26

Tatschke, T. 2006 (June). Early sensor data fusion techniques for collision mitigation purposes. *Pages 445–452 of: Procs. IEEE Intelligent Vehicles Symposium*. 31

TerraMax Team, Oshkosh Truck Corporation. 2005. *Oshkosh, Wisconsin*. "http://www.terramax.com/". 25

TerraMaxTeam. 2005. *Oshkosh Truck Corporation, DARPA Grand Challenge Technical Paper*. `"http://www.darpa.mil/grandchallenge04/TeamTechPapers/TerraMaxFinalTP.pdf"`. 26, 31

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., & Mahoney, P. 2006. Winning the DARPA Grand Challenge. *Journal of Field Robotics*. `"http://robots.stanford.edu/papers/thrun.stanley05.html"`. 25, 31

Tons, M., Doerfler, R., Meinecke, M.M., & Obojski, M.A. 2004 (June). Radar sensors and sensor platform used for pedestrian protection in the EC-funded project SAVE-U. *Pages 813–818 of: Procs. IEEE International Symposium on Intelligent Vehicles*. 25, 31

Toulminet, Gwenaelle, Bertozzi, Massimo, Mousset, Stephane, Bensrhair, Abdelaziz, Broggi, Alberto, & Member, Senior. 2006. Vehicle Detection by Means of Stereo Vision-Based Obstacles Features Extraction and Monocular Pattern Analysis. *Pages 2364–2375 of: IEEE Transactions on Image Processing*, vol. 15. 36, 54, 55, 80

Trepagnier, P.G., Nagel, J., Kinney, P.M., Koutsougeras, C., & Dooner, M. 2006. KAT-5: Robust systems for autonomous vehicle navigation in challenging and unknown terrain. *Journal of Field Robotics*, 509–526. `"www2.selu.edu/Academics/Faculty/ck/paps/JFR.pdf"`. 25, 31

van Rijsbergen, C. 1979. *Information Retrieval, Second Edition*. Butterworths. 68

Vapnik, V.N. 1998. *Statistical learning theory*. New York, USA: Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley and Sons. 80

Victor Tango team. 2007. *DARPA Urban Challenge-Technical Paper*. `"http://www.darpa.mil/grandchallenge/TechPapers/Victor_Tango.pdf"`. 31

Victor Tango team, Virginia Tech. 2007. *Blacksburg, Virginia*. `"http://www.me.vt.edu/urbanchallenge/"`. 26

Viola, P.A., & Jones, M.J. 2001. Rapid object detection using a boosted cascade of simple features. *Pages 511–518 of: IEEE Computer Vision and Pattern Recognition (CVPR)*. 40

Viola, P.A., Jones, M.J., & Snow, D. 2003. Pedetrian using patterns of motions and appearance. *Pages 734–741 of: In IEEE Int. Conf on Computer Vision*. 40, 43, 47

Wöhler, C., & Anlauf, J.K. 1999. An adaptable time-delay neural network algorithm for image sequence analysis. *Pages 1531–1536 of: IEEE Trans. Neural Networks*, vol. 10. 40

WHO. 2004. *The world health organization*. `"http://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/en/index.html"`. 2

Witten, Ian H., & Frank, Eibe. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. 2 edn. San Francisco: Morgan Kaufmann. 90

Zhao, L., & Thorpe, C. E. 2000 (September). Stereo- and neural network-based pedestrian detection. *Pages 148–154 of: IEEE Trans. Intelligent Transportation Systems*, vol. 1. 44, 47

List of Acronyms

NWIR Near Wavelength InfraRed. 13, 14

OD Obstacle Detection. 4, 20, 32, 33, 43, 44, 55, 65
ODR Obstacle Detection and Recognition. 1, 2, 4, 5, 9, 10, 12, 27, 30, 32, 46, 48, 49, 53, 55, 58–60, 86, 87, 110–113, 129, 130
OR Obstacle Recognition. 32, 33, 55, 60, 62, 80

POL polynomial. 81–83, 120, 121
PROTECTOR Preventive Safety for Unprotected Road User. 27
PSD power spectral density. 28

radar RAdio Detection And Ranging. 4, 5, 10–13, 15–22, 24–30, 48, 53
RALPH Rapidly Adapting Lateral Position Handler. 38
RBF Radial Basis Function. 39, 81–83, 120, 121
RCS radar cross section. 17, 18, 28
RLE Run Length Encoding. 71
ROI region of interest. 15, 16, 18, 19, 21, 23–25, 34–41, 43, 54, 58

sAFScores static Adaptive Fusion of Scores. 117, 119, 121, 122
SK single kernel. 81–83, 88, 109, 116, 117, 119–122, 124, 125
sonar SOund Navigation And Ranging. 10–12
SSD Sum of Squared Differences. 43
SVM Support Vector Machine. 1, 5, 37–39, 45, 53, 55, 58, 60–62, 65, 73, 80–86, 88, 99, 108–122, 124, 125, 129, 130
SWIR Short Wavelength InfraRed. 13, 14

TDNN Time Delay Neural Network. 41

UMRR Universal Multimode Range Resolution. 28
UTA Urban Traffic Assistant. 38, 39

VIS VISible. 1, 15, 17, 19, 20, 22, 24, 26, 32, 33, 46, 48, 49, 54–58, 61, 62, 69, 70, 72, 73, 75–77, 79–81, 84, 86, 98, 99, 101, 104–108, 110–122, 125, 129
VISIR VISible and InfraRed concatenated. 99, 101, 104, 106, 108, 110, 114, 116, 117, 121
VLWIR Very Long Wavelength InfraRed. 13
VRU Vulnerable Road Users. 25

List of Publications

International Conferences

- Besbes, B., **Apatean, A.**, Rogozan, A., Bensrhair, A., "Combining SURF-based Local and Global features for Road Obstacle Recognition in Far Infrared Images", accepted for publication at the *13th International IEEE Conference on Intelligent Transportation Systems - ITSC 2010*, to be held during September 19 - 22, 2010 at Madeira Island, Portugal

- **Apatean (Discant), A.**, Rusu, C., Rogozan, A., Bensrhair, A. "Visible-Infrared fusion in the frame of an obstacle recognition system", *IEEE International Conference on Automation, Quality and Testing Robotics, AQTR2010*, Cluj-Napoca, Romania, 28-30 May 2010

- **Apatean (Discant), A.**, Rogozan, A., Bensrhair, A., "SVM-based obstacle classification in visible and infrared images", *17th European Signal Processing Conference, (EUSIPCO 2009)*, August 24-28, 2009, Glasgow, Scotland, accepted for publication, http://www.eusipco2009.org/index.asp

- **Apatean (Discant), A.**, Rogozan, A., Bensrhair, A., "Information Fusion for Obstacle Recognition in Visible and Infrared Images", *International Symposium on Signals, Circuits and Systems (ISSCS2009)*, July 09-10, 2009, Iasi, Romania, http://scs.etc.tuiasi.ro/isscs2009/index.html

- **Apatean (Discant), A.**, Rogozan, A., Bensrhair, A., "Obstacle recognition using multiple kernel in visible and infrared images" *2009 IEEE Intelligent Vehicle Symposium (IV 2009)*, June 03-05, 2009, Xi'an, China, http://www.ieeeiv.net/

- **Apatean (Discant), A.**, Rogozan, A., Bensrhair, A., "Kernel and Feature Selection for Visible and Infrared based Obstacle Recognition", *11th International IEEE Conference on Intelligent Transportation Systems (ITSC2008)*, pp. 1130-1135, 12-15 Oct. 2008, Beijing, China, ISBN: 978-1-4244-2111-4

- **Apatean (Discant), A.**, Rogozan, A., Bensrhair, A., "Objects recognition in visible and infrared images from the road scene", *IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2008)*, Volume: 3, pp. 327-332, 22-25 May 2008, Cluj-Napoca, Romania, ISBN: 978-1-4244-2576-1AQTR

- **Apatean (Discant), A.**, Emerich, S., Lupu, E., Rogozan, A., Bensrhair, A., "Wavelets and Moments for Obstacle Classification", *The Third International Symposium on Communications, Control and Signal Processing ISSCSP2008*, 12-14 March, 2008, St. Julians, Malta

- **Discant, A.**, Emerich, S., Lupu, E., Rogozan, A., Bensrhair, A., "Ruttier Obstacle Classification by use of Fractional B-spline Wavelets and Moments", *Proceedings of The IEEE Region 8 Eurocon2007 Conference Computer as a Tool*, September 2007, Warsaw, Poland, **Best Poster Award** for this article.

- **Discant, A.**, Rogozan, A., Rusu, C., Bensrhair, A., "Sensors for Obstacle Detection - A Survey", *Proceedings of the 30th International Spring Seminar on Electronics Technology (ISSE2007)*, May 9-13, 2007, Cluj-Napoca, Romania, ISBN 978-973-713-174-4

National Journals

- **Apatean (Discant), A.**, Rogozan, A., Emerich, S., Bensrhair, A., "Wavelets as Features for Objects Recognition", *Acta Tehnica Napocensis*, vol 49 no. 2/2008, pp. 23-26, ISSN1221-6542

- **Discant, A.**, Rusu, C., Rogozan, A., Bensrhair, A., "Sensors for Obstacle Detection in Traffic Scene Situation", *Acta Tehnica Napocensis*, vol 48 no. 1/2007, pp. 29-34, ISSN1221-6542

ABSTRACT

To continue and improve the detection task which is in progress at INSA laboratory, we focused on the fusion of the information provided by visible and infrared cameras from the viewpoint of an Obstacle Recognition module, thus discriminating between vehicles, pedestrians, cyclists and background obstacles. Bimodal systems have been proposed to fuse the information at different levels: of features, SVM's kernels, or SVM's matching-scores. These were weighted according to the relative importance of the modality sensors to ensure the adaptation (fixed or dynamic) of the system to the environmental conditions. To evaluate the pertinence of the features, different features selection methods were tested by a KNN classifier, which was later replaced by a SVM. An operation of model search, performed by 10 folds cross-validation, provides the optimized kernel for the SVM. The results have proven that all bimodal VIS-IR systems are better than their corresponding monomodal ones.

Keywords: Fusion, Infrared cameras, Features extraction, Features selection, Support Vector Machine, Kernels, Matching-scores, Hyper-parameter optimization, Model search, 10 folds cross-validation.

RÉSUMÉ

Afin de poursuivre et d'améliorer la tâche de détection qui est en cours à l'INSA, nous nous sommes concentrés sur la fusion des informations visibles et infrarouges du point de vue de reconnaissance des obstacles, ainsi distinguer entre les véhicules, les piétons, les cyclistes et les obstacles de fond. Les systèmes bimodaux ont été proposées pour fusionner l'information à différents niveaux: des caractéristiques, des noyaux SVM, ou de scores SVM. Ils ont été pondérés selon l'importance relative des capteurs modalité pour assurer l'adaptation (fixe ou dynamique) du système aux conditions environnementales. Pour évaluer la pertinence des caractéristiques, différentes méthodes de sélection ont été testés par un PPV, qui fut plus tard remplacée par un SVM. Une opération de recherche de modèle, réalisée par 10 fois validation croisée, fournit le noyau optimisé pour SVM. Les résultats ont prouvé que tous les systèmes bimodale VIS-IR sont meilleurs que leurs correspondant monomodale.

Mot clés: Fusion, Caméras infrarouges, Extraction des caractéristiques, Sélection des caractéristiques, Séparateur a Vaste Marge, Noyau, Scores, Optimisation des hyper-paramèters, Recherche du modèle, 10 fois validation croisée.