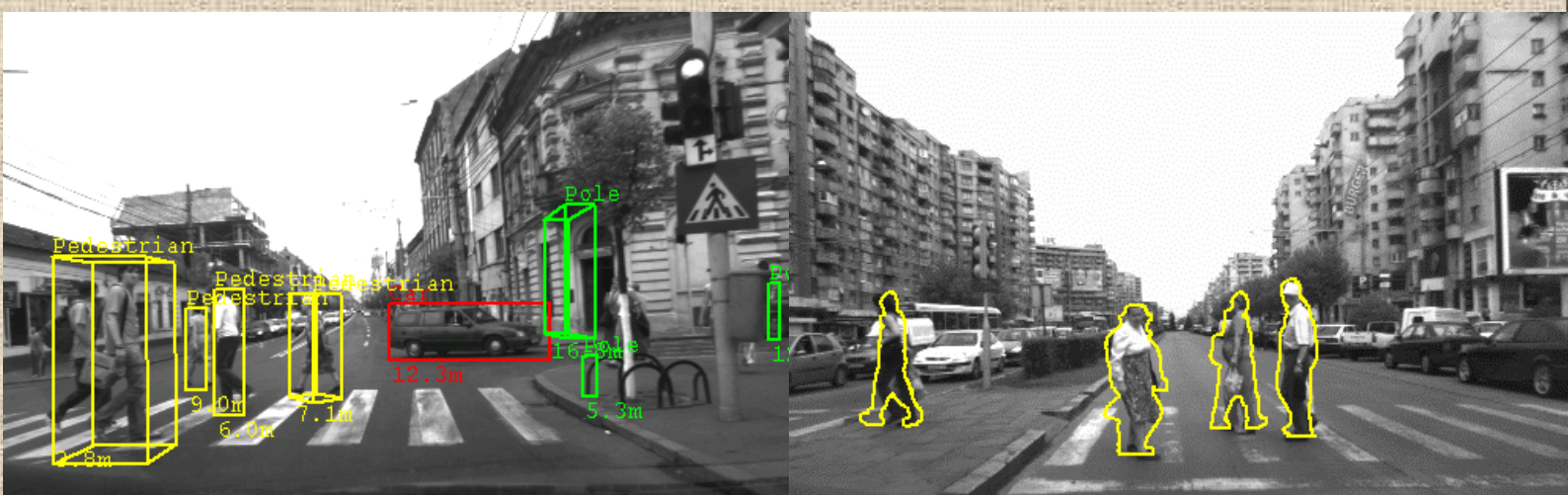


Ion GIOSAN

METODE TRADIȚIONALE DE PERCEPȚIE A SCENELOR DE TRAFIC



UTPRESS
Cluj-Napoca, 2020
ISBN 978-606-737-460-5

Ion GIOSAN

**METODE TRADIȚIONALE
DE PERCEPȚIE
A SCENELOR DE TRAFIC**



UTPRESS

Cluj-Napoca, 2020

ISBN 978-606-737-460-5



Editura U.T.PRESS
Str. Observatorului nr. 34
C.P. 42, O.P. 2, 400775 Cluj-Napoca
Tel.:0264-401.999
e-mail: utpress@biblio.utcluj.ro
<http://biblioteca.utcluj.ro/editura>

Director: Ing. Călin D. Câmpean

Recenzia: Conf.dr.ing. Mihai Negru
Ș.I.dr.ing. Raluca Brehar

Copyright © 2020 Editura U.T.PRESS
Reproducerea integrală sau parțială a textului sau ilustrațiilor din această carte este posibilă numai cu acordul prealabil scris al editurii U.T.PRESS.

ISBN 978-606-737-460-5

Cuprins

1. Introducere.....	5
2. Detecția obstacolelor din scenele de trafic.....	7
2.1. Detecția obstacolelor prin segmentarea imaginii în superpixeli.....	9
2.1.1. Componentele principale ale sistemului de detecție a obstacolelor	10
2.1.2. Extragerea superpixelilor pe baza intensităților din imagine	11
2.1.3. Extragerea trăsăturilor la nivelul fiecărui superpixel	13
2.1.4. Gruparea superpixelilor în obstacole.....	14
2.1.5. Validarea și rafinarea obstacolelor	16
2.1.6. Rezultate experimentale	19
2.2. Detecția obstacolelor prin analiza disparităților UV	23
2.2.1. Componentele principale ale sistemului de detecție a obstacolelor	23
2.2.2. Sistemul de stereoviziune	24
2.2.3. Spațiul disparităților <i>UV</i>	25
2.2.4. Extragerea suprafeței drumului	26
2.2.5. Etichetarea spațiului de disparitate	26
2.2.6. Delimitarea obstacolelor în scena de trafic	28
2.3. Detecția obstacolelor prin analiza vectorilor de mișcare relativă.....	31
2.3.1. Componentele principale ale sistemului de detecție a obstacolelor	31
2.3.2. Extragerea punctelor de interes	32
2.3.3. Calculul vectorilor de mișcare relativă și urmărirea punctelor de interes	33
2.3.4. Estimarea mișcării fundalului.....	34
2.3.5. Gruparea și validarea vectorilor de mișcare relativă în obstacole.....	36
2.3.6. Detecția coliziunii iminente cu un obstacol.....	38
2.3.7. Rezultate experimentale	40
2.4. Concluzii.....	42
3. Recunoașterea obstacolelor din scenele de trafic.....	43
3.1. Recunoașterea pietonilor folosind trăsături de contur și ierarhii de modele de contururi.....	45
3.1.1. Extragerea conturului pietonilor.....	45
3.1.2. Construirea ierarhiei cu modele de contururi de pietoni	49
3.1.3. Recunoașterea pietonilor pe baza trăsăturilor de contur.....	54
Arhitectura sistemului de recunoaștere a pietonilor	54

Recunoașterea pietonilor pe baza clasificării conturului.....	58
3.2. Recunoașterea claselor de obstacole de interes	61
3.2.1. Trăsături multiple pentru recunoașterea obstacolelor.....	61
Arhitectura sistemului de clasificare a obstacolelor	62
Extragerea trăsăturilor	63
Clasificarea obstacolelor pe baza trăsăturilor	67
Rezultate experimentale	68
3.2.2. Inferențe probabilistice pentru urmărirea rezultatelor clasificării	72
Arhitectura sistemului de urmărire a clasificării obstacolelor.....	73
Inferența Bayesiană	75
Urmărirea rezultatelor clasificării.....	76
Rezultate experimentale	77
3.3. Concluzii.....	82
Bibliografie.....	84

1. Introducere

Percepția scenelor de trafic văzute din interiorul unui autovehicul inteligent presupune în principal detecția și recunoașterea robustă și în timp real a tuturor participanților la trafic (obstacole) astfel încât să se poată lua ulterior deciziile de navigare corecte și la momentul oportun. Stereoviziunea presupune existența unui sistem de două camere video care achiziționează imagini pereche (stânga-dreapta) ale scenei de trafic și care ulterior sunt prelucrate cu scopul percepției mediului din preajma autovehiculului. În cadrul metodelor prezentate în această carte se utilizează un sistem de stereoviziune care achiziționează imagini în niveluri de gri, având un câmp vizual larg, cu un domeniu de detecție mediu în scenă și costuri relativ scăzute care încurajează montarea lui pe autovehicule în producții de serie. Sistemul stereo este conectat la un sistem de calcul, care preia imaginile și ulterior le procesează.

Modulele de percepție care detectează și recunosc obstacole din imagini ale scenelor de trafic sunt adesea întâlnite ca părți integrante în sisteme de asistență a conducătorului auto și în sisteme de navigare autonomă din vehiculele inteligente. Funcțiile de detecție și recunoaștere a obstacolelor trebuie să fie robuste indiferent de condițiile de trafic și vreme. Dacă pe autostradă scenele achiziționate sunt relativ simple conținând în principal obstacole ca și autovehicule, garduri și stâlpi, în scenariile de trafic urban acestea sunt mult mai complexe. Aici apar problemele legate de scenele foarte aglomerate în care ocluziile apar frecvent și fac problema detecției obstacolelor una foarte dificilă. Valorile primare care sunt utilizate pentru detecția obstacolelor utilizând stereoviziunea sunt disparitățile și punctele 3D rezultate în urma stereo-reconstrucției. Erorile și imposibilitatea de a realiza stereo-reconstrucția în unele zone de imagine duc la detecția imprecisă a obstacolelor sau chiar la nedetectarea acestora. Aceste probleme apar și din cauza condițiilor de iluminare a scenei: scene suprasaturate în lumină puternică sau opusul acestora scene în condiții de iluminare scăzută – noapte, seară, când numărul și calitatea punctelor 3D reconstruite scad drastic. În ceea ce privește recunoașterea obstacolelor, în scenariile de trafic urban apar cei mai vulnerabili participanți la trafic și anume pietonii. Problema recunoașterii pietonilor a atras un interes deosebit de ridicat din partea comunității științifice și a producătorilor naționali și internaționali de autovehicule, cu scopul creșterii siguranței acestora. Sistemele de asistare și de navigare autonomă trebuie să detecteze în avans situațiile potențial periculoase în care apar pietoni. De obicei, pietonii sunt recunoscuți în scenele de trafic, în mod tradițional, cu ajutorul clasificatoarelor de obstacole, pe baza trăsăturilor lor. Rata lor de recunoaștere trebuie să fie suficient de mare astfel încât sistemul de asistență să nu trimită semnale false șoferului în cazul prezenței altor obstacole și nici să nu rateze unii pietoni. Deși recunoașterea pietonilor este o problemă simplă pentru oameni, în cazul sistemelor de viziune artificială această problemă este una relativ dificilă. Acest lucru se datorează în special diferitelor ipostaze în care apar în trafic, îmbrăcămintei diferite și accesoriilor pe care le poartă. Și alte variabile precum variațiile fundalului scenelor, condițiile meteo, distanța față de camere și rezoluția lor, vibrațiile ego-vehiculului afectează recunoașterea nu numai a pietonilor dar și a obstacolelor în general. Metodele de clasificare sunt și ele afectate de

numărul claselor și de calitatea trăsăturilor folosite. De aceea numărul claselor de obstacole utilizate în procesul de recunoaștere nu trebuie să fie unul foarte mare și trăsăturile trebuie să fie cât mai discriminante.

Aspectul foarte important și motivant pentru sistemele de asistență a conducătorului auto și de navigare autonomă este acela că toate procesările trebuie să se facă într-un timp cât mai scăzut, pentru a permite execuția în timp real. Acest lucru este esențial întrucât, la viteze ridicate ale ego-vehiculului, sistemul trebuie să ofere un răspuns în timp util. Astfel, algoritmi de detecție și recunoaștere a obstacolelor trebuie să fie suficient de complecși pentru a rezolva problemele într-un mod robust dar și foarte eficienți pentru a oferi rezultatele în timp util. Aceste două cerințe, care se află una față de cealaltă într-un continuu compromis, trebuie satisfăcute fiecare într-o măsură cât mai mare.

În cadrul acestei cărți sunt prezentate câteva metode tradiționale care au fost dezvoltate, implementate și experimentate pentru detecția robustă a obstacolelor din scenarii de trafic urban utilizând în principal imagini de stereoviziune și apoi recunoașterea precisă a acestora cu un accent ridicat pe recunoașterea pietonilor întrucât sunt cei mai vulnerabili participanți la trafic. În cadrul detecției obstacolelor se fixează ca și obiectiv obținerea unei segmentări cât mai precise ale acestora într-un timp rezonabil pentru procesarea de timp real și care să poată fi folosite ulterior pentru recunoașterea lor. Sunt prezentați algoritmi noi de segmentare a obstacolelor, se îmbunătățesc metode existente de detecție a obstacolelor și se analizează comparativ rezultatele cu scopul evidențierii avantajelor și dezavantajelor fiecăreia. După obținerea unei segmentări de calitate a obstacolelor, se dorește recunoașterea acestora. În scenele de trafic urban se identifică trei clase importante de obstacole: mașini, stâlpi/copaci și pietoni. Recunoașterea obstacolelor are ca și obiectiv clasificarea obstacolelor detectate anterior în patru clase: cele trei specificate și încă o clasă cu alte obstacole. Accentul se pune pe recunoașterea pietonilor. Recunoașterea trebuie să se facă într-un timp foarte scurt de procesare pentru a permite sistemului să poată rula în timp real. Astfel se propun și se extrag trăsături noi specifice pietonilor, se construiesc și se organizează modele de pietoni, se definesc caracteristici ale obstacolelor, se dezvoltă metode noi și se îmbunătățesc metode existente în literatura de specialitate pentru clasificarea pietonilor și a obstacolelor în general. Rezultatele lor sunt analizate comparativ pentru a evidenția avantajele și dezavantajele fiecărei metode.

2. Detecția obstacolelor din scenele de trafic

În literatura de specialitate sunt folosite în principal trei tehnici pentru detecția obstacolelor din imagini monoculare. Prima tehnică este bazată pe extragerea și analiza trăsăturilor de culoare și a trăsăturilor texturale din imagini, cu scopul principal de a determina zona de drum din fața ego-vehiculului și de a putea face discriminare între suprafața drumului și zonele care ar putea aparține obstacolelor [1-3]. Această abordare este dificil de utilizat în scenarii în care aspectul (culoarea, textura) zonei drumului variază. De asemenea este dificil de separat zonele cu obstacole de zonele de fundal, acestea din urmă putând varia foarte mult în secvențe de imagini. A doua tehnică este bazată pe estimarea hărților de profunzime cu scopul definirii distanțelor punctelor din scenă față de sistemul de viziune [4-7]. Acestea sunt calculate cu ajutorul trăsăturilor 2D extrase pe regiuni. Din păcate aceste estimări nu au o acuratețe suficient de mare pentru a putea realiza o segmentare și o detecție bună a obstacolelor. A treia tehnică utilizează trăsături de flux optic cu scopul determinării zonelor aparținătoare obstacolelor aflate în mișcare [8-11]. În acest caz apare problema legată de erorile în determinarea vectorilor de mișcare – aceștia fiind calculați pe baza potrivirilor punctelor cheie între cadre succesive de imagini. Există și metode mixte care combină trăsăturile de apariție vizuală cu trăsături de mișcare [12-14]. În general toate tehnicile de detecție a obstacolelor din viziune monoculară nu au o acuratețe și robustețe ridicată, iar în cazurile în care acestea obțin rezultate promițătoare, timpul de execuție este foarte ridicat, fiind departe de procesarea în timp real, deosebit necesară în sistemele de asistență a condusului sau de navigare autonomă la viteze mari.

Stereoviziunea încearcă să rezolve problemele care sunt întâlnite în detecția obstacolelor din viziune monoculară. Există două mari direcții care sunt utilizate în detecția obstacolelor din imagini stereo: prima utilizează trăsături de apariție vizuală a scenei, iar cea de-a doua trăsături de mișcare a elementelor scenei. Trăsăturile de apariție vizuală sunt extrase, iar pe baza lor se determină zonele care pot candida ca fiind aparținătoare unor obstacole. Acestea sunt de obicei validate pe baza unor constrângeri impuse pe caracteristicile 3D ale obstacolelor. Informațiile 3D pot fi calculate mult mai precis cu ajutorul stereoviziunii [15-19]. Metoda transformării geometrice inverse (*IPM*) este utilizată pentru a detecta suprafața drumului și marcajele acestuia cu scopul ulterior de a determina obstacolele [20-22]. Trăsăturile de mișcare sunt foarte frecvent întâlnite în detecția obstacolelor. Acestea sunt extrase de obicei pe baza fluxului optic calculat între cadre succesive de imagini, permițând identificarea elementelor din scenă care au o mișcare relativă față de sistemul de camere stereo [23-28]. Ulterior se poate determina dacă aceste elemente aparțin sau nu obstacolelor. Problemele care pot să apară sunt cauzate de erorile de calcul al fluxului optic mai ales în cazurile în care există ocluzii ale elementelor din scenă. Astfel, valorile eronate ale vitezelor de mișcare pot duce la detecții greșite a unor obstacole sau la cazuri mai grave în care acestea nu sunt detectate.

O tehnică frecvent întâlnită pentru detecția obstacolelor este cea bazată pe analiza spațiului disparităților și a calculării hărților de ocupare a scenei [29-33] [34-37]. Acestea separă suprafața drumului de celelalte elemente din scenă și apoi încearcă gruparea lor în obstacole individuale. Altă tehnică este cea bazată pe calculul punctelor 3D reconstruite

și gruparea lor în obstacole [38-46]. Problemele care apar în ultimele două tehnici sunt cauzate de erorile de stereo-reconstrucție generate în urma unor stereo-corespondențe eronate a punctelor între imaginile stânga și dreapta dar și de zonele în care nu s-a putut realiza stereo-corespondența, neexistând puncte 3D reconstruite. Metodele multi-rezoluție sunt utilizate pentru a mări viteza de detecție a obstacolelor prin validarea zonelor componente la diferite rezoluții [47-49]. Există și metode care fuzionează informațiile stereo cu măsurători de profunzime mult mai precise obținute de la un senzor LASER sau LIDAR. Acestea îmbunătățesc calitatea detecției obstacolelor dar cresc costurile unor astfel de echipamente suplimentare [50, 51].

Utilizând stereoviziunea se pot obține rezultate bune ale detecției obstacolelor, în timp real de procesare și cu costuri relativ scăzute ale camerelor. Metodele de detecție a obstacolelor din imagini stereo sunt net superioare celor ce utilizează viziunea monoculară. Totuși, majoritatea au încă probleme legate de segmentarea precisă a obstacolelor în special la distanțe mari și în scenarii complexe de trafic. Într-un proces ulterior de înțelegere a scenelor de trafic, segmentarea imprecisă a obstacolelor poate duce la o clasificare greșită a lor.

În continuare sunt descrise trei metode pentru detecția obstacolelor în scene de trafic urban, iar în final se prezintă concluziile cu evidențierea avantajelor și dezavantajelor fiecăreia și fezabilitatea utilizării lor în sisteme de asistare a conducătorului auto și de conducere autonomă.

În secțiunea 2.1 este prezentată metoda de detecție a obstacolelor prin segmentarea în superpixeli a imaginii scenei de trafic. Se definește o paradigmă care integrează informația de intensitate, profunzime și de mișcare pe superpixeli cu scopul segmentării obstacolelor în scene de trafic. Se proiectează o strategie de extragere și definire a superpixelilor: calculul acestora pe baza informației de intensitate și apoi extragerea de trăsături multiple (profunzime, flux optic) pentru caracterizarea fiecărui superpixel. Se concepe o metodă de grupare a superpixelilor în obstacole în doi pași: întâi, informația de profunzime este utilizată pentru definirea obstacolelor prin setul de superpixeli aparținători, iar apoi vectorii de flux optic rafinează grupările de superpixeli în obstacole pe baza mișcării acestora. Metoda descrisă este evaluată cantitativ și calitativ și comparată cu alte tehnici de detecție a obstacolelor.

În secțiunea 2.2 este prezentată metoda de detecție a obstacolelor prin calculul și analiza spațiului de disparități UV . Se definește o metodă de detecție a planului drumului pe baza spațiului de disparitate V și detecția obstacolelor de pe drum pe baza analizei spațiului de disparitate U prin definirea unui algoritm pentru etichetarea acestora.

În secțiunea 2.3 este prezentată metoda de detecție a obstacolelor prin analiza vectorilor de mișcare relativă. Se definește astfel o metodă de detecție a obstacolelor prin: calculul fluxului optic, estimarea mișcării generale a fundalului din analiza cadrelor succesive, eliminarea fundalului, gruparea vectorilor de mișcare în obstacole pe baza locației, modulului și unghiului și validarea obstacolelor obținute.

2.1. Detecția obstacolelor prin segmentarea imaginii în superpixeli

Metodele de detecție a obstacolelor care folosesc informații provenite de la sisteme de stereoviziune au la bază calculul stereo-corespondențelor dintre imaginile stânga și dreapta achiziționate. Utilizând aceste stereo-corespondențe se calculează hărțile de disparitate sau hărțile de profunzime. Punctele 3D stereo-reconstruite stau la baza numeroaselor metode de detecție a obstacolelor. Pentru obținerea de performanțe ridicate în detecția obstacolelor trebuie ca aceste date primare să fie de înaltă calitate și timpul de procesare necesar lor să fie cât mai scăzut.

În cadrul acestei metode de detecție a obstacolelor s-a folosit metoda SORT-SGM (*stereo-reconstruction with sub-pixel level optimized semiglobal matching*) pentru calculul semi-global al stereo-corespondențelor, într-o implementare pe GPU, care oferă o calitate ridicată a hărții de profunzime într-un timp de execuție foarte scăzut [52]. Harta profunzimilor oferită de această metodă este mult mai densă și cu valori mai precise decât cea care se obține prin tehnica locală de realizare a stereo-corespondențelor implementată hardware (TYZX) [53]. În Figura 2.1 este prezentat un exemplu cu puncte reconstruite 3D calculate prin ambele metode și vizualizate comparativ. Imaginea scenei este de rezoluție 512x383 pixeli în niveluri de gri (Figura 2.1a). Reconstrucția stereo oferită de TYZX dă un număr de aproximativ 61000 de puncte (Figura 2.1b) pe când cea realizată cu metoda SORT-SGM are un număr de aproximativ 77000 de puncte (Figura 2.1c). În general se observă o îmbunătățire de cca 25% în ceea ce privește numărul de puncte stereo-reconstruite, constatându-se și o creștere a acurateții acestora.

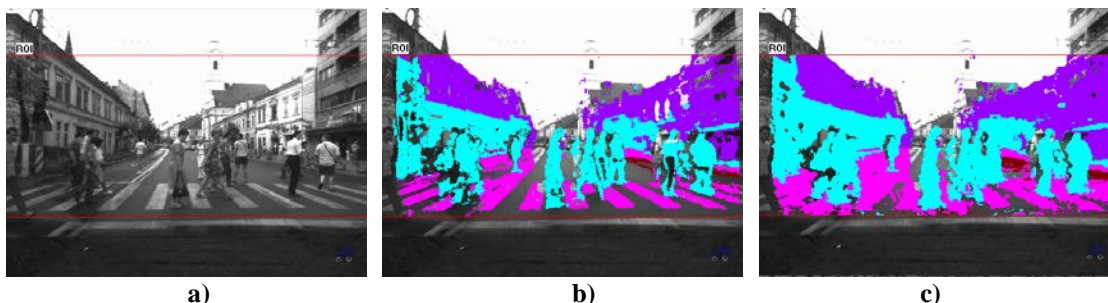


Figura 2.1. a) Imagine stângă achiziționată cu sistemul stereo; b) puncte reconstruite stereo cu TYZX; c) puncte stereo-reconstruite stereo cu SORT-SGM pe GPU

Pentru a putea detecta obstacolele trebuie în primul rând să existe o separație clară dintre punctele stereo-reconstruite ce aparțin suprafeței drumului și celelalte puncte din scenă. În general multe metode consideră drumul ca fiind o suprafață plană sau împarte suprafața acestuia în secțiuni care sunt approximate cu suprafețe plane. O astfel de metodă nu necesită extragerea de marcaje rutiere și ia în considerare doar informațiile legate de textură, umbre, muchii, etc. [29]. Metoda definită este bazată pe hărți de elevație cu scopul realizării acestei delimitări [42]. O suprafață modelată polinomial de grad doi este potrivită inițial la regiunea din fața ego-vehiculului. Procesul de potrivire este urmat de un proces de creștere a regiunilor, ghidat de modelul de incertitudine a punctelor 3D, cu rolul rafinării suprafeței modelate.

Superpixelii constau în grupări coerente de pixeli pe baza trăsăturilor locale extrase din imagini (Figura 2.2). Avantajul major al lor constă în faptul că reduc

semnificativ timpul de procesare și complexitatea sarcinilor cum ar fi estimarea profunzimilor, segmentarea imaginilor, estimarea suprafețelor obstacolelor și localizarea lor în scenă. Pentru detecția superpixelilor este prezentată o variantă modificată a algoritmului SLIC de segmentare a superpixelilor [54] care oferă rezultate foarte bune pe imaginile în niveluri de gri.

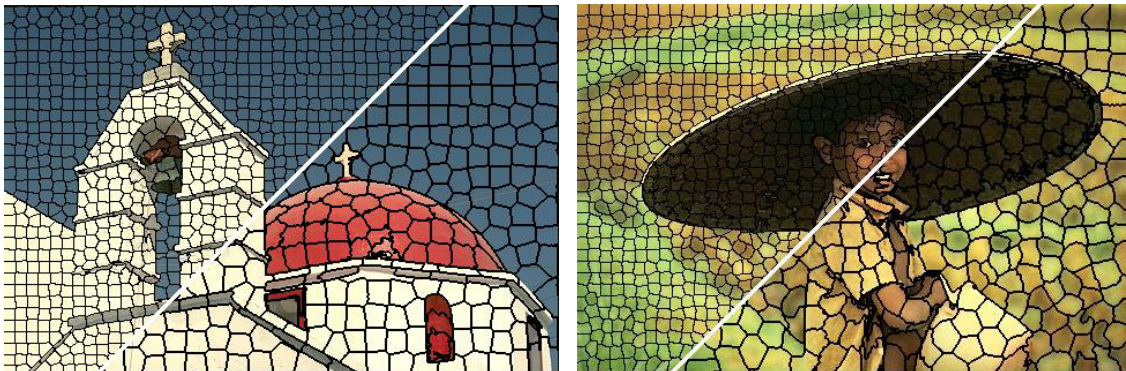


Figura 2.2. Exemple de segmentări cu superpixeli ale imaginilor color [54]

2.1.1. Componentele principale ale sistemului de detecție a obstacolelor

Arhitectura sistemului pentru detecția obstacolelor pe baza superpixelilor împreună cu modulele componente și interfețele lor (intrări și ieșiri) este prezentată în Figura 2.3.

Imaginile achiziționate de sistemul de stereoviziune sunt în niveluri de gri având rezoluția de 512x383 pixeli. Reconstrucția stereo este realizată prin algoritmul SORT-SGM precizat anterior și implementat pe un procesor grafic (GPU). Acesta primește ca și intrări imaginile de intensitate provenite de la camera stânga și dreapta, după ce în prealabil au fost eliminate distorsiunile lentilelor și s-a aplicat procedeul de rectificare, rezultând o hartă de profunzime densă și cu acuratețe ridicată, ceea ce este foarte important pentru procesările ulterioare. Această hartă memorează, pentru fiecare punct reconstruit stereo, distanța lui față de sistemul de camere. Pe același procesor grafic este implementat și fluxul optic, considerând ca și trăsături cele de tip *good-features-to-track* [55] într-o abordare piramidală [56] Lucas-Kanade [57].

Superpixelii SLIC sunt extrași pe baza informației de intensitate fiind utilizați apoi într-un modul de extragere a trăsăturilor care calculează trăsături specifice de intensitate, profunzime și de mișcare pentru fiecare superpixel. Pe baza adiacenței și asemănării dintre superpixeli se definește un algoritm de etichetare și grupare a lor în obstacole diferite. Suprafața drumului se calculează pe baza analizei hărților de disparitate [42].

Modulul de validare a obstacolelor are rolul de a filtra grupările de superpixeli, care nu au cum să reprezinte un obstacol valid, pe baza dimensiunilor acestora și considerând faptul că orice obstacol trebuie să aibă contact cu suprafața drumului. În final se definește și se aplică o procedură de rafinare a obstacolelor pe baza unei metode speciale cu operații morfologice de închidere și pe baza vectorilor de mișcare. Aceștia din urmă sunt deosebit de importanți, având puterea de a separa eventualele grupări eronate

de obstacole în entități de sine stătătoare. Toate aceste module sunt prezentate în detaliu în secțiunile următoare.

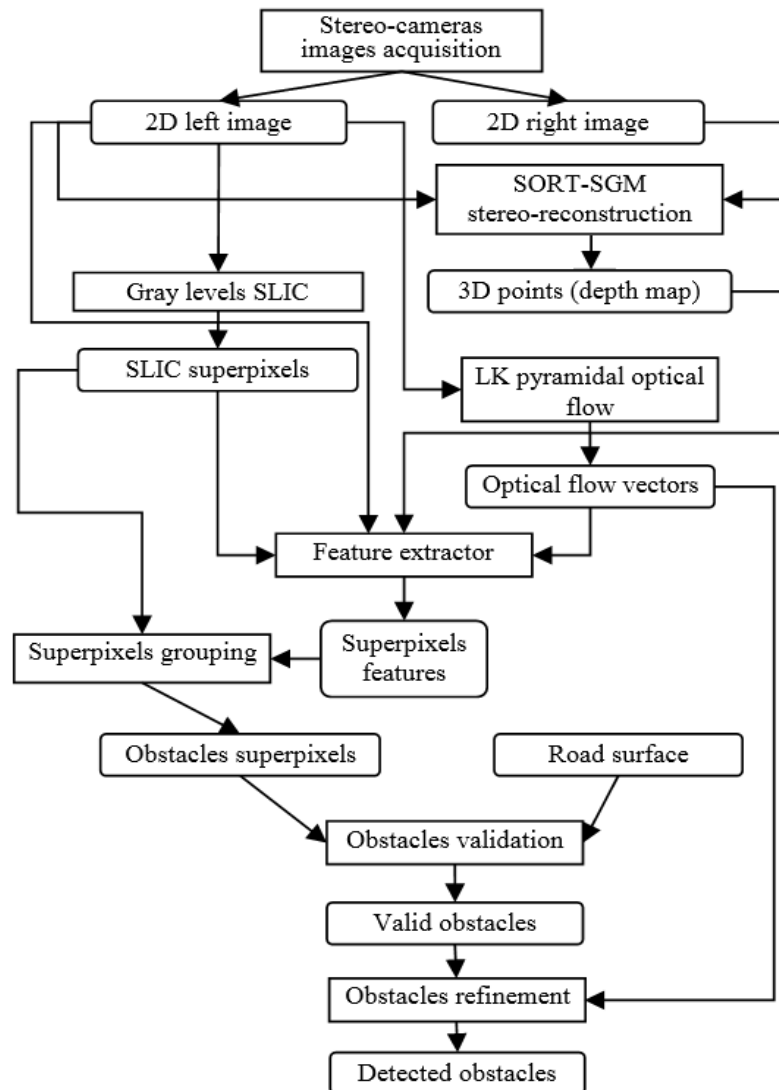


Figura 2.3. Schema bloc a sistemului de detecție a obstacolelor pe baza superpixelilor

2.1.2. Extragerea superpixelilor pe baza intensităților din imagine

Primul pas înspre detecția obstacolelor constă în extragerea superpixelilor sau segmentarea imaginii într-un set de superpixeli. Deoarece imaginile achiziționate sunt în niveluri de gri, s-a adaptat algoritmul SLIC pentru segmentarea acestor tipuri de imagini. S-a definit regiunea de interes (ROI) în care pot exista obstacole pe imaginea 2D de intensitate, procesările efectuându-se doar în această zonă cu scopul reducerii timpului de procesare. Poziția și dimensiunile acesteia sunt definite în imagine de dreptunghiul cu extremitățile ($left=0$, $top=100$, $right=512$, $bottom=320$) (Figura 2.4).

Se consideră un număr $N=2000$ de puncte distribuite uniform în regiunea de interes pe imaginea de intensitate și care constituie centrul fiecărui superpixel. Aria inițială (suprafața rectangulară) a fiecărui superpixel este ns și este dată de ecuația (2.1).

$$ns = \frac{\text{width}(ROI) \cdot \text{height}(ROI)}{N} \quad (2.1)$$

Inițial, fiecare astfel de centru definește un superpixel care are în jurul său un număr ns de pixeli aparținători. Aceste grupări de pixeli sunt notate cu $C_i, i=1 \dots N$, având un număr total de N superpixeli în imagine. Pentru a obține o segmentare bună trebuie ca aceste centre să fie deplasate în imagine într-o vecinătate în care valoarea gradientului este minimă. Dimensiunile ferestrei de căutare este de 3×3 pixeli în jurul locației inițiale a fiecărui centru. Pentru calculul derivatelor în imagine pe direcțiile ortogonale x și y s-au folosit nucleele de convoluție Prewitt de dimensiuni 3×3 pixeli. Modulul gradientului a fost calculat ca suma valorilor absolute a celor două derivate cu scopul creșterii vitezei de procesare.

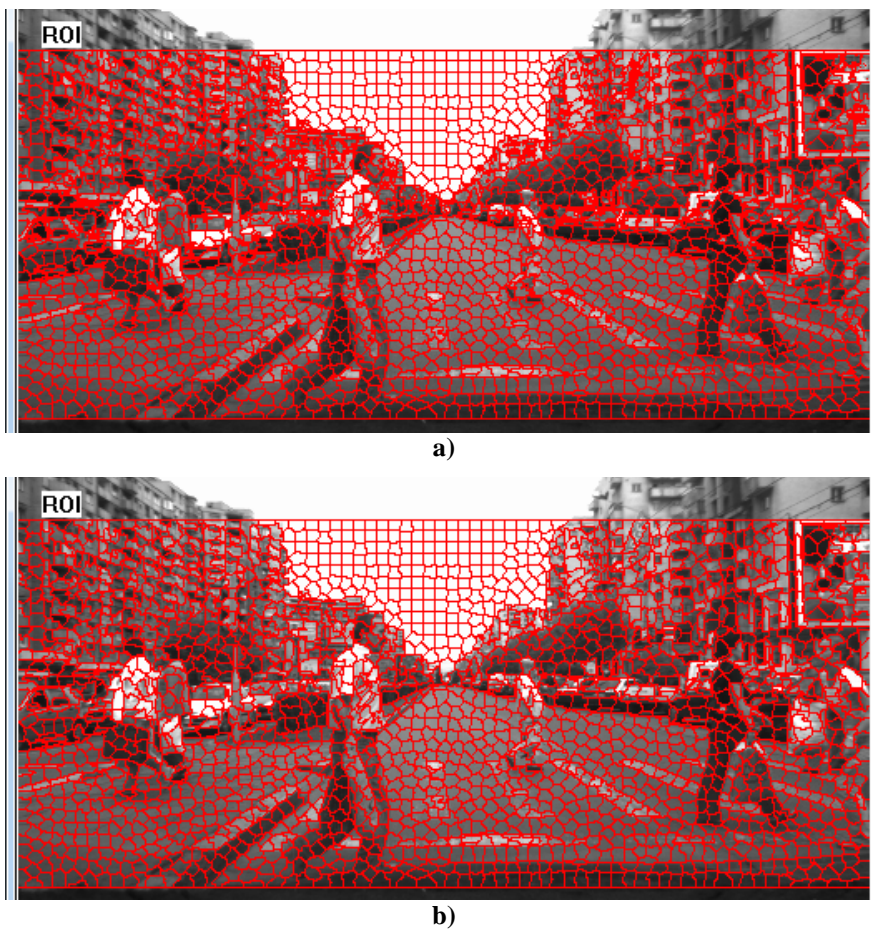


Figura 2.4. Celule-superpixeli: a) inclusiv celule rezultate de dimensiuni foarte mici; b) rafinare și rezultatul final obținut

Un algoritm de grupare *k-means* [58] este utilizat pentru atașarea fiecărui pixel din ROI la o anumită grupare (superpixel). Acesta este un proces iterativ. S-a considerat că un număr de 10 iterații este suficient pentru a obține o convergență care să ofere rezultate foarte bune. Centrele grupărilor sunt definite de media pe fiecare dintre cele trei componente: poziția dată de coordonatele x și y și nivelul de gri (intensitatea) *gray* corespunzător pixelului.

S-a definit o metrică de distanță $d(P, C_k)$ de la fiecare punct P din interiorul regiunii de interes la o grupare C_k pe baza locației pixelului (x, y) și intensității acestuia $gray$ prin formula (2.2). În cadrul acesteia, parametrul ng reprezintă un factor de ponderare, fiind considerat $ng=50$.

$$d(P, C_k) = \frac{|\text{gray}(P) - \text{gray}(C_k)|}{ng} + \frac{|x(P) - x(C_k)| + |y(P) - y(C_k)|}{2 \cdot ns} \quad (2.2)$$

În urma procesului de grupare pot apărea unii superpixeli care au o suprafață foarte mică (Figura 2.4a). Aceștia pot fi eliminați prin unificarea lor cu un vecin de dimensiuni mai mari. S-a considerat ca această unificare să fie făcută cu acel vecin care are suprafața cea mai mare. Un superpixel este introdus în procesul de unificare dacă suprafața lui este mai mică decât $\frac{1}{4} * ns$.

2.1.3. Extragerea trăsăturilor la nivelul fiecărui superpixel

La nivelul fiecărui superpixel determinat anterior, s-a calculat un set de trăsături F . Aria A a fiecărui superpixel reprezintă numărul total de puncte conținut de acesta. Pe toată suprafața unui superpixel sunt cunoscute: intensitatea $gray$ la nivelul fiecărui punct, pozițiile (X, Y, Z) în sistemul de coordonate al lumii pentru punctele reconstruite stereo și vectorii de mișcare OF calculați și descriși în sistemul de coordonate 2D de deplasamentele (p, q) .

$$F = (MI, MD, MedD, MedH, MedX, CD, MMagOF, MAngOF, COF) \quad (2.3)$$

Vectorul de trăsături F este descris în formula (2.3) și este caracterizat de MI media intensităților pixelilor componenți, MD media valorilor profunzimilor punctelor reconstruite, $MedD$ valoarea mediană a profunzimilor (Z), $MedH$ valoarea mediană a înălțimilor (Y), $MedX$ valoarea mediană a coordonatelor punctelor (X), CD densitatea punctelor 3D – formulele (2.4), $MMagOF$, $MAngOF$ care definesc modulul și unghiul vectorului mediu de mișcare al superpixelului și COF care arată densitatea vectorilor de mișcare din fluxul optic – formula(2.5).

$$MI = \frac{\sum_{P \in A} \text{gray}(P)}{\|A\|}, MD = \frac{\sum_{P \in RecPoints} Z(P)}{\|RecPoints\|}, CD = \frac{\|RecPoints\|}{\|A\|} \quad (2.4)$$

$$MedD = \text{median}_{P \in RecPoints}(Z(P)), MedH = \text{median}_{P \in RecPoints}(Y(P)), MedX = \text{median}_{P \in RecPoints}(X(P))$$

$$MMagOF = \sqrt{MMagOF_p^2 + MMagOF_q^2}, MAngOF = \arctg \frac{MMagOF_q}{MMagOF_p} \quad (2.5)$$

$$MMagOF_p = \frac{\sum_{P \in OF} p(P)}{\|OF\|}, MMagOF_q = \frac{\sum_{P \in OF} q(P)}{\|OF\|}, COF = \frac{\|OF\|}{\|A\|}$$

2.1.4. Gruparea superpixelilor în obstacole

Gruparea superpixelilor în obstacole se realizează pe baza asemărilor dintre vectorii de trăsături F corespunzători acestora. Obiectivul primar este de a clasifica fiecare superpixel în una din următoarele trei clase: superpixel de pe suprafața drumului, superpixel de obstacol, superpixel în afara zonei de mișcare a ego-vehiculului. Pe baza hărții de elevație și a indicatorilor punctelor reconstruite stereo (punct de pe suprafața drumului, punct deasupra suprafeței drumului), s-a calculat gradul de acoperire al fiecărui superpixel cu puncte de pe suprafața drumului conform formulei (2.6). Orice superpixel este considerat ca fiind aparținător suprafeței drumului dacă sunt îndeplinite două condiții: $Road > 0,25$ și $CD > 0,30$ (Figura 2.5).

$$Road = \frac{\sum_{P \in RecPoints} IsRoadPoint(P)}{\|RecPoints\|} \quad (2.6)$$

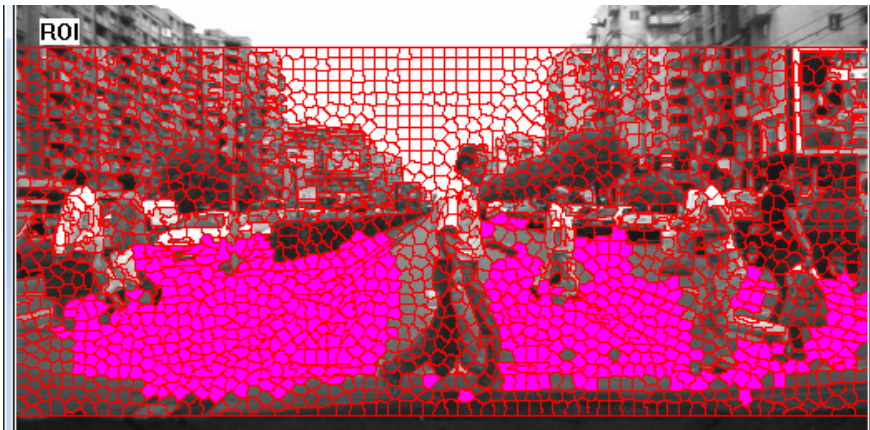


Figura 2.5. Superpixeli pe suprafața drumului (culoare purpurie)

S-a definit regiunea de interes 3D ca fiind volumul conținut de un paralelipiped dreptunghic delimitat de negarea inegalităților legate de $MedD$, $MedX$, $MedH$ din setul de condiții (2.7). Un superpixel este considerat ca fiind în afara zonei de interes 3D, dacă cel puțin una din următoarele condiții este adevărată:

$$\left\{ \begin{array}{l} CD = 0, \\ MedD > 40000 \text{ mm}, \\ MedH > 3500 \text{ mm}, \\ MedX < -10000 \text{ mm}, MedX > 10000 \text{ mm} \end{array} \right. \quad (2.7)$$

Acei superpixeli care nu sunt aparținători suprafeței drumului și sunt în interiorul regiunii de interes 3D sunt considerați ca fiind superpixeli de obstacole. S-a definit un algoritm de etichetare bazat pe vecinătatea superpixelilor și asemănarea trăsăturilor lor. Inițial nici un superpixel nu este etichetat. Matricea cu superpixeli este parcursă și, de fiecare dată când se întâlnește un superpixel neetichetat, se generează o nouă etichetă de obstacol și i se asignează acestuia. Cu ajutorul unei parcurgeri în lățime a grafurilor se propagă această etichetă la vecinii superpixelului, continuându-se până când nu se mai

întâlnesc superpixeli asemănători neetichetați. Rezultatul etichetării constă într-un set de grupări de superpixeli, fiecare astfel de grupare reprezentând un obstacol candidat. În secțiunea următoare s-a descris metodologia pentru validarea fiecărui obstacol candidat și rafinarea obstacolelor validate.

Un superpixel neetichetat T este considerat ca fiind asemănător cu un superpixel S care face deja parte dintr-un obstacol candidat, dacă următoarele două condiții sunt satisfăcute:

$$\begin{cases} CD_T > 0.55 \\ |MedD_S - MedD_T| < Thr(\min(MedD_S, MedD_T)) \end{cases} \quad (2.8)$$

Funcția $Thr(d)$ definește un prag variabil cu profunzimea d . Acest lucru este necesar datorită faptului că odată cu creșterea distanței, punctele reconstruite stereo sunt mai rarefiate și apar erori mai mari în determinarea poziției exacte a acestora în scenă. Astfel funcția prag $Thr(d)$ trebuie să aibă valori mai scăzute pentru puncte relativ apropiate și valori mai ridicate pentru puncte aflate la distanțe d mai mari. S-a ales empiric o astfel de funcție care dă rezultate bune pentru segmentarea obstacolelor. Expresia matematică este descrisă de formula (2.9) și graficul corespunzător acesteia este prezentat în Figura 2.6.

$$Thr(d) = \alpha \left(1 + \log^\gamma \left(1 + \frac{d}{\beta} \right) \right) \quad (2.9)$$

unde : $\alpha = 300 \text{ mm}$, $\beta = 2000 \text{ mm}$, $\gamma = 8$

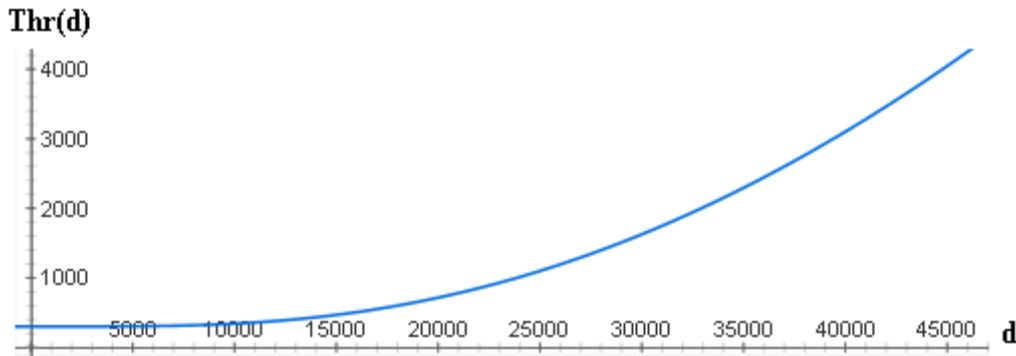


Figura 2.6. Graficul funcției $Thr(d)$ cu parametrii specificați

Etichetarea superpixelilor cu algoritmul și funcția prag definită obține trei categorii de superpixeli: superpixeli care aparțin suprafeței drumului, sunt în afara regiunii de interes 3D sau aparțin diferitelor obstacole. Rezultatele sunt prezentate în Figura 2.7. Fiecare grup de superpixeli este desenat cu o culoare diferită aleasă în mod aleatoriu.

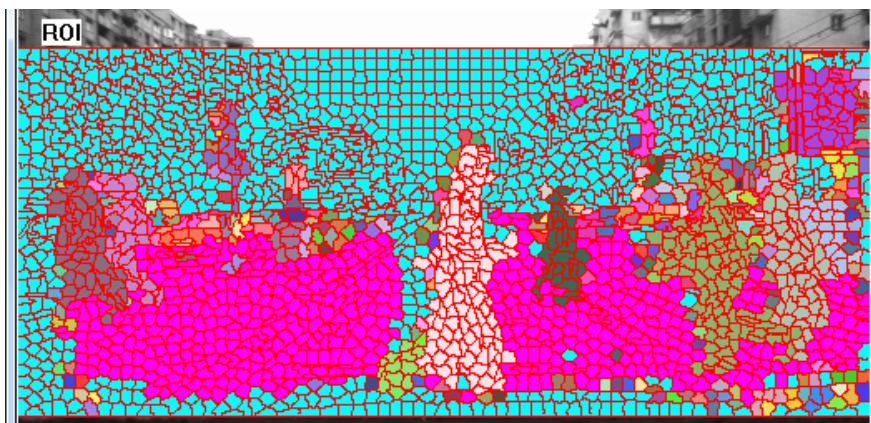


Figura 2.7. Grupări de superpixeli (suprafața drumului – culoare purpurie; zone în afara regiunii de interes – culoare turcoaz; obstacole candidate – fiecare cu culoare aleasă aleatoriu)

2.1.5. Validarea și rafinarea obstacolelor

Grupurile de superpixeli reprezentând obstacole candidate trebuie validate pentru a determina obstacolele reale care sunt în fața ego-vehiculului iar apoi acestea trebuie rafinate pentru a le defini forma lor finală. În Figura 2.7 se observă apariția unor grupări de superpixeli care reprezintă zgomote și nu pot reprezenta obstacole reale. Acestea sunt caracterizate de un număr relativ scăzut de superpixeli componenți, de faptul că nu se află pe suprafața drumului sau sunt determinate de mici denivelări de pe suprafața drumului. Astfel de obstacole trebuie eliminate. Un obstacol candidat este valid dacă setul de condiții (2.10) sunt satisfăcute. În cadrul acestora MY reprezintă media valorilor $MedY$ pentru superpixelii componenți ai obstacolului respectiv – formula (2.11). În Figura 2.8 este prezentată validarea obstacolelor din Figura 2.7. Se poate observa un număr de 8 obstacole valide.

$$\left\{ \begin{array}{l} \|ObstacleSuperpixels\| > 5 \\ \|RoadSuperpixels\|_{\text{neighbors of obstacle}} \geq 1 \\ MY > 300 \text{ mm}, \end{array} \right. \quad (2.10)$$

$$MY = \frac{\sum_{S \in ObstacleSuperpixels} MedY(S)}{\|ObstacleSuperpixels\|} \quad (2.11)$$

Există numeroase situații când pe suprafața obstacolelor apar mici discontinuități de superpixeli. Acesta este rezultatul calculării eronate a trăsăturilor din cauza măsurătorilor greșite a profunzimilor punctelor reconstruite stereo (Figura 2.9a). Aceste mici goluri sunt umplute printr-o operație morfologică specială de închidere. Astfel, se numără superpixelii vecini ai fiecărui gol care aparțin deja unui obstacol valid. Apoi superpixelul corespunzător golului este asignat aceluiași obstacol care are numărul maxim de superpixeli din vecinătate. Asignarea are loc numai dacă aceasta valoare maximă este cel puțin egală sau mai mare decât 4.

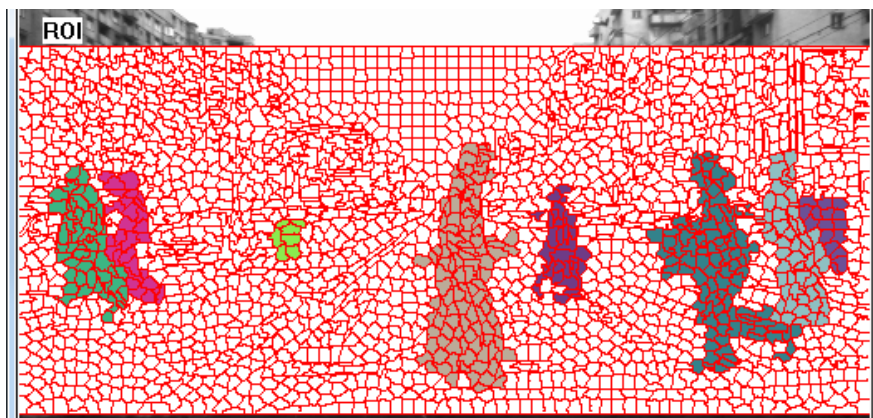
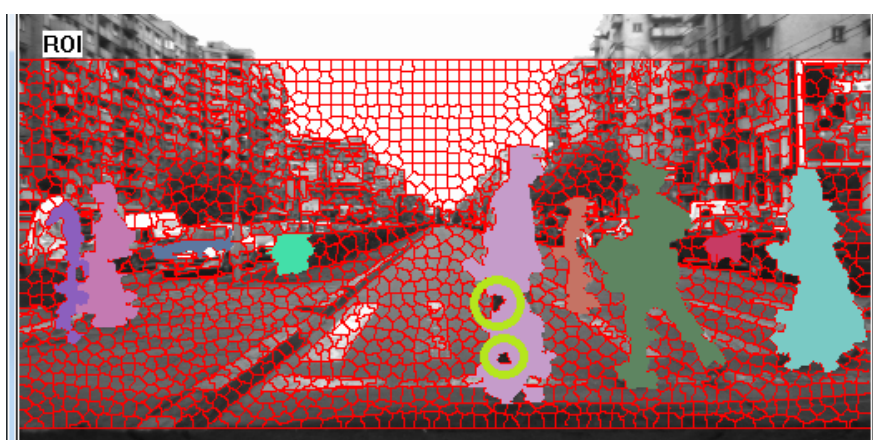
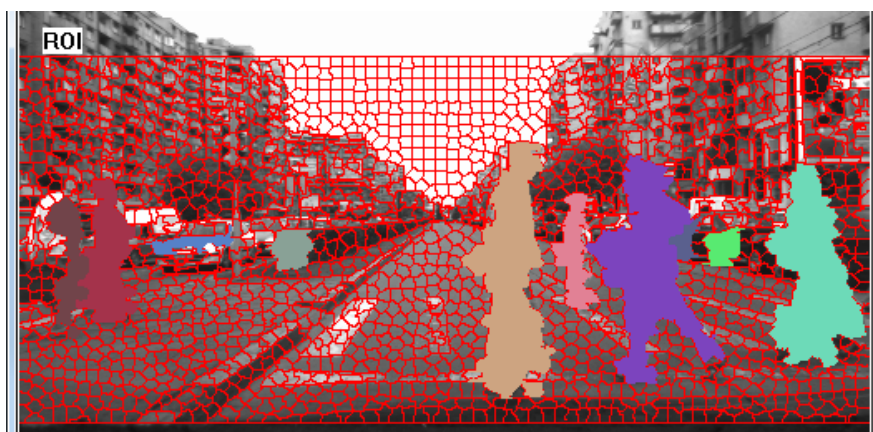


Figura 2.8. Obstacole valide (fiecare obstacol cu culoare aleasă aleatoriu; fundalul cu culoare albă)



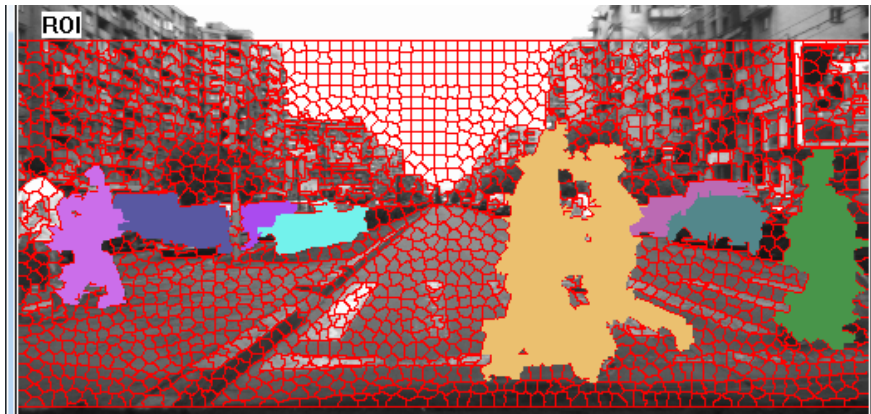
a)



b)

Figura 2.9. Obstacole valide: a) cu discontinuități interioare – marcate cu culoarea verde; b) după aplicarea operației morfologice speciale de închidere

Rafinarea finală constă în separarea grupărilor eronate de obstacole multiple în obstacole individuale. Această grupare multiplă eronată poate apărea când distanța dintre obstacole este foarte mică (Figura 2.10a). Procedura de rafinare are la bază utilizarea informației de flux optic, separația fiind posibilă numai în cazurile în care se poate observa o diferență majoră la nivelul direcțiilor vectorilor de mișcare (Figura 2.10b).



a)



b)

Figura 2.10. a) Obstacole multiple grupate eronat într-un singur obstacol; b) Vectori de mișcare (flux optic) – direcții opuse pentru cele două obstacole din centrul imaginii

Fiecare vector de mișcare OF este descompus pe cele două direcții ortogonale x și y . Modulul pe direcția orizontală este p , iar pe direcția verticală este q . Pe baza ecuației (2.5) se determină media modulelor și a orientărilor vectorilor fluxului optic în cadrul fiecărui superpixel. Se calculează o histogramă a orientării vectorilor de mișcare la nivelul fiecărui obstacol luând în considerare valorile medii calculate anterior la nivelul superpixelilor componente. Domeniul orientărilor este 0-360 grade, fiind divizat în 30 de acumuloare de dimensiuni identice (12 grade fiecare). Fiecare superpixel votează cu un vot în acumulatorul corespunzător unghiului vectorului de mișcare mediu OF . Se analizează histograma și dacă există mai mult de un vârf de maxim local atunci este cazul în care există mai multe obstacole care trebuie separate.

Figura 2.11 prezintă histograma orientărilor vectorilor de mișcare pentru obstacolele din centrul imaginii din Figura 2.10a care sunt greșit grupate într-un singur obstacol. S-a considerat o fereastră de scanare de 7 acumuloare centrată pe rând în cele 30 de acumuloare (de la 0 la 29). Valoarea unui acumulator se consideră a fi un maxim local dacă are număr maxim de voturi în cadrul ferestrei de scanare și este peste media valorilor acumuloarelor din fereastră cu 10 voturi. Maximele locale sunt marcate cu puncte de culoare roșie. Pragul de separație este setat la mijlocul distanței dintre oricare două valori maxime consecutive. Acestea sunt marcate prin linii verticale de culoare verde.

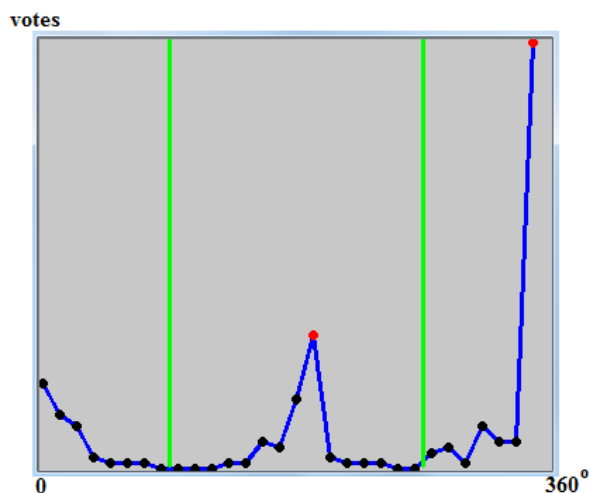


Figura 2.11. Histograma orientărilor vectorilor *OF* pentru cele două obiecte separabile din Figura 2.10a (două valori maxime locale, două linii de separație)

Pentru a re-eticheta superpixelii în obstacolele separate s-a definit o metodă bazată pe praguri multiple. Fiecare superpixel este asignat unui obstacol nou, determinat de cel mai apropiat maxim local. Rezultatul separației, bazat pe acești vectori de flux optic, este prezentat în Figura 2.12.

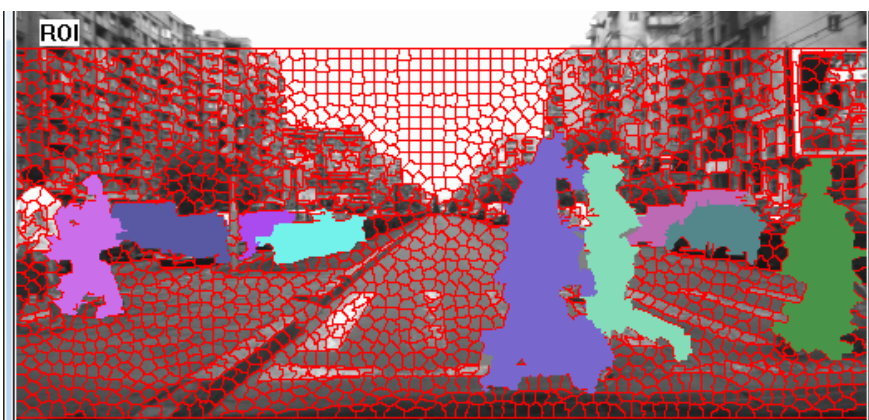


Figura 2.12. Separarea obstacolelor din mijloc pe baza informației fluxului optic

2.1.6. Rezultate experimentale

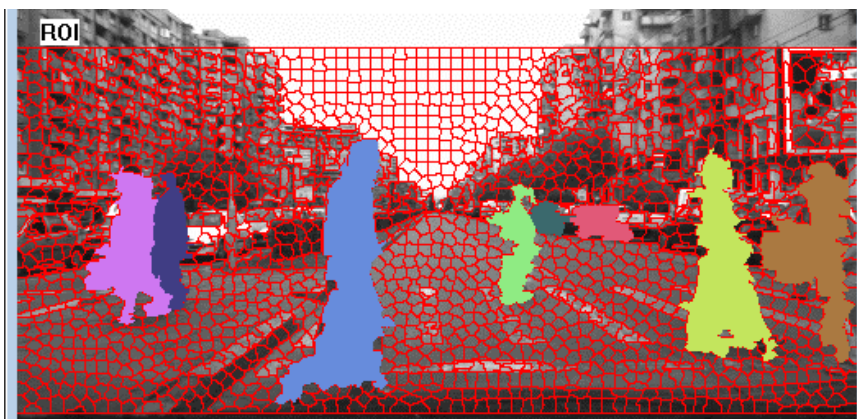
În această secțiune sunt prezentate rezultatele experimentale de detecție a obstacolelor prin superpixeli cu paradigma originală de combinare a informațiilor de intensitate, profunzime și flux optic descrisă anterior. Evaluarea s-a efectuat pe un set de secvențe video în niveluri de gri, conținând mii de cadre din diverse scenarii de trafic urban.

O evaluare calitativă (Figura 2.13) arată că obstacolele din fața ego-vehiculului sunt detectate foarte bine și sunt segmentate apropiat de forma lor reală. Pot să intervină și obstacole care nu sunt detectate, situațiile acestea fiind determinate de reconstrucția stereo eronată a punctelor 3D sau de lipsa acestora. Calitatea stereo-reconstrucției dată de metoda SORT-SGM face ca aceste cazuri să fie foarte rare. O altă problemă care apare

este cea legată de scenele de trafic foarte aglomerate, unde obstacolele care sunt foarte apropiate și care au trăsături similare de intensitate, profunzime și vectori de mișcare asemănători pot să apară detectate ca fiind un singur obstacol.



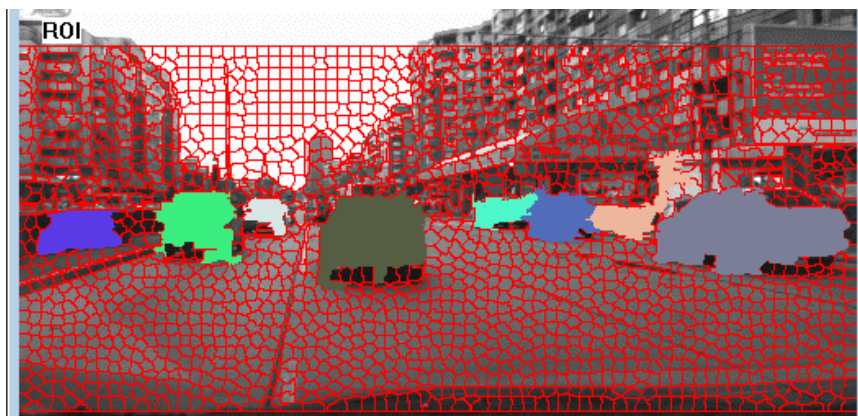
a)



b)



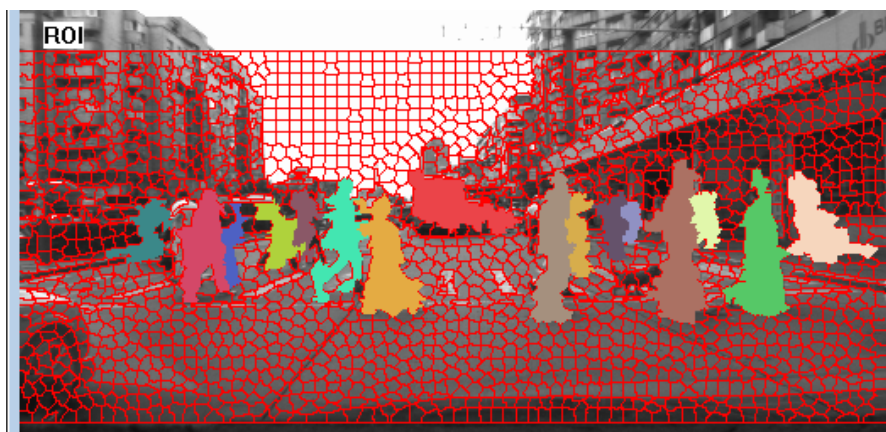
c)



d)



e)



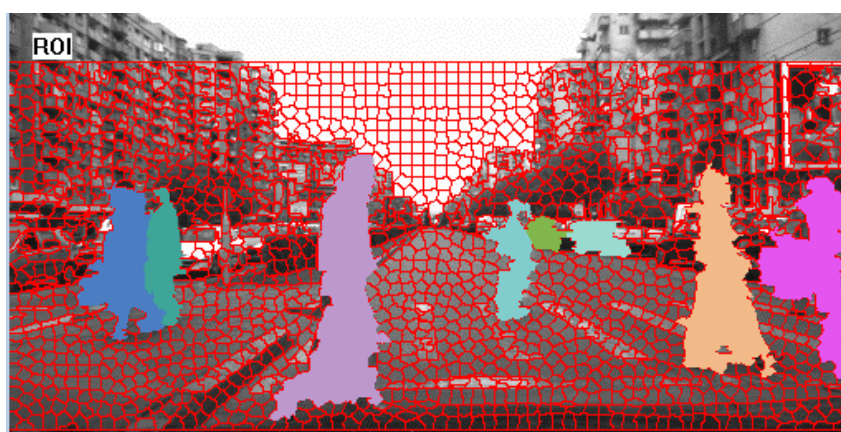
f)

Figura 2.13. Rezultatul detecției și segmentării obstacolelor cu superpixeli: a),c),e) imagini de intensitate a scenei; b),d),f) rezultatele corespunzătoare, fiecare obstacol cu o culoare aleatorie

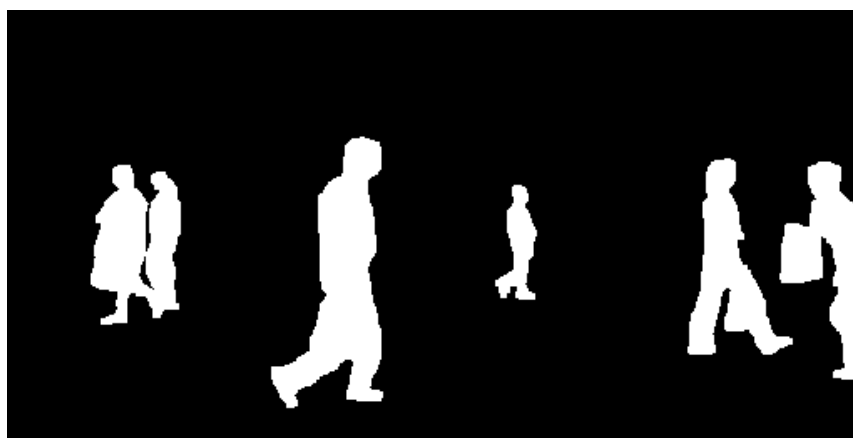
Evaluarea cantitativă a sistemului în termenii acurateții segmentării și detecției obstacolelor prin superpixeli este prezentată în Tabelul 2.1. S-a etichetat manual o serie de imagini de referință în care suprafețele fiecărui obstacol sunt segmentate precis. Evaluarea s-a făcut prin potrivirea suprafețelor obstacolelor detectate de sistem peste cele din setul etichetat (Figura 2.14). Procentele reprezintă gradul de acoperire (intersecția suprafețelor) a suprafețelor obstacolelor detectate în suprafețele obstacolelor etichetate manual și invers.

Tabelul 2.1. Acuratețea detecției și segmentării obstacolelor prin superpixeli

	Referință Obstacole	Referință	Obstacole
Suprafața (pixeli)	13,546,170	15,724,170	16,562,700
Procent de acoperire	X	86.2%	81.8%



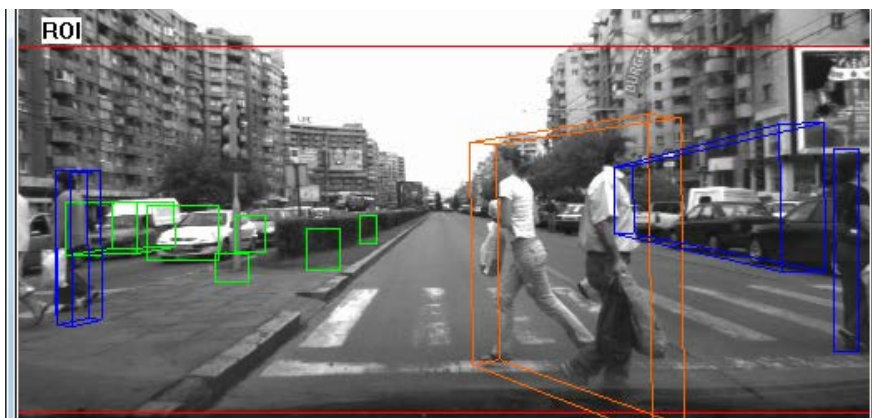
a)



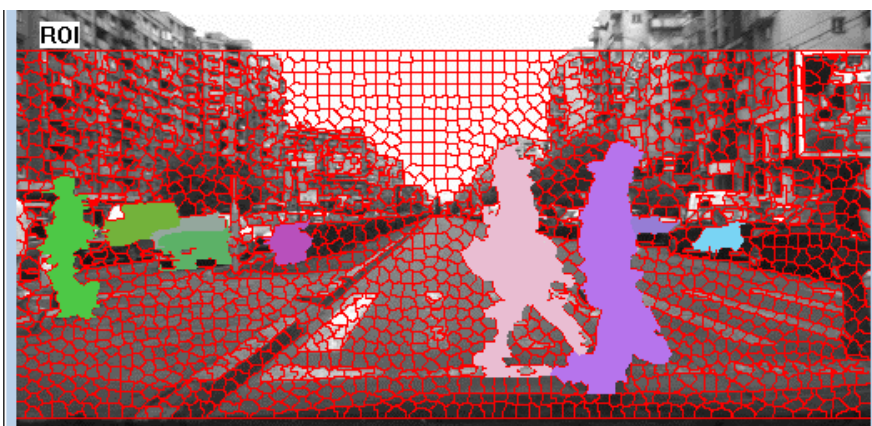
b)

Figura 2.14. Evaluarea metodei: a) imagine cu rezultate; b) imagine cu obstacole de referință etichetate manual

În comparație cu o altă tehnică de detecție a obstacolelor pe baza grupărilor punctelor 3D și a hărților de densități care le încapsulează în paralelipede dreptunghice, metoda descrisă bazată pe superpixeli, care integrează cele trei tipuri de informații intensitate, profunzime și mișcare, este superioară atât în detectarea obstacolelor cât și în segmentarea formei lor (Figura 2.15). Din punct de vedere al timpului de procesare, sistemul rulează în timp real, cu circa 20 cadre/s pe un procesor Intel i5 @ 3,33 GHz.



a)



b)

Figura 2.15. Comparație cu o abordare similară: a) detecția obstacolelor pe baza grupării punctelor 3D; b) rezultatele obținute cu metoda descrisă

2.2. Detecția obstacolelor prin analiza disparităților UV

În această secțiune este descrisă o metodă pentru detecția obstacolelor prin calculul și analiza hărților de disparitate UV . Obstacolele sunt detectate cu scopul utilizării lor ulterioare ca ipoteze într-un proces de recunoaștere a pietonilor. Metoda are succes și datorită faptului că se utilizează punctele stereo reconstruite dens și de înaltă calitate obținute prin aplicarea algoritmului SORT-SGM [52]. Se consideră că suprafața drumului din fața ego-vehiculului, pe care se află obstacolele de interes, este aproximativ planară. Se descrie o metodă de estimare a suprafeței drumului din analiza hărții de disparitate V , aceasta fiind utilizată ulterior pentru separarea punctelor de obstacole de cele de pe suprafața drumului. Harta de disparitate U este utilizată pentru gruparea punctelor reconstruite în obstacole, rezultând o listă de obstacole precis definite și poziționate în scenă. În final se definește o procedură de filtrare a obstacolelor pe bază de dimensiuni, cu rolul de a obține ipotezele de pietoni.

2.2.1. Componentele principale ale sistemului de detecție a obstacolelor

Modulele principale ale sistemului de detecție este prezentat în Figura 2.16. Imaginile achiziționate sunt de rezoluție 512x383 pixeli cu două camere în niveluri de

gri, montate într-o configurație stereo. Reconstrucția stereo este realizată cu algoritmul SORT-SGM pe imaginile obținute în urma proceselor de eliminare a distorsiunilor și de rectificare (Figura 2.1).

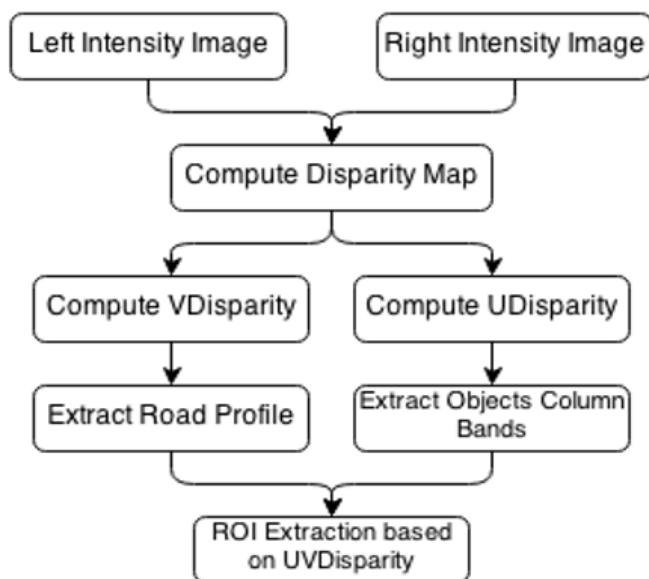


Figura 2.16. Modulele componente ale sistemului de detecție a obstacolelor și generarea regiunilor de interes (ipotezelor de pietoni)

Pentru a extrage regiunile de interes (*ROI*) cu obstacole, trebuie calculat spațiul disparităților *UV*. Acesta se bazează pe harta de disparitate, care este obținută din cele două imagini stânga și dreapta, în urma procesului de stereo-corespondență.

2.2.2. Sistemul de stereoviziune

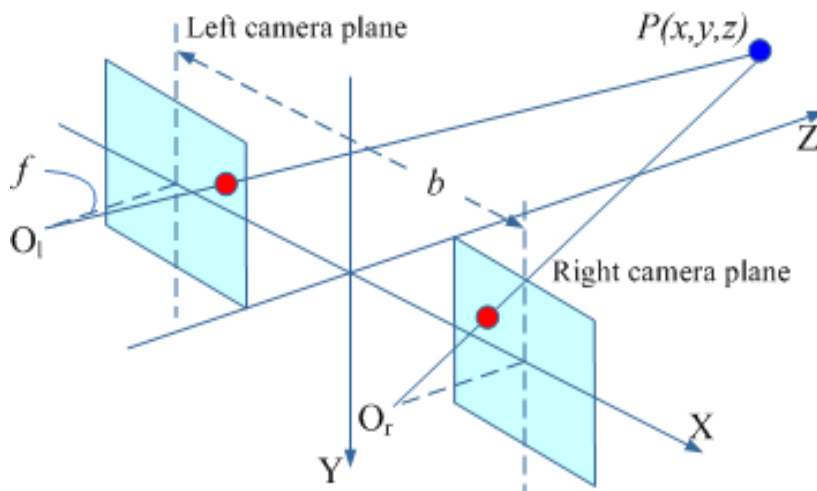


Figura 2.17. Sistem de stereoviziune canonic

Harta de disparitate se calculează pe baza punctelor 3D reconstruite. Cunoscând profunzimea fiecărui punct reconstruit și distanța dintre camerele stereo, disparitatea se calculează cu formula (2.12). Parametrii acesteia sunt: d – valoarea disparității, x_l și x_r

reprezintă valorile coordonatelor x în care pixelul apare în imaginea stângă respectiv dreaptă; f – distanța focală a camerelor; b – distanța dintre camere în sistemul de stereoviziune (Figura 2.17); Z – profunzimea punctului 3D în scenă.

$$d = x_l - x_r = f \frac{b}{Z} \quad (2.12)$$

2.2.3. Spațiul disparităților UV

Spațiul disparităților U este o matrice de acumuloare în care, pe fiecare coloană, votează disparitățile din harta de disparitate pe rândul corespunzător valorilor lor.

$$u_{id} = \sum_{j=0}^{rows} \Delta_{ij}, \quad \Delta_{ij} = \begin{cases} 1, & disp_{ij} = d \\ 0, & otherwise \end{cases} \quad (2.13)$$

În formula (2.13), u_{id} reprezintă valoarea din spațiul disparităților U care acumulează numărul de pixeli care au disparitatea d de pe coloana i din harta de disparitate.

Analog cu spațiul de disparitate U , spațiul de disparitate V este tot o matrice de acumuloare în care, pe fiecare rând, votează disparitățile din harta de disparitate pe coloana corespunzătoare valorilor lor.

$$v_{dj} = \sum_{i=0}^{cols} \Delta_{ij}, \quad \Delta_{ij} = \begin{cases} 1, & disp_{ij} = d \\ 0, & otherwise \end{cases} \quad (2.14)$$

În formula (2.14), v_{dj} reprezintă valoarea din spațiul disparităților V care acumulează numărul de pixeli care au disparitatea d de pe rândul j din harta de disparitate.



Figura 2.18. Exemplu de spații de disparitate UV corespunzătoare scenei: la dreapta V , dedesubt U

Următoarea fază constă în extragerea suprafeței drumului. Aceasta se aproximează ca fiind plană și se extrage din spațiul de disparitate V , urmând ca obstacolele să fie determinate pe benzi verticale din spațiul de disparitate U .

2.2.4. Extragerea suprafeței drumului

Suprafața planară a drumului determină o linie dreaptă în spațiul de disparitate V . Obiectivul este de a determina această linie. O metodă clasică este cea bazată pe transformata Hough [59]. Aceasta convertește toate punctele din spațiul cartezian într-un spațiu de coordonate polare în care fiecare punct din imaginea inițială determină o curbă. Intersecția dintre mai multe astfel de curbe determină coordonatele polare ale unei linii în imagine. Se definește un prag minim de intersecții pentru care o linie se consideră validă și apoi se iau în considerare toate liniile care îndeplinesc această condiție. Pentru determinarea suprafeței drumului trebuie să găsim linia cea mai bună care rezultă din algoritmul Hough (linia care are cele mai multe intersecții în spațiul coordonatelor polare).

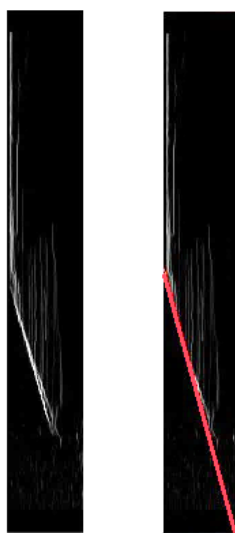


Figura 2.19. Linia cea mai bună (dreapta) extrasă din spațiul disparităților V (stânga) cu ajutorul algoritmului Hough

2.2.5. Etichetarea spațiului de disparitate

După ce suprafața drumului a fost determinată, se vor detecta și eticheta obstacolele de pe drum. Pentru aceasta, se utilizează coordonatele orizontale ale spațiului de disparitate U cu scopul de a delimita fiecare obstacol și de a defini câte o regiune de interes rectangulară în jurul acestora. În spațiul de disparitate U , un obstacol (plan vertical) din scenă apare aproximativ ca un segment de dreaptă. Spațiul de disparitate U prezintă zgomote, ceea ce face dificilă detecția precisă a segmentelor de dreaptă. Pentru eliminarea acestor zgomote cauzate fie de suprafața drumului fie de obstacole de dimensiuni foarte mici, s-a stabilit empiric și apoi aplicat un prag pentru înălțimea minimă a obstacolelor. Spațiul de disparitate U se binarizează cu acest prag (puncte de obstacol și puncte de fundal) și apoi sunt extrase segmentele de dreaptă cu un algoritm definit special pentru etichetare.

Algoritmul de etichetare definit are la bază metoda clasică [60] ce presupune scanarea imaginii spațiului de disparitate U pe rânduri de sus în jos și pe coloane de la stânga la dreapta. În momentul în care se găsește un punct de obstacol, se salvează, i se

atribuie o nouă etichetă și se caută toți vecinii lui. Fiecărui vecin i se atribuie aceeași etichetă care a fost asignată primului pixel găsit și procedeul se continuă până la etichetarea întregii componente conexe. După aceasta se etichetează și restul componentelor conexe, procesul continuând până când toate segmentele sunt etichetate, respectiv se întâlnește colțul din dreapta jos al imaginii și nu mai există puncte de obstacol neetichetate.

Pentru a putea segmenta cât mai bine obstacolele (discrimina cât mai bine segmentele de dreaptă în spațiul de disparitate binarizat U), căutarea vecinilor se face pe rânduri (fiecare rând având aceleași valori ale disparităților). Disparitatea punctelor de pe obstacole este invers proporțională cu distanța lor față de camere.

În cazul în care avem de a face cu obstacole aflate la distanțe mari (disparități foarte mici), în procesul de etichetare, trebuie să considerăm vecinii pe un domeniu foarte restrâns pe direcțiile orizontale și verticale iar în caz contrar domeniile trebuie să fie mai largi. De aceea, în algoritmul de etichetare, s-a permis ca vecinătatea unui punct să varieze cu distanța, așa cum este prezentat în Figura 2.20. Acest lucru este motivat și de faptul că în spațiul de disparitate U obstacolele apropiate pot avea fluctuații mai mari ale valorilor disparităților (axa verticală) și a poziției punctelor reconstruite (axa orizontală) (Figura 2.20).

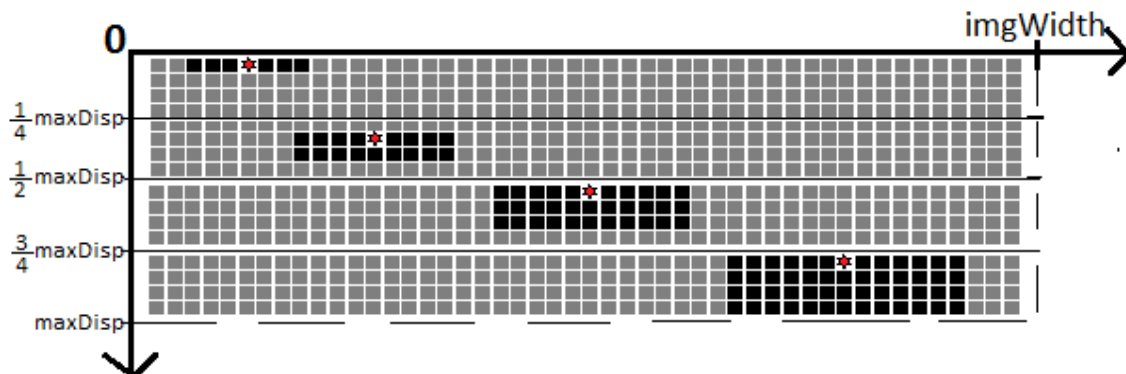


Figura 2.20. Vecinătatea (cu culoare neagră) fiecărui punct (roșu) în spațiul de disparitate U în funcție de valoarea posibilă a disparității

Rezultatul etichetării constă într-o listă de obstacole, fiecare obstacol având un set de puncte care fac parte, pe baza relației de vecinătate, din aceeași componentă conexă. Pentru fiecare obstacol astfel determinat se definesc patru puncte principale: ordonata punctului cu disparitatea cea mai mică (punctul 1 în Figura 2.21), abscisa punctului cu poziția cea mai din stânga (punctul 2 în Figura 2.21), abscisa punctului cu poziția cea mai din dreapta (punctul 3 în Figura 2.21), ordonata punctului cu disparitatea cea mai mare (punctul 4 în Figura 2.21). Aceste patru puncte definesc limitele regiunii de interes rectangulare care circumscrie fiecare obstacol.

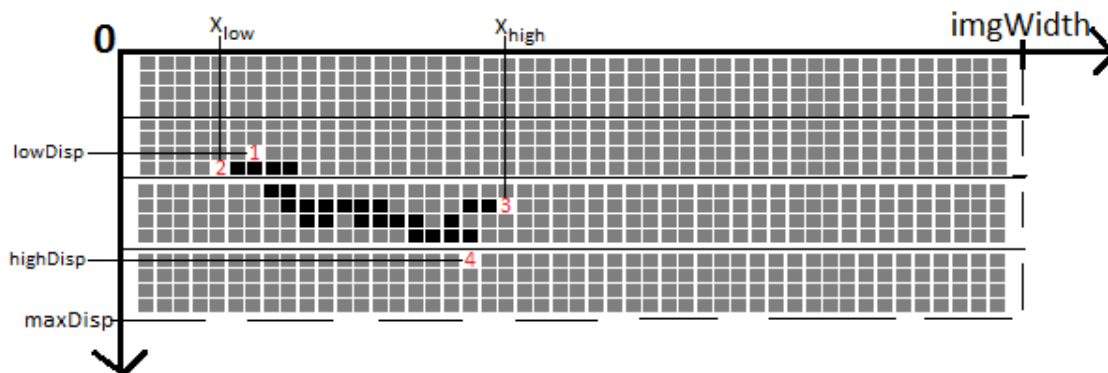


Figura 2.21. Obstacol etichetat și punctele lui de margine

În determinarea obstacolelor din spațiul de disparitate U s-a experimentat atât algoritmul de etichetare definit anterior cât și o metodă care găsește segmente de dreaptă pe baza transformatei Hough. Algoritmul de etichetare definit este superior metodei Hough. Metoda Hough, în cazurile în care există obstacole în scenă la aceeași distanță, are problema detecției unui singur segment de dreaptă ceea ce conduce eronat la găsirea unui singur obstacol. De asemenea, în cazurile în care valorile disparităților de pe suprafața vizibilă a unui obstacol sunt ușor diferite, metoda Hough va detecta eronat mai multe obstacole care ar trebui ulterior unificate. În Figura 2.22 se poate observa că algoritmul de etichetare definit detectează corect gruparea de segmente, ca fiind un singur obstacol, în timp ce algoritmul Hough detectează fiecare segment ca fiind un obstacol individual.

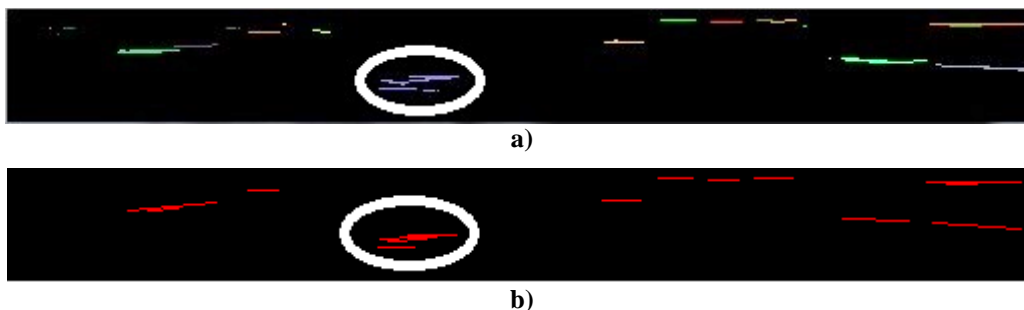


Figura 2.22. Comparație între asignarea segmentelor de dreaptă la obstacole în spațiul de disparitate U : a) prin algoritmul de etichetare definit – fiecare culoare reprezintă un obstacol separat; b) prin metoda Hough – fiecare linie reprezintă un obstacol separat

2.2.6. Delimitarea obstacolelor în scena de trafic

Pentru stabilirea regiunilor de interes din imaginea de intensitate trebuie parcursă lista segmentelor de dreaptă determinată anterior prin etichetare. Se consideră pe rând fiecare segment și se parcurge banda verticală de acoperire a acestuia (delimitată de valorile x_{low} și x_{high} în spațiul binarizat de disparitate U) cu scopul de a marca pixelii care au valorile disparităților cuprinse în intervalul $[lowDisp; highDisp]$ corespunzătoare obstacolului respectiv.

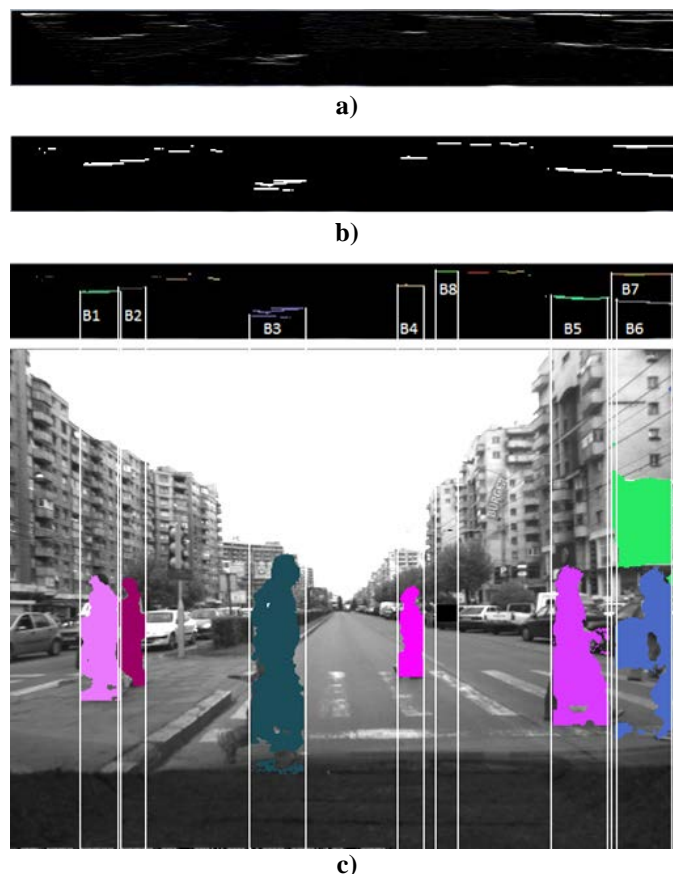


Figura 2.23. Procesarea spațiului de disparitate U : a) spațiul de disparitate U ; b) binarizarea spațiului de disparitate U ; c) delimitarea obstacolelor pe orizontală

În Figura 2.23 se pot observa benzile verticale B1, B2, ... , B8 care definesc obstacole și care sunt determinate cu ajutorul metodei descrise. Pentru reducerea numărului de benzi verticale și eliminarea zgomotelor s-a impus un prag de lățime bazat pe lățimea minimă a unui obstacol care poate fi ipoteză de pieton. Regiunile de interes rezultate sunt prezentate în Figura 2.24.

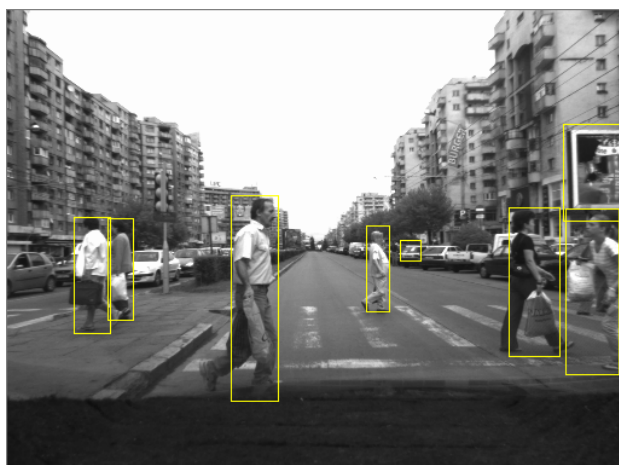


Figura 2.24. Regiuni de interes detectate prin analiza spațiilor de disparitate UV

Fiecare regiune de interes rectangulară care conține un obstacol este delimitată de patru linii: două linii verticale (x_{low} și x_{high}) care delimitează lățimea obstacolului (Figura

2.23c) și două linii orizontale care delimitează înălțimea obstacolului. Planul pe care este situată suprafața drumului se calculează utilizând algoritmul Hough în spațiul de disparitate V și a fost descris anterior. Deoarece pentru fiecare obstacol există un domeniu larg de disparități, se consideră valorile cele mai ridicate pentru a putea găsi valoarea y_{max} care determină în imagine linia orizontală $y=y_{max}$ cea mai de jos. Aceasta reprezintă separația dintre obstacol și suprafața drumului și este descrisă de ecuația (2.15) în care a , b , c sunt parametri determinați cu algoritmul Hough, $high_{Disp}$ este valoarea cea mai mare din intervalul disparităților.

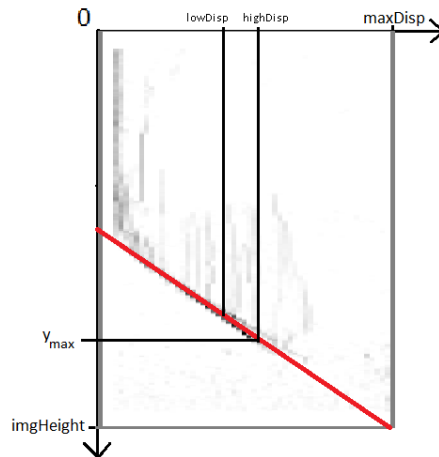


Figura 2.25. Calculul valorii y_{max} pe baza domeniului de disparitate U și a liniei detectate din spațiul de disparitate V

$$ahigh_{Disp} - by_{max} + c = 0 \quad (2.15)$$

Ultimul pas constă în filtrarea regiunilor de interes detectate eronat. Pentru aceasta s-a considerat un singur aspect: regiunile de interes cu obstacole care au număr de pixeli mai mic decât un prag prestabilit sunt eliminate. Acest prag este determinat experimental pe baza observațiilor dimensiunilor regiunilor de interes valide (Figura 2.26).

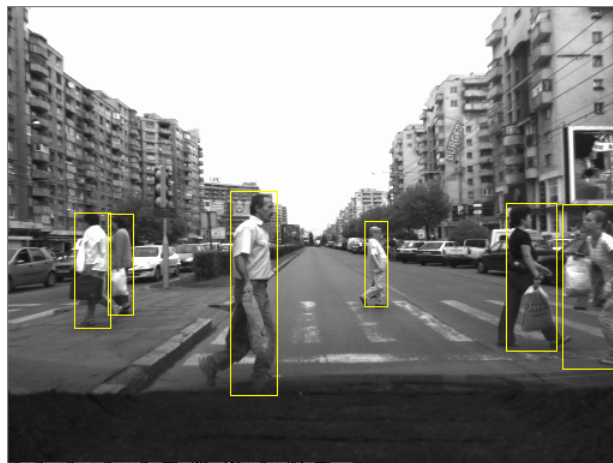


Figura 2.26. Regiuni de interes cu obstacole valide

Detecția obstacolelor din scenele de trafic rulează în timp real cu aproximativ 26 cadre/s pe un calculator cu procesor Intel Core i3 @ 2.39 GHz.

2.3. Detecția obstacolelor prin analiza vectorilor de mișcare relativă

În această secțiune este definită o metodă pentru detecția obstacolelor în trafic situate atât pe benzile de circulație cât și pe spațiile din preajma acestora, cum sunt trotuarele. Detecția obstacolelor are la bază analiza vectorilor de mișcare relativă din succesiunea de cadre. În primă fază, fluxul optic este calculat cu ajutorul algoritmului Lucas-Kanade [57] pe un set de puncte distribuit uniform pe o structură de raster peste întreaga imagine. Rezultatul aplicării constă în setul de vectori de mișcare relativă care sunt utilizați apoi atât pentru estimarea mișcării fundalului cât și pentru segmentarea obstacolelor. Algoritmul *RANSAC* [61] este apoi utilizat pentru determinarea mișcării fundalului cu scopul eliminării ei din setul de vectori de mișcare relativă. Obstacolele sunt definite ca fiind grupări de vectori de mișcare obținute în urma unui procedeu în doi pași de grupare ierarhică pe baza trăsăturilor de localizare, magnitudine și unghi. Obstacolele sunt apoi filtrate pentru a elimina eventualele grupări determinate de zgomote. În final este evaluată și semnalizată eventuala posibilitate de coliziune iminentă cu ego-vehiculul.

2.3.1. Componentele principale ale sistemului de detecție a obstacolelor

În Figura 2.27 este prezentată arhitectura sistemului de detecție a obstacolelor cu toate modulele lui componente: achiziția de secvențe de imagini color de rezoluție 352x288 pixeli, extragerea/fixarea punctelor de interes, urmărirea punctelor de interes (între cadre succesive prin calculul fluxului optic), estimarea fundalului și eliminarea acestuia, selecția punctelor candidate de obstacol, gruparea punctelor de interes în obstacole și în final validarea obstacolelor.

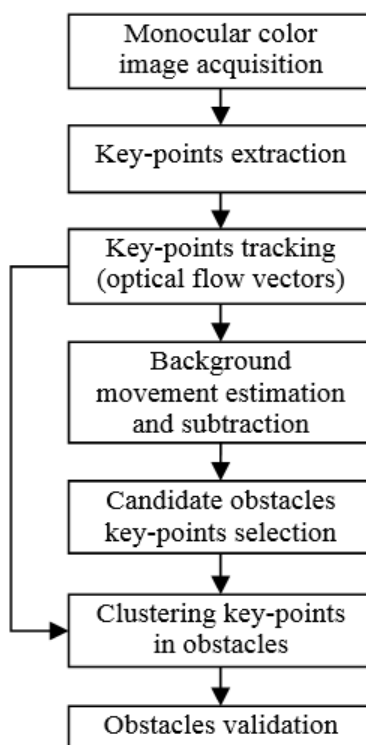


Figura 2.27. Sistemul de detecție a obstacolelor pe baza analizei vectorilor de mișcare relativă

2.3.2. Extragerea punctelor de interes

Imaginile color achiziționate conțin un număr foarte mare de pixeli care nu ar putea fi considerate toate pentru calculul și analiza vectorilor de mișcare relativă dintre cadrele succesive. De aceea, pentru a spori viteza de calcul, trebuie extrase și selectate anumite puncte de interes care vor fi utilizate ulterior pentru calculul fluxului optic. Obiectivul este de a estima inițial mișcarea fundalului din cadre consecutive cu ajutorul vectorilor de mișcare asociați punctelor de interes.

Punctele de interes sunt în general acele puncte pentru care se pot calcula trăsături robuste și care pot fi urmărite ușor între cadre (ex. puncte cu trăsături invariante *SIFT* [62]). Dezavantajul acestora constă în faptul că vor exista multe puncte de interes pe suprafețele cu textură și foarte puține sau chiar deloc pe suprafețele de culoare uniformă (care apar de obicei atât pe fundal cât și pe îmbrăcămintea persoanelor sau pe suprafețele altor obstacole). Acest lucru poate duce la imposibilitatea detecției corecte a mișcării fundalului.

Metoda de a evita aceste aspecte nedorite este de a fixa uniform punctele de interes pe o structură de raster și care nu țin cont de trăsăturile locale. Granularitatea acestui raster este descrisă de τ în ecuația (2.16) în care W și H reprezintă lățimea respectiv înălțimea imaginii, iar N este un parametru care specifică numărul total de puncte de interes.

$$\tau = \frac{W * H}{N} \quad (2.16)$$

Diferența între a considera punctele de interes pe baza trăsăturilor *SIFT* și cele determinate de distribuția lor uniformă pe raster este prezentată în Figura 2.28. Observația este că numărul de puncte *SIFT* este foarte scăzut în zonele cu textură uniformă.



Figura 2.28. Selecția punctelor de interes (puncte de culoare roșie): a) pe baza trăsăturilor *SIFT*; b) distribuite uniform pe o structura de tip raster

Numărul N de puncte de interes considerate este deosebit de important. În cazul în care acesta este foarte mare, timpul de procesare poate fi foarte ridicat, deci inacceptabil. În cazul contrar, cu un număr foarte redus de puncte de interes, detecția

obstacolelor prin analiza vectorilor de mișcare relativă ar putea fi dificil de realizat, neavând suficiente instanțe care să poată fi grupate. O soluție de compromis între acuratețe și timp de execuție constă în setarea valorii $N=1500$ de puncte de interes pentru imaginile achiziționate de rezoluție 352x288 pixeli.

2.3.3. Calculul vectorilor de mișcare relativă și urmărirea punctelor de interes

Următorul pas al algoritmului definit constă în calculul mișcării fiecărui punct de interes în secvențele de cadre consecutive. Cunoscând poziția fiecărui punct de interes în cadrul curent, se dorește găsirea corespondenței aceluiasi punct în cadrul următor. Acest lucru se poate realiza prin calculul fluxului optic care reprezintă o formă a mișcării aparente a elementelor din scenă determinată de mișcarea relativă dintre cameră și scenă. Fluxul optic se calculează pornind de la presupunerea că intensitatea în orice punct a scenei rămâne constantă, chiar dacă poziția lui în imagine se schimbă de la un cadru la altul – ecuația (2.17). Aproximarea în serie Taylor a ecuației (2.17) rezultă în (2.18).

$$I(x, y, t) = I(x + v_x(x, y), y + v_y(x, y), t + I) \quad (2.17)$$

$$I(x, y, t + I) = I'_x(x, y, t)v_x(x, y, t) + I'_y(x, y, t)v_y(x, y, t) + I_t(x, y, t) \quad (2.18)$$

Rezolvarea ecuației (2.18) oferă fluxul de mișcare v_x și v_y . Pentru rezolvarea acesteia trebuie definite constrângeri suplimentare. Algoritmul Lucas-Kanade presupune că valorile fluxului optic și ale intensității sunt constante în fiecare punct de interes și de asemenea și într-o fereastră în jurul acestora. Soluția poate fi găsită prin rezolvarea sistemului de ecuații (2.19) prin metoda celor mai mici pătrate.

$$\begin{aligned} I'_x v_x + I'_y v_y &= -I'_t \\ I''_x v_x + I''_y v_y &= -I''_t \\ &\dots \\ I''_x v_x + I''_y v_y &= -I''_t \end{aligned} \quad (2.19)$$

Stabilitatea acestei metode de rezolvare este dată de matricea $A^T A$, în care A este matricea sistemului de ecuații (2.19). În cazul în care aceasta nu are două valori proprii mari, sistemul nu poate fi rezolvat. Această condiție este echivalentă cu existența muchiilor pe două direcții ortogonale în fereastra de vecinătate considerată. Vectorii de flux optic sunt calculați cu algoritmul Lucas-Kanade, un exemplu fiind prezentat în Figura 2.29. Fiecare vector este caracterizat prin magnitudine și unghiul de mișcare. Aceste două valori sunt utilizate ulterior pentru găsirea și gruparea punctelor de interes în obstacole dar și pentru eliminarea zgomotului. Dacă punctul $P_1(x_{1i}, y_{1i})$ este un punct de interes în cadrul anterior și $P_2(x_{2i}, y_{2i})$ este corespondentul lui în cadrul curent, atunci trăsăturile date de fluxul optic sunt calculate cu ajutorul formulelor (2.20) unde $D_{i(1,2)}$ și $\theta_{i(1,2)}$ reprezintă magnitudinea respectiv unghiul mișcării relative.

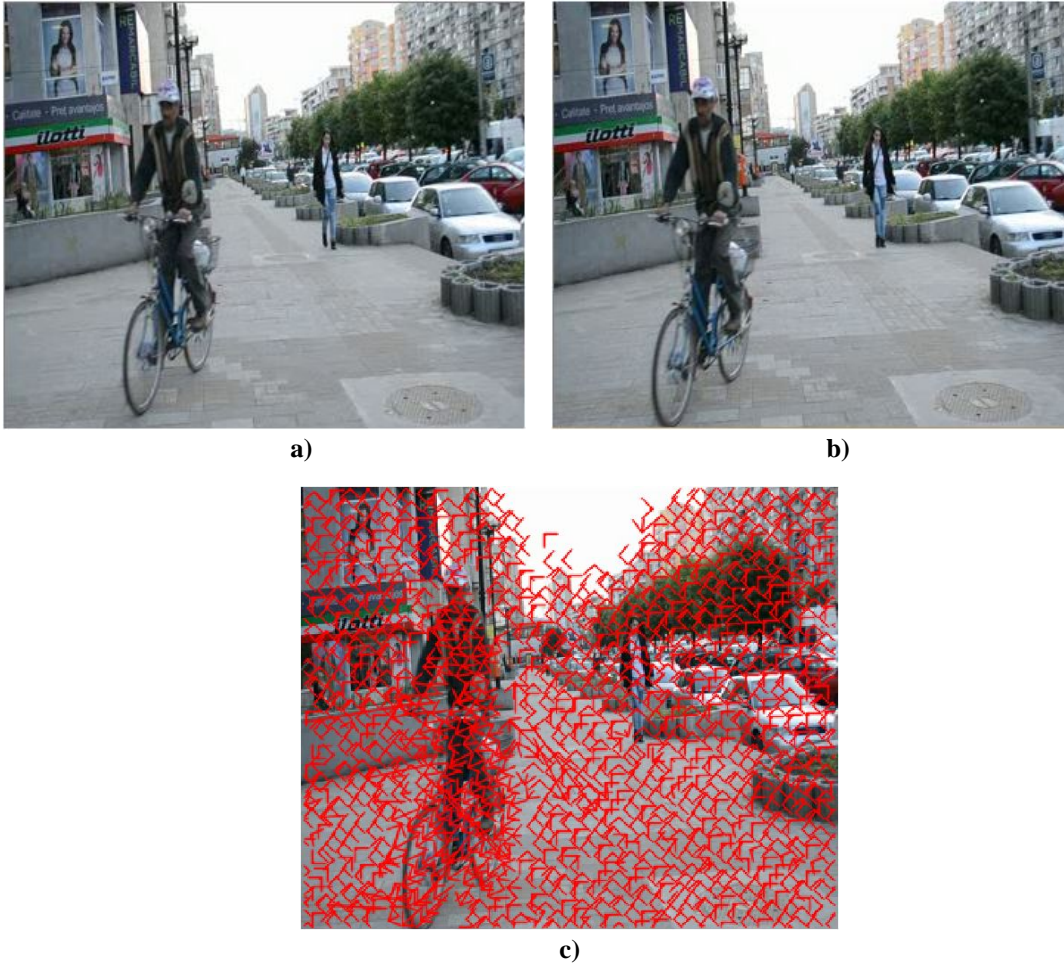


Figura 2.29. Calculul fluxului optic: a) cadrul precedent; b) cadrul curent; c) vectorii de mișcare (culoare roșie) peste cadrul curent

$$\begin{aligned}
 v_{ix} &= x_{2i} - x_{1i}, v_{iy} = y_{2i} - y_{1i} \\
 D_{i(1,2)} &= \sqrt{v_{ix}^2 + v_{iy}^2}, i = \overline{1, n} \\
 \theta_{i(1,2)} &= \arccos \frac{v_{ix}}{D_{i(1,2)}}, \theta \in [0, 2\pi]
 \end{aligned}
 \tag{2.20}$$

2.3.4. Estimarea mișcării fundalului

În secțiunea anterioară s-a descris modul de calcul al mișcării relative individuale a fiecărui punct de interes. Se extrag apoi punctele de interes care au mișcare diferită (mai mare sau mai mică) față de mișcarea generală a fundalului. Aceste puncte vor fi considerate ca fiind ipoteze de puncte aparținătoare obstacolelor.

Prin mișcarea generală a fundalului înțelegem mișcarea acelor pixeli care nu sunt pe suprafața obstacolelor (presupunând că suprafețele obstacolelor sunt mai restrânse decât suprafața fundalului). Considerând punctul P_1 în cadrul anterior ca fiind aparținător fundalului și aplicând o regulă generală de mișcare a fundalului H se poate estima corespondentul aceluiași punct în cadrul curent P_{2est} . Regula de transformare H este

considerată corectă dacă diferența dintre poziția estimată P_{2est} și poziția reală P_2 este suficient de mică. Matricea de transformare H se calculează cu scopul de a estima locațiile punctelor de fundal în cadrul curent pe baza locațiilor cunoscute din cadrul anterior – ecuația (2.21).

$$\begin{bmatrix} x_{2i}^{est} \\ y_{2i}^{est} \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \end{bmatrix} * \begin{bmatrix} x_{1i} \\ y_{1i} \\ 1 \end{bmatrix} \quad (2.21)$$

Eroarea de estimare este calculată utilizând metrica de distanță Manhattan – ecuația (2.22). Dacă eroarea este mai mare decât un prag ales empiric, atunci se poate considera că punctul nu respectă mișcarea generală a fundalului, însemnând că are o mișcare diferită și putând fi considerat ca fiind punct candidat de obstacol.

$$E = |p_{2i}^{est} - p_{2i}| = |x_{2i}^{est} - x_{2i}| + |y_{2i}^{est} - y_{2i}| \quad (2.22)$$

$$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \end{bmatrix} \quad (2.23)$$

S-a utilizat algoritmul *RANSAC* pentru calculul parametrilor matricei de transformare H (2.23). Setul de date de intrare pentru algoritm constă în perechi de puncte de interes corespondente din cadrul anterior în cadrul curent, calculate cu ajutorul fluxului optic. Algoritmul necesită mai multe iterații. În cadrul fiecărei iterații, acesta alege în mod aleatoriu perechile de puncte corespondente, calculează matricea H și eroarea totală asociată E . După N iterații, se alege matricea estimată H cu eroarea E cea mai mică. Matricea H are 6 parametri. Cu ajutorul unei singure perechi de puncte corespondente se poate scrie un set de două ecuații (2.24).

$$\begin{aligned} x_{2i}^{est} &= h_{00} * x_{1i} + h_{01} * y_{1i} + h_{02} \\ y_{2i}^{est} &= h_{10} * x_{1i} + h_{11} * y_{1i} + h_{12} \end{aligned} \quad (2.24)$$

La fiecare iterație trebuie să se aleagă trei perechi de puncte corespondente pentru a putea rezolva sistemul de ecuații în mod determinist cu rolul de a găsi parametrii matricei H care definesc mișcarea generală a fundalului. Eroarea de estimare – ecuația (2.22) – se calculează pentru fiecare punct. S-au definit două praguri: pragul inferior T_l și cel superior T_h . Dacă eroarea este sub pragul inferior atunci punctul se consideră aparținând fundalului. Dacă eroarea este peste pragul superior, atunci punctul se consideră ca fiind zgomet și este eliminat. Un punct este aparținător obstacolelor dacă eroarea de mișcare este între cele două praguri – inegalitatea (2.25). Valorile lor au fost stabilite la $T_l=4$ și $T_h=30$ de pixeli. Un exemplu de vectori de mișcare care au fost asigurați obstacolelor este prezentat în Figura 2.30.

$$T_l \leq E \leq T_h \quad (2.25)$$



Figura 2.30. Vectori de mișcare (culoare roșie) asignați obstacolelor

2.3.5. Gruparea și validarea vectorilor de mișcare relativă în obstacole

În continuare s-a descris metoda definită pentru gruparea vectorilor de mișcare relativă în obstacole individuale. Pentru definirea obstacolelor sunt considerate doar acele puncte de interes care nu au fost eliminate în pasul anterior ca fiind aparținătoare fundalului. Procesul de grupare este unul care pornește de la unificarea punctelor aflate la distanțele cele mai mici și continuând crescător spre valorile mai mari. Acesta are la bază algoritmul de grupare ierarhică [63] cu o metrică specifică de asemănare a trăsăturilor. Metoda este iterativă și pleacă inițial de la asignarea fiecărui punct în câte o clasă separată. La fiecare iterație două sau mai multe clase sunt unite pentru a forma o clasă nouă. Procesul de unificare continuă până când distanța minimă dintre oricare două clase este mai mare decât un prag prestabilit.

Asemănarea trăsăturilor a fost calculată relativ simplu. Între fiecare două puncte candidate de obstacol s-a luat în considerare locația, magnitudinea și unghiul vectorilor de mișcare asociați lor. În prima iterație se unifică perechea de două puncte care sunt cele mai apropiate, cu formarea unei clase noi. După acest pas se re-calculează distanța de la alte clase la această clasă nou formată, și procedeul continuă în același mod până la convergență. Există diverse posibilități de a defini distanța dintre două clase. S-a considerat ca aceasta să fie determinată de minimumul distanței dintre oricare pereche de puncte (unul din prima clasă și celălalt din a doua clasă).

Gruparea se face în doi pași: în primul pas se grupează vectorii de mișcare pe baza locației lor. Distanța dintre o pereche de vectori de mișcare este definită ca distanța Euclidiană dintre originile lor. Trebuie definit un prag de distanță care se alege experimental cu scopul de a obține rezultate bune. În cazul în care există un număr $N=1500$ de puncte de interes în imaginea scenei de 352×288 pixeli, o valoare potrivită a acestui prag T care definește distanța maximă permisă dintre două clase care pot fi unite este $T=20$ pixeli.

Pentru a realiza o grupare bună nu este suficientă considerarea doar a locației vectorilor de mișcare relativă. În calculul fluxului de mișcare și în procesul de eliminare a fundalului pot apărea diverse erori. De aceea, pentru eliminarea punctelor greșit asignate

în obstacole, trebuie să se analizeze trăsăturile comune ale majorității membrilor (punctelor) din fiecare grup (obstacol) și să se elimine acele puncte care diferă în trăsăturile lor. O trăsătură comună pentru toate punctele aparținătoare aceluiași obstacol este în general unghiul vectorilor de mișcare asociați punctelor de interes. Altă trăsătură comună este magnitudinea aceluiași vectori. În cadrul fiecărei clase, punctele componente trebuie să aibă aproximativ aceleași valori pentru unghi și magnitudine. S-a utilizat o histogramă pentru eliminarea punctelor care nu respectă unghiul și magnitudinea majorității punctelor din fiecare clasă (obstacol). Aceasta este o histogramă bi-dimensională care consideră magnitudinea și unghiul în definirea acumulatele. După calculul acestora se calculează valoarea maximă din acumulatele și se reține perechea ($Magn_{class}$, $Angle_{class}$) corespunzătoare acestui maxim. Aceasta este considerată referință pentru gruparea de vectori de mișcare corespunzătoare obstacolului. Toate celelalte puncte de interes i cu vectorii de mișcare asociați, care diferă în valoarea magnitudinii și/sau unghiului față de referință, sunt eliminate. Diferențele ($E_{magn,i}$ și $E_{angle,i}$) față de valorile de referință se calculează cu ajutorul distanței Manhattan:

$$\begin{aligned} E_{magn,i} &= |Magn_{class} - Magn_i| \\ E_{angle,i} &= |Angle_{class} - Angle_i| \end{aligned} \quad (2.26)$$

Se stabilesc experimental două praguri $T_{magn}=5$ și $T_{angle}=45^\circ$ pe valorile erorilor calculate anterior. Un punct de interes este eliminat dacă parametrii vectorului de mișcare relativă asociat respectă cel puțin una din următoarele condiții:

$$\begin{aligned} E_{magn,i} &> T_{magn} \\ E_{angle,i} &> T_{angle} \end{aligned} \quad (2.27)$$

În cazul unghiului vectorilor de mișcare se poate seta și o valoare mai mare de 45° pentru pragul T_{angle} întrucât, pentru un obstacol mai apropiat de sistemul de achiziție și care se deplasează cu viteză relativ mare, pot apărea pe suprafața obstacolului vectori de mișcare cu valori relativ diferite ale unghiurilor lor.



Figura 2.31. Diferențe mari între unghiurile vectorilor de mișcare relativă de pe suprafața aceluiași obstacol (cu culoare albastră)

După eliminarea punctelor care nu pot fi aparținătoare obstacolelor, se aplică încă o dată procedura de grupare a vectorilor de mișcare. Acest lucru este necesar pentru a

asigura că nu există grupări cu instanțe între care distanța este mai mare decât pragul admis. Un astfel de caz este prezentat în Figura 2.32. În prima iterație, punctele de interes A , B și C sunt grupate împreună. Se presupune că în urma procesului de filtrare, punctul B este eliminat. Distanța minimă dintre A și C devine mai mare decât pragul maxim admis. Astfel, aplicând procesul de re-grupare a vectorilor, punctele A și C vor fi distribuite în două clase (obstacole) separate.

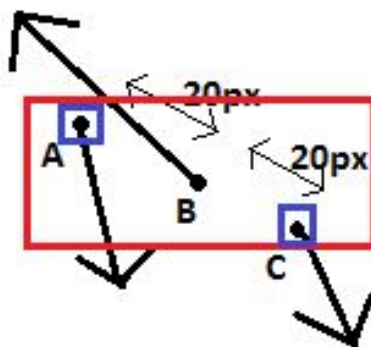


Figura 2.32. Trei vectori de mișcare corespunzători punctelor A , B și C

Gruparea rezultată de vectori de mișcare relativă corespunzători unui obstacol pot conține un număr foarte scăzut de instanțe. Acest caz poate apărea din cauza zgomotului iar obstacolul care este detectat nu trebuie să fie validat. Se consideră un prag $T_{points}=10$ care definește numărul minim de vectori de mișcare asociați punctelor de interes ai obstacolului. Astfel, un obstacol i este valid dacă condiția următoare este adevărată:

$$\|KeypointsSet\|_i > T_{points} \quad (2.28)$$

2.3.6. Detecția coliziunii iminente cu un obstacol

Obstacolele valide detectate cu ajutorul metodei descrisă în secțiunile anterioare urmează a fi clasificate în funcție de gradul lor de coliziune iminentă cu ego-vehiculul. Întrucât este dificilă stabilirea unei reguli generale, se consideră următoarele poziții relative ale obstacolelor față de cameră, care definesc două cazuri:

- Obstacolul se apropie de cameră cu un grad ridicat de coliziune iminentă;
- Obstacolul fie se apropie, fie se depărtează de cameră cu un grad scăzut de coliziune iminentă.

Se poate defini poziția fiecărui obstacol în imagine ca fiind dată de coordonatele centrului de masă (x_{center} , y_{center}). Aria acestuia este aproximată cu aria dreptunghiului circumscris punctelor lui de interes. Acesta este definit de colțurile (x_{min} , x_{max} , y_{min} , y_{max}). Luând în considerare vederea în perspectivă din viziunea monoculară, partea inferioară a obstacolului care apare în partea de jos a imaginii este considerată pentru a evalua riscul de coliziune iminentă. Astfel centrul obstacolului este definit de următoarele ecuații:

$$x_{center} = (x_{min} + x_{max}) * \frac{1}{2}$$

$$y_{center} = (y_{min} + y_{max}) * \frac{2}{3}$$
(2.29)

Elementele din scenă care sunt relativ departe, apar în partea superioară a imaginii și nu reprezintă un pericol de coliziune iminentă. Imaginea scenei este împărțită în trei părți orizontale identice. Treimea inferioară reprezintă regiunea cu risc ridicat de coliziune iminentă (Figura 2.33). Un obstacol este considerat că intră în această regiune dacă următoarea inegalitate este verificată:

$$y_{center} < y_{image}, \text{ unde } y_{image} = height_{image} * \frac{2}{3}$$
(2.30)



Figura 2.33. Zona de risc ridicat – treimea inferioară (marcată cu puncte roșii)

Nu este obligatoriu ca orice obstacol ce intră în zona de risc ridicat (Figura 2.33) să cauzeze o potențială coliziune iminentă. Punctul critic este definit în Figura 2.33 prin simbolul „X” situat pe ultima linie din imagine, în centrul acesteia. Pentru a evalua gradul de pericol se utilizează vectorul de mișcare global al obstacolului. Se definește pentru fiecare obstacol direcția de mișcare critică, ce determină gradul maxim de pericol, ca fiind dreapta ce unește punctul critic de punctul de centru calculat anterior al obstacolului.

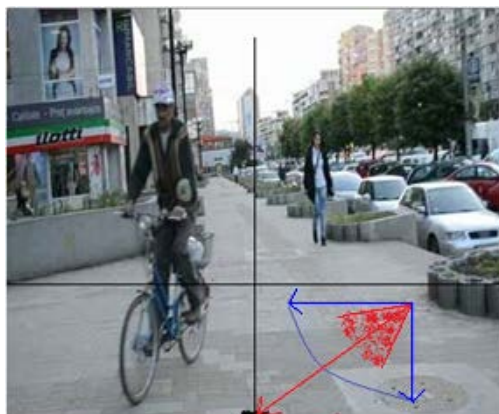


Figura 2.34. Zona unghiulară cu risc ridicat de coliziune, în jurul direcției de mișcare critică

Dacă mișcarea obstacolului are o abatere mică față de direcția de mișcare critică atunci el este considerat ca având un grad ridicat de posibilă coliziune iminentă. Pentru a lua această decizie a fost setat experimental un prag $T_{angdev}=15^\circ$. Un obstacol i este în zona unghiulară critică dacă unghiul lui de mișcare este în preajma direcției critice, adică se respectă următoarea inegalitate:

$$|Angle(Obstacle_i) - Angle(CriticalDir)| < T_{angdev} \quad (2.31)$$

2.3.7. Rezultate experimentale

Sistemul de detecție a obstacolelor pe baza analizei vectorilor de mișcare relativă a fost validat în două mari scenarii:

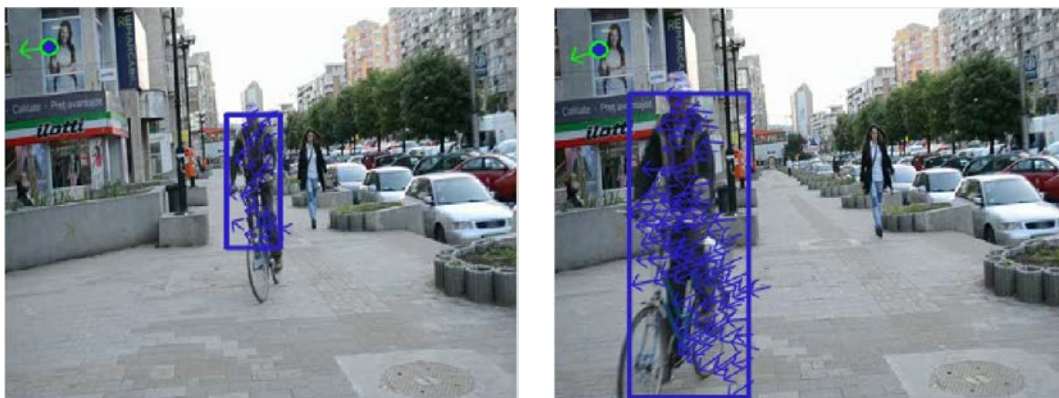
- Cameră fixă și obstacole în mișcare;
- Cameră în mișcare și obstacole fie fixe, fie aflate în mișcare.

Evaluarea s-a realizat pe două secvențe video din diverse scenarii de trafic, conținând mai mult de o mie de cadre consecutive. Rezoluția acestora este de 352x288 pixeli. Parametrii de execuție sunt cei care au fost setați și prezentați în secțiunile anterioare. Sistemul rulează în timp real, la cca. 20 cadre/s pe un procesor Intel Core i5@2,5 GHz. Performanțele sistemului au fost calculate în termenii ratei de detecții adevărate dar și de detecții false. Acestea sunt prezentate în Tabelul 2.2.

Tabelul 2.2. Acuratețea sistemului de detecție a obstacolelor cu mișcare relativă

	Rata de detecții adevărate	Rata de detecții false
Secvența 1 (1550 cadre)	76%	0%
Secvența 2 (1550 cadre)	87%	2%
GLOBAL	81.5%	1%

În secvența 1, camera a fost fixă iar obiectele din scenă erau în mișcare. În secvența 2, atât camera cât și obstacolele din scenă au fost fie fixe, fie în mișcare. Rezultate ale detecției sunt prezentate în Figura 2.35 și Figura 2.36.



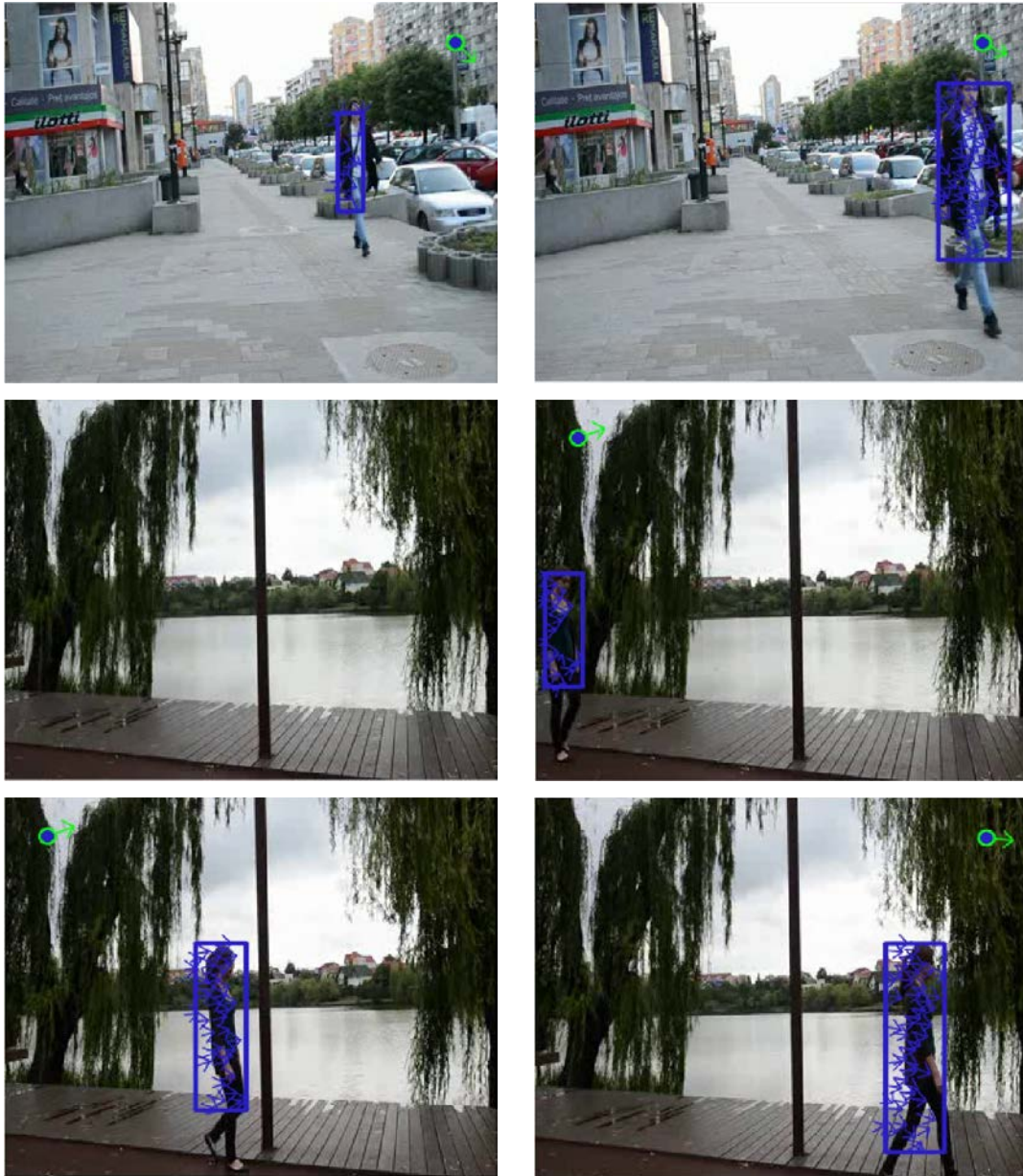


Figura 2.35. Detecția obstacolelor în secvența 1 (cameră fixă și obstacole în mișcare)

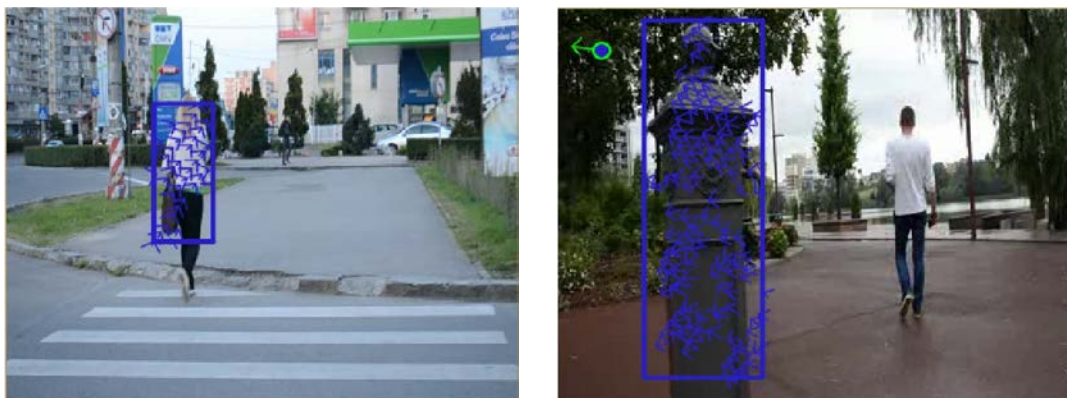


Figura 2.36. Detecția obstacolelor în secvența 2 (stânga – cameră în mișcare, obstacol în mișcare; dreapta – cameră în mișcare, obstacol fix)

2.4. Concluzii

În acest capitol au fost prezentate trei metode pentru detecția obstacolelor în scene de trafic.

În secțiunea 2.1 este definită o metodă de detecție a obstacolelor pe baza segmentării scenei în superpixeli. Aceasta beneficiază de avantajele folosirii unei configurații de imagini de stereoviziune în niveluri de gri și a unei stereo-reconstrucții de calitate pentru detecția obstacolelor. Algoritmul SLIC realizează o segmentare a scenei în superpixeli pe baza intensităților pixelilor din imagine. Viteza lui de execuție oferă posibilitatea utilizării altor metode ulterioare de procesare pentru detecția obstacolelor cu menținerea timpului real de procesare. S-a definit astfel o metodă nouă care constă în combinarea informației de intensitate, profunzime și de mișcare cu scopul realizării unei grupări precise a obstacolelor pe baza superpixelilor. Rezultatele arată o detecție foarte bună a obstacolelor cu segmentarea precisă a suprafețelor lor, ceea ce este deosebit de util pentru procesele ulterioare de detecție [64].

În secțiunea 2.2 este definită o metodă de detecție a obstacolelor din imagini de stereoviziune în niveluri de gri, pe baza calculului spațiilor de disparitate UV . S-a definit un algoritm de etichetare și modalitatea de validare a obstacolelor prin analiza spațiului de disparitate U . Acestea sunt combinate cu metoda Hough de detecție a suprafeței drumului din spațiul de disparitate V . Metoda oferă rezultate suficient de bune de detecție a regiunilor de interes care pot să fie utilizate ulterior pentru clasificare. Totuși aceasta are dezavantajele determinate de presupunerea suprafeței plane a drumului și de faptul că nu descrie obstacolele prin forma lor ci doar prin regiunea de interes rectangulară care le conține. Din cauza faptului că nu este utilizată informația de mișcare, apare posibilitatea detecției eronate a mai multor obstacole ca fiind unul singur în situațiile în care acestea sunt relativ apropiate în scenă [65].

În secțiunea 2.3 este definită o metodă de detecție a obstacolelor pe baza analizei vectorilor de mișcare relativă din secvențe de imagini color monoculare. Vectorii de mișcare sunt calculați între cadrele succesive, pe un set de puncte de interes distribuite uniform în imagini. Mișcarea fundalului este eliminată și vectorii aparținători potențialelor obstacole sunt grupate definind o abordare bazată pe algoritmul de grupare ierarhică. Obstacolele sunt validate și în final este definit un algoritm pentru semnalizarea unei eventuale coliziuni iminente cu acestea. Metoda beneficiază de avantajul informației de culoare care oferă rezultate superioare în calculul vectorilor de mișcare. Sistemul de detecție are avantajul costurilor reduse, putând fi implementat și pe dispozitive mobile. Obstacolele din scenă sunt detectate și semnalizate în cazul unei potențiale coliziuni iminente, dar nu sunt segmentate precis [66].

Analizând comparativ cele trei soluții pentru detecția obstacolelor se constată că, din punct de vedere ale performanțelor detecției obstacolelor și segmentării acestora în scenă indiferent de costurile și puterea de procesare necesare, cele mai bune rezultate sunt obținute utilizând metoda prezentată în 2.1 în care se evidențiază avantajele utilizării stereoviziunii, a segmentării imaginii pe baza superpixelilor și gruparea lor prin fuziunea informațiilor stereo (intensitate și profunzime) cu cele de mișcare.

3. Recunoașterea obstacolelor din scenele de trafic

Procesul de detecție a obstacolelor este urmat de obicei de cel de recunoaștere a lor. Dacă în cadrul detecției obstacolelor sunt găsite regiunile în care apar acestea în scena de trafic, în cadrul recunoașterii se va încerca identificarea sau mai precis clasificarea lor. În mod tradițional, clasificarea se face pe baza unor caracteristici (trăsături) specifice fiecărei clase cu ajutorul cărora se poate identifica orice obstacol din scenă. Sistemele de asistare a conducătorului auto și de conducere autonomă care integrează module de detecție și recunoaștere a obstacolelor au integrate de obicei și un modul de urmărire a lor cu rolul de a păstra coerența rezultatelor obținute de-a lungul procesării cadrelor succesive de imagini. Varietatea scenariilor de trafic în care sunt prezente multe tipuri de obstacole face procesul de recunoaștere unul dificil și complex și în care obținerea de rezultate de înaltă acuratețe sunt greu de atins.

Dacă în scenariile de autostradă, recunoașterea obstacolelor nu este foarte dificilă, în scenariile de trafic urban aceasta este mult mai complexă din cauza numărului mai mare de clase de obstacole prezente și a scenelor mult mai aglomerate. În aceste scenarii de trafic urban pietonii joacă cel mai important rol. Ei sunt cei mai vulnerabili participanți la trafic, astfel încât recunoașterea lor este una deosebit de importantă. Pe lângă facilitățile clasice de avertizare a conducătorului auto, există unele autovehicule inteligente care au în dotare accesorii speciale cum sunt airbag-urile externe sau capota mobilă care se declanșează automat sau respectiv își schimbă orientarea automat în momentul iminent al unui impact cu un pieton care nu mai poate fi evitat. Acestea au menirea de a reduce consecințele unui impact grav cu pietonii care ar putea duce la răniri sau chiar decese. Recunoașterea autovehiculelor este destul de importantă cu scopul ulterior de a găsi zonele în care sunt localizați pasagerii. În cazul unui impact cu un autovehicul sistemul de asistare va încerca evitarea coliziunii cu aceste zone și va devia ego-vehiculul pentru a lovi altă zonă mai puțin periculoasă a autovehiculului. Recunoașterea stâlpilor în trafic poate avea aplicații ulterioare în primul rând pentru a evita un eventual impact care ar putea fi deosebit de grav, iar apoi pentru a determina poziția culoarului navigabil (a benzilor de circulație navigabile) în raport cu aceștia și cu celelalte obstacole detectate. De asemenea, identificarea stâlpilor poate restrânge zona de căutare pentru module care detectează și clasifică semne de circulație sau chiar semafoare în trafic. Caracteristicile ce vor fi defini obstacolele trebuie să fie cât mai discriminante inter-clase și cât mai asemănătoare intra-clase, iar recunoașterea lor și în special a pietonilor trebuie să se facă cu o acuratețe cât mai ridicată, într-un timp util (procesări de timp real) și cu robustețe ridicată în diferitele scene de trafic și condiții de mediu.

Recunoașterea obstacolelor sau clasificarea obstacolelor se referă la asignarea unei clase fiecărui obstacol din scenă. Întreg procesul de clasificare [67] are în general câțiva pași de bază: preprocesarea imaginilor achiziționate în scopul eliminării zgomotului prin aplicarea de diverse filtre; detecția zonelor cu obstacole și extragerea de trăsături relevante pentru caracterizarea lor; antrenarea unor clasificatori pe baza trăsăturilor selectate; clasificarea propriu-zisă în care fiecărui obstacol detectat îi este asignată o clasă predefinită. Cele mai întâlnite tehnici de clasificare pentru recunoașterea

obstacolelor în lucrările din literatura de specialitate sunt: rețelele neuronale, arborii decizionali, mașinile cu vectori suport și clasificatorii fuzzy [68].

Trăsăturile folosite în general pentru caracterizarea obstacolelor sunt cele de mișcare [69-72], de apariție vizuală cu impunerea unor constrângeri geometrice [73-76], dar și cele vizuale cu construirea unor dicționare de trăsături [77-80]. Pentru obținerea de rezultate mai bune, se combină trăsăturile extrase din imagini cu cele provenite de la alți senzori [81-84]. În cazul special al recunoașterii pietonilor se utilizează adesea trăsături vizuale de formă [85, 86], textură [87], trăsături *wavelet* [88], trăsături de mișcare [89] și modele de descriere a formei prin contururi sau muchii [90]. Fuziunea trăsăturilor vizuale cu trăsături provenite de la alți senzori (RADAR, LIDAR, infraroșu) aduce și în acest caz o îmbunătățire a recunoașterii pietonilor [91-94].

Acuratețea clasificării este influențată direct de incertitudini și de erorile care se propagă în lanțul procesării imaginilor de la informațiile primare (calitatea imaginilor achiziționate, a stereo-reconstrucției etc.) până la momentul clasificării (detecția precisă a regiunilor candidate, trăsături discriminante, algoritmi robuști, etc.) și clasificarea propriu-zisă. Adesea este dificilă selecția unui clasificator potrivit pentru a se rezolva o anumită problemă. De aceea se studiază și se implementează diferiți clasificatori variind parametrii lor și se alege acela care oferă rezultatele cele mai bune cu setările corespunzătoare. Combinarea mai multor clasificatori se dovedește a fi foarte utilă, oferind rezultate superioare. Identificarea legăturilor slabe din acest lanț al procesărilor și reducerea incertitudinilor sunt critice pentru îmbunătățirea acurateței clasificării.

În acest capitol sunt definite mai multe metode pentru recunoașterea pietonilor și a altor clase de obstacole în scenariile de trafic. Unele metode de recunoaștere sunt bazate pe rezultatele furnizate de modulele de detecție a obstacolelor prezentate în capitolul anterior. Utilizarea acestor rezultate, care constau în regiunile de interes cu obstacole, restrâng spațiul de căutare, sporind viteza de clasificare a obstacolelor în scena de trafic.

În secțiunea 3.1 este definită o metodă pentru recunoașterea pietonilor în scene de trafic. Aceasta utilizează conturul integral al obstacolului ca trăsătură discriminantă, utilizând ierarhiile arborescente de modele de contururi de pietoni în procesul de potrivire a conturilor ipotezelor de pietoni, cu rolul măririi vitezei de procesare.

În secțiunea 3.2 este definită o metodă robustă de clasificare a obstacolelor prin definirea și selecția de trăsături multiple care includ scorul de potrivire a conturilor pietonilor descris în secțiunea 3.1. Sunt analizați și antrenați mai mulți clasificatori pe aceste trăsături cu scopul clasificării obstacolelor din scenele de trafic în patru clase: pietoni, autovehicule, stâlpi/copaci, alte obstacole. Ulterior este definită o metodă de urmărire a rezultatelor clasificării de-a lungul cadrelor succesive în secvențele de imagini cu scopul creșterii acurateței rezultatelor clasificării. În final este definită o metodă bazată pe construirea și utilizarea dicționarelor de trăsături vizuale cu scopul clasificării obstacolelor în aceleași patru clase.

3.1. Recunoașterea pietonilor folosind trăsături de contur și ierarhii de modele de contururi

Conturul este o trăsătură puternică pentru detecția pietonilor, întrucât elimină multe din problemele cauzate de varietatea îmbrăcăminte pe care o poartă aceștia. Conturul descrie forma pietonilor, fiind invariant la varietatea îmbrăcăminte care determină texturi diferite în imagini. Ideea de bază este de a avea o colecție de contururi de pietoni, având diverse atitudini și poziții care să acopere spațiul de căutare și care să poată fi folosite ulterior ca și modele într-un proces de potrivire. Contururile trebuie să descrie cât mai precis forma reală al pietonilor. De aceea, pentru a obține o descriere cât mai bună, acestea pot fi extrase prin metoda de eliminare a fundalului din secvențe de imagini cu fundal fix, achiziționate într-un mediu controlat.

Utilizarea ierarhiilor de modele de contururi de pietoni face ca timpul necesar procesului de potrivire a unui contur de obstacol la setul de modele de contururi de pietoni să se reducă considerabil, având posibilitatea obținerii de rezultate în timp real. Acest lucru se datorează reducerii numărului de potriviri prin parcurgerea și eliminarea potrivirilor cu unele ramuri ale ierarhiei arborescente. În general, ierarhiile utilizate în literatura de specialitate sunt fie arbori binari, fie arbori multicăi care au un număr constant de fii ai fiecărui nod [95, 96].

3.1.1. Extragerea conturului pietonilor

Pentru a putea obține un timp real de execuție trebuie utilizați un număr cât mai redus de pixeli în procesul de potrivire. De aceea, conturul pietonilor are un mare avantaj în procesul de potrivire, având un număr redus de pixeli, comparativ cu întreaga regiune de interes din imagine. Astfel, s-a definit un algoritm care să extragă conturul întregului pieton, atât din secvențe cu fundal fix (utilizate pentru generarea modelelor de contururi de pietoni), cât și din secvențe de stereoviziune cu fundal variabil (utilizate atât ca și modele de contururi de pietoni cât și pentru extragerea conturilor candidate a obstacolelor ce urmează a fi recunoscute).

Datele de intrare pentru algoritm constau în imaginea alb-negru a unui obstacol din scenă care poate fi formată din mai multe componente conexe separate. Rezultatul oferit de algoritm este o secvență de puncte de contur care descrie de jur-împrejur forma obstacolului respectiv. Algoritmul are patru mari pași:

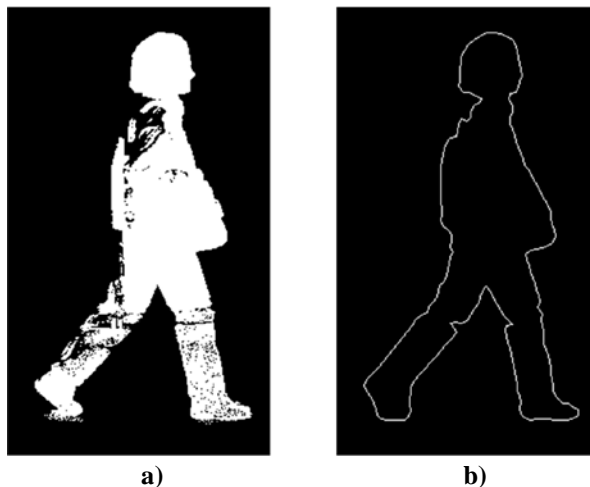
- Determinarea tuturor conturilor exterioare componentelor conexe care sunt conținute în imaginea obstacolului. Imaginea de lucru este o imagine alb-negru în care pixelii albi sunt aparținători pietonilor, iar cei negri sunt aparținători fundalului. Contururile sunt extrase utilizând un algoritm clasic de urmărire a marginilor obiectelor [97].
- Calcularea minimului distanțelor dintre toate conturile valide extrase. Distanțele sunt calculate cu ajutorul metricii Euclidiene între toate perechile de puncte de pe contururi. Distanța minimă dintre două contururi

este valoarea cea mai mică dintre toate distanțele perechilor de puncte (un punct din primul contur și celălalt din al doilea contur)

- Obținerea unei singure componente conexe prin unificarea tuturor conturilor componentelor conexe. Ordinea de unificare este dată de ordinea crescătoare a valorilor distanțelor dintre ele. Unificarea are loc între punctele de contur ale tuturor perechilor de componente conexe între care distanța este mai mică decât un prag determinat experimental. Rezultatul acestui pas este formarea unei singure componente conexe care aproximează obstacolul.
- Urmărirea conturului acestei singure componente conexe care oferă un singur contur aproximat al obstacolului și care reprezintă rezultatul final al algoritmului.

Contururile modelelor de pietoni în scene cu fundal fix

Pentru extragerea conturilor de modele de pietoni se consideră un mediu controlat cu fundal fix, în care există pietoni care se mișcă. Prin procedeul de eliminare a fundalului fix rezultă imaginile de confidență a pietonilor. Acestea definesc suprafețele (cu pixeli albi) în care există pietoni în scena cu fundal fix (pixeli negri). Pe aceste imagini este aplicat algoritmul descris anterior cu scopul obținerii unui singur contur continuu pentru fiecare pieton (Figura 3.1).



**Figura 3.1. Extragerea conturului modelelor de pietoni din scene cu fundal fix:
a) imaginea de confidență; b) conturul extras**

Contururile modelelor de pietoni și a obstacolelor în scene cu fundal variabil

Pentru extragerea atât a modelelor de contururi de pietoni cât și a conturilor obstacolelor candidate a fi pietoni se pot utiliza secvențe de imagini de stereoviziune cu fundal variabil.

Primul pas constă în delimitarea obstacolelor în imaginea de intensitate, lucru realizat fie manual fie este utilizat un algoritm de segmentare a regiunilor de interes cu obstacole din scena de trafic. Este deosebit de important ca regiunea de interes

rectangulară care circumscrie obstacolul în imagine să conțină cât mai puțin fundal (Figura 3.2b).

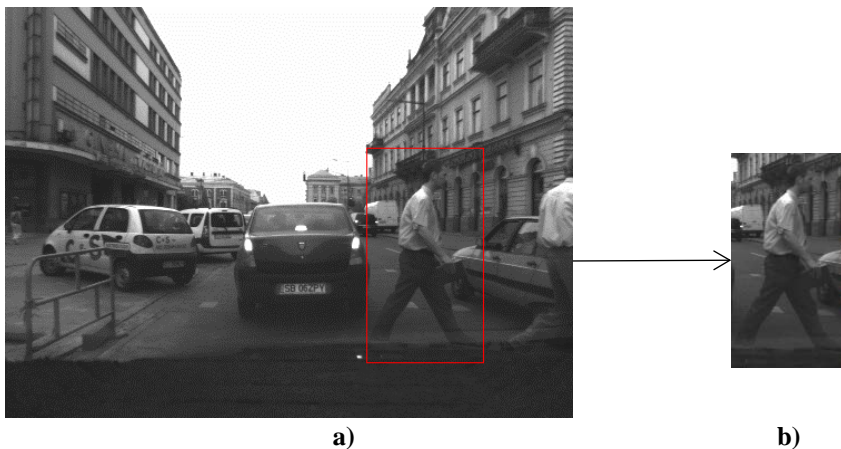


Figura 3.2. Selecția regiunii de interes a obstacolului:
a) imaginea scenei și zona de interes (cu culoare roșie); b) regiunea de interes

În regiunea de interes definită se calculează histograma profunzimilor. Valorile de profunzime sunt cunoscute în urma procesului de stereo-reconstrucție. În calculul acestei histogramme sunt ignorate valorile de 0 (pixeli 3D nereconstruiți) și infinit (puncte aflate foarte departe). Valorile histogrammei sunt apoi normalizate în intervalul $[0, 1]$. Aceasta este utilizată pentru a putea separa suprafața pietonului de suprafața fundalului (Figura 3.3b). Se definește un procent de abatere de 5% față de vârful de maxim iar acei pixeli care se află la acea distanță plus sau minus valoarea procentului sunt considerați că aparțin obiectului. Restul pixelilor sunt eliminați întrucât se consideră că aparțin fundalului. În urma acestui pas se obține o imagine binară în care pixelii albi sunt pixeli aparținători obiectului iar cei negri sunt considerați că fac parte din fundal (Figura 3.4a). Pasul următor constă în extragerea conturului pe imaginea binară, algoritmul descris anterior.

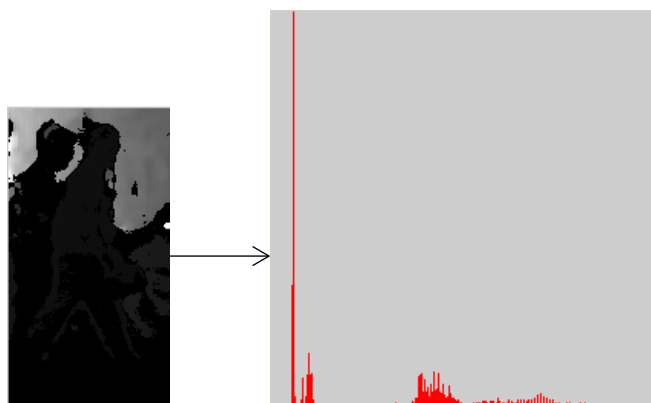


Figura 3.3. Calculul histogrammei valorilor de profunzime: imaginea de profunzime asociată regiunii de interes (stânga); b) histograma profunzimilor (dreapta)



Figura 3.4. Extragerea conturului obstacolului: puncte aparținătoare obstacolului (stânga); conturul extras (dreapta)

Pentru a spori calitatea conturului extras se aplică un proces de rafinare a acestuia. Conturul extras inițial are problemele abaterii de la forma obiectului datorită erorilor care pot apărea în reconstrucția punctelor 3D sau datorită lipsei punctelor 3D din anumite zone. Soluția este de a exploata informația din imaginea de intensitate 2D în completarea informației 3D. Abordarea constă în găsirea conturului exterior (determinat anterior) și rafinarea acestuia pe baza muchiilor. O problemă care apare în acest caz este determinată de absența muchiilor, caz în care este imposibil în a decide unde este marginea reală a obiectului respectiv.

În acest proces de îmbunătățire a conturilor extrase ale obiectelor s-a aplicat un pas de pre-procesare constând în efectuarea unei filtrări cu metoda NL-means [98]. Algoritmul calculează asemănările în cadrul fiecărui obiect, pentru fiecare pixel, într-o zonă vecină lui. Procedul reduce zgomotul din imagine și păstrează detaliile din cadrul acesteia. Filtrarea elimină muchiile zgomotoase păstrând și rafinând muchiile importante (Figura 3.5). Conturul este apoi retrasat ținând cont de lungimea și orientarea muchiilor, din preajma conturului inițial, extrase cu metoda Canny [99] pe imaginea filtrată.

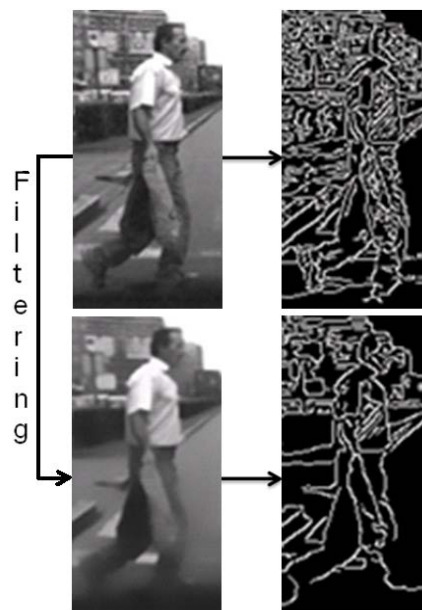


Figura 3.5. Imagine de intensitate originală și filtrată cu algoritmul NL-means (stânga); muchiile corespunzătoare extrase (dreapta)

Exemplificarea rafinării conturului pe baza muchiilor și a rezultatului final obținut sunt prezentate în Figura 3.6 respectiv Figura 3.7.

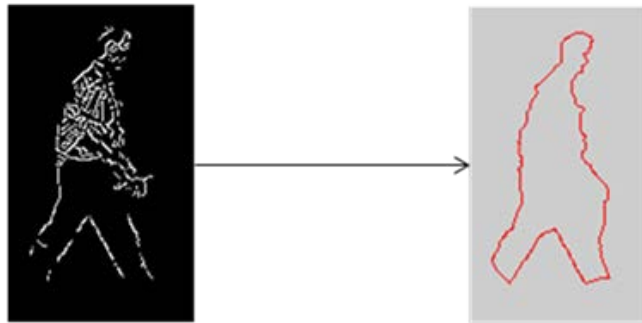


Figura 3.6 Stânga: imaginea de muchii a imaginii filtrate a obiectului; Dreapta: conturul rafinat pe baza muchiilor



Figura 3.7 Conturul final extras al obiectului exemplificat (un pieton din scenă)

3.1.2. Construirea ierarhiei cu modele de contururi de pietoni

Ierarhia de modele de contururi de pietoni este foarte importantă în procesul de detecție a pietonilor bazat pe potrivirea formelor (contururilor) întrucât reduce considerabil numărul de potriviri dintre contururile obstacolelor candidate din scenă și întregul set de modele de contururi de pietoni. Aceasta are structura unui arbore multicăi, fiecare nod reprezentând un model de contur de pieton obținut prin utilizarea metodelor descrise anterior.

Pentru construirea acesteia se utilizează un algoritm de grupare nesupervizată ce are la bază o variantă modificată (Figura 3.8) a algoritmului *k-means* [58]. Acesta este aplicat pentru construirea ierarhică a fiecărui nivel începând de la frunzele arborelui și continuând până la rădăcină. La fiecare nivel, datele de intrare ale algoritmului constau într-un set de modele de contururi de pietoni, frunzele conținând întregul set de contururi extrase. Rezultatul algoritmului constă într-un număr de grupuri de contururi, fiecare grup conținând acele modele care sunt asemănătoare și un prototip (centru de grup).

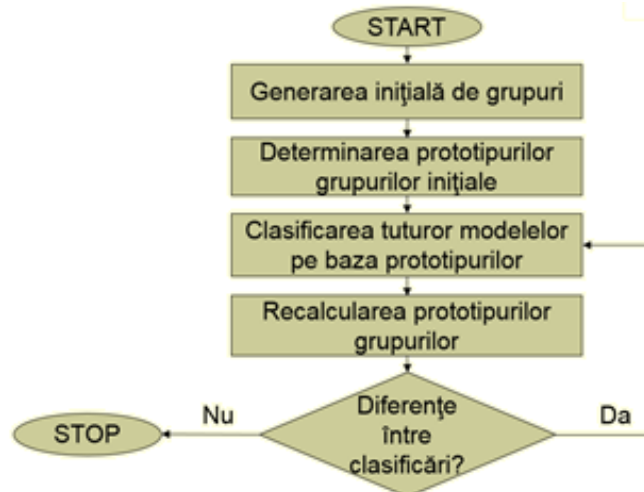


Figura 3.8. Schema bloc a algoritmului de construire a nivelurilor din ierarhia cu modele de contururi de pietoni

Pașii algoritmului definit pentru construirea ierarhiei cu modele de contururi de pietoni sunt:

- Calculul matricei de distanțe dintre toate modelele de contururi de pietoni. Considerând un număr N de contururi de pietoni, aceasta este o matrice de dimensiuni $N \times N$ în care fiecare element de pe poziția (i, j) reprezintă distanța Euclidiană de la conturul i la conturul j
- Generarea inițială de grupuri pe baza unei euristici definite. Se consideră inițial că nici un model nu aparține niciunui grup de contururi. Se ordonează apoi crescător toate distanțele din matricea calculată anterior. Urmează un proces iterativ: pornind de la cea mai mică distanță dintre două contururi și continuând până la cea mai mare din șirul ordonat crescător, sunt asignate aceluiași grup sau formează un grup nou dacă cel puțin un contur din cele două nu era aparținător niciunui grup până în acel moment. Rezultatul acestui pas este un set de grupuri de modele de contururi de pietoni.
- Se calculează prototipul fiecărui grup determinat anterior. Acesta este dat de acel model din grup care face ca cea mai mare distanță dintre el și restul modelelor din grup să fie minimă.
- Se asignează fiecare model de contur de pieton la unul din prototipurile calculate anterior față de care distanța lui este minimă.
- Se formează astfel grupuri noi, fiecare având câte un prototip. Se recalculază prototipul fiecărui grup și se repetă acești ultimi doi pași până la convergență (când nu mai există diferențe între ultimele două clasificări – asignări ale modelelor la grupuri).
- Rezultă un set stabil de grupuri de contururi de pietoni la nivelul curent din ierarhie, fiecare având desemnat câte un prototip. Prototipurile nivelului curent sunt considerate date de intrare pentru nivelul superior. Algoritmul este aplicat succesiv pe niveluri în sus până se ajunge la un singur model care reprezintă rădăcina arborelui multicăi, deci a ierarhiei.

Rezultatele experimentale ale metodelor de extragere a contururilor și a construirii ierarhiei cu modele ideale de contururi de pietoni au fost obținute prin rularea algoritmilor pe secvențe de imagini cu mii de cadre, în diferite scenarii de trafic. Calitatea extragerii contururilor pietonilor și a modului de grupare a acestora în ierarhie este ridicată.



a) b) c) d)
Figura 3.9. Contururi de pietoni extrase din secvențe de imagini cu fundal fix: a), c) imagini de încredință ale pietonilor; b), d) conturul corespunzător extras

În b) c) d)
 Figura 3.9 sunt prezentate două exemple de rezultate ale extragerii contururilor pietonilor din imaginile de încredință. Algoritmul de extragere a conturului oferă rezultate precise, eliminând golurile interioare și unificând componentele conexe din imaginile de încredință, rezultând un singur contur integral pentru fiecare pieton.



Figura 3.10. Scene cu fundal variabil și extragerea contururilor pietonilor (cu culoare roșie) în regiunile selectate (dreptunghiuri de culoare roșie)

În Figura 3.10 sunt prezentate patru situații de extragere a conturilor pietonilor din scene cu fundal variabil. Algoritmul de extragere și procedura de rafinare oferă rezultate bune care aproximează conturul real al pietonilor, în condițiile în care imaginile achiziționate ale scenelor de trafic sunt în nuanțe de gri.

Ierarhia construită conține 1872 de modele de contururi de pietoni (936 modele originale și încă 936 obținute prin rotirea celor originale în jurul axei verticale – obținându-se o direcție inversă de mers). Modelele conțin forma integrală a pietonilor (din cap până la picioare) și descriu diferite atitudini ale acestora. Modelele de contururi de pietoni au fost obținute prin procedeul de eliminare al fundalului în zonele pietonilor din diverse secvențe de imagini cu fundal fix. Imaginile pietonilor extrași au fost ajustate manual, eliminându-se micile imperfecțiuni, iar în final s-a aplicat algoritmul de extragere a conturului, rezultând modelele de contururi de pietoni.

În procesul de construire a ierarhiei aceste modele au fost redimensionate la o mărime constantă având factorul de formă specific pentru pietoni. Înălțimea și lățimea modelelor redimensionate au fost stabilite la 150 respectiv 71 pixeli, rezultând un factor de formă de aproximativ $1/2,1$. Aceste valori de referință au fost obținute prin medierea inițială a tuturor înălțimilor și lățimilor modelelor de contururi. În Figura 3.11 sunt prezentate câteva dintre modelele de contururi de pietoni redimensionate care sunt conținute de ierarhie.

Ierarhia de modele de contururi de pietoni este utilizată în algoritmul de potrivire în timp real, obținându-se scorul de potrivire care determină dacă o ipoteză de pieton (obiect din scenă) este sau nu este pieton.



Figura 3.11 Exemple de modele de contururi de pietoni redimensionate existente în ierarhie

Ierarhia care se obține este un arbore multicăi. Fiecare grup de modele (Figura 3.12) are un număr variabil de membri și un prototip care este modelul reprezentativ al acelu grup. Rădăcina structurii arborescente este prototipul singurului grup aflat la nivelul cel mai de sus.

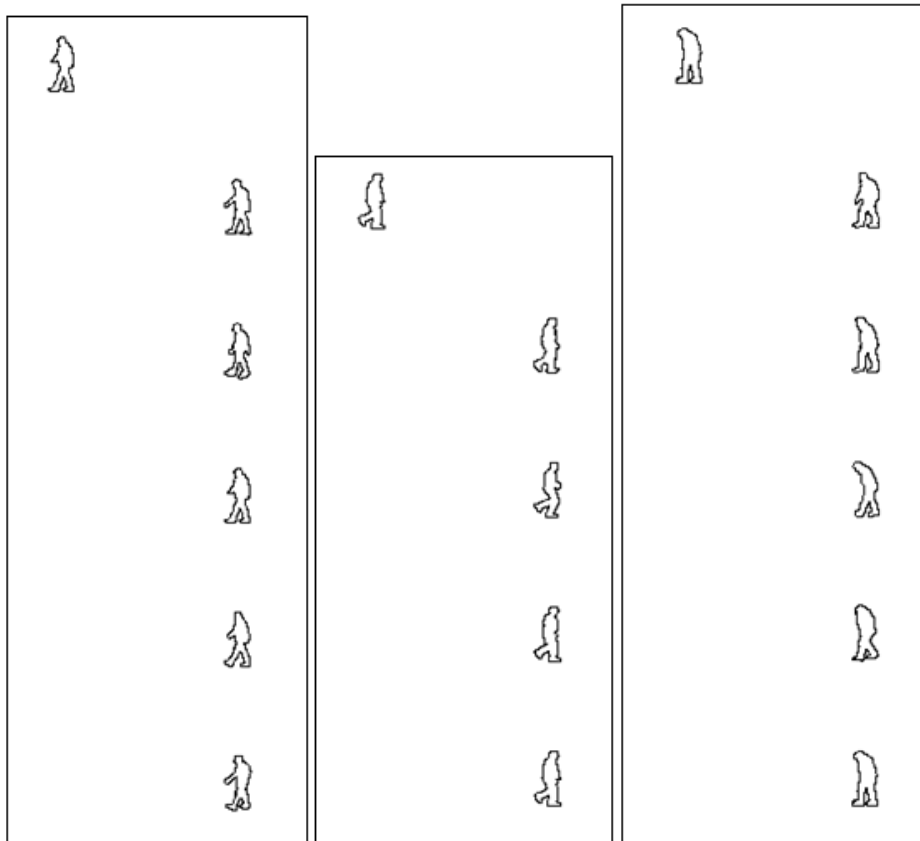


Figura 3.12 Exemplu de trei grupuri din cadrul ierarhiei de modele de contururi de pietoni (în stânga sus este reprezentat prototipul fiecărui grup; în coloana din dreapta sunt modelele din cadrul fiecărui grup)

Algoritmul de potrivire a modelelor cu obiectele din scenă este pornit în ierarhie dinspre rădăcină înspre frunze (Figura 3.13), memorând cel mai bun scor de potrivire și alegând întotdeauna pentru continuare ramura pe care se obține scorul minim de potrivire (cea mai bună potrivire).

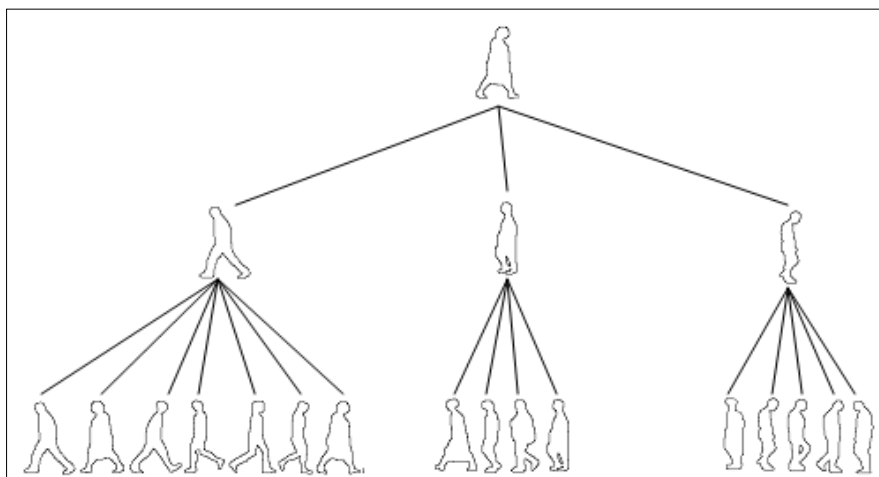


Figura 3.13 Exemplu de mini-ierarhie de contururi de pietoni

3.1.3. Recunoașterea pietonilor pe baza trăsăturilor de contur

În această secțiune este definit și prezentat un sistem împreună cu modulele și algoritmi necesari, bazați pe trăsăturile de contur, pentru recunoașterea pietonilor în scene de trafic.

Conturul întregului corp este exploatat ca o trăsătură puternică în procesul de potrivire cu un tip de ierarhie cu modele de contururi de pietoni descrisă în secțiunea 3.1.2. Prin utilizarea conturilor se realizează o descriere completă a formei pietonilor care oferă rezultate bune pentru recunoaștere.

Arhitectura sistemului de recunoaștere a pietonilor

În Figura 3.14 sunt prezentate modulele componente și fluxul de date din sistemul de recunoaștere a pietonilor pe baza trăsăturilor de contur. În schema bloc se evidențiază modulul de extragere a conturilor obiectelor din scenă (Figura 3.16) și cel de clasificare (Figura 3.17) cu rolul de a asigura fiecăruia clasa pieton sau non-pieton.

Pornind de la imaginile de intensitate de dimensiuni 512x383 pixeli, în niveluri de gri, achiziționate cu sistemul de stereoviziune și utilizând o implementare hardware [53] pentru stereo-reconstrucție, se obțin valorile profunzimilor punctelor 3D reconstruite.

Se utilizează un algoritm de detecție a obstacolelor cu rolul determinării regiunilor de interes 3D din scenă care ar putea fi ipoteze de pietoni (Figura 3.15). Toate punctele aparținătoare fundalului sunt eliminate din cadrul fiecărei regiuni de interes pe baza informației de profunzime.

Se etichetează apoi fiecare obstacol pe baza proiecțiilor punctelor 3D aparținătoare lor în imaginea stângă (Figura 3.18). Algoritmul descris în secțiunea 3.1.1 pentru extragerea conturului este aplicat pe fiecare obstacol, obținându-se câte un contur continuu pentru fiecare dintre ele. Acesta este ulterior rafinat pe baza muchiilor din vecinătatea lui extrase pe imaginea din imaginea de intensitate, care a fost în prealabil filtrată cu ajutorul algoritmului NL-means [98].

Pasul final constă în potrivirea conturului ipotezelor de pietoni cu scopul determinării dacă obstacolele respective sunt sau nu pietoni. Potrivirea se realizează între conturul ipotezei și ierarhia cu modele de contururi de pietoni (secțiunea 3.1.2). Aceste module și algoritmi sunt prezentați în detaliu în secțiunile următoare.

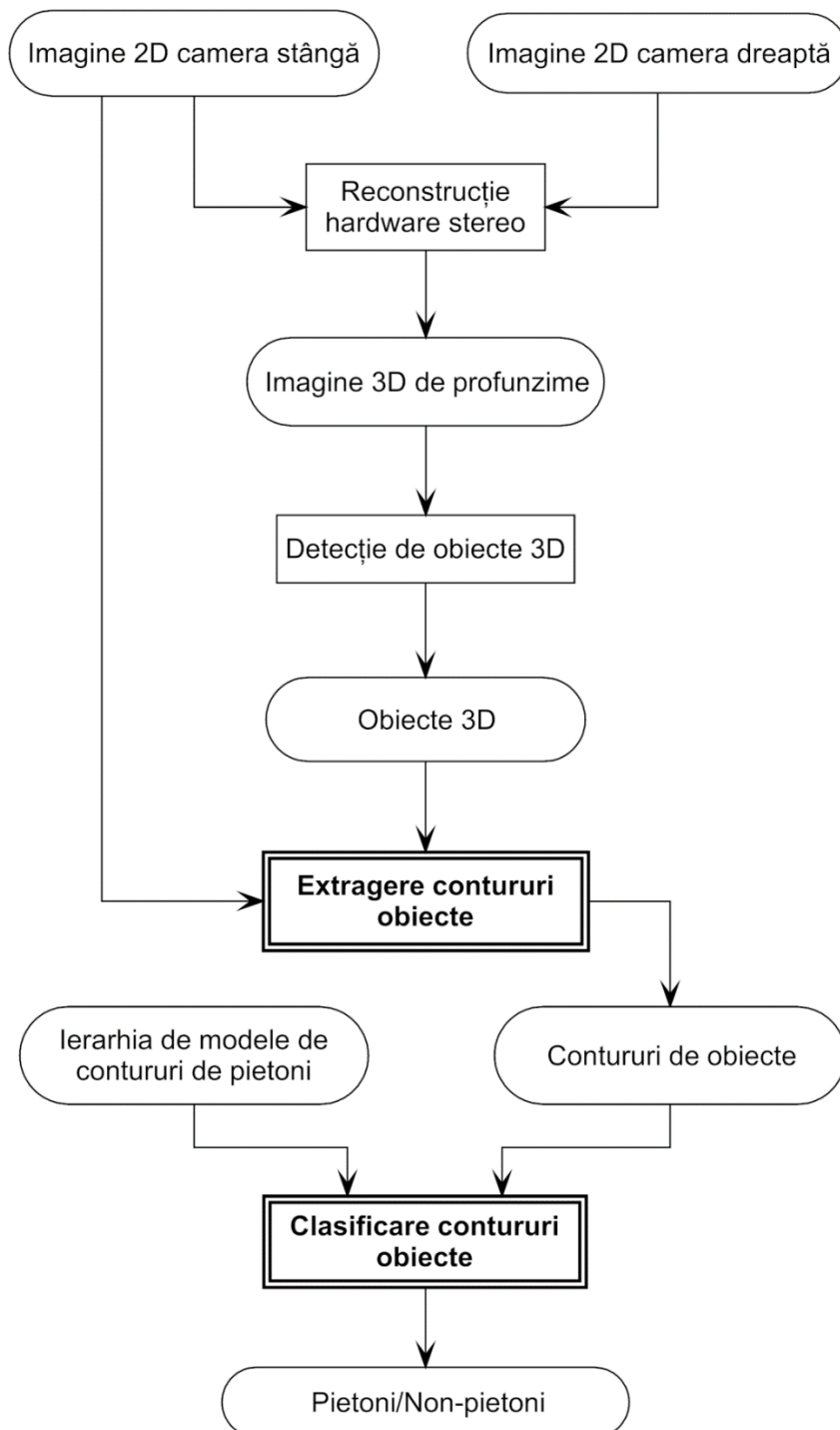


Figura 3.14. Arhitectura generală a sistemului de detecție a pietonilor pe baza clasificării trăsăturilor de contur ale obiectelor

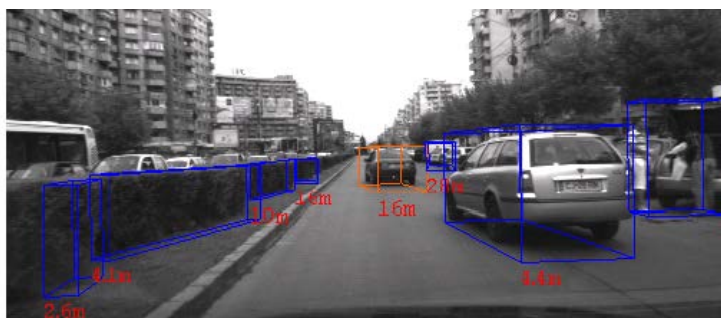


Figura 3.15. Regiuni de interes 3D cu obstacole din scenă

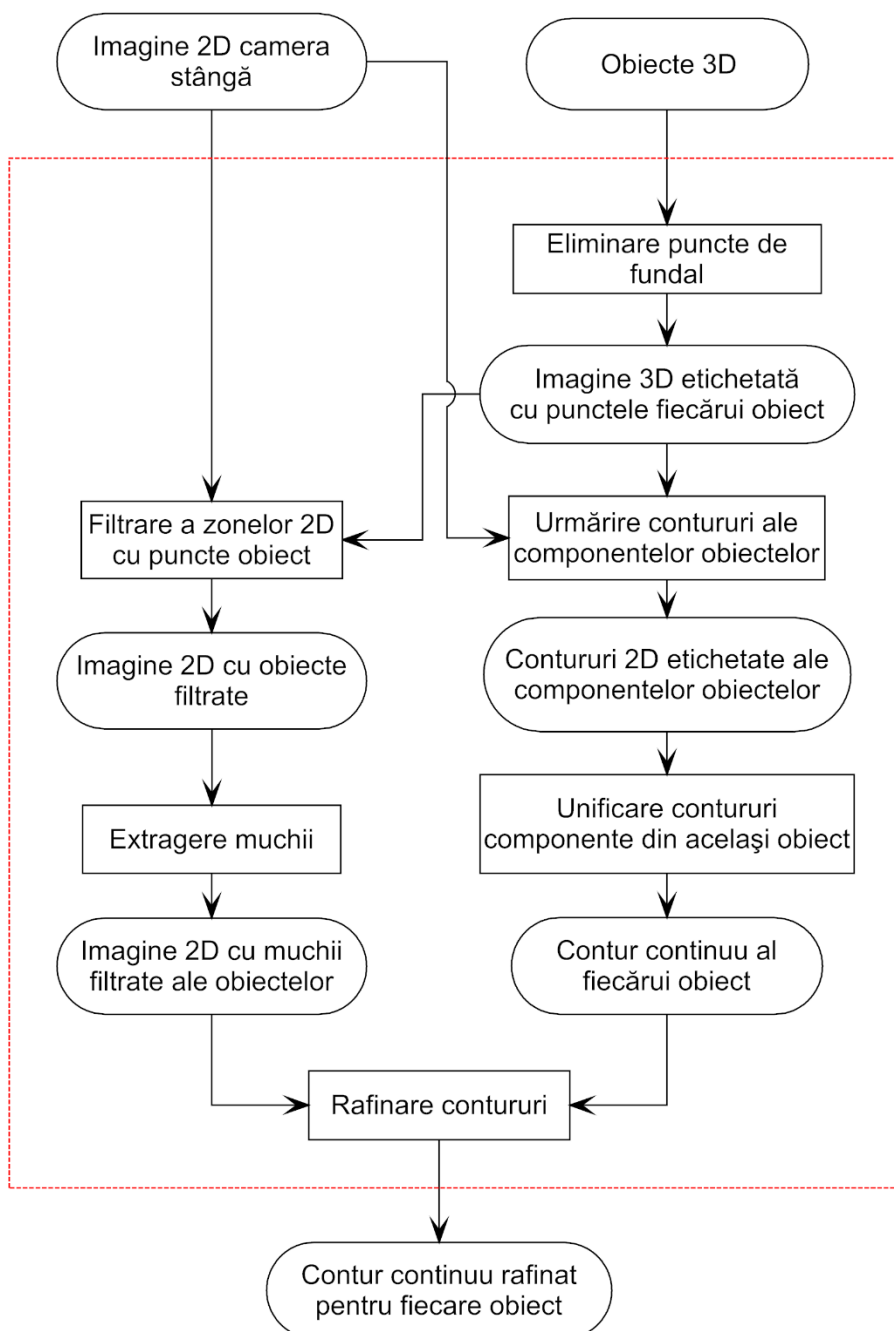


Figura 3.16. Modulul de extragere a conturilor obiectelor (marcat cu linie punctată) cu intrările și ieșirile lui

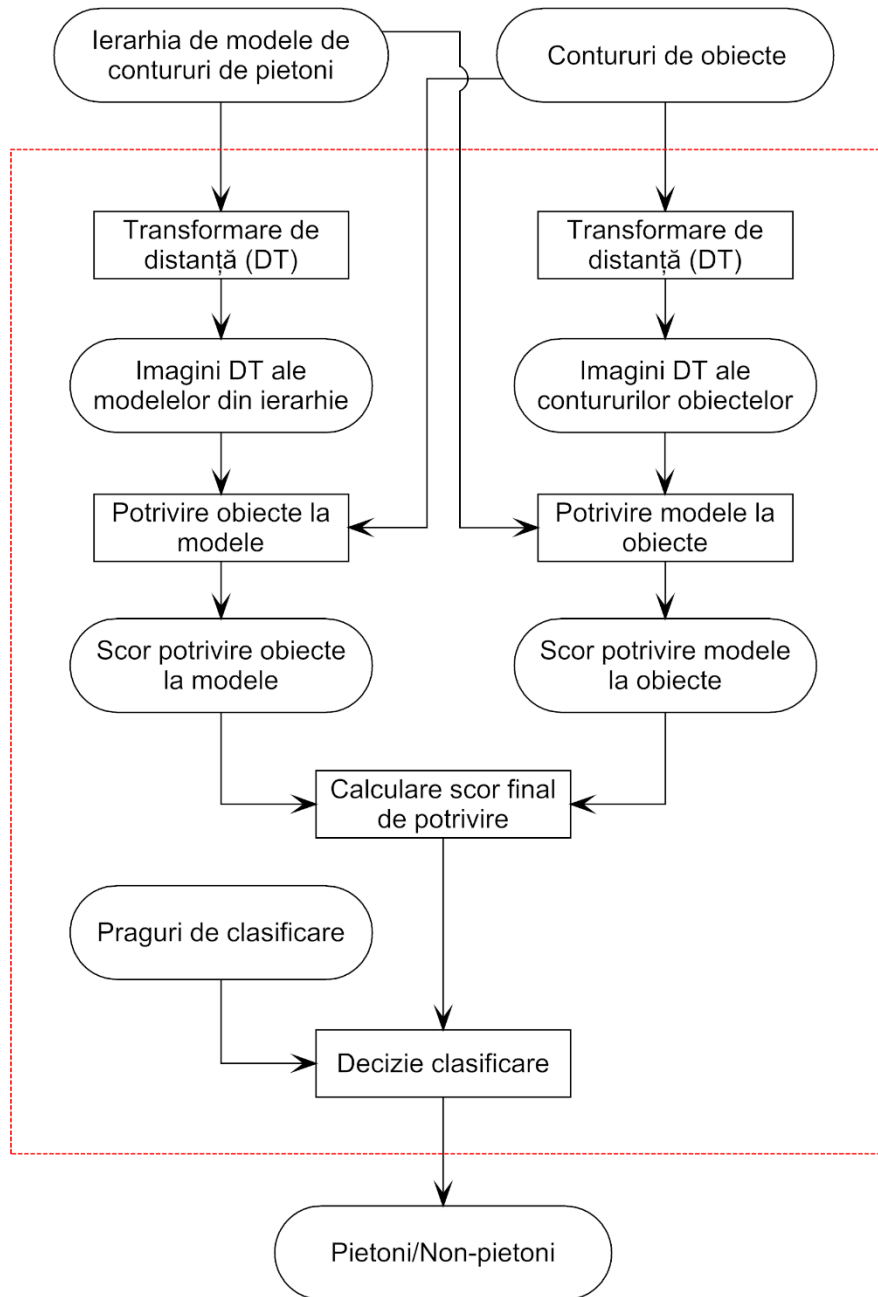


Figura 3.17. Modulul de clasificare a obiectelor pe baza conturilor extrase (marcat cu linie punctată) cu intrările și ieșirile lui



Figura 3.18. Etichetarea punctelor aparținătoare obiectelor din scenă: imaginea scenei (stânga); obiectele etichetate (dreapta)

Recunoașterea pietonilor pe baza clasificării conturului

Metoda dezvoltată pentru clasificarea obiectelor pe baza conturului constă în realizarea unei potriviri de contururi între modelele din ierarhie și obiectele (ipoteze de pietoni) din scenă. Procesul de potrivire este compus din două faze.

Prima fază constă în determinarea scorului de potrivire contur-la-model. În cadrul acesteia conturul ipotezei de pieton este suprapus peste un model de contur de pieton din ierarhie (se parcurge ierarhia, alegând întotdeauna ramura pe care potrivirea este cea mai bună). Se aplică transformata de distanță (DT) [100] pe imaginea conturului modelului din ierarhie. Scorul de potrivire constă în media valorilor intensităților (distanțelor) pixelilor din imaginea transformatei de distanță a conturului modelului care se află sub punctele de contur ale ipotezei (Figura 3.19c).

A doua fază este inversa primei faze. Aceasta constă în determinarea scorului de potrivire model-la-contur. Pentru calculul acestuia, conturul modelului de pieton din ierarhie este suprapus peste conturul ipotezei de pieton (se parcurge ierarhia, alegând întotdeauna ramura pe care potrivirea este cea mai bună). Se aplică transformata de distanță [100] pe imaginea conturului ipotezei de pieton. Scorul de potrivire constă în media valorilor intensităților (distanțelor) pixelilor din imaginea transformatei de distanță a conturului ipotezei de pieton care se află sub punctele de contur ale modelului (Figura 3.19d).

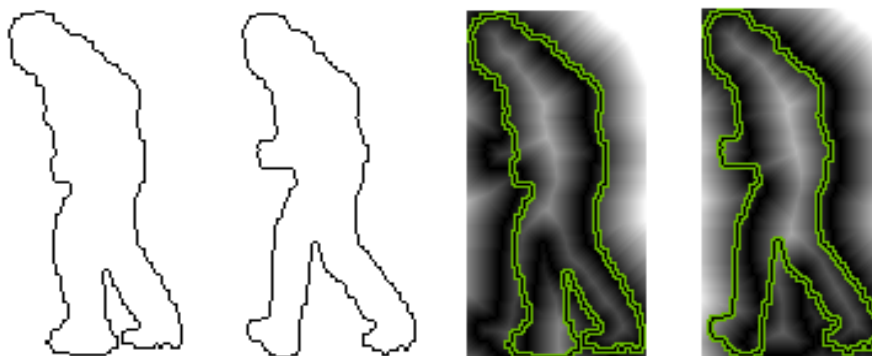


Figura 3.19. Potrivirea conturului modelului la ipoteza de contur și invers utilizând transformata de distanță: a) conturul ipotezei de pieton; b) conturul modelului din ierarhie; c) potrivirea contur-la-model; d) potrivirea model-la-contur

În cadrul procesului de potrivire, conturul ipotezei de pieton și conturul modelului din ierarhie sunt redimensionate la aceleași dimensiuni pentru calculul scorului de potrivire. Un anumit obiect este considerat a fi ipoteză de pieton (Figura 3.20) numai dacă are o anumită dimensiune (înălțimea între 1m-2m și lățimea între 0,25m-1m) și un factor de aspect (înălțime/lățime) între 4 și 1. Scorul de potrivire este calculat ca fiind media aritmetică a scorurilor contur-la-model și model-la-contur calculate anterior. O ipoteză de pieton este clasificată ca fiind pieton dacă acest scor de potrivire este mai mic decât un prag. Acest prag de potrivire se determină printr-un procedeu de învățare, în sensul că se alege un set larg de obiecte cu clasă cunoscută (pietoni sau non-pietoni), se aplică algoritmul de clasificare prin potrivirea conturilor și se calculează scorurile de potrivire

pentru fiecare dintre ele. Se alege valoarea pragului ca fiind acel scor pentru care majoritatea obiectelor sunt clasificate corect.

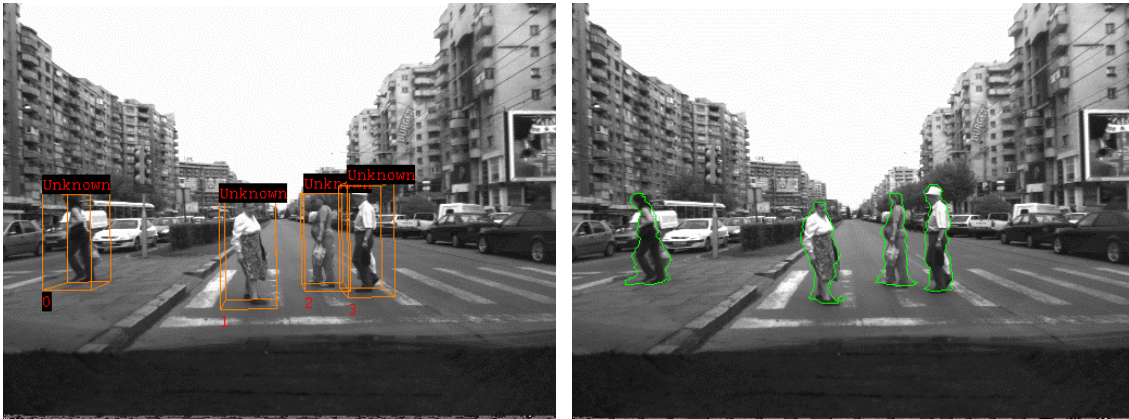


Figura 3.20. Obiecte încă neclasificate reprezentând ipoteze de pietoni (stânga) și conturul extras al acestora (dreapta)

Contururile ipotezelor de pietoni intră în procesul de clasificare, rezultând pentru fiecare decizia apartenenței la clasa pietonilor (Figura 3.21a). De asemenea pentru toate obiectele clasificate ca fiind pietoni sunt afișate modelele din ierarhie cu care s-au potrivit pietonii detectați (Figura 3.21b).

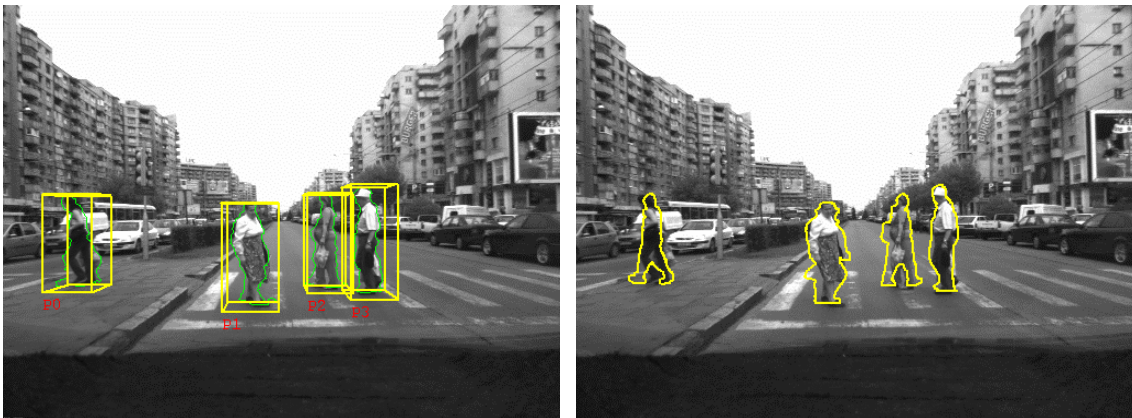


Figura 3.21. Obiecte clasificate ca fiind pietoni, marcate cu galben (stânga); modelul din ierarhie cu care s-a potrivit fiecare pieton detectat (dreapta)

Detecția pietonilor s-a realizat prin extragerea conturilor obiectelor din scenă și potrivirea acestora cu ierarhia de modele de contururi de pietoni. Metodele și algoritmi utilizați în detecția pietonilor au fost testați pe diverse secvențe având mii de cadre în componență.

Curba ROC (rata de detecții false comparată cu rata de detecții corecte pentru un anumit prag al scorului de potrivire) este prezentată în Figura 3.22. Pragul scorului de potrivire ales a fost determinat alegând comportamentul cel mai bun al clasificatorului pe un anumit set de obiecte pre-clasificate. Există evident un compromis între rata de detecții ratate și rata de detecții false.

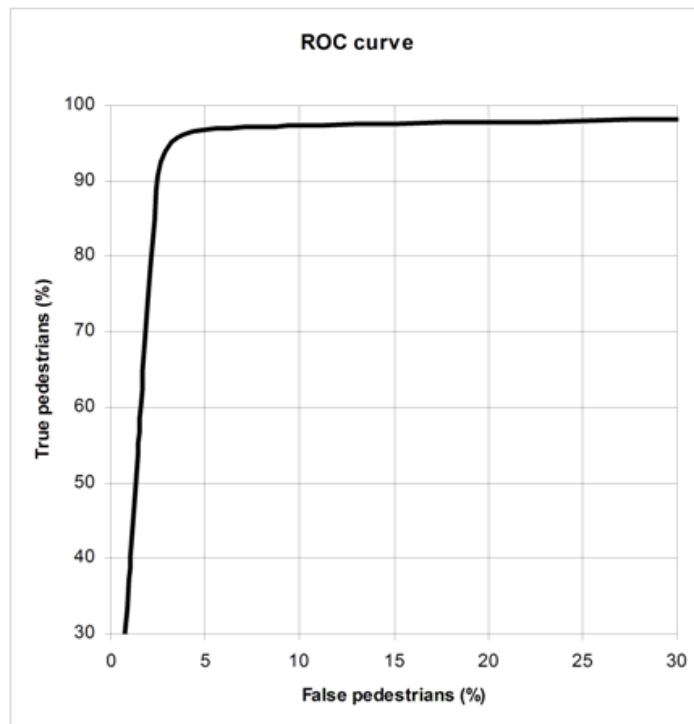


Figura 3.22. Curba de performanță (ROC) a clasificatorului

Rezultatele detecției pietonilor obținute pe o secvență achiziționată dintr-un mediu de complexitate medie-înaltă sunt prezentate în Tabelul 3.1. Câteva cadre din această secvență de test împreună cu rezultatul detecției sunt prezentate în Figura 3.23. Contururile extrase ale tuturor obiectelor sunt trasate cu culoarea verde. Modelele de contururi din ierarhie care se potrivesc peste obiectele detectate ca fiind pietoni sunt trasate cu culoare galbenă.

Tabelul 3.1 Rezultatele detecției pietonilor

Total obiecte	3195	100%
Pietoni	2703	84,6%
Alte obiecte	492	15,4%
Detecție corectă	2963	92,7%
Pietoni nedetecțați	130	4,1%
Pietoni detectați greșit	102	3,2%

Recunoașterea pietonilor utilizând această tehnică reușește să atingă o performanță de timp-real (25 de cadre/secundă pe un sistem de calcul având un procesor Intel Core 2 Duo cu frecvența de 2,66 GHz).



Figura 3.23. Recunoașterea pietonilor prin clasificarea conturului (conturul extras al obstacolelor trasat cu culoare verde și modelul de contur potrivit din ierarhie cu culoare galbenă)

3.2. Recunoașterea claselor de obstacole de interes

3.2.1. Trăsături multiple pentru recunoașterea obstacolelor

În această secțiune este definită o metodă bazată pe trăsături multiple pentru recunoașterea pietonilor, autovehiculelor și a stâlpilor/copacilor în scenarii de trafic urban. Este prezentat întregul sistem de recunoaștere care achiziționează imagini de stereoviziune, reconstruiește punctele 3D utilizând algoritmul SORT-SGM [52], detectează obstacolele pe baza punctelor 3D, urmărește obstacolele, extrage trăsături originale și cu o acuratețe ridicată, iar în final clasifică obstacolele pe baza trăsăturilor, rezultatul clasificării fiind urmărit de-a lungul mai multor cadre succesive. Este analizat modul în care calitatea reconstrucției punctelor 3D afectează acuratețea rezultatelor clasificării. Astfel, pentru reconstrucția stereo, este realizată o comparație între utilizarea metodei semi-globale SORT-SGM și a metodei locale TYZX implementate hardware [53]. O altă comparație este efectuată între utilizarea clasificatorului *Random Forest* și alte tipuri de clasificatoare. Este remarcată acuratețea obținută la utilizarea clasificatorului *Random Forest* cu metoda de reconstrucție SORT-SGM.

Arhitectura sistemului de clasificare a obstacolelor

În Figura 3.24 este prezentată arhitectura sistemului de clasificare a obstacolelor cu toate modulele și fluxul de date dintre acestea.

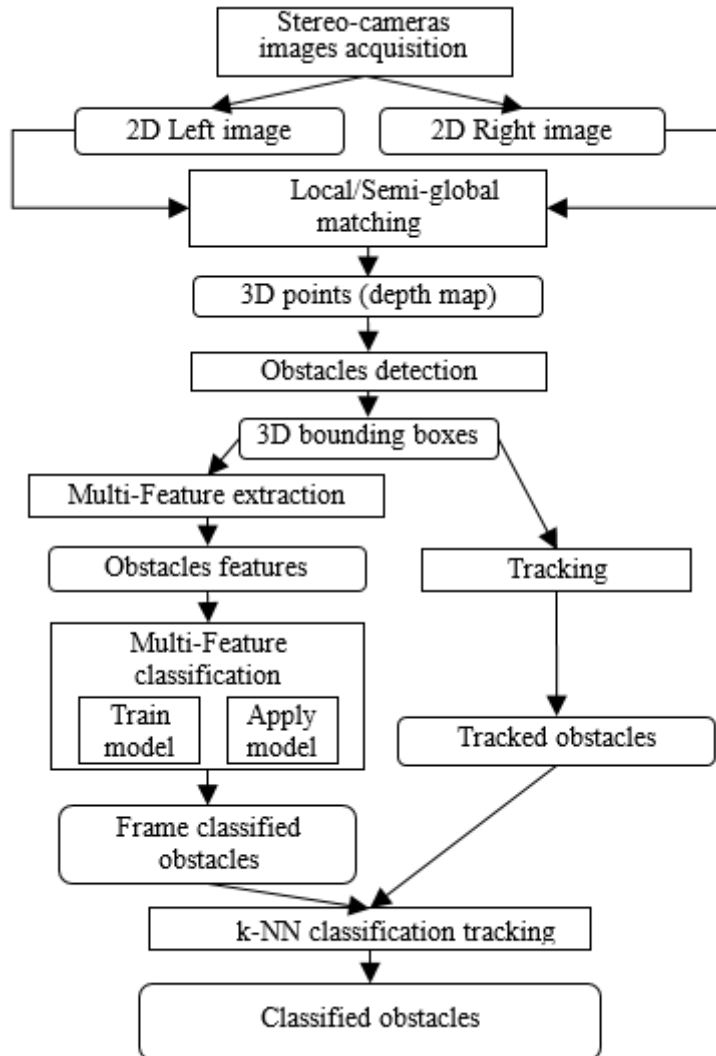


Figura 3.24. Sistemul de clasificare a obstacolelor

După ce imaginile stereo de intensitate de rezoluție 512x383 pixeli au fost achiziționate, acestea fac subiectul procesului de stereo-reconstrucție care s-a realizat cu ajutorul algoritmului SORT-SGM, implementarea fiind realizată pe un procesor grafic, beneficiile acestuia fiind prezentate în secțiunea 2.1 și evidențiate de Figura 2.1. Pentru detecția obstacolelor trebuie realizată în primul rând separația dintre punctele 3D reconstruite aparținătoare lor și cele de pe suprafața drumului sau de fundal. În Figura 3.25 se observă avantajul densității și acurateții punctelor 3D reconstruite cu algoritmul SORT-SGM în comparație cu reconstrucția TYZX. Acesta este un factor determinant în obținerea de acuratețe ridicată în clasificarea obstacolelor.

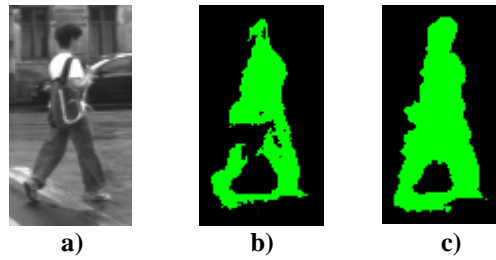


Figura 3.25. a) Imaginea de intensitate a unui pieton; b) punctele 3D reconstruite cu TYZX; c) punctele 3D reconstruite pe GPU cu algoritmul SORT-SGM

Detecția obstacolelor se face pe baza punctelor 3D reconstruite și constă în potrivirea unui paralelipiped dreptunghic care circumscrie fiecare obstacol din scenă. Este utilizată o multi-paradigmă bazată pe vecinătatea punctelor 3D pentru determinarea zonelor ocupate în imagine [101] și pe hărțile de densitate a lor [96]. În Figura 3.26 este prezentat un exemplu de segmentare a scenei în obstacole de interes.

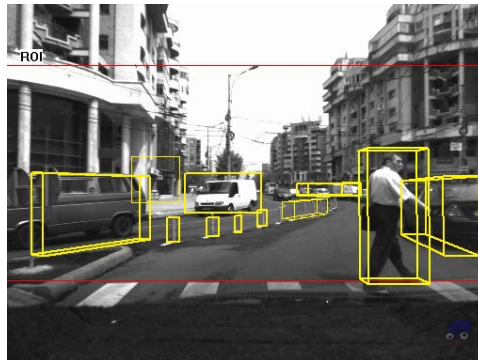


Figura 3.26. Obstacole detectate pe baza punctelor 3D reconstruite

Este utilizat apoi un algoritm de urmărire a obstacolelor pe baza punctelor 3D componente și a vectorilor de mișcare asociați [102]. Acesta definește un model probabilistic pentru obstacole și utilizează un model dinamic care se adaptează la diferitele clase de obstacole. Sunt definite relații ierarhice între obstacole, rezultatele arătând că performanțele modului de detecție a obstacolelor cresc odată cu posibilitatea urmăririi lor.

Extragerea trăsăturilor

Pentru a putea clasifica obstacolele în cele patru clase (pietoni, autovehicule, stâlpi/copaci, alte obstacole) trebuie extrase o serie de trăsături definitorii ale acestora. Vectorul de trăsături FV conține atât trăsături extrase pe baza informației de intensitate cât și pe cea de profunzime. În cele ce urmează sunt descrise succint trăsăturile definite și distribuția lor peste setul celor patru clase de obstacole. Datorită calității ridicate ale punctelor stereo-reconstruite, dimensiunile obstacolelor, contururile, suprafețele și proiecțiile au o acuratețe mult mai ridicată determinând un vector de trăsături mult mai apropiat de valorile reale.

$$FV = (H, W, L, PM, TD, HS, SX, SZ, AR, AC, HR, HC) \quad (3.1)$$

Dimensiunile obstacolelor

Dimensiunile paralelipipedului dreptunghic 3D care circumscrie fiecare obstacol (înălțimea H , lățimea W , lungimea L) sunt primele trăsături extrase și introduse în vectorul de trăsături. Se observă că înălțimea este o trăsătură discriminantă între pietoni sau autovehicule față de celelalte două clase, în timp ce lățimea discriminează autovehiculele de celelalte trei clase. Un obstacol este considerat ipoteză de pieton dacă respectă anumite constrângeri de dimensiuni: înălțimea între 1m și 2m; lățimea între 0,25m și 1m; factorul de aspect (înălțime/lățime) între 1 și 4.

Scorul de potrivire al conturilor pietonilor

Trăsătura PM constă într-un scor de potrivire (secțiunea 3.1.3) și calculată pentru fiecare ipoteză de pieton. Este extras întreg conturul ipotezei de pieton pe baza proiecției punctelor 3D reconstruite în imaginea 2D, rezultând conturul integral exterior pentru acel obstacol. Potrivirea se realizează între acel contur și o ierarhie conținând un număr $N=1872$ de modele de contururi de pietoni (secțiunea 3.1.2).

Procesul de potrivire are două faze. În prima fază se realizează potrivirea contur-model: conturul C al ipotezei este suprapus peste modelul de contur T din ierarhie. Se calculează transformata de distanță DT pe modelul de contur. Scorul de potrivire PM_{CT} este calculat prin însumarea valorilor intensităților pixelilor (distanțelor) din imaginea DT care se află sub pixelii conturului ipotezei. Cea de-a doua fază este asemănătoare primei, și calculează un scor de potrivire inversă PM_{TC} . Scorul de potrivire total este considerat a fi maximul dintre cele două scoruri. Cu cât valoarea acestuia este mai scăzută cu atât probabilitatea ca un obstacol detectat să fie pieton este mai ridicată.

$$\begin{aligned} PM_{CT_i} &= \text{Match}(C, DT(H(T_i))) \\ PM_{T_iC} &= \text{Match}(H(T_i), DT(C)) \\ PM &= \text{TreeSearchMin}_{i=1,N}(\text{Max}(PM_{TC}, PM_{CT_i})) \end{aligned} \quad (3.2)$$

Se observă că având o stereo-reconstrucție de calitate (Figura 3.25b) se obține un contur extras de calitate pe baza punctelor aparținătoare obstacolului, care aproximează mult mai precis forma reală a acestuia și care ulterior oferă calculul unui scor de potrivire mult mai corect.

Scorul de disimilaritate texturală

Scorul de disimilaritate texturală TD este o trăsătură care măsoară disimilaritatea verticală maximă care poate fi găsită pe suprafața obstacolului. Aceasta este calculată considerând zona verticală din mijloc a obstacolului în imaginea de intensitate 2D. Se calculează coordonatele centrului de masă al obstacolului pe baza proiecțiilor punctelor reconstruite în imaginea 2D. Zona de interes considerată (R) pentru analiza texturală este un dreptunghi având lățimea egală cu o treime din lățimea obstacolului și înălțimea

identică cu cea a obstacolului, fiind centrată în poziția coordonatei orizontale a centrului de masă.

Se consideră apoi un set de deplasamente V (1–5 pixeli) pentru calculul matricii de co-apariție a nivelurilor de gri (Figura 3.27). Coeficientul de disimilaritate texturală este calculat pentru fiecare deplasament ca fiind media diferențelor absolute a nivelurilor de gri I din matricea de co-apariție. Scorul final de disimilaritate texturală este considerat ca fiind maximul dintre coeficienții precizați anterior (3.3).

$$TD = \max_{v=1,5} (TD_v)$$

$$TD_v = \text{avg}_{\substack{(i,j) \in R, \\ (i+v,j) \in R}} (|I(i,j) - I(i+v,j)|) \quad (3.3)$$

Scorul de disimilaritate texturală are valori ridicate pentru pietoni datorită îmbrăcămintei lor și valori scăzute pentru stâlpi/copaci datorită aspectului uniform al suprafețelor lor. Această trăsătură este utilizată cu succes pentru discriminarea dintre pietoni și stâlpi/copaci.

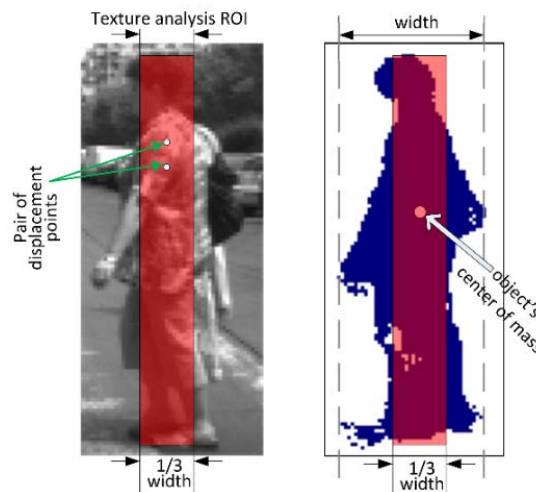


Figura 3.27. Analiza texturală: a) regiunea de interes ROI cu poziția și dimensiunea ei împreună cu o pereche de puncte cu deplasament utilizate pentru calculul co-aparițiilor; b) masca de profunzime

Scorul HoG

Scorul de potrivire a histogramei orientării gradientelor (*HoG*) HS este dat de un set de clasificatori. Aceștia lucrează pe baza histogramei orientării gradientelor [103] calculată din imaginea de intensitate 2D a fiecărui obstacol. Trăsăturile *HoG* sunt obținute prin împărțirea regiunii 2D din imaginea de intensitate corespunzătoare obstacolului în celule de dimensiuni fixe și calculul unei histogramme ponderate a orientării gradientelor în fiecare dintre acestea (Figura 3.28). Celulele sunt grupate apoi în blocuri care se suprapun și valorile histogramelor conținute de fiecare bloc sunt normalizate. Se utilizează un set de antrenare conținând pietoni și alte obstacole pentru a antrena o cascadă de clasificatori. Aceștia oferă un scor de predicție pentru clasificarea pieton-alt obstacol care reprezintă valoarea HS din vectorul de trăsături.

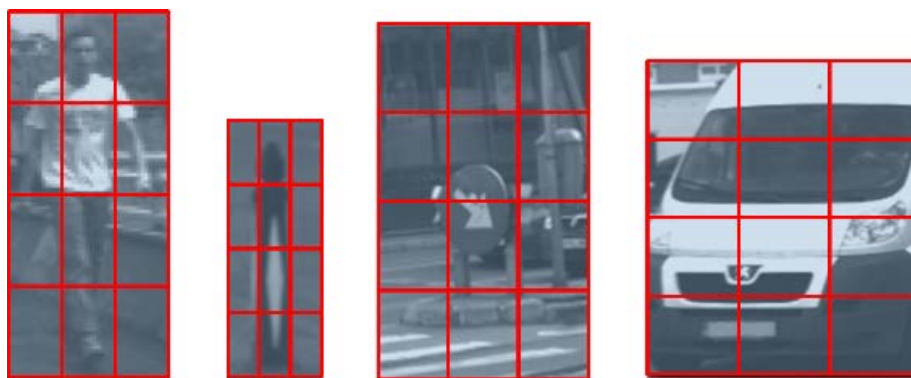


Figura 3.28. Imagini de intensitate cu pietoni și alte obstacole divizate în celule pentru calculul scorului *HoG*

Viteza

Viteza laterală SX și longitudinală SZ sunt extrase pentru toate obstacolele cu ajutorul modulului de urmărire a acestora. Acestea nu sunt foarte discriminante exceptând viteza longitudinală care în unele cazuri poate discrimina pietonii de autovehiculele aflate în mișcare și de alte obstacole.

Măsuri de distanță și arie

S-au definit și calculat două trăsături AR și AC prin înmulțirea ariei de proiecție a obstacolelor P_A (exprimată în pixeli) în imaginea de intensitate cu rădăcina pătratică a distanței D față de camere și respectiv cu cubul distanței (exprimate în milimetri).

$$\begin{aligned} AR &= P_A \sqrt{D} \\ AC &= P_A D^3 \end{aligned} \quad (3.4)$$

Trăsături specifice corpului pietonilor

S-au definit și calculat două trăsături pe baza părții superioare a obstacolelor. În general capul unui om reprezintă cca. 1/7 din înălțimea întregului corp. S-a modelat șeptimea superioară a corpului cu o elipsă și restul corpului cu un dreptunghi (Figura 3.29). Rația cap – restul corpului, HR este definită ca fiind raportul dintre diametrul transversal td a elipsei și lățimea dreptunghiului rw . Cealaltă trăsătură se referă la circularitatea capului HC care separă clar pietonii de celelalte clase de obstacole. Circularitatea este calculată ca fiind factorul de subțiere a zonei obiect din șeptimea superioară a corpului.

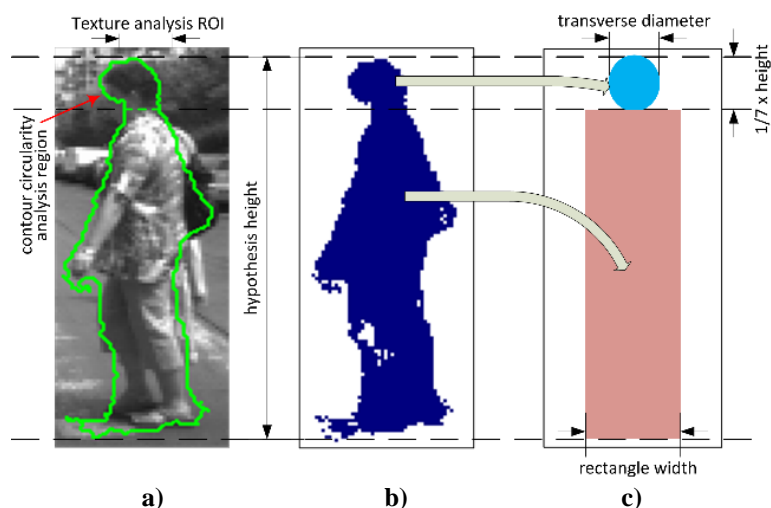


Figura 3.29. Trăsături specifice pietonilor: a) imaginea de intensitate cu conturul extras și regiunea de interes ROI pentru analiza circularității; b) masca de profunzime; c) dimensiunile pentru modelarea capului și a corpului

$$HR = \frac{td}{rw} = \frac{24}{\pi} \cdot \frac{Area_{Head}}{Area_{RestOfBody}} \quad (3.5)$$

$$HC = 4\pi \frac{Area_{Head}}{Perimeter_{Head}^2}$$

Clasificarea obstacolelor pe baza trăsăturilor

Pentru antrenarea clasificatorului s-au extras vectorii de trăsături corespunzătorii unui set larg de obstacole cu clasă cunoscută. Acestea au fost obținute în urma unui proces de etichetare manuală. S-a antrenat un clasificator de tip *Random Forest* pe baza vectorilor de trăsături ai obstacolelor utilizând mediul de antrenare WEKA [104]. S-a utilizat un număr identic de 26309 instanțe pentru fiecare cele patru clase: pietoni, autovehicule, stâlpi/copaci, alte obstacole (Figura 3.30). Dintre clasificatoarele frecvent utilizate în literatura de specialitate ca și *AdaBoost*, *J48 trees*, *multilayer perceptron*, clasificatorul *Random Forest* are avantajele faptului că nu necesită operații de *pruning* a arborilor, nu fac *overfitting* și nu sunt sensibili la măsurători eronate sau în lipsa valorilor unor trăsături. Rezultatul antrenării este un model care este aplicat pe seturile de obstacole detectate cu scopul clasificării lor.

Rezultatele clasificării nu sunt stabile de-a lungul secvențelor de cadre. Vectorul de trăsături al aceluiași obiect se schimbă ușor de la cadru la cadru putând astfel să apară rezultate diferite ale clasificării. De aceea, se utilizează o clasificare de tip *k-NN* [105] pe rezultatele valorilor claselor obținute în ultimele câteva cadre la nivelul aceluiași obiect. Aceasta crește acuratețea și stabilitatea clasificării obstacolelor care pot fi urmărite de-a lungul cadrelor succesive. Se consideră că în general un obstacol este clasificat corect în majoritatea cadrelor în care apare și în care poate fi urmărit, existând posibilitatea ca în câteva cadre acesta să fie clasificat greșit. Obiectivul este de a filtra aceste clasificări eronate și de a înlocui rezultatul greșit cu cel corect. Metoda *k-NN* este potrivită pentru realizarea acestui obiectiv. Se consideră valoarea *k* ca fiind numărul ultimelor cadre

successive în care obstacolul este urmărit. Valoarea W_f reprezintă clasa asignată obstacolului în cadrul f , în urma numărării aparițiilor fiecărei clase în cele k cadre.

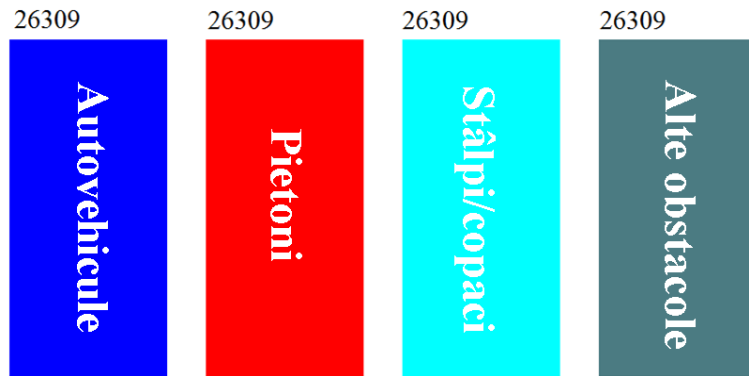


Figura 3.30. Distribuție uniformă a instanțelor claselor utilizate pentru antrenarea clasificatorului

$$C = \{Autovehicul, Pieton, Stâlp / Copac, AltObstacol\}$$

$$V(C_i) = \sum_{f=1}^k W_f(C_i), i = \overline{1,4} \quad (3.6)$$

$$W_f(C_i) = \begin{cases} 1, & \text{dacă obstacolul are clasa } C_i \text{ în cadrul } f \\ 0, & \text{altfel} \end{cases}$$

După însumarea voturilor din ultimele k cadre cu ajutorul formulelor (3.6), clasa obstacolului din cadrul curent este înlocuită cu clasa care are numărul maxim de voturi conform formulei (3.7).

$$W = \arg \max_{i=1,4} (V(C_i)) \quad (3.7)$$

Rezultate experimentale

Sistemul a fost testat pe secvențe de imagini stereo cu mii de cadre din diferite scenarii de trafic urban. S-a construit o bază de date conținând cca. 100000 de obstacole (pietoni, autovehicule, stâlpi/copaci, alte obstacole) împreună cu vectorii de trăsături și clasele asociate (printr-un proces de etichetare manuală). Distribuțiile trăsăturilor pe clase sunt prezentate în Figura 3.31. Există unele trăsături care nu sunt puternic discriminante dar care îmbunătățesc rezultatele finale ale clasificării.

S-a realizat o analiză comparativă a modului în care clasificarea este afectată de calitatea stereo-reconstrucției. Astfel s-au comparat tehnicile de reconstrucție locală cu TYZX și semi-globală SORT-SGM pentru determinarea punctelor din scenă.

Clasificatorul *Random Forest* cu k arbori componenți a fost antrenat cu scopul asignării uneia din cele patru clase la fiecare obstacol detectat din scenă. Adâncimea arborilor nu este limitată și selecția trăsăturilor de clasificare la nivelul fiecărui nod este aleatorie. În selecția aleatorie a fost considerat un număr de $\log_2(TNA)+1$ atribute, în care $TNA=12$ reprezintă numărul total de atribute din vectorul de trăsături. Valoarea k a

fost determinată experimental pe baza analizei curbei ROC (Figura 3.32). Cele mai bune rezultate au fost obținute pentru $k=30$, când s-a generat un model de clasificare rapid și de dimensiuni restrânse.

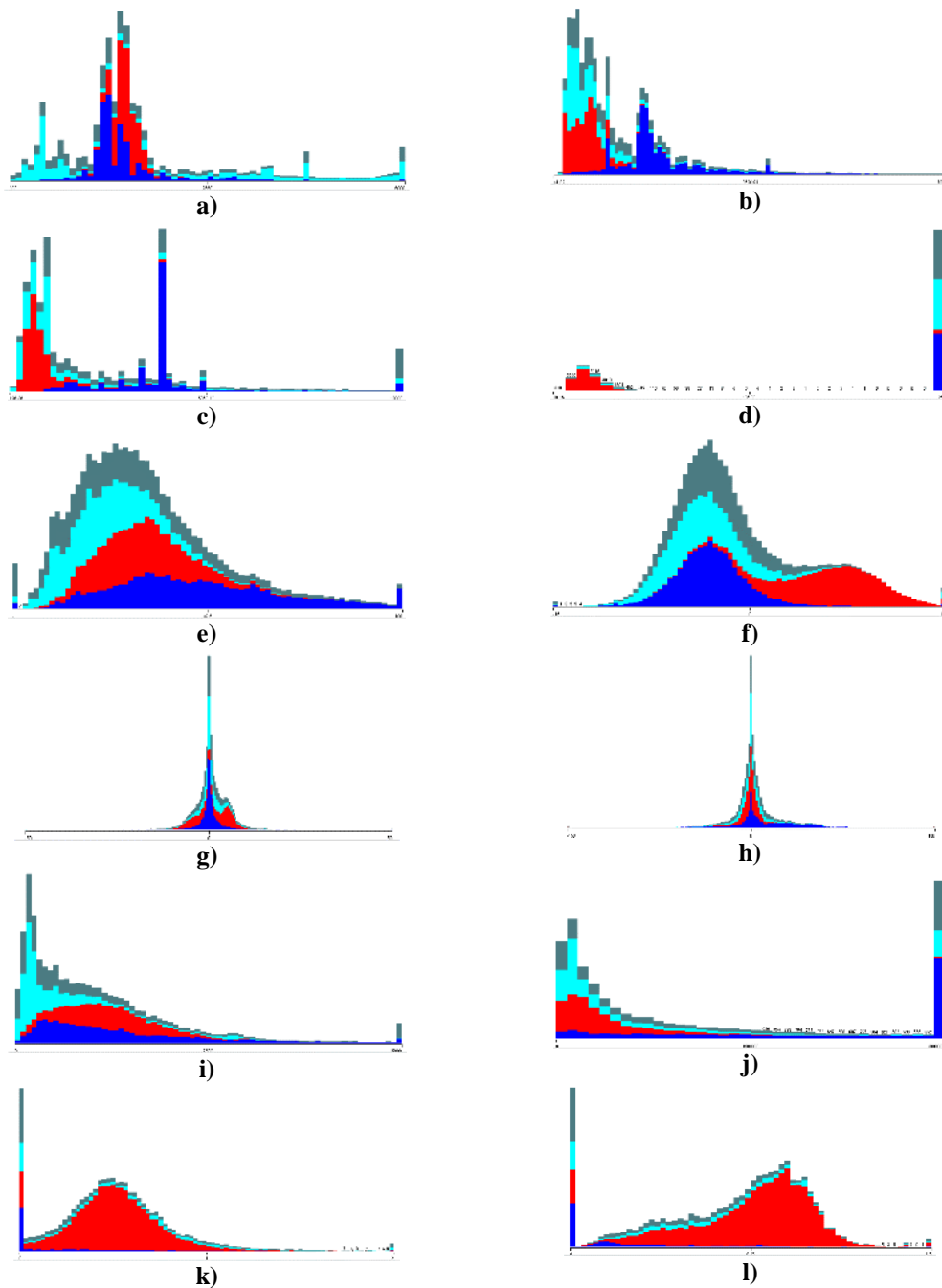


Figura 3.31. Distribuția trăsăturilor pe cele patru clase de obstacole: a) înălțimea; b) lățimea; c) lungimea; d) scorul de potrivire a conturului; e) scorul de disimilaritate texturală; f) scorul *HoG*; g) viteza laterală; h) viteza longitudinală; i) raportul arie-rădăcina pătratică a distanței; j) multiplicarea ariei cu cubul distanței; k) raportul cap-restul corpului; l) circularitatea capului

Avantajul utilizării trăsăturilor definite este evidențiat prin testarea recunoașterii pietonilor cu (Tabelul 3.2) și fără acestea (Tabelul 3.3). Au fost considerați pe rând mai mulți clasificatori pentru a evidenția beneficiile clasificatorului *Random Forest*. O

comparație a rezultatelor recunoașterii pietonilor cu ajutorul clasificatorilor *Random Forest*, *J48 trees*, *AdaBoostM1*, *perceptron multilayer*, *RBF network* este prezentată în Tabelul 3.4.

Tabelul 3.2. Recunoașterea pietonilor utilizând trăsăturile definite

k	Stereo-reconstrucție TYZX		Stereo-reconstrucție SORT-SGM	
	Fals pozitiv	Adevărat pozitiv	Fals pozitiv	Adevărat pozitiv
5	0.049	0.934	0.035	0.940
10	0.040	0.942	0.033	0.945
20	0.034	0.950	0.025	0.951
30	0.030	0.953	0.022	0.955
50	0.030	0.956	0.022	0.956

Tabelul 3.3. Recunoașterea pietonilor fără utilizarea trăsăturilor definite

k	Stereo-reconstrucție TYZX		Stereo-reconstrucție SORT-SGM	
	Fals pozitiv	Adevărat pozitiv	Fals pozitiv	Adevărat pozitiv
5	0.064	0.913	0.051	0.920
10	0.059	0.923	0.050	0.930
20	0.051	0.933	0.042	0.936
30	0.048	0.938	0.040	0.941
50	0.048	0.940	0.040	0.942

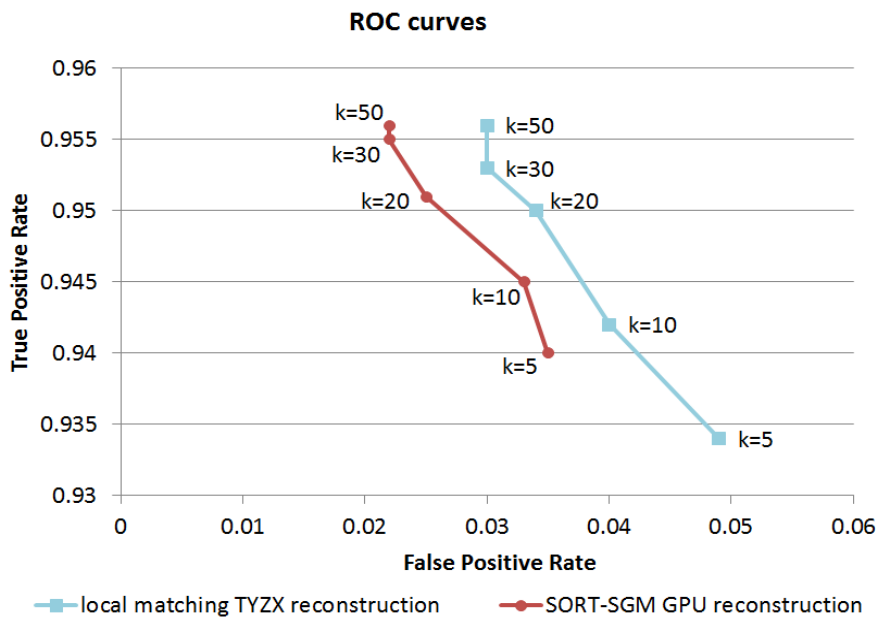


Figura 3.32. Curba ROC a recunoașterii pietonilor: reconstrucție stereo TYZX vs. reconstrucție stereo SORT-SGM

Tabelul 3.4. Rezultatele recunoașterii pietonilor cu diverși clasificatori

Tipul clasificatorului	Recunoașterea pietonilor (cu reconstrucție stereo SORT-SGM)	
	Fals pozitiv	Adevărat pozitiv
J48 trees	0.032	0.927
AdaBoostM1	0.084	0.932
Mutilayer perceptron	0.033	0.933
RBF network	0.036	0.891
Random forest (k=30)	0.022	0.955

Evaluarea clasificării s-a realizat prin metoda *stratified cross-validation* utilizând 10 eșantioane. Utilizând metoda SORT-SGM pentru stereo-reconstrucție împreună cu trăsăturile definite se obține o scădere a detecțiilor false a pietonilor cu cca. 2,6%. Acest lucru este deosebit de important pentru sistemele de asistență a conducătorului auto întrucât nu sunt emise alerte false sau nu este acționată automat frâna în situațiile detecțiilor false. Deși rata de adevăr pozitiv nu este îmbunătățită semnificativ (cca. 1,5%) există situații în care unii pietoni nedetecțai utilizând reconstrucția stereo TYZX sunt detectați prin utilizarea stereo-reconstrucției SORT-SGM. O astfel de situație este prezentată în Figura 3.34. În Figura 3.33 sunt prezentate câteva scenarii de recunoaștere a pietonilor în scenele de trafic urban.

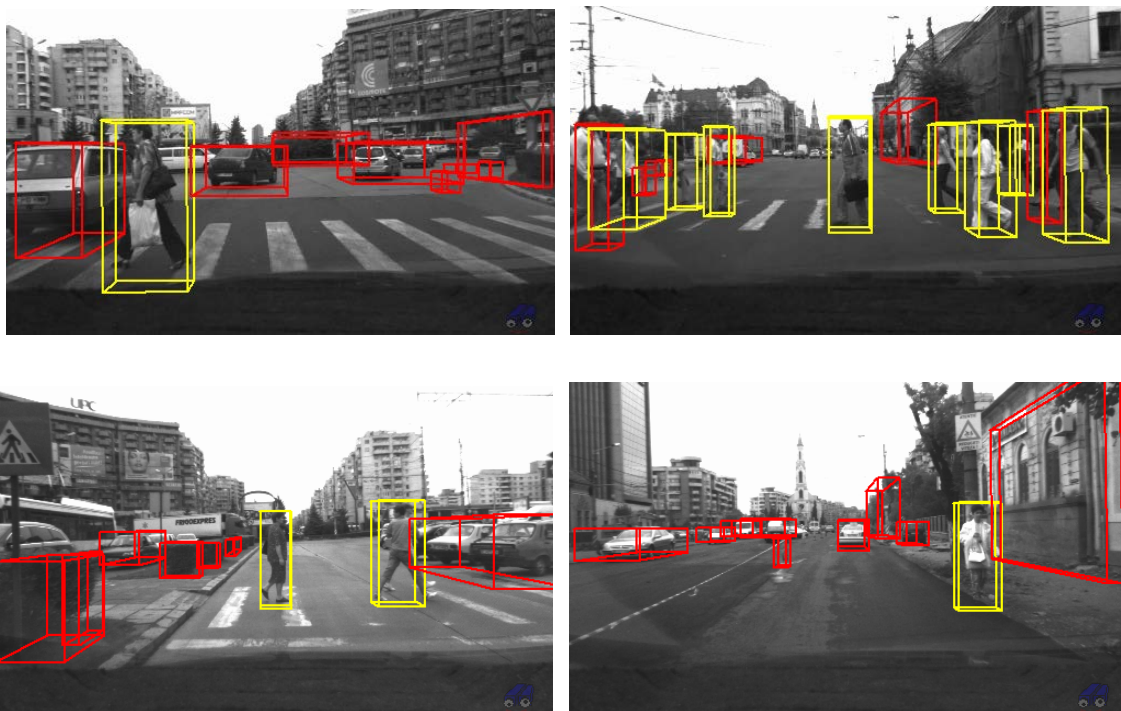


Figura 3.33. Rezultate ale recunoașterii pietonilor (pietoni – cu culoare galbenă; alte obstacole – cu culoare roșie)



Figura 3.34. Clasificări eronate rezolvate prin utilizarea stereo-reconstrucției SORT-SGM (pietoni – cu culoare galbenă; celelalte obstacole – cu culoare roșie: un fals pozitiv de pieton (stânga); un fals negativ de pieton (dreapta))

3.2.2. Inferențe probabilistice pentru urmărirea rezultatelor clasificării

În această secțiune este definit și prezentat un sistem de urmărire a rezultatelor de clasificare a obstacolelor de-a lungul ultimelor cadre procesate și care integrează metoda de clasificare descrisă anterior. Sistemul trebuie să recunoască în timp real și cu o acuratețe ridicată patru clase diferite de obstacole: pietoni, autovehicule, stâlpi/copaci și alte obstacole. Identificarea pietonilor este deosebit de importantă, ei fiind cei mai vulnerabili participanți în trafic, protecția lor fiind vitală în traficul urban. Recunoașterea autovehiculelor este de asemenea importantă deoarece în cazul unui impact ce nu mai poate fi evitat, trebuie ca acesta să nu aibă loc cu zonele în care se află pasagerii. Recunoașterea stâlpilor este utilă pentru evitarea eventualelor coliziuni cu aceștia, pentru inferarea poziției benzii curente de circulație, pentru detecția semnelor de circulație sau culoarea semafoarelor instalate pe aceștia.

Clasificarea obstacolelor trebuie să ofere rezultate foarte bune chiar în condiții dificile de vreme sau de trafic. Algoritmii de clasificare existenți nu sunt suficienți de puternici pentru a putea determina cu acuratețe clasa obstacolelor numai pe baza analizei cadrului curent. Valorile trăsăturilor aceluiși obstacol se pot ușor schimba de la un cadru la altul, ceea ce poate duce la o clasificare incorectă. În general există secvențe de cadre consecutive în care obstacolul este clasificat în majoritate corect, dar care conțin și cadre în care acesta nu este clasificat corect.

Acest neajuns este acoperit de către acest sistem prin definirea unei metode care să urmărească rezultatele clasificării și care să facă posibilă obținerea de rezultate robuste de-a lungul cadrelor succesive. Astfel, rezultatele preliminare ale clasificatorului care recunoaște obstacolele în cadrul curent sunt fuzionate cu cele ale unui modul de urmărire a obstacolelor de-a lungul cadrelor succesive. Este definită astfel o metodă probabilistică care combină aceste rezultate preliminare. Se evidențiază rezultate finale superioare obținute în clasificarea obstacolelor cu acest sistem în comparație cu alte sisteme de clasificare care nu utilizează urmărirea rezultatelor clasificării [106, 107].

Arhitectura sistemului de urmărire a clasificării obstacolelor

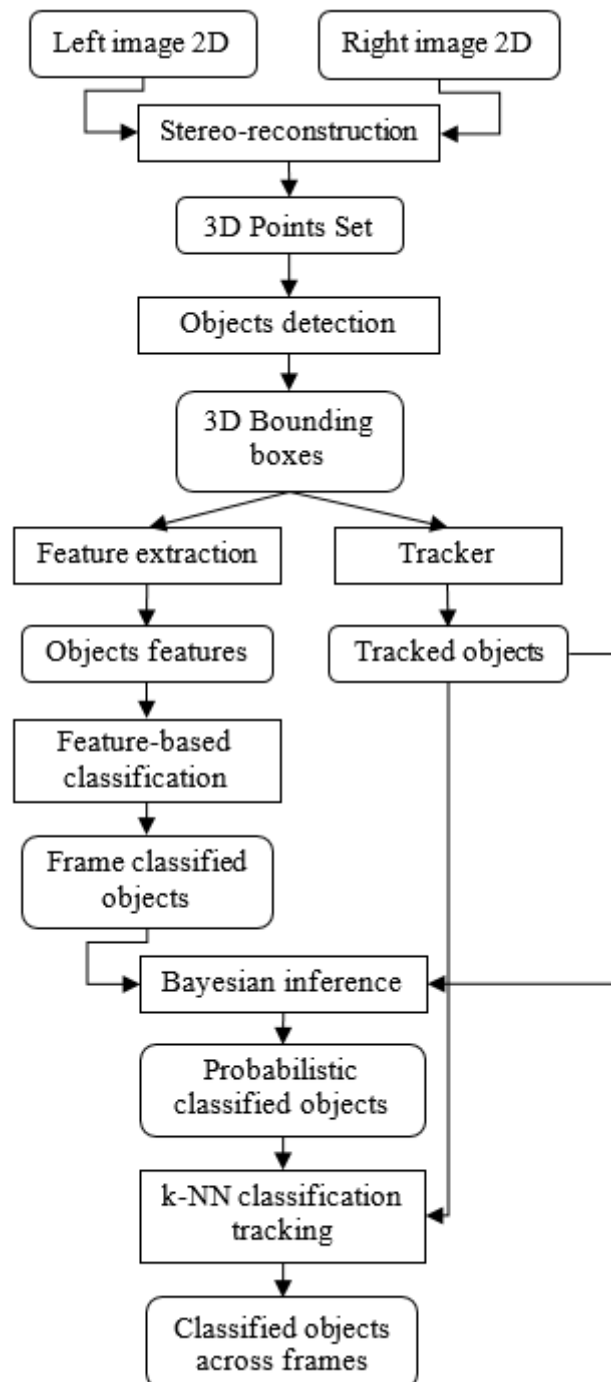


Figura 3.35. Arhitectura sistemului de urmărire a clasificării obstacolelor care integrează metoda definită de inferență probabilistică

În Figura 3.35 este prezentată arhitectura sistemului de urmărire a clasificării obstacolelor care integrează metoda definită pentru inferența probabilistică de-a lungul secvențelor de cadre succesive.

Pentru achiziționarea imaginilor s-a utilizat un sistem de stereoviziune, rezultând două imagini de rezoluție 512x383 pixeli. Stereo-reconstrucția a fost realizată cu ajutorul

unei mașini hardware [53] care realizează stereo-corespondențele pe baza trăsăturilor locale. Detecția obstacolelor a fost realizată pe baza punctelor 3D stereo-reconstruite [101], ulterior acestea urmând a fi clasificate. Algoritmii de urmărire a obstacolelor de-a lungul cadrelor succesive este bazat pe punctele stereo-reconstruite și pe calculul vectorilor de mișcare [108]. Pentru descrierea fiecărui obstacol s-au extras o serie de trăsături specifice care au fost descrise în detaliu în secțiunea precedentă. Acestea constau în general din atribute 3D, atribute de mișcare dar și atribute vizuale 2D. Cele mai relevante trăsături în procesul de clasificare sunt: dimensiunile obstacolelor, factorul de aspect, raportul arie-distanță, viteza laterală și longitudinală, scorul dat de potrivirea histogrammei orientării gradientilor pentru pietoni, scorul de potrivire cu conturul pietonilor, scorul de disimilaritate texturală care este util pentru discriminarea dintre pietoni și stâlpi/copaci. S-a construit o bază de date cu cca. 100000 de obstacole, s-au calculat trăsăturile și s-a asignat clasa fiecărui obstacol printr-un proces de etichetare manuală. Pe baza acestora s-a antrenat cu ajutorul uneia WEKA [104] un clasificator de tip *Random Forest* care a fost descris în secțiunea precedentă și care generează un model de clasificare. Modelul de clasificare este aplicat pe fiecare cadru, asignând câte o clasă pentru fiecare dintre obstacolele din scenă (Figura 3.36).

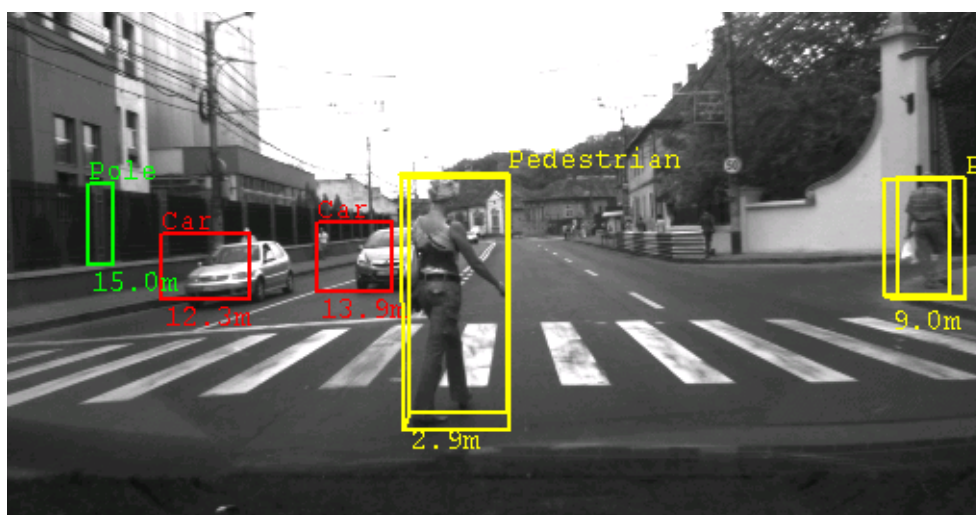


Figura 3.36. Clasificarea obstacolelor într-un singur cadru

Problema care apare în clasificarea obstacolelor în cadre independente este aceea că rezultatul clasificării poate să fie instabil de-a lungul cadrelor succesive pentru obstacolele care pot fi urmărite. Pentru soluționarea acestora s-a definit o abordare în două faze care să rafineze rezultatele clasificării obstacolelor. În primul pas se consideră o inferență Bayesiană care este aplicată pentru calculul probabilității de apartenență la o clasă a obiectelor ce pot fi urmărite de-a lungul cadrelor succesive. Cel de-al doilea pas constă în urmărirea și filtrarea clasei obstacolelor cu ajutorul unui clasificator *k-NN* care analizează rezultatele obținute în ultimele câteva cadre. Aceste metode sunt descrise pe larg în secțiunile următoare. Sunt de asemenea prezentate îmbunătățirile obținute prin utilizarea urmăririi rezultatelor clasificării comparativ cu metoda clasică care clasifică individual obstacolele fiecărui cadru al scenei de trafic.

Inferența Bayesiană

Actualizarea probabilității Bayes este o tehnică statistică importantă care poate fi utilizată în calculul probabilităților apartenențelor la clase pe care un clasificator le asignează unui anumit obstacol în procesul de clasificare. Aceasta este importantă în analiza rezultatelor clasificărilor în serie temporală (într-o secvență de date).

Clasificatorul RF utilizat generează distribuții uniforme și independente pentru apartenența fiecărui obstacol candidat O la clasa C_i . În cazul de față se consideră patru clase: pietoni, autovehicule, stâlpi/copaci, alte obstacole (3.8).

$$\begin{aligned}n &= 4, i = \overline{1, n} \\C_1 &= \text{pieton} \\C_2 &= \text{autovehicul} \\C_3 &= \text{stâlp / copac} \\C_4 &= \text{alt obstacol}\end{aligned}\tag{3.8}$$

Distribuția de probabilitate pentru fiecare dintre ele este necunoscută. Probabilitățile condiționate $P(O|C_i)$ sunt specifice pentru a defini modelele fiecărei clase C_i . Valoarea probabilității $P(C_i)$ reprezintă probabilitatea de apariție a clasei C_i fără a avea alte informații. Inițial se consideră setul de probabilități a priori a claselor astfel:

$$P(C_i) = \frac{1}{n} = 0.25, i = \overline{1, n}\tag{3.9}$$

La nivelul fiecărui cadru, pentru fiecare obstacol detectat O , clasificatorul asignează probabilitatea $P(O|C_i)$ ca obstacolul să aparțină uneia din cele patru clase $C_i, i = \overline{1, n}$ considerate. Această probabilitate reprezintă verosimilitatea $L(O|C_i)$ utilizată în inferența Bayesiană:

$$P(C_i|O) = \frac{L(O|C_i)}{P(O)} \cdot P(C_i), i = \overline{1, n}\tag{3.10}$$

S-au definit două metode pentru calculul verosimilității $L(O|C_i)$ în cadrul curent, astfel:

- Prima metodă consideră verosimilitatea obstacolului ca fiind egală cu probabilitatea asignată de clasificator în cadrul curent:

$$L(O|C_i) = P(O|C_i), i = \overline{1, n}\tag{3.11}$$

- A doua metodă constă în calculul verosimilității ca fiind media aritmetică a tuturor verosimilităților care au fost asignate de clasificator obstacolului de-a lungul apariției lui în secvența de cadre:

$$L(O|C_i) = \frac{1}{N} \sum_{f=1}^N P_f(O|C_i), \quad i = \overline{1, n} \quad (3.12)$$

În formula (3.12) N reprezintă numărul de cadre în care apărut același obstacol O , iar $P_f(O|C_i)$ este probabilitatea asignată lui de către clasificator în cadrul numărul f .

A doua metodă este potrivită numai pentru acele obstacole care pot fi urmărite de-a lungul multiplelor cadre succesive. În cazul în care un obstacol nu poate fi urmărit atunci metoda oferă valoarea verosimilității identică cu cea dată de prima metodă:

$$N = 1, \quad L(O|C_i) = P_1(O|C_i) = P(O|C_i), \quad i = \overline{1, n} \quad (3.13)$$

Probabilitatea a posteriori $P(C_i|O)$ poate fi calculată cu ajutorul formulei (3.10), având calculată valoarea verosimilității calculată fie cu prima metodă (3.11), fie cu cea de-a doua (3.12). Inferența probabilistică Bayesiană clasifică obstacolul în clasa cu indexul W care are probabilitatea maximă din setul tuturor probabilităților a-posteriori (3.12):

$$W = \arg \max_{i=\overline{1, n}} (P(C_i|O)) \quad (3.14)$$

Pasul final constă în actualizarea probabilităților a-priori cu valoarea probabilităților a-posteriori (3.15), repetând inferența în cadrul următor în care obstacolul este urmărit și continuând în acest mod până când acesta dispare din scenă.

$$P(C_i) = P(C_i|O), \quad i = \overline{1, n} \quad (3.15)$$

Discuția asupra cărei metode este mai potrivită pentru a calcula verosimilitatea asociată obstacolelor din fiecare cadru este prezentată în secțiunea dedicată rezultatelor experimentale.

Urmărirea rezultatelor clasificării

Algoritmul k -NN este o metodă pentru clasificarea obiectelor pe baza celor mai apropiate instanțe din spațiul trăsăturilor. Un obiect este clasificat astfel prin vot majoritar dat de vecinii lui, fiindu-i asignată clasa corespunzătoare instanțelor majoritare de același tip din cei k vecini.

Metoda k -NN a fost folosită pentru filtrarea clasificărilor false a fiecărui obstacol care a putut fi urmărit în cadre succesive. S-a presupus că obstacolul este clasificat corect în majoritatea cadrelor în care este urmărit, existând câteva cadre în care inferența Bayesiană dă un rezultat greșit. Obiectivul este de a elimina aceste clasificări greșite și de a le asigna clasa corectă prin utilizarea metodei k -NN.

Se poate determina numărul de apariții a fiecărei clase în cele k cadre considerând valoarea lui k ca fiind numărul ultimelor cadre în care obstacolul apare în scenă și cunoscând clasa W_f care a fost asignată obstacolului în cadrul f (Figura 3.37).

...	W_1	W_2	W_3	...	W_k
cadre anterioare	ultimele k -cadre				

Figura 3.37. Asignarea claselor aceluiași obstacol în ultimele k cadre

$$V(C_i) = \sum_{f=1}^k W_f(C_i), i = \overline{1, n}$$

$$W_f(C_i) = \begin{cases} 1, & \text{dacă obstacolul are clasa } C_i \text{ în cadrul } f \\ 0, & \text{altfel} \end{cases} \quad (3.16)$$

Clasificarea obstacolelor a fost realizată prin inferența Bayesiană, valorile claselor din formula (3.16) fiind calculate cu (3.14). După calcularea tuturor voturilor din ultimele k cadre cu ajutorul formulei (3.16), clasa W care este re-asignată obstacolului în cadrul curent este dată de cea care are numărul maxim de voturi conform formulei (3.17).

$$W = \arg \max_{i=\overline{1, n}} (V(C_i)) \quad (3.17)$$

Se remarcă faptul că rezultatul filtrării depinde de numărul k a ultimelor cadre care este considerat în procesul de votare. Dacă valoarea k este aleasă prea mică, urmărirea rezultatelor clasificării are tendința de a avea un efect de filtrare slabă cu rezultate similare cu cele obținute prin utilizarea simplă a inferenței Bayesiene. În cazul contrar, când k este ridicat, filtrarea poate să fie foarte puternică și să altereze rezultatele clasificării la valori total eronate.

Rezultate experimentale

În această secțiune sunt prezentate comparativ rezultatele clasificării obținute cu și fără inferența probabilistică și urmărirea rezultatelor clasificărilor per fiecare cadru. Este de asemenea realizată o comparație a rezultatelor obținute prin calculul verosimilității ca fiind valoarea din cadrul curent (3.11) și ca fiind media valorilor din mai multe cadre (3.12).

Clasificatorul RF , aplicat independent pe fiecare cadru, este antrenat pe baza unei baze de date cu aproximativ 100000 de obstacole (clasă cunoscută și vectori de trăsături calculați). S-a utilizat un procent de cca. 80% din acest set de date pentru antrenare și restul de 20% pentru testarea clasificatorului. Rezultatele obținute prin clasificarea simplă sunt prezentate în Tabelul 3.5.

Tabelul 3.5. Rezultatele clasificării individuale la nivelul fiecărui cadru

Clasa	Pozitiv adevărat	Fals pozitiv	Aria curbei ROC
Pietoni	0.908	0.032	0.9025
Autovehicule	0.877	0.101	0.8408
Stâlpi/copaci	0.853	0.094	0.8255
Alte obstacole	0.692	0.085	0.6804

Urmărirea rezultatelor clasificării cu calcularea individuală a verosimilității

Rezultatele clasificării obstacolelor considerând ultimele k cadre în clasificatorul k -NN și verosimilitatea calculată individual la nivelul fiecărui cadru pentru inferența Bayesiană sunt prezentate, pentru fiecare clasă, în Tabelul 3.6, Tabelul 3.7, Tabelul 3.8, Tabelul 3.9.

Tabelul 3.6. Clasificarea k -NN pentru clasa *pieton* utilizând verosimilitate calculată individual

k	3	4	5	7	8	10	12
FP Rate	0.0874	0.0404	0.0492	0.0417	0.029	0.059	0.048
TP Rate	0.9254	0.921	0.9314	0.9314	0.942	0.9186	0.9248
ROC value	0.8850	0.9112	0.9155	0.9197	0.9351	0.8994	0.9107

Tabelul 3.7. Clasificarea k -NN pentru clasa *autovehicul* utilizând verosimilitate calculată individual

k	3	4	5	7	8	10	12
FP Rate	0.1236	0.1316	0.1252	0.1169	0.106	0.1189	0.1115
TP Rate	0.8982	0.897	0.9037	0.9036	0.912	0.8972	0.884
ROC value	0.8398	0.8328	0.8420	0.8484	0.8622	0.8428	0.8391

Tabelul 3.8. Clasificarea k -NN pentru clasa *stâlp/copac* utilizând verosimilitate calculată individual

k	3	4	5	7	8	10	12
FP Rate	0.0992	0.1058	0.0904	0.074	0.0873	0.1088	0.0852
TP Rate	0.8618	0.872	0.8866	0.89	0.8826	0.8902	0.8606
ROC value	0.8298	0.8339	0.8549	0.8674	0.8536	0.8454	0.8366

Tabelul 3.9. Clasificarea k -NN pentru clasa *alt obstacol* utilizând verosimilitate calculată individual

k	3	4	5	7	8	10	12
FP Rate	0.081	0.098	0.1144	0.0843	0.075	0.1002	0.0951
TP Rate	0.6734	0.6811	0.6874	0.6936	0.7	0.6838	0.6708
ROC value	0.6635	0.6663	0.6671	0.6822	0.6907	0.6683	0.6573

În ceea ce privește valorile curbelor ROC, există variații pentru toate clasele de obstacole când parametrul k este variat între 3 și 12 ultime cadre. Din analiza curbelor ROC, se pot observa aceste variații și se poate remarca faptul că există un maxim local în aproape toate cele patru curbe în vecinătatea valorii $k=8$ (Figura 3.38). Astfel, s-a considerat valoarea $k=8$ ca fiind optimă pentru urmărirea rezultatelor clasificării considerând calcularea individuală a verosimilităților din inferențele Bayesiene.

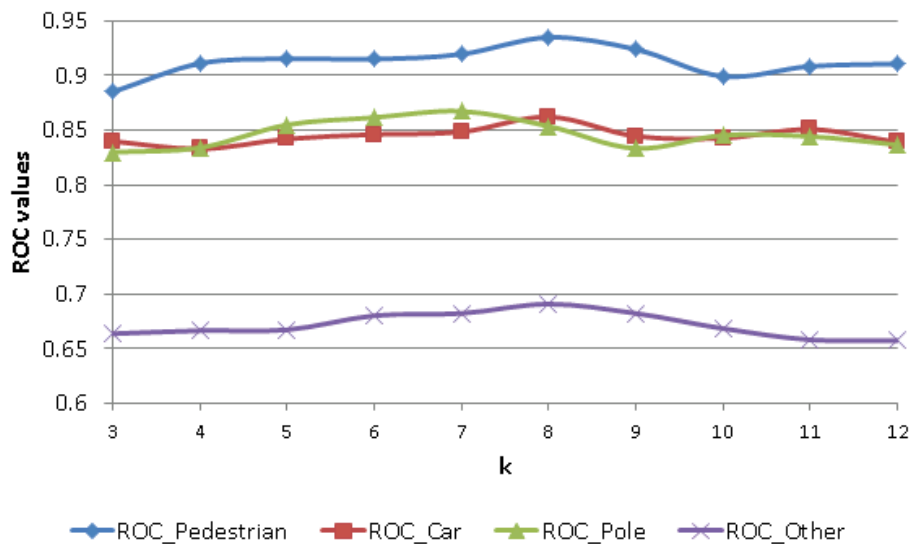


Figura 3.38. Curbele ROC pentru cele patru clase de obstacole – verosimilitate calculată individual

Urmărirea rezultatelor clasificării cu calcularea mediei verosimilităților

Rezultatele clasificării obstacolelor considerând ultimele k cadre în clasificatorul k -NN și verosimilitatea calculată prin media aritmetică pentru inferența Bayesiană sunt prezentate pentru fiecare clasă în Tabelul 3.10, Tabelul 3.11, Tabelul 3.12, Tabelul 3.13.

Tabelul 3.10. Clasificarea k -NN pentru clasa *pieton* utilizând verosimilitate calculată prin media aritmetică

k	3	4	5	7	8	10	12
FP Rate	0.0343	0.034	0.029	0.0398	0.0407	0.0414	0.0437
TP Rate	0.9321	0.9355	0.94	0.9298	0.923	0.918	0.918
ROC value	0.9239	0.9270	0.9333	0.9193	0.9129	0.9081	0.9070

Tabelul 3.11. Clasificarea k -NN pentru clasa *autovehicul* utilizând verosimilitate calculată prin media aritmetică

k	3	4	5	7	8	10	12
FP Rate	0.1121	0.1156	0.108	0.1186	0.1144	0.124	0.135
TP Rate	0.8972	0.8977	0.9	0.8995	0.892	0.878	0.866
ROC value	0.8479	0.8456	0.8528	0.8445	0.8426	0.8260	0.8097

Tabelul 3.12. Clasificarea k -NN pentru clasa *stâlp/copac* utilizând verosimilitate calculată prin media aritmetică

k	3	4	5	7	8	10	12
FP Rate	0.0795	0.076	0.0778	0.0805	0.0858	0.0802	0.1201
TP Rate	0.8882	0.89	0.8865	0.885	0.8756	0.869	0.872
ROC value	0.8628	0.8662	0.8623	0.8596	0.8488	0.8463	0.8244

Tabelul 3.13. Clasificarea k -NN pentru clasa *alt obstacol* utilizând verosimilitate calculată prin media aritmetică

k	3	4	5	7	8	10	12
FP Rate	0.0906	0.0867	0.08	0.0849	0.0942	0.0874	0.1195
TP Rate	0.686	0.6891	0.695	0.6876	0.69	0.6806	0.6712
ROC value	0.6731	0.6772	0.6846	0.6762	0.6760	0.6688	0.6501

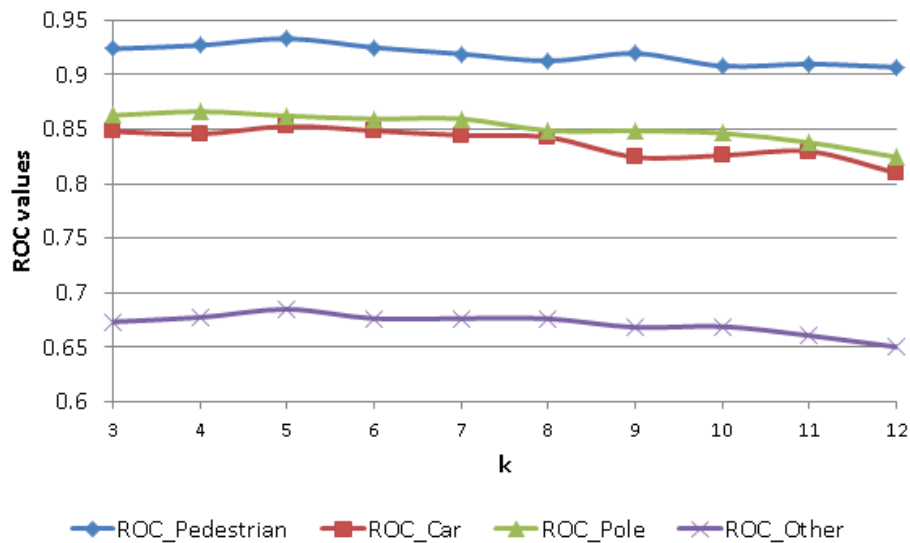
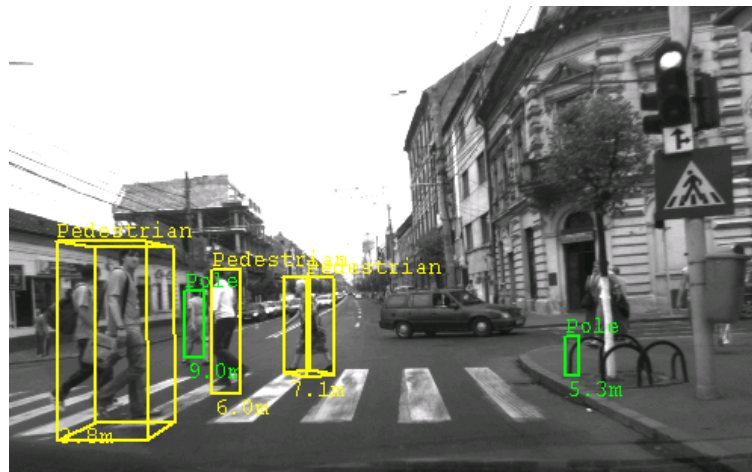


Figura 3.39. Curbele ROC pentru cele patru clase de obstacole – verosimilitate calculată prin medie aritmetică

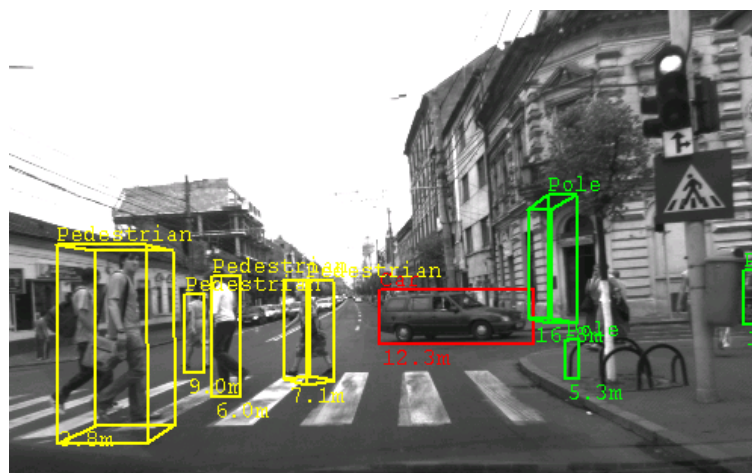
S-a aplicat aceeași procedură ca și în cazul descris anterior, variind valoarea lui k de la 3 la 12. Din graficele curbelor ROC se poate observa faptul că există variații foarte mici la diferite valori ale lui k (Figura 3.39). Se identifică un maxim local în jurul valorii $k=5$ pentru aproape toate curbele ROC. Valoarea $k=5$ este considerată optimă pentru urmărirea rezultatelor clasificării considerând calcularea prin medie aritmetică a verosimilităților din inferențele Bayesiene.

Statistic, acuratețea clasificării pentru aproape toate clasele de obstacole a fost îmbunătățită cu aproximativ 5% în ratele de adevărat pozitiv și fals pozitiv. Excepție face clasa altor obstacole care nu a beneficiat de nici o îmbunătățire. Acest lucru se explică prin faptul că această clasă conține și părți de obstacole aparținătoare celorlalte trei clase și care sunt dificil de urmărit de-a lungul secvențelor de cadre.

Un exemplu de rezultat obținut înainte și după aplicarea tehnicii descrise de urmărire probabilistică a rezultatelor clasificării este prezentat în Figura 3.40. Se remarcă faptul că utilizând tehnica descrisă sunt determinate mult mai precis clasele reale ale obstacolelor din scenă.



a) fără utilizarea urmării clasificării (câteva obstacole sunt clasificate greșit)



b) cu utilizarea urmării clasificării (obstacolele sunt clasificate corect)

Figura 3.40. Comparație între clasificarea obstacolelor cu și fără utilizarea tehnicii descrise pentru urmărirea rezultatelor

3.3. Concluzii

În acest capitol au fost prezentate soluții pentru recunoașterea pietonilor (secțiunea 3.1) și recunoașterea în general a obstacolelor din scenele de trafic prin clasificarea lor în patru clase diferite: pietoni, autovehicule, stâlpi/copaci, alte obstacole (secțiunea 3.2).

În secțiunea 3.1.1 este definit un algoritm pentru extragerea conturului pietonilor pe baza informațiilor de intensitate și profunzime. Este calculată histograma de profunzime și este extras conturul integral pe punctele 3D filtrate pe baza maximului din histogramă, proiectate în spațiul imagine 2D. Conturul este apoi rafinat pe baza muchiilor extrase din imaginea de intensitate filtrată în prealabil cu algoritmul *NL-Means*. Algoritmul de extragere a conturului poate fi utilizat atât pentru obținerea conturilor obstacolelor din scenă cât și pentru generarea de contururi ideale de pietoni.

În secțiunea 3.1.2 contururile extrase sunt folosite pentru construirea unei ierarhii de modele de contururi de pietoni cu structură de arbore multi-căi. Algoritmul de construire a ierarhiei nu consideră un număr fix de modele pentru toate grupările situate în nodurile din arborele ierarhiei. Este definită o euristică pentru generarea grupurilor inițiale. Aceasta oferă flexibilitate și posibilitatea grupării conturilor modelelor de pietoni după atitudinea și orientarea pe care o au. O astfel de ierarhie poate fi construită automat fără a fi necesară setarea unor parametri inițiali și este deosebit de utilă pentru creșterea calității detecției pietonilor prin metoda de potrivire a conturilor. Rezultatele au fost publicate în [109].

În secțiunea 3.1.3 este definită o metodă de recunoaștere a pietonilor pe baza conturului integral al ipotezelor de pietoni și un algoritm de potrivire a conturilor cu ierarhia de modele de contururi. Este descrisă o metrică de potrivire a conturilor ipotezelor de pietoni cu modelele din ierarhie prin parcurgerea arborelui multi-căi. Această parcurgere este orientată de valoarea scorului de potrivire. Rezultatele au fost publicate în [110].

În secțiunea 3.2.1 este definit un set de trăsături originale (disimilaritate texturală, trăsături specifice corpului uman, trăsături legate de distanța obstacolelor, viteză) utilizate pentru caracterizarea obstacolelor în scenele de trafic. Acestea, împreună cu trăsături clasice (scorul de potrivire *HOG*, potrivirea conturilor și dimensiunea obstacolelor) existente în literatura de specialitate, sunt utilizate pentru caracterizarea obstacolelor din scenele de trafic. Sunt antrenați mai mulți clasificatori (*J48 trees*, *AdaBoostM1*, *Random Forest*, *Mutilayer perceptron*, *RBF network*), rezultatele cele mai bune fiind obținute cu clasificatorul de tip *Random Forest*. De asemenea, este evidențiat avantajul stereo-reconstrucției de calitate obținute prin metoda SORT-SGM care afectează rezultatele clasificării în sens pozitiv. Este definită în final o metodă de tip *k-NN* care filtrează rezultatele clasificării de-a lungul cadrelor succesive de imagini. Rezultatele comparative, cu evidențierea îmbunătățirii recunoașterii pietonilor, au fost publicate în [111].

În secțiunea 3.2.2 este definită o metodă de urmărire probabilistică a rezultatelor clasificării obstacolelor în patru clase (pietoni, autovehicule, stâlpi/copaci, alte obstacole)

cu scopul creșterii acurateții acestora. Este definită utilizarea unei metode probabilistice de inferență Bayesiană pentru calculul probabilității apartenenței obstacolelor la clase. Pentru calculul verosimilității sunt prezentate două abordări: una care calculează verosimilitatea ca fiind valoarea probabilității curente de clasificare iar cealaltă ca fiind media aritmetică a valorilor probabilităților pe ultimele cadre. În final este utilizată metoda k - NN pentru filtrarea rezultatelor de-a lungul cadrelor succesive. Rezultatele arată o creștere a acurateții clasificării la nivelul majorității claselor, excepție făcând clasa altor obstacole care conțin părți de obstacole și care sunt dificil de caracterizat unitar și de urmărit în timp. Rezultatele au fost publicate în [112].

Bibliografie

- [1] I. Ulrich and I. R. Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 2000, pp. 866-871.
- [2] S. Lenser and M. Veloso, "Visual sonar: fast obstacle avoidance using monocular vision," in *Proceedings of International Conference on Intelligent Robots and Systems*, 2003, pp. 886-891 vol.1.
- [3] T. Taylor, S. Geva, and W. W. Boles, "Monocular Vision as a Range Sensor," in *International Conference on Computational Intelligence for Modelling, Control & Automation (CIMCA)*, Gold Coast, Australia, 2004, pp. 566-575.
- [4] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd International Conference on Machine learning*, Bonn, Germany, 2005, pp. 593-600.
- [5] A. Saxena, S. Chung, and A. Ng, "3-D Depth Reconstruction from a Single Still Image," *International Journal of Computer Vision*, vol. 76, pp. 53-69, 2008/01/01 2008.
- [6] K. Tien-Ying and L. Yi-Chung, "Depth estimation from a monocular view of the outdoors," *IEEE Transactions on Consumer Electronics*, vol. 57, pp. 817-822, 2011.
- [7] H. Kaiming, S. Jian, and T. Xiaoou, "Single Image Haze Removal Using Dark Channel Prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 2341-2353, 2011.
- [8] A. Wedel, U. Franke, J. Klappstein, T. Brox, and D. Cremers, "Realtime depth estimation and obstacle detection from monocular video," in *Proceedings of the 28th conference on Pattern Recognition*, Berlin, Germany, 2006, pp. 475-484.
- [9] C. Démonceaux and D. Kachi-Akkouche, "Robust obstacle detection with monocular vision based on motion analysis," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 527-532.
- [10] N. S. Boroujeni, S. A. Etemad, and A. Whitehead, "Fast obstacle detection using targeted optical flow," in *IEEE International Conference on Image Processing (ICIP)*, 2012, pp. 65-68.
- [11] B. Jia, R. Liu, and M. Zhu, "Real-time obstacle detection with motion features using monocular vision," *The Visual Computer*, pp. 1-13, 2014/01/28 2014.
- [12] S. Wybo, D. Tsishkou, C. Vestri, F. Abad, S. Bougnoux, and R. Bendahan, "Monocular vision obstacles detection for autonomous navigation," in *International Conference on Intelligent Robots and Systems*, 2008, pp. 4190-4195.
- [13] K. Yamaguchi, T. Kato, and Y. Ninomiya, "Moving Obstacle Detection using Monocular Vision," in *IEEE Intelligent Vehicles Symposium*, 2006, pp. 288-293.
- [14] S. Qian, J. K. Tan, H. Kim, S. Ishikawa, T. Morie, and T. Shinomiya, "Road region estimation and obstacles extraction using a monocular camera," *International Journal of Innovative Computing, Information and Control*, vol. 9, pp. 3561-3572, 2013.

- [15] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo vision-based vehicle detection," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2000, pp. 39-44.
- [16] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair, and A. Broggi, "Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis," *IEEE Transactions on Image Processing*, vol. 15, pp. 2364-2375, 2006.
- [17] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, *et al.*, "High accuracy stereo vision system for far distance obstacle detection," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 292-297.
- [18] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, *et al.*, "A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 276-283.
- [19] K. Huh, J. Park, J. Hwang, and D. Hong, "A stereo vision-based obstacle detection system in vehicles," *Optics and Lasers in Engineering*, vol. 46, pp. 168-178, 2// 2008.
- [20] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, pp. 62-81, 1998.
- [21] M. Bertozz, A. Broggi, and A. Fascioli, "Stereo inverse perspective mapping: theory and applications," *Image and Vision Computing*, vol. 16, pp. 585-590, 6// 1998.
- [22] A. Broggi, P. Medici, and P. P. Porta, "StereoBox: a robust and efficient solution for automotive short-range obstacle detection," *EURASIP Journal on Embedded Systems*, vol. 2007, pp. 6-6, 2007.
- [23] L. You and Y. Ruichek, "Moving objects detection and recognition using sparse spatial information in urban environments," in *IEEE Intelligent Vehicles Symposium*, 2012, pp. 1060-1065.
- [24] B. Lefaudeux and F. Nashashibi, "Real-time visual perception : detection and localisation of static and moving objects from a moving stereo rig," in *International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2012, pp. 522-527.
- [25] S. Heinrich, "Fast obstacle detection using flow/depth constraint," in *IEEE Intelligent Vehicle Symposium*, 2002, pp. 658-665 vol.2.
- [26] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6D-vision: fusion of stereo and motion for robust environment perception," presented at the Proceedings of the 27th DAGM conference on Pattern Recognition, Vienna, Austria, 2005.
- [27] A. Barth and U. Franke, "Estimating the Driving State of Oncoming Vehicles From a Moving Platform Using Stereo Vision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 560-571, 2009.
- [28] F. Erbs, A. Barth, and U. Franke, "Moving vehicle detection by optimal segmentation of the Dynamic Stixel World," in *IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 951-956.
- [29] R. Labayrade, D. Aubert, and J. P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," in *IEEE Intelligent Vehicle Symposium*, 2002, pp. 646-651 vol.2.

- [30] Z. Zhen, W. Yifei, J. Brand, and N. Dahnoun, "Real-time obstacle detection based on stereo vision for automotive applications," in *European DSP Education and Research Conference*, 2012, pp. 281-285.
- [31] M. Perrollaz, J. D. Yoder, A. Negre, A. Spalanzani, and C. Laugier, "A Visibility-Based Approach for Occupancy Grid Computation in Disparity Space," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 1383-1393, 2012.
- [32] H. Zhencheng and K. Uchimura, "U-V-disparity: an efficient algorithm for stereovision based scene analysis," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2005, pp. 48-54.
- [33] K. Jungwon and C. Myung-Jin, "Stereo-vision based free space and obstacle detection with structural and traversability analysis using probabilistic volume polar grid map," in *IEEE Conference on Robotics, Automation and Mechatronics*, 2011, pp. 245-251.
- [34] M. Perrollaz, A. Spalanzani, and D. Aubert, "Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 313-318.
- [35] I. Cabani, G. Toulminet, and A. Bensrhair, "Contrast-invariant Obstacle Detection System using Color Stereo Vision," in *International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 1032-1037.
- [36] U. Franke and A. Joos, "Real-time stereo vision for urban traffic scene understanding," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2000, pp. 273-278.
- [37] H. Badino, U. Franke, and R. Mester, "Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming," in *ICCV Workshop on Dynamical Vision*, 2007.
- [38] Q. Yu, H. Araujo, and H. Wang, "Stereo-vision based real time obstacle detection for urban environments," in *11th International Conference on Advanced Robotics*, Coimbra, Portugal, 2003.
- [39] A. Broggi, M. Buzzoni, M. Felisa, and P. Zani, "Stereo obstacle detection in challenging environments: The VIAC experience," in *International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1599-1604.
- [40] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, *et al.*, "High accuracy stereovision approach for obstacle detection on non-planar roads," in *IEEE Intelligent Engineering Systems (INES)*, 2004, pp. 211-216.
- [41] F. Oniga, S. Nedevschi, M. M. Meinecke, and T. Thanh-Binh, "Road Surface and Obstacle Detection Based on Elevation Maps from Dense Stereo," in *IEEE Intelligent Transportation Systems Conference*, 2007, pp. 859-865.
- [42] F. Oniga and S. Nedevschi, "Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 1172-1182, 2010.
- [43] C. Pocol, S. Nedevschi, and M. M. Meinecke, "Obstacle Detection Based on Dense Stereovision for Urban ACC Systems," in *International Workshop on Intelligent Transportation (WIT)*, Hamburg, Germany, 2008, pp. 13-18.
- [44] C. D. Pantilie, S. Bota, I. Haller, and S. Nedevschi, "Real-time obstacle detection using dense stereo vision and dense optical flow," in *International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2010, pp. 191-196.

- [45] C. D. Pantilie and S. Nedevschi, "Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow," in *International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2010, pp. 439-444.
- [46] R. Danescu, C. Pantilie, F. Oniga, and S. Nedevschi, "Particle Grid Tracking System Stereovision Based Obstacle Perception in Driving Environments," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, pp. 6-20, 2012.
- [47] M. Perrollaz, R. Labayrade, R. Gallen, and D. Aubert, "A Three Resolution Framework for Reliable Road Obstacle Detection Using Stereovision," in *Conference on Machine Vision Applications*, 2007, pp. 469-472.
- [48] S. Kramm and A. Benschair, "Obstacle detection using sparse stereovision and clustering techniques," in *IEEE Intelligent Vehicles Symposium*, 2012, pp. 760-765.
- [49] G. Dubbelman, W. Van der Mark, J. C. Van Den Heuvel, and F. C. A. Groen, "Obstacle detection during day and night conditions using stereo vision," in *International Conference on Intelligent Robots and Systems*, 2007, pp. 109-116.
- [50] A. Broggi, A. Cappalunga, C. Caraffi, S. Cattani, S. Ghidoni, P. Grisleri, *et al.*, "TerraMax Vision at the Urban Challenge 2007," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 194-205, 2010.
- [51] S. Kuthirummal, A. Das, and S. Samarasekera, "A graph traversal based algorithm for obstacle detection using lidar or stereo," in *International Conference on Intelligent Robots and Systems*, 2011, pp. 3874-3880.
- [52] C. D. Pantilie and S. Nedevschi, "SORT-SGM: Subpixel Optimized Real-Time Semiglobal Matching for Intelligent Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 61, pp. 1032-1042, 2012.
- [53] J. I. Woodlill, G. Gordon, and R. Buck, "Tyzx DeepSea High Speed Stereo Vision System," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 41-41.
- [54] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2274-2282, 2012.
- [55] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593-600.
- [56] J.-y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker," *Intel Corporation, Microprocessor Research Labs*, 2000.
- [57] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, Vancouver, BC, Canada, 1981, pp. 674-679.
- [58] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 881-892, 2002.
- [59] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, pp. 11-15, 1972.
- [60] L. Di Stefano and A. Bulgarelli, "A simple and efficient connected components labeling algorithm," in *International Conference on Image Analysis and Processing*, 1999, pp. 322-327.

- [61] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381-395, 1981.
- [62] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
- [63] T. Hastie, R. Tibshirani, and J. Friedman, "Hierarchical clustering," in *The Elements of Statistical Learning (2nd ed.)*, ed: New York: Springer, 2009, pp. 520-528.
- [64] I. Giosan and S. Nedevschi, "Superpixel-based obstacle segmentation from dense stereo urban traffic scenarios using intensity, depth and optical flow information," in *Proceedings of IEEE 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, 2014.
- [65] A. Iloie, I. Giosan, and S. Nedevschi, "UV disparity based obstacle detection and pedestrian classification in urban traffic scenarios," in *Proceedings of IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, 2014.
- [66] I. Giosan, E. Olti, and S. Nedevschi, "Traffic road obstacles detection based on analysis of relative motion vectors," *Automation, Computers, Applied Mathematics*, pp. 217-224, 2013.
- [67] P. Kamavidar, S. Saluja, and S. Agrawal, "A Survey on Image Classification Approaches and Techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, 2013.
- [68] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, pp. 823-870, 2007.
- [69] D. Toth and T. Aach, "Detection and recognition of moving objects using statistical motion detection and Fourier descriptors," in *12th International Conference on Image Analysis and Processing*, 2003, pp. 430-435.
- [70] E. Rivlin, M. Rudzsky, R. Goldenberg, U. Bogomolov, and S. Lepchev, "A real-time system for classification of moving objects," in *16th International Conference on Pattern Recognition*, 2002, pp. 688-691 vol.3.
- [71] O. Javed and M. Shah, "Tracking and Object Classification for Automated Surveillance," *Proceedings of the 7th European Conference on Computer Vision-Part IV*, pp. 343-357, 2002.
- [72] M. A. Dewan, M. J. Hossain, and O. Chae, "Moving Object Detection and Classification Using Neural Network," in *Agent and Multi-Agent Systems: Technologies and Applications*. vol. 4953, N. Nguyen, G. Jo, R. Howlett, and L. Jain, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 152-161.
- [73] Z. Lun, S. Z. Li, Y. Xiaotong, and X. Shiming, "Real-time Object Classification in Video Surveillance Based on Appearance Learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8.
- [74] A. Khammari, F. Nashashibi, Y. Abramson, and C. Lurgeau, "Vehicle detection combining gradient analysis and AdaBoost classification," in *Proceedings of Intelligent Transportation Systems*, 2005, pp. 66-71.
- [75] C. Peng, D. Hirvonen, T. Camus, and B. Southall, "Stereo-Based Object Detection, Classification, and Quantitative Evaluation with Automotive Applications," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops 2005*, pp. 62-62.

- [76] Q. Li and F. C. Sun, "Obstacle Detection in Cluttered Traffic Environment Based on Candidate Generation and Classification," *International Journal of Computers, Communications & Control*, vol. 1, pp. 93-99, 2006.
- [77] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. II-762-II-769 Vol.2.
- [78] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context," *International Journal of Computer Vision*, vol. 81, pp. 2-23, 2009.
- [79] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 739-746.
- [80] P. Kohli, Ladicky, x, L., and P. H. S. Torr, "Robust higher order potentials for enforcing label consistency," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
- [81] M. Bertozzi, L. Bombini, P. Cerri, P. Medici, P. C. Antonello, and M. Miglietta, "Obstacle detection and classification fusing radar and vision," in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 608-613.
- [82] U. Meis, W. Ritter, and H. Neumann, "Detection and classification of obstacles in night vision traffic scenes based on infrared imagery," *IEEE Intelligent Transportation Systems*, vol. 2, pp. 1140-1144, 12-15 Oct. 2003 2003.
- [83] M. Himmelsbach and H. J. Wuensche, "Tracking and classification of arbitrary objects with bottom-up/top-down detection," in *IEEE Intelligent Vehicles Symposium*, 2012, pp. 577-582.
- [84] G. Monteiro, C. Premebida, P. Peixoto, and U. Nunes, "Tracking and classification of dynamic obstacles using laser range finder and vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006.
- [85] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, "Shape-based pedestrian detection," in *IEEE Intelligent Vehicles Symposium*, 2000, pp. 215-220.
- [86] D. Gavrila, "Pedestrian Detection from a Moving Vehicle," *Proceedings of the 6th European Conference on Computer Vision-Part II*, pp. 37-49, 2000.
- [87] Y. Hee-Deok and L. Seong-Whan, "Multiple pedestrian detection and tracking based on weighted temporal texture features," in *17th International Conference on Pattern Recognition (ICPR)*, 2004, pp. 248-251 Vol.4.
- [88] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 193-199.
- [89] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *IEEE International Conference on Computer Vision*, 2003, pp. 734-741 vol.2.
- [90] D. M. Gavrila and V. Philomin, "Real-time object detection for "smart" vehicles," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 87-93 vol.1.

- [91] M. Bertozzi, A. Broggi, P. Grisleri, T. Graf, and M. Meinecke, "Pedestrian detection in infrared images," in *IEEE Intelligent Vehicles Symposium*, 2003, pp. 662-667.
- [92] H. Nanda and L. Davis, "Probabilistic template based pedestrian detection in infrared videos," in *IEEE Intelligent Vehicle Symposium*, 2002, pp. 15-20 vol.1.
- [93] F. Kruse, F. Folster, M. Ahrholdt, H. Rohling, M. Meinecke, and T. Thanh-Binh, "Target classification based on near-distance radar sensors," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 722-727.
- [94] G. Gate and F. Nashashibi, "Using targets appearance to improve pedestrian classification with a laser scanner," in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 571-576.
- [95] D. M. Gavrila and S. Munder, "Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle," *International Journal of Computer Vision*, vol. 73, pp. 41-59, 2007.
- [96] S. Nedeveschi, S. Bota, and C. Tomiuc, "Stereo-Based Pedestrian Detection for Collision-Avoidance Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 380-391, 2009.
- [97] F. Chang, C.-J. Chen, and C.-J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Computer Vision and Image Understanding*, vol. 93, pp. 206-220, 2004.
- [98] A. Buades, B. Coll, and J.-M. Morel, "A Non-Local Algorithm for Image Denoising," presented at the Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
- [99] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-698, 1986.
- [100] R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno, "2D Euclidean distance transform algorithms: A comparative survey," *ACM Computing Surveys (CSUR)*, vol. 40, pp. 1-44, 2008.
- [101] C. Pocol, S. Nedeveschi, and M. A. Obojski, "Obstacle Detection for Mobile Robots, Using Dense Stereo Reconstruction," in *IEEE International Conference on Intelligent Computer Communication and Processing*, 2007, pp. 127-132.
- [102] S. Bota and S. Nedeveschi, "Vision based obstacle tracking in urban traffic environments," in *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2011, pp. 231-238.
- [103] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886-893 vol. 1.
- [104] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, pp. 10-18, 2009.
- [105] A. Mucherino, P. Papajorgji, and P. Pardalos, "k-Nearest Neighbor Classification," in *Data Mining in Agriculture*. vol. 34, ed: Springer New York, 2009, pp. 83-106.
- [106] C. Tomiuc and S. Nedeveschi, "Real-time pedestrian classification exploiting 2D and 3D information," *Intelligent Transport Systems, IET*, vol. 2, pp. 201-210, 2008.
- [107] S. Bota, S. Nedeveschi, and M. Konig, "A framework for object detection, tracking and classification in urban traffic scenarios using stereovision," in *IEEE 5th*

- International Conference on Intelligent Computer Communication and Processing*, 2009, pp. 153-156.
- [108] S. Bota and S. Nedevschi, "Tracking multiple objects in urban traffic environments using dense stereo and optical flow," in *International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 791-796.
- [109] I. Giosan and S. Nedevschi, "Building Pedestrian Contour Hierarchies for Improving Detection in Traffic Scenes," *Computer Vision and Graphics - Lecture Notes in Computer Science*, pp. 154-163, 2009.
- [110] I. Giosan, S. Nedevschi, and S. Bota, "Real time stereo vision based pedestrian detection using full body contours," in *IEEE 5th International Conference on Intelligent Computer Communication and Processing*, 2009, pp. 79-86.
- [111] I. Giosan and S. Nedevschi, "Multi-feature Real Time Pedestrian Detection from Dense Stereo SORT-SGM Reconstructed Urban Traffic Scenarios," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, Lisbon, 2014.
- [112] I. Giosan and S. Nedevschi, "A solution for probabilistic inference and tracking of obstacles classification in urban traffic scenarios," in *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2012, pp. 221-227.