



FACULTATEA DE AUTOMATICA SI CALCULATOARE

Ing. Ciprian POCOL

# TEZĂ DE DOCTORAT

## TEHNICI BAZATE PE STEREO VIZIUNE PENTRU DETECȚIA OBSTACOLELOR ÎN CADRUL SISTEMELOR DE ASISTARE A CONDUCERII AUTO

**Conducător științific,  
Prof. dr. ing. Sergiu Nedevschi**

### **Comisia de evaluare a tezei de doctorat:**

**PREȘEDINTE:**

- Prof. dr. ing. *Liviu Miclea* – Decan, Facultatea de Automatică și Calculatoare, Universitatea Tehnică din Cluj-Napoca;

**MEMBRI:**

- Prof. dr. ing. *Sergiu Nedevschi* – Conducător științific, Universitatea Tehnică din Cluj-Napoca;
- Prof. dr. ing. *Ștefan-Gheorghe Pentiuc* – Referent, Universitatea “Ștefan cel Mare” Suceava;
- Prof. dr. *Bazil Pârv* – Referent, Universitatea “Babeș-Bolyai” Cluj-Napoca;
- Prof. dr. ing. *Ioan Salomie* – Referent, Universitatea Tehnică din Cluj-Napoca.



# Cuprins

<b>CAPITOLUL 1. INTRODUCERE .....</b>	<b>6</b>
1.1. DOMENIUL ȘI MOTIVAȚIA.....	6
1.2. OBIECTIVELE TEZEI.....	6
1.3. STRUCTURA TEZEI.....	7
1.4. MULȚUMIRI .....	9
<b>CAPITOLUL 2. SISTEME AVANSATE DE ASISTARE A CONDUCERII AUTO (ADAS).....</b>	<b>11</b>
2.1. FUNCȚII ADAS .....	11
2.1.1 De atenționare .....	11
2.1.2 De siguranță activă .....	13
2.1.3 De informare .....	14
2.1.4 Alte funcții .....	14
2.2. ACHIZIȚIA ȘI PROCESAREA DATELOR.....	15
2.2.1 Senzori montați pe vehicule .....	15
2.2.2 Suport din partea infrastructurii rutiere.....	17
2.2.3 Sisteme de informații geografice (GIS).....	17
2.2.4 Comunicare între vehicule.....	18
2.3. NECESITATEA SENZORILOR VIZUALI .....	18
2.3.1 Schema bloc a unui sistem ADAS bazat pe stereo viziune.....	20
2.4. CÂTEVA APLICAȚII POTRIVITE PENTRU SENZORI DE STEREO VIZIUNE .....	21
2.4.1 Asistent pentru congestii (stop-and-go).....	21
2.4.2 Asistent de conducere în intersecții urbane .....	22
2.4.3 Asistent lateral .....	23
2.5. CONCLUZII .....	24
<b>CAPITOLUL 3. ÎNȚELEGEREA, MODELAREA ȘI ANALIZA RECONSTRUCȚIEI STEREO .....</b>	<b>25</b>
3.1. PAȘII RECONSTRUCȚIEI 3D STEREO .....	25
3.1.1 Parametrii camerelor .....	26
3.1.2 Calibrarea.....	28
3.1.3 Rectificarea .....	29
3.1.4 Alegerea trăsăturilor .....	30
3.1.5 Stereo matching .....	31
3.1.6 Reconstrucția 3D .....	32
3.2. CALITATEA RECONSTRUCȚIEI .....	34
3.2.1 Acuratețea.....	34
3.2.2 Erori. Magnitudinea erorilor .....	35
3.2.3 Modelarea acurateței metrice .....	39
3.3. CANTITATEA ȘI DISTRIBUȚIA PUNCTELOR 3D .....	45
3.3.1 Cantitatea .....	45
3.3.2 Distribuția.....	45
3.4. CONCLUZII ȘI CONTRIBUȚII.....	47
<b>CAPITOLUL 4. MODELAREA OBSTACOLELOR.....</b>	<b>48</b>
4.1. TIPURI DE MODELE ȘI SISTEME DE COORDONATE .....	48
4.2. MODELE FOLOSITE ÎN PRACTICĂ .....	50
4.2.1 Modelul dreptunghiular neorientat.....	50
4.2.2 Modelul dreptunghiular orientat .....	51
4.2.3 Modelul octogonal .....	52
4.2.4 Modelul poligonal .....	53
4.2.5 Modelul poligonal cu rezoluție constantă și înălțime variabilă (stixel) .....	55
4.2.6 Modelul poligonal cu rezoluție variabilă și înălțime constantă.....	55

4.2.7 Modelul cuboidal cartezian .....	55
4.2.8 Modelul cuboidal polar .....	56
4.2.9 Modelul poliedral .....	56
4.2.10 Alte modele .....	57
4.2.11 Mai multe modele pentru un obstacol .....	58
4.3. CONCLUZII, CONTRIBUȚII .....	61
<b>CAPITOLUL 5. ABORDĂRI EXISTENTE ÎN DOMENIUL DETECȚIEI DE OBSTACOLE .....</b>	<b>62</b>
5.1. CLASIFICAREA ABORDĂRILOR .....	62
5.2. DETECȚIA DIN IMAGINI MONO, DE INTENSITATE .....	63
5.2.1 Simetrie .....	64
5.2.2 Umbra .....	66
5.2.3 Rețele neuronale .....	67
5.3. DETECȚIA DIN IMAGINI MONO, COLOR .....	68
5.4. DETECȚIA DIN SECVENȚE DE IMAGINI .....	70
5.4.1 Diferența imaginilor IPM succesive .....	71
5.4.2 Diferența imaginilor perspective succesive .....	73
5.4.3 Detecția mișcării în imagini IPM .....	74
5.4.4 Detecția mișcării în imagini perspective .....	75
5.5. DETECȚIA DIN IMAGINI STEREO .....	77
5.5.1 Profunzime exprimată prin disparități .....	80
5.5.2 Profunzime exprimată cartezian .....	83
5.5.3 Concluzii .....	92
5.6. CONCLUZII, CONTRIBUȚII .....	94
<b>CAPITOLUL 6. SOLUȚIE ORIGINALĂ PENTRU DETECȚIA OBSTACOLELOR .....</b>	<b>95</b>
6.1. EVIDENȚIEREA ZONELOR OCUPATE .....	96
6.1.1 Selecția punctelor 3D de interes .....	96
6.1.2 Compresia punctelor 3D într-un spațiu compact, de tip grid de celule top-view .....	98
6.1.3 Gruparea celulelor în obstacole, pe criterii de densitate și vecinătate .....	102
6.1.4 Rafinarea frontierei obstacolelor .....	104
6.2. PROCESAREA PE DIRECȚIA VERTICALĂ .....	109
6.2.1 Extinderea obstacolelor înalte .....	110
6.2.2 Fragmentarea verticală .....	115
6.2.3 Rafinarea verticală .....	116
6.3. VALORIFICAREA FORMELOR OBSTACOLELOR .....	118
6.3.1 Fragmentarea obstacolelor pe criterii de concavitate .....	118
6.3.2 Orientarea obstacolelor .....	121
6.3.3 Fragmentarea pe criterii de convexitate .....	126
6.4. ALTE PROCESĂRI .....	144
6.4.1 Absorbția obstacolelor .....	144
6.4.2 Clasificarea obstacolelor de fundal .....	146
6.4.3 Filtrarea obstacolelor .....	147
6.5. CONCLUZII .....	147
<b>CAPITOLUL 7. CONCLUZII, CONTRIBUȚII .....</b>	<b>149</b>
7.1. CONTRIBUȚIILE TEZEI .....	151
<b>ANEXE .....</b>	<b>153</b>
<b>A. REFERINȚE BIBLIOGRAFICE .....</b>	<b>153</b>
<b>B. PUBLICAȚII .....</b>	<b>160</b>
<b>C. CITĂRI .....</b>	<b>163</b>
<b>D. RAPOARTE TEHNICE .....</b>	<b>164</b>
<b>E. LISTA DE FIGURI .....</b>	<b>168</b>



<b>F. ARTICOLE SELECTATE .....</b>	<b>171</b>
<b>G. CURRICULUM VITAE .....</b>	<b>191</b>

# Capitolul 1. Introducere

## 1.1. Domeniul și motivația

Teza de față se încadrează în **domeniul** roboților autonomi în general, respectiv al autovehiculelor parțial sau total autonome în particular. Mai exact, analizează și propune metode noi pentru detecția obstacolelor, folosind stereo viziunea.

Sistemele avansate de asistare a conducerii (ADAS = Advanced Driver Assistance Systems), disponibile pe autovehiculele din zilele noastre, implementează diverse funcții (senzoriale sau de acționare) ale unui sistem de conducere autonomă. Realizarea sistemelor ADAS reprezintă totodată un proces iterativ înspre atingerea dezideratului conducerii autonome complete.

Realizarea unui sistem senzorial pentru conducerea autonomă a autovehiculelor, folosind vederea artificială de tip stereo viziune, constituie **motivația** care stă la baza întregii teze. Există și alte tipuri de senzori (radar, laser), dar vederea artificială este cea mai potrivită, pentru că întreaga infrastructură rutieră este proiectată pentru o astfel de percepție (vizuală).

**Importanța** domeniului rezultă din faptul că cerința principală a sistemelor de conducere autonomă este înțelegerea structurii scenei: a spațiului liber (driveable) și a obstacolelor, putând fi derivate traiectoriile obstacolelor și posibilele coliziuni.

În martie 2014, producătorul auto Mercedes-Benz a lansat pe piață un sistem de detecție a pietonilor folosind stereo viziunea, în vederea frânării automate în cazul posibilelor coliziuni cu cea mai vulnerabilă categorie de participanți la traficul urban, pietonii. Același sistem este folosit și pentru pre-determinarea calității carosabilului, în scopul reglării automate a parametrilor sistemului adaptativ de suspensie, pentru oferirea unui grad de confort ridicat. Dezvoltatorul și producătorul de componente auto Robert Bosch a anunțat pentru anul 2015 lansarea unui sistem de frânare de urgență, bazat tot pe stereo viziune. Așadar, teza se înscrie într-o direcție de cercetare de **actualitate**.

Deși există multe abordări pentru detecția de obstacole folosind stereo viziunea, niciuna nu investighează mai profund detecția din cadre individuale. Teza de față umple acest gol al **necesității** unei abordări elaborate, care să detecteze obstacole prin exploatarea a cât mai multor indicii disponibile în cadre stereo individuale.

## 1.2. Obiectivele tezei

Pentru realizarea unor vehicule autonome, este nevoie de perceperea senzorială a elementelor importante ale structurii scenei. Aceste elemente pot fi împărțite în:

- suprafața solului, asimilată adesea cu noțiunea de spațiu liber; detecția sa asigură și funcția de separare sol-obstacole, restrângând astfel spațiul pentru detecția obstacolelor;
- obstacolele, putând fi împărțite în:
  - o obstacole de prim-plan: prezintă potențial de impact pentru vehiculul propriu;

- obstacole de fundal: în timp, ele pot deveni obstacole de prim-plan (mai ales dacă sunt în mișcare); detecția lor poate fi importantă pentru sisteme de cartografiere a infrastructurii.

Multe obstacole prezente în scene tipice de trafic prezintă particularități și pot fi detectate mai ușor: stâlpi, clădiri, pietoni, bicicliști, autovehicule ș.a. Totuși, pot apărea și multe alte clase de obstacole: vehicule din cele mai diverse, diferite tipuri de garduri, clădiri atipice, mobile, tufe, recipiente de colectare a deșeurilor, diferite forme de trunchiuri de copaci, stâlpi care au atașate alte elemente rutiere sau pietonale (indicatoare, semafoare) etc.

Astfel, **obiectivul** principal al tezei este investigarea și implementarea de metode noi și performante pentru detecția de obstacole generice, în ipoteza existenței unui modul separat care realizează separarea sol-obstacole. În plus, detecția se va face în cadre stereo independente, pentru că este foarte importantă o bună înțelegere a posibilităților de detecție la nivel de cadru, înaintea introducerii informațiilor de mișcare în cadre succesive. Pentru reprezentarea obstacolelor, se va folosi modelul cuboidal, incluzând o posibilă orientare. O atenție deosebită se va acorda modelării obstacolelor prin cuboide cât mai fidele.

Alte obiective, secundare, vizează înțelegerea sistemelor avansate de asistare a conducerii auto, propunerea de modele analitice pentru reconstrucția stereo, modelarea obstacolelor și, nu în ultimul rând, studiul metodelor existente în domeniul detecției de obstacole.

### 1.3. Structura tezei

Conținutul tezei constă din mai multe capitole grupate în trei părți: studiul contextului și contribuții aduse odată cu aceasta, prezentarea detaliată a unei metode originale de detecția obstacolelor și rezumarea concluziilor. O secțiune de anexe este atașată la sfârșitul tezei, pentru a oferi informații suplimentare.

Prima parte începe cu prezentarea sistemelor avansate de asistare a conducerii auto (**capitolul 2**): funcțiile acestora, modalitățile de percepție și procesare a elementelor scenei, argumentarea necesității stereo viziunii etc. Câteva sisteme concrete sunt detaliate. Se propune o structură originală a unui astfel de sistem bazat pe stereo viziune.

Pentru atingerea unor performanțe ridicate în abordările de detecție de obstacole, s-a impus de la sine înțelegerea cât mai profundă a reconstrucției 3D stereo, fiind realizate numeroase experimente. Astfel, **capitolul 3** prezintă pașii reconstrucției stereo: parametrii camerelor, calibrarea, rectificarea, alegerea trăsăturilor, potrivirea stereo și reconstrucția 3D. O atenție deosebită s-a acordat unei analize originale asupra calității, cantității și distribuției punctelor 3D.

Pentru testarea diferitelor abordări de detecția obstacolelor, s-au implementat diferite modele pentru reprezentarea obstacolelor, atât 2D cât și 3D, atât mai schematic cât și mai detaliate. Toate acestea, dar și altele întâlnite în literatura de specialitate, sunt prezentate și analizate în **capitolul 4**: modelul dreptunghiular neorientat, modelul dreptunghiular orientat, modelul octogonal, modelul poligonal, modelul poligonal cu rezoluție constantă și înălțime variabilă (stixel), modelul poligonal cu rezoluție variabilă și înălțime constantă, modelul cuboidal cartezian, modelul cuboidal polar, modelul cuboidal cu orientări multiple, modelul multi-element, modelul curbat, modelul poliedral.

În **capitolul 5** se clasifică și se prezintă diverse abordări existente, grupate în funcție de sursa datelor folosite: imagini mono de intensitate, imagini mono color, secvențe de imagini și imagini stereo. De asemenea, se face o analiză critică a acestora, prezentându-se alegerea trasăturilor relevante, avantaje, dezavantaje, modelele folosite pentru reprezentarea obstacolelor, dar și posibile îmbunătățiri. O atenție sporită s-a acordat abordărilor bazate pe stereo viziune, acestea fiind grupate în mod original după spațiul de procesare folosit preponderent; se face o analiză critică a acestor spații de procesare și a echipelor de cercetare.

Partea a doua, constând din **capitolul 6**, prezintă o abordare originală pentru detecția de obstacole din cadre stereo individuale. Abordarea constă dintr-o suită de pași și subpași de procesare. Dintre aceștia, unii sunt aplicați tuturor obstacolelor, iar ceilalți doar pentru anumite cazuri, în urma unor decizii luate automat pe baza unor algoritmi specifici de analiză a necesității lor.

Pașii detecției pot fi grupați pe mai multe direcții. Prima este cea a evidențierii zonelor ocupate. În ipoteza existenței unui modul separat care realizează separarea sol-obstacole (nu face parte din această teză), se selectează doar punctele 3D care fac parte din structurile de deasupra solului, structuri care pot prezenta pericol de impact cu vehiculul propriu. Aceste puncte (carteziene) sunt apoi transformate într-un spațiu comprimat orizontal (de tip grid top-view), ținând seama de posibilitățile reale de percepție a scenei prin intermediul stereo viziunii. Pe axa laterală, percepția este una polară, astfel încât coloanele spațiului comprimat reprezintă direcții optice polare; astfel se pot face cu ușurință raționamente legate de erorile de reconstrucție, dar și raționamente despre ocluzii. Rândurile spațiului comprimat surprind puterea de discernere a profunzimii, prin stereo viziune. În acest spațiu, celulele sunt grupate folosind criteriile de vecinătate și de densitate. Au fost luate măsurile necesare compensării diferiților factori care influențează vecinătatea și densitatea. Urmează o îmbunătățire a frontierei fiecărui obstacol, prin procesări la nivel de coloane individuale și la nivel de obstacol.

A doua direcție este cea a procesărilor pe direcția verticală. Pentru detecția fidelă a obstacolelor mai înalte decât autovehiculul propriu, de exemplu a camioanelor, obstacolele sunt extinse drept în sus, în măsura în care sunt puncte 3D, dar nu mai sus de 4,5 m. Dacă deasupra unui obstacol aflat pe sol, din diverse motive se află și alte puncte 3D, o analiză a distribuției verticale ajută la stabilirea înălțimii corecte a obstacolului.

O direcție importantă pentru modelarea fidelă a obstacolelor prin cuboide este cea a valorificării formelor obstacolelor. Adesea, mai multe obstacole apropiate sunt grupate împreună, dar forma frontierei grupului poate oferi indicii suplimentare. Astfel, dacă frontiera vizibilă (înspre cameră) are concavități, acestea sunt folosite pentru determinarea punctelor de fragmentare a grupului în obstacole. Apoi, partea cvazi-liniară cea mai lungă a frontierei vizibile poate indica orientarea fiecărui obstacol. În acest punct fiecare obstacol este modelat printr-un cuboid, posibil orientat. În pasul următor, se evaluează fidelitatea cuboidului, prin suprafața spațiului liber înglobat între laturile vizibile ale cuboidului și frontiera vizibilă. Dacă această suprafață e semnificativă, atunci se aplică un algoritm de fragmentare. Desigur, se ține seama de cazul în care suprafața liberă este cauzată de ocluziile din partea unor obstacole de prim-plan sau de cazul în care o parte a cuboidului nu este vizibilă în ambele camere ale sistemului stereo.

Atunci când un obstacol mai mic se situează în interiorul unui obstacol mai mare, are loc un proces de absorbție. Pe baza analizei ocluziilor dintre obstacole, acestea sunt împărțite în obstacole de prim-plan și obstacole de fundal. Obstacolele mici sunt rejectate, în urma analizei numărului de puncte 3D, a dimensiunilor carteziene și a dimensiunilor polare; pentru obstacolele de fundal, se folosesc parametri care înlesnesc rejectarea.

În partea a treia, constând din **capitolul 7**, se prezintă concluziile și contribuțiile tezei. Se va putea observa că și în capitolele care se încadrează la “studiul contextului”, sunt prezentate unele propuneri, analize și viziuni originale. De asemenea, se va putea observa că metoda originală propusă pentru detecția obstacolelor din cadre stereo individuale, prin multe problematizări și soluționări, este cea mai elaborată metodă de ultimă oră.

Anexele oferite completează amplasarea științifică a tezei:

- Anexa A: referințe bibliografice;
- Anexa B: lista a 20 de articole publicate la conferințe, trei articole publicate în reviste naționale și două capitole de carte;
- Anexa C: prezintă o listă cu primele 5 cele mai citate lucrări, însumând 393 dintr-un total de 480 de citări;
- Anexa D: lista cu 28 de rapoarte tehnice întocmite în cadrul a 10 contracte de cercetare;
- Anexa E: lista de figuri;
- Anexa F: trei articole selectate;
- Anexa G: Curriculum Vitae.

## 1.4. Mulțumiri

În primul rând, mulțumesc îndrumătorului meu, profesorul Sergiu Nedevschi, care a avut multă răbdare cu mine, m-a inspirat și mi-a oferit cadrul logistic necesar activității de cercetare. Pentru noi toți, este un exemplu de energie debordantă, de dedicare, de capacitate intelectuală și de profesionalism. A ridicat constant nivelul și vizibilitatea grupului de cercetare, a catedrei de Calculatoare, a facultății de Automatică și Calculatoare, a Universității Tehnice din Cluj-Napoca și a României.

Mulțumesc de asemenea comisiei de evaluare a tezei de doctorat, pentru efortul considerabil de a-mi evalua teza și de a participa la susținerea publică.

De-a lungul anilor, mulți mentori au ajutat la formarea mea profesională. Fiind imposibil să mi-i amintesc pe toți, mă voi limita la o parte dintre aceștia, din perioada mea pre-universitară: Sabina Sigmirean, Rafila Ise, Csaba Ilieș, Monica Izgărian, Iosif Hossu, Mariana Udrescu, Daniela Rad, Sorin Vaida, Sabin Ilieș, Gelu Tomșa, Cristian Rusu, Alexandru Băban, Simion Miheț, Radu Odorheanu, Ioan Târnovan, Ana Negreanu.

Atenția mi se îndreaptă spre colegii din echipa de cercetare, alături de care am prins rădăcini în acest domeniu special. Totodată, în timpul multelor proiecte pentru Volkswagen AG, unii angajați ai acestei companii ne-au devenit adevărați colegi, care ne-au sprijinit în diverse moduri.

Sprijinul, înțelegerea și răbdarea familiei și a prietenilor mi-au conferit un adevărat confort sufletesc. De-atâtea ori le-am spus: “Azi nu pot, am de lucru la doctorat”, “Am de scris la teză”, “Termin la vară”, “Termin la iarnă”, “Termin anul viitor”, “Termin la vară”. Și m-au înțeles. Am văzut că mă înțeleg cu adevărat.

De asemenea, țin să amintesc tonifierea provenită din admirarea lumii alpine. Acolo am văzut perseverența și dăruirea pentru un anumit domeniu, munca îndelungată pentru proiecte personale nerăsplătite material, camaraderia, sinceritatea, angajamentul, cunoașterea și recunoașterea propriilor limite și respectarea limitelor colegilor. Sportul rege nu este nici fotbalul, așa cum zic mulți, nu este nici culturismul, așa cum zice Arnold Schwarzenegger. Sportul rege este alpinismul, pentru că traseul nu poate fi întrerupt în orice moment, ci trebuie continuat până la loc de adăpost (acesta fiind adesea la distanță de câteva ore de mers), de multe ori în condiții de oboseală, căldură, frig, foame, sete, ploaie, ninsoare, viscol, ceață, întuneric, răni, dificultăți de orientare. Cercetătorii, alpiștii și rockerii au un avantaj comun: își doresc să înțeleagă ce nu înțeleg majoritatea oamenilor.

Mai sunt și mulțumiri personale de ordin sufletesc, spiritual și contemplativ, care se transmit mai mult prin simțire și mai puțin prin rațiune, în duh, parcă în afara timpului și a spațiului.

\*\*\*

În contextul național al plagiarismului la nivel “înalt”, am ales finalizarea și publicarea acestei teze, pentru a răspunde cu bine la rău.

## Capitolul 2. Sisteme avansate de asistare a conducerii auto (ADAS)

Deși calea spre autovehicule autonome este încă lungă, sub-părți ale acestor sisteme sunt realizabile, sau vor fi realizabile în viitorul apropiat. Acestea sunt numite sisteme avansate de asistare a conducerii (ADAS = Advanced Driver Assistance Systems). Pe măsură ce se dovedesc ca funcționează corect și că sunt sigure, pot fi introduse în producția de masă.

În cazul în care un astfel de sistem nu este destul de fezabil, poate fi încadrat la categoria sisteme de confort, asistând șoferul la condus și doar atenționându-l în cazul apariției unor situații periculoase. E posibil ca unui astfel de sistem să i se permită efectuarea unor manevre, dar șoferul rămâne responsabil pentru supravegherea funcționării (de exemplu sistemul Autonomous Cruise Control – ACC [WikiACC]). Atât sistemele automate cât și șoferii umani au un grad de imperfecțiune. Pentru a se justifica folosirea unui sistem automat, conform [FHWA96], acesta ar trebui să fie de cel puțin două ori mai sigur decât șoferii umani, dar se recomandă să fie de cinci ori mai sigur.

Un sistem de conducere autonomă ar fi obținut prin mai multe nivele intermediare, fiecare fiind compatibil cu cele anterioare (conținându-le, eventual îmbunătățite) și anticipând nivelele viitoare. În [IEEE98] se propune o evoluție pe cinci nivele:

- Nivel 0: autonomous cruise control prin comanda accelerației (fără frânare);
- Nivel 1: comunicare între vehicule, eventual și comunicarea unor informații de către infrastructura rutieră (de exemplu viteza maximă);
- Nivel 2: frânare automată de urgență;
- Nivel 3: păstrarea automată în cadrul benzii de circulație curente;
- Nivel 4: schimbarea automată a benzii de circulație;
- Nivel 5: algoritmi de evitarea coliziunilor;
- Alte nivele ar putea viza funcții precum planificarea rutei și schimbarea automată a traseului pentru a evita segmente aglomerate.

Toate aceste funcții necesită componente foarte complicate, atât software cât și hardware (senzori, actuatori, unități de procesare, rețele de comunicare, interfețe om-masină – HMI: Human-Machine Interface).

### 2.1. Funcții ADAS

În principiu, funcțiile ADAS pot fi împărțite în mai multe categorii. În continuare sunt prezentate câteva dintre acestea.

#### 2.1.1 De atenționare

Funcțiile de atenționare pot fi parte a unor funcții de siguranță activă. Ele doar observă și atenționează, fără a acționa.

Câteva exemple sunt:

- Atenționare la părăsirea nejustificată a benzii curente (LDW: Lane Departure Warning): în principiu, este atenționat șoferul atunci când părăsește banda curentă fără a semnaliza corespunzător.
- Asistare la schimbarea benzii (LCA: Lane Changing Assistance): la schimbarea benzii curente este posibil ca pe noua bandă să se apropie din spate un alt vehicul. Acesta poate fi neobservat de către șofer dacă se apropie cu viteză mare sau dacă se află în unghiul mort. Dacă șoferul semnalizează intenția de schimbare a benzii, sistemul poate oferi asistență în astfel de cazuri.
- Atenționare pentru coliziune frontală (CWS: frontal Collision Warning System): crește siguranța atenționând șoferul despre posibile coliziuni cu obstacole aflate în față, statice sau în mișcare. Un senzor care detectează obstacolele din față s-ar putea să nu fie suficient, ci să fie necesari și senzori care detectează traficul lateral pentru a monitoriza vehicule care intră brusc în fața vehiculului propriu.
- Asistența la parcare (Parking Assistance): în timpul manevrelor de parcare, obstacolele din apropierea vehiculului pot fi lovite pe fondul vizibilității scăzute. Pentru aceasta pot fi folosiți senzori de proximitate. În cazul unei posibile coliziuni pot fi emise avertismente sonore sau poate fi acționată frâna automat. Unele sisteme mai noi pot afișa imaginea “din satelit” a vehiculului propriu și a spațiului înconjurător. Aceasta necesită patru camere care acoperă tot spațiul din jurul vehiculului (360°), iar cele patru imagini de tip perspectivă sunt transformate în imagini IPM, formând împreună o imagine satelit completă.
- Monitorizarea oboselii șoferului (Driver’s Drowsiness Monitoring): detectează dacă șoferul e pe punctul de a adormi sau dacă nu este capabil să ofere suficientă atenție condusului. Este monitorizată clipirea ochilor șoferului și se detectează un anumit șablon specific șoferilor obosiți.
- Monitorizarea atenției șoferului: un exemplu de caz periculos este atunci când șoferul se uită pentru un timp mai îndelungat în una din oglinzile retrovizoare sau la panoul de bord, iar vehiculul din față încetinește.
- Monitorizarea acționării pedalelor: de exemplu când șoferul ridică brusc piciorul de pe pedala de accelerație și începe să-l îndrepte spre pedala de frână. În acest caz vehiculul poate începe frânarea mai devreme.
- Monitorizarea ocupanților (Occupants Monitoring): în cazul unui accident, rănirea ocupanților din vehicul depinde de mulți factori, printre aceștia fiind și poziția pe scaun. Se studiază folosirea senzorilor termali sau a sistemelor de stereo viziune cu unghi mare pentru detectarea și monitorizarea poziției ocupanților. Nepurtarea centurii de siguranță, a dus la 42% din accidentele mortale pe timp de zi și la 61% pe timp de noapte, conform [NHTSA10].

Atenționarea șoferului poate fi făcută prin diferite tipuri de semnale:

- Semnale vizuale: LED-uri montate în oglinzi sau pe panoul de bord;
- Semnale acustice: pot fi emise prin sistemul audio al vehiculului sau prin sisteme dedicate;
- Semnale mecanice: mecanisme care induc vibrații, montate în volan sau în scaun. În cazul montării în scaun, s-ar putea folosi două astfel de dispozitive, câte unul pe fiecare parte, alertarea putând fi în partea stângă sau în partea dreaptă în funcție de direcția în care este părăsită banda curentă fără anunțarea intenției prin semnalizare (de exemplu pentru LDW).



## 2.1.2 De siguranță activă

Siguranța activă presupune preluarea controlului vehiculului în mod automat: frânare, accelerare, virare.

Funcțiile siguranței active sunt necesare în principal pentru evitarea accidentelor, dar și pentru confort:

- Frânare de urgență: se acționează automat frâna în cazul în care se detectează posibilitatea unei coliziuni. În cazul sistemului ACC (Autonomous Cruise Control), frânarea are loc pentru a limita viteza proprie la viteza vehiculului din față.
- Stop and go: folosirea automată a accelerației și a frânei pentru a menține o distanță de siguranță față de vehiculul din față, inclusiv prin oprirea totală. Reprezintă un succes pentru ACC.
- Asistent de păstrare a benzii (Lane Keeping Assistant – LKA): acționează direcția pentru a menține vehiculul în banda curentă de circulație. Atunci când șoferul acționează semnalizarea, acest sistem este dezactivat. Are nevoie de senzori care detectează scena din fața vehiculului.
- Schimbarea automată a benzii (Automatic Lane Changing): are nevoie de senzori care detectează toată scena din jurul vehiculului pentru a putea decide când sunt îndeplinite condițiile de siguranță pentru schimbarea benzii.
- Deplasarea în pluton (platooning): presupune existența unui vehicul condus de un șofer uman și alte vehicule care să-l urmeze automat, într-un lanț, fiecare urmându-l pe cel din fața sa. Este necesară folosirea unei rețele de comunicare între vehicule, pentru a fi comunicată tuturor vehiculelor din lanț intenția de frânare, accelerare și virare. De altfel, având diferite tipuri de vehicule în același lanț, este necesară luarea în considerare a tuturor parametrilor dinamici și statici: puterea motorului, eficiența frânării, încărcătura, lungimea, lățimea, parametrii de virare etc. Dacă vehiculele sunt relativ apropiate, un avantaj major este și acela al scăderii consumului de combustibil, pentru că doar primul vehicul întâmpină întreaga rezistență a aerului. O atenție deosebită e necesară algoritmilor pentru intrarea/ieșirea vehiculelor în/din lanț.
- Evitarea obstacolelor: pentru obstacole statice soluția este relativ simplă. În cazul apariției bruște a unui obstacol, problema devine complexă, pentru că și ceilalți participanți la trafic încearcă evitarea impactului sau limitarea daunelor, iar comportamentul lor este greu de prezis.
- Asistența la intersecție de drumuri (Roads Intersection Assistance): se referă la detecția indicatoarelor rutiere, a marcajelor de pe suprafața carosabilă, a semafoarelor, dar și a traficului transversal (care intră aparent brusc în scenă comparat cu traficul longitudinal). Într-un oraș aglomerat, este o mare probabilitate ca șoferii să nu observe elemente importante în intersecții: indicatoare și semafoare cu vizibilitate scăzută, marcaje de pe carosabil atunci când sunt obturate de alți participanți la trafic (ducând la încadrarea pe benzi greșite), treceri de pietoni, participanți la trafic de tip auto, velo, sau pietonal etc.
- Detectarea pietonilor (Pedestrian Detection): detecția participanților vulnerabili la trafic (pietoni, bicicliști, motocicliști), care apar brusc în calea vehiculului, este foarte importantă pentru că multe accidente cu urmări grave implică astfel de participanți.
- Parcare automată: Pe străzile aglomerate ale unui oraș, poate fi dificilă găsirea locurilor de parcare libere. Un sistem senzorial poate detecta locurile libere, urmând să anunțe șoferul la găsirea unui loc liber suficient de mare, iar apoi poate efectua automat manevre precise de parcare.

- Controlul stabilității (Stability Control Assistance): Atunci când viteza nu este adecvată la un anumit viraj și la anumite condiții de aderență, vehiculul poate derapa. Sisteme precum ESP (Electronic Stability Program) pot acționa frâna independent pe fiecare roată, astfel încât să contracareze fenomene de tip sub-virage sau supra-virage [WikiESP]. Totuși, trebuie precizat că aceste sisteme acționează doar în momentul când sesizează derapajul. Ar fi necesare sisteme preventive care să reducă viteza înainte de virajul în curbă sau înainte de manevre de evitare a obstacolelor.

### 2.1.3 De informare

În această categorie intră sisteme de hărți, de informații legate de traficul sau de condițiile rutiere ce urmează a fi întâlnite în drumul spre destinație, inclusiv informații de prognoză meteo.

Disponibilitatea diferitelor informații în vehicul poate ajuta la prevenirea accidentelor și oferă confort șoferului. Pentru funcțiile ADAS, este nevoie de informații privind managementul traficului, oferind: scurtarea timpului până la destinație, scăderea consumului de combustibil (și implicit scăderea cantității de noxe emirate), evitarea congestiilor etc.

Informațiile off-line și online pot fi folosite pentru o varietate de funcții:

- Sisteme de navigație: furnizează informații necesare șoferului pentru a urma calea către destinație. Acestea folosesc sisteme de poziționare globală (online – GPS) și unități optice (CD-ROM/DVD-ROM) pentru hărți rutiere digitale (off-line). Multe alte informații adiționale despre facilități și infrastructură pot fi stocate: restaurante, hoteluri, stații de alimentare cu carburant, service-uri auto, spitale, posturi de poliție etc.
- Pre-atenționare curbe: în funcție de viteza curentă, de poziția GPS și de curbura drumului, poate fi evaluat pericolul de derapaj.
- Vehiculul ar putea fi de asemenea anunțat despre viteza maximă prestabilită pentru un tronson de drum. Infrastructura rutieră poate transmite informații de acest gen. De importanță deosebită ar fi limitarea vitezei în zone aflate în construcție sau în reparații, sau în caz de condiții meteo dificile.

### 2.1.4 Alte funcții

- Apel automat de urgență 112 (Emergency auto-calling): în cazul detecției unui accident, sistemul ar urma să apeleze automat serviciile de urgență, comunicând și poziția exactă, scurtând astfel timpul de intervenție. În acest scop, ar putea fi disponibil și un buton “de urgență”, care să fie acționat manual pentru situații care nu pot fi detectate automat.
- Vedere îmbunătățită pe timp de noapte (Night vision): Chiar dacă numai 28% din conduci are loc noaptea, 55% din accidentele mortale au loc pe timpul nopții (conform unui studiu american făcut de NHTSA – National Highway Traffic Safety Administration). Sisteme bazate pe camere infraroșii pot fi folosite pentru a îmbunătăți percepția șoferului asupra traficului, pe timp de noapte. De exemplu, Cadillac Deville a făcut pionierat în acest sens, oferind un sistem care proiectează imaginea termică în partea de jos a parbrizului.

- Comenzi vocale: Șoferul poate fi semnificativ distrat în timpul manipulării a diferite butoane aflate în mașină sau în timpul navigării prin meniuri vaste. Sistemele cu comenzi vocale permit șoferului să nu-și ia ochii de la trafic în timp ce comandă diverse funcții prin folosirea vocii. Aceste sisteme sunt capabile să înțeleagă sute de comenzi, chiar și când șoferul vorbește cu accent.

## 2.2. Achiziția și procesarea datelor

Achiziția și procesarea datelor poate fi făcută:

- de către vehicul: prin folosirea de senzori care detectează suprafața drumului și a celorlalte vehicule, inclusiv a diverse obstacole generice;
- de către infrastructura rutieră: prin folosirea de emițători care comunică wireless informații despre viteza maximă, geometria drumului, geometria modificată a drumului în cazul lucrărilor de întreținere, condiții meteo, poziția altor vehicule care comunică cu infrastructura sau care sunt detectate de către infrastructură, existența unui accident etc.

Dintr-un alt punct de vedere, achiziția și procesarea datelor poate fi făcută:

- off-line: prin hărți digitale care descriu infrastructura statică a drumului;
- online (în timp real): prin descrierea vehiculelor dinamice și a obstacolelor statice existente în momentul de față.

Toate aceste sisteme de achiziție și de procesare a datelor trebuie interconectate pentru a compune un sistem integrat care să asigure caracteristici precum: robustețe, scalabilitate, redundanță, compatibilitate cu sistemele mai vechi și cu cele mai noi etc.

### 2.2.1 Senzori montați pe vehicule

Marele avantaj al senzorilor montați pe vehicule, în comparație cu dispozitivele care aparțin infrastructurii rutiere, este că necesită modificări doar pe vehicule individuale.

Există multe informații legate de principiile și limitările acestor senzori. Un sumar al celor mai folosiți senzori (radar, laser, camera) este prezentat în tabelul de mai jos.

După cum sugerează tabelul, camerele stereo reprezintă cel mai flexibil tip de senzor. Odată cu progresul tehnicii de calcul, a devenit posibilă stereo viziunea densă de timp real, cu imagini de rezoluție mare. Astfel, s-a deschis calea către multe aplicații complexe. Camerele cu apertură mare (wide angle) pot cuprinde o mare parte din scenă și pot surprinde detalii ale scenei îndepărtate în același timp. Senzorii bazați pe vedere artificială permit o multitudine de metode pentru detecția, urmărirea, clasificarea și recunoașterea obiectelor. De asemenea, astfel de senzori nu interferențiază între ei, camerele fiind din categoria senzorilor pasivi.

Totuși, camerele au unele limitări. De exemplu, acuratețea distanței măsurate prin stereo viziune scade repede cu distanța, mai exact într-o manieră pătratică cu distanța. Măsurarea vitezei obstacolelor se face doar indirect, prin urmărirea acestora în mai multe cadre. Perceperea structurii scenei poate fi afectată de condițiile atmosferice (ploaie, ceață), iar camerele obișnuite nu sunt suficient de sensibile în condiții slabe de iluminare (noaptea); camerele sensibile în spectrul infraroșu și camerele high dynamic range pot ajuta mult în astfel de condiții.

Senzor	Long Range Radar	Short Range Radar	Multi-Beam Lidar	Scanning Lidar	Camera
Tipul senzorului	77GHz Radar	24/79GHz Radar	Multi-beam Lidar	Scanning Lidar	Gray-scale/Colour Mono/Stereo Infra-red
Distanța măsurată de:	Timpul de întoarcere, schimbarea de fază		Timpul de întoarcere		Prin reconstrucție stereo
Viteza măsurată de:	Efectul Doppler		Variația distanței		Variația distanței
Distanța măsurată:	Foarte mare	Medie	Foarte mare	Medie	Mare, depinde de configurație
Apertura	Foarte mică	Mare	Mică	Foarte mare	Mare, depinde de configurație
Rezoluția laterală	Mică	Mare	Medie	Mare	Foarte mare
Rezoluția verticală	Nu		Doar cu mai multe raze		Foarte mare
Tehnologia	Pulse radar, FMCW, PSK		Pulse trigger, sampling, phase		CMOS, CCD, FIR
Localizarea senzorului	Posibil acoperit de material permisibil EM		Posibil acoperit de material permisibil IR		În spatele parbrizului
Senzitivitate la vreme	Mică		Ceață, ninsoare, ploaie (mică)		Ceață, ninsoare, ploaie (mică)
Senzitivitate la mizerie	Mica		Depinde		Da, dar ajuta stergatoarele
Detectia obstacolelor individuale	După distanță și viteză		După distanță și vecinătate		Sunt multe metode posibile
Clasificarea obstacolelor	Limitată		Limitată		Foarte flexibilă:, vehicule, pietoni, benzi, indicatoare, alte elemente etc
Constrângeri	Alocare oficială de frecvențe		Laser clasa 1 (nepericulos pentru ochiul uman)		Nu

### **Necesitatea fuzionării rezultatelor mai multor senzori (sensor fusion)**

Pentru că nu există un senzor care să funcționeze bine în toate condițiile (meteo, de iluminare, suprafață mare a spațiului analizat), se impune adesea folosirea simultană a mai multor senzori, de tipuri diferite, și fuzionarea rezultatelor acestora.

Senzorii radar long range pot detecta obstacole (inclusiv viteza acestora) aflate la distanță (profundime) mare, dar folosesc o apertură mică (cateva grade). Senzorii radar short range au o apertură mai mare (zeci de grade), dar sunt specializați pe distanțe mai mici. Problema este că undele radar nu sunt reflectate de materialele plastice (cum ar fi un tomberon de plastic), de

aceea senzorii radar sunt adesea montați cu succes în spatele părților din plastic ale caroseriei vehiculelor.

Senzorii laser (sau lidar) au o apertură maximă, limitată doar de alte elemente ale vehiculului care obstrucionează razele laser. Senzorul laser este unul rotativ, permițând o vedere de până la 360 de grade în cazul în care ar fi montat deasupra vehiculului. În caz de poluare excesivă, ceață, ninsoare sau ploaie, pot fi semnalate obstacole inexistente.

Sistemele bazate pe camere video au posibilități limitate de estimare a distanței, mai ales când aceasta este mare. Totuși, acestea sunt foarte potrivite pentru perceperea elementelor scenei pentru că întreaga infrastructură rutieră este realizată pentru a fi percepută în mod vizual.

În concluzie, pentru că fiecare tehnologie senzorială are propriile dezavantaje, se impune necesitatea folosirii simultane a mai multor tipuri de senzori, iar rezultatele să fie fuzionate ținând seama de avantajele și dezavantajele fiecărui senzor în parte.

### **2.2.2 Suport din partea infrastructurii rutiere**

Presupune comunicarea (wireless) dintre infrastructură și vehicule. Astfel, un vehicul care să beneficieze de informațiile furnizate de emițătorii infrastructurii trebuie să fie dotat cu echipament de comunicare wireless.

Pentru funcții precum menținerea automată a benzii sau schimbarea automată a benzii, posibile soluții ar fi:

- lipirea pe suprafața drumului a unor marcaje care să fie detectabile în orice condiții atmosferice, de exemplu prin senzori radar montați pe vehicule.
- implantarea unor emițători magnetici sub suprafața drumului și detectarea acestora prin magnetometre montate sub vehicule.

Ambele soluții necesită modificări relativ ieftine din partea infrastructurii rutiere și dispozitive ieftine din partea vehiculelor.

O soluție scumpă, dar foarte viabilă pe termen lung, este construirea de benzi adiționale dedicate în întregime vehiculelor autonome. Ar avea și avantajul unui nivel avansat de siguranță, pentru că nu ar interfera cu vehicule conduse manual. Echipamentul montat pe vehicule ar fi ieftin pentru că vehiculele ar fi controlate de infrastructură, care ar gestiona toate detaliile traficului. Pe alte benzi de circulație, aceste vehicule ar urma să fie conduse manual. Astfel de vehicule sunt numite “dual mode”.

### **2.2.3 Sisteme de informații geografice (GIS)**

S-au propus, experimentat și produs numeroase astfel de sisteme, și nu doar pentru domeniul auto. Câteva exemple sunt:

- Hărți off-line + GPS: folosite pentru sisteme de navigație ce furnizează informații necesare ghidării șoferului pe ruta spre destinație. Hărți rutiere digitale sunt stocate pe suport optic (CD-ROM/DVD-ROM), citite de sistemul audio, procesate de sistemul de navigație și afișate pe un ecran LCD. Un receptor GPS obține

- poziția geografică a vehiculului în timp real. Sistemul calculează ruta de la poziția curentă la destinația specificată și dă indicații vocale șoferului.
- Informații actualizate despre infrastructura rutieră și despre condițiile de trafic. O rețea de stații terestre deține informații actualizate despre congestiile de trafic, despre accidente și despre condițiile meteo de pe fiecare segment de drum. Aceste informații sunt transmise wireless la vehicule și apoi sunt comunicate șoferilor.

#### **2.2.4 Comunicare între vehicule**

Este de la sine înțeles că o rețea wireless care ar conecta vehiculele învecinate ar oferi multe beneficii, oferind o platformă pentru multe aplicații. Conectarea și conlucrarea cu stații terestre ar aduce și mai multe beneficii. În contrast cu rețelele clasice de calculatoare, de data aceasta ar fi vorba de conectivitate ad-hoc. Câteva aplicații utile ar fi:

- Anunțarea intenției: vehiculele își fac cunoscută intenția altor vehicule, punând în gardă șoferii acestora;
- Partajarea detecției: vehiculele echipate cu senzori pentru detecția elementelor scenei, pot partaja rezultatele detecției cu celelalte vehicule, acoperind astfel o arie mai mare, rezolvând problema ocluziilor, crescând siguranța/certitudinea detecției, ajutând vehiculele care au capacități mai scăzute de detecție sau fără astfel de capacități.
- Diseminarea datelor: în cazul congestiilor de trafic sau al accidentelor, această informație poate fi transmisă rapid și departe, din aproape în aproape, astfel încât vehiculele mai îndepărtate (la kilometri sau zece de kilometri) să poată alege rute alternative pentru a evita zona blocajului.
- Deschiderea unor porți, de exemplu la intrarea în instituții, pentru anumite vehicule.
- Acces la internet.
- Conversații vocale directe cu alți șoferi.

Un exemplu disponibil pe piață este sistemul CACC (Cooperative Adaptive Cruise Control) în care predecesorul său (sistemul ACC) a fost îmbunătățit prin comunicarea wireless a intenției de accelerare și frânare către ceilalți participanți la trafic, astfel încât sistemul CACC al acestora să poată răspunde mai repede [WikiCACC].

### **2.3. Necesitatea senzorilor vizuali**

Senzorii vizuali furnizează mai multe informații decât oricare alți senzori, dar necesită algoritmi complecși și putere mare de calcul. De-a lungul anilor, procesoarele au devenit tot mai puternice și mulți cercetători s-au ocupat de acest domeniu.

Fiind senzori pasivi, nu apare problema interferenței cu alți senzori și nici necesitatea alocării oficiale a unor frecvențe electromagnetice. Infrastructura rutieră nu trebuie modificată special pentru acești senzori.

Prin stereo viziune densă, se poate măsura profunzimea multor puncte, cu o precizie satisfăcătoare, dar este nevoie de o bună calibrare a parametrilor intrinseci și extrinseci care descriu geometria camerelor. Viteza obstacolelor poate fi determinată prin urmărirea acestora în cadre succesive. Procesul de urmărire are nevoie de siguranța că asociază același obstacol

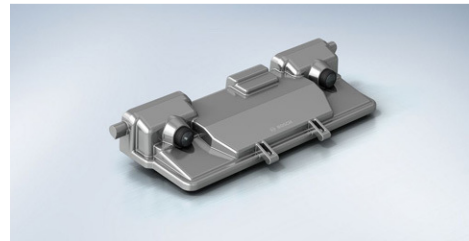
între oricare două cadre succesive. Acest lucru poate fi realizat doar prin verificări pe imagine, care nu e disponibilă în cazul altor tipuri de senzori. De asemenea, ocluziile parțiale pot fi tratate mai ușor folosind senzori vizuali.

Clasificarea obstacolelor este de asemenea mult mai potrivită senzorilor vizuali. Prin tehnici de recunoașterea formelor pot fi “citite” indicatoare rutiere, semafoare și semne desenate pe suprafața drumului (benzi de circulație, treceri de pietoni, săgeți de direcție, texte, diverse pictograme). Astfel de elemente sunt foarte importante pentru sisteme de păstrarea benzii curente, dar și pentru o conducere corectă în prezenta altor participanți la trafic.

Diverși producători au dezvoltat camere stereo pentru autovehicule:



a) Subaru



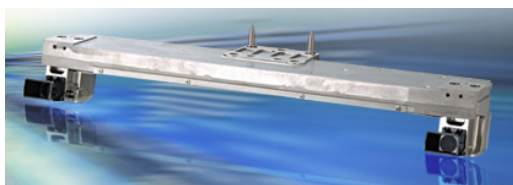
b) Robert Bosch



c) Continental



d) Mercedes-Benz



e) Kyocera



f) Hitachi



g) Carnegie Robotics

Fig. 2.3.1. Sisteme stereo comerciale sau demonstrative, pentru aplicații automotive

Pe timp de noapte, există tehnici potrivite pentru procesarea imaginilor termale. În plus, imaginile termale ar putea fi folosite pentru a oferi informații suplimentare atunci când imaginile normale sunt afectate de ninsoare, ceață sau ploaie. Unii senzori CMOS sunt sensibili atât la lumina din spectrul vizibil cât și la cea din spectrul infraroșu.

### 2.3.1 Schema bloc a unui sistem ADAS bazat pe stereo viziune

Un astfel de sistem are mai multe nivele, plecând de la componentele hardware și terminând cu decizii și acționări.

Pe primul nivel se află resursele hardware:

- vehiculul în sine, cu componente care asigură tracțiunea și schimbarea direcției;
- camerele stereo, care achiziționează imagini stânga/dreapta sincrone;
- senzorii de mișcare a vehiculului ajută la determinarea traiectoriei față de suprafața drumului și implicit față de componentele statice ale scenei; aceasta se poate determina și cu ajutorul odometriei vizuale, dar necesită algoritmi complecși și pot apărea probleme atunci când componentele statice sunt puține și cea mai mare parte a imaginii conține obstacole dinamice; în plus majoritatea autovehiculelor noi sunt echipate cu astfel de senzori necesari unor sisteme ca ABS, ASR, ESP, ACC;
- o unitate de procesare care să poată executa algoritmi de vedere artificială în timp real;
- în realizarea unui astfel de sistem, tot pe primul nivel se află cercetătorii – o resursă de bază –, după cum spunea regretatul profesor Alexandru Leluțiu: „Pentru mine, resursa umană este cea mai importantă resursă. O spun absolut, absolut categoric.”

Pe nivelul al doilea se află extragerea de informații primare din imagini, atât din punct de vedere stereo – măsurarea profunzimii –, cât și din punct de vedere mono – măsurarea fluxului optic. Tot aici intră și determinarea geometriei camerelor printr-un proces specific de calibrare. Un pas important în creșterea vitezei și simplificarea reconstrucției stereo dense este rectificarea imaginilor, adică transformarea imaginilor astfel încât să pară ca au fost achiziționate cu camere poziționate într-o configurație simplificatoare (numita configurație canonică).

Pe următorul nivel se află procesările la nivel de cadre individuale:

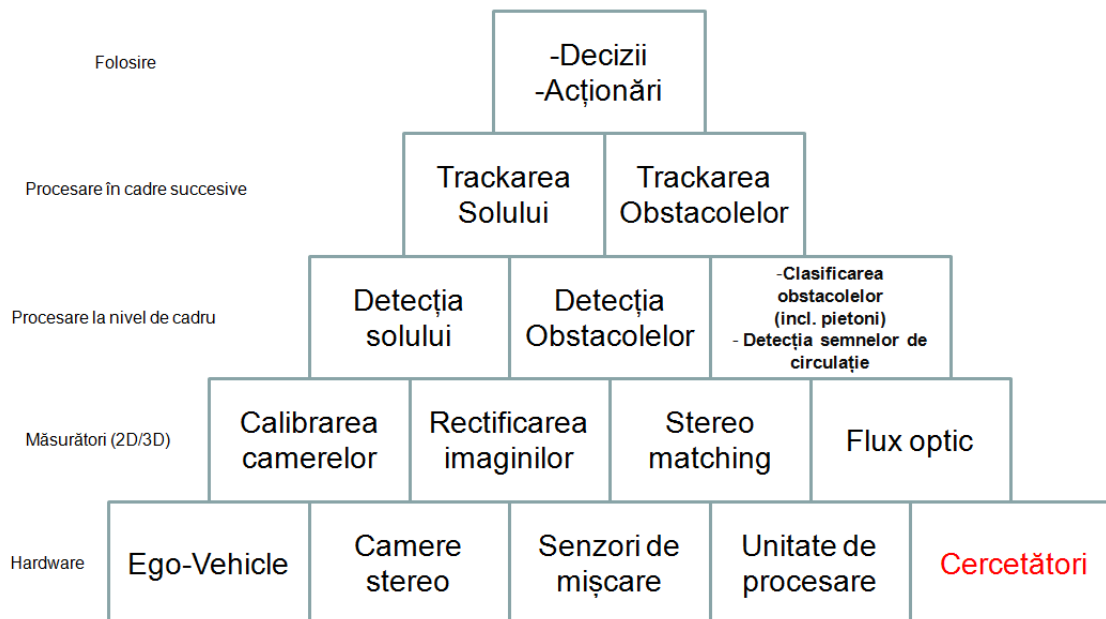
- detecția solului este cea mai importantă procesare pentru că oferă și un indiciu despre spațiul 3D în care se află obstacolele. Abordările pentru detecția solului exploatează unumite particularități ale geometriei acestuia;
- detecția obstacolelor trebuie să poată trata diverse situații: obstacole generice, obstacole dinamice care trec repede prin fața camerelor, ocluzii parțiale. Un caz particular e reprezentat de obstacolele scunde, gen borduri, care sunt o provocare pentru problema separării sol-obstacole;
- pentru diferite procesări de nivel mai înalt (decizii) este necesară clasificarea obstacolelor (de exemplu în cazul unui impact care este iminent indiferent de măsurile luabile, decizia corectă ar fi de exemplu evitarea accidentării unui pieton și lovirea unor elemente de vegetație). Detecția și recunoașterea semnelor de circulație (marcaje pe suprafața drumului, indicatoare rutiere, semafoare și chiar indicațiile poliștilului) sunt de mare importanță pentru înțelegerea corectă a contextului rutier.

Al patrulea nivel vizează acumularea și analiza rezultatelor obținute la nivel de cadre individuale, permițând: stabilizarea detecției, raționamente pe bază de votare, eliminarea artefactelor sporadice, medierea unor măsurători fluctuante, deschiderea de noi posibilități de



deteție (de exemplu deția semnalelor luminoase a altor vehicule) și determinarea vitezei de mișcare a obstacolelor.

Ultimul nivel este cel al deciziilor și a acționării. Toate celelalte nivele trebuie să funcționeze cât mai corect și precis, pentru că dețțiile fals pozitive sunt cele mai neplăcute și periculoase, materializate de exemplu printr-o frânăre puternică fără motiv la viteză mare. Pentru sisteme care sunt proiectate pentru funcții mai simple, deciziile pot fi destul de ușor de luat, dar cu cât ne apropiem de sisteme de tip conducere autonomă, cu atât trebuie înțeles cât mai corect contextul rutier, iar schemele decizionale devin mai complexe. Acționarea comenzilor acestor vehicule trebuie să fie de tipul drive-by-wire, pentru a putea fi conduse atât manual cât și automat.



## 2.4. Câteva aplicații potrivite pentru senzori de stereo viziune

### 2.4.1 Asistent pentru congestii (stop-and-go)

Obiectivele unui asistent în cazul unor congestii de trafic sunt:

- Reducerea necesității ca șoferul să conducă în situații de congestii de trafic, plictisitoare;
- Reducerea consumului de combustibil, prin optimizarea accelerărilor și a frânărilor;
- Reducerea numărului de accidente care au loc din cauza faptului că în situații de congestie atenția șoferului scade;
- Creșterea fluidității traficului;
- Urmărirea vehiculului din față, păstrând banda curentă.

Conceptul de stop-and-go presupune următoarele funcții:

- Păstrarea unei distanțe cât mai constante față de vehiculul din față;

- Păstrarea unei distanțe minime față de vehiculele laterale;
- Oprirea la lumina roșie a semaforului și la întâlnirea semnelor stop, chiar și atunci când vehiculul din față nu oprește;
- Cedarea priorității de dreapta atunci când este cazul;
- Evitarea coliziunilor cu pietonii care traversează strada.

Unele dintre aceste funcții sunt comune cu cele de la sisteme care asigură asistența în intersecții și de la sisteme care asigură asistența laterală.

O soluție pentru ghidarea vehiculului este bazată pe principiul remorcării virtuale: cunoscând distanța și unghiul vehiculului țintă, vehiculul propriu se va îndrepta către acesta, menținând o distanță constantă. Astfel, se obține aproximativ aceeași traiectorie ca și vehiculul țintă, dar în curbe strânse se taie curba. Acest comportament nedorit poate fi controlat dacă este cunoscută poziția vehiculului propriu în propria bandă de circulație [Franke99]. În cazul lipsei reprezentării benzilor, ar trebui să fie memorată și urmărită traiectoria recentă a vehiculului țintă.

Cel mai adesea, vehiculul țintă nu se află la distanță mare, astfel că stereo viziunea îl poate percepe cu o bună precizie. Resursele hardware sunt astăzi capabile să realizeze stereo viziune densă și flux optic dens, pe imagini cu rezoluție destul de mare, în timp real.

#### **2.4.2 Asistent de conducere în intersecții urbane**

În traficul urban, cele mai multe accidente au loc în intersecții. Astfel, un asistent de conducere în intersecții ar putea detecta posibilele situații periculoase atât la apropierea de intersecții cât și în timpul manevrelor din intersecții.

Sunt mai multe indicii ale prezenței intersecțiilor:

- treceri de pietoni;
- marcaje transversale desenate pe drum;
- săgeți desenate pe drum;
- indicatoare rutiere;
- semafoare;
- trafic transversal.

Prima sarcină a unui astfel de sistem este să recunoască apropierea de intersecție. Aceasta nu se poate rezolva cu ușurință folosind doar camere stereo, deoarece indiciile prezenței intersecțiilor pot suferi de probleme ca:

- lipsa acestora în unele intersecții;
- unele dintre acestea pot fi întâlnite și în alte situații;
- semafoarele și indicatoarele rutiere nu sunt întotdeauna ușor de detectat în scenariile urbane: pot fi ocluzate de alte vehicule din trafic sau de elemente statice ale scenei, locul amplasării lor nu este standardizat și poate fi diferit în funcție de diverse condiții existente la fața locului.

Indiciile care pot fi detectate de camere stereo sunt:

- treceri de pietoni și marcaje transversale;
- săgeți desenate pe suprafața drumului;

- traficul transversal: simpla detecție și trackare a vehiculelor care se deplasează transversal s-ar putea să nu funcționeze foarte bine, mai ales în prezența ocluziilor (parțiale) și a traficului rapid. O soluție mai fiabilă ar fi bazată pe detecția fluxului optic 3D: dacă acesta are multe componente transversale, atunci reprezintă trafic transversal (inclusiv pietoni pe trecere).

Odată recunoscută apropierea de intersecție, sistemul trebuie să determine topologia intersecției și să decidă regulile de circulație care trebuie aplicate. Aceasta este o altă sarcină dificilă pentru un sistem pe bază de viziune artificială, dar ar ajuta mult informațiile furnizate de un sistem de navigație, mai ales că acestea tind să conțină informații tot mai detaliate despre infrastructura rutieră.

Cunoscând regulile dintr-o anumită intersecție și intenția șoferului, sistemul poate detecta situații periculoase și poate atenționa șoferul sau chiar poate lua măsuri active. Pentru inferarea situațiilor periculoase, sistemul trebuie:

- să detecteze trecerile de pietoni;
- să detecteze (și să recunoască) indicatoarele rutiere;
- să detecteze starea semafoarelor;
- să detecteze vehiculul din față;
- să detecteze vehiculele care vin din partea stângă; acestea sunt periculoase în toate situațiile, mai puțin când virează la dreapta;
- să detecteze vehiculele care vin din partea dreaptă; acestea sunt periculoase în toate situațiile, mai puțin când vehiculul propriu virează la dreapta;
- să detecteze vehiculele care vin din față, în special când vehiculul propriu virează la stanga.

Sistemele de stereo viziune densă, cu lentile cu apertură mare (wide angle), pot detecta și urmări alte vehicule cu succes. Multele articole publicate pe această temă demonstrează fezabilitatea acestor sisteme.

### 2.4.3 Asistent lateral

Funcțiile unui sistem care oferă asistență laterală sunt:

- atenționarea șoferului în cazul părăsirii neintenționate (nesemnalizate) a benzii curente;
- ajutorul șoferului în menținerea benzii curente, în schimbarea benzii sau chiar în evitarea unor obstacole situate periculos de mult pe banda curentă;
- adaptarea unor parametri care țin de întreaga cinematică a vehiculului, în funcție de stilul de condus al șoferului (da, marii producători iau în serios acest aspect) și de alte considerente ce țin de situații concrete de trafic (de exemplu suprafață carosabilă cu aderență scăzută).

Pentru a realiza aceste funcții, sistemul trebuie să observe scena și: să recunoască banda curentă și benzile adiacente, să recunoască celelalte vehicule și pietonii, să recunoască obstacolele statice.

Sistemul trebuie să evalueze:

- tendința de a părăsi neintenționat banda curentă;
- tendința altor vehicule de a-și părăsi benzile lor;

- pericolul coliziunii cu obstacole statice;
- pericolul coliziunii cu obstacole aflate în mișcare (vehicule, pietoni).

În privința detecției benzilor de circulație, senzorii de stereo viziune sunt cei mai potriviți. Alte tipuri de senzori nu pot sesiza marcajele (radar) sau le sesizează cu un contrast slab față de restul carosabilului (lidar).

Recunoașterea obstacolelor – vehicule în mișcare, pietoni, obstacole statice – poate fi făcută și cu alte tipuri de senzori, dar viziunea artificială beneficiază de multe indicii suplimentare, mai ales că la proiectarea vehiculelor și infrastructurii rutiere se ține cont de posibilitățile vederii umane. De asemenea, prin vedere se pot face mai ușor clasificări ale obstacolelor: mașini, motociclete, biciclete, pietoni, stâlpi, ziduri etc.

## **2.5. Concluzii**

S-a analizat problematica sistemelor avansate de asistare a conducerii (ADAS = Advanced Driver Assistance Systems) la nivel structural, la nivel senzorial și prin detalierea unor sisteme concrete. S-a concluzionat că senzorii bazați pe stereo viziune sunt cei mai potriviți pentru detecția de obstacole în scene de trafic urban. În astfel de scene, profunzimea tipică este relativ mică (sub 30 de metri), iar complexitatea scenei este adesea prohibitivă pentru alte tipuri de senzori. Stereo viziunea oferă o cantitate mult mai mare de măsurători 3D decât senzorii de tip radar sau lidar. Vederea artificială are și avantajul că întreaga infrastructură rutieră este proiectată și realizată pentru a fi percepută în mod vizual.

S-a propus și o schemă generică, organizată pe mai multe nivele, pentru realizarea de astfel de sisteme bazate pe stereo viziune.

## Capitolul 3. Înțelegerea, modelarea și analiza reconstrucției stereo

Vederea artificială stereo (computer stereo vision) se referă la extragerea informațiilor 3D din imagini digitale, obținute de camere foto/video. Prin compararea informațiilor obținute din imagini achiziționate din două puncte diferite, se poate extrage informație 3D prin examinarea pozițiilor relative ale obstacolelor în cele două imagini. În biologie, procesul este numit stereopsis ([WikiStereoVision]).

În cazul amplasării orizontale a celor două camere, una în dreapta celeilalte, obiectele apar la aceeași înălțime în cele două imagini, dar la poziții orizontale diferite. Această diferență de poziție este numită *disparitate* (orizontală) sau, mai general, *disparitate binoculară*. Disparitatea poate fi ușor transformată în *profunzime*. Pentru oameni, disparitățile pot fi simulate prin generarea de imagini artificiale reprezentând două puncte de vedere diferite ale unei scene, câte una pentru fiecare ochi, prin folosirea unei tehnici numite *stereoscopie*. Perceperea profunzimii și a structurilor 3D este posibilă și prin folosirea unei singure camere video aflate în mișcare, chiar dacă nu la fel de acurat precum prin disparități binoculare ([WikiStereopsis]).

Liste mai mult sau mai puțin exhaustive, cu sisteme de stereo viziune, sunt oferite de [Stereo01] și [Stereo02]. În lucrarea de față se va considera un sistem cu două camere, montate fix pe un suport rigid, amplasate lateral una față de alta, numit pe scurt *sistem stereo*.

### 3.1. Pașii reconstrucției 3D stereo

În cazul în care sistemul stereo se află în mișcare sau unele obiecte din scenă se află în mișcare, o cerință pentru procesul de reconstrucție stereo este ca cele două camere să captureze imaginile simultan, exact în același moment. *Sincronizarea* se poate face prin trimiterea unui semnal electric (trigger) la cele două camere, de la un dispozitiv extern, camerele fiind prevăzute cu această posibilitate. O altă tehnică este funcționarea de tip master-slave, în care una dintre camere funcționează la o cadență dată și transmite și la cealaltă cameră semnalele necesare. Mai exact, cele două camere trebuie să aibă același moment de începere a expunerii și aceeași durată a expunerii.

O posibilitate propusă de [Catadioptric01] și de [Catadioptric02] este construirea unui sistem *catadioptric*, cu o singură cameră și cu un sistem de oglinzi care simulează amplasarea a două camere virtuale în două poziții diferite (Fig. 3.1.1). Se elimină astfel problema sincronizării, dar se înjumătățește FOV. În plus, e posibil ca soluția constructivă să necesite amplasarea sistemului pe verticală (stereo vertical), ceea ce nu-și găsește locul în sisteme ADAS.

O altă posibilitate, în condiții de cameră și scenă statice, este folosirea unei singure camere care este folosită pentru a achiziționa imagini din poziții diferite. Tehnica se numește *structure from motion* (structura din mișcare). Din nou, nu este aplicabil în sisteme ADAS.

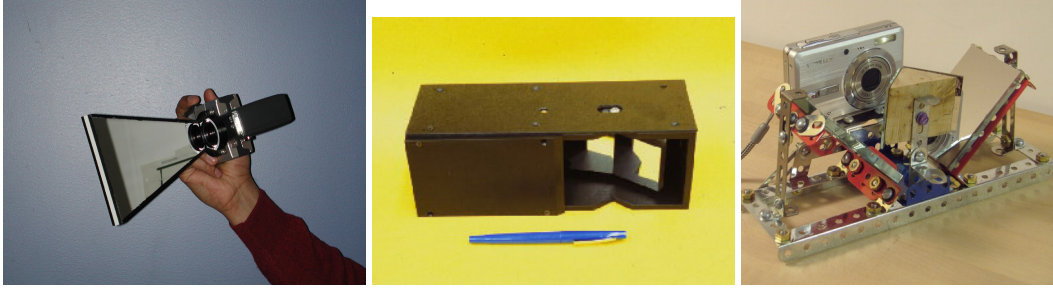


Fig. 3.1.1. Sisteme catadioptrice stereo, cu o singura cameră

Revenind la sistemul stereo cu două imagini sincronizate, pașii procesului de reconstrucție 3D, începând cu determinarea pozițiilor relative a celor două camere și terminând cu obținerea punctelor 3D, sunt prezentați în sub-secțiunile următoare.

### 3.1.1 Parametrii camerelor

După cum s-a mai spus, centrul stereoviziunii presupune identificarea unei trăsături (obiect, porțiune de obiect) dintr-o imagine în cealaltă imagine, cel mai adesea din imaginea stângă în imaginea dreaptă, folosind metrici de similaritate. Apar două mari întrebări:

- 1) Unde să se facă acea căutare în imaginea dreaptă? Dacă s-ar căuta în toată imaginea, ar fi foarte costisitor și ar fi și probabil să se găsească mai multe trăsături similare.
- 2) Odată trăsătura găsită, cum se poate calcula profunzimea?

Dacă s-ar cunoaște geometria sistemului stereo, răspunsurile la cele două întrebări ar fi:

1) Trăsătura reală din scenă împreună cu cele două poziții ale sale (fizic vorbind, în planele celor doi senzori ai camerelor), determină un plan numit *plan epipolar*, care, mai mult, include centrele optice ale celor două camere. Planul epipolar intersectează planele senzorilor generând câte o linie dreaptă pe fiecare senzor, numită *linie epipolară*. Deci spațiul de căutare se reduce la linia epipolară în imaginea dreaptă (Fig. 3.1.2 [Ares]).

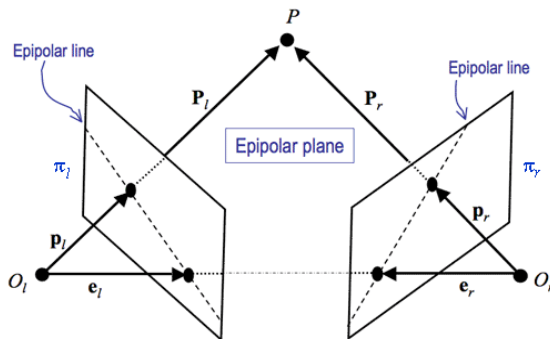


Fig. 3.1.2. Constrângerea epipolară

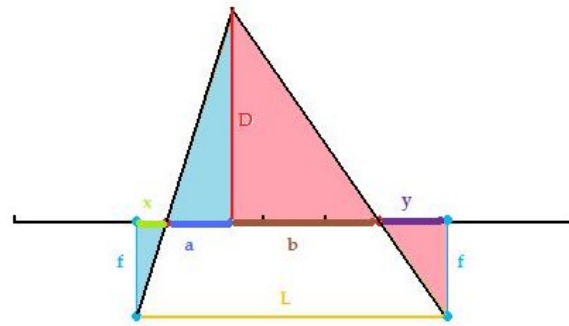


Fig. 3.1.3. Triangularea

2) Profunzimea se calculează folosind *triangularea*, adică un raționament geometric bazat pe triunghiuri asemenea din spațiul lumii reale și din spațiul camerei (Fig. 3.1.3 [StereoVision]).

Geometria sistemului stereo descrie procesul optic de proiecție a elementelor scenei pe planele imaginilor. În acest proces intervine geometria internă a fiecărei camere și poziția și orientarea fiecărei camere în scenă, adică geometria externă.

Geometria internă a unei camere este descrisă de *parametrii intrinseci* [OpenCVcalibmono]. Aceștia se împart în trei grupe:

- Coeficienții de distorsionare: fără a intra în detalii, aceștia sunt:
  - o radiali, având o influență direct proporțională cu distanța – raza – față de punctul principal; notându-i  $k_1$ ,  $k_2$  și  $k_3$ , pixelul  $(x_{\text{corrected}}, y_{\text{corrected}})$  are corespondent în imaginea inițială pixelul  $(x, y)$  prin transformarea:

$$\begin{aligned}x_{\text{corrected}} &= x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\y_{\text{corrected}} &= y(1 + k_1r^2 + k_2r^4 + k_3r^6)\end{aligned}$$

unde  $r$  este distanța de la centrul optic al imaginii.

Notă: coordonatele pixelilor sunt exprimate aici relativ la centrul optic al imaginii.

- o tangențiali, descriind imperfecțiunea paralelismului dintre planul lentilei și planul sensorului; notându-i  $p_1$  și  $p_2$ , pixelul  $(x_{\text{corrected}}, y_{\text{corrected}})$  are corespondent în imaginea inițială pixelul  $(x, y)$  prin transformarea:

$$\begin{aligned}x_{\text{corrected}} &= x + (2p_1xy + p_2(r^2 + 2x^2)) \\y_{\text{corrected}} &= y + (p_1(r^2 + 2y^2) + 2p_2xy)\end{aligned}$$

- Punctul principal: este locul în care raza optică principală (perpendiculară pe planul lentilei, în centrul acesteia) intersectează planul sensorului. Numeric acesta se exprimă în coordonate  $x, y$  în spațiul imaginii, exprimate în pixeli, cu precizie de sub-pixel. Fie acest punct  $(c_x, c_y)$ .
- Distanța focală: este distanța dintre centrul optic al lentilei și sensorul camerei. Ea se exprimă metric sau în pixeli, relația dintre cele două unități de măsură făcându-se prin intermediul dimensiunii fizice, metrice, a unui pixel în planul sensorului. Pentru generalizare, se pot folosi distanțe focale diferite pe cele două axe, fiind de ajutor atunci când se exprimă în pixeli iar forma unui pixel nu este patrică ci dreptunghiulară; le notăm cu  $f_x$  și  $f_y$ .

Pentru a descrie proiecția unui punct 3D  $(X, Y, Z)$  în spațiul imaginii într-un mod elegant, se apelează la calcul matricial, folosind homografia:

$$\begin{bmatrix}u \\ v \\ w\end{bmatrix} = \begin{bmatrix}f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1\end{bmatrix} \begin{bmatrix}X \\ Y \\ Z\end{bmatrix}$$

iar pentru obținerea coordonatelor  $x, y$  în spațiul imaginii mai rămân de făcut împărțirile  $x = u/w$  și  $y = v/w$ , cu observația că  $w = Z$ . Folosirea matricilor aduce beneficii precum:

- formule scrise simplificat,
- transformările inverse se obțin prin inversarea matricilor,
- mai multe transformări succesive pot fi condensate într-una singură obținută prin înmulțirea matricilor corespunzătoare.

Matricea  $A$  se numește matricea camerei (camera matrix), și înglobează toți parametrii constanți:

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Geometria externă a unei camere este descrisă de *parametrii extrinseci*. Aceștia presupun alegerea unui sistem cartezian de coordonate al scenei și descriu roto-translația fiecărei camere în acest sistem. Rotația (sau orientarea) constă din trei unghiuri, câte unul în jurul fiecărei axe, iar translația (sau poziția) constă din cele trei coordonate pe axe. Dacă sistemul este ales cu originea în camera stângă și cu orientarea ca și cea a camerei stângi, atunci parametrii extrinseci ai camerei drepte sunt numiți *parametrii relativi* ai celor două camere. Dintre parametrii relativi, poziția laterală a camerei drepte (în raport cu camera stângă) este numită *baseline*. În definitiv, se poate lucra cu mai multe sisteme de coordonate, între care se pot face transformări.

Dacă roto-translația e descrisă de parametrii  $r_{ij}$  și  $t_i$ , atunci întreaga transformare dintr-un sistem arbitrar în spațiul imaginii este:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

sau mai scurt:

$$m = A[R | t]M$$

### 3.1.2 Calibrarea

Prin procesul de calibrare se determină geometria sistemului stereo: parametrii intrinseci și parametrii extrinseci. Procesul folosește puncte 3D cunoscute din scena reală, pentru care se pot determina coordonatele 3D în sistemul scenei, coordonatele în imaginea stângă și coordonatele în imaginea dreaptă. Proiecția unui punct 3D în spațiul imaginii este descrisă de o ecuație ale cărei necunoscute sunt chiar parametrii intrinseci și extrinseci. Numărul de puncte 3D necesar calibrării este egal cu numărul de necunoscute, rezultând un sistem de ecuații (procedura de calibrare se alege în așa fel încât sistemul să fie determinat). În practică, au loc diferite fenomene care introduc erori, imprecizii. De aceea, se folosesc mult mai multe puncte 3D, pentru medierea erorilor, iar sistemul de ecuații se rezolvă într-o manieră iterativă, până când eroarea medie a reproiecției punctelor 3D, față de coordonatele inițiale în imagini, scade sub un prag stabilit (de exemplu 0,2 pixeli). Pentru îmbunătățirea stabilității acestei rezolvări iterative, soluția este ca mai întâi să se determine parametrii intrinseci ai fiecărei camere, individual, și apoi parametrii extrinseci [OpenCVcalibstereo].

O metodă binecunoscută pentru calibrarea intrinsecilor folosește o tablă de șah cu pătrățele albe și negre, având laturi de lățime fixă și cunoscută (Fig. 3.1.4 [Bouguet]). Punctele 3D folosite sunt *colțurile* acestor pătrățele, care sunt ușor de detectat în imagini, cu precizie de sub-pixel, fiind la intersecția a patru pătrățele cu contrast mare, două albe și două negre. În acest sens, colțurile exterioare ale pătrățelelor exterioare nu se folosesc. Sistemul de coordonate 3D al scenei se alege, de exemplu, cu originea în colțul stânga-sus. Toate colțurile fiind într-un singur plan, pot fi considerate ca având o coordonată nulă, de exemplu  $z = 0$ , iar celelalte două coordonate sunt ușor de calculat.



Pentru calibrarea extrinsecilor, procesul e similar cu cel de calibrare a intrinsecilor. Sunt necesare minimum patru puncte vizibile în ambele imagini (Fig. 3.1.5), iar dacă sunt mai multe, rezultatul este mai bun, prin medierea erorilor. Coordonatele 3D ale punctelor fie se calculează automat (de exemplu ca la intrinseci, printr-un șablon tip tablă de șah), fie se măsoară manual, într-un sistem ales arbitrar.

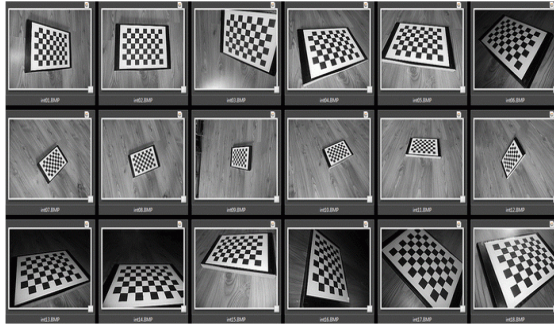


Fig. 3.1.4. Calibrarea intrinsecilor



Fig. 3.1.5. Calibrarea extrinsecilor (coordonatele 3D măsurate manual)

### 3.1.3 Rectificarea

Calibrarea parametrilor sistemului stereo fiind făcută, căutarea fiecărei trăsături din imaginea stângă se poate face în imaginea dreaptă de-a lungul liniei epipolare corespunzătoare. Totuși, această abordare are unele dezavantaje:

- Parametrii liniei epipolare corespunzătoare fiecărui punct din imaginea stânga trebuie calculați (sau precalculați);
- Căutarea în imaginea dreaptă are loc de-a lungul unei linii înclinată discrete, în trepte, provenind din linia epipolară matematică, continuă; o căutare la nivel de sub-pixel ar fi greu chiar și de imaginat.

Soluția este să se calculeze imaginile „rotite”, astfel încât liniile epipolare să devină orizontale, iar noile imagini să fie calculate prin interpolarea, la nivel de sub-pixel, a valorilor pixelilor din imaginile inițiale. Astfel, cele două dezavantaje amintite ar fi îndepărtate.

Transformarea este mai mult decât o rotație în planul imaginii, este o transformare în spațiul 3D, obținându-se imagini care arată ca și cum achiziția s-ar fi făcut cu două camere *canonice*. Configurația canonică a camerelor stereo presupune:

- Camere paralele, cele trei unghiuri de rotație relativă să fie nule;
- Deplasamentele între camere, pe înălțime și pe profunzime, să fie nule, doar cel pe lățime – baseline-ul – să fie nenul;
- Aceeași distanță focală;
- Același punct principal;
- În plus, trebuie să fie nedistorsionate (radial și tangențial); am putea numi această operație *desdistorsionare* (în eng. *undistort*).

Dacă s-ar lua aceste cerințe/transformări individual, ar fi necesare operații de rotație, translație, scalare, în spațiul imaginii și în spațiul 3D. Dintre acestea, translația 3D poate introduce artefacte, pentru că forțează privirea scenei dintr-un alt punct, din care fețele laterale ale obiectelor s-ar vedea altfel, iar ocluziile ar fi diferite.

Pentru a avea linii epipolare drepte, ambele imagini (stânga/dreapta) trebuie desdistorsionate. Desdistorsionarea poate fi evitată doar dacă parametrii de distorsionare sunt mici în raport cu

cerințele aplicațiilor concrete. În domeniul ADAS, fiind de dorit performanțe ale detecției la distanțe cât mai mari, se alege efectuarea desdistorsionării, astfel că, împreună cu celelalte transformări, ambele imagini sunt aduse la configurație canonică.

Transformarea necesară obținerii imaginilor canonice se numește *rectificare* (Fig. 3.1.6). Termenul nu include și înțelesul de desdistorsionare, dar acestea se fac împreună, pentru a avea o singură re-eșantionare (prin interpolare) a imaginilor și pentru a fi făcute într-o singură „trecere”. Există și posibilitatea unei transformări astfel încât doar una dintre imagini să fie efectiv rectificată, cealaltă fiind practic o copie 1 la 1 a celei inițiale, dar această variantă este mai optimă doar dacă nu se face desdistorsionare.

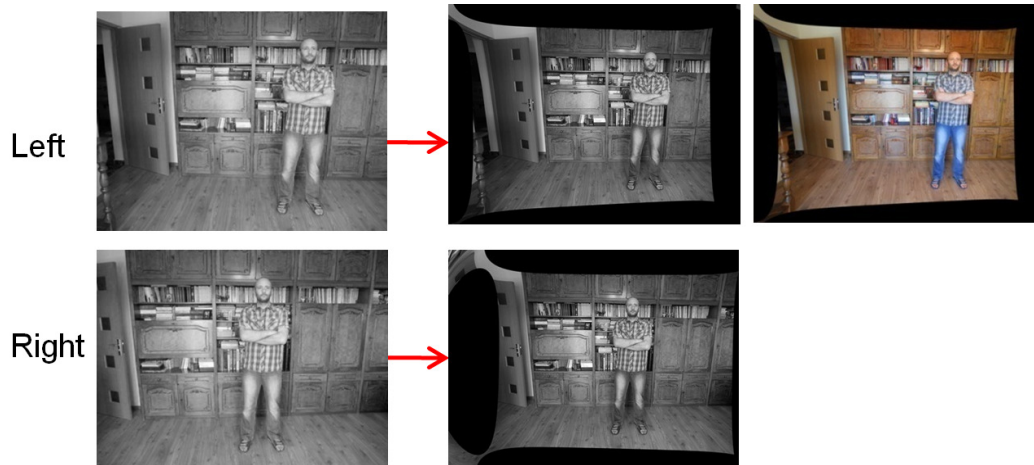


Fig. 3.1.6. Rectificarea imaginilor stânga și dreapta, inclusiv imaginea stânga color

Așadar, prin rectificare, ambele imagini vor deveni mai puțin clare, din cauza interpolării pixelilor din imaginile inițiale. Acesta este desigur un dezavantaj, dar soluțiile ingineresti sunt adesea imperfecte, de unde și frumusețea domeniului, arătând limitările și neputințele ființei umane, dar și provocând la autodepășire.

În urma rectificării, căutarea unei trăsături din imaginea stângă se face în imaginea dreaptă:

- pe aceeași linie ( $y$ , ca și în imaginea stângă) și
- în stânga coloanei curente ( $x$ ), până la o disparitate maximă, calculată în funcție de profunzimea minimă dorită. Trăsăturile aflate la profunzime foarte mare se află pe aceeași coloană în ambele imagini, disparitatea tinzând la zero.

### 3.1.4 Alegerea trăsăturilor

Încă de la început, trebuie precizat că sunt două abordări, una care face căutarea fiecărui pixel din imaginea stângă în imaginea dreaptă, numită *stereo viziune densă*, și una care face căutarea unui set restrâns de trăsături, numită *stereo viziune rară* (în eng. *sparse*). În secțiunea următoare se va vedea exact cum se face această căutare.

În stereo viziunea densă, prin termenul de „trăsătură” se înțelege fiecare pixel individual al imaginii, iar termenul de „alegere trăsăturilor” are astfel un înțeles diluat, alegându-se toți pixelii. Abia în pașii următori se va face o rejectare a pixelilor care nu au fost găsiți în imaginea dreaptă. Stereo viziunea densă este cea care este folosită în lucrarea de față.

În stereo viziunea rară, trăsăturile alese au semnificație în domeniul elementelor scenei: pixeli de muchie ([Oniga04]), pixeli înconjurați de textura puternică, segmente de muchii sau chiar petice de imagine. Față de stereo viziunea densă, are avantajul vitezei mari de procesare, prin alegerea unui set restrâns de trăsături, dar are dezavantajul unei consistențe mai scăzute, mai discontinue, a informației 3D. La nevoie, aceste trăsături pot fi folosite în stereo viziunea densă, pentru a aduce un plus de informații.

### 3.1.5 Stereo matching

Pasul cheie al măsurării profunzimilor prin stereo viziune este cel al găsirii trăsăturilor dintr-o imagine, care se potrivesc în mod unic cu trăsături din cealaltă imagine, rezultând astfel disparități. După cum s-a mai spus, căutarea, de regulă a unei trăsături din imaginea stângă în imaginea dreaptă, se face de-a lungul liniilor epipolare, din pixel în pixel. Potrivirea (*matching*-ul) poate fi evaluată folosind diverse metrice. Pentru creșterea unicității, se folosește pixelul curent împreună cu vecinii săi. Vecinătăți uzuale sunt de 5x5, 7x7, 9x9 pixeli, dimensiunea fiind impară pentru ca pixelul curent să fie central.

Una dintre metricile cele mai folosite este *suma diferențelor absolute* (SAD = Sum of Absolute Differences): se face suprapunerea vecinătății *pixelului căutat* (din imaginea stânga) peste vecinătatea *pixelului candidat* (din imaginea dreaptă) și se calculează suma dintre valorile absolute ale diferențelor pixelilor corespondenți. În cazul imaginilor grayscale cu 8 biti/pixel, valorile diferențelor sunt în intervalul -255..+255. Pixelul candidat care are valoarea SAD minimă este cel ales. Mai mult, folosind și valorile SAD ale celor doi pixeli vecini, disparitatea poate fi estimată cu precizie de sub-pixel.

Formula generică a valorii SAD este:

$$SAD = \sum_{x=-n}^{+n} \sum_{y=-n}^{+n} |I_L(x_L + x, y_L + y) - I_R(x_R + x, y_R + y)|$$

unde vecinătatea are dimensiunea de  $2n+1$ ,  $I_L$  și  $I_R$  sunt imaginile stânga/dreapta, iar  $(x_L, y_L)$  și  $(x_R, y_R)$  sunt pozițiile pixelilor centrali în cele două imagini.

În cazul imaginilor rectificate,  $y_R = y_L$ , iar despre  $x_R$  se poate afirma că este între  $x_L - MaxDisparity$  și  $x_L$ , astfel că, dacă  $d$  este disparitatea, putem scrie formula valorii SAD astfel:

$$SAD = \sum_{x=-n}^{+n} \sum_{y=-n}^{+n} |I_L(x_L + x, y_L + y) - I_R(x_L - d + x, y_L + y)|$$

Alte metrice sunt:

- Suma pătratelor diferențelor (SSD = Sum of Squared Differences): dă pondere prea mare diferențelor mai mari și pondere mică asemănarilor (diferențele mici).
- Transformata CENSUS: nu folosește direct valorile grayscale, ci folosește o descriere a pixelilor vecini în comparație cu pixelul central. Are avantajul insensibilității la cazul în care o trăsătură apare cu intensități diferite în cele două imagini, dar necesită putere mare de calcul, fiind pretabilă unor implementari hardware ([TYZX]) sau unor sisteme non-real-time.

Alte considerente:

- Folosirea unui subset al pixelilor vecini: permite vecinătăți mai mari, la putere de calcul limitată.
- Folosirea de raționamente suplimentare în cazul în care sunt mai mulți pixeli aproape la fel de potriviți ca și cel mai potrivit.
- Rejectarea pixelilor căutați atunci când eșuează găsirea celui mai potrivit candidat. Astfel, nu toți pixelii din imagine vor avea o profunzime determinabilă.
- Stereo-matching-ul din imaginea stânga în imaginea dreaptă, urmat de cel din imaginea dreaptă în imaginea stângă, poate beneficia de raționamente suplimentare.
- De asemenea, raționamente suplimentare pot fi aplicate luând în considerare simultan potrivirile tuturor pixelilor din imagine ([Haller10, Gehrig09]), nu doar pentru fiecare individual: tehnici global matching (GM) și tehnici semi global matching (SGM).
- Disponibilitatea recentă a unor tehnologii de calcul paralel (instrucțiuni SIMD, multiprocesoare, calcule făcute de procesorul plăcii grafice) au permis implementarea de tehnici real-time noi ([Pantilie11]).

Rezultatul pasului de stereo matching este o hartă (imagine) de disparități, valorile mai mari însemnând profunzimi mai mici. După cum s-a văzut, această hartă poate fi discontinuă.

### 3.1.6 Reconstrucția 3D

Disparitatea este măsura nativă a profunzimii rezultate din stereo viziune, iar împreună cu poziția în imagine oferă o vedere polară a structurii scenei. Pasul următor, numit *reconstrucție 3D*, face trecerea spre o vedere carteziană, metrică, a structurii scenei. Această trecere are, la rândul său, doi pași: trecerea în sistemul cartezian al camerei stângi, rezultând *puncte 3D*, și transformarea în sistemul cartezian al scenei (a doua fiind opțională).

Relativ la camera stângă, profunzimea exprimată metric, numită adesea și *distanță*, este invers proporțională cu disparitatea:

$$Z = \frac{b \cdot f}{D}$$

unde  $b$  este distanța laterală dintre camere (baseline-ul),  $f$  este distanța focală, exprimată în pixeli,  $D$  este disparitatea exprimată în pixeli, iar  $Z$  este distanța exprimată în aceeași unitate de măsură ca și  $b$  (de regulă m sau mm). Pozițiile laterală și pe înălțime sunt direct proporționale cu distanța și cu poziția în imagine față de punctul principal, formulele rezultând din triunghiuri asemenea:

$$X = (x - PP_x) \frac{Z}{f}$$

$$Y = (y - PP_y) \frac{Z}{f}$$

Unde  $x$  și  $y$  reprezintă poziția în spațiul imaginii, iar  $PP_x$  și  $PP_y$  reprezintă poziția punctului principal, toate acestea fiind exprimate în pixeli.

Pentru a avea punctele 3D exprimate într-un spațiu cartezian, cu o semnificație mai mare în spațiul scenei și independent de poziționarea sistemului stereo, se mai poate face o

transformare matematică (amintită și mai sus) între sistemul de coordonate al camerei stângi și sistemul de coordonate al scenei. Transformarea folosește parametri extrinseci obținuți în procesul de calibrare. Sistemul scenei este ales arbitrar, în funcție de aplicație: colțul camerei, punctul de pe drum de sub centrul botului ego-car etc. Dacă e nevoie, se pot face transformări suplimentare, în alte sisteme de coordonate, de exemplu într-un sistem static al lumii, în cazul în care ego-car se mișcă și se cunosc parametrii mișcării. De notat că aceste transformări pot fi comasate, folosind calcul matricial, pentru a face transformări din sistemul camerei direct în sistemul static al lumii, de exemplu pentru a acumula punctele 3D obținute în cadre succesive ([Geiger11]).

În Fig. 3.1.7, sunt prezentate punctele 3D, văzute dintr-o poziție arbitrară, obținute din reconstrucția scenei prezentate în Fig. 3.1.6, iar culorile punctelor fiind obținute din imaginea stângă color.

În general, sistemele de coordonate 3D folosite au următoarea orientare a axelor: axa X reprezintă distanța laterală, cu valori pozitive înspre dreapta, axa Y reprezintă înălțimea, cu valori pozitive în jos, iar axa Z reprezintă distanța, orientată în direcția în care privește observatorul. În acest fel, sistemul de coordonate este direct orientat.

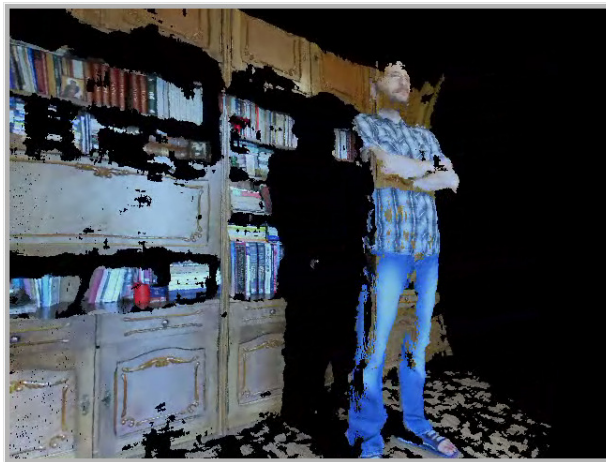


Fig. 3.1.7. Reconstrucție 3D

## 3.2. Calitatea reconstrucției

Vom împărți imperfecțiunile reconstrucției în două tipuri:

- *Acuratețea*, ca reprezentând limitările inerente de precizie a măsurării profunzimii, atunci când procesul de stereo matching găsește corect pixelul corespondent în imaginea dreaptă. Se mai poate numi și eroarea inerentă la nivel de sub-pixel, sau, și mai scurt, *eroarea de sub-pixel*.
- *Eroarea*, ca fiind imperfecțiunea cauzată de găsirea incorectă a pixelului corespondent în imaginea dreaptă.

Din punctul de vedere al algoritmilor de detecția obstacolelor, ambele tipuri de imperfecțiuni sunt adesea numite generic *erori*, fiind abateri ale profunzimii măsurate față de profunzimea reală. Înțelegerea acestora este necesară conform motto-ului:

„O buna detecție a obstacolelor necesită o buna înțelegere a erorilor reconstrucției.”

### 3.2.1 Acuratețea

Acuratețea reconstrucției este direct legată de acuratețea disparității. Aceasta este influențată de diverși factori, printre care:

- distribuirea semnalului luminos, de natură continuă, în pătrățelele discrete ale senzorului camerei;
- focalizarea imperfectă;
- zgomotul valorilor pixelilor. Acesta, la rândul său, are cauze multiple și simultane: sensibilitatea senzorului, amplificatorul semnalului electric, zgomotul termic, sursa de curent, convertorul analogic-digital, erori de transmisie și stocare digitală etc;
- în cazul camerelor color: modul de obținere a componentelor RGB și conversia în valori grayscale;
- limitările calibrării parametrilor intrinseci și extrinseci;
- re-eșantionarea ce are loc în pasul rectificării (opțional cu desdistorsionare inclusă);
- limitările procesului de stereo matching;
- limitările modului de determinare a părții fracționare, de sub-pixel, a disparității;
- o textură mai slabă determină diferențe mai mici între măsura potrivirii pixelului corect și a celor doi vecini, ducând la o imprecizie mai mare a sub-pixelului disparității.

În literatura de specialitate, se consideră că acuratețea este de ordinul a 0,2 – 0,25 pixeli de disparitate (în [PointGrey12] este 0,22). Pare neplăcut de imprecis, dar se explică prin multitudinea de factori enumerați mai sus.

Acuratețea în domeniul disparităților se poate considera constantă, dar în domeniul profunzimilor metrice (coordonata  $Z$ ) acuratețea e variabilă, după cum se arată în Fig. 3.2.1. Procentual, raportat la distanță, avem acuratețe de 1% la 5 m, 2% la 10 m, 4% la 20 m și 11% la 50 m. La 100 de metri se poate considera că este inacceptabilă, fiind de 28%.

Pentru calculul acurateței metrice, trebuie specificat că s-a folosit un sistem cu  $f = 380$  pixeli,  $b = 320$  mm, imagini de rezoluție 512x383 și punctul principal în centrul imaginii.

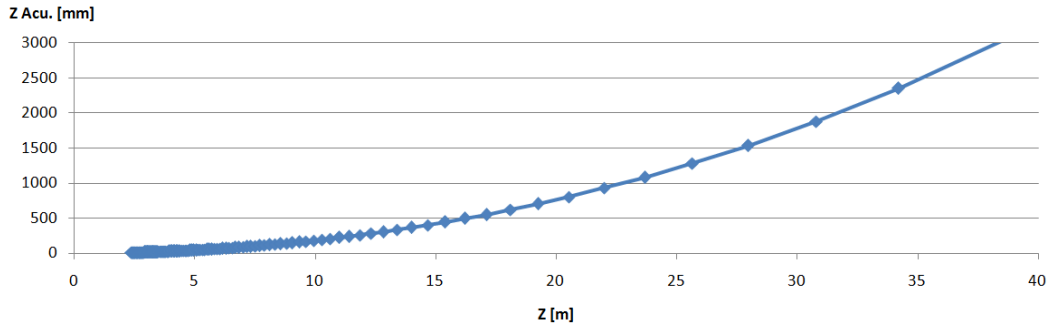


Fig. 3.2.1. Acuratețea metrică

Conform secțiunii „Reconstrucția 3D”, acuratețea coordonatei  $X$  ( $X = (x - PP_x) \frac{Z}{f}$ ) este acuratețea coordonatei  $Z$  înmulțită cu  $\frac{(x - PP_x)}{f}$ . Ea tinde spre zero în apropierea punctului principal, iar spre marginea imaginii are valori de ordinul de mărime al acurateței coordonatei  $Z$ . Acuratețea coordonatei  $Y$  este similară.

### 3.2.2 Erori. Magnitudinea erorilor

După cum s-a văzut, acuratețea reprezintă clasa erorilor cu cele mai mici magnitudini, adică atunci când pixelul corespondent este găsit corect. În continuare vor fi prezentate clase mai mari de magnitudini ale erorilor, atunci când pixelul corespondent este găsit incorect. Desigur, împărțirea în clase are un caracter orientativ și subiectiv, având semnificație din punct de vedere al cauzelor dar și al detecției de obstacole. Cauzele vor fi exemplificate în Fig. 3.2.2 și Fig. 3.2.3.

Magnitudinile (orientative ale) erorilor disparității pot fi împărțite în următoarele patru clase:

- **Mici:** de la 0 la 0,3 pixeli; desigur, aceasta este clasa magnitudinilor acurateței, sau a erorilor inerente la nivel de sub-pixel;
- **Medii:** de la 0,5 la 1,5 pixeli; reprezintă mici alunecări de la disparitatea corectă;
- **Mari:** de la 2 la 5 pixeli; alunecări serioase de la disparitatea corectă, inclusiv prin găsirea incorectă a pixelului corespondent;
- **Foarte mari:** de la 5 pixeli la disparitatea maximă folosită; cauzate de găsirea incorectă a pixelului corespondent.

Cauzele erorilor sunt:

- **Eroarea de sub pixel:** adică acuratețea; factorii săi au fost prezentați pe larg în secțiunea anterioară.

În Fig. 3.2.2 și Fig. 3.2.1, se poate observa că acest tip de erori au magnitudini mici pentru obstacolele din prim-plan, dar devin semnificative la distanțe mai mari, împrăștiind punctele 3D ale mașinii aflate departe pe drum, pe o distanță de 10 metri.

În vederea de sus a scenei, la distanțe mari, se mai poate observa fenomenul unei distribuții a punctelor 3D pe rânduri (Fig. 3.2.2). Aceasta provine din reprezentarea disparităților ca numere reale în virgulă fixă. De exemplu, placa de reconstrucție hardware TYZX ([TYZXhw]) folosește 5 biți pentru partea subunitară, adică 32 de valori discrete. Pentru intervalul 40 .. 65 m, disparitatea are valori între 3 și 1.9 pixeli, rezultând

că, în mare, 25 de metri sunt împărțiți în  $1,1 \cdot 32 = 35$  de valori discrete ale disparității, adică aproximativ 0,7 m între două disparități reprezentabile consecutive. Oricum, partea bună este că această acuratețe a reprezentării disparităților este mult mai fină decât cea a erorilor inerente ale stereo matching-ului (de 8 ori). Situația e similară pentru motorul de reconstrucție software din biblioteca open source oferită de Intel: OpenCV ([OpenCV]), care folosește 4 biți pentru partea subunitară.

#### - **Pixeli la intersecția dintre prim-plan și fundal**

Se poate observa că, la intersecția dintre elementele de prim-plan ale scenei și cele de fundal, apar puncte reconstruite la profunzimi între cele două elemente, sau chiar în afara acestui interval. Explicația este că atât textura din prim-plan cât și cea din fundal contribuie la valoarea metricii folosite la potrivire.

Magnitudinea acestui tip de erori variază de la 0,5 la câțiva pixeli, putând fi încadrate în clasele de erori medii și mari.

#### - **Suprafețe reale înclinate**

Orientarea unei suprafețe este descrisă de cele trei unghiuri: pitch, yaw și roll. Multe combinații ale acestora pot face ca suprafețele proiectate în cele două camere să arate suficient de diferit încât să ducă la scăderea perfecțiunii metricii de potrivire. Orice suprafață, care are pitch nenul sau yaw nenul, suferă un astfel de fenomen. Pentru suprafețe particulare pot fi folosite abordări particulare, cum ar fi cea din [Williamson99] pentru suprafața drumului.

- **Muchii orizontale:** de-a lungul unei muchii orizontale, toți pixelii pot fi înconjurați de aceeași textură, monotonă, astfel că potrivirea este la limită. Magnitudinea poate fi mică, medie sau mare.

- **Textura slabă:** este similar cu tipul muchii orizontale, dar textura este monotonă în toate direcțiile. S-a observat că se încadrează în clasele de magnitudini: medii, mari și foarte mari.

Din fericire, în cele mai multe cazuri, problema apare rar la distanțe mai mari de 15-20 de metri. Peticele de textură slabă, de pe mașini și ziduri, apar mai mici în imagini atunci când sunt la distanțe mai mari. Astfel, vecinătatea metricii de potrivire (care e la fel de mare pentru toți pixelii) tinde să acopere și petice cu textură puternică. Astfel potrivirea devine corectă. Oricum, întotdeauna pot apărea obstacole îndepărtate și mari (inclusiv suprafața drumului), cu petice mari de imagine monotonă (Fig. 3.2.2).

#### - **Reflexii pe suprafețe lucioase/transparente**

Când un obstacol O1 este reflectat de suprafața lucioasă a obstacolului O2, obstacolul O1 este perceput ca fiind la o distanță egală cu suma dintre distanța de la cameră și O1 plus distanța de la O1 la O2. În general, suprafețele vehiculelor nu sunt plate, ci curbate convex, făcând ca O2 să pară și mai departe. În plus textura lui O1 se combină cu textura reflectată a lui O2. Efectele negative asupra procesului de stereo matching sunt evidente. Similar și în cazul suprafețelor transparente, sau care sunt și lucioase și transparente în același timp. Eroarea disparității se încadrează în clasele de magnitudini: medii, mari și foarte mari.



Din fericire, reflexiile obstacolelor pe caroseriile autovehiculelor nu sunt observabile la distanțe mai mari (15m). Totuși, reflexiile Soarelui și ale obstacolelor mari pot persista și până la 40 m, mai ales pe parbrize.

**- Șablon repetitiv**

E specific atunci când procesul de stereo matching găsește mai mulți pixeli candidați având vecinătăți foarte similare. Unele abordări preferă să rejeteze aceste cazuri.

Erorile disparității sunt mari sau foarte mari. Nici nu pot să fie mici sau medii, pentru că șabloanele repetitive nu pot fi amplasate foarte aproape în imagine (deoarece ar fuziona).

**-Lipsa reconstrucției 3D:** chiar dacă nu poate fi considerată o eroare, totuși poate duce la discontinuități ale obstacolelor.

Reamintim că eroarea profunzimii este direct legată de eroarea disparității. Această legătură este exemplificată în Fig. 3.2.4 și Fig. 3.2.5.

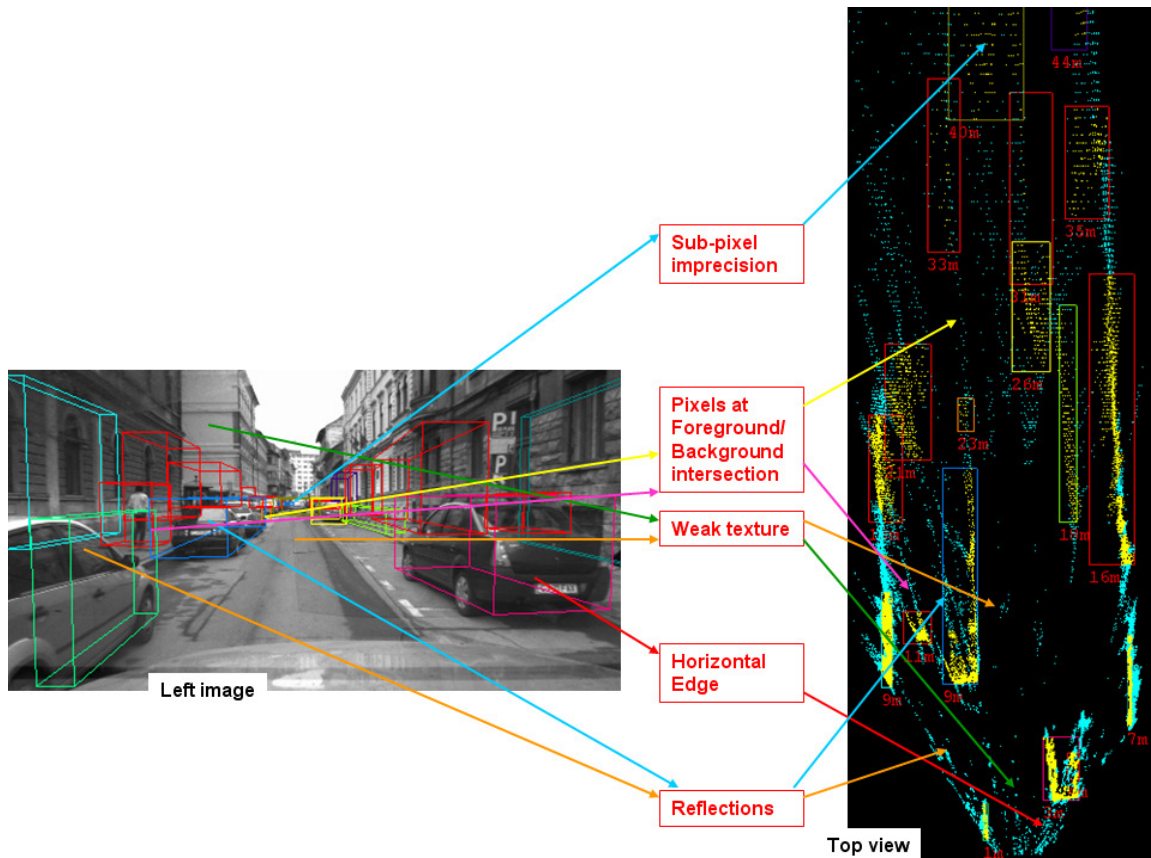


Fig. 3.2.2. Cauzele erorilor (notă: obstacolele roșii au fost clasificate ca obstacole de fundal)

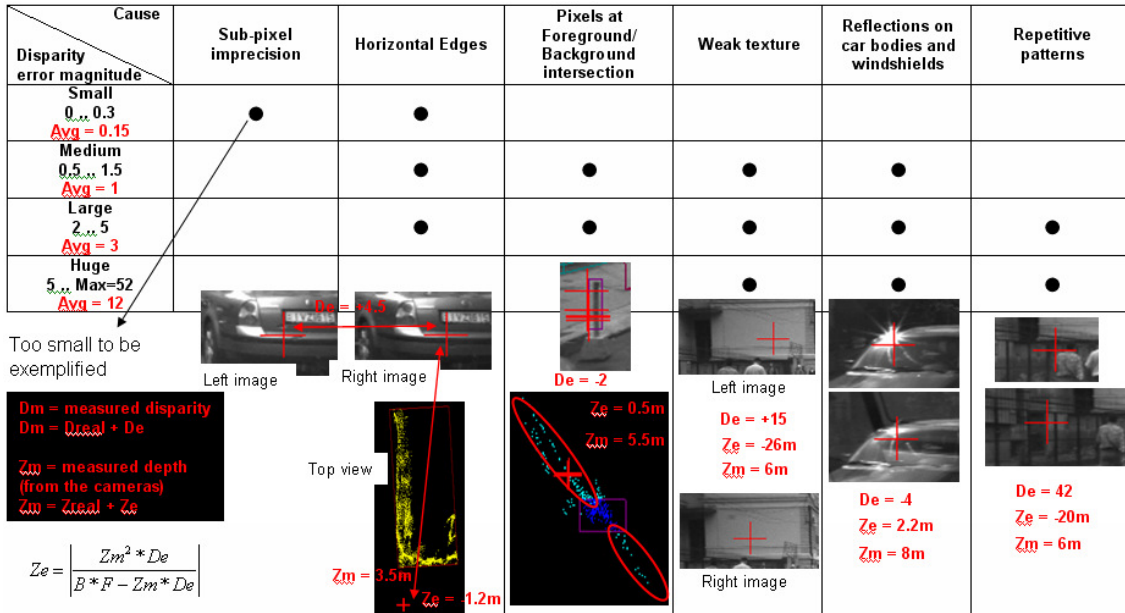


Fig. 3.2.3. Magnitudinile claselor de erori, cu exemple în spațiul imaginii și spațiul top-view

Magnitude type of measured disparity error	Magnitude value of measured disparity error $D_e$ [pixels]	Measured Depth error magnitude [m] (for measured Avg disparity error magnitude)									
		Depth range 2.34 .. 10 m ~7.5 m		Depth range 10 .. 20 m 10 m		Depth range 20 .. 30 m 10 m		Depth range 30 .. 40 m 10 m		Depth range 40 .. 65 m 25 m	
		Disparity range 52 .. 12 (disp. range = 40)		Disparity range 12 .. 6 (disp. range = 6)		Disparity range 6 .. 4 (disp. range = 2)		Disparity range 4 .. 3 (disp. range = 1)		Disparity range 3 .. 1.9 (disp. range = 1.1)	
		+De	-De	+De	-De	+De	-De	+De	-De	+De	-De
Small	0 .. 0.3 Avg = 0.15	0.01 .. 0.12	0.01 .. 0.12	0.12 .. 0.50	0.12 .. 0.48	0.50 .. 1.1	0.48 .. 1.07	1.15 .. 2	1.07 .. 1.8	2 .. 5.6	1.8 .. 4.8
Medium	0.5 .. 1.5 Avg = 1	0.04 .. 0.9	0.04 .. 0.75	0.9 .. 4	0.75 .. 2.8	4 .. 10	2.8 .. 6	10 .. 19	6 .. 10	19 .. 75	10 .. 22
Large	2 .. 5 Avg = 3	0.14 .. 3.2	0.13 .. 2	3.2 .. 19	2 .. 6.5	19 .. 85	6.5 .. 12	85 .. ∞	12 .. 20	Impossible	20 .. 40
Huge	5 .. Max=52 Avg = 12	0.7 .. ∞	0.4 .. 5	Impossible	5 .. 13	Impossible	13 .. 22	Impossible	22 .. 32	Impossible	32 .. 56

$$Z_e = \left| \frac{Z_m^2 * D_e}{B * F - Z_m * D_e} \right|$$

Fig. 3.2.4. Exemple de erori ale disparităților și profunzimilor, pe diferite intervale de profunzime

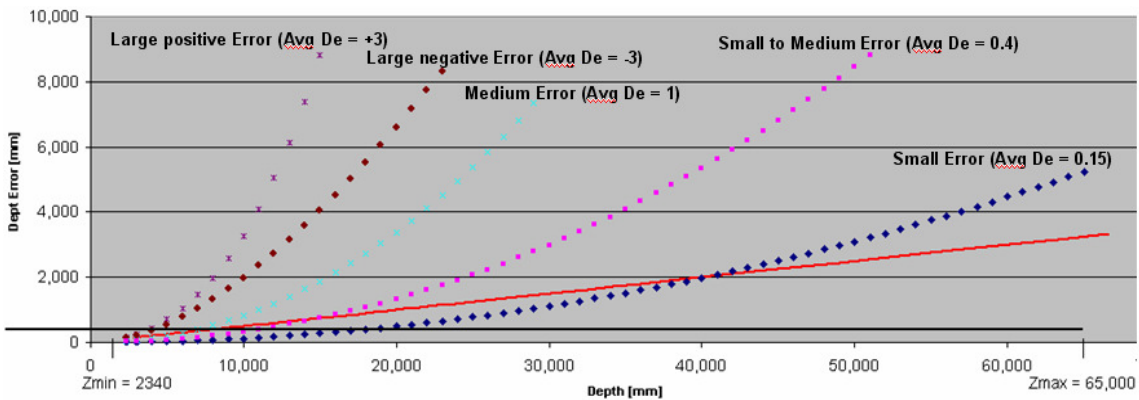


Fig. 3.2.5. Exemple de magnitudinea erorilor distanțelor, pentru câteva erori ale disparității

Din punctul de vedere al detecției de obstacole, concluziile sunt:

- Eroarea distanței metrice, pentru eroare constantă a disparității, crește pătratic cu distanța.
- Efectul erorilor este mai mic asupra obstacolelor mai mari.
- Erorile mici ale disparității nu au un efect semnificativ asupra structurii obstacolelor aflate la profunzime mică (până la 20 m), pentru profunzime medie (20 .. 40 m) apar distorsiuni tot mai mari, iar la profunzimi mai mari, erorile metrice tind să fie comparabile cu lungimea obstacolelor reale.
- Erorile medii și mici (în jur de 0,5 pixeli) nu au un efect semnificativ până la 15 m.
- Erorile medii nu au un efect semnificativ până la 8 m.
- Pentru erori medii, mari și foarte mari, punctele 3D pot să apară în afara obstacolelor, partea bună fiind că nu mai afectează structura.
- Adesea, un petic de imagine are propria semnificație 2D și 3D și propriul tip de eroare, adică grupuri de pixeli vecini pot fi afectați de același tip de eroare.

**Observatie:** Fiecare pixel poate fi afectat de o combinație de tipuri de erori, unul sau două fiind predominante.

### 3.2.3 Modelarea acurateții metrice

Studierea acurateții metrice prin **modele** matematice oferă o analiză științifică și elegantă a modului în care domeniul problemei – scena reală – este perceput în domeniul soluției – scena măsurată prin stereo viziune.

#### Modelul nativ

În mod nativ, stereo viziunea măsoară profunzimea metrică prin intermediul disparităților:

$$Z = \frac{b \cdot f}{D} \quad (1)$$

în care  $b$  și  $f$  sunt constante, putând nota  $b \cdot f = C \Rightarrow Z \cdot D = C$  (2)

Scriem  $D$  ca  $D_m = D_r + D_e$  (3)

cu semnificațiile:

- $D_m$  – disparitatea măsurată,
- $D_r$  – disparitatea reală,
- $D_e$  – eroarea disparității.

Aici, eroarea disparității este acuratețea, dar din considerente de generalizare, prin eroare, se pot înțelege toate clasele de eroare (inclusiv acuratețea).

În mod similar, putem scrie și  $Z_m = Z_r + Z_e$ , cu aceleași semnificații. Astfel, eroarea metrică este

$$Z_e = Z_m - Z_r = \frac{C}{D_m} - \frac{C}{D_m - D_e} = C \frac{-D_e}{D_m(D_m - D_e)} = -\frac{C \cdot D_e}{D_m(D_m - D_e)} \quad (4)$$

Pentru a avea o măsură a erorii distanței metrice în funcție de însăși distanța metrică, putem duce mai departe relația anterioară:

$$Z_e = -\frac{C \cdot D_e}{\frac{C}{Z_m} (\frac{C}{Z_m} - D_e)} = -\frac{D_e}{\frac{1}{Z_m} \frac{C - D_e Z_m}{Z_m}} = -\frac{D_e Z_m^2}{C - D_e Z_m} \Rightarrow Z_e = -\frac{D_e Z_m^2}{C - D_e Z_m} \quad (5)$$

Ordonatele din Fig. 3.2.1 au fost generate folosind această formulă. În Fig. 3.2.5 sunt prezentate graficele erorilor metrice, pentru câte o eroare a disparităţii din fiecare clasă de magnitudini de erori.

Este interesant de discutat semnul lui  $Z_e$  din formula (5). În practică, se disting două cazuri:

1)  $D_e > 0$ , adică s-a găsit o disparitate mai mare decât cea reală: din (2) rezultă  $C = D_m \cdot Z_m$ , rezultă că numitorul  $C - D_e Z_m = D_m Z_m - D_e Z_m = (D_m - D_e) Z_m$

Avem două subcazuri:

11)  $D_m > D_e \Rightarrow$  numitorul e pozitiv, deci  $Z_e$  din (5) e negativ. Aceasta era de aşteptat: disparitatea mai mare decât cea reală ( $D_e > 0$ ) produce o distanţă mai mică decât cea reală ( $Z_e < 0$ ).

12)  $D_m < D_e$ , ar rezulta că  $D_m - D_e < 0$ . Din (3) ar rezulta că disparitatea reală ( $D_r$ ) este negativă, ceea ce e imposibil (pentru că s-a considerat un sistem stereo canonic).

2)  $D_e < 0$ , adică s-a găsit o disparitate mai mică decât cea reală: numitorul din (5) va fi pozitiv, deci  $Z_e > 0$ . Şi aceasta era de aşteptat: disparitatea mai mică decât cea reală ( $D_e < 0$ ) produce o distanţă mai mare decât cea reală ( $Z_e > 0$ ).

### Maniera creşterii erorii distanţei metrice

Se pune întrebarea: în ce manieră creşte eroarea distanţei metrice, în funcţie de însăşi distanţa metrică? În (5), numărătorul e o funcţie pătratică şi numitorul e o funcţie afină. La numitor, ar fi util de analizat ce relaţie este între  $C$  şi  $D_e \cdot Z_m$ .  $C = D_m \cdot Z_m$ , rezultând  $C - D_e \cdot Z_m = D_m \cdot Z_m - D_e \cdot Z_m = (D_m - D_e) \cdot Z_m$ .

Distingem trei cazuri:

1) pentru  $D_m$  „suficient” de mare (de exemplu  $D_m > 10 \cdot D_e = 2,5$  pixeli), termenul de grad 1 ( $D_e \cdot Z_m$ ) devine nesemnificativ în comparaţie cu termenul de grad 0 ( $D_m \cdot Z_m = C$ ), deci putem

face aproximaţia  $Z_e = -\frac{D_e Z_m^2}{(D_m - D_e) Z_m} \approx -\frac{D_e Z_m^2}{D_m Z_m} = -\frac{D_e Z_m^2}{C}$  adică eroarea metrică creşte într-o

**manieră pătratică.**

2)  $D_m$  e relativ mic, tinde spre  $D_e$ , dar nu mai mic decât  $D_e$ :  $D_m$  şi  $D_e$  devin comparabile, deci termenii de grad 0 şi 1 ( $D_m Z_m = C$  şi  $D_e Z_m$ ) devin comparabili, gradul numitorului crescând de la 0 spre 0,5. Şi cum numărătorul are gradul 2, întreg raportul tinde să scadă de la gradul 2 spre gradul 1,5, adică eroarea metrică creşte într-o manieră de tip “între pătratic şi liniar”.

3) pentru disparităţi măsurate ( $D_m$ ) mai mici decât acurateţea ( $D_e$ ), aceste ecuaţii nu mai sunt valabile, zgomotul fiind prea mare; incertitudinea măsurătorilor (metric vorbind) e atât de mare încât toate punctele aflate la disparităţi mai mici decât  $D_e$  pot fi considerate ca fiind la aceeaşi distanţă, de exemplu  $Z = C/D_e$  sau  $Z = +\infty$  (cu semnificaţia „puncte aflate la distanţă prea mare pentru a fi măsurabilă”). În practică, e de preferat să nu se dea încredere distanţei

metrice dincolo de  $Z = \frac{C}{k \cdot D_e}$ , unde  $k = 2$  sau  $k = 3$ .

Per global, se poate spune că eroarea metrică crește într-o manieră cvazi-pătratică cu profunzimea.

### Modelul liniar (contribuție originală)

Se știe că o funcție pătratică, pentru valori mici ale variabilei, are o creștere care poate fi considerată liniară. Ne propunem să vedem care este diferența dintre modelul nativ al erorii și un model liniar (exemplificat printr-o dreaptă de culoare roșie în Fig. 3.2.5):

$$\hat{Z}_e = a \cdot Z_m, \quad a > 0.$$

$$\text{Diferența, în funcție de distanță, este } H(Z_m) = |\hat{Z}_e| - |Z_e| = aZ_m - \frac{D_e Z_m^2}{C - D_e Z_m}.$$

Analiza unor astfel de funcții, folosind metode de analiză matematică, conduce la ecuații de grad mare, pentru care nu există formule ale soluțiilor (grad mai mare decât 3) și nici nu se încadrează în clase particulare de ecuații. De aceea, s-a apelat la metode numerice și iterative. Astfel, pe intervalul 3 – 50 m, prin încercarea a diferite valori pentru  $a$ , s-au observat valorile lui  $H$ . De exemplu, pentru  $a = 0,0887$ ,  $H$  are graficul:

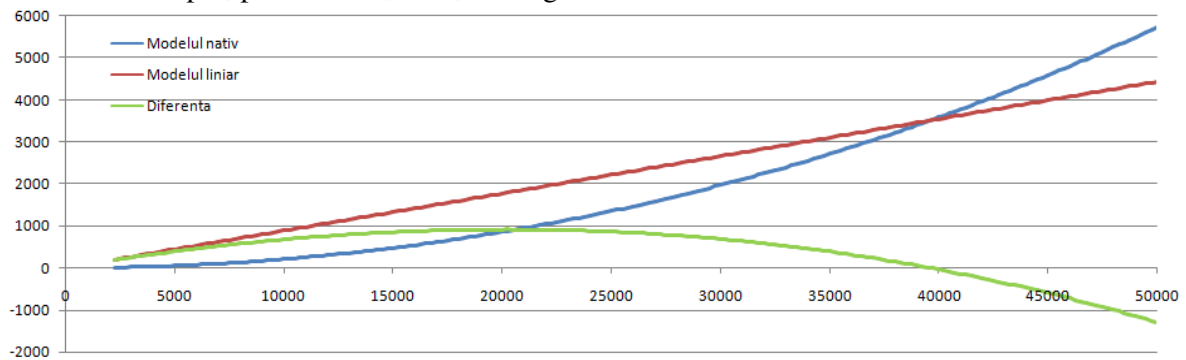


Fig. 3.2.6. Modelul nativ ( $Z_e$ ), modelul liniar ( $\hat{Z}_e$ ) și diferența dintre ele ( $H$ ).

Valoarea lui  $a$  a fost aleasă empiric, astfel încât să ofere un echilibru între subestimarea acurateții native, în prima parte a intervalului și supraestimarea acurateții native, în a doua parte a intervalului; s-a ținut seama și de valorile diferențelor absolute și de diferențele relative dintre cele două modele.

Astfel, maximul de 0,9 m se obține la distanța de 20 m. Conform acestui model, la distanța de 20 m nu se pot distinge obstacole aflate la 1-2 m între ele (pe profunzime). În același timp, erori de 0,9 m vor fi ignorate. În jurul distanței de 40 m, semnificația este similară cu cea a modelului nativ. La 50 m, modelul liniar va încerca să supraestimeze acuratețea, coborând-o sub nivelul corespunzător modelului nativ.

În Fig. 3.2.7, modelul liniar este comparat cu modelul nativ pentru diferite clase de magnitudini ale erorilor, arătând că:

- Acoperă acuratețea (erori mici) până la 40 m;
- Acoperă erori medii până la 15 m;
- Acoperă erori mari până la 7 m;
- Acoperă erori foarte mari (pozitive și negative) până la 2 m, interval în care nu se află obstacole;

- Linia orizontală groasă și neagră reprezintă distanța minimă dorită de separare între obstacole (0,5 m). Modelul liniar respectă această distanță minimă până la 8 m. Dincolo de 8 m, se consideră că distanța minimă dorită de separare nu mai poate fi respectată, din cauza erorilor de reconstrucție; aceasta din punctul de vedere al modelului liniar al erorilor, însă se observă că modelul nativ merge până la 18 m.

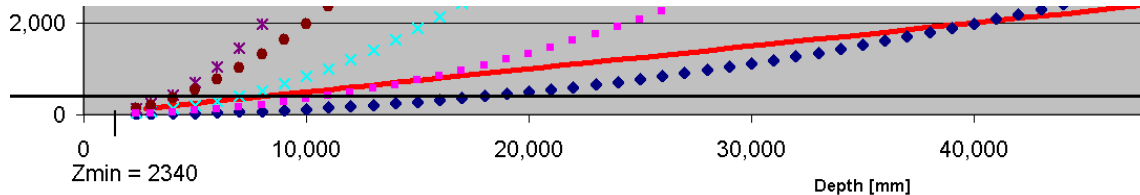


Fig. 3.2.7. O vedere mai îndeaproape a modelului liniar (dreapta roșie) din Fig. 3.2.5

Dezavantajele modelului liniar sunt:

- Pentru alegerea și evaluarea sa, e necesară stabilirea unui interval de distanțe în care se va face detecția de obstacole (sau alte procesări);
- Se îndepărtează relativ mult de modelul nativ, excepție făcând distanțele foarte mici (care oricum nu sunt folosite pentru că între camere și botul ego-car nu sunt obstacole) și un interval destul de scurt în jurul distanței unde cele două modele se intersectează (după cum s-a văzut mai sus);
- Tratează într-un mod neuniform intervalul de distanțe folosit;
- Necesită o alegere atentă a pantei ( $a$ ) și, în plus, e vorba de o alegere empirică.

Avantajele modelului liniar sunt:

- Este cel mai simplu model;
- În prima parte a intervalului, poate ignora/îngloba erori relativ mari, cauzate de reflexii, transparențe, muchii orizontale sau imperfecțiuni ale calibrării camerelor.

Per global, modelul liniar are mai multe dezavantaje decât avantaje, dar studiul său a dus la o mai bună înțelegere a erorilor de reconstrucție, mai ales prin meditarea legată de echilibrul avantajelor contra dezavantajelor.

\*\*\*

Modelul liniar este folosit în [Pocol07], creându-se un spațiu top-view comprimat, tip matrice, în care fiecare linie a matricii corespunde unui interval pe axa distanțelor. Intervalele succesive au lungimi tot mai mari, crescând liniar cu distanța, folosind așadar modelul liniar al erorilor de reconstrucție.

Astfel, intervalul metric corespunzător primului rând al spațiului comprimat este:

$$[Z_0 .. Z_0 + IntervalLength(Z_0)] \text{ sau } [Z_0 .. Z_1]$$

unde  $Z_0 = Z_{min}$  este ales convenabil pentru detecția de obstacole, și  $IntervalLength(Z) = a * Z$  este lungimea intervalului care începe la distanța  $Z$ . Astfel

$$Z_0 + IntervalLength(Z_0) = Z_0 * (1 + a) = Z_1.$$

Intervalul metric corespunzător celui de al doilea rând al spațiului comprimat este:

$$[Z_1 .. Z_1 + IntervalLength(Z_1)], \text{ sau } [Z_1 .. Z_2]$$

unde  $Z_1 = Z_0 + IntervalLength(Z_0)$ , adică al doilea interval începe unde se termină primul:

$$Z_1 = Z_0 * (1 + a) \text{ și}$$

$$Z_2 = Z_1 + IntervalLength(Z_1) = Z_1*(1+a) = (Z_0 + IntervalLength(Z_0))*(1+a)^2 = Z_0*(1+a)*(1+a) = Z_0*(1+a)^2$$

Sfârșitul celui de-al doilea interval este începutul celui de-al treilea interval:

$$Z_2 = Z_0*(1+a)^2$$

Intervalul metric corespunzător celui de-al  $n$ -lea rând din spațiul comprimat este:

$$[Z_n .. Z_n + IntervalLength(Z_n)], \text{ unde}$$

$$Z_n = Z_0*(1+a)^n$$

Pentru un punct 3D dat, intervalul corespunzător este  $[Z_i .. Z_i + IntervalLength(Z_i)]$  sau  $[Z_i .. Z_{i+1}]$ . Din formula

$$Z = Z_0*(1+a)^i,$$

se determină formula inversă și se consideră partea întreagă a rezultatului:

$$Row = i = \lceil \log_{1+a} \frac{Z}{Z_0} \rceil$$

În practică, se precalculează tabele, pentru a evita astfel de calcule costisitoare.

\*\*\*

O comparație între intervalele magnitudinilor acurateții, pentru cele două modele, este prezentată în Fig. 3.2.8. Gridul verde împarte spațiul metric top-view în pătrate cu latura de 10 metri. Axa Z este desenată cu roșu. Se poate observa că acuratețea nativă este foarte bună la distanțe mici, scăzând pătratic în prima parte și “între pătratic și liniar” în a doua parte, după cum s-a demonstrat mai sus. De asemenea, s-a demonstrat, și se poate observa și aici, că modelul liniar subestimează modelul nativ în prima parte și îl supraestimează în a doua parte.

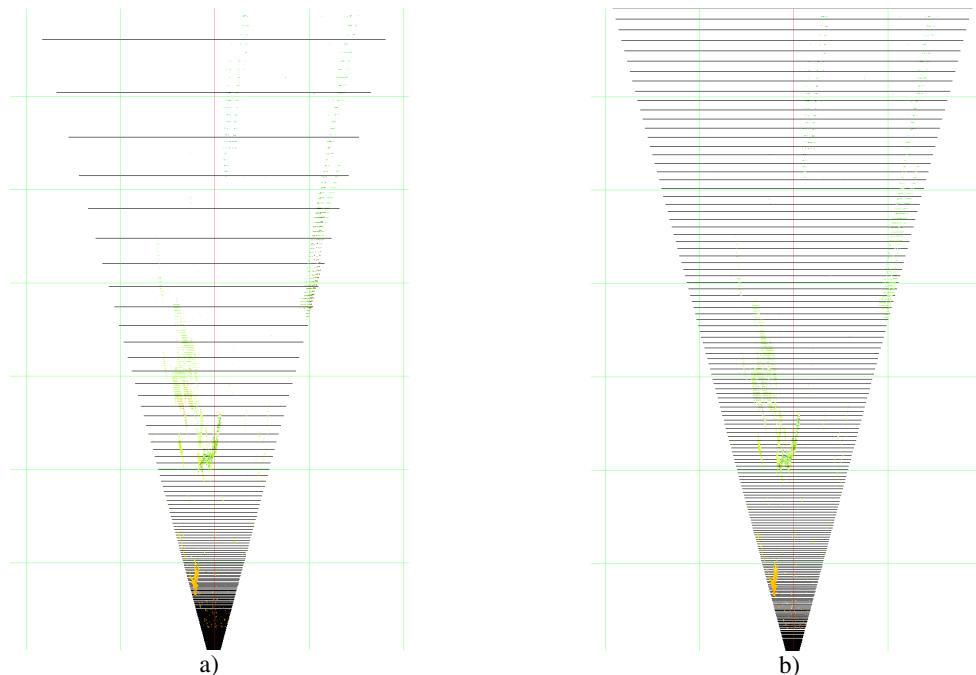


Fig. 3.2.8. Magnitudinile acurateții metrică pentru modelul nativ (a) și pentru modelul liniar (b)

## Modelul probabilistic

În [Thrun01] eroarea măsurătorii unui senzor de profunzime se definește într-un mod probabilistic, în ipoteza în care punctele măsurate aparțin unei suprafețe plane. Planul  $\theta$  este descris prin:

$$\theta = \{\alpha, \beta\}$$

unde  $\alpha$  este un grup de trei parametri care descriu normala la plan, iar  $\beta$  este offset-ul față de originea sistemului de coordonate.

Distanța unei măsurători  $z_i$  la planul  $\theta$  este una euclidiană și se notează cu  $d(z_i, \theta)$ , fiind ușor de folosit formule standard pentru distanța de la un punct la un plan.

Presupunând o eroare gaussiană, se definește *modelul măsurătorii* ca:

$$p(z_i | \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{d^2(z_i, \theta)}{2\sigma^2}}$$

Distribuția erorii e dată de un gaussian cu varianța  $\sigma$ . În acest mod se poate determina simultan și suprafața  $\theta$ .

Mai mult, lucrarea citată prezintă o modelare extinsă, care determină structura scenei printr-un set de suprafețe plane. În acest scop, modelul unei suprafețe este extins prin parametri care descriu un poligon rectangular conținut în suprafața (infinită a) planului.



### 3.3. Cantitatea și distribuția punctelor 3D

#### 3.3.1 Cantitatea

Indiferent de structura scenei, numărul de puncte 3D este limitat la numărul de pixeli din imagine. În practică, nu toate punctele pot fi reconstruite, adică nu se pot stabili disparități clare pentru toți pixelii, din două mari motive:

- ocluziile stereo: pixeli vizibili în imaginea stângă și nevizibili în cea dreaptă;
- incertitudinea potrivirii: atunci când mai mulți pixeli candidați din imaginea dreaptă se potrivesc bine cu pixelul curent din imaginea stângă și nu sunt disponibile raționamente de dezambiguare.

În general, 50-90% din pixeli sunt reconstruiți 3D.

În cazul reconstrucției rare (sparse), cantitatea punctelor 3D este limitată la numărul trăsăturilor alese. Este în special cazul sistemelor cu resurse de calcul limitate.

#### 3.3.2 Distribuția

Dacă în spațiul imaginii, punctele 3D pot fi distribuite continuu și uniform, în spațiul 3D acestea au o distribuție de o natură mult diferită, având în vedere următoarele fenomene, adesea simultane:

- 1) Distribuția laterală și pe înălțime (Fig. 3.3.1) este direct influențată de împărțirea spațiului 3D în mod polar, prin însăși geometria proiectivă a matricii de pixeli ai camerei; fiind tot mai răsfirate pe lățimea și înălțimea spațiului 3D, cu cât profunzimea este mai mare.
- 2) Distribuția longitudinală (pe profunzime), pentru o structură planară orientată spre cameră (Fig. 3.3.1), este dată de erorile de reconstrucție, inclusiv cele mici, ale acurateței (și chiar până la precizia de reprezentare numerică a disparității –după cum s-a discutat anterior).

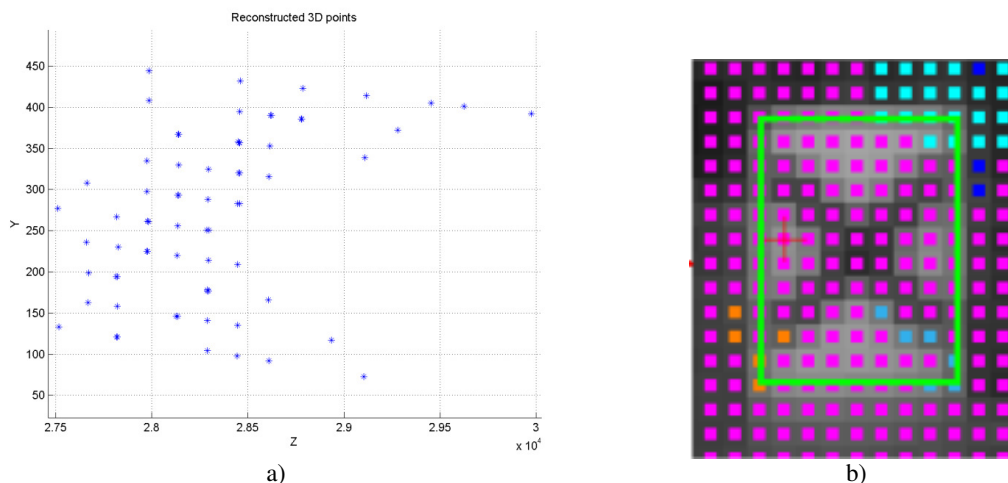


Fig. 3.3.1. Distribuția punctelor 3D pe înălțime și profunzime (a) a punctelor de pe o suprafață plană – doar cele din dreptunghiul verde (b) (culorile punctelor nu au semnificație aici).

Distribuția longitudinală, pentru o suprafață plană orientată spre cameră, este folosită pentru evaluarea a diferite sisteme de reconstrucție. Singurele tipuri de erori sunt aici acuratețea și muchiile orizontale.

Pentru surprinderea primelor două fenomene, s-au propus abordări care folosesc spații polare pe lățime[Pocol07]/înălțime[Labayrade02] și comprimate special pe profunzime: [Labayrade02, Pocol07, Pocol15, Badino07].

- 3) Înclinarea unei suprafețe reale (cu unghi pitch nenul sau yaw nenul) duce la micșorarea proiecției în spațiul imaginii, deci la puncte mai puține și mai distanțate între ele (datorită înclinării înseși). În plus, reconstrucția are erori mai mari, după cum s-a văzut în secțiunea care analizează magnitudinea și sursele erorilor. Acest fenomen este cel mai puțin tangibil și tratabil, pentru că depinde strict de structura scenei, putând fi doar măsurat prin stereoviziune, dar acuratețea scăzută la distanțe mai mari este prohibitivă. Pot fi aplicate doar tratări particulare pentru structuri particulare, cum ar fi suprafața drumului, care are înclinații mai mari la distanțe mai mari, relativ la camera, și unde se poate face o compensare a densității punctelor, precum în [Oniga10].
- 4) Proiecția perspectivă, care face ca obstacolele mai îndepărtate să apară mai mici. Acest fenomen este abordat în [Pocol15] prin normalizarea densității punctelor 3D în spațiul comprimat orizontal.
- 5) Ocluziile cauzează conuri de „umbră”, în spatele obstacolelor nemaifiind nimic vizibil (Fig. 3.1.7).

\*\*\*

Comparativ, stereo reconstrucția densă aduce oportunități noi față de cea rară:

- În interiorul obstacolelor sunt mai multe puncte 3D, mai apropiate între ele, punând în evidență consistența obstacolelor și a legăturii dintre sub-părțile lor.
- În exteriorul obstacolelor, distanțele dintre acestea, chiar și dintre cele mai apropiate, este considerabil mai mare decât distanțele dintre punctele 3D din interiorul obstacolelor, făcând astfel mai facilă detecția de obstacole.
- Frontiera obstacolelor este mai bine conturată, deschizând de exemplu oportunitatea detecției orientării obstacolelor (Fig. 3.3.2).

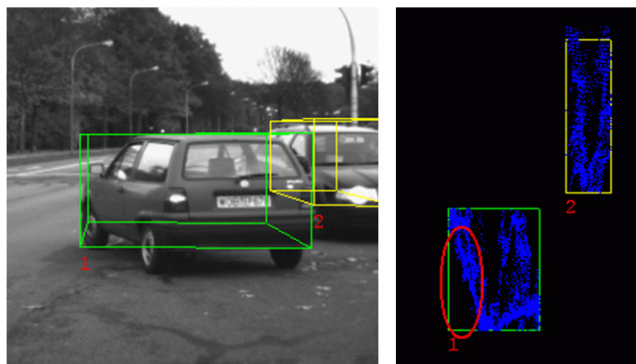


Fig. 3.3.2. Obstacole bogat reconstruite; frontiere bine conturate.

### **3.4. Concluzii, contribuții**

Au fost experimentați și analizați toți pașii necesari stereo viziunii: înțelegerea parametrilor geometrici ai camerelor, calibrarea camerelor, rectificarea imaginilor, alegerea trăsăturilor, stereo matching și reconstrucția 3D. S-a demonstrat că, în ciuda cantității mari de calcule, stereo matching-ul rulează în timp real, pe CPU.

Asupra stereo reconstrucției s-a făcut o analiză originală legată de aspecte precum: acuratețea nativă și modelarea acesteia, erori și clase de magnitudini ale erorilor, cantitatea și distribuția punctelor 3D. Tot aici, s-a pus accentul pe rezoluția cu care senzorul de stereo viziune percepe lumea reală. S-a arătat că pe direcțiile orizontală și verticală, percepția este una polară și că în spațiul cartezian rezoluția scade liniar cu profunzimea. Pe direcția profunzimii, percepția se face prin disparități, rezultând în spațiul cartezian o rezoluție care scade cvazi-pătratic cu profunzimea.

Astfel, una din contribuțiile principale ale tezei este evidențierea și accentuarea în mod repetat a posibilităților de percepție a scenei prin stereo viziune.

## Capitolul 4. Modelarea obstacolelor

În lumea reală, obstacolele pot avea cele mai diverse forme și proprietăți. Un scenariu tipic de trafic conține obstacole din diferite categorii:

- vehicule, inclusiv motocicletele și biciclete;
- pietoni;
- clădiri;
- stâlpi, uneori având indicatoare rutiere la înălțime mică;
- stâlpișori;
- trunchiuri de copaci;
- borduri înalte;
- senzori giratorii;
- bare de protecție;
- tufe;
- puncte de colectare a deșeurilor;
- ridicături semnificative ale solului.

Stereo-viziunea oferă puncte de vedere și perspective particulare ale acestor obstacole, iar formele 3D sunt măsurate cu acuratețe limitată, în special pe axa profunzimii.

Algoritmii de detecție și de procesare a obstacolelor au nevoie de reprezentări numerice ale formelor obstacolelor. Aceste reprezentări trebuie să îndeplinească proprietăți precum:

- să fie ușor de procesat;
- să fie schematice;
- să surprindă cât mai fidel forma obstacolelor reale;
- să fie optime pentru cât mai multe obstacole, adică să aibă un caracter *generic*.

În terminologia de specialitate, aceste reprezentări se numesc *modele*. În lipsa unor modele, ar fi aproape de neimaginat modul în care s-ar putea descrie un obstacol. Un set de puncte 3D neordonat sau ordonat în vreun fel? Porțiunea de imagine în care apare obstacolul? Toate acestea ar oferi informații sărăcăcioase și insuficiente.

Modelele pot furniza cu ușurință informații de nivel înalt despre obstacole: dimensiuni, orientare, formă, axa de alungire poate sugera direcția de mișcare. De asemenea, analiza modelelor unui grup de obstacole poate evidenția relațiile dintre acestea: de exemplu, mai multe obstacole cvazi-liniare și apropiate între ele ar putea sugera existența unui șir de clădiri sau de mașini parcate. Totodată, se poate evidenția relația dintre ego-car și obstacole, cum ar fi un posibil impact, fie cu un obstacol static, fie cu unul dinamic.

### 4.1. Tipuri de modele și sisteme de coordonate

În general, modelele pot fi 2D sau 3D și se bazează pe anumite forme geometrice: dreptunghi, linie poligonală, paralelipiped, poliedru etc. Formele curbe nu se prea folosesc, fiind greu de potrivit pe obstacolele reale, și fiindcă inacuratețea reconstrucției nu prea permite aprecierea corectă a formelor curbe. Totuși, ar fi un caz particular: obstacolele mari și rotunde, cum sunt senzurile giratorii.

În spațiul 2D, linia poligonală oferă o flexibilitate mare, putând fi destul de fidelă obstacolului real. Ea poate fi deschisă sau închisă (rezultând un poligon). Un caz particular este dreptunghiul, putând fi paralel cu axele de coordonate sau având și un unghi de rotație (orientare). În ciuda flexibilității sale, linia poligonală poate fluctua mult în imagini succesive, pentru același obstacol; în schimb dreptunghiul e mai constant, chiar dacă mai puțin fidel obstacolului real. În practică, pentru obstacolele aflate în mișcare, este plauzibil să se folosească modele dreptunghiulare, potrivitându-se atât pietonilor cât și vehiculelor, cu excepția celor mari și articulate (cum este capul tractor al unui tir față de compartimentul pentru marfă). Pentru obstacolele statice, în spațiul top-view, o analiză a frontierei poligonale ar putea decide care model e mai potrivit: linie poligonală sau dreptunghi.

Spațiile 2D cele mai folosite sunt:

- spațiul imaginii: este spațiul nativ al camerei video; obstacolele sunt “văzute” printr-o proiecție perspectivă, dimensiunile și detaliile depinzând de profunzime, liniar cu profunzimea; din punctul de vedere al surprinderii profunzimii scenei, este un spațiu polar.
- spațiul polar top-view: după cum îi spune și numele, este o vedere de sus a scenei; împarte scena de la stânga la dreapta în fâșii polare, similar cu modul de formare al imaginilor în camerele video; împarte scena, de la profunzime mică la profunzime mare, în intervale egale sau variabile, de exemplu în intervale care modelează acuratețea măsurării profunzimii de către camerele stereo [Pocol08].
- spațiul cartezian top-view: oferă și acesta o vedere de sus a scenei, neținând seama de principiile procesului de percepere a scenei prin camera video, ci fiind legat de spațiul cartezian 3D al scenei și respectiv al obstacolelor din aceasta.

În spațiul 3D, suprafețele poliedrale, închise sau deschise, oferă flexibilitate mare, putând fi destul de fidele obstacolelor reale. Un caz particular este paralelipipedul, numit și *cuboid*, orientat sau neorientat față de sistemul de coordonate. Similar cu cazul liniei poligonale, și suprafața poliedrală poate varia mult pentru același obstacol detectat în cadre succesive. În măsura în care există indicii suficiente, se poate alege în mod automat dacă un obstacol se pretează a fi modelat printr-un cuboid sau printr-un mesh poliedral.

Alegerea originii și orientării axelor de coordonate ale spațiului 3D rămâne la latitudinea aplicațiilor concrete. Câteva posibilități sunt:

- originea în camera stânga: este cel mai apropiat de formarea imaginilor;
- originea pe sol sub centrul botului ego-car: are avantajul poziționării în spațiul scenei rutiere;
- originea pe sol sub camera stângă: este un compromis între primele două, valorile profunzimilor fiind egale;
- originea pe sol sub centrul osiei spate a ego-car: prezintă avantaje din punctul de vedere al cinematicii, fiind tangent la cercul mișcării circulare a ego-car;
- axa Ox de la stânga la dreapta: este în concordanță cu axa Ox a imaginii;
- axa Ox dinspre aproape înspre departe: este sugestivă pentru modelarea mișcării longitudinale a ego-car;
- axa Oy de sus în jos: este în concordanță cu axa Oy a imaginii;
- axa Oy de jos în sus: descrie în mod firesc elevația elementelor din scenă;
- axa Oz este de regulă destinată exprimării profunzimii: dinspre aproape înspre departe.

## 4.2. Modele folosite în practică

În practică, se folosesc diferite modele de obstacole, adesea fiind modele mixte, pornind de la modele mai simple. Considerentele care conduc la alegerea unui model concret sunt dintre cele mai diverse:

- modul nativ în care lucrează senzorul;
- natura erorilor;
- particularitățile algoritmilor de procesare;
- considerente legate de afișarea rezultatelor, inclusiv a celor intermediare;
- necesități de depanare;
- schimbul de informații cu alte sisteme;
- necesitatea unor reprezentări independente sau standardizate;
- compatibilitatea cu alte modele;
- considerente legate de timpii de procesare sau de cantitatea de memorie necesară;
- nu în ultimul rând, depinde de inspirația proiectanților și dezvoltatorilor.

Văzând toate aceste considerente, este lesne de înțeles că o aplicație concretă poate necesita folosirea simultană a mai multor modele pentru un singur obstacol. Adesea, atunci când unul dintre modele este modificat, este necesar ca și celelalte modele să fie ajustate corespunzător.

În continuare vor fi prezentate câteva modele folosite frecvent în domeniu, de către diferiți autori. Majoritatea au fost implementate și analizate și în cadrul tezei de față, mai mult ca urmare a necesității decât a inspirației din literatură.

### 4.2.1 Modelul dreptunghiular neorientat

Un obstacol (Fig. Fig. 4.2.1.a) văzut de sus poate fi încadrat într-un dreptunghi paralel cu axele sistemului de coordonate ( $xOz$ ), deci neorientat (Fig. 4.2.1.b) [Nedevschi05]. Se observa că laturile dreptunghiului reprezintă limitele obstacolului pe fiecare axă. Avantajul este că, dându-se mulțimea de puncte 3D a obstacolului, determinarea dreptunghiului se rezumă la o problemă de minim și maxim pe fiecare axă. Se pretează la scenarii de autostradă, unde marea majoritate a obstacolelor sunt cvazi-paralele cu ego-car. Se mai folosește și modelul dreptunghiular 2D (neorientat) în spațiul imaginii, pentru a delimita zona în care apare obstacolul în spațiul senzorului video.

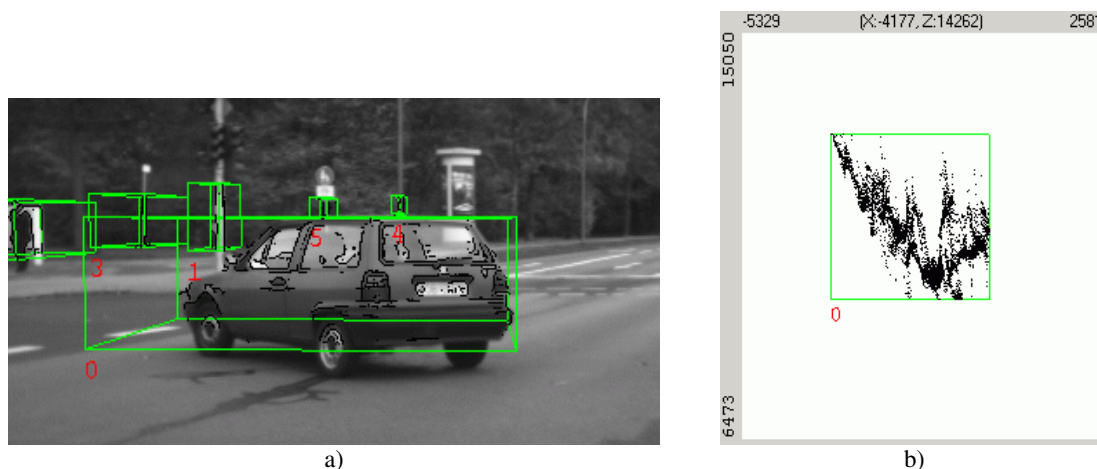


Fig. 4.2.1. Obstacolul real (a) și modelarea sa printr-un dreptunghi neorientat.

Dezavantajul este că nu modelează fidel obstacolele orientate, incluzând și spațiu liber. Un exemplu mai pregnant este prezentat în Fig. 4.2.2. Un dreptunghi care înscrie un obstacol lung, îngust și cu o orientare mare față de axele sistemului de coordonate, va îngloba un spațiu mult mai mare decât obstacolul real, spațiu care poate fi liber sau chiar poate conține alte obstacole.

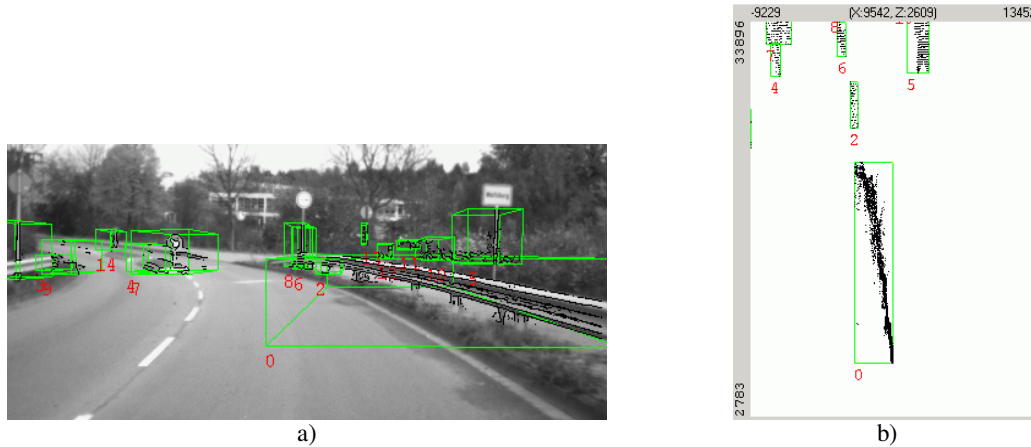


Fig. 4.2.2. Includerea spațiului liber în interiorul dreptunghiului neorientat.

#### 4.2.2 Modelul dreptunghiular orientat

Algoritmii care determină orientarea obstacolelor pot aduce un plus de fidelitate a modelului față de obstacolele reale, folosind modelul dreptunghiular orientat (Fig. 4.2.3) [Pocol08]. Orientarea are sens mai mult în jurul axei verticale (orientare în plan orizontal), ajutând și faptul că baza celor mai multe obstacole este de formă dreptunghiulară. Orientările în jurul celorlalte două axe sunt îngreunate din cauza formelor variate ale obstacolelor (vehicule în particular); totodată aceste orientări sunt adesea nule față de ego-car, fiind amplasate pe aceeași suprafață plană cu ego-car.

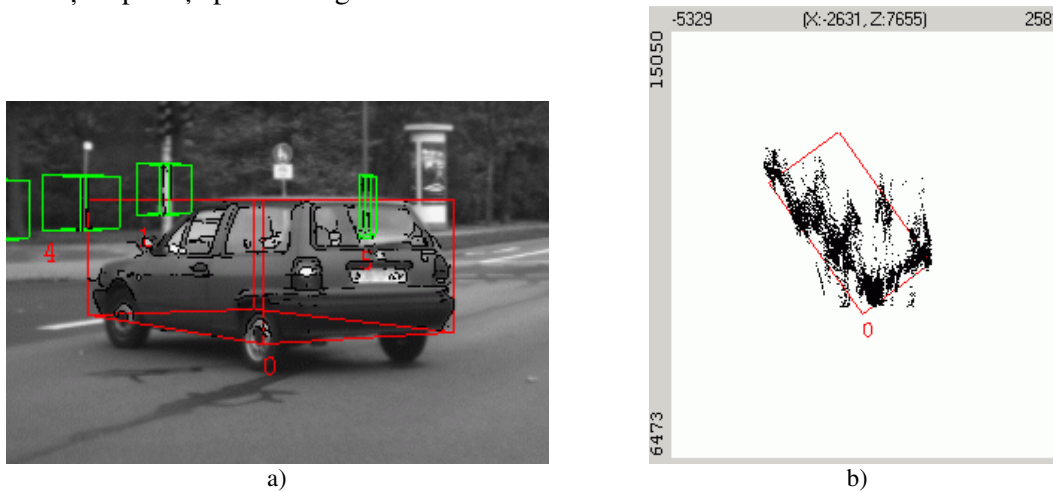


Fig. 4.2.3. Modelul dreptunghiular orientat.

### 4.2.3 Modelul octogonal

Se poate întâmpla, și nu rareori, ca forma top-view a unor obstacole reale să nu fie potrivită pentru modelul dreptunghiular, fie acesta și orientat. Cu alte cuvinte, nu poate fi găsită o linie poligonală, sub formă de dreptunghi, care să înscrie suficient de fidel obstacolul. Astfel, următorul pas propus este o linie poligonală sub formă de octogon (Fig. 4.2.4). Un mod mai facil de a obține un octogon, este de a pleca de la dreptunghiul neorientat, care înscrie obstacolul, și de a delimita colțurile goale ale dreptunghiului prin secante ale acestuia. Această abordare tinde să modeleze orientarea obstacolului, în timp ce obstacolul nu este dreptunghiular. Desigur, unele laturi pot avea lungime zero, simulând astfel poligoane cu mai puțin de opt laturi.

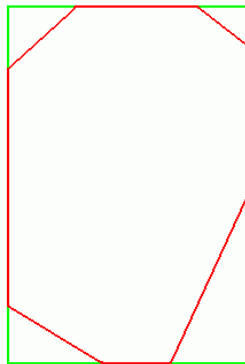


Fig. 4.2.4. Un exemplu generic de model octogonal.

În Fig. 4.2.5, e prezentat un exemplu concret. Aplicația poate alege să dea importanță diferită laturilor vizibile și celor nevizibile în spațiul camerei. Majoritatea punctelor 3D sunt reconstruite destul de corect pentru a indica structura obstacolului real, dar de regula sunt și puncte care alterează această structură. Calitatea potrivirii unui model octogonal pe structura reală depinde de iscusința algoritmilor folosiți.

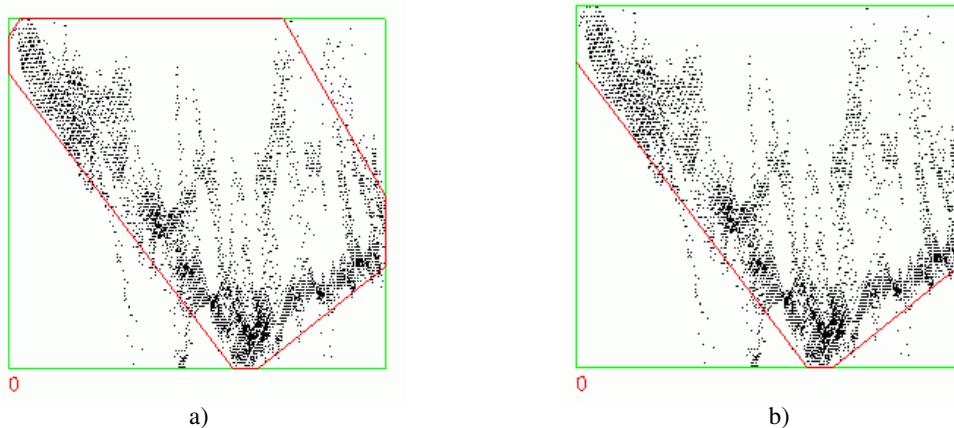


Fig. 4.2.5. Modelul octogonal, cu (a) și fără laturile nevizibile (b).

Unele obstacole reale pot avea forme mai complexe, cum ar fi un obstacol rotund. În astfel de cazuri, modelul octogonal propus, obținut dintr-un dreptunghi neorientat cu colțurile tăiate, poate să nu fie suficient de fidel (Fig. 4.2.6). O soluție ar fi plecarea de la un dreptunghi orientat, dar algoritmi se complică, ducând chiar la fluctuații mari în cadre succesive.



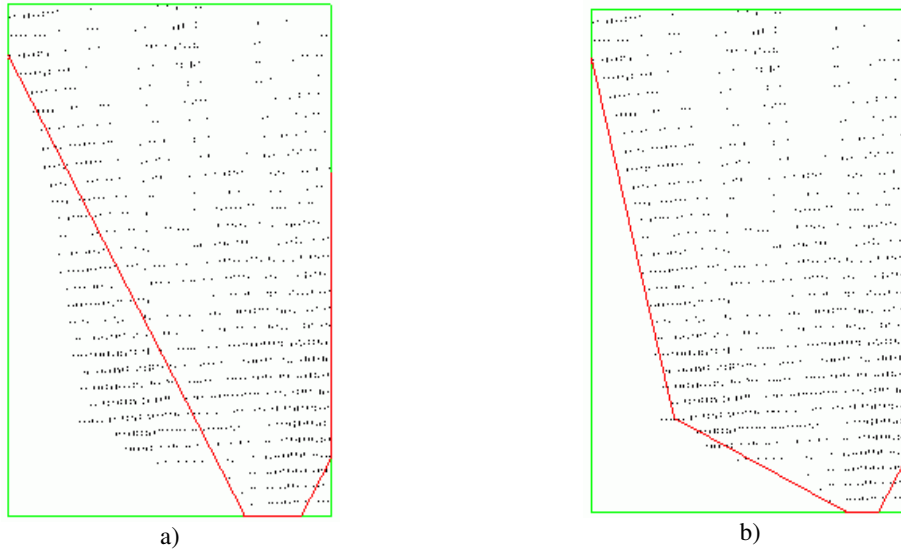


Fig. 4.2.6. Exemplu de obstacol care nu e potrivit pentru modelul octogonal (a). Un model fidel, dar nu octogonal (b).

#### 4.2.4 Modelul poligonal

Pentru obstacole generice, un model poligonal, cu număr de laturi după necesități, descrie cel mai fidel obstacolele [AMAA09] (Fig. 4.2.7 – spațiul top-view). Se poate obține pe baza înfășurătorii punctelor 3D, fiind și aceasta o linie poligonală. În plus, sunt necesari algoritmi de îmbunătățire a frontierei obstacolului, prin eliminarea punctelor 3D cu erori mari de lângă obstacolul real și prin evidențierea formei obstacolului real. Pentru partea din spatele obstacolului (cea nevizibilă), o astfel de linie poligonală poate fi de asemenea determinată, dar ar oferi doar o vagă idee despre limitele din spate ale obstacolului, adesea obstacolul real fiind mai lung de atât.

Beneficii ale modelului poligonal:

- Orientarea obstacolelor: în Fig. 4.2.8 lanțul de laturi, care determină orientarea, este desenat în verde, celelalte laturi vizibile fiind desenate în negru. În plus, poate fi ușor determinat un model dreptunghiular orientat, având orientarea lanțului verde și înscriind laturile vizibile ale modelului poligonal. Obstacolele mici au dimensiuni comparabile cu acuratețea reconstrucției, modelul poligonal neputând evidenția vreo orientare, astfel că nu ar trebui aplicați algoritmi pentru determinarea orientării.

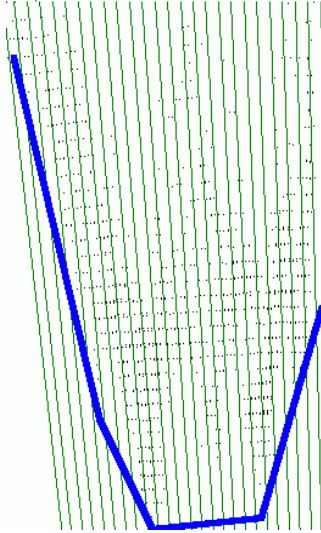


Fig. 4.2.7. Exemplu de model poligonal top-view.

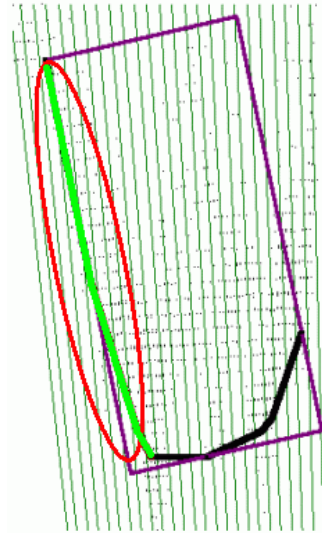


Fig. 4.2.8. Determinarea unui model dreptunghiular orientat pe baza modelului poligonal.

- Ocluzia obstacolelor: dacă se descompun laturile modelului poligonal în segmente de lățime egală în concordanță cu direcțiile optice, polare, ale camerei, se pot face ușor raționamente despre ocluziile de pe fiecare direcție optică: segmentul obstacolului aflat la profunzime mai mică (*ocluzant*) poate ocluziona segmentul obstacolului aflat la profunzime mai mare (*ocluzat*), fiind necesare în plus doar raționamente legate de înălțimea în spațiul imaginii. Dacă segmentul ocluzat are informație 3D, înseamnă că e cel puțin parțial vizibil, deoarece:

- segmentul ocluzant nu este suficient de mare față de segmentul ocluzat;
- segmentul ocluzant nu este continuu, permițând ca unele părți din spatele scenei să fie vizibile.

Informațiile provenite din analiza ocluziilor au aplicații precum:

- pentru un obstacol parțial ocluzat, evaluarea orientării ar putea fi evitată, sau nu, în funcție de gradul de ocluzie;
- un algoritm de tracking poate folosi orientarea și dimensiunile unui obstacol care trece în spatele altui obstacol;
- pentru simplificare, un segment ocluzat parțial poate fi considerat ocluzat total de unele procesări, sau vizibil total de alte procesari, după caz.

Un exemplu de folosire a modelului poligonal în spațiul imaginii este prezentat în figura următoare [Pocol03]:



Fig. 4.2.9. Exemplu de model poligonal în spațiul imaginii.

#### 4.2.5 Modelul poligonal cu rezoluție constantă și înălțime variabilă (stixel)

Dacă la fiecare latură a modelului poligonal top-view se adaugă și informație de înălțime, se obține un model 3D bazat pe un model 2D. Modelul este propus de [Badino09] și este compus dintr-un șir de elemente dreptunghiulare verticale, de lățimi polare egale, numite stixeli, denumire derivată din cuvintele stick și pixel. Fiecare stixel începe pe suprafața solului și are înălțime proprie. În plus, se clasifică în stixeli de prim-plan și stixeli de fundal, fiecare direcție optică putând avea câte un stixel din fiecare clasă.

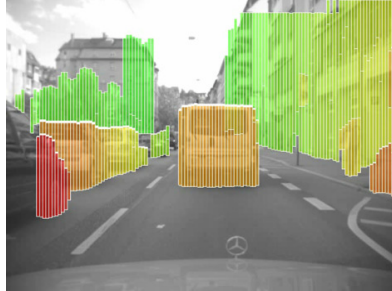


Fig. 4.2.10. Modelul poligonal cu rezoluție constantă și înălțime variabilă.

#### 4.2.6 Modelul poligonal cu rezoluție variabilă și înălțime constantă

Modelul poligonal propus în [Vătavu12] are rezoluție variabilă a laturilor, atât din punct de vedere polar cât și metric, dar pe axa înălțimii metrice au înălțime constantă. Are avantajul simplificării dar și stabilizării înălțimii întregului obstacol, altfel, înălțimile variabile ar putea fluctua mult în cadre succesive.

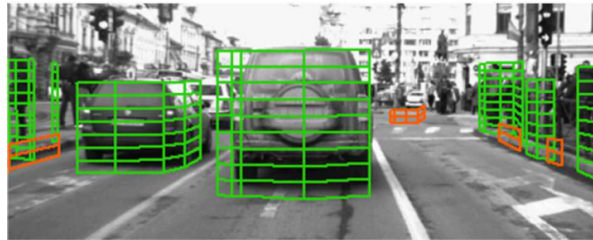


Fig. 4.2.11. Model poligonal cu rezoluție variabilă și înălțime constantă.

#### 4.2.7 Modelul cuboidal cartezian

Este printre cele mai folosite modele (Fig. 4.2.3.a) pentru că este generic și fidel multor obstacole reale, în special celor dinamice. Este schematic și sugestiv. Poate fi orientat sau neorientat (paralel cu axele de coordonate). Fiind descris numeric în sistemul de coordonate cartezian al scenei, înlătură limitările spațiului perspectiv al camerei și se pretează la analize în domeniul obstacolelor reale, cum ar fi dimensiunile, poziția, orientarea. De mare folos este și în cazul trackingului obstacolelor [Danescu07], aceasta fiind o tehnică ce procesează obstacolele din mai multe cadre succesive, deci are nevoie de o decuplare de spațiile perspective a acelor imagini și de o reprezentare într-un spațiu comun lor.

#### 4.2.8 Modelul cuboidal polar

Modelul cuboidal este prin excelență exprimat numeric într-un sistem de coordonate cartezian 3D, dar provine dintr-o detecție efectuată în imagini perspective, polare. Un cuboid 3D poate fi exprimat numeric într-un sistem polar, prin vectori descriși de magnitudine și unghi (Fig. 4.2.12). De exemplu, limitele stânga/dreapta ale feței cuboidului pot fi descrise fiecare prin distanța și prin coloana de pixeli corespunzătoare în spațiul imaginii. Coordonatele pixelilor în spațiul imaginii pot foarte bine și firesc să substituie unghiurile efective, făcând legătura directă cu spațiul senzorului, spațiul camerei. Centrul cuboidului poate fi și el exprimat numeric prin distanța și prin unghiurile față de axa optică principală a camerei.

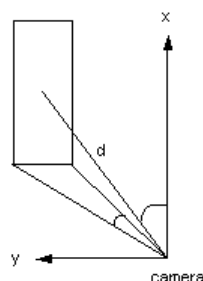


Fig. 4.2.12. Modelul cuboid polar – top-view.

#### 4.2.9 Modelul poliedral

O extensie firească a modelului poligonal, folosit în spațiul top-view și în spațiul imaginii, este modelul poliedral. Acesta este în principiu în spațiul 3D și provine din spațiul 2D al imaginii la care se adaugă și dimensiunea profunzimii. Un exemplu în care suprafața vizibilă (și măsurabilă ca profunzime) a unui obstacol este descrisă de un *mesh* compus din triunghiuri este:

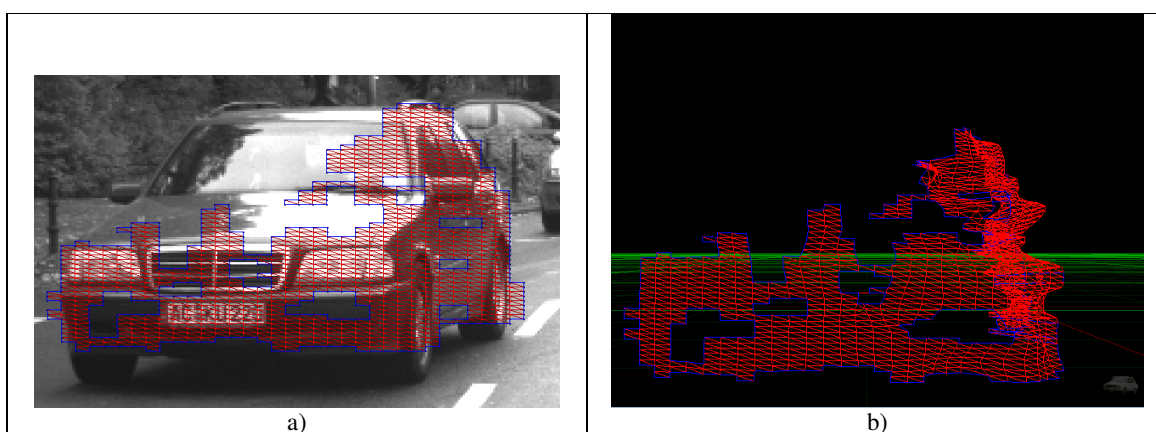


Fig. 4.2.13. Mesh de triunghiuri (a). Vedere dintr-un punct arbitrar (b).

Descompunerea în triunghiuri este utilă atunci când se dorește creșterea rezoluției locale, prin divizarea triunghiurilor, având avantajul că un triunghi împărțit în două va produce tot triunghiuri. Atunci când un obstacol relativ mare se apropie de ego-car, poate fi de dorit să-i crească rezoluția locală pentru acele părți ale mesh-ului care apar, în spațiul imaginii, mai mari decât erau inițial (Fig. 4.2.14). Condiția pentru creșterea rezoluției locale ar putea fi ca lungimea unui arc proiectat în imagine să depășească cu 50% lungimea inițială. Prin divizarea

arcului, în punctul de divizare (mijlocul arcului) vor rezulta două sub-arce în locul celui inițial, iar cele două triunghiuri la care era adiacent vor fi divizate introducând noi arce care să unească punctul de divizare cu vârfurile opuse ale acelor două triunghiuri. Creșterea rezoluției locale sau globale este numită *teselare*.

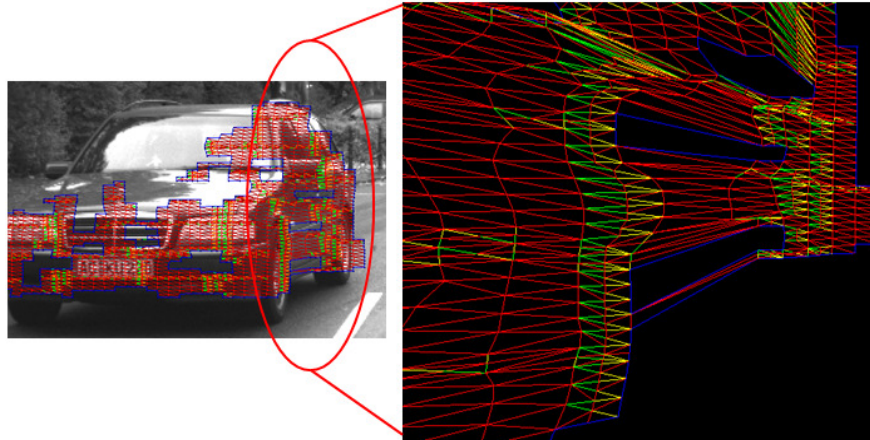


Fig. 4.2.14. Creșterea rezoluției locale. Divizarea unui arc produce două sub-arce (verde și galben).

Avantajele acestui model sunt: modelarea oricărei forme 3D, creșterea rezoluției locale prin teselare, păstrarea rezoluției maxime atunci când obstacolul se îndepărtează, clasificarea nodurilor și a arcelor (vizibil, nevizibil, la limita vizibilității, la marginea mesh-ului, provenit din teselare etc). Principalul dezavantaj este complexitatea structurilor de date și a algoritmilor de procesare.

O altă posibilitate ar fi împărțirea suprafeței obstacolului în dreptunghiuri. Așa cum modelul poligonal top-view cu rezoluție constantă împarte spațiul orizontal al camerei în direcții optice polare, tot așa și modelul mesh-ului de dreptunghiuri împarte spațiul bidimensional al camerei în piramide optice polare. Împărțirea în dreptunghiuri complică mult operația de teselare. În schimb, divizarea fiecărui dreptunghi în două triunghiuri duce repede la un mesh de triunghiuri, așa cum s-a obținut și mesh-ul din Fig. 4.2.14.

Modelul poliedral înlesnește analiza ocluziei parțiale a obstacolelor, poate oferi o bază pentru o analiză mai avansată a formelor 3D, poate memora textura sau poate conduce la descoperirea de noi tehnici de evaluare și corectare a erorilor de reconstrucție.

#### 4.2.10 Alte modele

Se pot imagina și alte modele:

- modelul poligonal cu rezoluție constantă și înălțime constantă: are simplitatea rezoluției constante și stabilitatea înălțimii;
- modelul poligonal cu rezoluție variabilă și înălțime variabilă: schematic și flexibil;
- modelul poligonal cu rezoluție constantă adaptabilă: rezoluția întregului poligon este crescută atunci când obstacolul se apropie, fiind practic o operație de teselare;
- modelul cuboidal cu orientări multiple: compus din mai multe cuboide cu aceeași înălțime, calculat pentru mai multe orientări posibile; util atunci când sunt mai multe orientări posibile, dar niciuna nu e foarte bună;



- modelul cuboidal orientat în jurul celor trei axe: util atunci când suprafața solului nu este planară și nu se poate face supoziția că obstacolul are același tangaj și ruluu ca și ego-car;
- modele polare: diferite modele construite după principiile modelului cuboidal polar;
- modelul multi-element (Fig. 4.2.15) și modelul “curbat” (Fig. 4.2.16).



Fig. 4.2.15. Modelul multi-element



Fig. 4.2.16. Modelul “curbat”

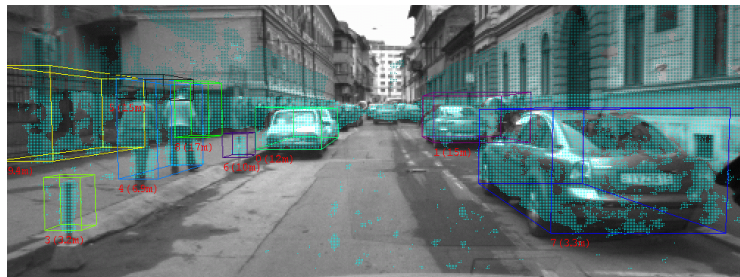
\*\*\*

Unele modele sunt catalogate a fi 3D, dar ar fi mai corect să fie numite 2.5D. De exemplu, un model top-view poligonal + înălțime este totuși limitat la un șir de dreptunghiuri verticale. Însă modelul poliedral este cu adevărat 3D, putând modela orice formă 3D.

#### 4.2.11 Mai multe modele pentru un obstacol

În partea stângă a scenei din Fig. 4.2.17.a sunt două obstacole subțiri, doi stâlpișori detectați drept obstacolele 3 și 6. În Fig. 4.2.17.c, vederea de sus relevă faptul că adâncimea detectată este mai degrabă dată de inacuratețea punctelor 3D reconstruite, în timp ce adâncimea părților vizibile a obstacolelor reale este foarte mică. Fiind obstacole subțiri, nu se încearcă determinarea unei orientări. Mai mult, când astfel de obstacole sunt în lateralul imaginii, modelul cuboidal neorientat pare și mai mare, cuprinzând mult mai mult spațiu decât obstacolul real.

Soluția este ca astfel de obstacole să aibă atașat și un model dreptunghiular în spațiul imaginii. În Fig. 4.2.17.b se poate vedea acest model calculat pentru toate obstacolele din scenă, putând fi folosit de diverse procesări precum clasificarea obstacolelor pe baza texturii și a formei 2D în spațiul imaginii (cum ar fi clasificarea stâlpilor).



a)

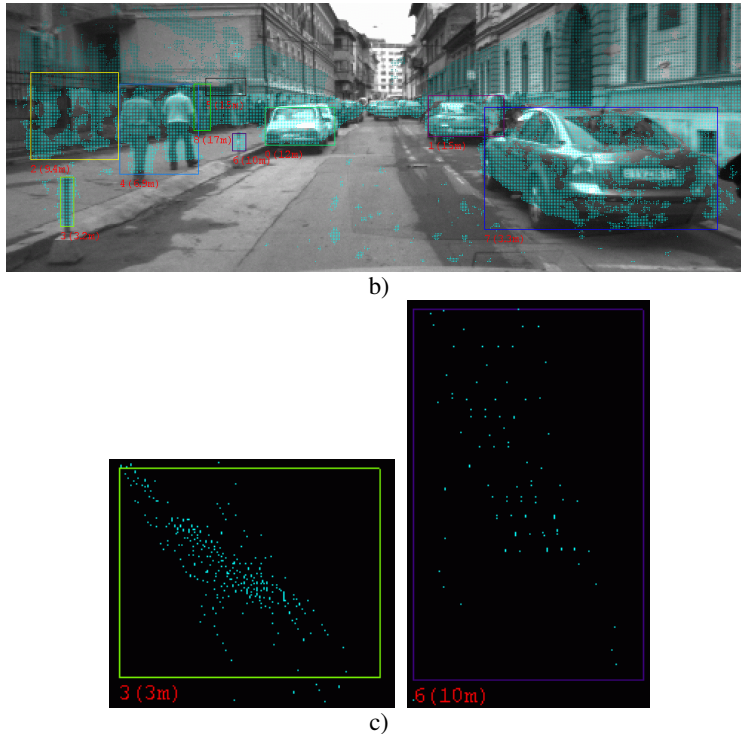
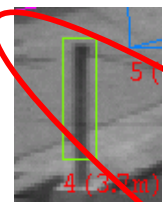


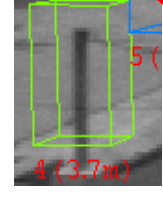

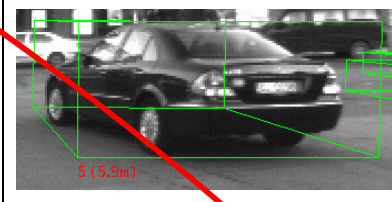
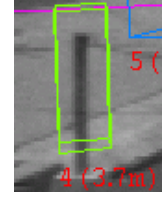

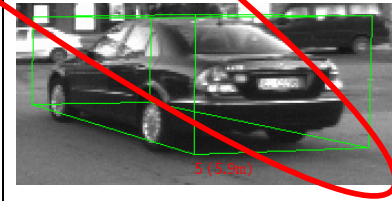


Fig. 4.2.17. Obstacole modelate prin: a) cuboide, b) dreptunghiuri în imagine, c) dreptunghi top-view pentru obstacolele 3 și 6.

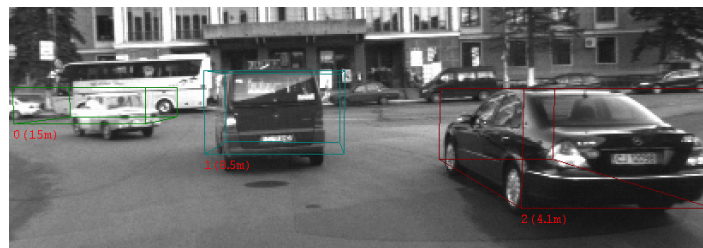
Necesitatea ca un singur obstacol să fie modelat simultan prin mai multe modele rezultă atât din motive estetice, de afișare, cât și din motive de procesare, diferiți algoritmi necesitând diferite modele. Desigur, o problemă tipică este menținerea consistenței tuturor modelelor simultan.

Un exemplu concret ar fi folosirea simultană a modelelor: cuboid neorientat, cuboid orientat și dreptunghi în spațiul imaginii. În fiecare moment trebuie să fie disponibile, pentru fiecare obstacol, și indicii despre cât de bine se potrivește fiecare model aceluia obstacol. De exemplu, indicele “plat” arată că obstacolul nu are o adâncime considerabilă, iar indicele “orientat” indică o orientare bine determinată. Tabelul de mai jos exemplifică modelul afișat, în funcție de opțiunea utilizatorului:

- dreptunghi în spațiul imaginii
- cuboid neorientat
- cuboid orientat
- Auto: se folosește modelul cel mai potrivit, după următorul raționament:
  - o dacă e “plat”, atunci se folosește dreptunghiul în spațiul imaginii,
  - o altfel, dacă este “orientat”, se folosește cuboidul orientat,
  - o altfel se folosește cuboidul neorientat.

Plat	Cuboid neorientat (obstacolul e undeva în lateral)	Cuboid orientat	Tipul obstacolului Modelul afișat
			<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Image Rectangle</li> <li><input type="checkbox"/> 3D Box</li> <li><input type="checkbox"/> 3D Oriented Box</li> <li><input type="checkbox"/> Auto (Best Suited)</li> </ul>
			<ul style="list-style-type: none"> <li><input type="checkbox"/> Image Rectangle</li> <li><input checked="" type="checkbox"/> 3D Box</li> <li><input type="checkbox"/> 3D Oriented Box</li> <li><input type="checkbox"/> Auto (Best Suited)</li> </ul>
			<ul style="list-style-type: none"> <li><input type="checkbox"/> Image Rectangle</li> <li><input type="checkbox"/> 3D Box</li> <li><input checked="" type="checkbox"/> 3D Oriented Box</li> <li><input type="checkbox"/> Auto (Best Suited)</li> </ul>
			<ul style="list-style-type: none"> <li><input type="checkbox"/> Image Rectangle</li> <li><input type="checkbox"/> 3D Box</li> <li><input type="checkbox"/> 3D Oriented Box</li> <li><input checked="" type="checkbox"/> Auto (Best Suited)</li> </ul>

Modelul cuboidal neorientat și modelul dreptunghiular în spațiul imaginii sunt ușor de calculat, pentru orice obstacol. Nu la fel stau lucrurile cu modelul cuboidal orientat. Pentru obstacole a căror orientare nu poate fi determinată, nu se setează indiciul “orientat”, dar se calculează totuși un cuboid, cât de cât valid, care este orientat de-a lungul axei optice care trece prin centrul obstacolului. Un exemplu este obstacolul 1 din figura de mai jos:



a)



b)



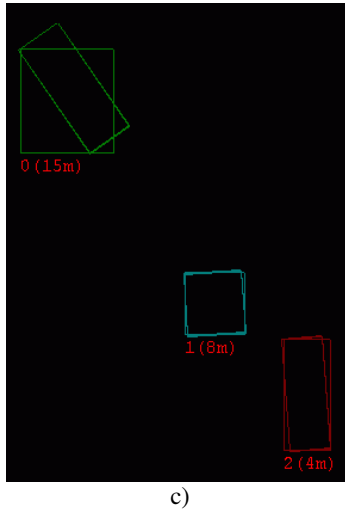


Fig. 4.2.18. a) – Cuboide neorientate, b) – cuboide orientate, c) top-view – atât cuboide orientate cât și cuboide neorientate.

### 4.3. Concluzii, contribuții

Acest capitol detaliază modelele implementate: dreptunghiular neorientat, dreptunghiular orientat, octogonal, poligonal, cuboidal neorientat, cuboidal orientat, poliedral, multi-element și curbat; dar și altele din literatură: poligonal cu rezoluție variabilă și înălțime constantă, poligonal cu rezoluție constantă și înălțime variabilă (stixel), cuboidal polar etc. Majoritatea au fost implementate în cadrul tezei de față, mai mult ca urmare a necesității decât a inspirației din literatură. Analiza modelelor este originală.

Tot aici, se prezintă sistemele de coordonate folosite pentru modelarea domeniului problemei (scena reală) și domeniului soluției (stereo viziunea). Astfel, se prezintă sistemul de coordonate cartezian 3D potrivit pentru descrierea scenei reale, dar și sistemul de coordonate 3D nativ al sensorului de stereo viziune, potrivit pentru posibilitățile reale de percepție a scenei.

## Capitolul 5. Abordări existente în domeniul detecției de obstacole

Domeniul detecției de obstacole a fost amplu studiat în ultimele decade, de către echipe de cercetători din toată lumea. Astfel, studiul de față își propune să prezinte mai în detaliu doar o parte a domeniului, încercând să amintească totuși cât mai multe direcții cunoscute. În sub capitolul abordărilor care folosesc stereo viziunea, este originală clasificarea acestor abordări în funcție de profunzimea folosită și în funcție de spațiul de procesare folosit preponderent.

### 5.1. Clasificarea abordărilor

Există mai multe posibilități de a clasifica abordările existente în domeniul detecției de obstacole. [Hilario2005] propune trei categorii de abordări (orientate pentru obstacole de tip vehicul):

- **Bottom-up.** Întâi se extrag din imagine diferite trăsături specifice care aparțin în mod uzual obiectelor (muchii, zone cu simetrie verticală, zone mai întunecate – posibile umbre sub și lângă vehicule) și apoi acestea se grupează generând ipoteze de obstacole. Validarea ipotezelor se face prin urmărirea lor în imagini succesive. Aceste abordări pot da greș pentru acele obstacole (în particular vehicule) pentru care lipsesc una sau mai multe trăsături specifice.
- **Top-down.** Se caută obstacole cu anumite forme, care sunt reprezentate prin modele (de ex. spatele mașinilor are un model rectangular). Aceste modele se caută în imagine, calculându-se valoarea unei funcții de asemănare. Pentru asemănare mare se generează ipoteze de obstacole. Din nou, validarea ipotezelor se face prin urmărirea lor în imagini succesive. Aceste abordări sunt mai robuste decât cele din prima categorie, dar sunt mai încete și sunt limitate la a detecta doar obstacole care respectă modelele folosite.
- **Bazate pe învățare.** În principal, acestea folosesc rețele neuronale antrenate cu multe imagini care conțin tipurile de obstacole ce se doresc a fi detectate. Deoarece rețelele neuronale necesită putere de calcul mare, sunt aplicate doar pe zonele de imagine care conțin ipoteze de obstacole generate prin metode din primele două categorii.

Deși, în general, abordările de mai sus au fost orientate pe detecția în imagini statice, există și extinderi ale lor care se orientează pe procesarea imaginilor în mișcare (de ex. rețele neuronale care recunosc anumite șabloane de mișcare în imagini succesive: [Fang03]).

[Regensburger94] propune tot trei categorii de abordări, orientate pe căutarea unor modele în fiecare imagine, analizarea mișcării în imagini succesive sau exploatarea informațiilor 3D:

- Abordări bazate pe **modele de obiecte**. Obstacolele sunt localizate prin detectarea unor trăsături particulare, relaționate într-un anumit mod.
- Abordări bazate pe **analiza mișcării** în imagini succesive. Pentru un set de trăsături de interes, se calculează deplasarea lor în imagini succesive. Detecția de obstacole exploatează faptul că trăsăturile de pe obiecte se deplasează diferit față de trăsăturile de pe structurile statice.
- **Reconstrucția 3D** a scenei și analiza relațiilor de poziționare a structurilor 3D.

În lucrarea de față, se propune o clasificare asemănătoare cu cea din [Regensburger94], punând accentul pe capacitatea abordărilor de a recunoaște obstacole generice și de a furniza informații despre alcătuirea 3D a scenei.

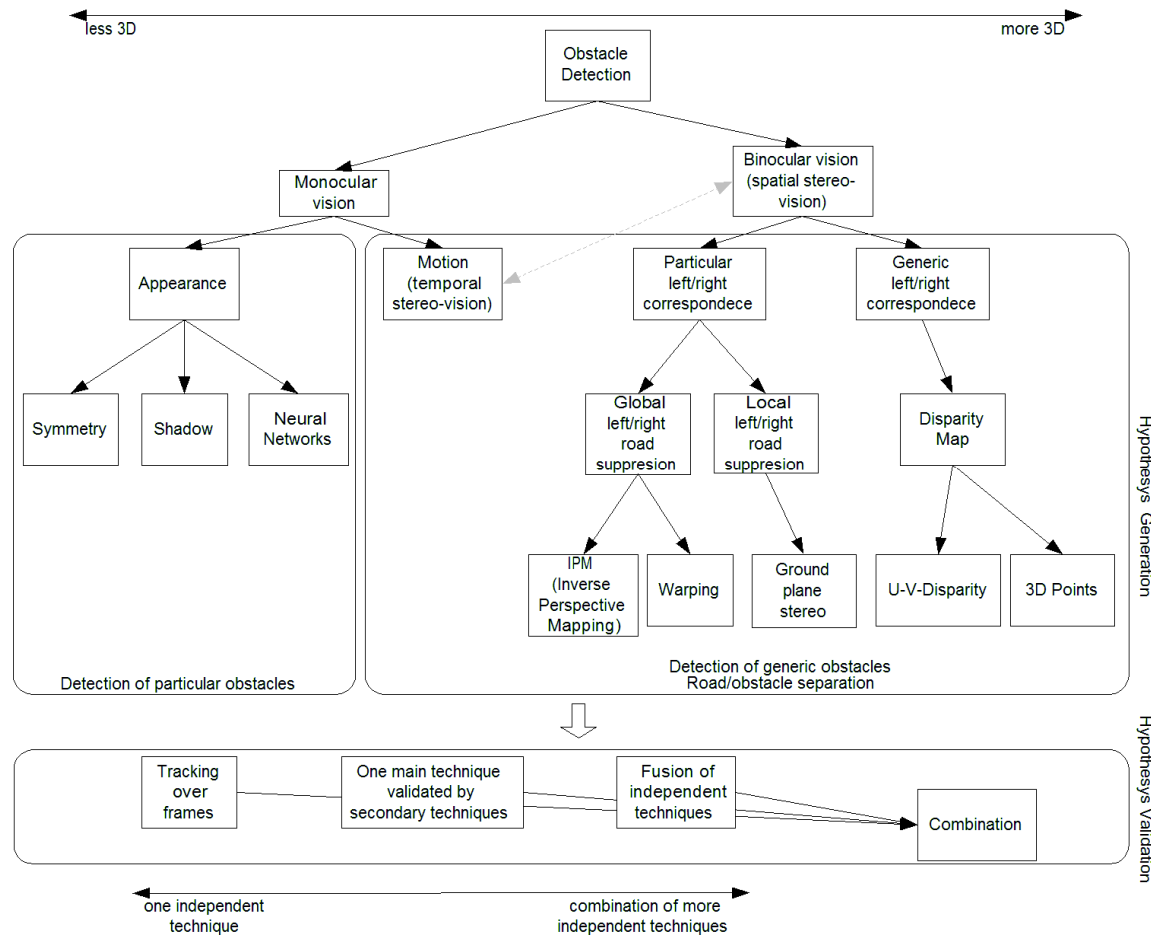


Fig. 5.1.1. Clasificarea abordărilor pentru detecția de obstacole.

În secțiunile următoare vor fi prezentate diferite metode, grupate după tipul de informație folosită: imagini mono de intensitate, imagini mono color, secvențe de imagini mono, imagini stereo. Vor fi propuse și posibile îmbunătățiri, atât la nivel de cadre individuale cât și de urmărire în cadre succesive. În plus, există și îmbunătățiri extrinseci metodelor (Fig. 5.1.1, partea de jos):

- validarea ipotezelor generate de o metodă, printr-o altă metodă independentă;
- fuziunea rezultatelor obținute prin metode independente, inclusiv prin urmărirea în timp a acestor rezultate.

## 5.2. Detecția din imagini mono, de intensitate

Folosirea unui sistem monocular este motivată de simplitatea echipamentului și, implicit, de costul scăzut. De asemenea, procesul de calibrare a unui astfel de sistem este destul de simplu, iar acuratețea parametrilor obținuți prin calibrare nu este critică.

Un mare dezavantaj este acela că, în cele mai multe abordări, monoviziunea doar estimează distanțele până la obstacole. Calcularea cu precizie a distanțelor nu este posibilă.

Abordările bazate pe monoviziune se împart în două mari categorii: acelea care pot detecta obstacole în imagini singulare și acelea care detectează obstacolele prin analiza imaginilor succesive.

Cele din prima categorie folosesc indicii particulare bazate pe aparența obstacolelor în imagine. Dezavantajul lor este că sunt limitate la a detecta doar obstacolele care prezintă aceste particularități.

În toate scenariile de autostrada, toate vehiculele se deplasează în același sens cu ego-car și pe traiectorii cvazi-paralele cu acesta. Așadar, partea din spate a acestora va fi cea vizibilă în imaginile achiziționate. Astfel, multe abordări orientate pe aparență exploatează diferite proprietăți ale spatelui vehiculelor. Dezavantajul major este că nu pot fi detectate obstacolele care nu au aceste proprietăți (inclusiv a vehiculelor care sunt orientate și poziționate în așa fel încât partea din spate nu este suficient de vizibilă). Pentru creșterea robusteții, implementările descrise în literatură combină exploatarea mai multor proprietăți. În subsecțiunile următoare, vor fi descrise aceste proprietăți și exploatarea lor.

În cazul căutării unor obstacole a căror aparență respectă un anumit model, modelele sunt caracterizate de o serie de trăsături particulare (muchii orizontale și verticale, zone cu simetrie verticală, zone mai întunecate – posibile umbre sub vehicule) relaționate între ele într-un mod specific. Astfel, detecția de obstacole se face prin localizarea acestor trăsături.

### 5.2.1 Simetrie

În general, spatele unui vehicul prezintă o puternică simetrie verticală:

- forma caroseriei și a bandoului este simetrică;
- caroseria este vopsită într-o singură culoare;
- numărul de înmatriculare este situat în mijloc;
- cele două blocuri optice sunt situate la extremități și au construcție simetrică; roțile sunt de asemenea situate la extremități și arată la fel;
- parbrizul prezintă simetrie verticală, adesea incluzând ceea ce se vede în interiorul vehiculului (tetiere, ocupanți), cu toate că ocupanții și alte obiecte din interiorul vehiculului pot fi elemente de asimetrie.

În cele ce urmează, se prezintă o metodă de punere în evidență a zonelor cu simetrie verticală, într-o imagine de intensitate [Regensburger94]. Pentru început se alege, în mod empiric, lățimea  $w$  în pixeli (40) a spatelui vehiculelor (obiectelor) ce se doresc a fi detectate. Desigur, pentru aceasta, se ține cont de rezoluția imaginii și de distanța cea mai probabilă la care s-ar putea afla vehicule. Pentru fiecare pixel din zona de interes, se calculează o măsură a simetriei pixelilor aflați de o parte și de alta, pe același rând de imagine. Pentru fiecare pereche de pixeli concentrici față de pixelul curent ( $w/2$  perechi), se scade valoarea intensității pixelului din dreapta din valoarea intensității pixelului din stânga (Fig. 5.2.1.a). Valoarea măsurii simetriei se obține prin însumarea valorilor absolute ale acestor diferențe. Cu cât se obțin valori mai mici, cu atât simetria stânga/dreapta este mai mare. Valorile acestei măsuri, a simetriei tuturor pixelilor aflați de-a lungul unui rând de pixeli, vor descrie funcția de simetrie. În Fig. 5.2.1.b se arată graficele funcțiilor de simetrie: câte un grafic pentru fiecare rând de imagini. De-a lungul unui rând de imagine, minimele locale ale funcției de simetrie indică centrele de simetrie verticală (în Fig. 5.2.1.c sunt arătați pixelii cu cele mai

mici minime locale). Coloana de imagine, cu cele mai multe și mai mici minime locale, indică centrul unui obiect a cărui imagine este vertical simetrică. Dacă  $w$  este semnificativ mai mic decât lățimea obiectului, atunci apare riscul ca, în mijlocul zonei simetrice, toate valorile funcției de simetrie să fie mici și să nu poată fi localizat minimul local cu precizie. Dacă  $w$  este semnificativ mai mare decât lățimea obiectului, atunci apare riscul ca valorile funcției de simetrie, de pe colona din mijlocul obiectului, să fie puternic influențate de un background asimetric; astfel minimul local este mutat spre stânga sau spre dreapta.

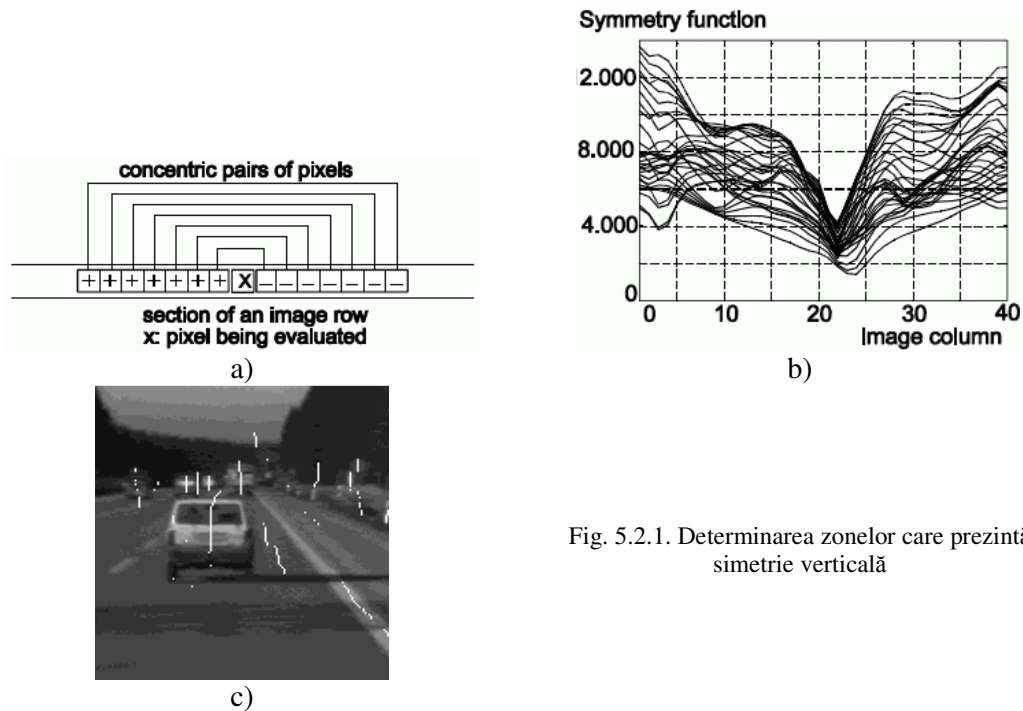


Fig. 5.2.1. Determinarea zonelor care prezintă simetrie verticală

Posibile îmbunătățiri:

- Alegerea empirică a lățimii zonelor simetrice țintite, este eliminată în [Broggi94], unde se folosește o schemă multi-rezoluție. Astfel, metoda este capabilă să detecteze spatele vehiculelor aflate la diferite distanțe și, în același timp, să scurteze timpii necesari detecției vehiculelor aflate la distanțe mici și care apar mari în imagini.
- Pentru mărirea robusteții, pe lângă simetria verticală a imaginii de intensitate, se pot impune și simetrii a două imagini suplimentare: imaginea muchiiilor verticale și imaginea muchiiilor orizontale [Broggi94, Bertozzi01, Bombini06]. În general, muchiile verticale și cele orizontale surprind formele cvazi-dreptunghiulare ale caroseriei, parbrizului, plăcii numărului de înmatriculare (dacă este montată central), bandoului și a umbrei de sub mașină. Simetria imaginii de intensitate poate fi puternic afectată de iluminarea diferită a părților caroseriei și de artefacte desenate sau atașate pe caroserie (Fig. 5.2.2), în timp ce muchiile verticale și orizontale sunt mai puțin sensibile la astfel de factori și tind să surprindă elementele cvazi-dreptunghiulare enumerate mai sus.

Limitări:

- atunci când lumina soarelui luminează diferit partea stângă și cea dreaptă (Fig. 5.2.2.a);
- atunci când forma caroseriei nu este simetrică (Fig. 5.2.2.b) sau
- când culoarea vopselei nu este uniformă (Fig. 5.2.2.c);

- numărul de înmatriculare nu este situat în mijloc (Fig. 5.2.2.c).



a)



b)



c)

Fig. 5.2.2. Cazuri când imaginea spatelui vehiculului este mai puțin simetrică.

### 5.2.2 Umbra

Umbra care apare pe șosea, sub vehicule, în orice condiții meteo, este o trăsătură invariantă pentru toate vehiculele și poate fi folosită ca un bun indiciu pentru detecție.

Intensitatea umbrelor este mai mică decât intensitatea drumului. În [Krips04], pentru determinarea intensității medii ( $\mu$ ) a pixelilor aparținând drumului, se scanează coloanele imaginii, de jos în sus, până la întâlnirea de muchii. Totodată, se calculează și dispersia ( $\sigma$ ) acestor intensități. În [Hoffmann04],  $\mu$  și  $\sigma$  se calculează într-o zonă dreptunghiulară din partea de centru-jos a imaginii. În cazul în care  $\sigma$  este mare, înseamnă că în zona analizată există și marcaje sau artefacte, astfel că se va căuta automat altă zonă din partea de jos a imaginii, cu  $\sigma$  mic. Intensitatea pixelilor aparținând umbrelor se impune a fi mai mică decât  $\mu - 3\sigma$ .

Prin identificarea zonelor din imagine conținând intensități corespunzătoare umbrelor, s-ar găsi o mulțime de zone care nu aparțin vehiculelor. Pentru eliminarea acestora se impun unele constrângeri suplimentare:

- zona trebuie să fie cvazi-orizontală;
- imaginea de deasupra umbrei (posibilul spate de vehicul) trebuie să aibă o entropie mare (adică histograma nivelelor de gri să aibă vârfuri puține și cu distanță mare între ele);
- la marginile zonei trebuie să existe variații mari ale intensității;
- mai jos de zonă, trebuie să fie suprafața drumului (media și dispersia intensităților să fie apropiate de  $\mu$  și  $\sigma$ ): se elimină zonele întunecate care nu sunt pe drum;
- lățimea și înălțimea zonei să îndeplinească constrângeri geometrice obișnuite obstacolelor de tip vehicul.

Posibile îmbunătățiri:

- Urmărirea în cadre succesive a ipoteticului obstacol de deasupra umbrei; dacă umbra nu are aceeași traiectorie va fi ignorată.

- Cunoscând înclinația aproximativă a camerei, se poate determina cu o bună precizie poziția verticală maximă în imagine a drumului (vanishing point), inclusiv a umbrelor, care sunt incluse în drum. Astfel, ipotezele din partea de sus a imaginii ar fi din start eliminate.

Limitări:

- Metoda nu funcționează corect atunci când pe suprafața drumului există variații mici ale intensității, dar care se întind pe suprafețe mai mari (variații de tip dégradé), neexistând muchii. De asemenea, ipoteza că intensitatea suprafeței drumului este constantă și determinabilă nu mai este validă. Cu alte cuvinte, vecinătatea drum-umbră trebuie evidențiată altfel. De exemplu: zone întunecate și de intensitate aproximativ constantă (umbre) aflate deasupra unor zone mai luminoase și cu intensitate aproximativ constantă (drum).

### 5.2.3 Rețele neuronale

Aplicarea pe întreaga imagine a algoritmilor bazați pe rețele neuronale este mult prea înceată. De aceea, soluția este de a găsi ipoteze de obstacole folosind alte criterii (simetrie, umbră, s.a.), și de a aplica rețele neuronale doar pe zonele de imagine unde se află aceste ipoteze.

Rețeaua neuronală este antrenată pentru a recunoaște unul sau mai multe tipuri de vehicule. Cea prezentată în [Ran99] este de tipul feed-forward, back-propagation cu două nivele. Ca intrare se folosesc imagini de 30x30 pixeli. Ieșirea rețelei constă în clasificarea ipotezei ca automobil, camion sau non-vehicul (Fig. 5.2.3).

Cel mai important factor care determină performanța rețelei este modul de reprezentare a intrării. Există două posibilități:

- Trăsături considerate reprezentative (muchii, pete de culoare), extrase prin diverse metode.
- Imaginea grayscale a ipotezei. Rețeaua neuronală urmează să surprindă trăsăturile reprezentative.

În abordarea prezentată se folosește a doua posibilitate pentru că se auto-adaptează. Deoarece imaginea ipotezei poate avea diferite dimensiuni, dar aceasta este scalată la dimensiunea de 30x30 pixeli.

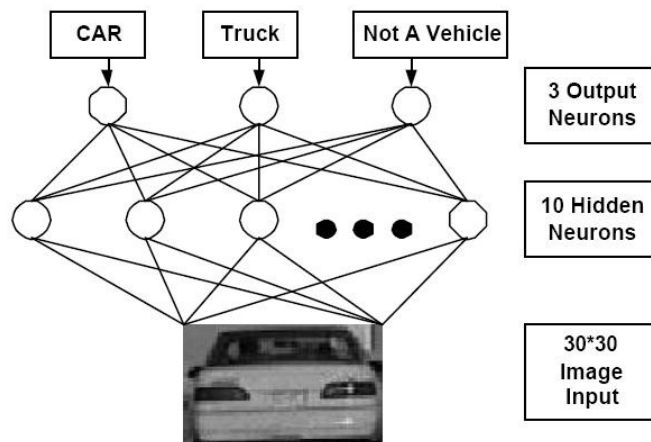


Fig. 5.2.3. Rețeaua neuronală.

Fiecare din cei trei neuroni ai ieșirii reprezintă un tip de obstacol. Astfel, unul dintre ei va avea valoarea 1 și ceilalți vor avea valoarea 0. De exemplu, ieșirea este 0, 1, 0 dacă rețeaua recunoaște un camion. Practic, având imagini de intrare imperfecte, valorile de ieșire nu vor fi perfecte, ci vor fi numere între 0 și 1. De exemplu, ieșirea ar putea avea tripletul de valori 0.2, 0.95, 0.1. Astfel, rezultatul rețelei va fi analizat de funcția de activare astfel încât ieșirea să aibă o valoare egală cu 1 și celelalte egale cu 0 (în exemplul dat, aceasta va fi: 0, 1, 0). Rezultatul acestei post-procesări este cel folosit în cele din urmă.

S-au experimentat diferite configurații prin varierea numărului de nivele ascunse, a dimensiunilor nivelelor și a modului de conectare între nivele. Concluzia este că aproape toate configurațiile indentifică rezonabil obiectele. Niciuna din ele, indiferent de complexitate, nu dă rezultate semnificativ mai bune decât cea din Fig. 5.2.3: un singur nivel cu 10 neuroni conectați total.

Funcția de activare a rețelei neuronale este de tip log-sigmoid. Funcția de activare log-sigmoid a fost aleasă pentru că are valori în intervalul 0..1, având semnificație booleană.

Rețeaua neuronală este antrenată folosind imagini din diferite scenarii: parcuri, străzi urbane, rurale și autostrăzi. Antrenarea se consideră completă când rata de recunoaștere este mai mare de 90%.

### 5.3. Detecția din imagini mono, color

Pentru fiecare pixel, sensorul unei camere color furnizează cantitățile culorilor de bază: roșu, verde și albastru (RGB). Acest triplet de valori se poate converti în tripletul HSI:

- Hue: frecvența din spectrul vizibil: roșu .. violet;
- Saturation: saturația sau gradul de puritate/murdărire, de amestec cu alte culori;
- Intensity: luminozitatea.

Componenta I este informația dată de imaginea de intensitate (achiziționată cu o cameră grayscale). Componentele H și S sunt cele care sunt furnizate în plus de către o cameră color. Componenta H a culorii unui obiect este valoroasă pentru că are aceeași valoare indiferent de modul în care este iluminată scena. Totuși, pentru culori care au saturație (S) mică, componenta H este foarte sensibilă la zgomete, devenind inutilizabilă. Cu cât este mai mică saturația, cu atât culoarea tinde spre gri.

Detecția obstacolelor din imagini color cunoaște două direcții principale:

1. Îmbunătățirea metodelor de detecție din imagini de intensitate (grayscale), prin adăugarea informației de culoare ([Rehrmann97] – detecția optical flow). Mai exact, se poate face deosebirea între două elemente din scenă, care produc aceeași intensitate luminoasă (aceeași "culoare" în imaginile grayscale), dar care au culori de frecvență și purități diferite, informații disponibile în imaginile color (Fig. 5.3.1). Nu se va insista pe discutarea acestei direcții, metodele în sine fiind prezentate în celelalte capitole.





Fig. 5.3.1. Imagine color (a) percepută de o camera grayscale (b).

2. Pe baza ipotezei ca în general obstacolele au altă culoare decât cea a drumului, se determină culoarea drumului și se cataloghează celelalte zone de imagine, care au altă culoare, ca fiind posibile obstacole [Soto00, Ulrich00].

Se presupune că în partea de jos a imaginii nu se află obstacole, ci doar suprafața rulabilă (drum, în particular). Culoarea drumului se determină într-un trapez din partea de jos a imaginii, numit “zonă de referință” (Fig. 5.3.2 a și c).

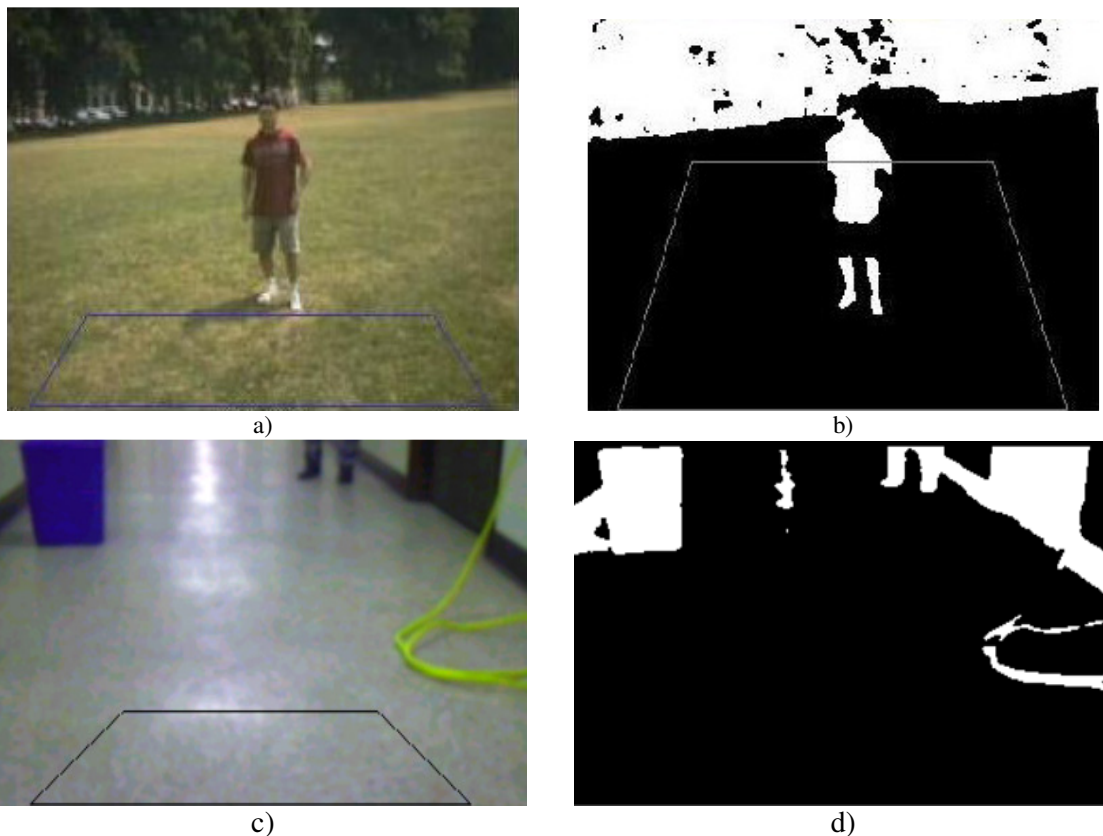


Fig. 5.3.2. a și c: Culoarea drumului se determină într-un trapez din partea de jos a imaginii; b și d: rezultatul detecției.

În zona de referință se consideră pixeli valizi doar aceia care au intensitatea și saturatia peste niște praguri minime. Cu acești pixeli se construiesc două histograme: una a componentei H și una a componentei I. Asupra fiecărei histograme se aplică o operație de netezire și una de normalizare (în intervalul 0 .. 100). Pasul principal este cel al clasificării pixelilor din imagine în pixeli de pe suprafața rulabilă și pixeli ai obstacolelor. Un pixel aparține suprafeței rulabile

dacă intrările corespunzătoare în cele două histogramme au cel puțin valoarea 60 pentru H și 80 pentru I. Ceilalți pixeli aparțin obstacolelor (Fig. 5.3.2 b și d).

Pe distanțe relativ scurte, considerând că suprafața rulabilă este plană și cunoscându-se înclinarea camerei, pentru fiecare coloană din imagine se poate estima cu o bună precizie distanța până la cel mai apropiat obstacol folosind înălțimea în imagine a primului pixel (de jos în sus) clasificat ca obstacol. Desigur, pentru obstacole care nu se sprijină pe sol, distanța măsurată va fi una eronată, mai mare decât cea reală.

Limitări și posibile îmbunătățiri:

- Asumpția că obstacolele au culoare diferită de cea a suprafeței rulabile poate conduce la nedetectarea obstacolelor care au culoare apropiată de cea a suprafeței rulabile. În cazul în care în zona de referință se află un obstacol, acesta va fi detectat ca suprafață rulabilă.

Totuși, conform [Ulrich00], în cazul roboților parțial supervizați, operatorul uman poate antrena sistemul într-o parte a scenei unde zona de referință conține doar suprafață rulabilă. În acest caz, cele două histogramme sunt calculate doar în faza de antrenare și folosite pentru imaginile următoare. Și în acest caz apar probleme atunci când se ajunge într-o zonă în care suprafața rulabilă are alte culori decât cele din momentul antrenării. Conform autorilor, soluția ar fi o învățare continuă a culorilor suprafeței rulabile, dar autorii omit să specifice că la apariția unei noi culori, sistemul nu știe dacă aparține unui obstacol sau suprafeței rulabile. Totuși, autorii propun rularea robotului sub supravegherea unui operator uman, care să comande robotul atunci când acesta se oprește fără a avea un obstacol real. Concluzia este ca în cazul sistemelor ce se doresc a fi nesupervizate sau a sistemelor de asistare a șoferilor, abordarea nu reușește să separe corect suprafața rulabilă de obstacole în toate situațiile.

## 5.4. Detecția din secvențe de imagini

Datorită unor mecanisme complexe, inclusiv cel de învățare, vederea umană este capabilă să identifice cu ușurință elementele ce compun scena în imagini statice. Percepția din secvențe video este cu atât mai bună.

În cazul vederii artificiale, chiar și în medii structurate, identificarea elementelor din scenă, prin analiza unei imagini statice, este limitată la acele elemente care prezintă particularități de aparență: formă, aranjament al muchiilor, simetrie verticală, prezența umbrei (zona mai întunecată), aparența învățată prin rețele neuronale etc.

Prin analiza secvențelor video, detecția poate fi extinsă și la acele elemente care nu prezintă particularități de aparență, dar care au o mișcare diferită de celelalte elemente din jurul lor. Așadar, elementul de bază este detecția mișcării, a deplasării între imagini succesive.

Se analizează vectorii de deplasare, în imagini succesive, a unor trăsături mici – compuse din câteva zeci de pixeli vecini. Trăsăturile mici sunt aproximativ nedeformabile în timp. Pentru a crește șansa ca o trăsătură să fie găsită în mod unic în imaginea următoare, trebuie folosite trăsături cu grad mare de unicitate precum colțurile – intersecția a două sau mai multe muchii.

### 5.4.1 Diferența imaginilor IPM succesive

Prin însuși modul de capturare a imaginii scenei folosind camere, se obțin imagini de tip perspectivă. Astfel, două obiecte identice, deci care au aceleași dimensiuni în spațiul 3D al scenei, dacă se află la profunzimi diferite, vor avea dimensiuni diferite în spațiul 2D al imaginii. De asemenea, suprafața drumului apare mai îngustă pe măsură ce crește profunzimea.

Maparea inversă perspectivei (Inverse Perspective Mapping = IPM) este o tehnică ce transformă imaginea perspectivă într-o imagine top-view, imagine tip satelit, în care scena este văzută de sus (Fig. 5.4.1). În timp ce în imaginea perspectivă lățimea drumului variază în funcție de profunzime, în imaginea IPM lățimea drumului este constantă, indiferent de profunzime. Imaginea IPM mai este numită și imaginea drumului, o cameră virtuală având planul imaginii paralel cu suprafața drumului și fiind amplasată undeva sus deasupra acestuia. În Fig. 5.4.3, intervalele de drum având lungime 3D constantă, arătate în imaginea IPM, sunt puse în corespondență cu cele din imaginea perspectivă. După cum se poate observa, partea de sus a imaginii IPM are o calitate mai slabă decât partea de jos, de vreme ce pixelii sursă din imaginea perspectivă sunt mai puțini. O altă observație, legată de Fig. 5.4.1 și Fig. 5.4.3, este că imaginea IPM este deformată, fiind de 5 ori mai comprimată pe înălțime decât pe lățime; imaginea nedeformată este prezentată în Fig. 5.4.2.

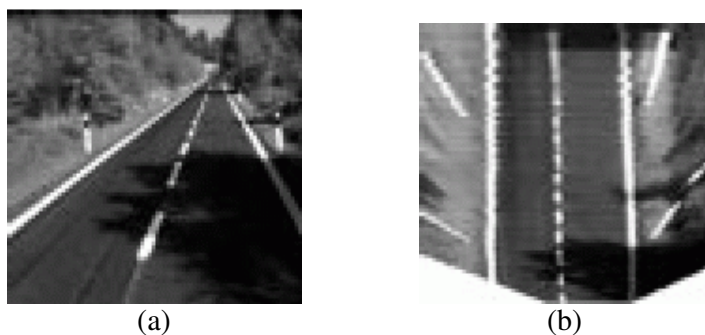


Fig. 5.4.1. Transformarea IPM: (a) imaginea perspectivă, (b) imaginea IPM.

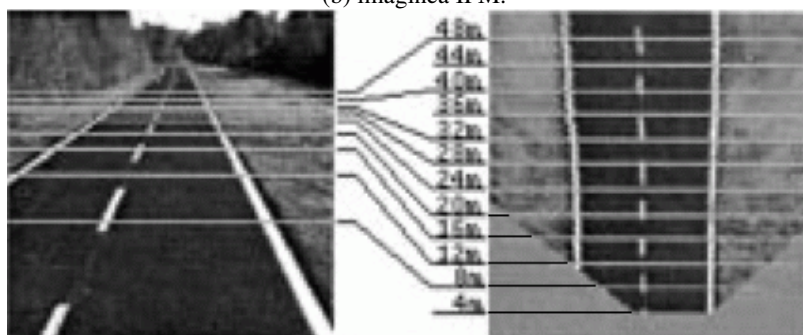


Fig. 5.4.3. Corespondența intervalelor de drum, în imaginea perspectivă și în imaginea IPM.

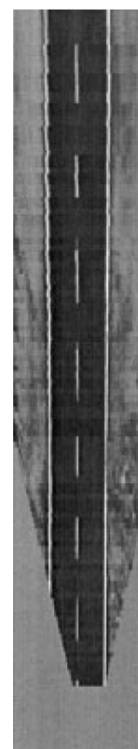


Fig. 5.4.2. Aspect 1:1.

Tehnica este potrivită atât pentru detecția benzii de circulație, cât și pentru detecția obstacolelor, atât în sisteme mono cât și sisteme de stereo viziune. În cazul stereo, a avut succes în anii '90, fiind implementată în proiectul italian ARGO [Bertozzi97].

În cazul sistemelor monoculare, detecția benzii de circulație este făcută prin localizarea marcajelor. Acestea sunt depistate ca linii luminoase cvazi-verticale, de lățime constantă și înconjurată de regiuni mai închise [Broggi95]. Orientarea și poziția ego-car față de banda de circulație sunt ușor de calculat din parametrii ecuațiilor care descriu liniile marcajelor.

Pentru detecția obstacolelor aflate în mișcare, se face o transformare a imaginii IPM de la momentul  $t + \delta$  peste imaginea IPM de la momentul  $t$ , pe baza parametrilor cinematicii ego-car (viteza și yaw-rate-ul – obținute de la sensori specializați) și timpul scurs între cele două imagini ( $\delta$ ). Astfel, regiunile comune ale drumului se suprapun. Prin calcularea diferenței celor două imagini, se localizează obstacolele aflate în mișcare. Tehnica se numește Temporal-IPM. Obstacolele statice nu sunt evidențiate astfel, decât dacă se află la mică profunzime, după cum propune [Bertozzi05], detaliat în secțiunea următoare. În [Dănescu14], se propune o metodă care detectează atât obstacolele în mișcare, cât și pe cele statice, folosind imagini IPM individuale.

Tehnica presupune că drumul este plan, iar caroseria ego-car nu are oscilații. Atunci când aceste presupuneri nu sunt adevărate, efectul asupra detecției benzii de circulație este minor, dar pentru detecția obstacolelor, situația este diferită:

- o curbă verticală ascendentă a drumului va muta obstacolul câțiva pixeli în susul imaginii, care conduce la deplasări serioase în imaginea IPM, iar estimarea distanței suferă erori mari. Pentru profunzimi mici drumul poate fi considerat plan, fără să apară probleme;
- în același mod, oscilațiile caroseriei conduc la oscilații ale obstacolelor în susul și în josul imaginii, ducând din nou la variații serioase în imaginea IPM, confuzând detecția când profunzimea este mai mare și când sunt mai multe obstacole. Mișcarea în planul imaginii perspective este aceeași indiferent de profunzimea la care se află obstacolele, dar în imaginea IPM deplasările cresc exponențial cu profunzimea.

Prin adoptarea celor două presupuneri, tehnica IPM pentru un sistem de mono viziune este limitată la distanțe mici. În cele ce urmează, va fi prezentat mai în detaliu un astfel de sistem.

În [Bertozzi05], se propune o soluție pentru detecția obstacolelor aflate la profunzime mică și când ego-car se deplasează cu viteză mică. Un astfel de sistem ar fi ieftin și ar putea fi folosit pentru mașinile mari, atunci când efectuează manevre mai simple sau mai complexe în spații mici (parcări), și când șoferului îi este mai greu să supravegheze vecinătatea mașinii. De exemplu, i-ar fi greu să observe un copil care trece chiar prin fața mașinii.

Când ego-car se mișcă cu viteză mică, două imagini succesive IPM sunt scăzute, ținând cont de roto-translația corespunzătoare cinematicii ego-car; proces care suprapune aceleași porțiuni ale drumului în cele două imagini. Imaginea diferență pune în evidență atât obstacolele aflate în mișcare cât și obstacolele staționare:

- cele în mișcare, pentru că își schimbă poziția pe drum, nerespectând mișcarea aparentă așteptată a drumului;
- și cele staționare, pentru că aparența lor în imaginile IPM se schimbă diferit de aparența drumului (Fig. 5.4.4). Pentru profunzimi mai mari, obstacolele staționare nu pot fi evidențiate pentru că aparența lor nu se schimbă semnificativ de cea a drumului, în imaginile IPM.

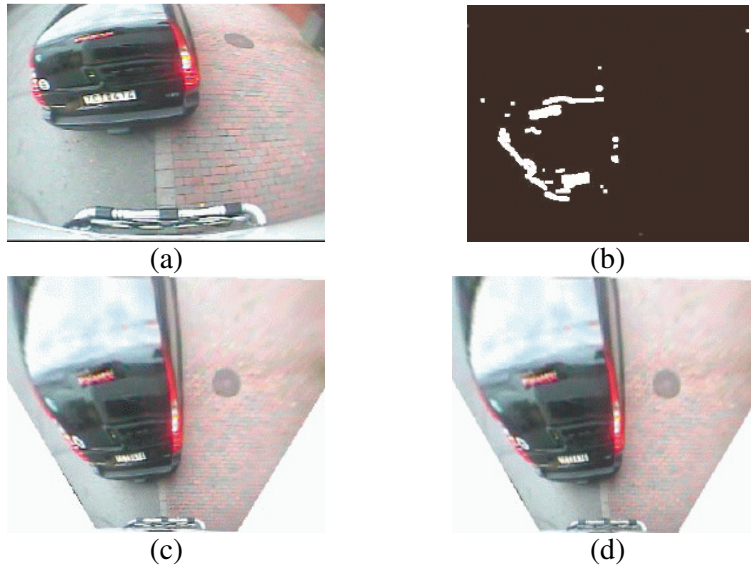


Fig. 5.4.4. Temporal-IPM (a) imaginea perspectivă curentă (camera fish-eye), (b) imaginea binarizată a diferenței dintre: (c) imaginea IPM curentă și (d) imaginea IPM roto-translatată anterioară.

#### 5.4.2 Diferența imaginilor perspective succesive

Tot în [Bertozzi05], se propune o abordare pentru cazul în care ego-car nu se mișcă. În acest caz, algoritmul este asemănător cu cei din domeniul camerelor de supraveghere (surveillance), detectând obstacolele aflate în mișcare și lucrând doar cu imaginile perspective. Când scena nu se schimbă deloc, se obține o imagine de fundal de referință (Fig. 5.4.5.b). Pentru îmbunătățirea sa, se mediază mai multe imagini succesive, reducând astfel zgomotele. Zgomotele au două surse: vibrațiile caroseriei și zgomotul semnalului video. Ulterior, imaginile achiziționate (Fig. 5.4.5.a) sunt scăzute din imaginea de referință. Valorile non-zero (Fig. 5.4.5.c) sunt îmbunătățite prin filtre morfologice (eroziune, dilatare) și sunt binarizate (Fig. 5.4.5.d). În imaginea binară, pixelii albi învecinați sunt grupați, obținând obstacolele în mișcare. Chiar dacă un obstacol real este detectat prin mai multe astfel de grupuri, pentru sistemul care avertizează șoferul (sau chiar acționează automat frâna) este suficientă semnalarea zonelor în care sunt obstacole în mișcare.

Dacă s-ar pune problema unei înțelegeri mai coerente a scenei, câteva limitări ar fi inerente:

- zgomotul nu poate fi definit foarte bine, și respectiv nu poate fi eliminat întotdeauna;
- calitatea imaginii binarizate este limitată;
- o grupare corectă este greu de obținut: în cazul exemplului prezentat, pietonul este detectat ca două obstacole (dar o urmărire în mai multe cadre ar putea ajuta);
- metoda nu este fezabilă la distanțe mai mari și în cazul mai multor obstacole în mișcare;
- la viteze mai mari ar trebui o frecvență mai mare de achiziție a imaginilor.

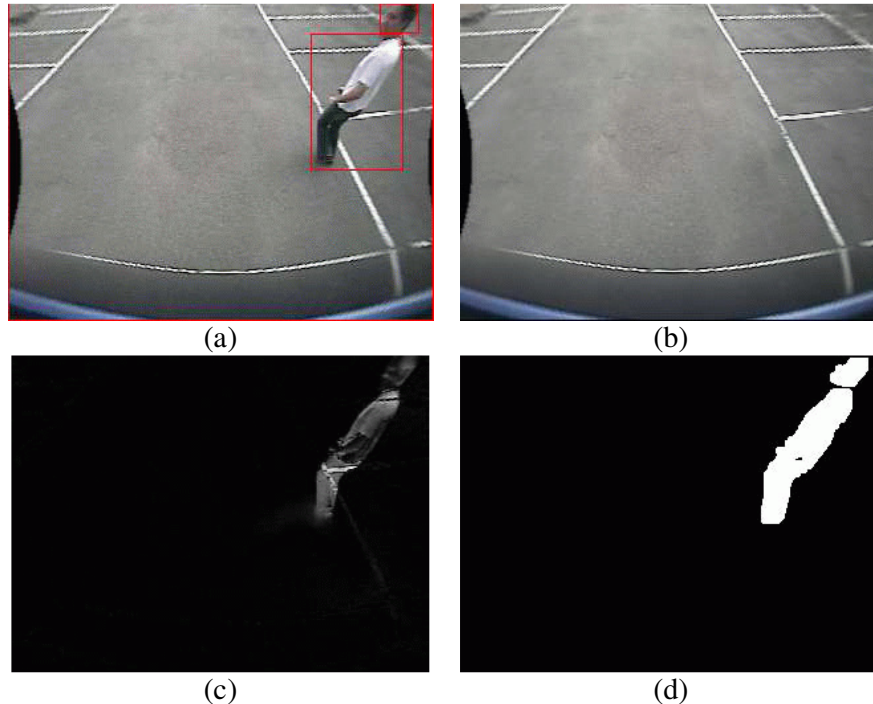


Fig. 5.4.5. (a) Imaginea curentă, inclusiv detecția, (b) imaginea de referință, de fundal, (c) imaginea diferență și (d) imaginea binarizată după filtrarea morfologică.

### 5.4.3 Detecția mișcării în imagini IPM

O altă abordare este aceea a urmării de trăsături ale obstacolelor în imagini IPM succesive. Un exemplu în acest sens este [Techmer04], în care se urmăresc muchii ale imaginii IPM. Căutarea muchiilor dintr-o imagine în imaginea următoare se face cu ajutorul Distance Transform – DT (principiul e prezentat în [Gavrila98]). Astfel, suprapunerea muchiilor din imaginea curentă în diferite poziții peste DT al imaginii anterioare, conduce la determinarea unei măsuri a potrivirii. Poziția corespunzătoare celei mai bune potriviri va fi cea care arată locul unde s-a deplasat obiectul (prin “obiect” se înțelege aici orice element al scenei, mai exact poate fi un marcaj de pe drum sau un obstacol – deasupra drumului). Astfel, fiecărei muchii  $i$  se atribuie un vector de mișcare. Muchiile, care sunt vecine și ale căror vectori de mișcare au aproximativ același sens și aceeași magnitudine, sunt grupate într-un obiect. Astfel, sunt găsite toate obiectele din scenă. Obiectele sunt încadrate în dreptunghiuri. În funcție de direcția de deplasare, relativă la ego-car, obiectele sunt clasificate în trei categorii (Fig. 5.4.6): care se îndepărtează, care se apropie, și staționare. Evident, un obiect “staționar” se mișcă la fel ca și ego-car. Obiectele care se apropie sunt cele de maxim interes fiind cele care s-ar putea ciocni cu ego-car.

Câteva rezultate sunt exemplificate în Fig. 5.4.7. Desigur, obiectele reale, care sunt apropiate unele de altele și care au aceeași direcție de deplasare, pot fi cu ușurință grupate într-un singur obiect detectat (Fig. 5.4.7.c – toate muchiile staționare au fost grupate într-un singur obiect mare). De asemenea, muchiile care sunt greșit potrivite conduc la semnalarea unor obiecte false (Fig. 5.4.7.a, b, și c – o parte din muchiile de pe obiectul din dreapta sus sunt clasificate ca făcând parte dintr-un fals obiect staționar).



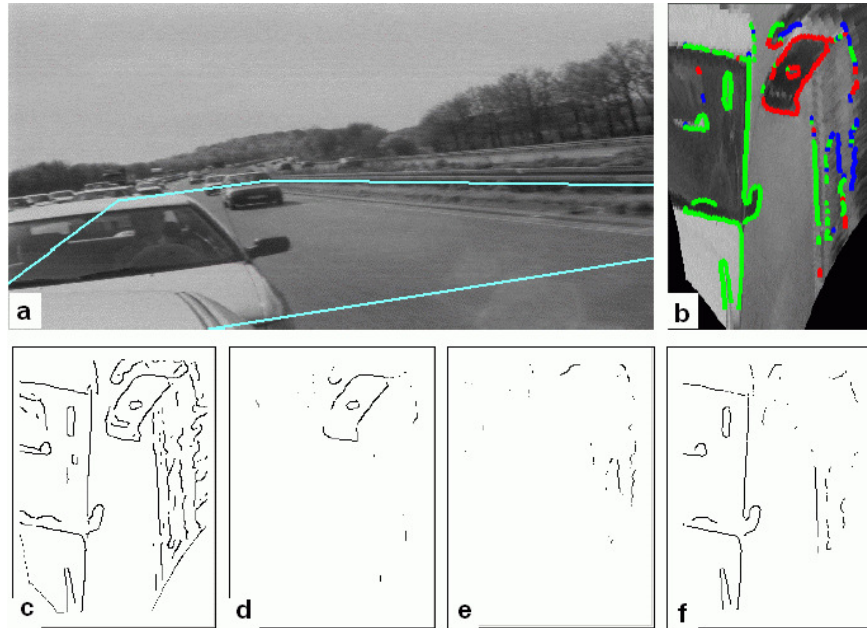


Fig. 5.4.6. Clasificarea mișcării obiectelor: (a) imaginea perspectivă, (b) imaginea IPM cu punctele de muchie clasificate (roșii – se apropie, albastre – se îndepărtează, verzi – staționare), (c) imaginea de muchii, (d) muchiile care se apropie, (e) muchiile care se îndepărtează, și (f) muchiile staționare.

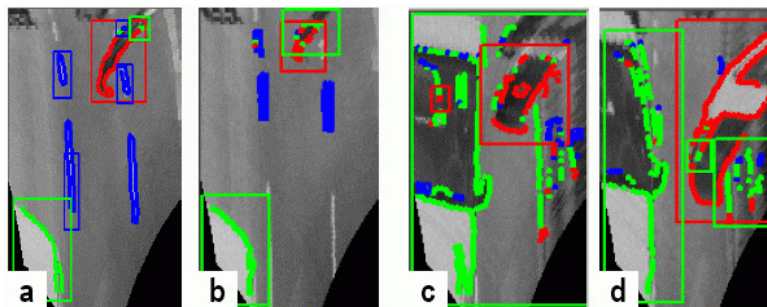


Fig. 5.4.7. Câteva exemple de detecție și clasificare a obiectelor. În imaginile (c) și (d) sunt observabile greșeli.

Limitări:

- pentru muchiile care sunt cvazi-drepte, paralele cu direcția de deplasare și care se întind pe toată imaginea (cum ar fi marcajele continue), potrivirea dintr-o imagine în următoarea se poate face la fel de bine în mai multe poziții, deci ar fi ambiguu. Un astfel de exemplu este prezent în Fig. 5.4.6 b și f;
- o grupare corectă a muchiilor în obiecte este limitată la cazuri simple.

#### 5.4.4 Detecția mișcării în imagini perspective

Mișcarea aparentă în spațiul imaginii se datorează compunerii a trei fenomene:

- mișcarea ego-car,
- mișcarea altor obstacole în spațiul scenei,
- proiecția perspectivă.

Dintre acestea, fenomenul proiecției perspective este cel care complică cel mai mult situația, deformând mișcarea reală în moduri diferite pentru componenta longitudinală și pentru componentele laterală/verticală.

Pentru calcularea fluxului optic dintr-o imagine (Fig. 5.4.8) în imaginea următoare (Fig. 5.4.10), există algoritmi specifici, împărțiți în două clase:

- pentru flux optic rar (Fig. 5.4.9): calculează vectorii de mișcare a unui număr redus de pixeli, aleși pe criteriul unei probabilități mari de a fi unici (de exemplu pixeli aflați la intersecții de muchii – colțuri);
- pentru flux optic dens (Fig. 5.4.11): încearcă să calculeze vectorii de mișcare pentru toți pixelii din imagine. Timpul de execuție este mai mare, iar afișarea orientării și magnitudinii vectorilor se impune a fi limitată la encodarea acestora prin coduri de culori.



Fig. 5.4.8. Imagine la momentul  $t$



Fig. 5.4.9. Flux optic rar



Fig. 5.4.10. Imagine la momentul  $t+1$



Fig. 5.4.11. Flux optic dens

Din cauza complexității și a varietății abordărilor care detectează obstacole din informația oferită de mișcarea în imagini perspectivă, studiul de față nu își propune să trateze aceste abordări.



## 5.5. Detecția din imagini stereo

Măsurători 3D se pot face și folosind o singură cameră aflată în mișcare (structure from motion), în fond făcându-se și determinarea poziției camerei față de poziția sa în cadrul anterior (se determină parametrii de mișcare), simulându-se astfel principiile stereo viziunii. Principalele dezavantaje sunt: determinarea structurii scenei simultan cu determinarea parametrilor de mișcare și faptul că nu se poate determina scala scenei, fiind necesare indicii suplimentare (de exemplu înălțimea camerei față de planul solului).

Dar prin folosirea a două camere distincte, fixate rigid între ele, se obține un sistem real de stereo viziune, având avantaje net superioare sistemelor de mono viziune:

- poziția unei camere este întotdeauna aceeași în raport cu cealaltă și poate fi predeterminată cu precizie, folosind metode specifice de calibrare;
- poate determina structura scenei chiar și în lipsa mișcării;
- structura scenei e măsurabilă cu precizie, deoarece calibrarea sistemului restrânge mult incertitudinea corespondențelor dintre cele două imagini.

Având disponibilă informația de profunzime, este tentantă și apelarea la informații de mișcare, în cadre succesive (optical flow) [Franke05]. Totuși, aceasta crește complexitatea procesărilor. Astfel, în primul rând, este necesară o bună înțelegere a procesărilor la nivel de cadre independente (single frame).

Unele abordări mizează pe anumite particularități ale scenei. De exemplu, [Bertozzi97A] face asumția unei suprafețe plane a solului, astfel că diferența imaginilor IPM (Inverse Perspective Mapping) pune în evidență localizarea obstacolelor: Fig. 5.5.1. În [Bertozzi98A], autorii propun o rezolvare pentru cazul suprafeței neplanare a drumului, făcând observația că în acest caz marcajele benzilor de circulație nu sunt paralele în imaginea IPM. Astfel că, profilul vertical al drumului este dedus tocmai din acest neparalelism.

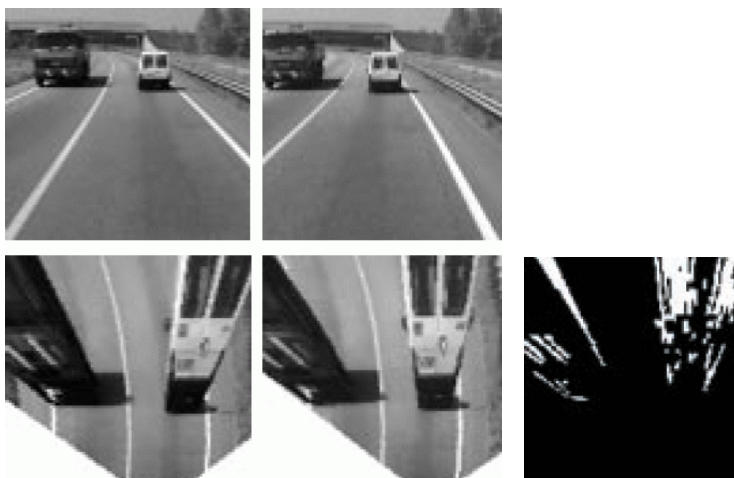


Fig. 5.5.1. Imaginile inițiale stânga/dreapta, imaginile IPM stânga/dreapta și diferența imaginilor IPM

Tehnica IPM transformă imaginile stânga/dreapta într-un spațiu comun în care compară cele două imagini. În [Kubota07], spațiul comun folosit este cel al imaginii stângi, astfel că doar o singură imagine este transformată, cea dreaptă, în spațiul imaginii stângi. Tehnica este cunoscută sub numele de *warping*. Ca și la IPM, este necesar profilul drumului, acesta fiind presupus plan și cu o înclinare dată (de exemplu prin calibrare).

[Williamson99] folosește aceeași presupunere pentru a face o reconstrucție rapidă a solului, restrângând foarte mult spațiul disparităților.

În aceeași secțiune, vor fi prezentate și analizate abordări care detectează obstacole generice, în scene cu sol cu profil generic, folosind un singur cadru stereo.

Detectia de obstacole poate avea părți comune sau similare cu detectia solului. De aceea, vor fi prezentate pe scurt și modurile în care e detectat solul, ca parte a separării sol-obstacole. Unele abordări detectează întreaga suprafață vizibilă a solului ([Oniga10]), iar altele doar limita spațiului liber, adică frontiera dintre suprafața solului și începutul obstacolelor ([Badino07]).

Spațiul sursă al informațiilor 3D este imaginea perspectivă îmbogățită prin stereo viziune cu profunzimi exprimate prin disparități, numită hartă de disparități, cunoscută și sub numele de spațiu U-V-disparity. În [Hu05], coordonatele U și V sunt definite relativ la centrul optic al imaginii. Rezoluția coordonatelor U și V poate diferi de cea a imaginii: de exemplu  $1U = 2$  pixeli, pentru a asigura o compresie a datelor. Adesea se folosesc histogramele U-disparity și V-disparity, care acumulează pixelii cu aceleași coordonate (U, disparity), respectiv (V, disparity).

Din harta de disparități se pot obține puncte 3D reprezentabile într-un spațiu polar sau cartezian. Diferite abordări folosesc, cel mai adesea, spații derivate din spațiul disparităților sau din spațiul 3D.

Sistemul de coordonate al hărții de disparități are caracter polar pe lățime și înălțime, conform geometriei proiective a imaginii, cu axele U, respectiv V, iar profunzimea este reprezentată prin disparități. Sistemul de coordonate cartezian are axele X și Y paralele cu axele U și V ale imaginii, iar axa Z reprezintă profunzimea (unele abordări denumesc altfel axele de coordonate).

De regulă, algoritmi sunt strâns legați de un spațiu de procesare preponderent, dar folosesc simultan și alte spații:

- pentru a face raționamente în spațiul cartezian al scenei, în timp ce se ține cont de capacitatea de percepere din spațiul nativ; sau vice-versa;
- pentru a folosi informații care s-au pierdut o dată cu obținerea unor spații particulare, prin scăderea dimensionalității.

În figura următoare este prezentată o diagramă a spațiilor de procesare:

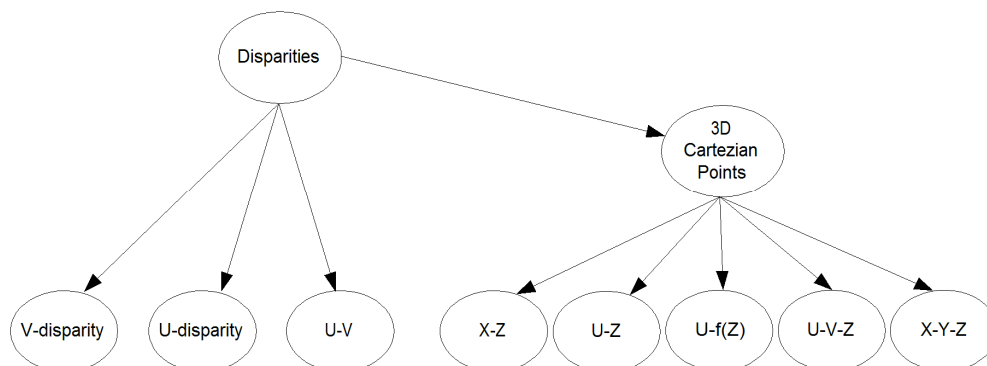


Fig. 5.5.2. Spații de procesare

Din punct de vedere cronologic, mai întâi s-a folosit profunzimea exprimată prin disparități, aceste valori fiind obținute prin însuși procesul de stereo matching. Mai apoi, s-a trecut la folosirea punctelor carteziene 3D (exprimate metric), obținute din harta de disparități prin reconstrucție stereo 3D.

În continuare, vor fi prezentate abordări existente, grupate după profunzimea folosită și după spațiul de procesare folosit preponderent la detectarea obstacolelor. Schematic, acestea sunt enumerate în tabelul de mai jos. Pentru fiecare abordare vor fi prezentate: organizația (companie/universitate/grup de cercetare), echipa (1-2 membri), anul publicării (dacă sunt mai multe articole, atunci se alege unul dintre cele mai semnificative, amintit la secțiunea de referințe), iar modul de detecție a solului și a obstacolelor va fi indicat prin următoarele simboluri:












↑ = detecția solului se limitează la detecția spațiului liber, adică a spațiului din fața ego-car și până la frontiera începutului obstacolelor;

▲ = se determină un profil vertical 3D al solului (eventual și unul orizontal al benzii de circulație);

□ = detecția de obstacole se limitează la evidențierea zonelor care conțin obstacole;

■ = se detectează obstacolele individuale și eventual se reprezintă prin modele de nivel înalt (de exemplu prin cuboide).

**Observații despre tabel.** Tabelul e limitat ca bogăție de informații, detalii fiind prezentate în subsecțiunile următoare și în articolele citate. Ordinea spațiilor de procesare sugerează cât de compacte sunt datele, de la foarte compacte (prima coloană) la foarte puțin compacte (ultima coloană). Ordinea abordărilor încearcă să urmărească ordinea spațiului de procesare folosit pentru detecția de obstacole, primele abordări folosind cele mai compacte spații de procesare. Pe lângă acele simboluri, pentru a fi mai sugestiv, pot fi indicate și anumite cuvinte cheie (de regulă modelul folosit).

Spațiul de procesare preponderent		Spațiul disparităților			Spațiul Cartezian 3D				
		V-disparity	U-disparity	U-V	U-f(Z)	U-Z	U-V(+z)	X-Z	X-Y-Z
Abordări									
Labayrade '02		▲ Flat-road ■ Rectangles							
Hu '05			▲ Flat-road ■ Polygons						
Broggi '05				□ Point sets					
Caraffi '07				□ Point sets					
Nedevschi '04, Pocol '08					■ Cuboids				▲ Clothoid
Badino '07						↑□ Frontier			
Manduchi '04, Mark '07							↑□ Point sets		
Badino, Pfeiffer '07, '09							□ Stixels		▲ B-Spline
Gehrig '04								■ Rectangles	
Oniga '10, Vatavu '11								▲ Quadratic ■ Polylines	
Broggi '13									Flat-road ■ Voxels

## 5.5.1 Profunzime exprimată prin disparități

### Abordări în spațiul histogramei V-disparity

La începutul anilor 2000, Raphael Labayrade a investigat posibilitățile de detecție oferite de spațiul compact al histogramei V-disparity [Labayrade02], în care fiecare celulă numără punctele care au aceleași coordonate (V, disparity). În această matrice liniile corespund liniilor imaginii (V), iar coloanele corespund planelor verticale cu disparitate constantă, înșirate de la disparitate mică (profunzime mare) la disparitate mare (profunzime mică): Fig. 5.5.3.a.

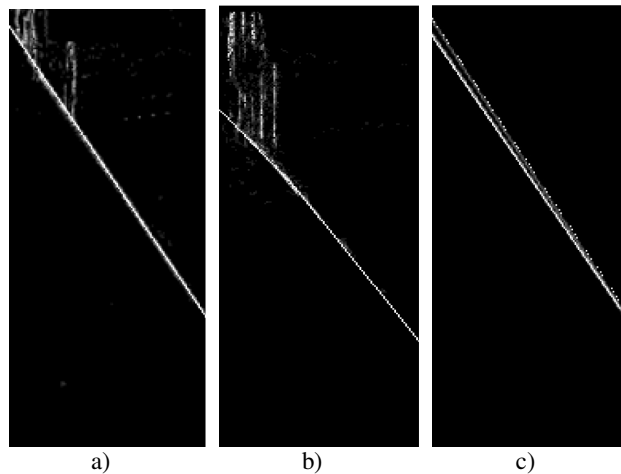


Fig. 5.5.3. Spațiul V-disparity.



Fig. 5.5.4. Delimitarea obstacolelor în spațiul imaginii.

Linia albă lungă și oblică pune în evidență suprafața solului. În cazul unui sol planar, linia solului este o dreaptă și este detectată folosind transformata Hough. În cazul unui sol neplanar, acesta este împărțit în părți cvazi-planare (Fig. 5.5.3.b). Orizontul este localizat la capătul liniei solului.

Obstacolele sunt evidențiate prin linii verticale. Acestea sunt ușor de identificat într-o histogramă unidimensională care numără punctele din planele verticale cu disparitate constantă. Această histogramă este ușor de obținut din histograma V-disparity, iar maximele locale ale ei evidențiază obstacolele.

Dreptunghiul care înscrie un obstacol în spațiul imaginii poate fi determinat (Fig. 5.5.4). Partea de jos, contactul cu solul, este dată de coordonata V a liniei solului, la disparitatea obstacolului curent, în spațiul V-disparity. Partea de sus poate fi determinată în matricea V-disparity la colona obstacolului curent: se determină ultima celulă cu densitate considerabilă, pornind de la linia solului în sus. Limitele stânga/dreapta pot fi determinate printr-o analiză a

distribuției laterale a punctelor care au disparitatea obstacolului curent și sunt între limitele de sus și de jos ale dreptunghiului. Mai exact, această analiză se face în spațiul U-disparity, care este o vedere de sus (view from above) a spațiului U-V-disparity. Totodată, se pot diferenția obstacolele multiple aflate la aceeași profunzime.

În Fig. 5.5.3.c, este arătat spațiul V-disparity cu liniile solului din două cadre diferite, arătând influența oscilațiilor caroseriei ego-car și importanța calculării unghiului pitch și înălțimii camerei față de sol, pentru separarea sol-obstacole, fapt ignorat în alte lucrări, precum [Broggi13].

În cazul în care solul are și un unghi roll, e nevoie de procesări suplimentare, prezentate în [Labayrade03]. O limitare serioasă a acestei abordări apare atunci când obstacolele nu prezintă o față paralelă cu planul imaginii și se întind pe mai multe disparități: în matricea V-disparity vor fi difuze.

### Abordări în spațiul histogramei U-disparity

Așa cum în [Labayrade02] se detectează suprafețe planare particulare în spațiul de acumulare V-disparity, în [Hu05] se extinde ideea și pentru suprafețe plane particulare detectabile în spațiul histogramei U-disparity. Totuși, suprafețele plane cu orientări neparticulare rămân nedetectabile.

Detectarea și etichetarea liniilor drepte au loc independent în histogramele V-disparity (Fig. 5.5.5) și U-disparity (Fig. 5.5.6). În spațiul imaginii, se construiesc grupuri de pixeli. Toți pixelii dintr-un grup au aceeași etichetă în histograma V-disparity și aceeași etichetă în histograma U-disparity. Se acceptă și cazul în care toți pixelii din grup au etichetă doar în unul din spații. Fiecare grup e încadrat într-un poligon (Fig. 5.5.7), iar suprafețele corespunzătoare acestor grupuri se împart în trei clase:

- orizontale (sol, tavan – albastre); au etichetă doar în histograma V-disparity;
- verticale orientate spre cameră (obstacole – rosii); au etichete în ambele histograme;
- verticale cu alte orientări (structuri de pe marginea drumului – galbene); au etichetă doar în histograma U-disparity.

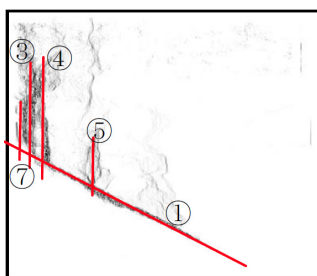


Fig. 5.5.5. Spațiul histogramei V-disparity.

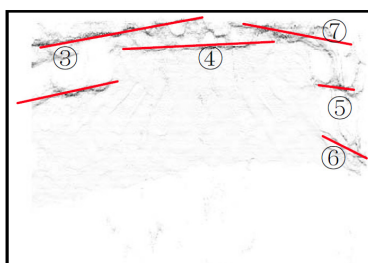


Fig. 5.5.6. Spațiul histogramei U-disparity.

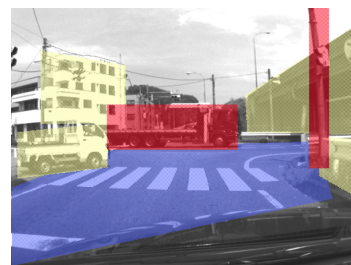


Fig. 5.5.7. Spațiul U-V.

Autorii folosesc formule care înglobează și unghiul pitch al camerei, pentru cazurile când acesta este pronunțat. Nu se face gruparea suprafețelor plane în obstacole individuale, deși ar fi important în scenariile de trafic și mai puțin important/aplicabil în scenariile indoor.

## Abordări în spațiul U-V

În [Broggi05], detecția solului se face în spațiul V-disparity, dar e posibilă doar când profilul transversal e drept (în ciuda propunerii de la sfârșitul articolului).

Evidențierea obstacolelor se face în spațiul U-V, prin agregarea pixelilor cu disparități similare (Fig. 5.5.8), fiind practic o evidențiere a suprafețelor verticale. Nu sunt detectate obstacolele individuale, ci doar o hartă polară a prezenței acestora.



Fig. 5.5.8. Prezența obstacolelor (culoarea codifică profunzimea).

\*\*\*

În [Caraffi07], pe fiecare coloană, se determină disparitatea care apare mai des, obținându-se câte o fâșie de pixeli care au această disparitate (Fig. 5.5.9). Diferite filtre păstrează doar fâșiile care au proprietăți specifice obstacolelor. Criteriile sunt: pixelii din fâșie să fie compacți, fără prea multe goluri; fâșiile vecine să aibă disparități apropiate cu fâșia curentă – vecinătatea este în funcție de profunzime, mai îngustă pentru profunzime mai mare; panta să fie semnificativ mai mare decât panta locală a solului (determinată în spațiul V-disparity); înălțimea metrică să fie peste un prag. După cum se poate observa, spațiul folosit preponderent este cel al disparităților, dar se apelează și la metrici carteziene. Rezultatul e prezentat în Fig. 5.5.10.

Dacă sunt mai multe obstacole pe o coloană, doar cel care apare mai mare în imagine va fi considerat. De asemenea, nici obstacolele înguste și înclinate lateral nu vor fi detectate. Nu sunt detectate obstacolele individuale, ci doar o hartă polară a prezenței obstacolelor.

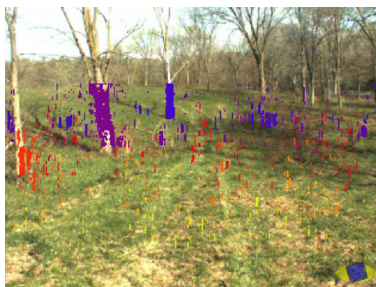


Fig. 5.5.9. Fâșiile cu disparitatea preponderentă.



Fig. 5.5.10. Rezultate finale, după aplicarea filtrelor.



## 5.5.2 Profunzime exprimată cartezian

### Abordări în spațiul X-Z

În abordările mai vechi ale echipei Daimler [Daimler], suprafața solului este presupusă a fi plană și orizontală. Astfel, în [Gehrig04] punctele 3D de interes pentru detecția de obstacole sunt cele care au înălțimea între 20 cm și 2 m deasupra solului și sunt într-un dreptunghi de 30x50 m în fața ego-car. Spațiul cartezian top-view este împărțit în celule de 30x30 cm. Practic, se creează o histogramă bidimensională, în care fiecare celulă numără punctele 3D pe care le cuprinde, iar coordonata înălțimii se pierde. Celulele care au un număr semnificativ de puncte sunt grupate. Aceasta se face cu un algoritm de analiză a componentelor conexe, căruia i se specifică și distanța maximă dintre două celule conectabile.

Obstacolele astfel obținute sunt modelate prin dreptunghiuri care le înscriu, atât în spațiul cartezian top-view (Fig. 5.5.11), cât și în spațiul imaginii (Fig. 5.5.12). Se observă că uneori mai multe obstacole reale sunt grupate împreună.

Autorii nu amintesc de faptul că densitatea punctelor 3D ale obstacolelor scade mai mult decât pătratic cu distanța și nici despre cum aleg pragul minim de puncte dintr-o celulă validă. De asemenea, nu specifică nici modul de alegere a distanței maxime dintre două celule conectabile, o distanță prea mare putând duce la o unire a obstacolelor apropiate care nu sunt departe de ego-car, iar o distanță prea mică la o fragmentare a obstacolelor mari care sunt departe.

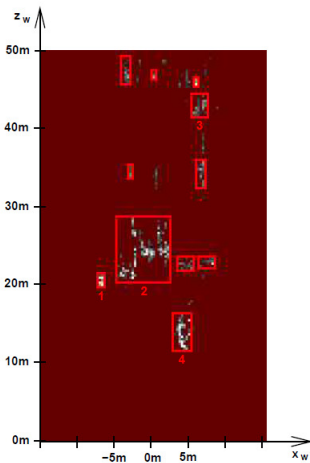


Fig. 5.5.11. Grupare în spațiul cartezian top-view.

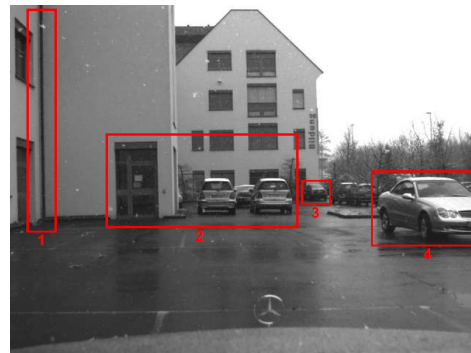


Fig. 5.5.12. Obstacolele în spațiul imaginii, prin reconsiderarea coordonatei înălțimii (obstacolele de fundal nu sunt arătate).

\*\*\*

Echipea de la Universitatea Tehnică din Cluj-Napoca [IPPRG] are o abordare bazată pe hărți de înălțime, care separă structurile scenei în trei clase: sol, platforme joase (obstacole puțin înalte – minim 5 cm) și obstacole. Abordarea este prezentată începând cu [Oniga07] și culminând cu [Oniga10]. Spațiul cartezian top-view este împărțit în celule de 10x10 cm, care au asociate informații precum densitatea de puncte și înălțimea medie a punctelor. Printr-un algoritm elaborat, se determină o suprafață matematică parametrizată (cuadratică:  $Y = -a \cdot X - a' \cdot X^2 - b \cdot Z - b' \cdot Z^2 - c$ ), care modelează cât mai fidel suprafața solului. Astfel, pe lângă

evidențierea localizării obstacolelor obișnuite, pot fi evidențiate separat și obstacolele joase, cum ar fi trotuarele, delimitate de borduri (Fig. 5.5.13).

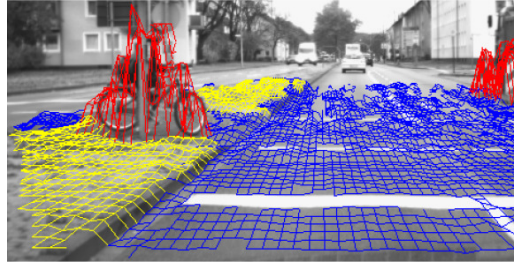


Fig. 5.5.13. Sol, platforme joase și obstacole.

Celulele care aparțin obstacolelor sunt grupate printr-un algoritm simplu de etichetare folosind vecinătăți de 3x3, cu riscul supra-fragmentării [Vatavu11]. Sunt tratate doar obstacolele de prim-plan. Contururile poligonale ale frontierei vizibile ale obstacolelor (Fig. 5.5.14) sunt determinate printr-o scanare radială din poziția ego-car folosind un pas unghiular adaptabil cu distanța (pentru a nu rata obstacole mici aflate la distanțe mari). Pentru a lua în considerare și obstacolele aflate în spatele obstacolelor joase, se folosesc două treceri de acest gen.

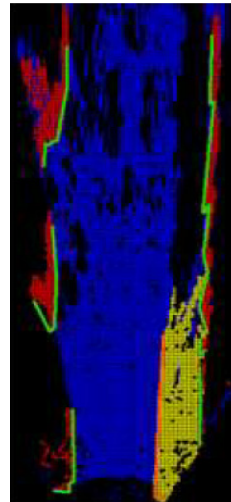


Fig. 5.5.14. Descriere poligonală.

Abordarea are avantajul modelării oricărei forme, într-un mod compact. Dezavantajul este că implementarea devine laborioasă, pentru că lucrează în spațiul cartezian, iar un spațiu polar ar fi simplificat mult scanarea radială.

### Abordări în spațiul U-f(Z)

O altă abordare a grupului de la Universitatea Tehnică din Cluj-Napoca detectează solul pe baza benzii de circulație și obstacolele prin etichetare în spațiul polar U-f(Z).

Banda de circulație se detectează în spațiul cartezian 3D [Nedevschi04A]. În prima fază, se detectează unghiul de pitch folosind punctele 3D din fața ego-car, unde suprafața solului este cvazi-plană. Apoi se detectează ceilalți parametri ai suprafeței benzii de circulație, descrisă printr-un model clotoidal, folosindu-se indicii ale marcajelor desenate pe sol sau alte delimitatoare: Fig. 5.5.15.



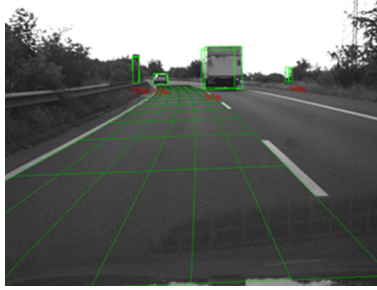


Fig. 5.5.15. Detectia benzii de circulatie.

Pentru detectia de obstacole se folosesc doar punctele 3D dintr-un spatiu de interes aflat deasupra solului, dar nu mai sus de inaltimea ego-car: Fig. 5.5.16.

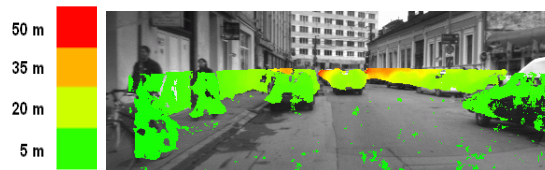


Fig. 5.5.16. Punctele 3D din spatiul de interes.

Similar cu abordarea din [Gehrig04], descrisa mai sus, intr-o serie de articole, incepand cu [Nedevschi04] si terminand cu [Pocol08], se creeaza o histograma a punctelor 3D vazute de sus (Fig. 5.5.17), dar este folosit un spatiu mixt, de tip U-f(Z), avand avantajul procesarii obstacolelor la o rezolutie care urmareste puterea de perceptie a sistemului stereo. Pe axa laterala, camerele percep scena intr-un mod polar, astfel ca se foloseste coordonata U. Pe axa profunzimii, spatiul Cartezian e impartit in intervale care surprind atat calitatea reconstructiei 3D cat si diverse aspecte ale consistentei scenei: suprafete reflectiv/transparente/omogene, suprafete cu diferite orientari etc. Mai exact, pentru fiecare interval se foloseste o lungime proportionala cu distanta:  $IntervalLength(Z) = k * Z$  (k e ales empiric).

Problema pragului, folosit pentru alegerea celulelor cu densitate consistenta a punctelor 3D, este simpla: densitatea tinde sa fie constanta, trebuind totusi o ajustare liniara cu distanta metrica, pentru ca inaltimea spatiului de interes scade, in spatiul imaginii, proportional cu distanta. Nici problema conectivitatii nu exista: punctele 3D ale unui obstacol se afla in celule invecinate.



Fig. 5.5.17. Histograma punctelor obstacolelor din Fig. 5.5.16.

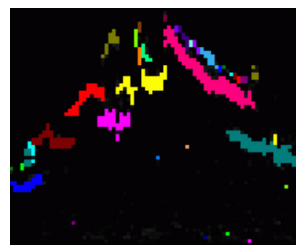


Fig. 5.5.18. Etichetarea obstacolelor din Fig. 5.5.16.

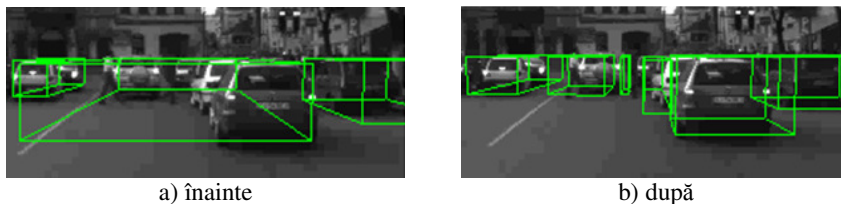
Gruparea celulelor in obstacole se face cu un algoritm de etichetare specializat (Fig. 5.5.18), care:

- grupeaza coloanele de celule ale fiecarui obstacol, de la stanga la dreapta;
- coloanele de celule sunt construite din celule consistente consecutive, prin permiterea unor goluri de cel mult echivalentul a 50 cm;

- lungimea unei coloane de celule este limitată pentru a evita conectarea greșită a obstacolelor aflate pe aceeași direcție optică;
- la gruparea coloanelor de celule se permit goluri laterale limitate și se impune o limită maximă a diferenței de profunzime.

În încercarea de a modela obstacolele prin cuboide cât mai fidele, urmează un șir de procesări:

- se face o rafinare a frontierei fiecărui obstacol, atât la nivel de coloane individuale de celule, cât și la nivelul întregului obstacol, diminuându-se astfel erorile de reconstrucție;
- obstacolele cu concavități sunt fragmentate, rezultând mai multe obstacole fără această problema (Fig. 5.5.19);
- o analiză a înfășurătorii poate evidenția orientarea obstacolului (Fig. 5.5.20);
- unele obstacole sunt departe de a avea o formă cuboidală. Și acestea sunt fragmentate (Fig. 5.5.21).



a) înainte

b) după

Fig. 5.5.19. Fragmentarea în obstacole fără concavități.

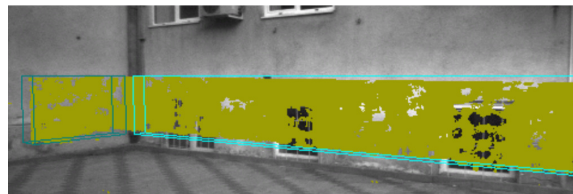


Fig. 5.5.20. Obstacole orientate.



a) înainte

b) după

Fig. 5.5.21. Fragmentarea în cuboide fidele.

## Abordări în spațiul U-Z

După mai mulți ani în care grupul de Image Understanding de la Daimler [Daimler] a implementat algoritmi în spațiul cartezian top-view, a urmat explorarea de algoritmi în spațiul U-Z, în care, se consideră că pentru fiecare direcție optică, structura scenei e compusă din: spațiu liber, obstacol de prim-plan și structuri de fundal. Abordarea din [Badino07] determină *frontiera* tuturor obstacolelor de prim-plan, analizând un grid al spațiului U-Z (Fig. 5.5.22). Această frontieră împarte spațiul în două: partea corespunzătoare spațiului liber (profunzimi mai mici) și partea fundalului din spatele obstacolelor de prim-plan (profunzimi mai mari). Vom numi această frontieră și *frontiera sol-obstacole*.

Frontiera constă din câte o celulă pe fiecare coloană (direcție optică), urmărindu-se un șir de celule de cost minim, de la stânga la dreapta. Fiecărei celule  $i$  se atribuie un cost care este invers proporțional cu densitatea, adică celulele cu densitate mai mare vor fi mai eligibile. În plus, pentru fiecare pereche de celule de pe coloane adiacente, se asociază un cost proporțional cu diferența de profunzime dintre cele două celule, adică se penalizează variațiile mari; totuși, pentru a permite diferențe reale mari de profunzime, acest cost este saturat la o valoare maximă (câțiva metri).

Folosind un algoritm de determinare a drumului de cost minim, de la stânga la dreapta spațiului polar, se determină frontiera optimă. Practic, se analizează un graf, în care nodurile sunt celulele spațiului U-Z, iar arcele sunt perechile de celule din coloane adiacente. Autorii folosesc algoritmi de programare dinamică, pentru că asigură găsirea soluției optime în cel mai scurt timp.

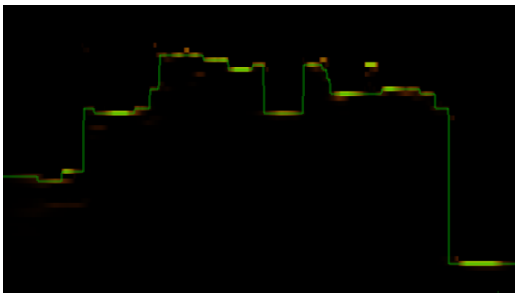


Fig. 5.5.22. Frontiera obstacolelor de prim-plan: grid în spațiul U-Z

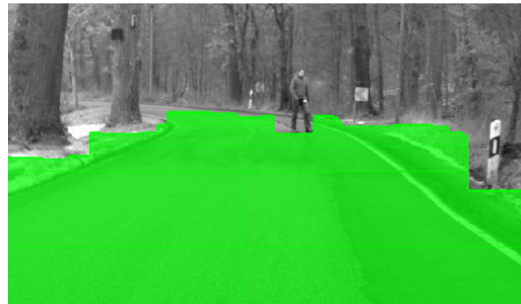


Fig. 5.5.23. Frontiera dintre spațiul liber și începutul obstacolelor de prim-plan: reprezentare în spațiul imaginii

Nu se ignoră nici avantajele folosirii datelor temporale, din cadrele anterioare. Astfel, densitatea punctelor din celule este acumulată în cadre succesive, ținând cont de cinematica ego-car. Deoarece cadrele succesive au diferite sisteme de coordonate polare, pentru procesul de acumulare se folosește un spațiu cartezian comun. Autorii prezintă fundamentele matematice ale acestor spații și ale transformărilor dintre ele, folosind atât măsurătorile 3D cât și varianța (gradul de încredere  $\alpha$ ) acestora. Astfel, pe lângă costurile celulelor și costurile diferențelor spațiale ale perechilor de celule, se folosesc și costuri ale diferențelor temporale, care penalizează diferențele față de frontiera optimă din cadrul anterior. Și în acest caz, pentru a permite mișcări reale ale obstacolelor, costul este saturat la o valoare maximă.

Fig. 5.5.23. arată frontiera în spațiul imaginii. Autorii nu-și propun identificarea obstacolelor individuale, decât în articole ulterioare.

### Abordări în spațiul U-V

Abordarea din [Manduchi04] este capabilă să trateze și scenarii off-road, în doi pași, separând mai întâi punctele 3D ale obstacolelor de cele ale solului, și identificând apoi obstacolele individuale.

Primul pas se aplică pentru fiecare pixel reconstruit (punct 3D). La început, toate punctele se consideră ca aparținând solului, urmând apoi să fie evidențiate cele care aparțin obstacolelor. În principiu, în jurul unui punct  $P_i$ , într-o vecinătate specială, se caută puncte  $P_j$ , care determină un segment cu panta mare împreună cu  $P_i$ . Dacă se găsesc astfel de puncte, fiecare pereche  $(P_i, P_j)$  indică prezența unui obstacol și se numesc puncte *compatibile*. Două puncte

sunt considerate compatibile indirect dacă sunt conectabile printr-un lanț de perechi de puncte compatibile. Vecinătatea folosită este reprezentată de două trunchiuri de con verticale, cu vârful în  $P_i$ , ca în Fig. 5.5.24.  $\theta_{\max}$  reprezintă panta maximă pe care poate urca ego-car ( $\theta_{\max} = 40^\circ$ ). Totuși, o pantă mare, dar care nu e mai înaltă de  $H_{\min}$ , poate fi abordată de ego-car, deci nu e considerată obstacol ( $H_{\min} = 20$  cm).  $H_{\max}$  este folosit pentru a evita unirea a două puncte care sunt vecine în spațiul imaginii, dar sunt depărtate în spațiul 3D ( $H_{\max} = 1$  m). Punctele dintr-o astfel de pereche de puncte au o relație simetrică, adică unul se află mai jos și unul mai sus, sau vice-versa, astfel că este suficientă căutarea în conul de sus.

În practică, trunchiurile conurilor 3D sunt transformate în trunchiuri de triunghiuri (trapeze) 2D în spațiul imaginii: Fig. 5.5.25. Autorii discută și posibilitatea de a avea conuri/triunghiuri înclinate, în cazurile în care ar fi necesar să se folosească un bias pentru unghiurile de pitch și roll, de exemplu când ego-car se află pe o suprafață neorizentală, unghiurile respective fiind măsurate cu dispozitive specifice (IMU). Timpul de căutare a perechilor de puncte compatibile este liniar cu numărul de puncte.

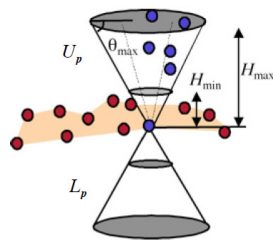


Fig. 5.5.24. Spațiul de căutare.

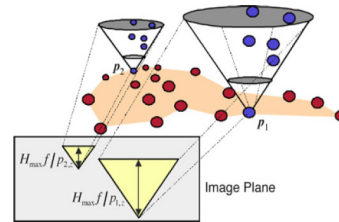


Fig. 5.5.25. Transformarea conurilor în triunghiuri.

În pasul al doilea, perechile de puncte compatibile sunt grupate în obstacole individuale. Pentru aceasta se construiește un graf: nodurile sunt punctele 3D, iar arcele sunt perechile de puncte compatibile. Astfel, un algoritm de identificare a subgrafurilor conexe determină obstacolele individuale (Fig. 5.5.26), care pot fi folosite pentru raționamente de nivel înalt. Un astfel de raționament este cel al rejectării obstacolelor mici (provenite probabil din puncte 3D reconstruite greșit), în urma analizei dimensiunilor în spațiul imaginii, dar și în spațiul 3D.

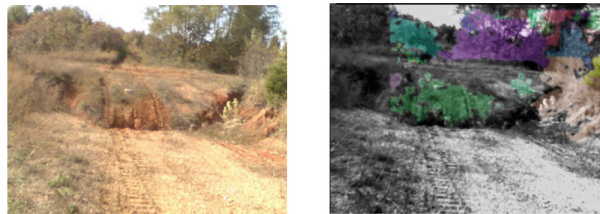


Fig. 5.5.26. Gruparea în obstacole individuale.

Un avantaj al abordării este că poate detecta și obstacole care prezintă zone nereconstruite, cu condiția ca toate punctele reconstruite să fie compatibile, direct sau indirect.

\*\*\*

Față de [Badino07] (vezi mai sus), în [Badino09] se aduc câteva îmbunătățiri mai importante:

- În prezența unui modul separat de detecție a solului [Wedel08], se folosesc doar punctele 3D aflate deasupra solului.

- Determinarea frontierei dintre spațiul liber și începutul obstacolelor de prim-plan are loc tot în spațiul U-Z. În faza de preprocesare, pentru eliminarea structurilor din fundal, de-a lungul fiecărei coloane, se păstrează doar prima celulă a cărei densitate este mai mare decât un prag.
- Urmează determinarea înălțimii obstacolelor, în spațiul imaginii. În acest sens, se introduce noțiunea de stixeli: un șir de dreptunghiuri verticale care se sprijină pe frontiera sol-obstacole și se înalță până la limita superioară a obstacolelor (Fig. 5.5.27).
- Pentru determinarea limitei superioare a obstacolelor, se face o analiză în spațiul imaginii (U-V). Ca și la determinarea frontierei sol-obstacole, și aici soluția optimă este determinată cu un algoritm de programare dinamică, pe bază de costuri (Fig. 5.5.28) pentru celulele U-V individuale și pentru perechile de celule de pe coloanele adiacente. Costul unei celule U-V reflectă diferența de profunzime față de frontieră, pe coloana curentă. Costul unei perechi de celule reflectă diferența de înălțime dintre cele două celule. În practică, autorii folosesc versiuni mai elaborate ale acestor costuri, combinând metrici carteziene cu metrici native sensorului de stereo viziune.

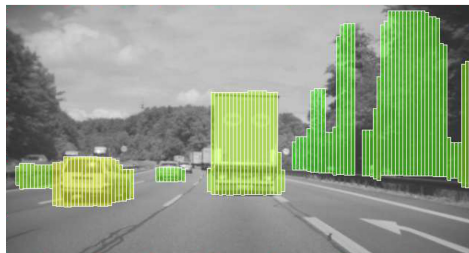


Fig. 5.5.27. Obstacolele delimitate de stixeli.

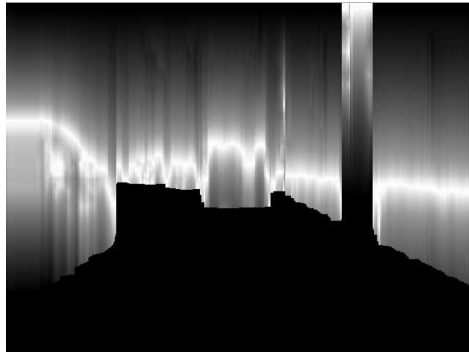


Fig. 5.5.28. Harta costurilor pentru segmentarea dintre prim-plan și fundal; costurile mai mici sunt desenate mai alb (scena diferă de cea din Fig. 5.5.27!).

În [Pfeiffer10], față de [Badino07] și [Badino09], se aduc următoarele îmbunătățiri mai importante:

- Frontiera sol-obstacole se determină acum în spațiul U-V, fiind mai compact decât spațiul U-Z, mai ales pentru profunzimi mai mari, dar și pentru că e spațiul care surprinde foarte bine ocluziile. În fond, procedura este aceeași: folosind un algoritm de programare dinamică, se determină drumul optim de celule, de la stânga la dreapta, folosind costuri asociate celulelor și costuri asociate perechilor de celule din coloanele adiacente.
- Se introduc stixelii dinamici, obținuți prin trackarea stixelilor statici (Fig. 5.5.29). Poziția și viteza sunt exprimate într-un sistem cartezian al lumii comun ultimelor două cadre, folosind parametri cinematicii ego-car, obținuți prin odometrie vizuală. Coordonatele înălțimii se consideră constante, presupunând că elementele scenei nu au o mișcare verticală. Modelarea mișcării se face cu filtru Kalman, considerând viteza constantă. Asocierea dintre stixelii trackați și cei măsurați se face



folosind optical flow. Pentru migrarea trackării din spațiul cartezian în spațiul imaginii, este necesară trecerea la filtrare prin filtru Kalman extins.

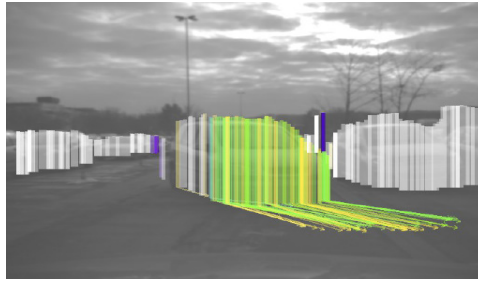


Fig. 5.5.29. Vectorii stixelilor dinamici, corespunzători unei mișcări de 0,5 secunde.

Abordarea aceasta constă din doi pași consecutivi pentru determinarea stixelilor: delimitarea spațiului liber de obstacolele din prim-plan, urmată de determinarea înălțimii obstacolelor de prim-plan. Astfel, posibilele erori ale primului pas pot fi derutante pentru al doilea pas. În plus, se determină doar frontiera obstacolelor de prim-plan, nefiind tratate obstacolele parțial ocluzate, care sunt destul de importante pentru că pot ajunge în prim-plan într-un timp foarte scurt și pot fi subiectul coliziunii cu ego-car.

Astfel, în [Pfeiffer11] s-a adoptat o abordare care folosește un singur pas pentru determinarea tuturor stixelilor, inclusiv prin împărțirea fiecărei coloane în mai mulți stixeli aflați la diferite profunzimi (Fig. 5.5.30). Extragerea stixelilor este modelată ca o problemă de estimare a maximului a posteriori (MAP = maximum a posteriori probability) și este rezolvată prin programare dinamică.



Fig. 5.5.30. Mai mulți stixeli pentru o direcție optică.

Gruparea stixelilor în obstacole este tratată tot într-o manieră de estimare MAP [PfeifferErbs12], fiecare stixel și fiecare pereche de stixeli consecutivi având asociat un cost care penalizează diferențele de profunzime, la care se pot adăuga și diferențele de mișcare. În plus, este folosit și indicele a-priori conform căruia, în funcție de poziția în spațiul imaginii, obstacolele dinamice au anumite probabilități de mișcare conform [Erbs12]. Etichetarea optimă este obținută folosind  $\alpha$ -expansion graph cut: Fig. 5.5.31. Parametrii optimi ai algoritmului au fost determinați prin folosirea unei baze de date adnotate manual, constând din 30.000 de imagini.

Abordări similare, mai complexe, se găsesc în [Erbs13] și [Erbs14].



Fig. 5.5.31. Obstacole detectate.

### Abordări în spațiul X-Y-Z

O împărțire a spațiului cartezian 3D în celule cubice cu o dimensiune constantă și suficient de mică pentru o fidelă discretizare a obstacolelor, produce mult mai multe celule goale decât celule cu puncte 3D. Pentru a evita acest neajuns, în [Broggi13], echipa VisLab de la Universitatea din Parma [VisLab] a adoptat o organizare de tip octomap/octree care împarte spațiul de interes în opt spații de dimensiune egală. Fiecare spațiu astfel obținut este împărțit în aceeași manieră, arborescent, până se ajunge la o dimensiune minimă convenabil aleasă (25 cm). Împărțirea se face doar pentru spațiile în care sunt puncte 3D, obținând astfel o structură minimalistă. Spațiile de dimensiune minimă sunt numite aici voxel: Fig. 5.5.32.

Autorii evită să specifice explicit modelul folosit pentru suprafața solului, dar specifică faptul că doar voxelii aflați la înălțime pozitivă sunt folosiți pentru detecția obstacolelor. Rezultă astfel că solul este considerat plat și că ego-car nu oscilează față de planul solului.

Gruparea voxelilor în obstacole se face cu un algoritm de tip region-growing, plecând de la un voxel inițial, neconsiderat încă, și adăugând la grup voxelii vecini neconsiderați, care se află la cel mult o distanță (de tip Chebyshev) aleasă dinainte (nespecificată) și care nu diferă prea mult din punct de vedere cromatic. Diferența cromatică este evaluată ca suma diferențelor, în spațiile RGB și HSL, dintre culoarea voxelului și culoarea medie a grupului, dar nu se specifică nimic despre alegerea pragului diferenței maxime.

Adesea, obstacolele sunt compuse din părți cu cromatică diferită. Astfel, pasul următor este cel al agregării grupurilor apropiate (nu se precizează cât de apropiate): Fig. 5.5.33. Grupurile mici, rămase neagregate, sunt eliminate, cel mai probabil fiind cauzate de erori de reconstrucție. Metoda prezintă risc de grupare a obstacolelor învecinate.

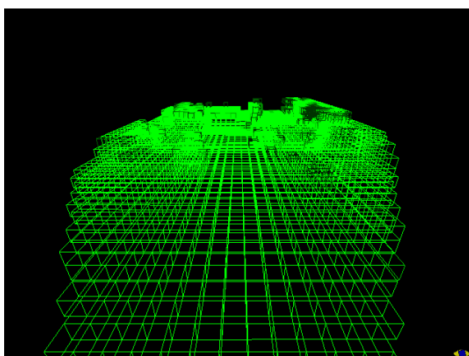


Fig. 5.5.32. Împărțirea spațiului ocupat în voxelii.



Fig. 5.5.33. Gruparea voxelilor în obstacole.

### 5.5.3 Concluzii

Reamintim că o bună detecție din imagini video stereo necesită mai întâi o bună înțelegere a detecției din cadre stereo individuale. Astfel, în acest articol sunt prezentate și analizate diverse abordări pentru detecția obstacolelor din cadre stereo individuale. S-a amintit și modul de separare dintre sol și obstacole.

S-a făcut o clasificare originală a abordărilor în funcție de profunzimea folosită și în funcție de spațiul de procesare folosit preponderent, algoritmi și calitatea rezultatelor depinzând de acestea. În urma înțelegerii profunde a abordărilor, se pot face comparații între spațiile de procesare și între abordări.

#### Comparație între spațiile de procesare

Spațiul 3D al scenei reale este prea vast comparativ cu datele rezultate din spațiul imaginii îmbogățit cu informație de profunzime. Astfel că diverse abordări de detecția obstacolelor încearcă să reducă spațiul de procesare.

Dacă ar fi să facem o comparație scurtă între spațiile carteziene și cele bazate pe disparități, am putea spune că spațiile carteziene nu deformează realitatea și că spațiile disparităților urmăresc posibilitățile reale de percepție prin stereo viziune.

În continuare, vor fi ordonate spațiile de procesare, începând cu cele mai slabe (atenție: ordinea este totuși aproximativă și poate fi diferită în contextul diferitelor aplicații concrete!):

- **Spațiul V-disparity.** Singurul avantaj e că este compact pentru detecția solului atunci când este plan și fără unghi de roll. Nu separă obstacolele aflate la aceeași profunzime, iar obstacolele neparalele cu camera devin difuze, mai ales la profunzime mică.
- **Spațiul cartezian 3D.** E bun doar ca spațiu independent, comun, în cazul folosirii simultane a mai multor senzori sau în cazul folosirii mai multor cadre a sistemului stereo în mișcare. Dezavantaje: având rezoluție constantă, e prea grosier pentru profunzime mică și prea detaliat pentru profunzime mare; e greu de procesat; e mai mult gol decât plin; dacă e reconstrucție slabă, atunci obstacolele devin discontinue.
- **Spațiul U-V** cu variantele **U-V-disparity** și **U-V-Z.** Avantaje: informație destul de compactă; fiind însuși spațiul imaginii, permite legătura directă cu diverși algoritmi de procesarea imaginilor; are o legătură 1 la 1 cu punctele 3D. Dezavantaje: zonele nereconstruibile pot fragmenta obstacolele; pixeli învecinați ai aceluiași obstacol pot avea variații mari de profunzime.
- **Spațiul cartezian X-Z.** Avantaje: ca și cel cartezian 3D, e bun la folosirea mai multor senzori/mai multe cadre; permite raționamente euclidiene în spațiul scenei reale. Dezavantaje: ca și la cel cartezian 3D, rezoluția constantă nu e potrivită pentru toate profunzimile; densitatea punctelor 3D scade repede cu profunzimea (mai mult decât pătratic).
- **Spațiul U-disparity.** Avantaje: informație destul de compactă; chiar și în cazul reconstrucției deficitare (erori sau lipsa reconstrucției), obstacolele apar contigue; surprinde bine distribuția obstacolelor în scenă. Dezavantaj: pentru spațiu de interes înalt, unește obstacolele care au aceeași poziție laterală și aceeași profunzime.
- **Spațiul U-Z.** Avantaje: e compact; combină percepția laterală polară cu percepția profunzimii carteziene. Dezavantaje: interpretarea consistenței percepției structurilor de la diferite profunzimi este empirică. O variantă este **spațiul U-f(Z).**



Cele mai bune rezultate se obțin dacă evidențierea și localizarea obstacolelor se face în spațiul U-disparity (care surprinde cel mai bine puterea de percepere prin stereo viziune), iar la post procesări se folosesc și raționamente în spații carteziene 2D/3D (care sunt mai apropiate de structura scenei reale).

## Abordările echipelor

Unele echipe și persoane s-au ocupat constant de acest domeniu [VisLab, Daimler, IPPRG], în timp ce altele au avut preocupări sporadice.

Câteva echipe mai importante ar fi:

- Echipa VisLab [VisLab], de la Universitatea din Parma, Italia, condusă de prof. Alberto Broggi, cu activitate în domeniu de la începutul anilor '90, se remarcă prin inițiative de pionierat (de ex: autovehicule autonome). În privința detecției de obstacole din cadre stereo individuale, spațiul de procesare folosit preponderent este spațiul U-V al imaginii, exprimând profunzimea metric sau prin disparități, permițând abordări naturale, similare cu vederea umană. Din păcate, abordările lor sunt simpliste, având limitări importante.
- Echipa Daimler [Daimler], Germania, condusă de prof. Uwe Franke, cu activitate în domeniu de la începutul anilor '90, s-a remarcat prin seriozitate, uneori cu abordări față de care alții au fost reticenți (de ex: 6D-vision). În privința detecției de obstacole din cadre stereo individuale, au avut o abordare mai slabă în 2004 [Gehrig04], prin folosirea spațiului cartezian top-view. Dar, începând din 2007 și până în prezent, au perfecționat o abordare care generează o segmentare optimă a obstacolelor, pe baza unor costuri asociate diferitelor decizii posibile. Algoritmul este unul scalabil, putând îngloba diferite informații și decizii de natură spațială și temporală, într-o manieră probabilistică. Începând cu 2013 [Erbs13], autorii prezintă cazurile nerezolvate, propunând soluții. Limitările provin din tratarea tuturor situațiilor într-un mod generalizat.
- Echipa IPPRG [IPPRG] de la Universitatea Tehnică din Cluj-Napoca, România, condusă de prof. Sergiu Nedeveschi, cu activitate în domeniu începând din 2001, se remarcă prin abordări elaborate (de ex: detecția solului). În privința detecției de obstacole din cadre stereo individuale, evidențiază zonele cu obstacole în spațiul U-f(Z), după care aplică un șir de procesări pentru obținerea unei modelări cât mai fidele a obstacolelor, prin cuboide [Pocol08]. Limitările provin din modelarea forțată, prin cuboide, a unor obstacole necuboidale. O abordare alternativă este cea a modelării prin frontiere poligonale [Vatavu11].

Raphael Labayrade et al. și apoi Zhencheng Hu et al. au evidențiat anumite posibilități de detecție a solului și a obstacolelor în spațiile V-disparity și U-disparity. Deși metodele lor sunt limitate la detecția unor fațete plane cu orientări particulare (ce-i drept, compunând majoritatea obstacolelor), au meritul aprofundării cunoștințelor despre consistența informațiilor bazate pe disparități.

Roberto Manduchi et al. propun o metodă de separare a solului și a obstacolelor, inclusiv în scenarii off-road, cu procesări în spațiul imaginii. Din păcate, autorii se limitează la modelarea obstacolelor individuale prin mulțimi conexe de puncte 3D.

Este "interesant" că mulți autori își califică abordările drept robuste și cu rezultate bune, după care nu se mai aude nimic de ele.

## 5.6. Concluzii, contributii

Acest capitol face un studiu a celor mai importante abordări din literatura de specialitate, care încearcă să rezolve aceleași probleme ca și teza de față. Sunt prezentate abordări de viziune mono de intensitate, mono color, secvențe de imagini și imagini stereo.

În lucrarea de față, se propune o clasificare asemănătoare cu cea din [Regensburger94], punând accentul pe capacitatea abordărilor de a recunoaște obstacole generice și de a furniza informații despre alcătuirea 3D a scenei.

În cazul stereo viziunii, o contribuție originală este gruparea abordărilor în funcție de spațiul de procesare folosit preponderent. S-a observat că, pentru simplificarea procesărilor și compactarea datelor, se folosesc adesea proiecții particulare, fie ale spațiului cartezian 3D, fie ale spațiului nativ al camerei stereo. Originale sunt și comparațiile între spațiile de procesare și comparațiile între cele mai importante echipe de cercetători. S-a argumentat faptul că cel mai bun este spațiul top-view U-disparity, folosit și în teza de față. Spațiul cartezian ar trebui folosit pentru procesări secundare sau pentru criterii suplimentare în cadrul procesărilor principale.

## Capitolul 6. Soluție originală pentru detecția obstacolelor

La modul ideal, un algoritm de detecția obstacolelor ar trebui să nu împartă un obstacol real în mai multe obstacole detectate și invers, să nu unească mai multe obstacole reale într-un singur obstacol detectat. Acestea sunt două cerințe opuse. Și sunt cu atât mai greu de îndeplinit cu cât scena are mai multe obstacole apropiate unele de altele. În plus, se mai adaugă erorile de reconstrucție și deformările proiecției perspective prin care se obțin imaginile. Metodele care folosesc și informația de mișcare întâmpină și erorile de la detecția mișcării în imagini (optical flow).

Vom avea în vedere următoarele considerente:

- Modelul folosit pentru reprezentarea obstacolelor va fi cel cuboidal, având dimensiuni (lățime, lungime, înălțime), poziția 3D a centrului cuboidului și o posibilă orientare în planul orizontal (yaw); acestea vor fi exprimate în sistemul cartezian al ego-car;
- Prin *cameră* vom înțelege camera stângă, camera dreaptă fiind considerată ca utilă doar pentru reconstrucția stereo.

Vor fi prezentați mai mulți pași, grupați pe câteva direcții, începând cu evidențierea zonelor ocupate, continuând cu diverși pași de rafinare a obstacolelor și terminând cu câteva procesări aplicabile obstacolelor generice. S-a mers pe ideea, și s-a ajuns la concluzia!, că nu există un singur pas magic care să producă obstacole detectate fidele obstacolelor reale.

Pașii abordării, grupați pe câteva direcții, sunt:

Evidențierea zonelor ocupate

- Selecția punctelor 3D de interes: nu toate punctele furnizate de reconstrucția stereo sunt în spațiul care conține obstacolele de interes.
- Spațiul cartezian 3D nu este propice procesării punctelor 3D. În schimb, este construit un spațiu orizontal comprimat.
- Celulele spațiului comprimat sunt grupate în obstacole. Un astfel de obstacol poate cuprinde mai multe obstacole reale.
- Frontiera obstacolelor este rafinată, în vederea micșorării inacurateții reconstrucției stereo.

Procesări pe direcția verticală

- Urmează câteva procesări care analizează obstacolele pe verticală.

Valorificarea formelor obstacolelor

- Obstacolele concave sunt fragmentate, pentru că nu sunt modelabile prin cuboide, și pentru că sunt modelabile printr-o succesiune de cuboide mai mici.
- Determinarea orientării obstacolelor.
- Obstacolele care încă nu sunt modelabile prin cuboide fidele se fragmentează mai departe în urma analizei naturii convexității lor.

Alți pași.

- Obstacolele mai mari absorb obstacolele mai mici din interiorul lor.
- Clasificarea obstacolelor de fundal este necesară altor procesări ale obstacolelor.
- Micile obstacole fantomă pot fi eliminate.

Proiectarea și implementarea de metode de detecție de obstacole din puncte 3D a avut un caracter de pionierat, anterior folosindu-se cu precădere spațiul disparităților.

## 6.1. Evidențierea zonelor ocupate

Abordarea de față preferă ca, în primă fază, să unească mai multe obstacole reale într-un singur obstacol detectat, iar apoi să aplice diverse criterii de fragmentare și rafinare. Din acest motiv, obstacolele obținute în prima fază se mai numesc și *zone ocupate*, și abia cele finale se pretează mai bine la a fi numite obstacole, fiind mai fidele obstacolelor reale.

### 6.1.1 Selecția punctelor 3D de interes

Încă de la început, putem observa că nu toate punctele 3D ne sunt de folos. De exemplu, punctele aparținând solului nu aparțin obstacolelor. Pe scurt, avem nevoie doar de punctele aflate deasupra solului, aflate nu prea sus și nu prea departe. Putem defini punctele de care avem nevoie din punct de vedere inclusivist sau exclusivist. Din punct de vedere exclusivist, clasele de puncte de care nu avem nevoie sunt:

- Puncte aflate sub sol: cel mai adesea acestea sunt reconstruite greșit, dar pot fi și puncte reconstruite corect și care se află în șanțuri și gropi accentuate.
- Puncte aparținând solului: sunt puncte reconstruite corect și, în mod firesc, nu fac subiectul semnalizării prezenței obstacolelor, ci a zonelor libere, care ar putea fi parcurse de ego-car.
- O clasă aparte este cea a punctelor aflate deasupra solului, în imediata vecinătate a acestuia, așa-numitele “puncte de pe sol”. De ce sunt acestea aparte? Pentru că, din cauza imperfecțiunii algoritmilor de detecție a solului, e de preferat ca aceste puncte să nu fie folosite nici în modelarea suprafeței solului, nici în detecția de obstacole.
- Punctele care sunt mult deasupra solului. Acestea nu se află în calea ego-car și fac parte din ramurile înalte ale copacilor, clădiri înalte, stâlpi înalți, cabluri aeriene, panouri suspendate, poduri etc.

În principiu, acestea sunt clase ale unei stratificări verticale, de jos în sus. Punctele din clasele prezentate pînă aici sunt rezultatul clasificării de către algoritmi de detecția solului. Acești algoritmi au un rol esențial în înțelegerea structurii scenei în cazul sistemelor de asistare a conducerii, furnizând totodată indicii importante pentru selecția punctelor de interes ale algoritmilor de detecția obstacolelor. Diferite abordări sunt prezentate în [Aufre01, Bertozzi98, Badino07, Danescu08, Oniga10], dar nu fac obiectul lucrării de față. În funcție de abordare și obiective, acești algoritmi pot fi de tipul: lane detection (determină cu exactitate banda curentă de circulație, eventual și benzile adiacente), road detection (se ocupă mai mult de profilul vertical al drumului, indiferent de împărțirea semantică a sa în benzi de circulație) sau ground detection (specific mai degrabă scenariilor off-road, detectează toate locurile în care ego-car ar putea ajunge).

Se pot folosi și alte criterii de rejectare:

- Puncte care sunt prea departe: acest criteriu se bazează în special pe faptul că, începând de la o distanță, profunzimea exprimată metric are erori prea mari în comparație cu dimensiunile obstacolelor, dar și cu distanțele dintre acestea.
- Puncte care sunt în același timp și departe și în lateral: cel mai probabil că traiectoria ego-car nu se va intersecta cu traiectoria acelora.
- O altă justificare e legată de situația în care se află ego-car: de exemplu face o operație de parcare și nu prezintă interes obstacolele îndepărtate.
- Structura scenei prezintă particularități precum:

- Prezența unui șanț semnificativ: nu prezintă interes ceea ce se află dincolo de el.
- Prezența unui obstacol scund și de lățime considerabilă (o bordură înaltă, un gard) sau prezența multor obstacole în prim-plan: la fel, nu prezintă interes obstacolele mai îndepărtate. Facem totuși observația că aceste obstacole ar trebui detectate și catalogate în cadrele anterioare.
  - Dacă este disponibil și un algoritm de detecție a benzii curente de circulație, pot fi îngorate punctele care sunt departe de aceasta.
  - Alte criterii și indicii specifice unor situații particulare sau unor sisteme particulare de asistare a conducerii.

Pe baza acestor criterii, se ajunge la conștientizarea noțiunii de spațiu de interes (SOI = Space Of Interest). Observăm că aceasta este o noțiune inclusivă. Forma spațiului de interes folosit urmărește suprafața solului și se înalță până la înălțimea ego-car.

Un exemplu de selecție a punctelor de interes este prezentat în figura următoare:



Fig. 6.1.1. Selecția punctelor 3D de interes: puncte pe sol (verzi), puncte de interes pentru detecția de obstacole (roșii), puncte mult deasupra solului (maro)

Observăm că, din cauza unor erori de reconstrucție mai mari, unele puncte de pe drum au fost reconstruite la o distanță mai mică (de-a lungul axelor optice pe care se află) și astfel apar deasupra drumului. Rejectarea acestor puncte este prezentată într-una din secțiunile următoare.

### 6.1.2 Compresia punctelor 3D într-un spațiu compact, de tip grid de celule top-view

După cum s-a discutat mai pe larg în capitolul dedicat reconstrucției stereo, punctele 3D sunt mai rare la profunzimi mai mari. Acesta este un fenomen absolut firesc în procesul geometriei proiective a formării imaginilor digitale de tip perspectivă. Mai simplu spus: un obstacol apare mai mic în spațiul imaginii atunci când se află la profunzime mai mare în spațiul 3D. Pentru a contracara acest fenomen, se folosește un spațiu comprimat.

Spațiul comprimat este de fapt o matrice (Fig. 6.1.2.b), fiecare celulă corespunzând unei zone trapezoidale din spațiul real top-view (Fig. 6.1.2.a). Coloanele din matrice, de la stânga la dreapta, corespund direcțiilor optice ale camerei, de la stânga la dreapta. Întreg FOV al camerei este împărțit în părți de lățimi egale, numite *direcții optice*. La modul practic, fiecare direcție optică are o lățime de 6 pixeli, rezultând direcții optice cvazi egale, deoarece unghiul exact corespunzător unei coloane de pixeli depinde de poziția acesteia în imagine. La alegerea lățimii de 6 pixeli s-a luat în considerare:

- să nu cuprindă mai multe obstacole, deci să nu fie prea mare și
- să aibă consistență, adică să cuprindă un număr relevant de puncte, deci să nu fie prea mică.

Rândurile din matrice reprezintă intervale consecutive de profunzime carteziană comprimată după modelul nativ, care e mai bun decât modelul liniar, după cum s-a argumentat în capitolul despre modelarea acurateței metrice.

Astfel, dacă  $D_e$  este acuratețea disparității și  $D_{max}$  este diparitatea maximă, pentru un punct care are coordonata  $x$  în spațiul imaginii și disparitatea  $D$ , avem:

$$Col = \frac{x}{6}, \quad Row = \frac{D_{max} - D}{D_e}$$

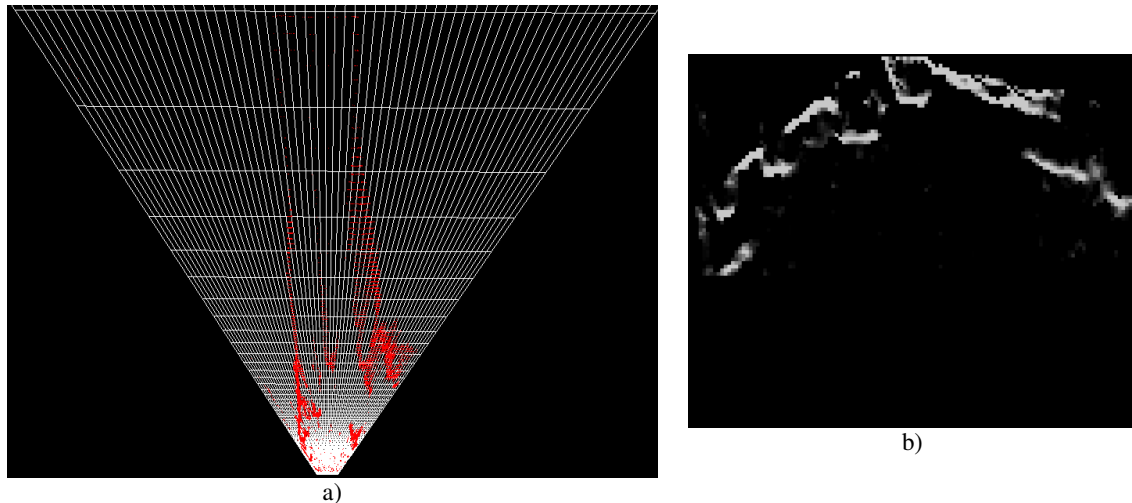


Fig. 6.1.2. a) FOV orizontal este divizat în rânduri și coloane; b) Spațiul comprimat.

În Fig. 6.1.2.a, observăm o ușoară înclinare spre dreapta a FOV, deoarece există un unghi yaw între sistemul de coordonate al ego-car și orientarea camerei. Toate direcțiile optice converg în cameră. Profunzimea este limitată de către utilizator.

În Fig. 6.1.3, celulele top-view (corespunzătoare celulelor din spațiul comprimat) sunt reprojecate în spațiul perspectiv al imaginii (stângi). În cazul particular în care suprafața solului ar fi planară și orizontală, s-ar crea impresia vizuală că gridul celulelor ar fi întins pe suprafața solului. Dar în cazul general, reproiecția în spațiul imaginii este utilă doar pentru a arăta corespondența dintre coloanele spațiului comprimat și direcțiile optice al camerei, iar rândurile nu au nicio semnificație.



Fig. 6.1.3. Reproiecția în imagine a spațiului comprimat

Într-o primă fază, pentru detecția de obstacole, s-au folosit doar punctele 3D aflate la cel mult 1 metru deasupra drumului. Această decizie se bazează pe argumentele:

- Punctele de pe suprafața parbrizelor au mari șanse să fie reconstruite greșit din două motive:
  - o Prezintă reflexii ale altor obiecte din scenă (stâlpi, semafoare, cabluri suspendate, clădiri) sau chiar a soarelui
  - o Datorită transparenței, imaginea devine compusă din: obiecte din interiorul mașinii, obiecte aflate dincolo de mașină, inscripții pe geam (și reflexii – după cum s-a mai zis)
  - o Partea de sus a mașinilor are adesea pixeli saturați
- La cele mai multe vehicule, partea de sus începe mai în spate decât partea de jos (de exemplu parbrizul este mai în spate decât capota) și astfel s-ar favoriza unirea cu obstacolele din spate.
- Se reduce numărul de puncte 3D procesate (sub 10000).

Mai târziu, după multe experimente, s-a luat totuși hotărârea folosirii tuturor punctelor aflate deasupra drumului și la cel mult înălțimea ego-car.

Vom vedea mai târziu că, după localizarea obstacolelor folosind punctele 3D descrise mai sus, sunt folosiți alți algoritmi pentru delimitarea verticală cât mai corectă a obstacolelor.

Spațiul comprimat are asociată o histogramă bidimensională, fiecare celulă contorizând numărul de puncte 3D pe care le conține. Ne putem imagina că fiecare celulă corespunde unei prisme verticale având drept bază acea celulă. În imaginile prezentate, profunzimea crește

începând de la rândurile de jos (*primele rânduri*) spre rândurile de sus (*ultimele rânduri*). Celulele care nu au puncte 3D (*celule vide*) pot fi considerate ca reprezentând spațiu liber. Celulele care au puține puncte 3D pot fi și ele considerate libere, cel mai probabil că acele puncte 3D au fost reconstruite greșit (de exemplu puncte de pe drum, unde textura e slabă, Fig. 6.1.1). Celelalte celule, având suficiente puncte, semnaleză prezența obstacolelor și le vom numi celule *valide*. Din cauza faptului că sistemul de stereo viziune este complex, numărul minim de puncte pentru o celulă validă este ales empiric (de exemplu 20). În figura următoare este prezentată o astfel de histogramă, celulele mai albe având mai multe puncte.

Punctele 3D folosite aparțin unui spațiu care are o înălțime constantă din punct de vedere metric. Dar din punctul de vedere al proiecției perspective înălțimea acestui spațiu scade liniar cu profunzimea. Acesta este unul dintre factorii care influențează densitatea punctelor, respectiv influențează valorile din histogramă (Fig. 6.1.4.a). Pentru a contracara aceasta, valorile rândurilor vor fi amplificate proporțional cu profunzimea (Fig. 6.1.4.b). Mai este încă o particularitate care trebuie luată în calcul: cel mai adesea, obstacolele aflate la profunzime mică au puține puncte reconstruite (Fig. 6.1.1 masina din dreapta) deoarece prezintă suprafețe mari, în spațiul imaginii, cu textură slabă. Pentru aceste obstacole, reconstrucția reușește mai mult în jurul muchiilor, rezultând un număr relativ constant de puncte reconstruite, pentru că muchiile sunt relativ constante la diferite profunzimi mici. Prin observații directe, s-a constatat că acest fenomen are loc până la o profunzime de aproximativ 10 metri, iar obstacolele de la profunzime mai mare au aproximativ toate punctele reconstruite. În practică, acest prag (10 metri) variază în funcție de rezoluția imaginii și de apertura camerei, dar ambii factori sunt cuprinși de unul singur: distanța focală exprimată în pixeli. Pragul de 10 metri s-a stabilit pentru o distanță focală de 400 de pixeli. Pentru a funcționa corect cu orice sistem stereo, acest prag este scalat cu raportul dintre distanța focală curentă și distanța focală de 400 de pixeli. Așadar, numai rândurile de la o distanță mai mare de acest prag vor fi amplificate cu distanța. Factorul de amplificare are formula:

$$\text{Amplif} = \begin{cases} 1, & \text{pentru } Z < \delta \\ \frac{Z}{\delta}, & \text{altfel} \end{cases}, \quad \delta = 10m \frac{\text{FocalLength}}{400}$$

unde  $Z$  este distanța metrică și  $\delta$  este profunzimea până la care apar zone mari de pixeli nereconstruiți.



a) Histograma inițială

b) amplificare cu profunzimea

c) mediere tip histereză

Fig. 6.1.4. Histograma bidimensională a punctelor 3D.

Punctele 3D și pozițiile lor sunt rezultatul unui lanț de procese, fiecare având un grad de hazard, astfel că distribuția punctelor în celule este afectată și ea de zgomot. Se poate ajunge astfel la situația în care un obstacol real contiguu și distribuit constant atât în spațiul 3D cât și în cel al imaginii, să aibă unele celule mai slab populate cu puncte și altele mai puternic



populate. Soluția este să analizăm fiecare celulă validă în raport cu celulele vecine: dacă o celulă este slab populată și are celule puternic populate în jur sau dacă este puternic populată și are celule slab populate în jur, valoarea ei se va înlocui cu media valorilor dintr-o vecinătate de 3x3. Astfel se implementează un principiu de tip histereză: celulele slab populate sunt întărite dacă au celule puternic populate în jur și invers, celulele puternic populate sunt slăbite dacă au celule slab populate în jur. În fond, are loc un proces de redistribuire a densității punctelor în vederea uniformizării, lucrând desigur în vecinătăți mici. Rezultatul e vizibil în Fig. 6.1.4.c.

Multe puncte reconstruite greșit sunt împrăștiate în scenă într-un mod destul de aleatoriu. Acestea vor produce celule slab populate și astfel vor putea fi ignorate. Totuși, uneori e posibil ca un grup de puncte să fie reconstruite greșit și să ajungă în aceeași celulă (eventual în două celule vecine), care va fi puternic populată. Astfel de celule se numesc *singulare* și urmează a fi identificate și eliminate (șterse). Un raționament posibil este: o celulă se consideră singulară dacă este validă și dacă are cel mult o celulă vecină validă.

La modul practic, fiecare din acești ultimi doi pași, care modifică valoarea unei celule în funcție de celulele vecine, trebuie să aibă în vedere să nu folosească valoarea unei celule proaspăt modificate, ci să folosească valoarea ei dinainte de modificare. Altfel, se poate obține un fenomen de trenare, de propagare a unui fenomen local la mai multe celule vecine succesive. Mai concret, soluția este de a folosi o matrice drept sursă și alta drept destinație.

Celulele valide vor fi folosite mai departe în procesul de detecție a obstacolelor.

### 6.1.3 Gruparea celulelor în obstacole, pe criterii de densitate și vecinătate

Este rezonabil să considerăm că, în acel spațiu de interes aflat deasupra drumului și până la înălțimea ego-car, nu avem obstacole care, într-o vedere top-view, să se suprapună între ele. Așa ceva ar putea să aibă loc în situații foarte rare în care pe drum s-ar afla un obstacol relativ scund, iar ceva mai sus s-ar afla un alt obstacol susținut cumva în aer. Cu alte cuvinte, în vederea top-view toate obstacolele sunt disjuncte. Astfel, în unele secțiuni, inclusiv aceasta, coordonata Y (înălțimea) a punctelor 3D va fi ignorată și toate procesările vor folosi doar coordonatele X și Z (top-view).

După cum s-a discutat în secțiunile anterioare, punctele de interes din spațiul cartezian 3D sunt folosite pentru a popula un spațiu comprimat, de tip grid (matrice), compresia pe cele două axe urmând îndeaproape posibilitățile de percepție ale unui senzor de stereo viziune.

În cele ce urmează, în spațiul comprimat va fi aplicat un algoritm de grupare a celulelor în obstacole, având la bază algoritmi clasici de etichetare (sau labeling), de segmentare și de region growing, adăugând criterii și raționamente specifice domeniului de față.

Algoritmul de grupare lucrează în modul următor (explicațiile pot fi urmărite pe Fig. 6.1.5.d): matricea spațiului comprimat este parcursă coloană cu coloană, de la stânga la dreapta, iar fiecare coloana este parcursă de jos în sus (de la profunzimi mici la profunzimi mari); când se întâlnește o celulă validă (având suficiente puncte 3D) se inițializează un nou obstacol; mai departe, gruparea are un comportament de tip region growing pe două direcții:

- Analizând coloana curentă și coloanele următoare. Se poate avansa la coloana următoare doar dacă aceasta prezintă un *început de coloană* aflat la cel mult 500 mm pe profunzime față de începutul coloanei curente. Termenul de *început de coloană* se referă la o celulă validă care nu are alte celule valide în fața sa (profunzimi mai mici), pe aceeași coloana, pe un interval de 500 mm. Coloana următoare nu trebuie să fie neapărat cea adiacentă coloanei curente, ci, în lipsa reconstrucției, se pot admite coloane care nu au *început de coloană*, dar nu mai mult de 300 mm în lateral.
- Analizând celulele de-a lungul acestor coloane. Se poate avansa la celulele valide următoare, dar trebuie avut în vedere să nu ne unim cu obstacolul aflat în spatele obstacolului curent, acest lucru fiind uneori posibil deoarece reconstrucția poate furniza artefacte sub formă de trene lungi de puncte 3D (Fig. 6.1.5.b și Fig. 6.1.5.c). Pentru a evita această problemă, algoritmul nu merge mai mult de  $k = 5$  celule față de începutul de coloană (trena este limitată). În plus, următoarele câteva celule sunt invalidate (cele desenate cu alb în Fig. 6.1.5.d). Un alt aspect avut în vedere este că se permit goluri de cel mult 500 mm.

Procesul de region growing al obstacolului curent se termină când nu se mai găsesc *începuturi de coloană*. De asemenea, vom avea în vedere mecanisme de protecție pentru a nu ieși în afara spațiului comprimat.

După cum se poate observa, spațiul minim dintre două obstacole a fost ales să fie 500 mm pe adâncime și 300 mm pe lățime. Aceste dimensiuni sunt convertite în număr de celule, iar pentru a crește viteza de calcul sunt precalculate și stocate în LUT-uri, fiind obținute niște constante specifice fiecărui rând de celule.

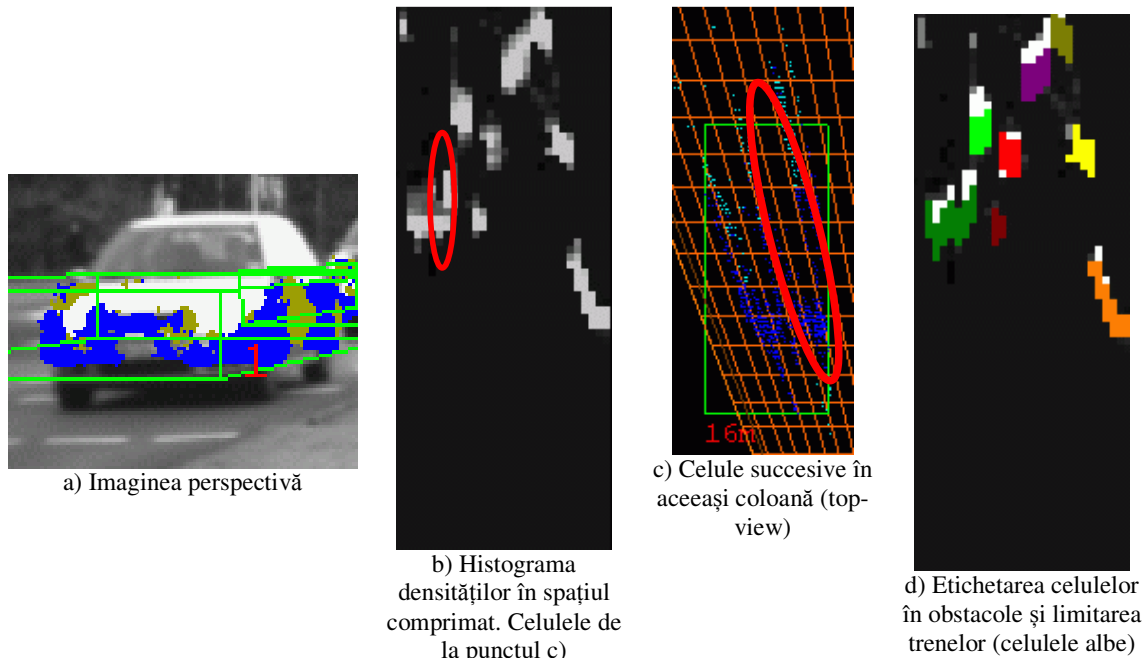


Fig. 6.1.5. Gruparea celulelor în obstacole

Algoritmul de etichetare poate fi sintetizat prin următoarea secvență de pseudo-cod:

```

for Col = 0 .. ColMax
  for Row = 0 .. RowMax
    if cell(Col, Row) has enough points and was not used yet
      init new obstacle
      while finds columns to the right (300mm gaps allowed)
        if the starting cell of new column is in the proximity of the starting cell of previous column
          while finds further cells (500mm gaps allowed), but not further than k rows
            add cell to the current obstacle
            mark cell as used by the current obstacle
          endwhile
        endif
      endwhile
    endif
  endfor
endfor

```

După cum s-a mai spus, aceste grupuri de celule, pot fi numite *obstacole*, dar pot fi numite și *zone ocupate*, deoarece este posibil să reprezinte grupuri de obstacole, urmând ca, în secțiunile următoare, să fie prezentate criteriile prin care aceste zone ocupate pot fi fragmentate și rafinate, rezultând obstacole detectate cât mai apropiate de obstacolele reale.

Algoritmul de bază folosit pentru gruparea celulelor a fost publicat pentru prima dată în cadrul conferinței *IEEE Intelligent Vehicles Symposium* din 2004, referința exactă fiind: S. Nedeveschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, T. Graf, "High Accuracy Stereo Vision System for Far Distance Obstacle Detection," in *Proc. IEEE Intelligent Vehicles Symposium (IV'04)*, Parma, Italy, 14-17 June 2004.

### 6.1.4 Rafinarea frontierei obstacolelor

Având grupuri de celule care pun în evidență prezența și localizarea obstacolelor, putem face procesări la nivel de grup de celule. O primă categorie de procesări este la nivelul frontierei grupurilor. Din cauza erorilor de reconstrucție, forma frontierei nu va respecta fidel forma obstacolului real. Problema e că nu cunoaștem forma obstacolelor. Astfel, dacă frontiera detectată este zimțată, nu putem face raționamente de tipul: „obstacolul real este zimțat și frontiera detectată este fidelă” sau „zimții frontierei detectate sunt cauzați de erori de reconstrucție și cel mai probabil că obstacolul real e neted”.

Un lucru e sigur: erorile de reconstrucție există (Fig. 6.1.6). Un punct 3D real poate fi reconstruit (măsurat) la o profunzime mai mică decât cea reală, caz în care eroarea e considerată *negativă*, sau poate fi reconstruit la o profunzime mai mare decât cea reală, caz în care eroarea e considerată *pozitivă*. Întotdeauna punctul măsurat se află de-a lungul razei optice corespunzătoare pixelului respectiv din imaginea stângă. Pe scurt, nu cunoaștem nici magnitudinea și nici sensul erorii, cunoaștem doar direcția.

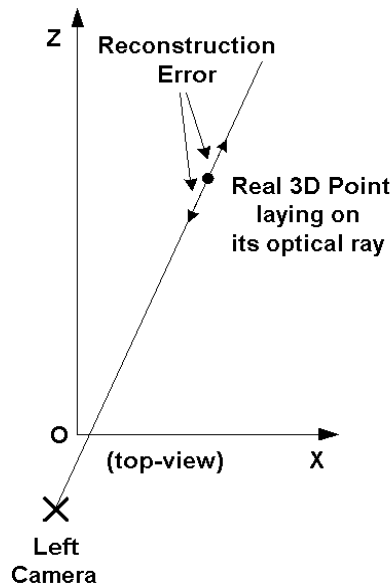


Fig. 6.1.6. Eroarea de reconstrucție a unui punct 3D

Cele mai multe erori sunt în gama impreciziei inerente a procesului de stereo matching (erori *mici*), dar acestea se încadrează în magnitudinea profunzimii unei celule din spațiul comprimat; însuși spațiul comprimat a fost construit pe acest raționament. Rezoluția numerică a profunzimilor este mai bună decât cea a celulelor (de regulă de  $2^2$  [OpenCV] sau  $2^3$  ori [TYZX]), astfel încât putem zice că avem rezoluție continuă vs. rezoluție discretă, astfel că un punct 3D poate să cadă în rândul corespunzător  $\pm 1$  rând.

Următoarea categorie este cea a erorilor *medii*, acestea fiind în gama disparităților de 0.5-1.5 pixeli (2-6 rânduri). De regulă punctele cu erori mici și medii sunt adiacente și grupabile cu întreg obstacolul. Punctele cu erori mai mari de atât sunt de regulă rejectabile prin raționamente prezentate în secțiunile anterioare.

În Fig. 6.1.7 și Fig. 6.1.8, se prezintă un exemplu ilustrativ pentru modul în care punctele 3D de la începutul unei coloane urmează a fi erodate (eliminate).

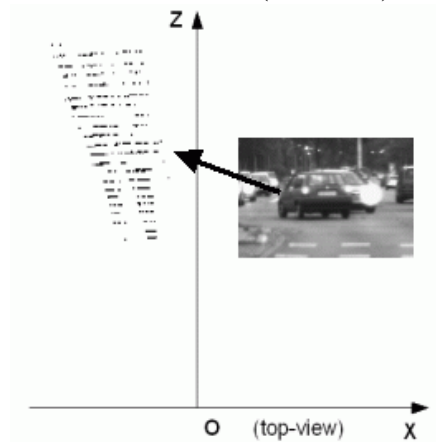


Fig. 6.1.7. Punctele 3D ale unui obstacol

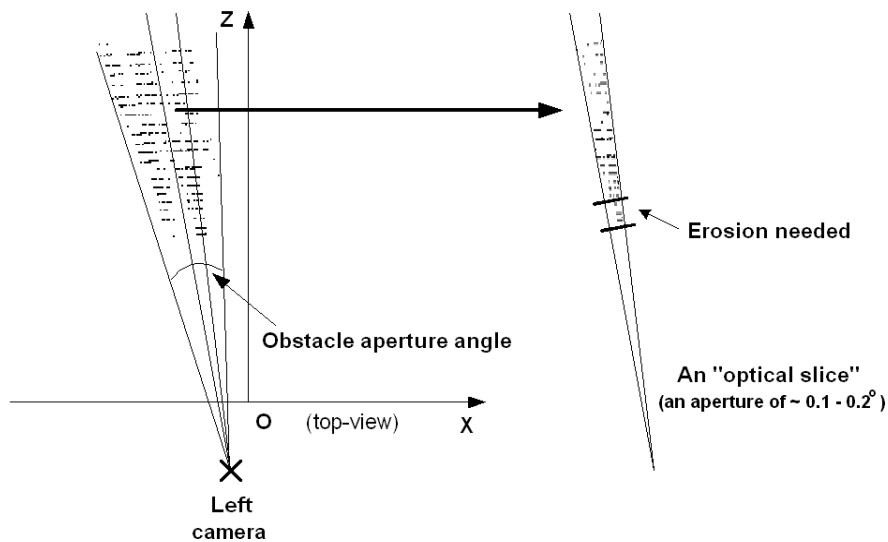


Fig. 6.1.8. O direcție optică (coloană în spațiul comprimat – optical slice) și punctele propuse spre erodare

Frontiera obstacolului are două părți:

- cea *din fața* obstacolului, înspre cameră, iar cum coloanele spațiului comprimat reprezintă direcții optice ale camerei, frontiera din față e dată de începuturile coloanelor. Punctele 3D cu erori negative vor dilata obstacolul, cel mai adesea producând o frontieră zimțată. Punctele cu erori pozitive vor fi în interiorul obstacolului și nu vor influența frontiera din față, cel mult vor influența frontiera din spate.
- cea *din spatele* obstacolului, dată de sfârșitul coloanelor. În acest caz, punctele 3D cu erori pozitive vor altera frontiera măsurată. Totuși, cel mai adesea spatele obstacolului nu este vizibil și nu putem vorbi de fidelitatea frontierei măsurate din moment ce frontiera reală nici nu este vizibilă. Numai pentru obstacolele joase pot fi vizibile margini ale feței din spate, dar nu fața însăși.

Pentru rafinarea frontierei obstacolelor s-au dezvoltat metode care erodează frontiera atât la nivel de coloană individuală, cât și la nivel de obstacol. Cele două erodări împreună sunt limitate la a eroda 20-25% din punctele 3D.

### La nivel de coloană

Erodarea la nivel de coloană (Fig. 6.1.9) are în vedere cele două categorii de erori: mici și medii. Întâi se erodează erorile medii și apoi cele mici (dacă s-ar face invers, ar însemna să fie întâi erodate mici părți ale erorilor medii). Astfel, cei doi pași sunt:

1) erodarea procentuală a 10% din numărul de puncte 3D ale coloanei, pentru eliminarea punctelor răzlețe, cu eroare medie negativă. Erodarea procentuală are un caracter de auto-adaptabilitate la consistența punctelor 3D.

2) erodare de o jumătate de celulă, pentru eliminarea punctelor care au eroare mică negativă. Nu se aplică pentru coloane foarte scurte. Cum e posibilă erodarea de o jumătate de celulă? Se construiește un spațiu comprimat cu rezoluție mai fină, de exemplu fiecare celulă din *spațiul comprimat mare* este împărțită în mai multe celule ale *spațiului comprimat fin*. Pentru această procesare, este suficient ca un rând *mare* să fie împărțit în două subrânduri *fine*.

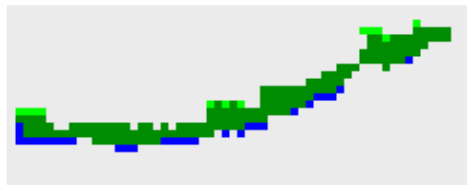


Fig. 6.1.9. Erodarea la nivel de coloană

Cele două erodări de mai sus au sens considerând că erorile, pe fiecare coloană individuală, au o distribuție gaussiană. Cazurile posibile sunt:

- da, e eroare gaussiană, ceea ce se întâmplă în majoritatea cazurilor, sau
- fie punctele sunt reconstruite mai aproape, creând excrescențe artificiale mai mari; caz în care erodarea la nivel de coloană s-ar putea să nu fie suficientă și va ajuta și erodarea la nivel de obstacol,
- fie punctele sunt reconstruite mai departe, creând goluri artificiale care sunt amplificate cu erodările 1) și 2). Cazul c) apare mai rar decât celelalte, fiind partea negativă a abordării, dar partea bună e că nu va fi afectat de erodarea la nivel de obstacol.

S-a făcut totuși și erodare la sfârșitul coloanelor, pentru o mai bună delimitare a zonei în care se află obstacolele.

Nu se aplică la coloane foarte scurte (de cel mult 2 rânduri), pentru că acesta e un indiciu că reconstrucția nu are erori semnificative în cadrul unei astfel de coloane.

Observăm că această erodare e benefică și pentru rafinarea concavităților, fiind necesară rafinarea începutului fiecărei coloane independent de celelalte coloane. Concavitățile vor fi tratate în una din secțiunile următoare.

## La nivel de obstacol

În partea dreaptă a Fig. 6.1.10, sunt prezentate coloanele unui obstacol, într-o vedere de tip top-view. Linia poligonală desenată cu albastru reprezintă înfășurătoarea celulelor. Nodurile înfășurătorii evidențiază excrescențele cauzate de erorile medii. Erodarea la nivel de obstacol se aplică în vârfurile înfășurătorii. În plus, se evită erodarea vârfurilor care au multe puncte, pentru că acestea pot prezenta porțiuni ale obstacolului real, în timp ce cele cu puncte puține pot fi cauzate de trene de puncte eronate.

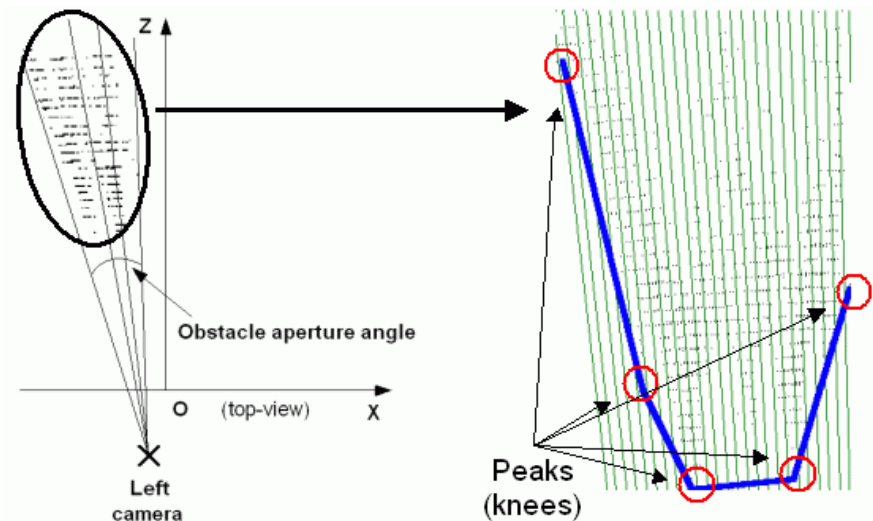


Fig. 6.1.10. Stânga: exemplu de obstacol, împreună cu apertura și punctele sale. Dreapta: direcțiile optice ale obstacolului și înfășurătoarea vârfurilor direcțiilor optice

Pentru atingerea acestor deziderate, se face o erodare repetată: se elimină câte  $k = 5$  puncte de la începutul coloanei fiecărui vârf, apoi se recalculază o nouă înfășurătoare și procesul se reia (Fig. 6.1.11). În acest fel, vârfurile care au multe puncte vor rezista mai mult decât cele care au puține. Se mai observă că, la recalcularea înfășurătorii, pot dispărea unele vârfuri și pot apărea altele noi. Pentru o mai bună localizare a obstacolului, procesul include și înfășurătoarea din spatele obstacolului (nu apare în figurile prezentate). Procesul se repetă până la eliminarea a 10-15% din punctele 3D.

Următoarea secvență pseudo-cod descrie erodarea:

```
ErodedPoints = 0
while ErodedPoints < 15% TotalPoints
    generate the convex hull on the top-view of the 3D points
    for each vertex of the convex hull
        erode the first 5 points along the optical direction (vertex's column)
        ErodedPoints += 5
    endfor
endwhile
```

Un exemplu de eroziune improprie (a înfășurătorii din fața obstacolului) e prezentat în Fig. 6.1.12. În acest exemplu fiecare celulă are  $k = 5$  puncte 3D și la fiecare înfășurătoare se elimină câte o celulă întregă pentru fiecare vârf al înfășurătorii (laturile și vârfurile înfășurătorii sunt desenate cu linii roșii, respectiv puncte roșii). La început înfășurătoarea are

o latură și două vârfuri, câte unul pentru fiecare capăt al laturii drepte din fața obstacolului. După aplicarea de trei ori a eroziunii, celulele aflate în mijlocul laturii inițiale rămân ne-erodate, iar celulele celelalte sunt cu atât mai erodate cu cât sunt mai aproape de capetele laturii inițiale.

O posibilă soluție ar fi să se aplice eroziunea de-a lungul laturilor înfășurătorii, nu doar în vârfurile acesteia.

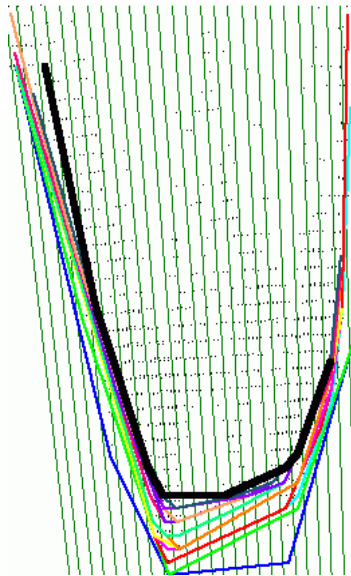


Fig. 6.1.11. Înfășurătorile succesive. Prima e desenată cu albastru și ultima cu negru îngroșat.

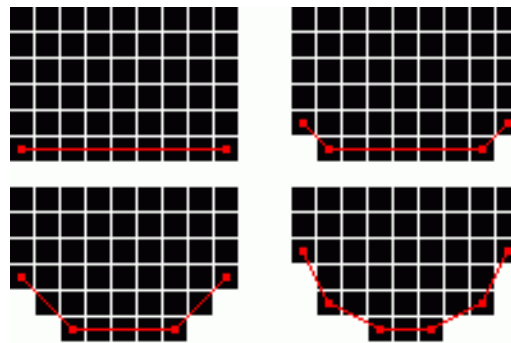


Fig. 6.1.12. Un exemplu de eroziune improprie

O altă observație, care ar putea fi investigată, este că s-ar putea face o corecție a punctelor 3D folosind informații de nivel înalt, cum ar fi faptul că un grup de puncte 3D aparțin unui obstacol. Este vorba de aceeași informație folosită și la rafinarea la nivel de obstacol. Ideea ar fi: să nu fie eliminate acele puncte 3D ci să fie împinse în interiorul obstacolului. De această corecție ar putea beneficia alte module de procesare.



## 6.2. Procesarea pe direcția verticală

Până în acest punct s-au prezentat doar procesări în spațiul comprimat orizontal – bidimensional. Fără analiza distribuției pe verticală a punctelor 3D, se pot obține obstacole detectate greșit. În Fig. 6.2.1, sunt prezentate trei surse de erori:

- structuri aeriale care nu prezintă un pericol pentru ego-car,
- puncte reconstruite cu erori mari,
- gruparea împreună cu obstacole înalte aflate în apropiere: stâlpi, copaci, ziduri.

Fiecare din aceste surse de erori, sunt tratate cu abordări specifice, prin analiza distribuției verticale a punctelor 3D. Așadar, este momentul să exploatăm și componenta verticală.

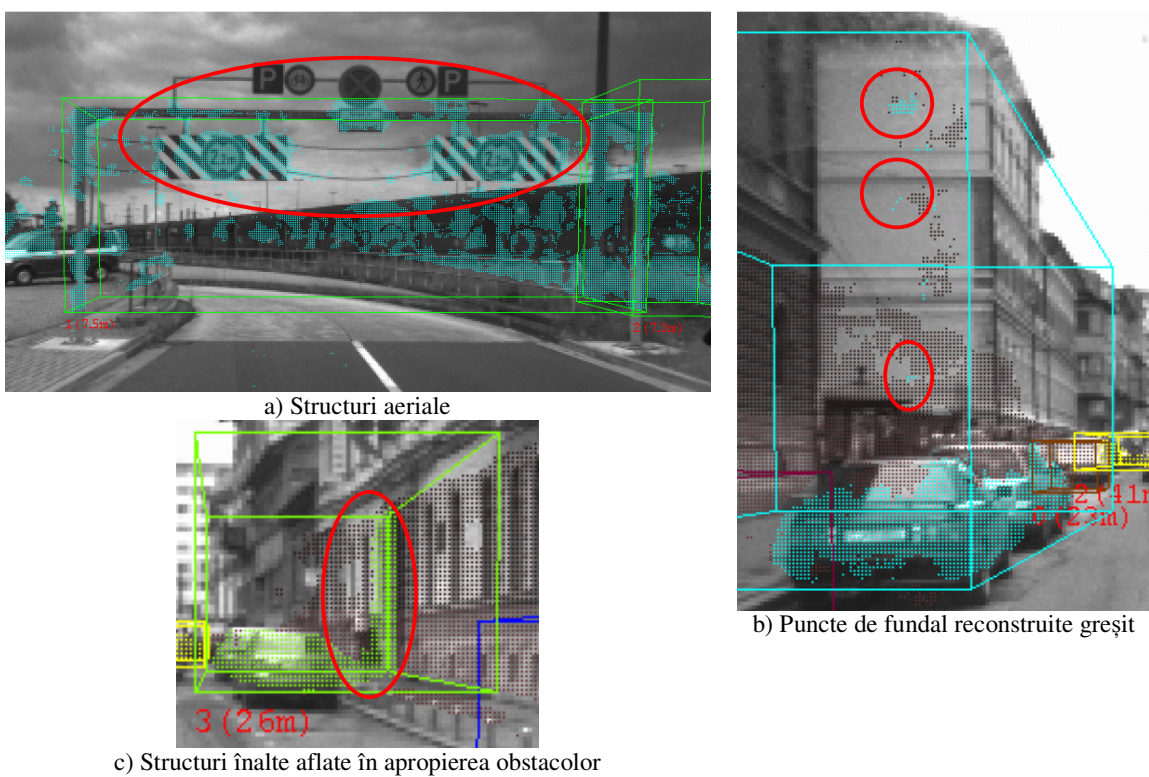


Fig. 6.2.1. Detectie greșită în lipsa analizei distribuției verticale a punctelor 3D

Pentru procesările pe verticală se folosește un spațiu comprimat al imaginii perspectivă (Fig. 6.2.2), similar cu spațiul orizontal comprimat. Pe lățime, imaginea este împărțită în grupuri de câte 6 coloane de pixeli, obținându-se coloane comprimate exact ca la spațiul orizontal comprimat, rezultând o strânsă legătură între cele două spații, relația fiind de 1 la 1. Pe înălțime, imaginea perspectivă este împărțită în rânduri de câte 2 pixeli (se poate folosi și altă granularitate). Astfel se obține o *imagine perspectivă comprimată* (observăm că raportul de aspect nu este 1 la 1), având și avantajul procesării rapide.

Aceste celule pot avea asociate diferite informații. Una dintre acestea este profunzimea exprimată metric. Deoarece sunt mai multe puncte 3D într-o astfel de celulă, se folosește principiul cunoscut în literatură sub numele de *Z-Buffer*. Inițial toate celulele primesc o valoare foarte mare (dincolo de spațiul de interes), iar apoi se ia fiecare punct 3D și dacă

profundzimea sa este mai mică decât cea din celulă, atunci celula primește profundzimea punctului 3D. Celulele fără puncte 3D vor avea în final acea valoare foarte mare.

O alta informație asociată celulelor este profundzimea exprimată ca: rândul din spațiul comprimat orizontal, astfel obținând o puternică legătură între cele două spații comprimate. Practic, fiecare celulă a imaginii comprimate ne duce astfel la celula corespunzătoare din spațiul orizontal comprimat. Celulele fără puncte 3D vor avea în final valoarea celui mai îndepărtat rând plus 1. Similar cu cazul profundzimilor metrice, aceste valori creează o hartă de tip *Row-Buffer* și se calculează direct din *Z-Buffer* prin transformarea coordonatei *Z* în rândul corespunzător.

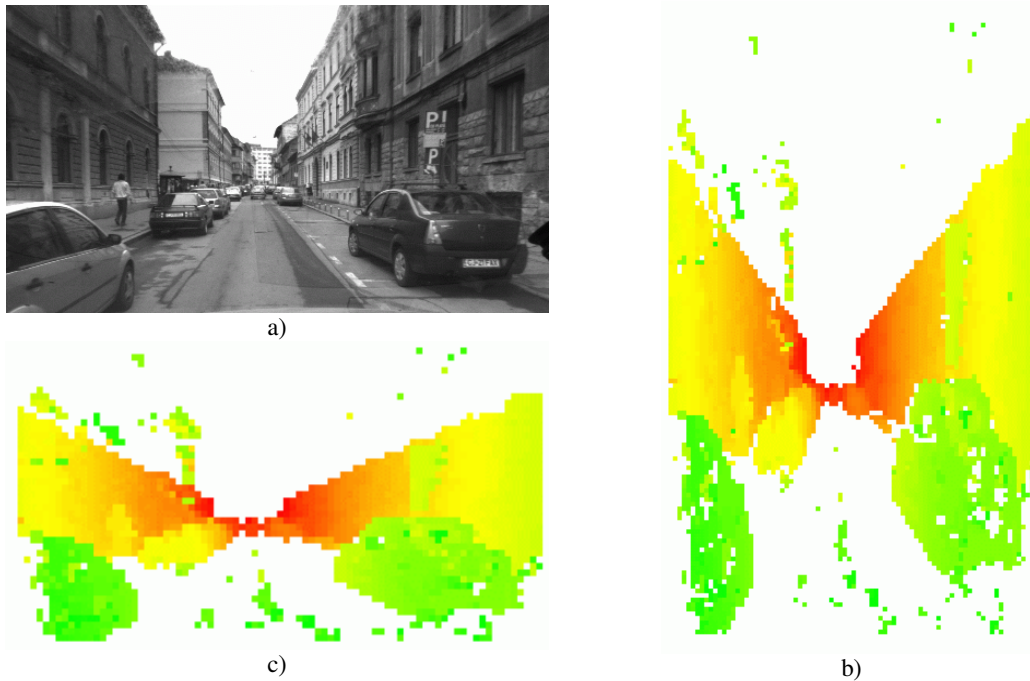


Fig. 6.2.2. Row-Buffer: a) imaginea gray-scale, b) Row-Buffer, c) Row-Buffer cu aspect 1:1 (e mai facil dpdv vizual pentru privitorul uman)

Spațiile comprimate mai poartă și numele de *hărți*. Pentru calcularea acestor hărți s-au folosit doar punctele din SOI (înalt de 4,5 metri).

Odată cu procesările “pe verticală”, fiecărui obstacol *i* se calculează dreptunghiul 2D care-l înscrie în spațiul imaginii. Astfel, pe lângă modelul cuboidal, este descris și de un model în spațiul imaginii.

### 6.2.1 Extinderea obstacolelor înalte

În Fig. 6.3.1, zidurile au fost detectate doar până la înălțimea ego-car, deși ele se continuă în sus. Dacă s-ar dori ca obstacolele mai înalte să fie detectate la reala lor înălțime, o soluție tentantă este folosirea punctelor 3D aflate mai sus de ego-car (Fig. 6.2.3). Noțiunea de *obstacol înalt* se definește după necesitățile aplicației; de exemplu, dacă se dorește detecția fidelă a camioanelor, acestea au înălțimea maximă de 4,5 metri.

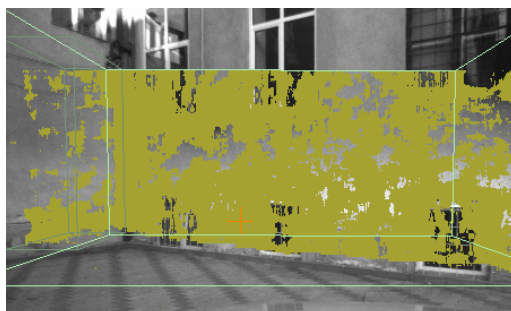


Fig. 6.2.3 Extinderea obstacolelor înalte

În această ipoteză, în Fig. 6.2.4, niște obstacole reale (cei doi stâlpi) au fost unite cu structuri aeriale. *Structurile aeriale* se află deasupra ego-car, dar nu și deasupra obstacolelor înalte (pe care ne propunem să le detectăm), astfel încât nu prezintă risc de coliziune pentru ego-car.



Fig. 6.2.4. Obstacole aeriale

Necesitatea de a detecta obstacole generice, inclusiv camioane și stâlpi, implică folosirea unui spațiu de interes având înălțimea unui camion. Spațiul de interes are o înălțime constantă și urmărește profilul vertical al solului. Punctele 3D din spațiul de interes sunt folosite pentru a construi spațiul comprimat orizontal (înălțimea punctelor nefiind folosită). Celulele valide ale spațiului comprimat sunt grupate în obstacole. În Fig. 6.2.5 sunt prezentate grupurile de celule corespunzătoare obstacolelor din Fig. 6.2.4. Un observator uman își dă seama că, de fapt, aceasta e o situație în care niște structuri solide sunt cumva susținute fix în aer și că ego-car poate trece în siguranță pe sub ele, și că doar stâlpii laterali care le susțin reprezintă obstacole reale. Exemple similare sunt: ramurile copacilor, tuneluri, intrări în garaje, poduri.



Fig. 6.2.5. Celulele spațiului orizontal comprimat, pentru obstacolele din Fig. 6.2.4

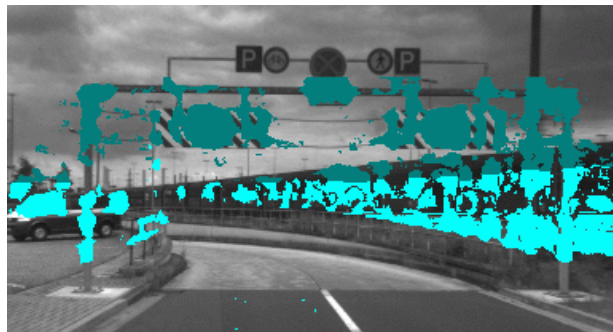
Așadar, soluția folosirii unui spațiu de interes de înălțime cât ego-car detectează incorect obstacolele înalte, iar dacă este de înălțime mai mare consideră că structurile aeriale sunt obstacole. Detecția cât mai fidelă a obstacolelor, inclusiv cele înalte, are beneficii precum: clasificarea obstacolelor, înțelegerea scenei etc.

## Soluția corectă pentru tratarea obstacolelor înalte: Folosirea a două spații de interes

Soluția propune folosirea a două spații de interes (Fig. 6.2.6):

- primul, situat deasupra profilului solului și având înălțimea egală cu cea a ego-car: punctele 3D din acest spațiu de interes sunt folosite pentru popularea spațiului orizontal comprimat și semnalează și localizează obstacolele;
- al doilea, continuă deasupra primului spațiu de interes și până la înălțimea prestabilită a obstacolelor înalte (4,5 metri pentru camioane): punctele 3D din acest spațiu sunt folosite pentru a extinde în sus obstacolele localizate în primul spațiu de interes.

Primul spațiu de interes va fi numit *spațiul de jos*, iar al doilea *spațiul de sus*.



a)



b)

Fig. 6.2.6. Două spații de interes: a) pentru scena din Fig. 6.2.4, b) pentru scena din Fig. 6.2.2.a

Limitele laterale și de profunzime ale obstacolelor sunt obținute din spațiul orizontal comprimat (obținute din punctele 3D din spațiul de jos). Limitele verticale vor fi obținute din spațiul imaginii comprimate (Row-Buffer).

Pentru un anumit obstacol celulele sale din imaginea comprimată se identifică simplu, făcând legătura dinspre spațiul imaginii comprimate înspre spațiul orizontal comprimat, pe aceeași coloană (sunt în relație 1 la 1) și obținând rândul spațiului orizontal din Row-Buffer: dacă celula “orizontală” astfel obținută aparține obstacolului curent, atunci se etichetează și celula din imaginea comprimată ca aparținând obstacolului curent. Cu alte cuvinte, sunt alese acele celule (de imagine) care se află deasupra celulelor spațiului orizontal ale obstacolului.

Obstacolele etichetate în spațiul imaginii comprimate sunt prezentate în Fig. 6.2.7. Facem precizarea că, în exemplul prezentat, detecția a fost limitată la o profunzime de 10 metri, pentru a evidenția doar obstacolele din prim-plan. De asemenea, toate figurile care prezintă



spațiul imaginii comprimate au fost generate la un raport de aspect de 1 la 1 cu imaginea inițială astfel că fiecare celulă desenată reprezintă câte un grup vertical de 3 celule.

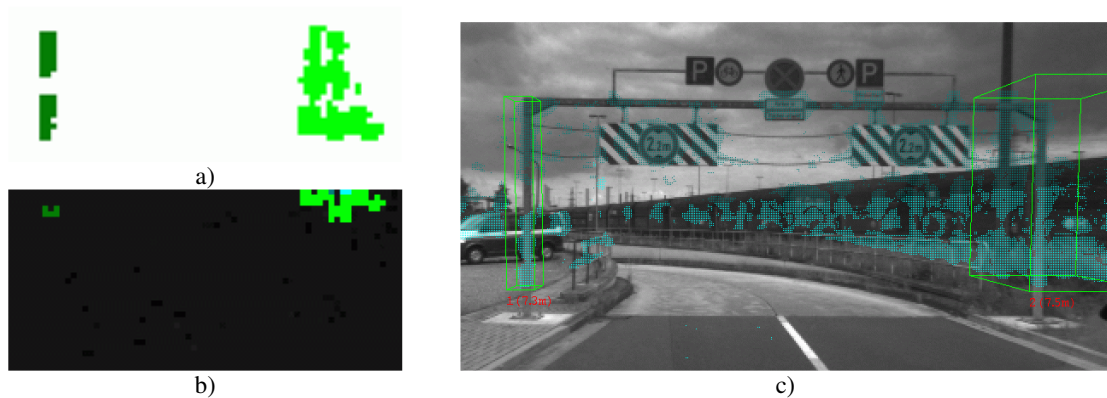


Fig. 6.2.7. a) Imaginea comprimată etichetată, b) spațiul orizontal etichetat, c) obstacolele 3D

Limitele verticale 3D ale oricărei celule din spațiul imaginii comprimate pot fi calculate cu formule simple. Astfel se calculează ușor limitele verticale ale întregului obstacol folosind toate celulele sale din imaginea comprimată. Limitele verticale nu mai sunt limitate la spațiul de jos, ci se obțin din ambele spații.

Rezumând: folosirea a două spații de interes, unul peste celălalt, are avantajul că structurile aeriile care nu prezintă un pericol pentru ego-car vor fi ignorate (neaparținând spațiului de jos), iar obstacolele înalte sunt extinse în spațiul de sus, având baza în spațiul de jos. Cu alte cuvinte, punctele 3D din spațiul de sus, care nu au puncte sub ele, în spațiul de jos, sunt ignorate.

Mai multe rezultate sunt prezentate în Fig. 6.2.8. Au fost alese patru scenarii:

- 1) indicatoare atârinate deasupra drumului;
- 2) camion;
- 3) crengi de copaci;
- 4) tunel.

Pentru fiecare exemplu sunt prezentate trei abordări:

- a) un singur spațiu înalt (4,5 metri), cu intenția de a detecta fidel camioanele (scenariul 2), dar eșuează când ceva e atârnat în aer (scenariile 1, 3, 4);
- b) un singur spațiu scund (1,8 metri), cu intenția de a evita structurile atârinate în aer mai sus de înălțimea ego-car, dar eșuează să detecteze fidel înălțimea camioanelor (scenariul 2);
- c) două spații de interes ( $1,8 \text{ m} + 2,7 \text{ m} = 4,5 \text{ m}$ ), unul peste celălalt, în ideea de a localiza obstacolele care se află sub înălțimea ego-car (în spațiul de jos) și apoi de a le extinde în sus (în spațiul de sus).



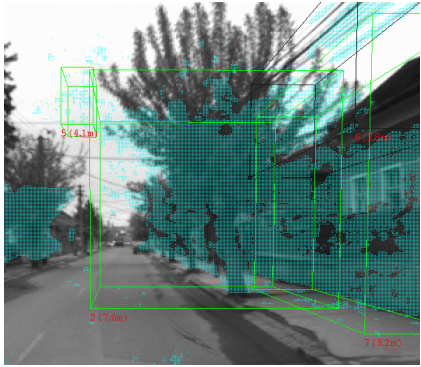
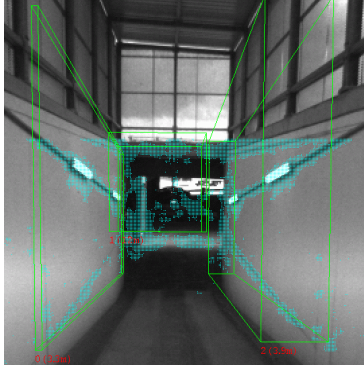

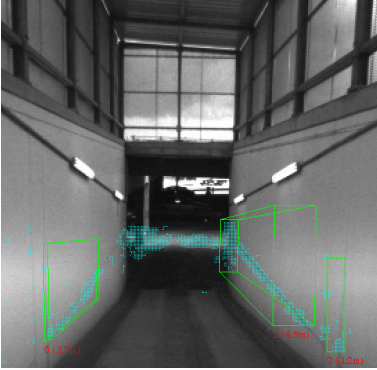
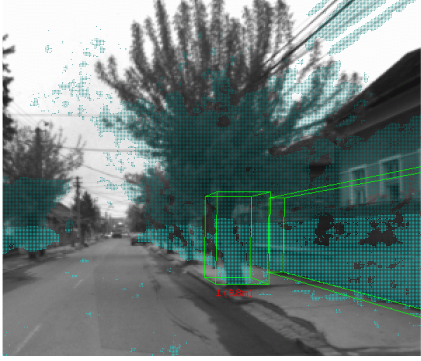
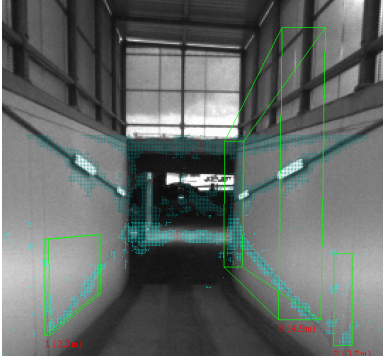
Un spațiu înalt (4,5m)	 <p>a3)</p>	 <p>a4)</p>
Un spațiu scund (1,8m)	 <p>b3)</p>	 <p>b4)</p>
Două spații (1,8m + 2,7m = 4,5m)	 <p>c3)</p>	 <p>c4)</p>

Fig. 6.2.8. Rezultate obținute folosind: a) un spațiu înalt, b) un spațiu scund, c) două spații  
 Scenarii: 1), 2), 3), 4) Distanța a fost limitată pentru a pune în evidență exemplele ilustrative din prim-plan

## 6.2.2 Fragmentarea verticală

După cum se arată în Fig. 6.2.1.b și Fig. 6.2.9, doar analiza într-un spațiu top-view, fără o analiză verticală, poate conduce la detecția de obstacole mai înalte decât cele reale.

Pentru fiecare obstacol, se construiește o histogramă verticală care contorizează numărul de celule de pe fiecare rând din imaginea comprimată. În aceasta se identifică porțiunile fără celule (valori zero) și astfel obstacolul se împarte în sub-obstacole. Sub-obstacolele mici sunt eliminate, cel mai probabil că sunt produse de puncte 3D având erori de reconstrucție mari;



de regulă acestea sunt puncte aparținând obstacolelor din fundal dar reconstruite la profunzimi mai mici.



Fig. 6.2.9. a) Încă un exemplu în care obstacolul detectat este mai înalt decât cel real, din cauza unor puncte reconstruite greșit, b) imaginea perspectivă comprimată și porțiunea verticală fără celule

În Fig. 6.2.10 se observă că obstacolul real este corect detectat și că punctele reconstruite cu erori mari sunt ignorate. Tot în acest pas, fiecărui obstacol i se atașează și modelul dreptunghiular 2D în spațiul imaginii.



Fig. 6.2.10. Rezultatul după fragmentarea verticală

### 6.2.3 Rafinarea verticală

Un alt caz în care obstacolul detectat nu este fidel celui real este atunci când în apropierea obstacolului se află un alt obstacol înalt (de exemplu: stâlp, copac, zid), iar cele două sunt grupate într-unul singur (Fig. 6.2.11.a).

Soluția folosește observația că, în astfel de situații, cea mai mare parte a volumului obstacolului detectat (70%) se află în partea de jos și că restul volumului – partea de sus – se înalță semnificativ (minim 0,8 m). Dacă se îndeplinesc aceste condiții, partea de sus se elimină.

Pentru această analiză este necesară calcularea unei histograme verticale, fiecare intrare – corespunzătoare unui rând din imaginea comprimată – reprezentând volumul unei felii orizontale a obstacolului.



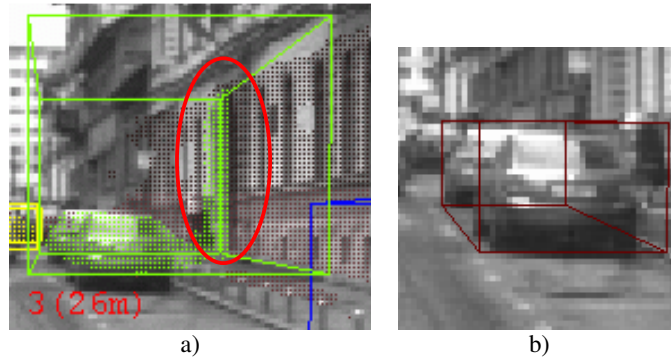


Fig. 6.2.11. Grupare gresita (a) si corectata (b)

Se consideră că volumul obstacolului este mărginit de două suprafețe:

- una orizontală și plană, numită *baza obstacolului*, reprezentată de celulele spațiului orizontal comprimat, imaginându-ne că spațiul orizontal comprimat se află pe suprafața solului (de unde și numele de bază);
- cealaltă determinată de celulele imaginii comprimate, neplană, ca un fel de plasă așezată peste partea vizibilă a obstacolului.

Calcularea histogramei se face prin parcurgerea volumului în felul următor: de la fiecare celulă a spațiului orizontal comprimat se “urcă” pe rândurile între care este cuprins obstacolul în spațiul imaginii comprimate și dacă pe coloana curentă (comună celor două spații) și pe rândul din imagine, se găsește în Row-Buffer o valoare R1 mai mică decât rândul curent R2 al spațiului orizontal comprimat, înseamnă că ne aflăm în interiorul obstacolului; astfel vom incrementa intrarea în histogramă, corespunzătoare rândului curent din imaginea comprimată.

Pentru a veni în ajutorul înțelegerii acestei explicații, avem următorul pseudo-cod:

```

for ImgRow = BottomImgRow .. TopImgRow
  Hist(ImgRow) = 0
endfor
for Col = LeftCol .. RightCol
  for HorizRow = NearHorizRow .. FarHorizRow
    if HorizCell(HorizRow, Col) belongs to the current obstacle
      for ImgRow = BottomImgRow .. TopImgRow
        if ImgRowBuffer(ImgRow, Col) < HorizRow
          Hist(ImgRow)++
        endif
      endfor
    endif
  endfor
endfor

```

Stabilirea rândului corespunzător a 70% din volum și măsurarea înălțimii părții de sus a obstacolului conduc la decizia necesității eliminării părții de sus și a punctului de la care se face eliminarea (Fig. 6.2.11.b).

Nu se aplică la obstacole scunde (sub 1,2 m).

## 6.3. Valorificarea formelor obstacolelor

Modelarea obstacolelor folosind cuboide devine nefidelă atunci când obstacolele au concavități, când obstacole cuboidale având diverse orientări sunt modelate prin cuboide neorientate (paralele cu axele de coordonate) sau când obstacolele reale nu au formă cuboidală.

### 6.3.1 Fragmentarea obstacolelor pe criterii de concavitate

De multe ori, un obstacol detectat care are concavități înglobează mai multe obstacole reale, iar punctul de concavitate maximă delimitează acele obstacole. Colțurile unei camere sau colțurile concave ale unei clădiri (Fig. 6.3.1.a) sunt exemple intuitive, dar astfel de exemple sunt adesea întâlnite și în trafic (Fig. 6.3.2.a).

Astfel se ajunge la necesitatea fragmentării obstacolelor care prezintă concavități. Concavitățile sunt spațiile libere dintre frontierele și înfășurătorile obstacolelor. Atât frontierele cât și obstacolele sunt analizate în spațiul comprimat. O concavitate mare semnaleză existența a cel puțin două obstacole reale, iar coloana în care concavitata este cea mai adâncă va fi cea în care se va face fragmentarea. Obstacolele astfel obținute vor fi reanalizate și fragmentate, de câte ori va fi nevoie.

Decizia de fragmentare și fragmentarea recursivă sunt descrise succint prin următoarea secvență pseudo-cod:

```
subroutine ConcavityBasedFragmentation(Obstacle O)
  generate the convex hull
  DeepestFreeSpace = 0
  for each side of the convex hull
    for Col = each column (optical direction) of the side
      FreeSpace = 0
      for Row = Row on side .. Row of the first obstacle's cell
        FreeSpace++
      endfor
      if FreeSpace > DeepestFreeSpace
        DeepestFreeSpace = FreeSpace
        FragmentationColumn = Col
      endif
    endfor
  endfor
  if DeepestFreeSpace is significant
    Fragment obstacle O, by the FragmentationColumn, into two obstacles O1 and O2
    call ConcavityBasedFragmentation(O1)
    call ConcavityBasedFragmentation(O2)
  endif
endsubroutine
```

În Fig. 6.3.1.c, latura de jos a înfășurătorii celulelor din Fig. 6.3.1.b delimitează o concavitate mare, având adâncimea maximă pe coloana unde celulele sunt desenate cu roșu. Prin convenție, acea coloană nu va aparține niciunuia din (sub-)obstacolele obținute. Pentru

fiecare sub-obstacol s-a calculat (și desenat) înfășurătoarea, dar nu s-au găsit concavități mari pentru noi fragmentări. Un exemplu cu mai multe fragmentări succesive este prezentat în Fig. 6.3.2.

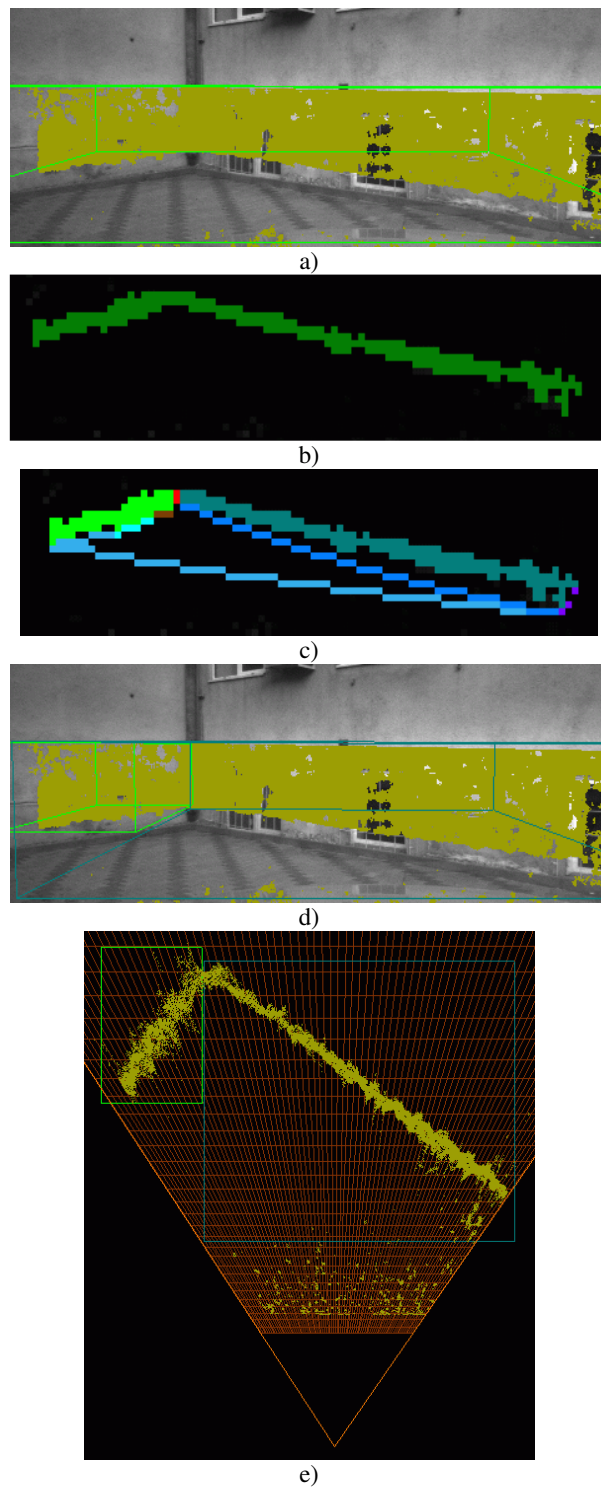


Fig. 6.3.1. Fragmentarea unui obstacol. a) obstacolul inițial, b) celulele grupate în spațiul comprimat, c) înfășurătorile obstacolului inițial și a sub-obstacolelor obținute, d) sub-obstacolele obținute – în spațiul imaginii, e) sub-obstacolele obținute – în spațiul top-view

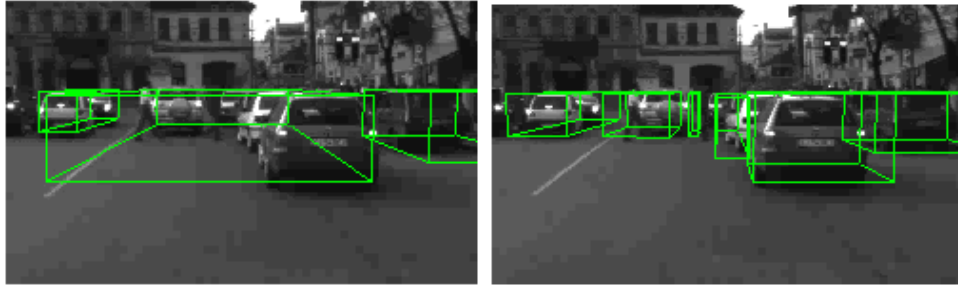


Fig. 6.3.2. Exemplu în trafic: obstacolul central (a) este fragmentat în patru sub-obstacole (b)

Prin reconsiderarea coordonatelor (inclusiv  $Y$ ) a punctelor 3D, care au determinat acele celule în spațiul comprimat, se determină limite cuboidului care înscrie obstacolul: Fig. 6.3.1.d (vedere perspectivă) și Fig. 6.3.1.e (top-view).

O posibilă îmbunătățire s-ar putea obține prin trackarea deciziilor de fragmentare în mai multe imagini succesive, obținându-se fragmentări mai stabile sau s-ar putea folosi fragmentări din imaginile anterioare atunci când deciziile sunt la limită.

Fragmentarea bazată pe criterii de convexitate este unul din pașii originali ai prezentei lucrări, fiind prezentată pentru prima dată la nivel internațional în 2007 în: S. Nedevschi, R. Danescu, T. Marita, F. Oniga, **C. Pocol**, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, M. A. Obojski, "A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision," in *Proc. Intelligent Vehicles (IV'07)*, Istanbul, Turkey, 13-15 June 2007.

### 6.3.2 Orientarea obstacolelor

În Fig. 6.3.3.a, este prezentat dreptunghiul care delimitează un obstacol, într-o vedere top-view. De fapt, acest dreptunghi este baza cuboidului obstacolului. Dreptunghiul este paralel cu axele sistemului de coordonate, dar obstacolul real este orientat oblic în acest sistem, astfel că dreptunghiul cuprinde și spațiu liber pe lângă obstacolul real. “Silueta” punctelor 3D ale acestui obstacol pare să aibă o latură dreaptă, sugerând orientarea reală. Înfășurătoarea punctelor poate modela această siluetă, iar o analiză a laturilor sale vizibile poate determina orientarea. Laturile vizibile sunt cele din fața obstacolului, dinspre cameră. Dacă analiza nu poate determina o orientare preponderentă a acestor laturi, cuboidul rămâne paralel cu axele sistemului de coordonate: este neorientat.

În Fig. 6.3.3.b este prezentată o astfel de înfășurătoare. Algoritmul caută lanțuri cvazi-drepte de laturi consecutive. Condiția de lanț cvazi-drept este ca dispersia unghiurilor pantelor laturilor componente să fie mică (formula de mai jos). Dacă lungimea celui mai lung lanț este de cel puțin 70% din lungimea laturilor vizibile, orientarea obiectului e calculată ca media ponderată a unghiurilor pantelor laturilor componente ale aceluia lanț. Ponderea unei laturi se calculează ca produsul dintre lungimea sa metrică și unghiul polar sub care e văzută în imagine; astfel laturile lungi din punct de vedere metric dar înguste din punct de vedere polar nu vor influența așa de mult lanțul după principiul “ce nu se vede bine trebuie să aibă o pondere mai mică”.

Dispersia unghiurilor ponderate ale pantelor este:

$$\sigma_{j \rightarrow k}^2 = \frac{\sum_{i=j}^k (l_i \alpha_i - \mu_{j \rightarrow k})^2}{k - j + 1}$$

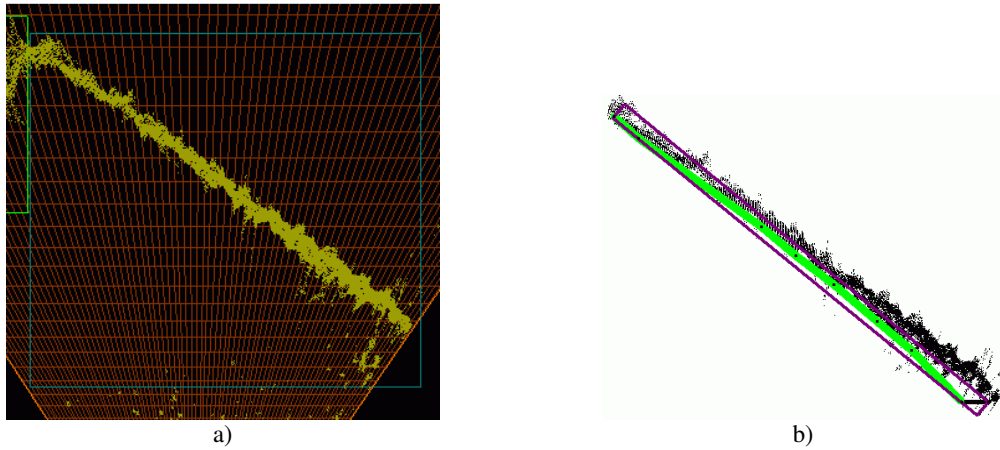
$0 \leq j < k \leq n$

unde  $n$  este numărul de vârfuri ale înfășurătorii,  $j$  și  $k$  sunt indicii vârfurilor capetelor unui lanț,  $l_i$  și  $\alpha_i$  sunt lungimea și unghiul polar (apertura optică) a laturii  $i$ , iar  $\mu_{j \rightarrow k}$  este media laturilor ponderate:

$$\mu_{j \rightarrow k} = \frac{\sum_{i=j}^k l_i \alpha_i}{k - j + 1}$$

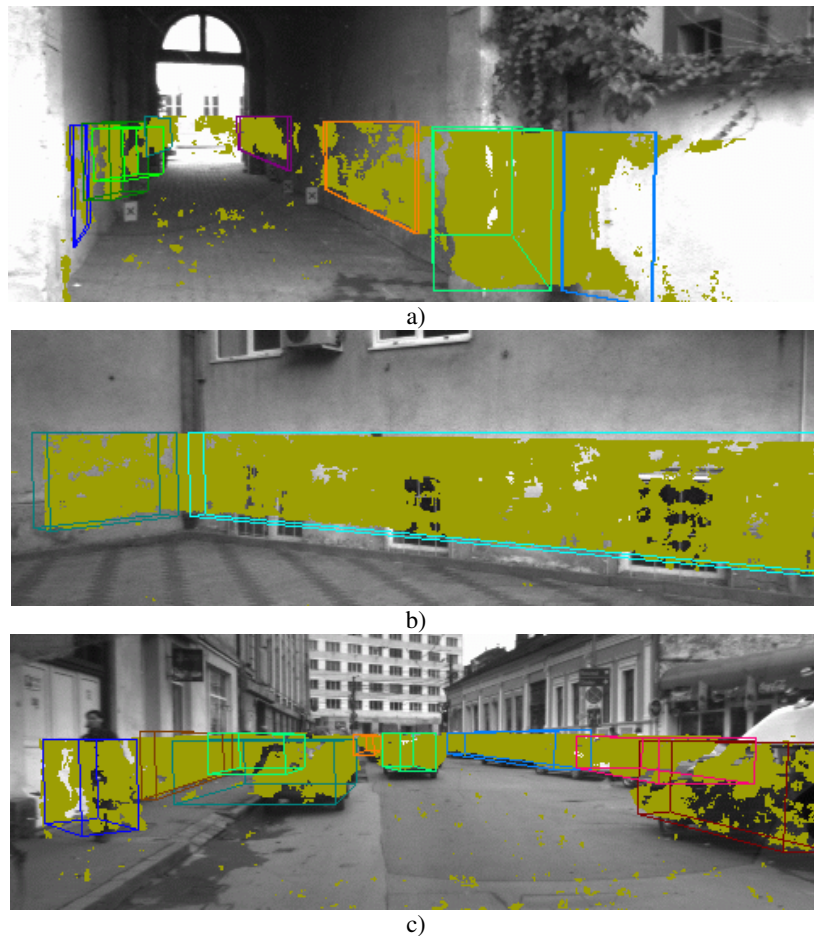
Apoi se determină dreptunghiul care are orientarea astfel calculată și care delimitează înfășurătoarea. Acest dreptunghi este baza cuboidului orientat. Rezultatul este prezentat în Fig. 6.3.3.b, făcând precizarea că înfășurătoarea nu este chiar convexă, este puțin concavă, din cauza unor aproximări folosite pentru scăderea timpului de calcul. În practică, dreptunghiul orientat poate cuprinde doar laturile vizibile sau toate laturile sau o combinație între acestea. În figura de mai jos, s-au folosit doar cele vizibile.

Mai trebuie precizat că înfășurătoarea este calculată după pasul rafinării frontierei obstacolelor, pas prezentat într-o secțiune anterioară, astfel că punctele reconstruite eronat au o influență minimă (sau chiar nulă) asupra formei înfășurătorii.



a)  
b)  
Fig. 6.3.3. Orientarea obstacolelor. a) dreptunghiul neorientat  
b) cel mai lung lanț de laturi vizibile și cvazi-liniare ale înfășurătorii (verde) și dreptunghiul orientat (mov)

Alte rezultate sunt prezentate în figura următoare:



a)  
b)  
c)  
Fig. 6.3.4. Alte rezultate: a) orientarea celui de-al doilea obstacol din dreapta nu a putut fi determinată, b) un caz ideal, c) din cauza grupării pietonului împreună cu poarta din spate (stânga), orientarea e improprie.

Pentru obstacole mici, eroarea de reconstrucție poate fi comparabilă cu dimensiunile obstacolului, astfel că nu se încearcă o estimare a orientării. E posibil chiar să nu aibă o orientare, cum ar fi în cazul stâlpilor, pietonilor sau a tulpinilor arborilor.

Convenim că dacă avem un cuboid neorientat, unghiul yaw este  $-0,99 \cdot 90^\circ$ . Dacă este orientat atunci unghiul yaw poate avea orice valoare între  $-90^\circ$  și  $+90^\circ$ , inclusiv valoarea zero.

Drept posibilă îmbunătățire, s-ar putea folosi tehnici de tracking pe mai multe imagini succesive, stabilizându-se orientările, sau s-ar putea infera în imaginile în care nu s-ar putea determina.

Unele obstacole, mai ales dintre cele mai mari, chiar dacă au o formă convexă, dacă nu sunt rectangularizate sunt improprii pentru a avea o orientare fidelă. În secțiunea următoare, va fi prezentată fragmentarea acestora în cuboide mai fidele, orientabile sau neorientabile.

\*\*\*

În continuare se va evalua acuratețea orientării obstacolelor. Aceasta este necesară pentru a avea o evaluare a algoritmului, dar și deoarece cuboidele orientate pot fi procesate mai apoi de alte aplicații: tracking de obstacole, fuziune (și comparare) cu rezultate obținute de la alți senzori de obstacole (radar, laser) etc.

Metoda a fost aplicată în condițiile în care obstacolul analizat se deplasează liniar, iar în ce privește ego-car fie e staționar, fie se deplasează liniar, fie mai e nevoie să fie luată în calcul variația orientării sale, astfel încât să poată fi separată orientarea obstacolelor detectate analizate.

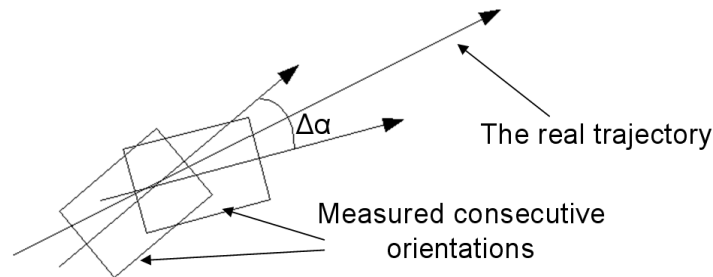


Fig. 6.3.5. Zgomotul orientărilor măsurate

După cum se poate observa în Fig. 6.3.5, în ipoteza că erorile (zgomotul) orientării sunt Gaussiene, diferența dintre două orientări măsurate ( $\Delta\alpha = \alpha_i - \alpha_{i-1}$ ) este dublul erorii fiecărei orientări. Adică eroarea este jumătate din diferență. Astfel, o estimare a zgomotului orientării, folosind multe imagini consecutive, poate fi considerată drept media jumătăților schimbărilor orientării:

$$\text{Obstacle Orientation Noise} = \frac{\sum_{n \text{ Frames}} |\alpha_i - \alpha_{i-1}| / 2}{n}$$

Pentru mai multe obstacole formula anterioară poate fi extinsă la:

$$OrientationNoise = \frac{\sum_{\substack{All\ Obstacles(m) \\ All\ Frames(n)}} |\alpha_i - \alpha_{i-1}| / 2}{m \cdot n}$$

Pentru multe imagini (10000) valoarea obținută este 2,4 grade. Ipoteza că obstacolele analizate se deplasează liniar poate fi ignorată pentru că cele mai multe obstacole nu fac viraje și pentru că adesea variația orientării datorate virajelor este mai mică decât zgomotul.

Pentru a avea o reprezentare vizuală a zgomotelor, s-a construit histograma acestora, pentru 10000 de imagini (Fig. 6.3.6). Rezoluția histogrammei este de 0,5 grade pe intrare. Se observă că sunt multe zgomote mici și unele sunt mai mari. Se observă o distribuție echilibrată în jurul lui zero, astfel că variația standard este reprezentată chiar de media valorilor absolute. Nucleul celor mai frecvente zgomote este dublul acestei medii,  $\sigma = 4,8$  grade.

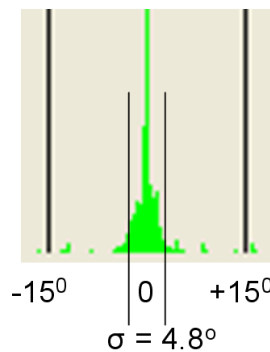


Fig. 6.3.6. Histograma zgomotelor

Prezentăm mai îndeaproape două cazuri concrete; în ambele ego-car e static.

A. În primul caz obstacolul are traiectorie rectilinie (Fig. 6.3.7). Astfel, variația orientării măsurate reprezintă însuși zgomotul.

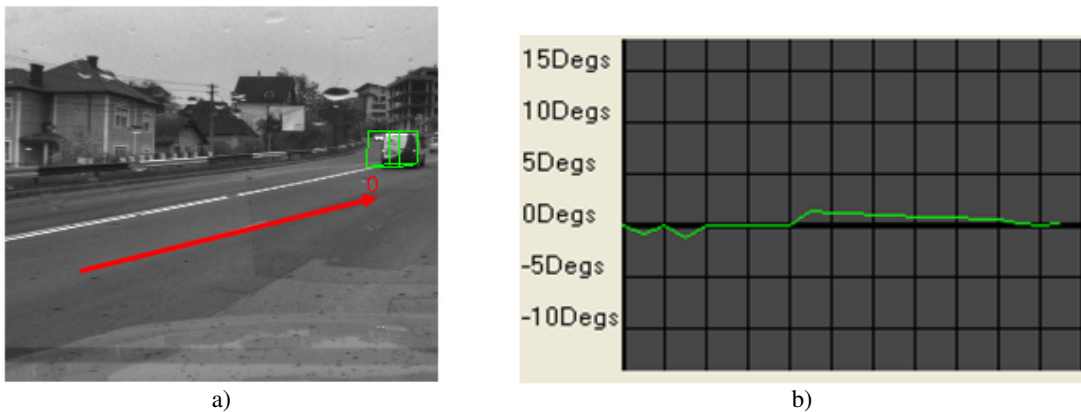


Fig. 6.3.7 a) Traiectorie rectilinie, b) Variația orientării

B. În al doilea caz obstacolul are traiectorie curbilinie (Fig. 6.3.8)

Pentru exemplul prezentat, istoricul variației orientării conține doar valori pozitive pentru că acestea includ și rotația reală.



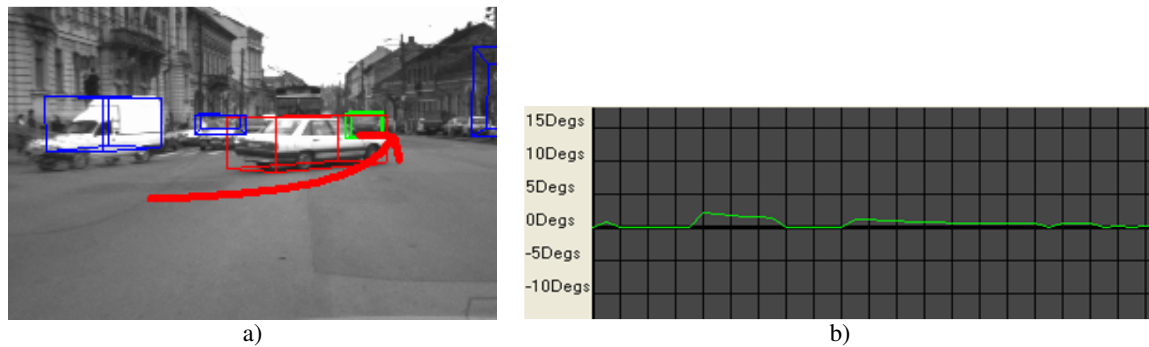


Fig. 6.3.8 a) Traiectorie curbilinie, b) Variația orientării

S-ar mai putea face potrivirea unei curbe matematice pe punctele traiectoriei pentru a surprinde cât mai corect traiectoria reală. Astfel, zgomotul orientării s-ar putea evalua ca fiind abaterea de la acea curbă.

Detecția orientării obstacolelor, prin analiza laturilor înfășurătorii convexe, a fost publicată în anul 2007 la conferința ICCP, organizată de departamentul de Calculatoare a Universității Tehnice din Cluj-Napoca, în articolul: **C. Pocol**, S. Nedevschi, "Obstacle Detection for Mobile Robots, Using Dense Stereo Reconstruction," in *Proc. IEEE International Conference on Intelligent Computer Communication and Processing (ICCP'07)*, Cluj-Napoca, România, 6-8 September 2007.

### 6.3.3 Fragmentarea pe criterii de convexitate

În ipoteza modelării obstacolelor prin cuboide, apare următoarea problemă: Ce se întâmplă dacă un obstacol mai mare, văzut pe top-view, nu are concavități, dar nici nu are o formă aproximabilă cu un dreptunghi? Pentru orice orientare propusă, un colț al cuboidului ar cuprinde un spațiu liber semnificativ (Fig. 6.3.9).



Fig. 6.3.9. Un exemplu de obstacol necuboidal

Modelul cuboidal conține doar unghiuri drepte. O primă soluție propune detectarea colțurilor care nu sunt în unghi drept, iar apoi obiectul mare să fie împărțit în sub-obstacole conform cu acele colțuri. Clasificarea colțurilor e prezentată în Fig. 6.3.9 și o posibilă fragmentare e prezentată în Fig. 6.3.10. Problema acestei soluții este că ar trebui să lucreze cu înfășurătoarea obstacolului, iar în practică cel mai adesea aceasta se depărtează mult de cazul ideal: un singur colț cu două laturi clare. În plus, ar fi o analiză relativ locală a obstacolului și nu ar avea o vedere de ansamblu a acestuia.

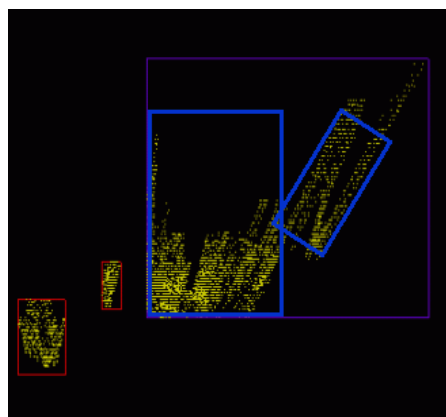


Fig. 6.3.10. O posibilă fragmentare a unui obstacol ce nu respectă modelul cuboidal, în obstacole care-l respectă

O soluție mai bună (și cea adoptată) este orientată pe analiza spațiului liber înglobat de cuboid. Acest spațiu este cel aflat între laturile cuboidului și laturile înfășurătorii punctelor sale 3D. Desigur, ca și în cazul determinării orientării obstacolelor, vorbim de înfășurătoarea rafinată, care minimizează influența punctelor reconstruite eronat pe frontiera obstacolelor.

Soluția se bazează pe faptul că spațiul liber cu suprafață mare, aflat în cuboid, indică imposibilitatea modelării corecte a obstacolului printr-un cuboid și că trebuie modelat prin cel puțin două cuboide.

Prin *spațiu liber* se înțelege cel vizibil în imagine, vizibil din poziția camerei. Spațiul liber din spatele obstacolelor sau cel cauzat de ocluzii nu poate fi luat în considerare.

Sunt două situații în care spațiul liber se datorează lipsei reconstrucției din cauza ocluziilor:

1) Ocluzia apare pentru că între cameră și obstacolul analizat se află un alt obstacol (Fig. 6.3.11). Trebuie detectat și măsurat spațiul ocluzat, iar apoi trebuie exclus din calculul spațiului liber.



Fig. 6.3.11. Fragmentarea nu este necesară: obstacolul din spate este parțial ocluzat de obstacolul din față, provocând o formă necuboidală a reconstrucției obstacolului din spate

2) Reconstrucția obstacolului este limitată pentru că e parțial vizibil în imagini (Fig. 6.3.12). În plus, una dintre camere poate surprinde o parte a obstacolului care nu e vizibilă în cealaltă cameră; aceasta se întâmplă deoarece FOV a celor două camere este ușor diferit. FOV al sistemului stereo este intersecția FOV a celor două camere. Astfel, trebuie calculate limitele FOV al sistemului stereo și trebuie calculată poziția obstacolului relativ la aceste limite, iar la calculul spațiului liber trebuie excluse aceste porțiuni ce nu pot fi reconstruite.

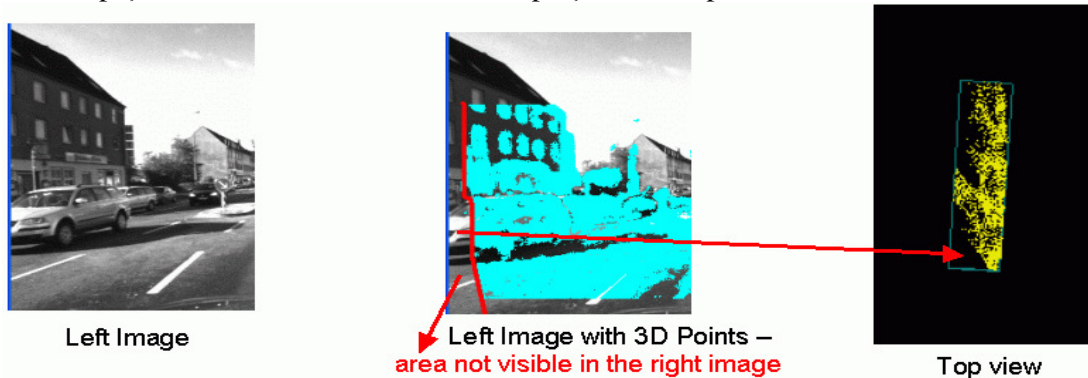


Fig. 6.3.12. Fragmentarea nu este necesară: obstacolul este parțial ocluzat de limitele FOV ale sistemului stereo, provocând o formă necuboidală a reconstrucției obstacolului

Detaliile analizei ocluziilor și ale folosirii rezultatelor vor fi prezentate în secțiunile “Analiza ocluziilor” și respectiv “Folosirea ocluziilor”.

### Evaluarea spațiului liber din cuboide

Spațiul liber este delimitat de laturile cuboidului (pe top-view este un dreptunghi orientat) și înfășurătoarea punctelor 3D (Fig. 6.3.13). Înfășurătoarea este întotdeauna un poligon convex. Deoarece cuboidul înscrie înfășurătoarea, aceasta este tangentă la laturile cuboidului și colțurile cuboidului sunt mai mult sau mai puțin goale, reprezentând spațiul gol din cuboid. Dacă spațiul liber din colțuri este mic, înseamnă că avem un cuboid confident obstacolului real. În schimb, dacă este spațiu liber mare înseamnă că nu este un obstacol modelabil cuboidal. Aria spațiului liber din fiecare colț urmează a fi calculată pentru a putea fi evaluată semnificația gradului de potrivire a cuboidului pe obstacolul real. Aria ar putea fi măsurată în metri pătrați. Totuși, acuratețea percepției scenei, folosind imaginile perspectiva, scade cu distanța: probabil  $100 \text{ m}^2$  la distanța de 50 m sau  $1 \text{ m}^2$  la distanța de 5 m au aceeași dimensiune în imaginea perspectivă. Astfel, trebuie scalată aria spațiului liber cu pătratul distanței, fiind o normalizare pentru menținerea unei semnificații constante, indiferent de distanță. O astfel de normalizare este deja implementată prin însuși spațiul comprimat folosit la detecția obstacolelor prin criterii de densitate și vecinătate a punctelor 3D. Fig. 6.3.14 prezintă această împărțire a top-view în celule care sunt mai late și mai lungi la distanțe mai mari. Astfel, aria spațiului liber este măsurată ca număr de celule ale spațiului comprimat. Totodată, algoritmul calculează și aria în metri pătrați, în caz că e nevoie. Deoarece spațiul liber se poate întinde pe multe rânduri și coloane ale spațiului comprimat, o transformare directă a ariei exprimate în metri pătrați în arie exprimată în număr de celule nu este corectă. Soluția implementată este capabilă să calculeze numărul de celule într-o manieră corectă.

Forma spațiului liber din colțuri nu este una simplă. După stabilirea colțurilor vizibile, acestea vor fi descompuse în suprafețe simple, conform unei scheme mai complicate.

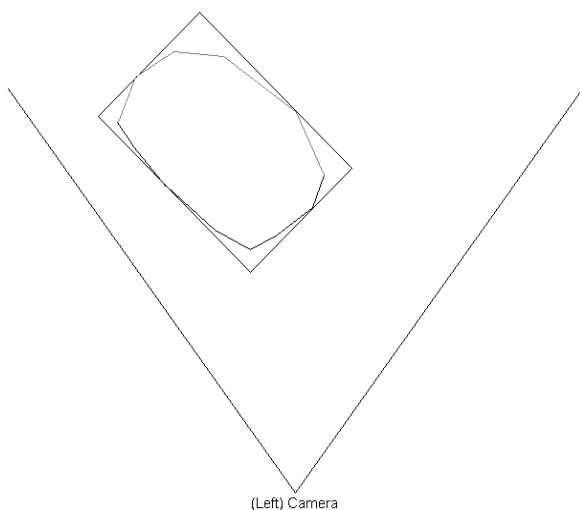


Fig. 6.3.13. Înfășurătoarea punctelor 3D și cuboidul care înscrie un obstacol (top-view)

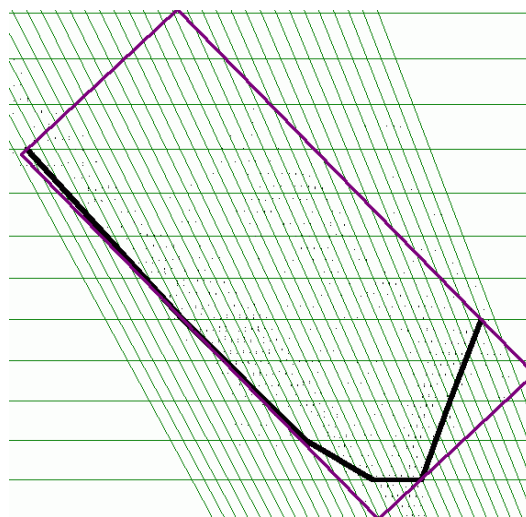


Fig. 6.3.14. Un caz real desenat peste celulele spațiului comprimat

## Stabilirea colțurilor vizibile

Orientarea unui obstacol este dată de înfășurătoarea sa, mai exact de sub-lanțul care este cvazi liniar și care are o lungime semnificativă (în Fig. 6.3.15, desenat cu roșu). Orientarea este un unghi în intervalul  $[-\pi/2 .. \pi/2]$ . Unghiurile negative reprezintă obstacole orientate spre stânga (Fig. 6.3.15), iar cele pozitive reprezintă obstacole orientate spre dreapta (Fig. 6.3.16). În practică, pentru o implementare uniformă, obstacolele neorientabile (a căror orientare nu este detectabilă pentru că cel mai lung sub-lanț cvazi liniar al înfășurătorii nu este suficient de lung) primesc o orientare de  $0.99 * (-\pi/2)$  (Fig. 6.3.17).

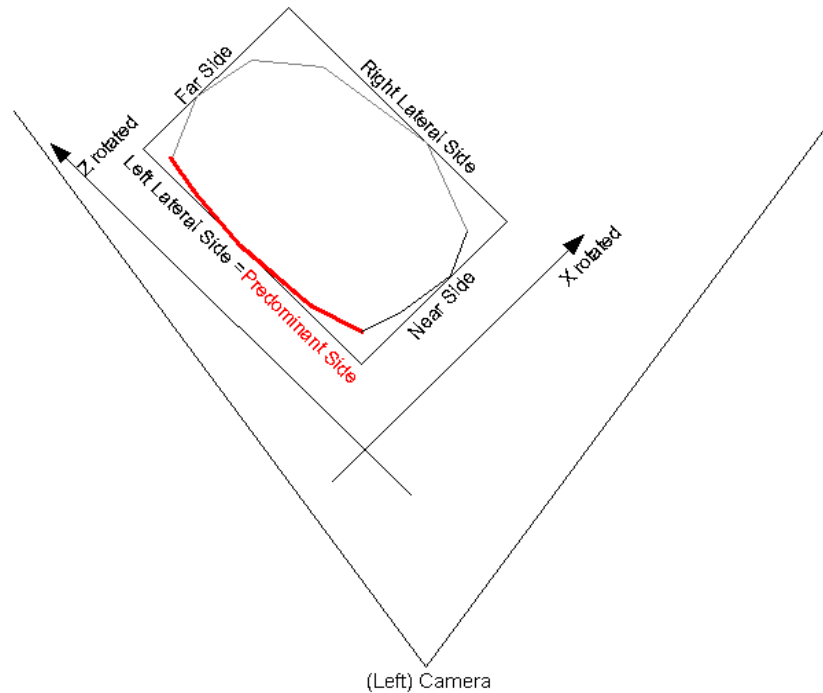


Fig. 6.3.15. Cuboid orientat spre stânga

Pentru fiecare obstacol, se folosește un sistem de coordonate local/propru (cu axele X rotated și Z rotated în figurile prezentate), axa sa Z fiind paralelă cu sub-lanțul care dă orientarea. Lanțurile sunt numite:

- “Near Side”: paralelă cu axa X rotated și având Z rotated minim
- “Far Side”: paralelă cu axa X rotated și având Z rotated maxim
- “Left Lateral Side”: paralelă cu axa Z rotated și având X rotated minim
- “Right Lateral Side”: paralelă cu axa Z rotated și având X rotated maxim

Este util să introducem noțiunea de *latură predominantă*. Pentru *obstacolele orientate-stânga* (unghi negativ), latura “Left Lateral Side” e considerată latură “Predominant Side”, fiind cea care este tangență la sub-lanțul care dă orientarea și care, la rândul său, dă orientarea, inclusiv pentru obstacolele neorientabile, care nu au un astfel de lanț. Pentru *obstacolele orientate-dreapta* (unghi pozitiv), latura “Predominant Side” este latura “Right Lateral Side”. Latura opusă laturii predominante se numește *latură non-predominantă*.

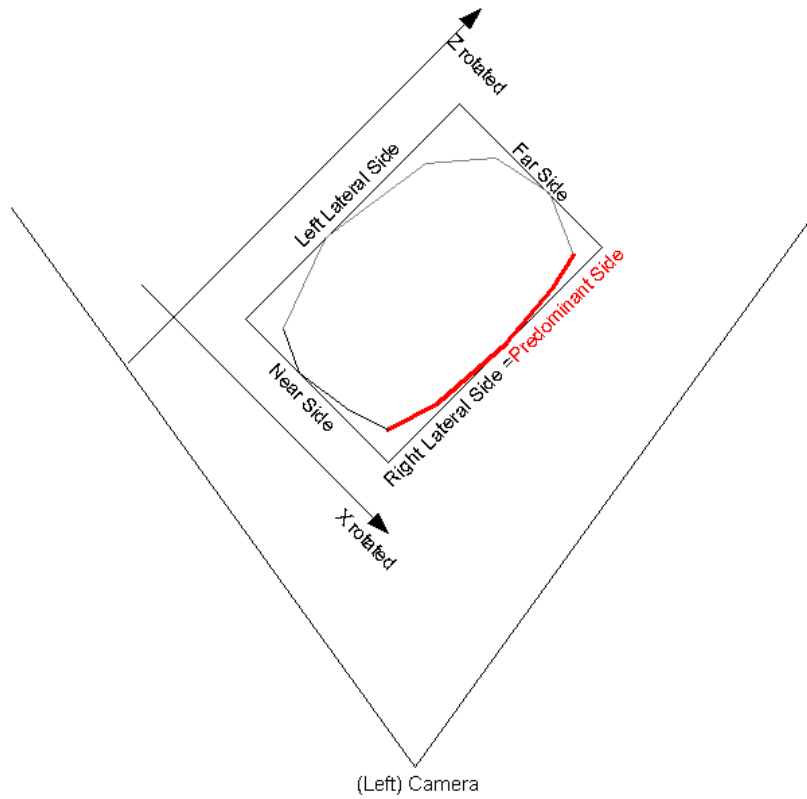


Fig. 6.3.16. Cuboid orientat spre dreapta

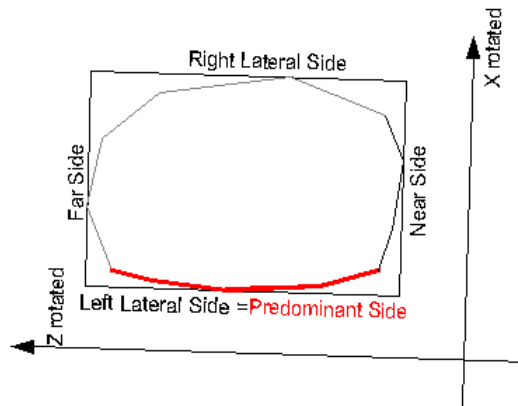


Fig. 6.3.17. Pentru un obstacol neorientabil, se folosește o orientare de  $0.99 * (-\pi/2)$ .

Spațiul liber din cele patru colțuri se clasifică astfel (Fig. 6.3.18):

- între înfășurătoare și partea stânga a laturii predominante,
- între înfășurătoare și partea stânga a laturii non-predominante,
- între înfășurătoare și partea dreaptă a laturii predominante,
- între înfășurătoare și partea dreaptă a laturii non-predominante.

Clasificarea în *parte stânga* sau *parte dreaptă* se face privind dinspre latura predominantă spre cea non-predominantă.

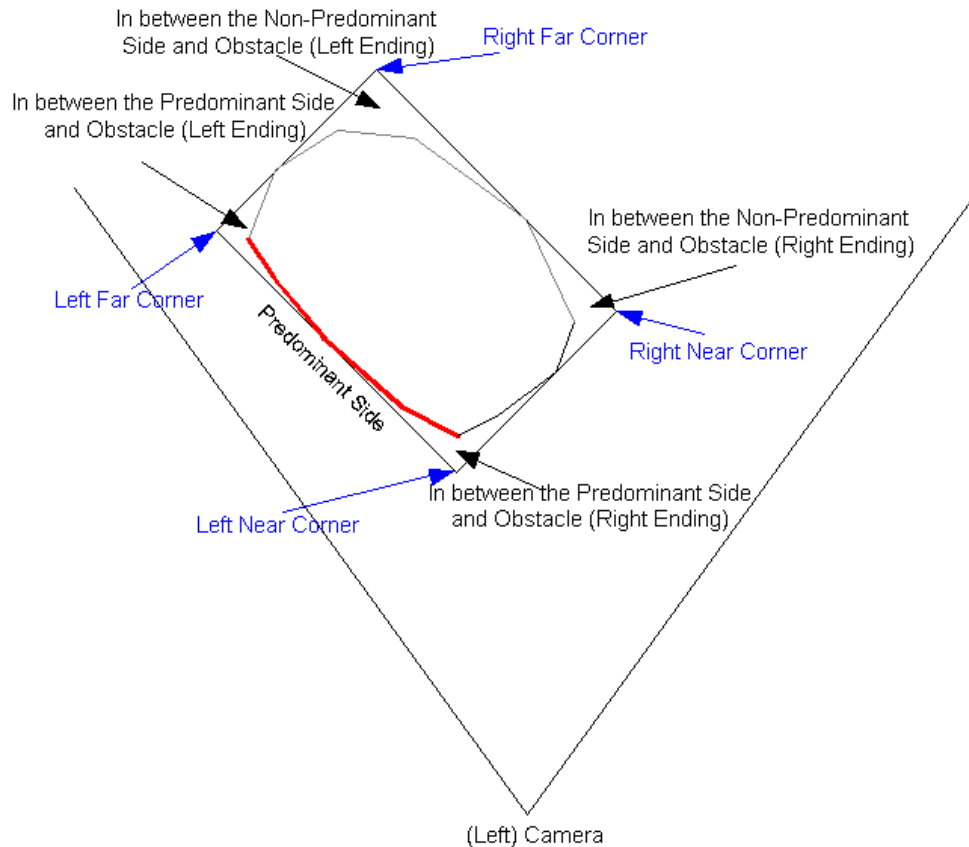


Fig. 6.3.18. Denumirea celor patru colțuri (cu albastru) și a spațiului liber în cele patru colțuri

Întotdeauna, numai două sau trei colțuri sunt vizibile sau chiar parțial vizibile. Unul sau două sunt întotdeauna nevizibile. Cele două colțuri care sunt adiacente laturii predominante sunt întotdeauna vizibile. Al treilea colț, care e posibil să fie vizibil, este adiacent laturii non-predominante. Condiția de vizibilitate este explicată pentru colțul care este la capătul drept al laturii non-predominante, în cazul când ostacolul este orientat-stânga:

punctele “Left Camera Position”, “Right Near Corner” (Fig. 6.3.18) și “Left Near Corner”, în această ordine, trebuie să fie în sens trigonometric (Fig. 6.3.19).

De fapt, condiția de vizibilitate verifică vizibilitatea laturii far/near (după caz), care este adiacentă colțului în cauză. Colțul în cauză este vizibil sau cel puțin parțial vizibil când latura adiacentă far/near este vizibilă. Punctele care trebuie să respecte sensul trigonometric, pentru fiecare din situații, sunt prezentate în tabelul următor:

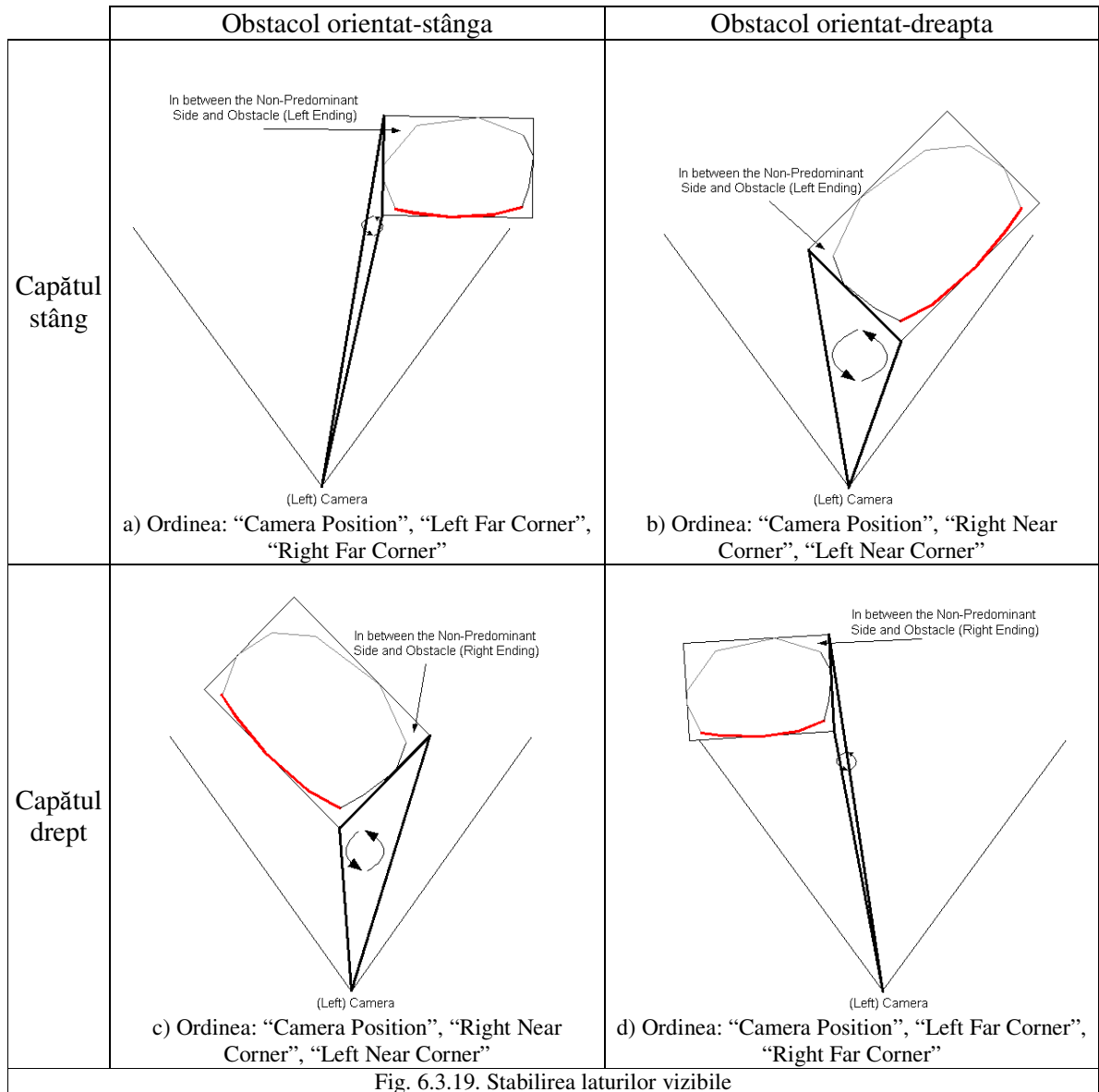


Fig. 6.3.19. Stabilirea laturilor vizibile

### Calcularea spațiului liber

În acest moment, prin *spațiu vizibil* se înțelege spațiul din cuboid care nu este ocluzat de însăși înfășurătoarea obstacolului curent, adică este vizibil din poziția camerei. Forma spațiului liber din colțuri nu este una simplă, folosindu-se o schemă de descompunere a acestuia în suprafețe simple. Descompunerea are trei niveluri. Suprafețele obținute la nivelul 3 sunt adunate pentru a obține suprafețele de la nivelul 2, iar suprafețele obținute la nivelul 2 sunt adunate pentru a obține suprafețele de la nivelul 1. Pentru toate aceste suprafețe, se calculează aria exprimată în număr de celule ale spațiului comprimat, dar și aria exprimată în metri pătrați. Celulele acoperite parțial se calculează ca atare, fracțional, nu se fac rotunjiri la cel mai apropiat număr întreg.



## Descompunerea la nivelul 1

După cum s-a mai discutat, spațiul liber din colțuri este delimitat de cuboid (un dreptunghi pe top-view) și înfășurătoarea punctelor 3D ale obstacolului curent. Înfășurătoarea are laturi vizibile (înspre cameră) și laturi nevizibile (înspre spatele obstacolului). Spațiul care nu este vizibil nu poate fi luat în calcul pentru că nu se știe ce se află în partea nevizibilă a obstacolului. Astfel, se pune problema spațiului vizibil din colțuri. În Fig. 6.3.20, liniile drepte punctate reprezintă cea mai din stânga și cea mai din dreapta rază optică a obstacolului. Acestea sunt determinate de cel mai din stânga și respectiv cel mai din dreapta vârf al părții vizibile a înfășurătorii.

În practică, pentru a lucra mai ușor, cuboidul și poziția camerei sunt orientate astfel încât laturile laterale (cea predominantă și cea non-predominantă) să devină verticale (Fig. 6.3.20).

Pentru fiecare colț, spațiul liber luat în calcul este descompus în:

- Spațiul liber dintre laturile vizibile ale înfășurătoarei și latura laterală. Care este latura laterală? Pentru cele două colțuri adiacente laturii predominante, aceasta este latura predominantă, iar pentru al treilea colț, adiacent laturii non-predominante, aceasta este latura non-predominantă. Acest spațiu liber este descompus în trapeze dreptunghice, având laturile paralele orizontale, latura perpendiculară fiind parte a laturii laterale și cea de-a patra latură fiind una din laturile înfășurătoarei. Altfel spus, laturile paralele sunt perpendicularele duse din vârfurile înfășurătoarei pe latura laterală (a cuboidului).
- Spațiul liber vizibil dincolo de ultimul colț vizibil al înfășurătoarei are o formă:
  - o trapezoidală, dacă este adiacent la latura predominantă (colțul far left în Fig. 6.3.20) și are: o latură paralelă inclusă în latura far, o latură paralelă care este perpendiculara dusă din ultimul vârf vizibil al înfășurătoarei pe latura predominantă, o latură neparalelă care este inclusă în latura predominantă și cea de-a patra latură care este inclusă în raza optică a ultimului vârf vizibil al înfășurătoarei; laturile paralele sunt orizontale;
  - o triunghiulară, dacă este adiacentă cu latura non-predominantă (colțul right near în Fig. 6.3.20) și are: o latură care este perpendiculara dusă din ultimul vârf vizibil a înfășurătorii pe latura non-predominantă, o latură care este inclusă în raza optică a ultimului punct vizibil a înfășurătorii și o latură inclusă în latura non-predominantă; în practică, pentru uniformizarea implementării, acest triunghi poate fi modelat ca un trapez dreptunghic cu laturile paralele orizontale (latura paralelă scurtă având lungime zero).

Sunt situații în care întreg colțul este vizibil (colțul "Left Near" în Fig. 6.3.20).

Mai condensat, putem exprima suprafața spațiului liber prin:

$$SCorner = SLevel = \sum_i STrapezoid_i$$

Trapezele sunt descrise de coordonatele vârfurilor lor. Modul prezentat pentru descompunerea spațiului produce doar trapeze orizontale, ale căror vârfuri sunt ușor de calculat. Nivelul 2 de descompunere se va ocupa de calcularea ariilor acestor trapeze, care se vor însuma apoi pentru fiecare colț.

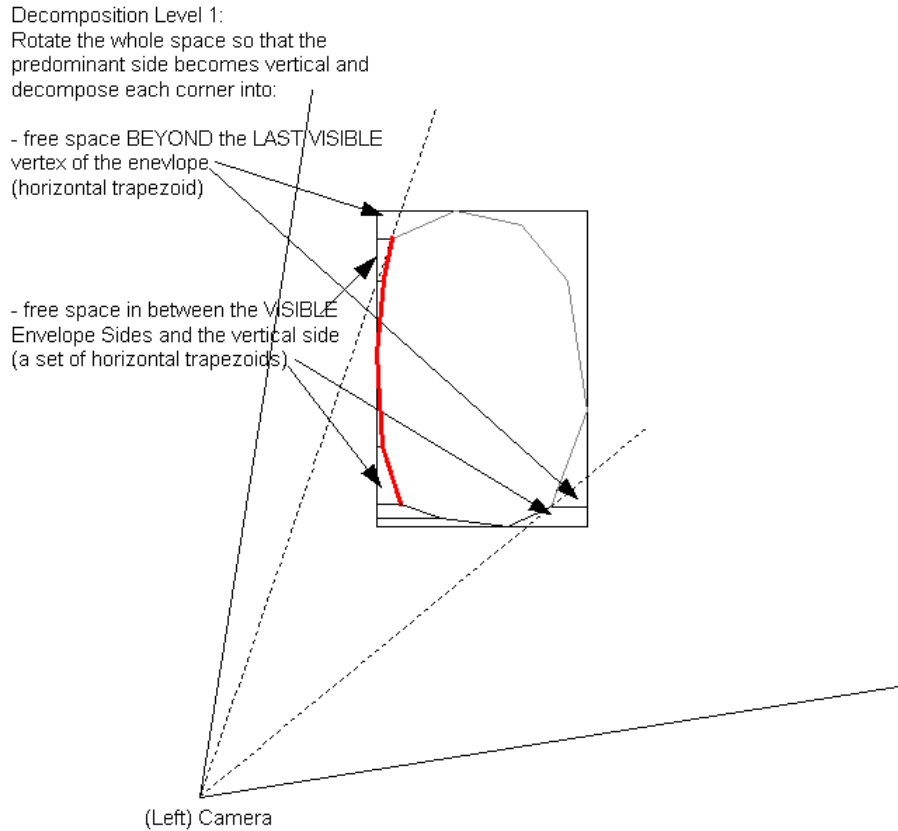


Fig. 6.3.20. Descompunerea la nivelul 1

## Descompunerea la nivelul 2

După descompunerea de nivel 1, se execută o contra-rotație pentru a ajunge înapoi în sistemul de coordonate inițial (Fig. 6.3.21). Fiecare trapez este descompus în patru triunghiuri cu câte o latură orizontală. Aceasta are loc în doi pași. În primul pas, trapezul este împărțit în două triunghiuri folosind o diagonală a trapezului. În pasul al doilea, fiecare triunghi astfel obținut este împărțit în două triunghiuri cu câte o latură orizontală, prin trasarea unei linii orizontale prin vârful din mijloc (considerat pe verticală) și care intersectează latura celorlalte două puncte.

Astfel avem:

$$STrapezoid_i = SLevel2 = \sum_{j=1}^4 STriangle_j$$

La fel ca trapezele, și triunghiurile sunt descrise de coordonatele vârfurilor lor, calculabile destul de ușor, având vârfuri re folosibile de la trapeze și unele vârfuri calculabile din drepte orizontale. Nivelul 3 de descompunere se va ocupa de calcularea ariilor acestor triunghiuri, care se vor însuma apoi pentru fiecare trapez și vor fi furnizate nivelului 1.

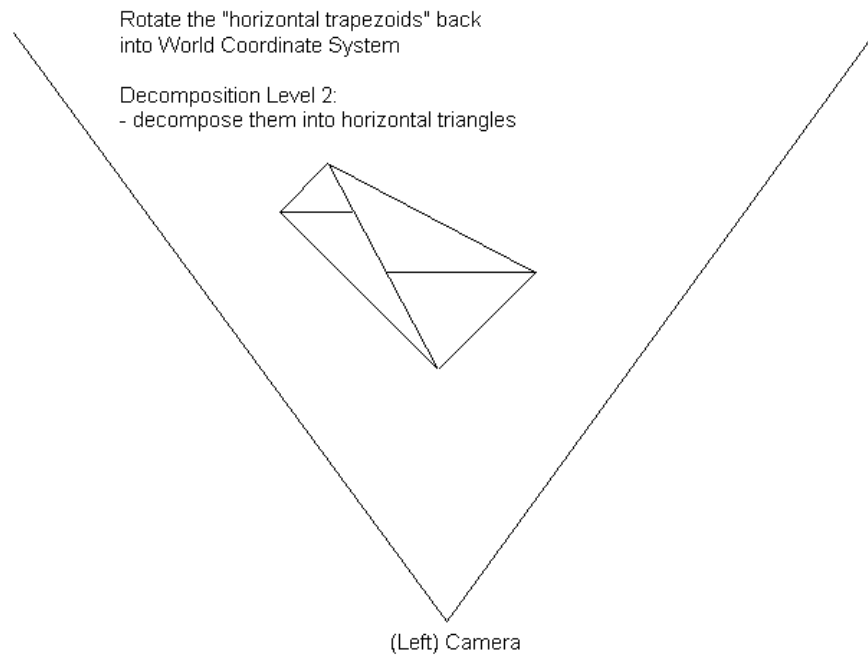


Fig. 6.3.21. Descompunerea la nivelul 2

### Descompunerea la nivelul 3

Mai departe, triunghiurile cu o latură orizontală sunt împărțite în rânduri ale spațiului comprimat (desenate cu verde în Fig. 6.3.22), rezultând trapeze orizontale, prin intersecția laturilor neorizontale ale triunghiurilor cu aceste rânduri. Pentru fiecare astfel de trapez, se calculează aria exprimată în metri pătrați (știindu-i coordonatele vârfurilor). Facem observația ca toate celulele de pe un rând au aceeași arie. Astfel, calcularea ariei acestor trapeze, exprimată în număr de celule, se face prin împărțirea ariei metrice a trapezului la aria metrică a unei celule.

Formal, putem scrie:

$$STriangle_j = SLevel3 = \sum_k S(row_k \cap Triangle_j)$$

Ariile rezultate sunt însumate și furnizate nivelului de descompunere 2, care la rândul său le va furniza nivelului 1.

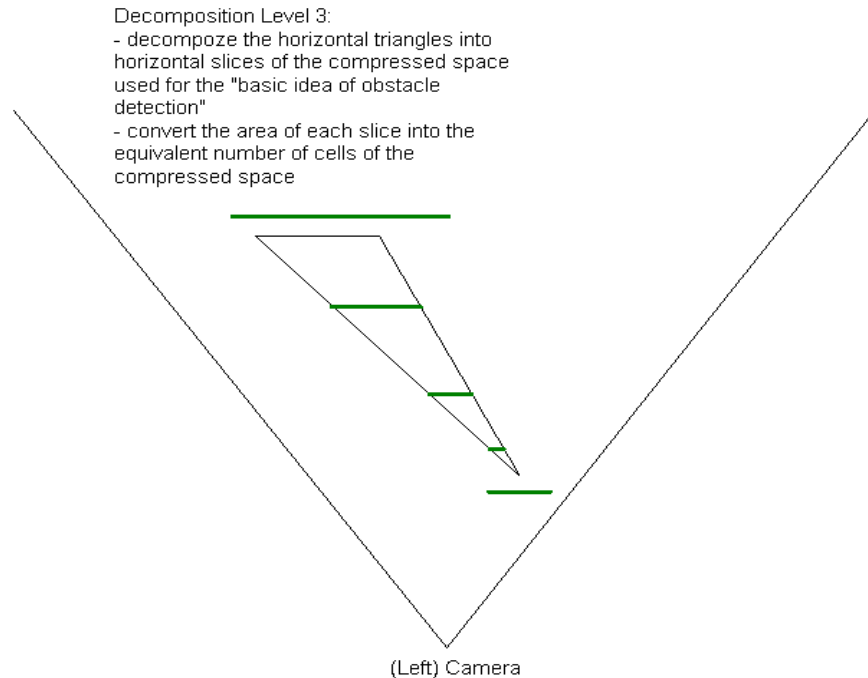


Fig. 6.3.22. Descompunerea la nivelul 3

## Analiza ocluziilor

Această secțiune descrie abordarea folosită pentru evaluarea ocluziilor obstacolelor, pentru ambele tipuri de ocluzii prezentate în Fig. 6.3.11 și Fig. 6.3.12. Aceste rezultate sunt folosite la nivelul de descompunere 3 (prezentat mai sus) pentru calcularea ariilor libere din colțurile cuboidelor obstacolelor, prin excluderea ariilor ocluzate. Detaliile folosirii efective vor fi prezentate în secțiunea următoare.

Ocluzia unui obstacol este exprimată în spațiul imaginii stângi, prin perechea (*LeftOcclusionColumn*, *RightOcclusionColumn*). Între cele două coloane, obstacolul nu este ocluzat; ocluzia are loc la stânga de *LeftOcclusionColumn* și la dreapta de *RightOcclusionColumn*.

După cum se poate vedea în Fig. 6.3.11, un obstacol din fundal poate fi parțial ocluzat de un obstacol din prim-plan. În procesul de evaluare a ocluziilor, obstacolul din fundal va fi numit *obstacol curent*, fiindcă este obstacolul pentru care se evaluează ocluzia, iar obstacolul din prim-plan va fi numit *obstacol ocluzant*.

După cum s-a văzut în secțiunea anterioară, sunt două situații în care un obstacol este ocluzat:

(1) din cauza altui sau altor două obstacole:

a) un obstacol ocluzant la stânga; ocluzia e descrisă de *LeftOcclusionColumn\_OtherObstacle*, cu înțelesul că ocluzia are loc la stânga acestei coloane; pentru cazul în care nu are loc ocluzie la stânga, se poate considera *LeftOcclusionColumn\_OtherObstacle* =  $-\infty$ ;

b) un obstacol ocluzant la dreapta; ocluzia e descrisă de *RightOcclusionColumn\_OtherObstacle*, cu înțelesul că ocluzia are loc la dreapta acestei

coloane; pentru cazul în care nu are loc ocluzie la dreapta, se poate considera  $RightOcclusionColumn\_OtherObstacle = +\infty$ ;

c) Notă: există și situații de ocluzie parțială, de exemplu când un obstacol ocluzant este scund și în spate se vede obstacolul curent. Este totuși rezonabil să se presupună că partea rămasă vizibilă este suficientă pentru a se surprinde forma top-view a obstacolului curent. Oricum, s-ar putea face un studiu mai detaliat pentru dezavantajele acestei presupuneri și pentru soluții mai avansate.

(2) din cauza FOV al sistemului stereo: descris de perechea ( $LeftOcclusionColumn\_stereoFOV$ ,  $RightOcclusionColumn\_stereoFOV$ ).

Astfel, fuzionând cele două situații, ocluzia obstacolului curent este descrisă de perechea ( $LeftOcclusionColumn$ ,  $RightOcclusionColumn$ ) unde:

$$LeftOcclusionColumn = \max (LeftOcclusionColumn\_stereoFOV, LeftOcclusionColumn\_OtherObstacle)$$

și

$$RightOcclusionColumn = \min (RightOcclusionColumn\_stereoFOV, RightOcclusionColumn\_OtherObstacle)$$

Observație: în cazul în care nu este nicio ocluzie în spațiul imaginii (nici din cauza vreunui obstacol ocluzant, nici din cauza limitelor stereo FOV), ocluzia este considerată ca ( $LeftOcclusionColumn\_stereoFOV$ ,  $RightOcclusionColumn\_stereoFOV$ ). Decizia este importantă deoarece cuboidul poate fi ocluzat de FOV chiar și atunci când obstacolul este cu totul vizibil în imagine. Aceasta se poate întâmpla deoarece cuboidul înscrie punctele obstacolului devenind de regulă mai mare decât obstacolul.

În continuare, se prezintă detaliat cele două situații.

### Ocluzii din cauza altor obstacole

Ocluzia poate avea loc în partea stânga sau în partea dreaptă a obstacolului curent. Evaluarea pentru partea stângă constă în analiza imaginii stângi (comprimate), în jurul părții stângi a obstacolului curent, și verificarea dacă acolo sunt puncte 3D (celule) care aparțin altui obstacol și sunt situate la profunzime mai mică decât cele ale părții stângi a obstacolului curent. Procesarea având loc în spațiul imaginii comprimate, are avantajele:

- concentrează informația,
- umple spațiile mici fără reconstrucție,
- crește viteza de procesare.

Mai exact, se folosesc imaginea comprimată etichetată (pentru evidența obstacolelor detectate) și imaginea Row-Buffer (pentru a avea o măsură a profunzimii).

După cum se arată în Fig. 6.3.23, pe imaginea (stânga) comprimată, în jurul părții stângi a obstacolului curent, se găsesc celule care aparțin altui obstacol și au profunzime mai mica. Prin “în jurul părții stângi” se înțelege coloana de celule aflate la stânga celei mai din stânga coloane a obstacolului curent.

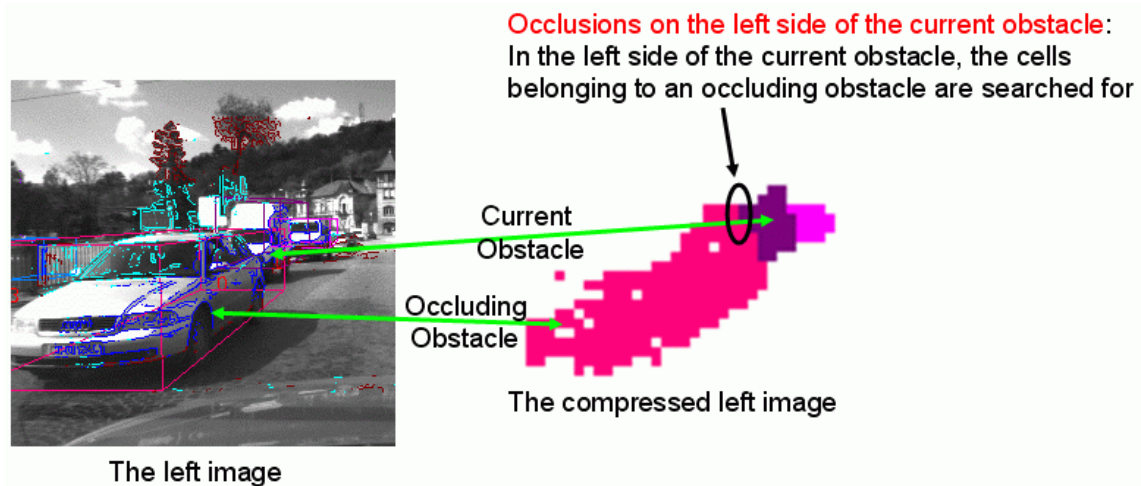


Fig. 6.3.23. Analiza ocluziei în partea stânga a obstacolului curent

Analiza în partea dreaptă a obstacolului curent este puțin diferită pentru că, în spațiul comprimat al imaginii stângi, ar apărea o lipsă a reconstrucției între obstacolul curent și obstacolul ocluzant (Fig. 6.3.24). Acest fenomen este normal și este cunoscut sub numele de *stereo shadow*: în imaginea stângă este vizibilă o parte a obstacolului curent care nu este vizibilă în imaginea dreaptă, fiind ocluzată doar în imaginea dreaptă, nu și în imaginea stângă. O soluție ar fi calcularea imaginii drepte comprimate. Aceasta s-ar face prin deplasarea celulelor imaginii stângi comprimate spre stânga, conform cu disparitățile lor, mai exact cu disparitățile comprimate. Soluția implementată simulează aceasta doar pentru celulele din dreapta obstacolului curent. Se caută celulele ocluzante, în spațiul imaginii stângi, în primele  $k$  coloane care se află la dreapta părții drepte a obstacolului curent:

$$k = MaxDisparity - RightSideDisparity$$

unde,  $MaxDisparity$  este disparitatea maximă (comprimate) folosită de reconstrucția stereo,  $RightSideDisparity$  este disparitatea (comprimate) a părții drepte a obstacolului curent. Astfel,  $k$  are semnificația de lățimea (comprimate) cea mai mare a stereo shadow.

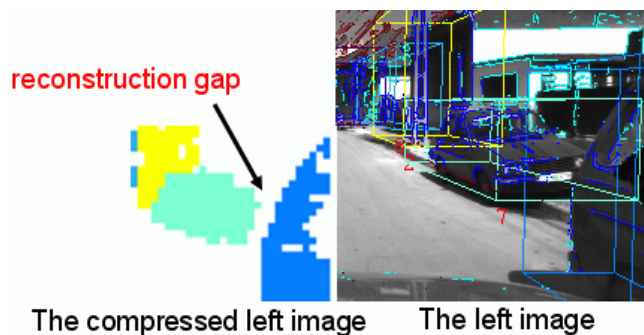


Fig. 6.3.24. Analiza ocluziei în partea dreaptă a obstacolului curent

## Ocluzii din cauza FOV al sistemului stereo

Un caz special de stereo shadow, adică de porțiuni ale scenei vizibile doar în una dintre camere, este și acela cauzat de amplasarea spațială diferită a celor două camere. Astfel, în partea cea mai stângă a imaginii stângi, va fi vizibilă o porțiune care nu este vizibilă în imaginea dreaptă. În Fig. 6.3.25, pe top-view, sunt desenate cele două FOV (a fiecărei camere) și FOV stereo (comun – în culoare roșie). Limita dreaptă a FOV stereo este dată de limita dreaptă FOV a camerei stângi. Limita stângă a FOV stereo este data de limita stângă FOV a camerei drepte. Coloanele dincolo de care apare acest fenomen se numesc *coloane ocluzante*. În practică, se ia în considerare și faptul că reconstrucția stereo nu se face pentru primele cele mai din stânga și ultimele cele mai din dreapta coloane de pixeli (de exemplu 5 coloane).

Coloana ocluzantă datorată limitei drepte a FOV stereo este ușor de calculat:

$$\text{RightOcclusionColumn\_stereoFOV} = \text{ImageWidth} - 5$$

Coloana ocluzantă datorată limitei stângi a FOV stereo este:

$$\text{LeftOcclusionColumn\_stereoFOV} = 5 + \text{Disparity of obstacle},$$

unde  $\text{Disparity of obstacle} = b * f / Z_{\text{Obstacle}}$ ,  $b$  este distanța dintre camere (exprimată metric),  $f$  este distanța focală a camerei (exprimată în pixeli) și  $Z_{\text{Obstacle}}$  este profunzimea obstacolului (exprimată metric).

Cu alte cuvinte,  $\text{LeftOcclusionColumn\_stereoFOV}$  este reproiecția în spațiul imaginii stângi, a punctului în care limita stângă a FOV a camerei drepte intersectează cuboidul obstacolului curent. În Fig. 6.3.25, raza optică a acestui punct este desenată cu o linie de culoare roșie subțire.

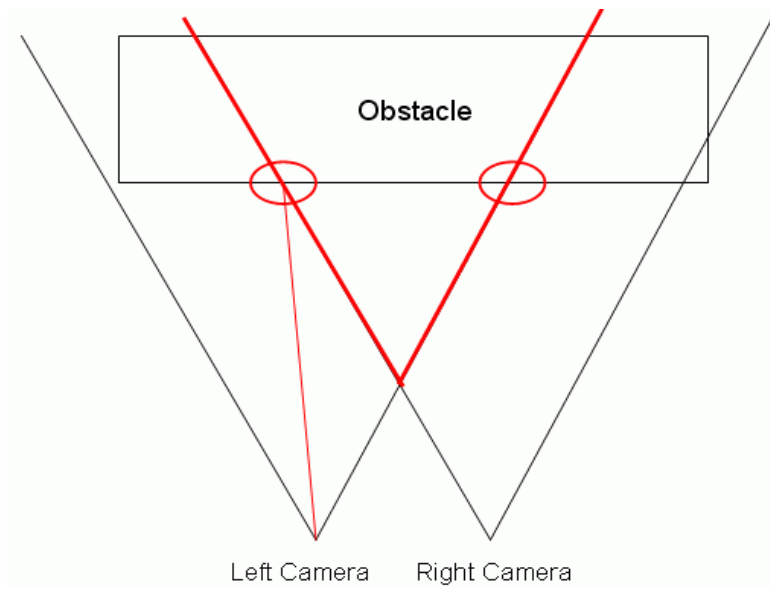


Fig. 6.3.25. Vizibilitatea unui obstacol în cele două camere

## Folosirea ocluziilor

Coloanele ocluzante, în spațiul imaginii stânga, sunt reprezentate de raze optice desenate cu roșu în Fig. 6.3.26. La nivelul de descompunere 3, suprafața fiecărui trapez orizontal este intersectată de cele două raze optice. Astfel, se calculează fracțiunea vizibilă a fiecărui trapez. Apoi fracțiunea vizibilă e aplicată ariilor trapezului, atât celei exprimate metric, cât și celei exprimate în număr de celule.

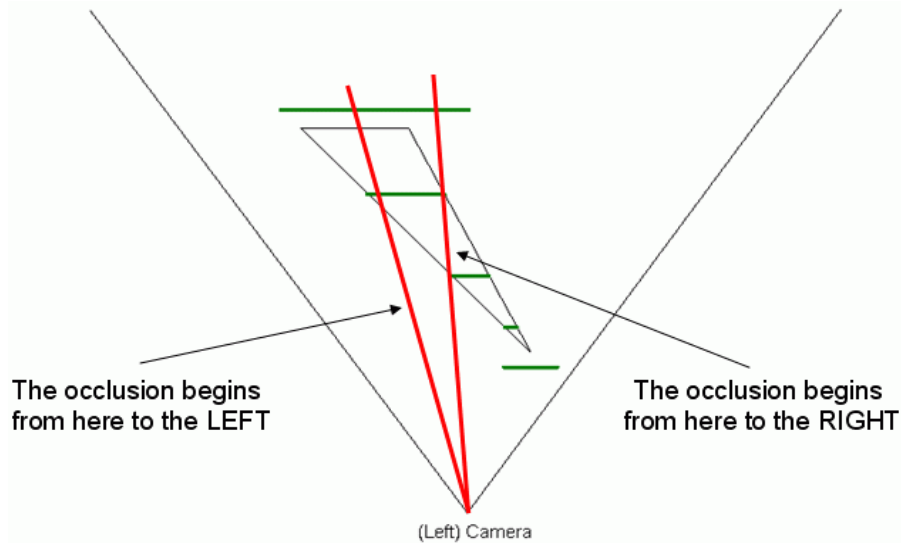


Fig. 6.3.26. Folosirea ocluziilor, la nivelul 3 de descompunere

## Fragmentarea în obstacole cuboidale

Din ultimele două secțiuni, sumarizând calcularea spațiului liber dintr-un colț al unui cuboid, avem:

$$SCorner = SLevel1 = \sum_i STrapezoid_i$$

$$STrapezoid_i = SLevel2 = \sum_{j=1}^4 STriangle_j$$

$$STriangle_j = SLevel3 = \sum_k S(row_k \cap Triangle_j \cap non\_occluded\_FOV)$$

Atunci când aria celui mai mare colț gol este mai mare de un prag de 10 celule, se consideră că obstacolul nu este modelabil printr-un cuboid. Soluția este să modelăm obstacolul prin mai multe cuboide, mai multe sub-obstacole, fiecare fiind modelabil printr-un cuboid. Practic, obstacolul inițial este împărțit în două obstacole mai mici, iar acestea sunt la rândul lor analizate în vederea determinării unei orientări și evaluării dacă sunt modelabile prin cuboide. Procesul se repetă recursiv până când nu mai sunt necesare fragmentări.



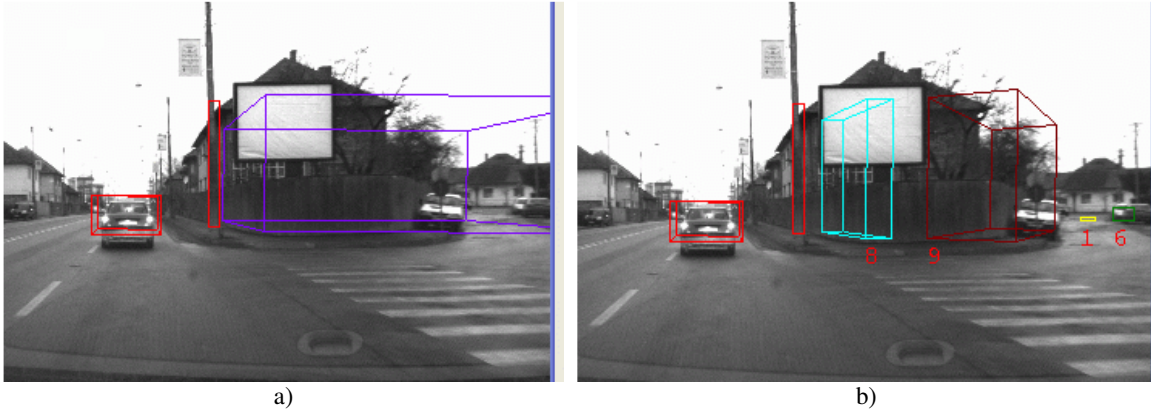


Fig. 6.3.27. Cu (a) și fără (b) folosirea criteriului de fragmentare bazat pe convexități

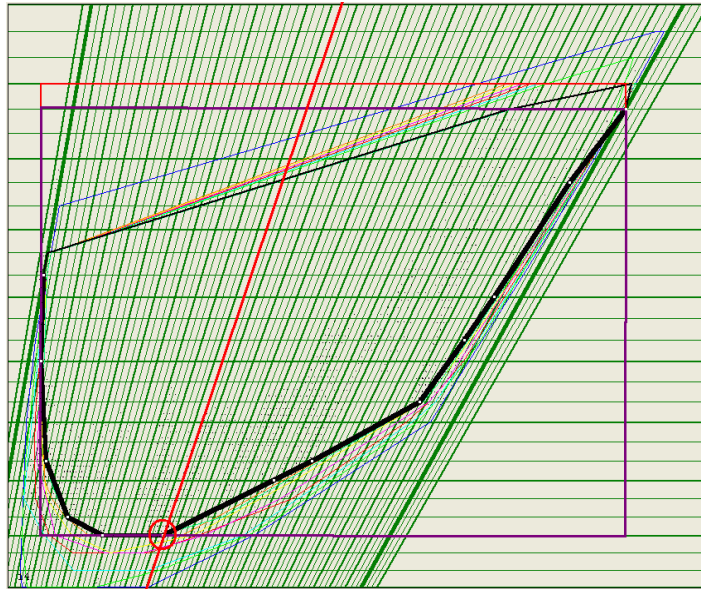
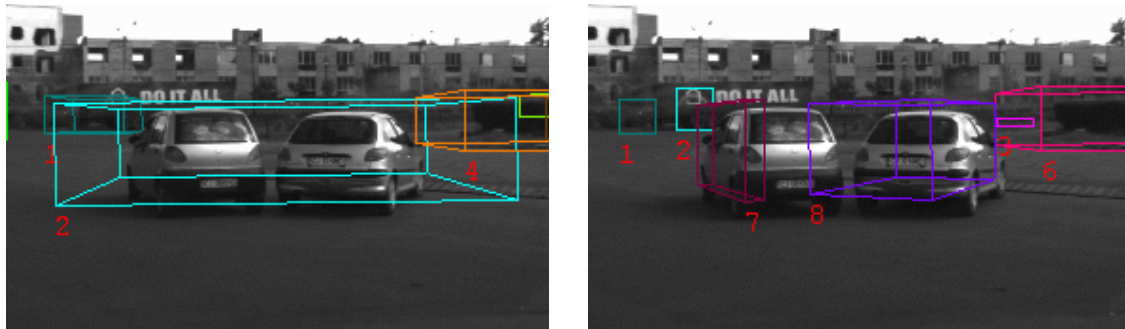


Fig. 6.3.28. Alegerea punctului de fragmentare pentru exemplul din Fig. 6.3.27

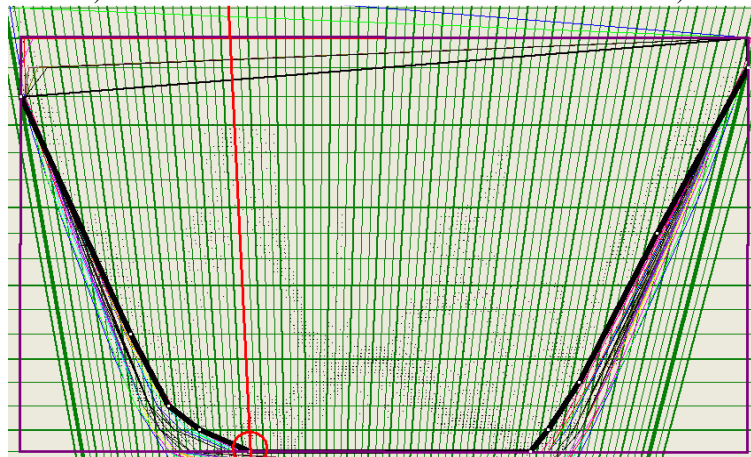
Se pune problema alegerii punctului de fragmentare. Acesta este unul dintre punctele de capăt ale înfășurătoarei vizibile, care delimitează colțul gol cel mai mare. Fiind vorba de două astfel de puncte, este ales cel care evită împărțirea într-un obstacol mare și unul mic. În exemplul prezentat este ales punctul din stânga. În particular, în acest exemplu, punctul din dreapta ar fi împărțit obstacolul într-un obstacol care ar fi fost obstacolul inițial și un obstacol de dimensiune zero. Fragmentarea se face de către raza optică ce trece prin punctul de fragmentare. Desigur, în spațiul orizontal comprimat, aceasta este o coloană, o verticală.

Alte exemple:

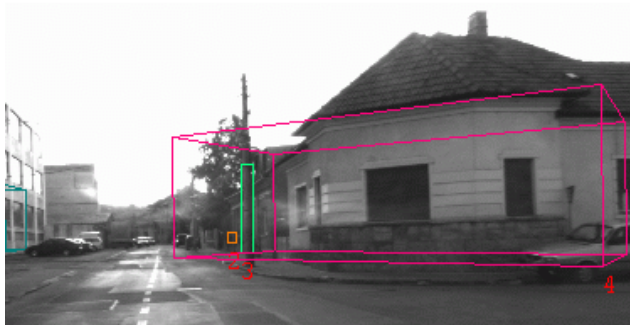


a)

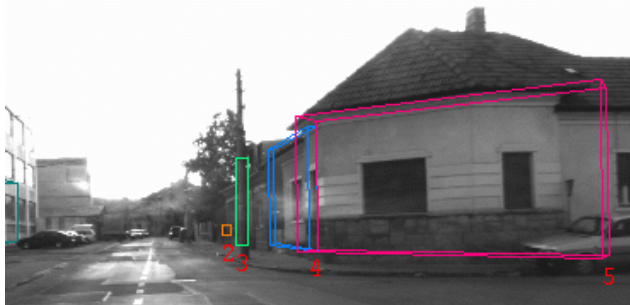
b)



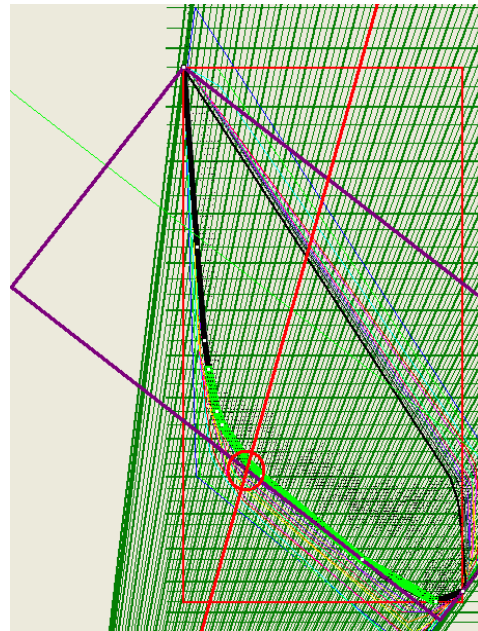
c) vederea top-view a înfășurătoarei și a punctului și razei optice a fragmentării  
Fig. 6.3.29. Cu (a) și fără (b) folosirea criteriului de fragmentare bazat pe convexități



a)



b)



c) vederea top-view a înfășurătoarei și a punctului și razei optice a fragmentării  
Fig. 6.3.30. Cu (a) și fără (b) folosirea criteriului de fragmentare bazat pe convexități

## Concluzii

Desideratul de a avea obstacole reale detectate și descrise de cuboide cât mai fidele a fost dus la limita sa maximă: pe top-view, forma părții vizibile a oricărui obstacol este modelată printr-o linie poligonală având doar unghiuri drepte, adică printr-un dreptunghi (baza cuboidului asociat). Așadar, considerând împreună criteriul fragmentării în cazul concavităților și criteriul fragmentării în cazul convexităților care nu sunt în unghi drept, zonele ocupate din scenă sunt descompuse acum în obstacole fără concavități și având doar unghiuri drepte.

O problemă importantă a detecției de obstacole este că nu se cunoaște nici forma obstacolelor și nici erorile de reconstrucție. Lucrarea de față folosește diferite indicii și tehnici pentru a diminua această problemă.

## Posibile îmbunătățiri

Pentru stabilizarea detecției în cadre succesive, s-ar putea urmări deciziile de fragmentare și punctele de fragmentare. Aceasta ar ajuta la eliminarea deciziilor sporadice, atât cele pozitive (de fragmentare), cât și cele negative (de nefragmentare), dar și a fluctuațiilor punctelor de fragmentare.

Fragmentarea obstacolelor în sub-obstacole cuboidale, prin analiza formei înfășurătorii convexe, a fost publicată în: **C. Pocol**, S. Nedevschi, Marc-M. Meinecke, “Obstacle Detection Based on Dense Stereovision for Urban ACC Systems,” in *Proc. Workshop on Intelligent Transportation (WIT’08)*, Hamburg, Germany, 18-19 March 2008.

## 6.4. Alte procesări

Există și procesări care nu modifică cuboidele obstacolelor: filtrări și clasificări. Acestea sunt prezentate în cele ce urmează.

### 6.4.1 Absorbția obstacolelor

Algoritmul de grupare a punctelor în obstacole, prezentat într-o secțiune anterioară, își propune dezideratul de a detecta obstacole generice, neavând cunoștințe legate de forma acestora. În încercarea sa de a nu grupa împreună obstacole din prim-plan cu obstacole de fundal, tinde să folosească mai mult începuturile coloanelor spațiului comprimat, începând de la frontiera din față a obstacolelor. Astfel că celulele din interiorul obstacolelor rămân ne-etichetate, urmând să fie grupate ulterior în noi obstacole (Fig. 6.4.1 a și c). Acest fenomen apare doar la obstacole aflate la profunzime mică, care se întind pe multe rânduri ale spațiului comprimat.

În urma detectării tuturor obstacolelor și a îmbunătățirii lor prin diferite tehnici (rafinarea frontierei, determinarea orientării, trei tipuri de fragmentări), se pot face noi raționamente privind structura obstacolelor din scenă. Este plauzibil să se considere că tot ce se află în interiorul unui obstacol cuboidal aparține aceluși obstacol. Astfel, dacă în interior ajunge să fie detectat un alt obstacol, mai mic, acesta trebuie inclus, absorbit, de obstacolul mai mare (Fig. 6.4.1 b și d). De altfel, astfel de cazuri erau de așteptat: de exemplu partea din față a unei mașini e reconstruită, partea de deasupra mașinii e și ea reconstruită, având trăsături, dar partea reprezentând capota și parbrizul este slab reconstruită, sugerând un spațiu gol între un obstacol aflat mai în față și altul aflat mai în spate.

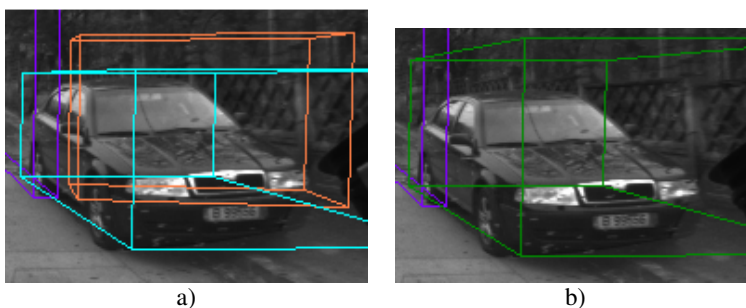
Soluția e simplă: se face un test de incluziune între cuboide, iar acesta trebuie să releve o intersecție semnificativă (cel puțin 80% din volumul cuboidului mai mic să fie inclus în cel mai mare) pentru ca obstacolul mai mic să fie absorbit de cel mai mare:

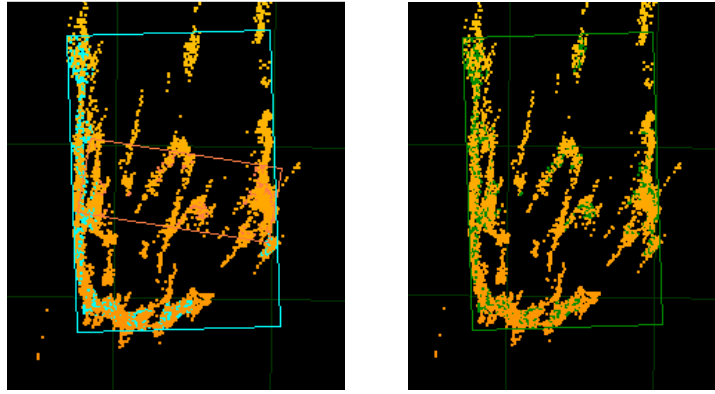
în cazul în care volumul obstacolului  $i$  este mai mare decât volumul obstacolului  $j$ , avem:

*if* ( $V_{Obstacle_i} \cap V_{Obstacle_j} > 80\% V_{Obstacle_j}$ ) *then absorb*  $Obstacle_j$  *into*  $Obstacle_i$

Similar, în cazul în care volumul obstacolului  $j$  este mai mare decât volumul obstacolului  $i$ , avem:

*if* ( $V_{Obstacle_i} \cap V_{Obstacle_j} > 80\% V_{Obstacle_i}$ ) *then absorb*  $Obstacle_i$  *into*  $Obstacle_j$





c)

d)

Fig. 6.4.1 Absorbția obstacolelor



## 6.4.2 Clasificarea obstacolelor de fundal

La un moment dat, în scenă pot fi identificate multe obstacole, de ordinul zecilor. Există mai multe motive pentru care, din multitudinea de obstacole, să fie identificate cele care sunt de fundal, adică în spațiul (polar! al) imaginii sunt în spatele obstacolelor din prim-plan, dar sunt suficient de vizibile și de distincte pentru a putea fi detectate în spațiul comprimat orizontal.

Motivele principale ale necesității clasificării obstacolelor de fundal sunt:

- legat de semantica pentru ego-car: din moment ce sunt în spatele obstacolelor din prim-plan, nu prezintă un pericol;
- totuși, obstacolele în mișcare pot ajunge în calea ego-car, de aceea o ignorare a lor ar fi greșită;
- unele obstacole statice de fundal, odată cu deplasarea ego-car, pot deveni obstacole de prim-plan, astfel că pot furniza informații încă dinainte de a deveni obstacole de prim-plan;
- de regulă au o acuratețe mai scăzută și o fluctuație mai mare a detecției în cadre succesive; astfel pot deveni problematice unor algoritmi de tracking;
- fiind parțial vizibile, pot crea probleme algoritmilor de clasificare bazați pe aparența obstacolelor în imagine;
- chiar și pentru pașii de post-procesare ai detecției de obstacole, pași de rafinare și fragmentare, poate fi utilă o astfel de informație;
- ar putea fi utilizate la un nivel superior de evaluare a scenei, de exemplu obstacolele înalte de-o parte și de alta a traiectoriei ego-car, ar putea sugera existența unor ziduri, a unei scene de tip urban.

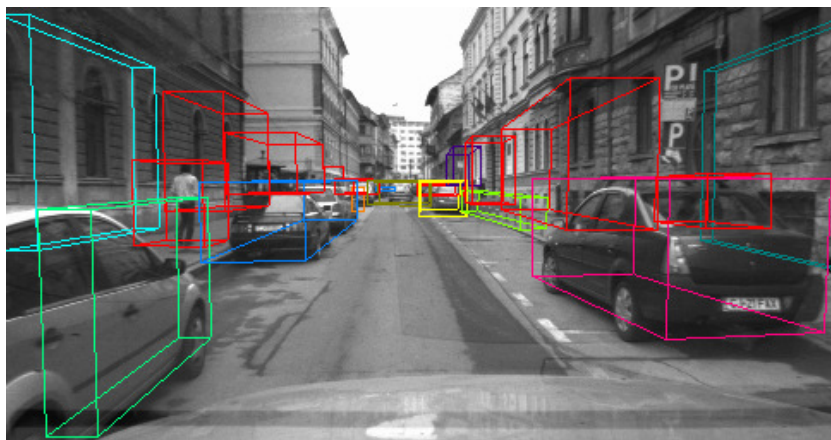


Fig. 6.4.2 Clasificarea obstacolelor de fundal (desenate în roșu)

Un obstacol e considerat de fundal dacă, în spațiul comprimat al imaginii stângi, cel puțin 80% din coloanele sale au sub ele celule aparținând unor obstacole ocluzante, adică cu profunzime mai mică decât obstacolul curent. Obstacolele mici, aflate pe sol (de exemplu un stâlpișor, o piatră mai mare), nu sunt considerate ocluzante, ci trebuie să aibă o înălțime de minim 500 mm pentru aceasta. În cazul în care un obstacol ar avea într-o parte multă informație 3D, și în cealaltă parte puțină informație, coloanele sale nu pot avea aceeași pondere în clasificarea obstacolului ca fiind de fundal. Astfel, fiecare coloană este ponderată cu numărul de celule din coloana corespondentă a spațiului orizontal al obstacolului curent, și abia apoi se compară cu acel prag de 80%.

### 6.4.3 Filtrarea obstacolelor

În cadrul reconstrucției stereo, pasul de stereo matching poate găsi greșit disparitatea unui grup mai mare de puncte 3D. Dacă grupul e suficient de mare, pot să apară mici obstacole fantomă în părți ale scenei unde nu este niciun obstacol real. Din fericire aceste obstacole sunt mici.

Separarea sol-obstacole are și ea o acuratețe limitată, putând cu ușurință să catalogheze grupuri mai mari de puncte, aflate puțin deasupra solului, drept puncte aparținând spațiului de interes al obstacolelor. Și în acest caz, astfel de puncte pot conduce la detecția unor obstacole false, dar de dimensiuni mici.

O altă situație în care pot să apară obstacole mici este în urma pașilor de fragmentare. De regulă, aceste obstacole sunt doar înguste și puțin adânci, dar dacă sunt reale atunci înălțimea e considerabilă.

Astfel se impune un raționament de decizie a situației când un obstacol e rejectabil și când nu. Măsurătorile luate în calcul sunt în spațiul orizontal comprimat, în spațiul imaginii comprimate și în spațiul cartezian 3D. Măsurătorile din spațiile comprimate au avantajul normalizării cu profunzimea, măsurătorile 3D ale obstacolelor detectate de la profunzimi mai mari neavând acuratețea celor de la profunzimi mai mici. Măsurătorile 3D au avantajul raționării în spațiul obstacolelor reale. Câteva raționamente pro eliminării unui obstacol sunt:

- înălțimea 3D este mai mică decât un prag (de exemplu 250 mm);
- înălțimea în spațiul imaginii comprimate este mai mică decât un prag (de exemplu două rânduri);
- lățimea în spațiul orizontal comprimat este mai mică decât un prag (de exemplu 3 coloane);
- profunzimea în spațiul orizontal comprimat este mai mică decât un prag (de exemplu 2 rânduri);
- pentru obstacolele de fundal, aceste praguri sunt puțin mai mari.

Filtrarea obstacolelor este apelată în următoarele momente:

- după gruparea celulelor spațiului orizontal comprimat în obstacole;
- după fragmentarea pe criterii de concavitate;
- după fragmentarea pe verticală;
- după fragmentarea pe criterii de convexitate.

## 6.5. Concluzii

Se prezintă o abordare originală pentru detecția de obstacole din cadre stereo individuale. Abordarea constă dintr-o suită de pași și subpași de procesare.

Pașii abordării sunt împărțiți pe mai multe direcții:

- Evidențierea zonelor ocupate:  
Punctele 3D sunt reprezentate într-un spațiu comprimat orizontal (de tip grid top-view), ținând seama de posibilitățile reale de percepție a scenei prin intermediul stereo viziunii (spațiul U-disparity). În acest spațiu, celulele sunt grupate folosind criterii de vecinătate și de densitate. Au fost luate măsurile necesare compensării diferiților factori care

influențează vecinătatea și densitatea. Urmează o îmbunătățire a frontierei fiecărui obstacol, prin procesări la nivel de coloane individuale și la nivel de obstacol.

- Procesări pe direcția verticală:  
Pentru detecția fidelă a vehiculelor mai înalte decât autovehiculul propriu, de exemplu a camioanelor, obstacolele sunt extinse drept în sus, în măsura în care sunt puncte 3D, dar nu mai sus de 4,5 m. Alte două procesări stabilesc limitele verticale corecte, prin fragmentări și rafinări.
- Valorificarea formelor obstacolelor:  
Adesea, mai multe obstacole apropiate sunt grupate împreună, dar forma frontierei grupului poate oferi indicii suplimentare. Astfel, dacă frontiera vizibilă (înspre cameră) are concavități, acestea sunt folosite pentru determinarea punctelor de fragmentare în obstacolele constitutive. Apoi, partea cvazi-liniară cea mai lungă a frontierei vizibile poate indica orientarea fiecărui obstacol. După acest pas fiecare obstacol este modelat printr-un cuboid, posibil orientat. În pasul următor, se evaluează fidelitatea cuboidului, prin suprafața spațiului liber înglobat între laturile vizibile ale cuboidului și frontiera vizibilă. Dacă această suprafață e semnificativă, atunci se aplică un algoritm de fragmentare. Desigur, s-a ținut seama de cazul în care suprafața liberă este cauzată de ocluziile din partea unor obstacole de prim-plan sau de cazul în care o parte a cuboidului nu este vizibilă în ambele camere ale sistemului stereo.

Majoritatea pașilor și sub-pașilor sunt de concepție și/sau soluționare originală. De multe ori, prin aprofundarea unui pas de procesare, s-au deschis direcții noi. De exemplu, rafinarea limitelor obstacolelor a condus la studiul frontierei și a înfășurătoarei convexe, iar mai apoi s-au dezvoltat algoritmi pentru fragmentarea grupurilor de obstacole și determinarea orientării obstacolelor individuale. În final, s-a ajuns la un grad înalt de modelare fidelă a obstacolelor reale prin cuboide.



## Capitolul 7. Concluzii, contribuții

Domeniul tezei este acela al vederii artificiale pentru autovehicule rutiere. Mai exact, obiectivul principal vizează detecția de obstacole generice din cadre stereoscopice individuale, în ipoteza existenței unui modul care realizează separarea sol-obstacole.

De-a lungul realizării tezei, studiul diferitelor aspecte ale domeniului s-a impus și s-a dovedit a fi un proces iterativ. Astfel, capitolul 2 analizează problematica sistemelor avansate de asistare a conducerii (ADAS = Advanced Driver Assistance Systems) la nivel structural, la nivel senzorial și prin detalierea unor sisteme concrete. S-a concluzionat că senzorii bazați pe stereo viziune sunt cei mai potriviți pentru detecția de obstacole în scene de trafic urban. În astfel de scene, profunzimea tipică este relativ mică (sub 30 de metri), iar complexitatea scenei este adesea prohibitivă pentru alte tipuri de senzori. Stereo viziunea oferă o cantitate mult mai mare de măsurători 3D decât senzorii de tip radar sau lidar. Vederea artificială are și avantajul că întreaga infrastructură rutieră este proiectată și realizată pentru a fi percepută în mod vizual. S-a propus și o schemă generică, organizată pe mai multe nivele, pentru realizarea de astfel de sisteme bazate pe stereo viziune.

Întregul capitol 3 este dedicat înțelegerii stereo viziunii. Astfel, au fost experimentați și analizați toți pașii necesari stereo viziunii: înțelegerea parametrilor geometrici ai camerelor, calibrarea camerelor (cu [OpenCV]), rectificarea imaginilor (cu OpenCV), alegerea trăsăturilor, stereo matching (cu OpenCV) și reconstrucția 3D. S-a demonstrat că, în ciuda cantității mari de calcule, stereo matching-ul rulează în timp real, pe CPU (40ms pe procesor single core din 2007, rezoluție VGA). Asupra stereo reconstrucției s-a făcut o analiză originală legată de aspecte precum: acuratețea nativă și modelarea acesteia, erori și clase de magnitudini ale erorilor, cantitatea și distribuția punctelor 3D. Tot aici, s-a pus accentul pe rezoluția cu care sensorul de stereo viziune percepe lumea reală. S-a arătat că pe direcțiile orizontală și verticală, percepția este una polară și că în spațiul cartezian rezoluția scade liniar cu profunzimea. Pe direcția profunzimii, percepția se face prin disparități, rezultând în spațiul cartezian o rezoluție care scade cvazi-pătratic cu profunzimea. Astfel, una din contribuțiile principale ale tezei este evidențierea și accentuarea în mod repetat a posibilităților de percepție a scenei prin stereo viziune.

O detecție fiabilă a obstacolelor este capabilă să trateze diverse obstacole, indiferent de forma 3D. Pentru descrierea localizării și a dimensiunilor, se impune nevoia unor reprezentări, a unor modele care să descrie obstacole generice. În decursul realizării tezei s-a simțit nevoia implementării a diferite modele de reprezentare a obstacolelor. Capitolul 4 detaliază modelele implementate: dreptunghiular neorientat, dreptunghiular orientat, octogonal, poligonal, cuboidal neorientat, cuboidal orientat, poliedral, multi-element și curbat; dar și altele din literatură: poligonal cu rezoluție variabilă și înălțime constantă, poligonal cu rezoluție constantă și înălțime variabilă (stixel), cuboidal polar etc. Majoritatea au fost implementate și analizate în cadrul tezei de față, mai mult ca urmare a necesității decât a inspirației din literatură. Tot aici, se prezintă sistemele de coordonate folosite pentru modelarea domeniului problemei (scena reală) și domeniului soluției (stereo viziunea). Astfel, se prezintă sistemul de coordonate cartezian 3D potrivit pentru descrierea scenei reale, dar și sistemul de coordonate 3D nativ al sensorului de stereo viziune, potrivit pentru posibilitățile reale de percepție a scenei.

Capitolul 5 face un studiu a celor mai importante abordări din literatura de specialitate, care încearcă să rezolve aceleași probleme ca și teza de față. Sunt prezentate abordări de viziune mono de intensitate, mono color, secvențe de imagini și imagini stereo. În cazul stereo viziunii, este originală gruparea abordărilor în funcție de spațiul de procesare folosit preponderent. S-a observat că, pentru simplificarea procesărilor și compactarea datelor, se folosesc adesea proiecții particulare, fie ale spațiului cartezian 3D, fie ale spațiului nativ al camerei stereo. Originale sunt și comparațiile între spațiile de procesare și comparațiile între cele mai importante echipe de cercetători. S-a argumentat faptul că cel mai bun este spațiul top-view U-disparity, folosit și în teza de față. Spațiul cartezian ar trebui folosit pentru procesări secundare sau pentru criterii suplimentare în cadrul procesărilor principale.

În capitolul 6 se prezintă o abordare originală pentru detecția de obstacole din cadre stereo individuale. Abordarea constă dintr-o suită de pași și subpași de procesare. Punctele 3D sunt reprezentate într-un spațiu comprimat orizontal (de tip grid top-view), ținând seama de posibilitățile reale de percepție a scenei prin intermediul stereo viziunii (spațiul U-disparity). În acest spațiu, celulele sunt grupate folosind criterii de vecinătate și de densitate. Au fost luate măsurile necesare compensării diferiților factori care influențează vecinătatea și densitatea. Urmează o îmbunătățire a frontierei fiecărui obstacol, prin procesări la nivel de coloane individuale și la nivel de obstacol. Pentru detecția fidelă a vehiculelor mai înalte decât autovehiculul propriu, de exemplu a camioanelor, obstacolele sunt extinse drept în sus, în măsura în care sunt puncte 3D, dar nu mai sus de 4,5 m. Alte două analize, pe direcția verticală, stabilesc limitele verticale corecte.

Adesea, mai multe obstacole apropiate sunt grupate împreună, dar forma frontierei grupului poate oferi indicii suplimentare. Astfel, dacă frontiera vizibilă (înspre cameră) are concavități, acestea sunt folosite pentru determinarea punctelor de fragmentare în obstacolele constituente. Apoi, partea cvazi-liniară cea mai lungă a frontierei vizibile poate indica orientarea fiecărui obstacol. După acest pas fiecare obstacol este modelat printr-un cuboid, posibil orientat. În pasul următor, se evaluează fidelitatea cuboidului, prin suprafața spațiului liber înglobat între laturile vizibile ale cuboidului și frontiera vizibilă. Dacă această suprafață e semnificativă, atunci se aplică un algoritm de fragmentare. Desigur, s-a ținut seama de cazul în care suprafața liberă este cauzată de ocluziile din partea unor obstacole de prim-plan sau de cazul în care o parte a cuboidului nu este vizibilă în ambele camere ale sistemului stereo.

Majoritatea pașilor și sub-pașilor sunt de concepție și/sau soluționare originală. De multe ori, prin aprofundarea unui pas de procesare, s-au deschis direcții noi. De exemplu, rafinarea limitelor obstacolelor a condus la studiul frontierei și a înfășurătoarei convexe, iar mai apoi s-au dezvoltat algoritmi pentru fragmentarea grupurilor de obstacole și determinarea orientării obstacolelor individuale. În final, s-a ajuns la un grad înalt de modelare fidelă a obstacolelor reale prin cuboide.

Obiectivele propuse au fost atinse și chiar depășite. În partea de înțelegere a contextului s-a studiat domeniul vehiculelor autonome (cu accent pe folosirea senzorilor de stereo viziune), s-a studiat reconstrucția stereo și posibilitățile oferite de aceasta, s-au testat diferite modele pentru reprezentarea obstacolelor și s-au studiat principalele abordări existente în domeniul detecției de obstacole folosind senzori video. Soluția originală propusă pentru detecția obstacolelor din cadre stereo individuale, prin multele problematizări și soluționări, este cea mai elaborată soluție la nivel internațional.

Contribuțiile tezei au fost dezvoltate și folosite în cadrul a 10 contracte de cercetare finanțate de producătorul auto Volkswagen AG în perioada 2001-2009 (anexa D). Soluția propusă a continuat să fie principala metodă de detecție a obstacolelor (în special a celor statice – pentru care nu există informație de mișcare) în grupul de procesarea imaginilor și recunoașterea formelor de la Universitatea Tehnică din Cluj-Napoca.

Pe lângă teza propriu-zisă, au mai rezultat 24 articole, două capitole de carte și 28 rapoarte tehnice pentru client. Totodată, s-a efectuat activitate de cercetare științifică la Centrul de Cercetare din Wolfsburg a Volkswagen AG, însumând o perioadă de timp de 6 luni. Recunoașterea rezultatelor este apreciată prin 480 de citari (din care 250 în ultimii 5 ani), majoritatea fiind independente.

## 7.1. Contribuțiile tezei

- S-a argumentat faptul că detecția de obstacole pe bază de stereo viziune este potrivită în scene complexe de trafic.
- S-a propus o schemă generică, organizată pe mai multe nivele, pentru realizarea de sisteme ADAS, bazate pe stereo viziune.
- Asupra stereo reconstrucției s-a făcut o analiză originală legată de aspecte precum: acuratețea nativă și modelarea acesteia, erori și clase de magnitudini ale erorilor, cantitatea și distribuția punctelor 3D.
- S-a pus accentul pe rezoluția cu care senzorul de stereo viziune percepe lumea reală.
- S-au studiat modelele folosite pentru reprezentarea obstacolelor, multe dintre ele fiind implementate și experimentate în cadrul tezei.
- În studiul abordărilor din literatura de specialitate, este originală clasificarea abordărilor stereo în funcție de spațiul de procesare folosit preponderent. Se face comparație între spațiile de procesare. Este argumentat faptul că cel mai bun este spațiul top-view U-disparity, folosit și în teza de față.
- S-a dezvoltat o abordare proprie pentru detecția de obstacole din cadre stereo individuale. Abordarea constă dintr-o suită de pași și subpași, mulți dintre aceștia fiind o premieră în domeniu și sunt împărțiți pe mai multe direcții:
  - Evidențierea zonelor ocupate:
    - S-a construit un spațiu comprimat de tip grid top-view, ținând seama de mai multe aspecte legate de posibilitățile de percepție a scenei prin stereo viziune.
    - S-a dezvoltat un algoritm de etichetare specializat, ținând seama de natura densității și vecinătății celulelor obstacolelor.
    - S-a rafinat frontiera obstacolelor, atât la nivel de coloane individuale cât și la nivel de obstacol.
  - Procesări pe direcția verticală:
    - S-a găsit o soluție pentru detecția corectă atât a obstacolelor joase cât și a celor înalte.
    - Fragmentări și rafinări.
  - Valorificarea formelor obstacolelor:
    - În cazul în care un obstacol nu are formă cuboidală, bazat pe analiza formei frontierei obstacolului, s-au dezvoltat doi algoritmi de fragmentare în sub-obstacole cuboidale, unul pe criterii de concavitate și altul pe criterii de convexitate a frontierei.

- Orientarea obstacolelor s-a determinat tot pe baza formei frontierelor acestora.
- Cazurile în care un obstacol este parțial ocluzat de alte obstacole de prim-plan sau iese parțial din câmpul vizual stereo sunt semnalizate și se ține cont de modificarea formei frontierei din aceste cauze.
- S-a ajuns la un grad înalt de modelare fidelă a obstacolelor reale prin cuboide.
- O problemă importantă a detecției de obstacole este că nu se cunoaște nici forma obstacolelor și nici erorile de reconstrucție. Lucrarea de față folosește diferite indicii și tehnici pentru a diminua această problemă.
- Proiectarea și implementarea de metode de detecție de obstacole din puncte 3D a avut un caracter de pionierat, anterior folosindu-se cu precădere spațiul disparităților.
- Metoda originală propusă pentru detecția obstacolelor din cadre stereo individuale, prin multele problematizări și soluționări, este cea mai elaborată metodă la nivel internațional.

Un exemplu tipic de scenă urbană, împreună cu rezultatul detecției de obstacole, este prezentat în figura de mai jos:



Fig. 7.1.1. Rezultate ale detecției de obstacole

# ANEXE

## A. Referințe bibliografice

[AMAA09] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Bota, M.-M. Meinecke, M. A. Obojski, "Stereovision-Based Sensor for Intersection Assistance", chapter in "[Advanced Microsystems for Automotive Applications 2009: Smart Systems for Safety, Sustainability and Comfort](#)".

[Ares] M. Ares, S. Royo, "Low-cost 3D scanning technology based on active stereovision".

[Aufreere01] R. Aufreere, R. Chapuis, F. Chausse, "A model-driven approach for real-time road recognition", *Machine Vision and Applications*, Springer-Verlag, 2001.

[Badino07] H. Badino, U. Franke, R. Mester, "Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming", *International Conference on Computer Vision*, 2007.

[Badino09] H. Badino, U. Franke, D. Pfeiffer, „The Stixel World - A Compact Medium Level Representation of the 3D-World", *DAGM Symposium on Pattern Recognition*, 2009.

[Bertozzi97] M. Bertozzi and A. Broggi, "Vision-Based Vehicle Guidance," in *The Computer Journal*, pp. 49-55, July 1997.

[Bertozzi97A] M. Bertozzi, G. Conte, A. Fascioli and A. Broggi, "Obstacle and lane detection on ARGO," *Proc. Intelligent Transportation Systems Conference*, 1997.

[Bertozzi98] M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. on Image Processing*, 1998.

[Bertozzi98A] M. Bertozzi, A. Broggi and A. Fascioli, "An extension to the Inverse Perspective Mapping to handle non-flat roads," in *Proceedings IEEE Intelligent Vehicles Symposium*, pages 305-310, Stuttgart, Germany, 1998.

[Bertozzi01] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miché, S. Mousset, and G. Toulminet, "A Cooperative Approach to Vision-based Vehicle Detection", in *Proc. Intelligent Transportation Systems Conference (ITSC'01)*, August 2001.

[Bertozzi05] M. Bertozzi, "Obstacle Detection for Start-Inhibit and Low Speed Driving," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2005.

[Bombini06] L. Bombini, P. Cerri, P. Medici and G. Alessandretti, "Radar-vision fusion for vehicle detection," in *Proc. International Workshop on Intelligent Transportation (WIT'06)*, 2006.

[Bouguet] J.Y. Bouguet, Camera Calibration Toolbox for Matlab:  
[www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/).

[**Broggi94**] A. Broggi, P. Cerri and P.C. Antonello, "Multi-Resolution Vehicle Detection using Artificial Vision," in *Proc. Intelligent Vehicles (IV'04)*, June 2004, pp. 310-314. Available: <http://www.ce.unipr.it/people/bertozzi/pap/cr/iv2004-vd-crf.pdf>

[**Broggi95**] A. Broggi, "Robust Real-time Lane and Road Detection in Critical Shadow Conditions," in *Proc. IEEE International Symposium on Computer Vision*, November 1995, pp. 353-358. Available: <http://www.ce.unipr.it/people/broggi/publications/coralgables.pdf>

[**Broggi05**] A. Broggi, C. Caraffi, R. I. Fedriga and Paolo Grisleri, "Obstacle detection with stereo vision for off-road vehicle navigation," In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[**Broggi13**] A. Broggi, S. Cattani, M. Patander, M. Sabbatelli and P. Zani, "A full-3D voxel-based dynamic obstacle detection for urban scenario using stereo vision", In *Proc. IEEE Intl. Conf. on Intelligent Transportation Systems*, 2013.

[**Caraffi07**] C. Caraffi, S. Cattani and P. Grisleri, "Off-Road path and obstacle detection using decision networks and stereo vision", *IEEE Intl. Conf. on Intelligent Transportation Systems 2007*.

[**Catadioptric01**] sistem catadioptric:  
[www1.cs.columbia.edu/CAVE/projects/cad\\_stereo/cad\\_stereo.php](http://www1.cs.columbia.edu/CAVE/projects/cad_stereo/cad_stereo.php)

[**Catadioptric02**] sistem catadioptric: [www.lhup.edu/~dsimanek/3d/stereo/3dgallery7.htm](http://www.lhup.edu/~dsimanek/3d/stereo/3dgallery7.htm)

[**Daimler**] Grupul de Image Understanding de la Daimler are o buna parte dintre realizari prezentate pe: [www.6d-vision.com](http://www.6d-vision.com)

[**Danescu07**] R. Danescu, S. Nedevschi, M.M. Meinecke, T. Graf, „Stereovision Based Vehicle Tracking in Urban Traffic Environments”, *Intelligent Transportation Systems Conference*, 2007.

[**Danescu08**] R. Danescu and S. Nedevschi, "Adaptive and robust road tracking system based on stereovision and particle filtering", *International Conference on Intelligent Computer Communication and Processing*, 2008

[**Danescu14**] R. Danescu and R. Itu, "An Efficient Obstacle Awareness Application for Android Mobile Devices", in *IEEE Conf. on Intelligent Computer Communication and Processing*, 2014.

[**Erbs12**] F. Erbs and U. Franke, "Stixmentation - probabilistic stixel based traffic scene labeling," In *British Machine Vision Conference*, 2012.

[**Erbs13**] F. Erbs, B. Schwarz, U. Franke: "From stixels to objects – a conditional random field based approach", *IEEE Intelligent Vehicles Symposium IV 2013*.

[**Erbs13A**] F. Erbs, A. Witte, T. Scharwaechter and R. Mester, "Spider-based stixel object segmentation," *IEEE Intelligent Vehicles Symposium 2013*.

- [**Fang03**] C. Y. Fang, C. S. Fuh, S. W. Chen, and P. S. Yen, "A Road Sign Recognition System Based on Dynamic Visual Model," in *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, Wisconsin, Vol. 1, pp. 750-755, 2003.
- [**FHWA96**] U.S. Department of Transportation, Federal Highway Administration, "An Hypothesized Evolution of an Automated Highway System," Publication No. FHWA-RD-96-051, January 1996 (<http://faculty.washington.edu/~jbs/itrans/ahsevolv.htm>)
- [**Franke99**] U. Franke, D. Gavrilu, S. Görzig, F. Lindner, F. Paetzold and C. Wöhler, "Autonomous Driving approaches Downtown," in *Intelligent Systems 1999*.
- [**Franke05**] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6D-vision: fusion of stereo and motion for robust environment perception," In *Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM05)*, Vienna, Austria, Sep. 2005.
- [**Gavrila98**] D. M. Gavrilu and V. Philomin, "Real-time Object Detection using Distance Transforms," *Proc. of IEEE Intelligent Vehicles Symposium*, 1998.
- [**Gehrig04**] S. Gehrig, J. Klappstein and U.Franke, „Active stereo for intersection assistance," in *Vision, Modeling, and Visualization 2004*.
- [**Gehrig09**] S.Gehrig, F.Eberli, T.Meyer, "A Real-time Low-Power Stereo Vision Engine Using Semi-Global Matching", *International Conference on Computer Vision Systems*, 2009.
- [**Geiger11**] A. Geiger, J. Ziegler and C. Stiller, "StereoScan: Dense 3D Reconstruction in Real-time", *Intelligent Vehicles Symposium*, 2011.
- [**Haller**] I. Haller, C. Pantilie, F. Oniga, and S. Nedeveschi, "Real-time semi-global dense stereo solution with improved sub-pixel accuracy", in *2010 IEEE Intelligent Vehicles Symposium*.
- [**Hancock97**] J. A. Hancock, "High-Speed Obstacle Detection for Automated Highway Applications," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1997.
- [**Hoffmann04**] C. Hoffmann, T. Dang and C. Stiller, "Vehicle detection fusing 2D visual features," in *Proc. Intelligent Vehicles (IV'04)*, June 2004, pp. 280-285.
- [**Hu05**] Z. Hu, F. Lamosa, and K. Uchimura, "A complete U-V-disparity study for stereovision based 3D driving environment analysis," in *3-D Digital Imaging and Modeling (3DIM'05)*, 2005.
- [**IEEE98**] "Automated Highway Systems, Incremental Deployment as a Solution for The Future of Transportation" (<http://www.wise-intern.org/journal98/POSTEMA.PDF>)
- [**IPPRG**] Image Processing and Pattern Recognition Group (IPPRG), Technical University of Cluj-Napoca: [cv.utcluj.ro](http://cv.utcluj.ro)
- [**Krips04**] M. Krips, J. Velten, A. Kummert and A. Teuner, "AdTM tracking for blind spot collision avoidance," in *Proc. Intelligent Vehicles (IV'04)*, June 2004, pp. 544-548.

[**Kubota07**] S. Kubota, T. Nakano, and Y. Okamoto, “A global optimization algorithm for real-time on-board stereo obstacle detection systems,” in *IEEE Intelligent Vehicles Symposium, 2007*, pp. 7–12.

[**Labayrade02**] R. Labayrade, D. Aubert, J.-P. Tarel, “Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through "V-disparity" Representation", *IEEE Intelligent Vehicles Symposium, 2002*.

[**Labayrade03**] R. Labayrade and D. Aubert, “A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision,” *IEEE Intelligent Vehicles Symposium, 2002*.

[**Manduchi04**] R. Manduchi, A. Castano, A. Talukder, L. Matthies, “Obstacle detection and terrain classification for autonomous off-road navigation,” *Autonomous Robots, 2004*.

[**Mark07**] W. van der Mark, J.C van der Heuvel and F. Groen, “Stereo based obstacle detection with uncertainty in rough terrain,” In *Proc. of the IEEE Intelligent Vehicles Symposium, 2007*.

[**Nedevschi04**] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, T. Graf, “[High Accuracy Stereo Vision System for Far Distance Obstacle Detection](#),” in *Proc. IEEE Intelligent Vehicles Symposium (IV'04)*, Parma, Italy, 14-17 June 2004, pp. 292-297.

[**Nedevschi04A**] S. Nedevschi, R. Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, “3D lane detection system based on stereovision,” in *IEEE Intelligent Transportation Systems Conference*, Oct. 2004, pp. 161-166.

[**Nedevschi05**] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, S. Sobol, T. Graf, R. Schmidt, “[Driving Environment Perception Using Stereovision](#),” in *Proc. IEEE Intelligent Vehicles Symposium 2005*.

[**NHTSA10**] “2010 Motor Vehicle Crashes: Overview”  
(<http://www-nrd.nhtsa.dot.gov/Pubs/811552.pdf>)

[**Oniga04**] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, Thorsten Graf, Rolf Schmidt, “High Accuracy Stereovision Approach for Obstacle Detection on Non-Planar Roads”, *Proceedings of IEEE Intelligent Engineering Systems 2004*.

[**Oniga10**] F. Oniga and S. Nedevschi, “Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection”, *IEEE Transactions on Vehicular Technology*, 2010.

[**OpenCVcalibmono**] calibrarea camerelor mono:  
[http://docs.opencv.org/doc/tutorials/calib3d/camera\\_calibration/camera\\_calibration.html](http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html)

[**OpenCVcalibstereo**] calibrarea camerelor stereo:  
[docs.opencv.org/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html)



[**OpenCV**] biblioteca open source oferită de Intel, având sute de funcții din domeniul computer vision, inclusiv stereo viziune: [opencv.org](http://opencv.org)

[**Pantilie11**] C. D. Pantilie and S. Nedeveschi, "Real-time semi-global matching using segmentation and plane fitting for improved accuracy on the GPU", *IEEE Conference on Intelligent Transportation Systems, 2011*.

[**PatentDaimler**] Hernan Badino, Uwe Franke. Method for detecting objects. U.S. Patent 8548229 B2, filed Feb 4, 2010.

[**Pfeiffer10**] D. Pfeiffer and U. Franke, "Efficient representation of traffic scenes by means of dynamic stixels," In Intelligent Vehicles Symposium 2010.

[**Pfeiffer10**] D. Pfeiffer and U. Franke, "Towards a global optimal multi-layer stixel representation of dense 3D data," In British Machine Vision Conference, 2011.

[**Pfeiffer10**] D. Pfeiffer, F. Erbs and U. Franke, "Pixels, stixels, and objects", Workshop on Computer Vision in Vehicle Technology, European Computer Vision Conference 2012.

[**PointGray12**] "Stereo Accuracy and Error Modeling":  
[http://www.ptgrey.com/support/downloads/documents/TAN2004006\\_Stereo\\_Accuracy\\_Error\\_Modeling.pdf](http://www.ptgrey.com/support/downloads/documents/TAN2004006_Stereo_Accuracy_Error_Modeling.pdf)

[**Pocol03**] C. Pocol, „Object Detection and Localization from 3D Points”, *lucrare de dizertatie*.

[**Pocol07**] C. Pocol, S. Nedeveschi, „Obstacle Detection for Mobile Robots, Using Dense Stereo Reconstruction”, *IEEE International Conference on Intelligent Computer Communication and Processing, 2007*.

[**Pocol08**] C. Pocol, S. Nedeveschi, Marc-M. Meinecke, "[Obstacle Detection Based on Dense Stereovision for Urban ACC Systems](#)," in *Proc. Workshop on Intelligent Transportation (WIT'08)*, Hamburg, Germany, 18-19 March 2008.

[**Pocol15**] lucrarea de față.

[**Ran99**] B. Ran, H. X. Liu R., "Development of A Vision-Based Vehicle Detection and Recognition System For Intelligent Vehicles," TRB Annual Meeting, 1999.

[**Regensburger94**] U. Regensburger and V. Graefe, "Visual Recognition of Obstacles on Roads," in *Proc. Intelligent Robots and Systems '94 (IROS'94)*, September 1994, volume 2, pp. 980-987. Available: <http://robotik.w3.rz.unibw-muenchen.de/PDF/obsroa.pdf>

[**Rehrmann97**] V. Rehrmann and M. Rothhaar, "Detection and tracking of moving objects in color outdoor scenes," in *Proc. Intelligent Dedicated Conference on Robotics, Motion and Machine Vision in the Automotive Industry*, 1997, pp. 16-19. Available: <http://www.uni-koblenz.de/~lb/publications/Rehrmann1997DAT.pdf>

[**Smith96**] S.M. Smith, "Integrated Real-Time Motion Segmentation and 3D Interpretation," in *Proc. 13th Int. Conf. on Pattern Recognition (ICPR'06)*, volume 3, pp. 49-55.

[**Soto00**] A. Soto et al., “Dynamic and Distributed Reconnaissance and Surveillance Using All Terrain UGVs,” in *Proc. Int. Conf. on Field and Service Robotics*, August 1999, pp. 329-334. Available:

[http://www.ri.cmu.edu/pub\\_files/pub2/soto\\_alvaro\\_1999\\_1/soto\\_alvaro\\_1999\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub2/soto_alvaro_1999_1/soto_alvaro_1999_1.pdf)

[**Stereo01**] lista de sisteme stereo: [www.stereoscopy.com](http://www.stereoscopy.com)

[**Stereo02**] lista de sisteme stereo: [www.lhup.edu/~dsimanek/3d/3dpage.htm](http://www.lhup.edu/~dsimanek/3d/3dpage.htm)

[**StereoVision**] tutorial stereo viziune: [inperc.com/wiki/index.php?title=Stereo\\_vision](http://inperc.com/wiki/index.php?title=Stereo_vision)

[**Techmer04**] A. Techmer, “Real-time Motion Analysis for Monitoring the Rear and Lateral Road,” in *Proc. Intelligent Vehicles (IV’04)*, June 2004, pp. 704-709.

[**Thrun01**] S. Thrun, W. Burgard, D. Chakrabarti, R. Emery, Y. Liu and C. Martin, „A Real-time Algorithm for Acquiring Multi-Planar Volumetric Models with Mobile Robots,” in *Robotics Research, The Tenth International Symposium, ISRR*, November 2001, pp. 21-37.

[**TYZX**] Sight in an Unstructured World – 3D Stereo Vision and Systems that See:

[http://www.tyzx.com/PDFs/tyzx\\_wp.pdf](http://www.tyzx.com/PDFs/tyzx_wp.pdf)

[**TYZXhw**] producătorul unei plăci de reconstrucție hardware: [www.tyzx.com](http://www.tyzx.com)

[**Ulrich00**] I. Ulrich and I. Nourbakhsh, “Appearance-Based Obstacle Detection with Monocular Color Vision,” in *Proc. AAAI National Conference on Artificial Intelligence*, August 2000, pp. 866-871 Available: <http://www.cs.cmu.edu/~illah/PAPERS/abod.pdf>

[**Vatavu11**] A. Vatavu, S. Nedevschi, and F. Oniga, “Real time environment representation in driving scenarios based on object delimiters extraction,” in *Lecture Notes in Electrical Engineering*, 2011.

[**Vatavu12**] A. Vatavu, R. Danescu, S. Nedevschi, „Real-Time Dynamic Environment Perception in Driving Scenarios Using Difference Fronts”, *Intelligent Vehicles Symposium, 2012*

[**VisLab**] The Artificial Vision and Intelligent Systems Laboratory (VisLab) of Parma University: [vislab.it](http://vislab.it)

[**Wedel08**] A. Wedel, U. Franke, H. Badino and D. Cremers, “B-spline modeling of road surfaces for freespace estimation,” In *Intelligent Vehicles Symposium 2008*.

[**WikiACC**] [http://en.wikipedia.org/wiki/Autonomous\\_cruise\\_control\\_system](http://en.wikipedia.org/wiki/Autonomous_cruise_control_system)

[**WikiCACC**] [http://en.wikipedia.org/wiki/Cooperative\\_Adaptive\\_Cruise\\_Control](http://en.wikipedia.org/wiki/Cooperative_Adaptive_Cruise_Control)

[**WikiESP**] [http://en.wikipedia.org/wiki/Electronic\\_stability\\_control](http://en.wikipedia.org/wiki/Electronic_stability_control)

[**WikiStereoVision**] [en.wikipedia.org/wiki/Computer\\_stereo\\_vision](http://en.wikipedia.org/wiki/Computer_stereo_vision)

[**WikiStereopsis**] [en.wikipedia.org/wiki/Stereopsis](http://en.wikipedia.org/wiki/Stereopsis)

[**Williamson97**] T. A. Williamson, "A High-Performance Vision System for Obstacle Detection," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania, September 1998. Available: <http://www.cs.cmu.edu/~toddw/Thesis.pdf>

[**Williamson99**] T. Williamson and C. Thorpe, "A Trinocular Stereo System for Highway Obstacle Detection", *International Conference on Robotics and Automation*, 1999.

## B. Publicații

### Capitole de carte:

1. S. Nedevschi, R. Danescu, T. Marita, F. Oniga, **C. Pocol**, S. Bota, M.-M. Meinecke, M. A. Obojski, "Stereovision-Based Sensor for Intersection Assistance", chapter in "[Advanced Microsystems for Automotive Applications 2009: Smart Systems for Safety, Sustainability and Comfort](#)", editors G. Meyer, J. Valldorf, W. Gessner, published by Springer, pp.129-164, ISBN 978-3-642-00744-6
2. S. Nedevschi, R. Danescu, T. Marita, F. Oniga, **C. Pocol**, S. Bota and C. Vancea, "A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision", chapter in "Stereo Vision" editor A. Bhatti, published by InTech Education and Publishing, Vienna, 2008, pp. 235-272, ISBN 978-953-7619-22-0

### În reviste naționale cotate CNCSIS, categorie B:

1. **C. Pocol**, S. Nedevschi, I. Giosan, "Obstacle Detection Based on Single Frame Stereo Vision," *J. Automation, Computers and Applied Mathematics (ACAM)*, 2013, pp. 246-254, Ed. Mediamira, Cluj-Napoca, Romania, ISSN 1221-437X.
2. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, Thorsten Graf, Rolf Schmidt, "High Accuracy Stereovision Approach for Obstacle Detection on Non-Planar Roads," *J. Automation, Computers and Applied Mathematics (ACAM)*, vol.14, no. 2, pp. 173-178, 2005, Ed. Mediamira, Cluj-Napoca, Romania, ISSN 1221-437X
3. S. Nedevschi, T. Marita, M. Vaida, R. Danescu, D. Frentiu, F. Oniga, **C. Pocol**, "Camera Calibration Method for Stereo Measurements – Extended Version," *J. Control Engineering and Applied Informatics (CEAI)*, vol. 4, no. 2, pp. 21-28, June 2002, Bucharest, Romania, ISSN 1454-8658

### Conferințe Internaționale ISI / IEEE Explore (majoritatea):

4. I. Giosan, S. Nedevschi, **C. Pocol**, "Shape improvement of traffic pedestrian hypotheses by means of stereo-vision and superpixels," in *Proc. IEEE International Conference on Intelligent Computer Communication and Processing (ICCP'15)*, Cluj-Napoca, România, 3-5 September 2015, pp. 217-222, ISBN 978-1-4673-8200-7
5. **C. Pocol**, S. Nedevschi, Marc-M. Meinecke, "[Obstacle Detection Based on Dense Stereovision for Urban ACC Systems](#)," in *Proc. Workshop on Intelligent Transportation (WIT'08)*, Hamburg, Germany, 18-19 March 2008
6. S. Nedevschi, R. Danescu, **C. Pocol**, Marc-M. Meinecke, "[Stereo Image Processing for ADAS and Precrash Systems](#)," in *Proc. Workshop on Intelligent Transportation (WIT'08)*, Hamburg, Germany, 18-19 March 2008
7. **C. Pocol**, S. Nedevschi, "Obstacle Detection for Mobile Robots, Using Dense Stereo Reconstruction," in *Proc. IEEE International Conference on Intelligent Computer*

*Communication and Processing (ICCP'07)*, Cluj-Napoca, România, 6-8 September 2007, pp. 127-132, ISBN 1-4244-1491-1, DOI 10.1109/ICCP.2007.4352151

8. S. Nedevschi, R. Danescu, T. Marita, F. Oniga, **C. Pocol**, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, M. A. Obojski, "[A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision](#)," in *Proc. Intelligent Vehicles (IV'07)*, Istanbul, Turkey, 13-15 June 2007, pp. 278-286, ISBN 1-4244-1068-1

9. S. Nedevschi, T. Marita, R. Danescu, F. Oniga, **C. Pocol**, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, "Stereovision Sensor for Driving Assistance," in *Proc. IEEE International Conference on Intelligent Computer Communication and Processing (ICCP'06)*, Volume 1, Cluj-Napoca, Romania, 1-2 September 2006, pp. 105-111, ISBN 978-973-662-234-2

10. S. Nedevschi, S. Bota, T. Marita, F. Oniga, **C. Pocol**, "[Real-Time 3D Environment Reconstruction Using High Precision Trinocular Stereovision](#)," in *Proc. IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR'06)*, Volume 2, Cluj-Napoca, Romania, 25-28 May 2006, pp. 333 – 338, ISBN 1-4244-0361-8

11. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, S. Sobol, T. Graf, R. Schmidt, "[Driving Environment Perception Using Stereovision](#)," in *Proc. IEEE Intelligent Vehicles Symposium (IV'05)*, Las Vegas, USA, June 2005, pp. 331-336, ISBN 0-7803-8961-1/05

12. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, "Moving Cameras Rotation Estimation Using Horizon Line Features' Motion Field," in *Proc. International Carpathian Control Conference (ICCC'05)*, Volume 2, Lilafured Miskolc, Hungary, 24-26 May 2005, pp. 449-454, ISBN 963-661-645-0

13. S. Nedevschi, T. Marita, R. Danescu, D. Frentiu, F. Oniga, **C. Pocol**, "Camera Calibration method for high-accuracy stereovision," in *Proc. Joint-Hungarian-Austrian Conference on Image Processing and Pattern Recognition*, Veszprem, Hungary, 11-13 May 2005, pp. 303-310, ISBN 3-85403-192-0

14. S. Nedevschi, R. Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, "[3D Lane Detection System Based on Stereovision](#)," in *Proc. IEEE Intelligent Transportation Systems Conference (ITSC'04)*, Washington, USA, 3-6 October 2004, pp. 161-166, 0-7803-8501-2

15. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, Thorsten Graf, Rolf Schmidt, "[High Accuracy Stereovision Approach for Obstacle Detection on Non-Planar Roads](#)," in *Proc. IEEE Intelligent Engineering Systems (INES'04)*, Cluj Napoca, Romania, Sept. 2004, pp. 211-216, ISBN 973-662-120-0

16. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, R. Schmidt, T. Graf, "Stereovision Approach For Obstacle Detection On Non-Planar Roads," in *Proc. International Conference on Informatics in Control, Automation and Robotics (ICINCO'04)*, Setubal, Portugal, 25-28 August 2004, pp. 11-18, ISBN 972-8865-12-0

17. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, R. Schmidt, T. Graf, "[High Accuracy Stereo Vision System for Far Distance Obstacle Detection](#)," in *Proc. IEEE Intelligent Vehicles Symposium (IV'04)*, Parma, Italy, 14-17 June 2004, pp. 292-297, ISBN 0-7803-8311-7
18. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, "Extraction of Dynamic Traffic Description Using Multiple Stereovision Equipped Vehicles," in *Proc. Computing, Communications and Control Technologies (CCCT'04)*, Austin, Texas, USA, 15-17 August 2004, pp. 410-445, ISBN 980-6560-17-5
19. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, "[Spatial Grouping of 3D Points from Multiple Stereovision Sensors](#)," in *Proc. International Conference of Networking, Sensing and Control (ICNSC'04)*, Taipei, Taiwan, 21-23 March 2004, pp. 874-879, ISBN 0-7803-8194-7
20. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, "[Dynamic traffic description using stereovision equipped vehicles and ad-hoc wireless networking](#)," in *Proc. IEEE-TTTC International Conference on Automation, Quality Testing and Robotics (AQTR'04)*, Volume 2, Cluj Napoca, Romania, May 2004, pp. 153-158, ISBN 973-713-047-2
21. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, "[3D Environment Reconstruction Using Multiple Moving Stereovision Sensors](#)," in *Proc. microCAD International Scientific Conference*, Miskolc, Hungary, March 2004, pp. 93-98, ISBN 1-85338-855-6
22. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, "[Real-Time Extraction of 3D Dynamic Environment Description Using Multiple Stereovision Sensors](#)," in *Proc. International Conference on Computers, Communication and Control (CCCT'03)*, Orlando, Florida USA, 29 July – 1 August 2003, pp. 520-524, ISBN 980-6560-05-1
23. S. Nedevschi, T. Marita, R. Danescu, F. Oniga, D. Frentiu, **C. Pocol**, "Camera Calibration Error Analysis in Stereo Measurements," in *Proc. microCAD International Scientific Conference*, Miskolc, Hungary, 6-7 March 2003, pp. 51-57
24. S. Nedevschi, T. Marita, M. Vaida, R. Danescu, D. Frentiu, F. Oniga, **C. Pocol**, "Camera Calibration Method for Stereo Measurements," in *Proc. IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics (AQTR'02)*, Cluj-Napoca, Romania, May 2002, pp. 111-118, ISBN 973-9357-11-1

## C. Citări

În august 2015, numărul total de citări se ridică la 480, din care 260 în ultimii 5 ani. Cifrele sunt oferite de Google Scholar, incluzând toate citările, dintre care cel puțin 80% sunt independente.

Primele 5 cele mai citate lucrări sunt:

1. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, R. Schmidt, T. Graf, "[High Accuracy Stereo Vision System for Far Distance Obstacle Detection](#)," în *Proc. IEEE Intelligent Vehicles Symposium (IV'04)*, Parma, Italy, 14-17 June 2004, pp. 292-297, ISBN 0-7803-8311-7

cu un total de 124 citări

2. S. Nedevschi, R. Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, "[3D Lane Detection System Based on Stereovision](#)," în *Proc. IEEE Intelligent Transportation Systems Conference (ITSC'04)*, Washington, USA, 3-6 October 2004, pp. 161-166, 0-7803-8501-2

cu un total de 114 citări

3. S. Nedevschi, R. Danescu, T. Marita, F. Oniga, **C. Pocol**, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, M. A. Obojski, "[A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision](#)," în *Proc. Intelligent Vehicles (IV'07)*, Istanbul, Turkey, 13-15 June 2007, pp. 278-286, ISBN 1-4244-1068-1

cu un total de 68 citări

4. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, Thorsten Graf, Rolf Schmidt, "[High Accuracy Stereovision Approach for Obstacle Detection on Non-Planar Roads](#)," în *Proc. IEEE Intelligent Engineering Systems (INES'04)*, Cluj Napoca, Romania, Sept. 2004, pp. 211-216, ISBN 973-662-120-0

cu un total de 58 citări

5. S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, **C. Pocol**, S. Sobol, T. Graf, R. Schmidt, "[Driving Environment Perception Using Stereovision](#)," în *Proc. IEEE Intelligent Vehicles Symposium (IV'05)*, Las Vegas, USA, June 2005, pp. 331-336, ISBN 0-7803-8961-1/05

cu un total de 34 citări

## D. Rapoarte tehnice

Cea mai mare parte a acestei teze a fost finanțată de clientul Volkswagen AG, în cadrul a 10 contracte de cercetare acoperind perioada 2001 – 2009. Fiecare contract a fost împărțit în câte patru părți finalizate prin rapoarte tehnice conținând descrierea amănunțită a problemelor, a soluțiilor posibile, a soluțiilor implementate precum și analiza performanțelor obținute.

### 2008:

1. S. Nedevschi, T. Marita, C. Pocol, S. Bota, R. Borca, I. Giosan, C. Pantilie “Dense Stereo-Based Object Tracking and Classification for Pre-Crash Applications (StereoClass): Technical Report WP2,” Technical University Cluj-Napoca, December 2008.
2. S. Nedevschi, T. Marita, C. Pocol, R. Danescu, F. Oniga, C. Vancea, S. Bota, R. Borca, M. Negru, I. Giosan, A. Vatavu, C. Pantilie “Stereo System Modular Design (MODULAR): Technical Report WP1,” Technical University Cluj-Napoca, December 2008.
3. S. Nedevschi, T. Marita, C. Pocol, R. Danescu, S. Bota, R. Borca, I. Giosan, C. Pantilie “Dense Stereo-Based Object Tracking and Classification for Pre-Crash Applications (StereoClass): Technical Report WP1,” Technical University Cluj-Napoca, September 2008.

### 2007:

4. S. Nedevschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, “Dense Stereo-Based Object Detection, Tracking And Pedestrian Classification In Traffic Environments (StereoPed): Technical Report WP2,” Technical University Cluj-Napoca, December 2007.
5. S. Nedevschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, “Stereo Sensor for City Automatic Cruise Control (StereoSens): Technical Report WP2&3,” Technical University Cluj-Napoca, December 2007.
6. S. Nedevschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, “Dense Stereo-Based Object Detection, Tracking And Pedestrian Classification In Traffic Environments (StereoPed): Technical Report WP1,” Technical University Cluj-Napoca, October 2007.
7. S. Nedevschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, “Dense Stereo-Based Object Detection, Tracking And Pedestrian Classification In Traffic Environments (DESPED): Technical Report WP4,” Technical University Cluj-Napoca, July 2007.



8. S. Nedeveschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, "Stereo Sensor for City Automatic Cruise Control (StereoSens): Technical Report WP1," Technical University Cluj-Napoca, July 2007.
9. S. Nedeveschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, "Dense Stereo-Based Object Detection, Tracking And Pedestrian Classification In Traffic Environments (DESPED): Technical Report WP2," Technical University Cluj-Napoca, February 2007.
10. S. Nedeveschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, "Dense Stereo-Based Object Recognition System for Automatic Cruise Control in Urban Traffic Environments (DESBOR): Technical Report WP5," Technical University Cluj-Napoca, February 2007.

#### **2006:**

11. S. Nedeveschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, "Dense Stereo-Based Object Detection, Tracking And Pedestrian Classification In Traffic Environments (DESPED): Technical Report WP1," Technical University Cluj-Napoca, November 2006.
12. S. Nedeveschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, R. Borca, M. Negru, S. Lazar, A. Vatavu, "Dense Stereo-Based Object Recognition System for Automatic Cruise Control in Urban Traffic Environments (DESBOR): Technical Report WP4," Technical University Cluj-Napoca, October 2006.
13. S. Nedeveschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, "Dense Stereo-Based Object Recognition System for Automatic Cruise Control in Urban Traffic Environments (DESBOR): Technical Report WP3," Technical University Cluj-Napoca, June 2006.

#### **2005:**

14. S. Nedeveschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, "Dense Stereo-Based Object Recognition System for Automatic Cruise Control in Urban Traffic Environments (DESBOR): Technical Report WP2," Technical University Cluj-Napoca, December 2005.
15. S. Nedeveschi, T. Marita, R. Danescu, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, S. Bota, "Dense Stereo-Based Object Recognition System for Automatic Cruise Control in Urban Traffic Environments (DESBOR): Technical Report WP1," Technical University Cluj-Napoca, September 2005.

**2004:**

16. S. Nedeveschi, T. Marita, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, C. Tomiuc, R. Schnell, C. Vancea, "Stereo-Camera Based Object Recognition System for Vehicle Application – 3 extension (SCABOR3Ex): Technical Report," Technical University Cluj-Napoca, December 2004.
17. S. Nedeveschi, T. Marita, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, C. Tomiuc, R. Schnell, C. Vancea, "Stereo-Camera Based Object Recognition System for Vehicle Application - 3 (SCABOR3): Final Report, Technical Report WP4," Technical University Cluj-Napoca, September 2004.
18. S. Nedeveschi, T. Marita, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, C. Tomiuc, R. Schnell, C. Vancea, "Stereo-Camera Based Object Recognition System for Vehicle Application - 3 (SCABOR3): Algorithm implementation, integration and testing, Technical Report WP3," Technical University Cluj-Napoca, July 2004.
19. S. Nedeveschi, T. Marita, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, C. Tomiuc, R. Schnell, C. Vancea, "Stereo-Camera Based Object Recognition System for Vehicle Application - 3 (SCABOR3): Algorithm Design, Technical Report WP2," Technical University Cluj-Napoca, April 2004.

**2003:**

20. S. Nedeveschi, T. Marita, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, C. Tomiuc, R. Schnell, S. Sobol, "Stereo-Camera Based Object Recognition System for Vehicle Application - 3 (SCABOR3): Survey, Experiments, Specifications, Achievements, Technical Report WP1," Technical University Cluj-Napoca, December 2003
21. S. Nedeveschi, T. Marita, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, C. Tomiuc, R. Schnell, "Stereo-Camera Based Object Recognition System for Vehicle Application - 2 (SCABOR2): Final Report, Technical Report WP4," Technical University Cluj-Napoca, September 2003.
22. S. Nedeveschi, T. Marita, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, C. Tomiuc, R. Schnell, "Stereo-Camera Based Object Recognition System for Vehicle Application - 2 (SCABOR2): Algorithm Testing and Optimization, Technical Report WP3," Technical University Cluj-Napoca, July 2003.
23. S. Nedeveschi, T. Marita, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, C. Tomiuc, R. Schnell, "Stereo-Camera Based Object Recognition System for Vehicle Application - 2 (SCABOR2): Algorithm Design & Encoding, Technical Report WP2," Technical University Cluj-Napoca, May 2003.

**2002:**

24. S. Nedeveschi, T. Marita, D. Moga, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, C. Tomiuc, R. Schnell "Stereo-Camera Based Object Recognition System for Vehicle Application - 2 (SCABOR2): Survey, Experiments, Specifications, Achievements, Technical Report WP1," Technical University Cluj-Napoca, December 2002.

25. S. Nedevschi, M. Vaida, T. Marita, D. Moga, R. Danescu, D. Frentiu, C. Maris, F. Oniga, C. Pocol, C. Rotaru, "Stereo-Camera Based Object Recognition System for Vehicle Application (SCABOR): Final Report, Technical Report WP4," Technical University Cluj-Napoca, July 2002.
26. S. Nedevschi, M. Vaida, T. Marita, D. Moga, R. Danescu, D. Frentiu, C. Maris, F. Oniga, C. Pocol, C. Rotaru, "Stereo-Camera Based Object Recognition System for Vehicle Application (SCABOR): SCABOR Environment Sensorial System, Technical Report WP3," Technical University Cluj-Napoca, May 2002.
27. S. Nedevschi, M. Vaida, T. Marita, D. Moga, R. Danescu, D. Frentiu, C. Maris, F. Oniga, C. Pocol, C. Rotaru, "Stereo-Camera Based Object Recognition System for Vehicle Application: Algorithm Development (SCABOR), Technical Report WP2," Technical University Cluj-Napoca, March 2002.

\*\*\*

Totodată, s-a efectuat activitate de cercetare științifică la Centrul de Cercetare din Wolfsburg a Volkswagen AG, însumând 6 luni.

## E. Lista de figuri

Fig. 2.3.1. Sisteme stereo comerciale sau demonstrative, pentru aplicații automotivă .....	19
Fig. 3.1.1. Sisteme catadioptrice stereo, cu o singură cameră .....	26
Fig. 3.1.2. Constrângerea epipolară .....	26
Fig. 3.1.3. Triangularea .....	26
Fig. 3.1.4. Calibrarea intrinsecilor .....	29
Fig. 3.1.5. Calibrarea extrinsecilor (coordonatele 3D măsurate manual) .....	29
Fig. 3.1.6. Rectificarea imaginilor stânga și dreapta, inclusiv imaginea stânga color .....	30
Fig. 3.1.7. Reconstrucție 3D .....	33
Fig. 3.2.1. Acuratețea metrică .....	35
Fig. 3.2.2. Cauzele erorilor (notă: obstacolele roșii au fost clasificate ca obstacole de fundal) .....	37
Fig. 3.2.3. Magnitudinile claselor de erori, cu exemple în spațiul imaginii și spațiul top-view .....	38
Fig. 3.2.4. Exemple de erori ale disparităților și profunzimilor, pe diferite intervale de profunzime .....	38
Fig. 3.2.5. Exemple de magnitudinea erorilor distanțelor, pentru câteva erori ale disparității .....	38
Fig. 3.2.6. Modelul nativ ( $Z_e$ ), modelul liniar ( $\hat{Z}_e$ ) și diferența dintre ele ( $H$ ). .....	41
Fig. 3.2.7. O vedere mai îndeaproape a modelului liniar (dreapta roșie) din Fig. 3.2.5. ....	42
Fig. 3.2.8. Magnitudinile acurateței metrică pentru modelul nativ (a) și pentru modelul liniar (b) .....	43
Fig. 3.3.1. Distribuția punctelor 3D pe înălțime și profunzime (a) a punctelor de pe o suprafață plană – doar cele din dreptunghiul verde (b) (culorile punctelor nu au semnificație aici) .....	45
Fig. 3.3.2. Obstacole bogat reconstruite; frontiere bine conturate. ....	46
Fig. 4.2.1. Obstacolul real (a) și modelarea sa printr-un dreptunghi neorientat .....	50
Fig. 4.2.2. Includerea spațiului liber în interiorul dreptunghiului neorientat. ....	51
Fig. 4.2.3. Modelul dreptunghiular orientat. ....	51
Fig. 4.2.4. Un exemplu generic de model octogonal. ....	52
Fig. 4.2.5. Modelul octogonal, cu (a) și fără laturile nevizibile (b). ....	52
Fig. 4.2.6. Exemplu de obstacol care nu e potrivit pentru modelul octogonal (a). Un model fidel, dar nu octogonal (b) .....	53
Fig. 4.2.7. Exemplu de model poligonal top-view. ....	54
Fig. 4.2.8. Determinarea unui model dreptunghiular orientat pe baza modelului poligonal. ....	54
Fig. 4.2.9. Exemplu de model poligonal în spațiul imaginii. ....	54
Fig. 4.2.10. Modelul poligonal cu rezoluție constantă și înălțime variabilă. ....	55
Fig. 4.2.11. Model poligonal cu rezoluție variabilă și înălțime constantă. ....	55
Fig. 4.2.12. Modelul cuboid polar – top-view. ....	56
Fig. 4.2.13. Mesh de triunghiuri (a). Vedere dintr-un punct arbitrar (b) .....	56
Fig. 4.2.14. Creșterea rezoluției locale. Divizarea unui arc produce două sub-arce (verde și galben). ....	57
Fig. 4.2.15. Modelul multi-element .....	58
Fig. 4.2.16. Modelul “curbat” .....	58
Fig. 4.2.17. Obstacole modelate prin: a) cuboide, b) dreptunghiuri în imagine, c) dreptunghi top-view pentru obstacolele 3 și 6. ....	59
Fig. 4.2.18. a) – Cuboide neorientate, b) – cuboide orientate, c) top-view – atât cuboide orientate cât și cuboide neorientate. ....	61
Fig. 5.1.1. Clasificarea abordărilor pentru detecția de obstacole. ....	63
Fig. 5.2.1. Determinarea zonelor care prezintă simetrie verticală .....	65
Fig. 5.2.2. Cazuri când imaginea spatelui vehiculului este mai puțin simetrică. ....	66
Fig. 5.2.3. Rețeaua neuronală. ....	67
Fig. 5.3.1. Imagine color (a) percepută de o camera grayscale (b). ....	69
Fig. 5.3.2. a și c: Culoarea drumului se determină într-un trapez din partea de jos a imaginii; b și d: rezultatul detecției. ....	69
Fig. 5.4.1. Transformarea IPM: (a) imaginea perspectivă, .....	71
Fig. 5.4.2. Aspect 1:1 .....	71
Fig. 5.4.3. Corespondența intervalelor de drum, în imaginea perspectivă și în imaginea IPM. ....	71
Fig. 5.4.4. Temporal-IPM (a) imaginea perspectivă curentă (camera fish-eye), (b) imaginea binarizată a diferenței dintre: (c) imaginea IPM curentă și (d) imaginea IPM roto-translatată anterioară. ....	73
Fig. 5.4.5. (a) Imaginea curentă, inclusiv detecția, (b) imaginea de referință, de fundal, (c) imaginea diferență și (d) imaginea binarizată după filtrarea morfologică. ....	74
Fig. 5.4.6. Clasificarea mișcării obiectelor: (a) imaginea perspectivă, (b) imaginea IPM cu punctele de muchie clasificate (roșii – se apropie, albastre – se îndepărtează, verzi – staționare), (c) imaginea de muchii, (d) muchiile care se apropie, (e) muchiile care se îndepărtează, și (f) muchiile staționare. ....	75

Fig. 5.4.7. Câteva exemple de detecție și clasificare a obiectelor. În imaginile (c) și (d) sunt observabile greșeli.	75
Fig. 5.4.8. Imagine la momentul $t$ .	76
Fig. 5.4.9. Flux optic rar	76
Fig. 5.4.10. Imagine la momentul $t+1$	76
Fig. 5.4.11. Flux optic dens	76
Fig. 5.5.1. Imaginile inițiale stânga/dreapta, imaginile IPM stânga/dreapta și diferența imaginilor IPM	77
Fig. 5.5.2. Spații de procesare	78
Fig. 5.5.3. Spațiul V-disparity.	80
Fig. 5.5.4. Delimitarea obstacolelor în spațiul imaginii.	80
Fig. 5.5.5. Spațiul histogramei V-disparity.	81
Fig. 5.5.6. Spațiul histogramei U-disparity.	81
Fig. 5.5.7. Spațiul U-V.	81
Fig. 5.5.8. Prezența obstacolelor (culoarea codifică profunzimea).	82
Fig. 5.5.9. Fășile cu disparitatea preponderentă.	82
Fig. 5.5.10. Rezultate finale, după aplicarea filtrelor.	82
Fig. 5.5.11. Grupare în spațiul cartezian top-view.	83
Fig. 5.5.12. Obstacolele în spațiul imaginii, prin reconsiderarea coordonatei înălțimii (obstacolele de fundal nu sunt arătate).	83
Fig. 5.5.13. Sol, platforme joase și obstacole.	84
Fig. 5.5.14. Descriere poligonală.	84
Fig. 5.5.15. Detecția benzii de circulație.	85
Fig. 5.5.16. Punctele 3D din spațiul de interes.	85
Fig. 5.5.17. Histograma punctelor obstacolelor din Fig. 5.5.16.	85
Fig. 5.5.18. Etichetarea obstacolelor din Fig. 5.5.16.	85
Fig. 5.5.19. Fragmentarea în obstacole fără concavități.	86
Fig. 5.5.20. Obstacole orientate.	86
Fig. 5.5.21. Fragmentarea în cuboide fidele.	86
Fig. 5.5.22. Frontiera obstacolelor de prim-plan: grid în spațiul U-Z.	87
Fig. 5.5.23. Frontiera dintre spațiul liber și începutul obstacolelor de prim-plan: reprezentare în spațiul imaginii	87
Fig. 5.5.24. Spațiul de căutare.	88
Fig. 5.5.25. Transformarea conurilor în triunghiuri.	88
Fig. 5.5.26. Gruparea în obstacole individuale.	88
Fig. 5.5.27. Obstacolele delimitate de stixeli.	89
Fig. 5.5.28. Harta costurilor pentru segmentarea dintre prim-plan și fundal; costurile mai mici sunt desenate mai alb (scena diferă de cea din Fig. 5.5.27!).	89
Fig. 5.5.29. Vectorii stixelilor dinamici, corespunzători unei mișcări de 0,5 secunde.	90
Fig. 5.5.30. Mai mulți stixeli pentru o direcție optică.	90
Fig. 5.5.31. Obstacole detectate.	91
Fig. 5.5.32. Împărțirea spațiului ocupat în voxelii.	91
Fig. 5.5.33. Gruparea voxelilor în obstacole.	91
Fig. 6.1.1. Selecția punctelor 3D de interes: puncte pe sol (verzi), puncte de interes pentru detecția de obstacole (roșii), puncte mult deasupra solului (maro).	97
Fig. 6.1.2. a) FOV orizontal este divizat în rânduri și coloane; b) Spațiul comprimat.	98
Fig. 6.1.3. Reproiecția în imagine a spațiului comprimat	99
Fig. 6.1.4. Histograma bidimensională a punctelor 3D.	100
Fig. 6.1.5. Gruparea celulelor în obstacole	103
Fig. 6.1.6. Eroarea de reconstrucție a unui punct 3D	104
Fig. 6.1.7. Punctele 3D ale unui obstacol	105
Fig. 6.1.8. O direcție optică (coloană în spațiul comprimat – optical slice) și punctele propuse spre erodare ...	105
Fig. 6.1.9. Eroarea la nivel de coloană.	106
Fig. 6.1.10. Stânga: exemplu de obstacol, împreună cu apertura și punctele sale. Dreapta: direcțiile optice ale obstacolului și înfășurătoarea vârfului direcțiilor optice	107
Fig. 6.1.11. Înfășurătorile succesive. Prima e desenată cu albastru și ultima cu negru îngroșat.	108
Fig. 6.1.12. Un exemplu de eroziune improprie	108
Fig. 6.2.1. Detecție greșită în lipsa analizei distribuției verticale a punctelor 3D.	109
Fig. 6.2.2. Row-Buffer: a) imaginea gray-scale, b) Row-Buffer, c) Row-Buffer cu aspect 1:1 (e mai facil dpdv vizual pentru privitorul uman)	110
Fig. 6.2.3 Extinderea obstacolelor înalte	111

Fig. 6.2.4. Obstacole aeriale .....	111
Fig. 6.2.5. Celulele spațiului orizontal comprimat, pentru obstacolele din Fig. 6.2.4 .....	111
Fig. 6.2.6. Două spații de interes: a) pentru scena din Fig. 6.2.4, .....	112
Fig. 6.2.7. a) Imaginea comprimată etichetată, b) spațiul orizontal etichetat, c) obstacolele 3D.....	113
Fig. 6.2.8. Rezultate obținute folosind: a) un spațiu înalt, b) un spațiu scund, c) două spații .....	115
Fig. 6.2.9. a) Încă un exemplu în care obstacolul detectat este mai înalt decât cel real, din cauza unor puncte reconstruite greșit, b) imaginea perspectivă comprimată și porțiunea verticală fără celule.....	116
Fig. 6.2.10. Rezultatul după fragmentarea verticală .....	116
Fig. 6.2.11. Grupare greșită (a) și corectată (b) .....	117
Fig. 6.3.1. Fragmentarea unui obstacol. a) obstacolul inițial, .....	119
Fig. 6.3.2. Exemplu în trafic: obstacolul central (a) este fragmentat în patru sub-obstacole (b) .....	120
Fig. 6.3.3. Orientarea obstacolelor. a) dreptunghiul neorientat .....	122
Fig. 6.3.4. Alte rezultate: a) orientarea celui de-al doilea obstacol din dreapta nu a putut fi determinată, b) un caz ideal, c) din cauza grupării pietonului împreună cu poarta din spate (stânga), orientarea e improprie.....	122
Fig. 6.3.5. Zgomotul orientărilor măsurate .....	123
Fig. 6.3.6. Histograma zgomotului .....	124
Fig. 6.3.7 a) Traiectorie rectilinie, b) Variația orientării.....	124
Fig. 6.3.8 a) Traiectorie curbilinie, b) Variația orientării .....	125
Fig. 6.3.9. Un exemplu de obstacol necuboidal .....	126
Fig. 6.3.10. O posibilă fragmentare a unui obstacol ce nu respectă modelul cuboidal, în obstacole care-l respectă .....	126
Fig. 6.3.11. Fragmentarea nu este necesară: obstacolul din spate este parțial ocluzat de obstacolul din față, provocând o formă necuboidală a reconstrucției obstacolului din spate.....	127
Fig. 6.3.12. Fragmentarea nu este necesară: obstacolul este parțial ocluzat de limitele FOV ale sistemului stereo, provocând o formă necuboidală a reconstrucției obstacolului.....	127
Fig. 6.3.13. Înfășurătoarea punctelor 3D și cuboidul care înscrie un obstacol (top-view).....	128
Fig. 6.3.14. Un caz real desenat peste celulele spațiului comprimat .....	128
Fig. 6.3.15. Cuboid orientat spre stânga .....	129
Fig. 6.3.16. Cuboid orientat spre dreapta.....	130
Fig. 6.3.17. Pentru un obstacol neorientabil, se folosește o orientare de $0.99*(-\pi/2)$ . .....	130
Fig. 6.3.18. Denumirea celor patru colțuri (cu albastru) și a spațiului liber în cele patru colțuri .....	131
Fig. 6.3.19. Stabilirea laturilor vizibile .....	132
Fig. 6.3.20. Descompunerea la nivelul 1 .....	134
Fig. 6.3.21. Descompunerea la nivelul 2 .....	135
Fig. 6.3.22. Descompunerea la nivelul 3 .....	136
Fig. 6.3.23. Analiza ocluziei în partea stânga a obstacolului curent .....	138
Fig. 6.3.24. Analiza ocluziei în partea dreaptă a obstacolului curent .....	138
Fig. 6.3.25. Vizibilitatea unui obstacol în cele două camere .....	139
Fig. 6.3.26. Folosirea ocluziilor, la nivelul 3 de descompunere .....	140
Fig. 6.3.27. Cu (a) și fără (b) folosirea criteriului de fragmentare bazat pe convexități .....	141
Fig. 6.3.28. Alegerea punctului de fragmentare pentru exemplul din Fig. 6.3.27 .....	141
Fig. 6.3.29. Cu (a) și fără (b) folosirea criteriului de fragmentare bazat pe convexități .....	142
Fig. 6.3.30. Cu (a) și fără (b) folosirea criteriului de fragmentare bazat pe convexități .....	142
Fig. 6.4.1 Absorbția obstacolelor.....	145
Fig. 6.4.2 Clasificarea obstacolelor de fundal (desenate în roșu) .....	146
Fig. 7.1.1. Rezultate ale detecției de obstacole.....	152

## F. Articole selectate

Primul dintre ele, publicat în Proc. Computing, Communications and Control Technologies 2004 (Austin, Texas), prezintă cooperarea mai multor senzori de stereo viziune și fuziunea obstacolelor detectate:

### Extraction of Dynamic Traffic Description Using Multiple Stereovision Equipped Vehicles

Sergiu NEDEVSCI, Radu DANESCU, Dan FRENTIU, Tiberiu MARITA, Florin ONIGA, Ciprian POCOL  
Computer Science Department, Technical University of Cluj-Napoca  
3400 Cluj-Napoca, ROMANIA

#### ABSTRACT

In this paper an Intelligent Vehicle Interaction System is presented. By connecting neighbor vehicles into an ad-hoc wireless network, each of them can broadcast to the others their GPS position, sizes and dynamic behavior, allowing them to build a dynamic map of the local traffic. However, some pieces can be missed: not all the vehicles are equipped with such systems, damages can appear, general obstacles can exist on the road. The solution is to use obstacle detection sensors. A stereovision sensor can detect obstacles in front of the vehicle, but it is still not perfect having a limited field of view and a limited detection distance. If each stereovision equipped vehicle broadcasts its detection, all the receiver vehicles can improve their perception of the surrounding scene. Each involved vehicle has to assemble its detection results and the results provided by others, as well as the information reported by vehicles about themselves. All the traffic data must be in the same coordinate system, and the GPS coordinates are the most suitable for this. The main aim of the system is to offer assistance to the driver. Since current GPS navigation systems can provide only a static map, such a system can bring dynamic information about the surrounding traffic.

**Keywords:** stereovision, sensor fusion, symbolic environment description, communication, distributed computation, driver assistance.

#### 1. INTRODUCTION AND PROBLEM DESCRIPTION

Stereovision is becoming more and more popular as a 3D measurement tool, having the advantage of being a passive method and also of providing a rich amount of 3D data. A stereovision sensor can identify objects in front of it measuring their position and size. The detection can be noisy and can even miss objects, thus a tracking procedure can reduce the noise by averaging measurements and fill in the detection gaps.

Due to the fact that a single sensor covers a limited area, and because of the imperfection of detection, the fusion of multiple sensors becomes necessary.

Note: for the readability purpose of this paper, all the objects detected by the stereovision sensor are subject to be considered and named as "vehicles".

The field of view and the size of acquired images give the maximum distance of acceptable accuracy of 3D reconstruction and the object detection quality. For constant image size, larger field of view can see a larger slice of the scene, but it will cause less detail for far objects. For a constant field of view, higher image size brings more details and claims more processing power, since smaller images are less time consuming but far objects are worse perceived.

A good detection means high distance and large field of view. As shown above this would need large images and long processing time. For real-time detection, a short processing time is required. Thus, having the processing power of a high-end computer, the request of **real-time object detection up to 100m** is achieved with a 30 degrees horizontal field of view and VGA images (640x480).

In figure 1 it is shown a scenario of detected vehicles by the stereovision sensor mounted on vehicle V. Near vehicles (closer than 50m), like vehicle V3, are detected with high accuracy. Further vehicles (from 50m to 100m), like vehicles V4 and V5, are detected with lower and lower accuracy as the distance grows. Vehicles detected further than 100 meters, are considered not to be reliable in terms of their existence, and they have a low accuracy of their size and position. As the boldness of depicted vehicles V3, V4 and V5 shows, the quality of detection is lower for higher distances. Vehicles V1 and V2 are not detected at all, being out of the field of view (30 degrees). Vehicle V6 is too far. Also, partial or total occlusions can cost the quality of detection.

Having the same scenario, let's consider that in figure 2, vehicles V1 and V4 are equipped with such stereovision sensors.

For vehicle V1, the detection could be as follows:

- V2: very good;
- V4: acceptable;
- V5: totally occluded;
- V and V3: out of field of view;
- V6: too far, not detected.

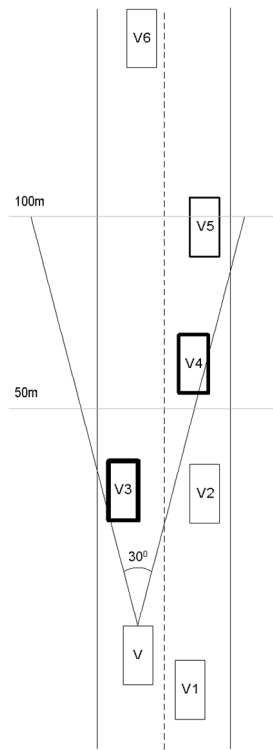


Fig 1. Vehicles V3, V4 and V5 are detected by vehicle V

For vehicle V4, the detection could be as follows:

- V5: very good;
- V6: acceptable;
- V, V1, V2 and V3: out of field of view.

If vehicles V, V1 and V4 are interconnected in a wireless network, each of them can get the detection results of the others and improve the perceiving of surrounding traffic. These vehicles are transmitters and receivers in the same time. Each of them, performs a reunion of its own detected vehicles and the vehicles detected by the others, by using a sensor fusion procedure to identify the common vehicles reported by different transmitters. Even more, the description about themselves is communicated to the others, having full reliability (its width and length are known a priori and the GPS information is reliable). If a vehicle doesn't have the stereovision sensor, but it has the GPS sensor and the wireless transceiver, it still can work by sending information about itself and fusing the information from the others.

A GPS sensor can provide information like: latitude, longitude, elevation, velocity, heading and time.

The transmission range of wireless networks is 250m. If enough stereovision equipped vehicles run on a 500m portion of road, the central one can have a very good description of the surrounding traffic.

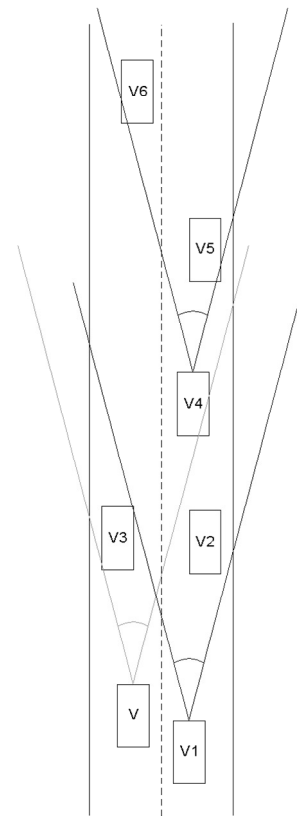


Fig 2. Vehicles V2, V4, V5 and V6 are detected by vehicles V1 and V4

## 2. DEFINITION OF THE ENVIRONMENT MODEL

The environment is described as a set of objects (vehicles). Each stereovision sensor outputs its own set of objects. The fusion algorithm will join all object sets into a final one.

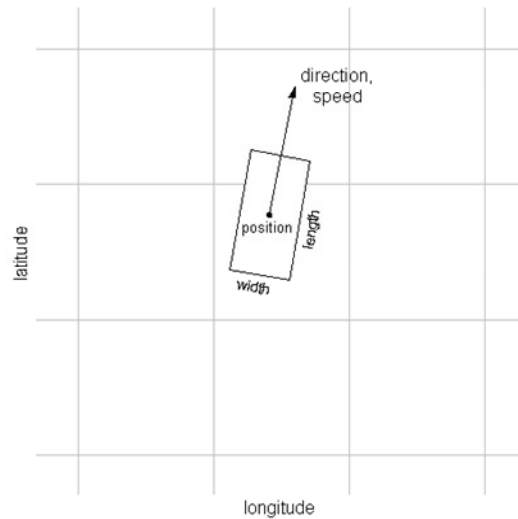


Fig 3. Vehicle's representation



Each sensor relieves 3D measurements of the perceived objects. Due to the specific of the application, the height of the vehicles is used only in the detection step. It is not used to describe the traffic because the space occupied on the road's surface is the aim of the project. But, if necessary, the height information could be used in a 3D rendered visualization. The used measurements are: length, width, 2D position on the Earth's surface, speed and direction of movement (fig 3).

The scene has associated a unique coordinate system. The data from all the stereovision sensors is relative to this unique coordinate system. The GPS coordinate system responds to this request.

### 3. THE SENSORIAL SYSTEM ARCHITECTURE AND FUNCTIONS

The stereovision sensor consists of two video cameras mounted on a rigid rig and a computing device (computer) – figure 4. The distance between cameras (base line) is about 30-40 cm. The rig is mounted inside the car, just before windshield. The stereovision detection is done in a coordinate system parallel with the car and with the origin situated where the GPS sensor is placed.

Using a wireless transceiver, our vehicle sends the set of detected vehicles, and receives the sets from other stereovision equipped vehicles located in nearby.

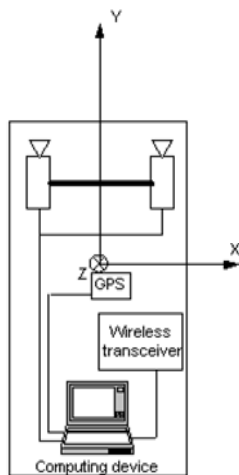


Fig 4. Needed equipment

The image processing computer performs stereo 3D reconstruction cycles on the synchronously acquired image pairs. The reconstructed 3D points are grouped into 3D objects (cuboids) which are tracked. The tracked objects represent the sensor's output and are broadcasted via the wireless transceiver. In the same time, the transceiver gets the sets of detected vehicles from the other stereovision equipped vehicles in nearby.

The synchronization of all the detected vehicles is done based on the GPS time. The GPS time is issued once per second by everybody. Aligned to the GPS time, at every 200 milliseconds, each sensor acquires a stereo pair of images, process it and broadcasts the set of detected vehicles together with the timestamp of images acquisition moment. In the period of 200ms from the acquisition moment, all the detection results should be broadcasted by transmitters and received by receivers and fused. The fusion is quite simple, having the same timestamp for all the data.

### 4. STEREOVISION SENSORS CALIBRATION

In order to reconstruct and measure the 3D environment using stereo cameras, the cameras must be calibrated. The calibration process estimates the camera's intrinsic parameters (which are related to its internal optical and geometrical characteristics) and extrinsic ones (which are related to the 3D position and orientation of the camera relative to a global world coordinate system).

The intrinsic parameters of each camera are calibrated individually. The estimated parameters are the focal length and the principal point coordinates and the lens distortions. The parameters are estimated by minimizing the projection error from multiple views of a set of control points placed on a coplanar calibration object with known geometry. For a stereo system of two cameras, the obtained intrinsic parameters can be refined by inferring the stereo information available. This is done by introducing a new constraint in the estimation process which considers also the projection error of the control points image coordinates from one image to another [1].

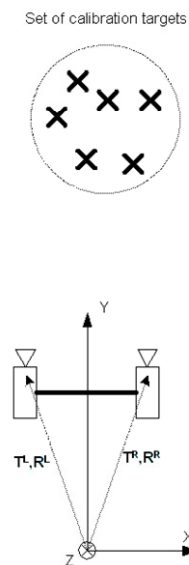


Fig. 5. Calibration setup for calibrating the extrinsic parameters.

The extrinsic parameters of the cameras are estimated by minimizing against the extrinsic parameters the projection error for a set of 3D control points with measured coordinates in a world reference system [2,3]. Each stereo pair of cameras is calibrated using a set of control points measured in the coordinate system of the vehicle the cameras are mounted on (fig. 5). The origin of the coordinate system is where the GPS sensor is mounted, this way eliminating the transformation of the detected vehicles from the car's coordinate system to the GPS coordinate system.

The obtained extrinsic parameters for each camera are a translation vector (T) and a rotation vector (R) of the camera in the car's coordinate system. In this way, the calibration allows us to measure the coordinates of the reconstructed 3D object in the car's coordinate system.

### 5. STEREO 3D RECONSTRUCTION

The stereo reconstruction algorithm used is mainly based on the classical stereovision principles available in the existing literature [4]: find pairs of left-right correspondent points and map them into the 3D world using the stereo system geometry determined by calibration.

Constraints, concerning real-time response of the system and high confidence of the reconstructed points, must be used. In order to reduce the search space, only edge points of the left image are correlated to the right image points. For robust detection of the image edges, a Canny-based [5] edge detector was implemented. By focusing to the image edges, not only the response time is improved, but also the correlation task is easier, since these points are placed in non-uniform image areas. The sum of absolute differences (SAD) function [6] is used as a measure of similarity, applied on a local neighborhood. Parallel processing features of the processor are used to implement this function. For a given left image point the search is performed along the epipolar line computed from the stereo geometry.

After this step of finding correspondences, each left-right pair of points is mapped into a unique 3D point [4]. Using the camera geometry, two 3D projection rays are traced, one for each point of the pair. By computing the intersection of the two projection rays, the coordinates of the 3D point are determined.

The result of reconstruction is a set of 3D points that must be clustered into objects. The grouping is performed mainly based on the local density of the points and the vicinity criteria: a local group of points must be dense enough to be considered as candidate and two points are considered to be in the same group if they are close to each other. Both these criteria are adapted to the fact that the density of reconstructed points per object decreases

with the distance (due to the perspective projection) and their positioning error increases with the same distance. For each cluster of points, the circumscribing box is built.

### 6. OBJECT TRACKING

Tracking is employed in order to estimate the dynamic parameters of an object. The dynamic parameters are the positions and the speeds on both X and Y axes. The geometric sizes can also be established filtering them against the detection noise. The geometric sizes are: width, length and height.

The position and speed of the object are tracked through a linear Kalman filter, using the uniform motion model (assumption of constant speed). The speed vector is established as the resultant of the speeds on individual axes. The modulus of the speed vector gives the speed value itself and the vector's orientation shows the direction of movement.

The size of the object is tracked up to a certain point, through a simple averaging of the individual measurements. After a certain number of frames, the size of the object is considered to be established, and the tracker will modify it no more.

The tracker will output the set of objects in the environment model format, being ready to be broadcasted.

### 7. DEALING WITH DIFFERENT COORDINATE SYSTEMS

The detection results of the stereovision sensor are relative to our vehicle. Some transformations have to be done, to move them into the global coordinate system.

In figure 6 our vehicle (V), is situated at position (*lat*, *long*)

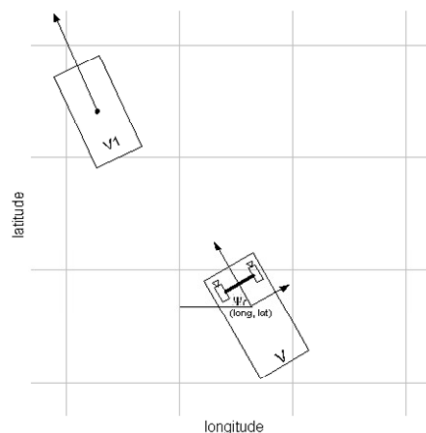


Fig 6. Our vehicle (V) and the detected vehicle (V1), in the global coordinate system.

The detected vehicles are in a coordinate system parallel with the car and with the origin situated where the GPS sensor is placed. By knowing the heading (orientation angle -  $\psi$ ) and the global position (longitude and latitude) of vehicle V (more precisely of the GPS sensor), the detected vehicles are easily converted in the global coordinate system.

Our vehicle is added to the set of detected vehicles. Thus, its center, which differs by the GPS sensor's place, has to be correctly expressed by knowing the GPS sensor placement in the vehicle's geometry.

## 8. SENSOR FUSION ALGORITHM

Not all of the received detected vehicles are usable from temporal and spatial aspects. Some messages can reach to the receiver too late, being outdated, and will be discarded. Other vehicles can be out of our space of interest. On highway, the broadcasted messages from the other driving direction are not important. Other similar situation is at intersection of highways, where one passes over other. Vehicles which are too far behind, don't represent a danger and can be disregarded. The space of interest can be considered 100m behind and 200m in front of us, even if messages can be transmitted over longer distances.

As mentioned, when transmitting the set of detected vehicles, first, data about itself is also transmitted in the same format as the detected vehicles. In this way, it is known that the first vehicle in a set, represents a full reliable one.

Each detected vehicle has associated a confidence degree, depending on its distance, number of 3D points inside it and the number of frames it was observed by tracking, all of these being weighted with their importance. Our vehicle has a 100% confidence degree.

When attempting to fuse the results of the sensors into a global result, we must make the difference between the case when an object is detected by only one sensor, and the case when it is detected by two or more sensors. In the first case, the act of fusion is simply to add this object to the global result set. In the second case, the result must be a combination of the sensor readings, taking into consideration the confidence degrees of each sensor.

The main simplification of the problem comes from the fact that the cuboids are defined in the same coordinate system.

The criterion that two objects occupy the same space (and therefore they could be joined) is literally interpreted: the intersection area of their rectangles is computed (figure 7). If this area, represents at least 50% of the smallest

rectangle, the two objects can be joined. Because of imperfection of detection, tracking and GPS position, it is possible to have two very close objects obeying our criterion, apparently the same one, but if their direction of movement differs too much they will not be joined.

If they are declared as joinable, the resulted one is computed from the two as a weighted sum of their parameters, using the confidence degree as the weight.

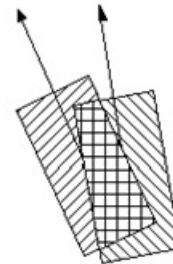


Fig. 7 Intersection area of two vehicles

## 9. RESULTS

For testing of the algorithm we have used two stereovision setups. The two setups were calibrated using the method described in the calibration section, having the coordinate system on the ground, just under the stereo rig and oriented as the cameras. The perspective views of the scene for each stereovision sensor are presented in the left side of (fig. 8.a and 8.b). The reconstruction results for each stereovision sensor is presented as a bird-eye view of the scene in the right part of the same images, and as white cuboids projected on the original perspective image.

The two sets of results were gathered and, by knowing the relative position of the sensors, the fusion could be done. The position of the first sensor was considered as being the origin of GPS coordinate system.

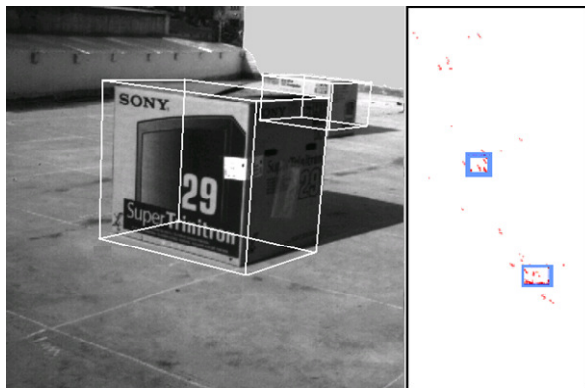
The fusion results are displayed in fig. 8.c as a bird-eye view. The final result corresponds to the aim of the algorithm: combining together the scene description of different sensors and refining the measurements of each sensor against each other, in the case where the same object is viewed by more than one sensor and in the case where an object is not viewed by one sensor but it is viewed by other one.

A highway scenario is presented in fig. 9.

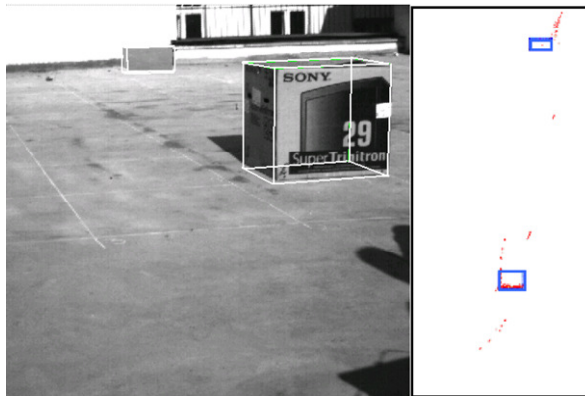
## 10. CONCLUSIONS

A method for extracting the 3D scene description from multiple stereovision sensors has been presented. The stereovision sensors are able to perform real-time image pair processing and extract 3D points of the environment. These points are then grouped into cuboids which are

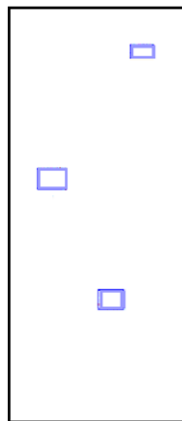
tracked in successive frames. The scene description of each sensor is sent to the other sensors through a wireless connection. Fusion of scene descriptions is performed in order to obtain a more complete description of the scene. The fused description has the advantage of increasing global field of view by uniting the fields of view of each sensor, and the advantage of refining the description of individual objects, if they are viewed by more than one sensor.



a – Results of the first stereo sensor

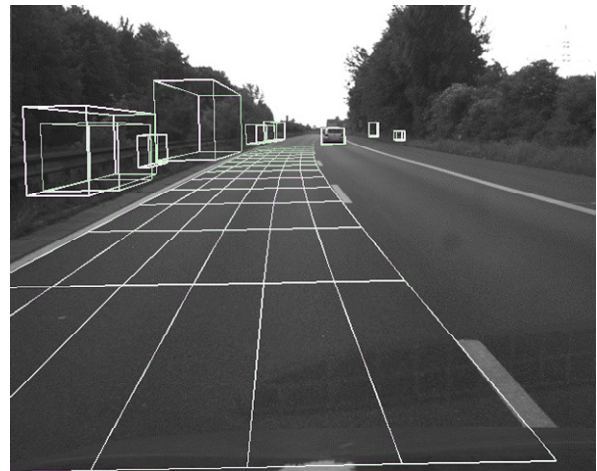


b – Results of the second stereo sensor

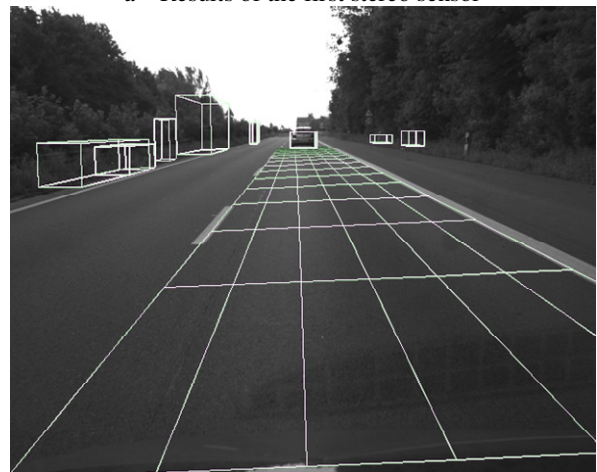


c – Results of the sensor fusion

Fig. 8. Stereo reconstruction and object fusion results



a – Results of the first stereo sensor



b – Results of the second stereo sensor

Fig 9. Highway scenario

## 11. REFERENCES

- [1] Jean-Yves Bouguet, **Camera Calibration Toolbox for Matlab**, MRL - Intel Corp., [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/), 2003.
- [2] S. Nedevschi, T. Marita, M. Vaida, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, D. Moga, **Camera Calibration Method for Stereo Measurements**, *Journal of Control Engineering and Applied Informatics (CEAI)*, Vol.4, No. 2, pp.21-28, 2002, Bucuresti, Romania.
- [3] S. Nedevschi, T. Marita, R. Danescu, F. Oniga, D. Frentiu, C. Pocol, **"Camera Calibration Error Analysis in Stereo Measurements"**, *microCAD International Scientific Conference, March 2003, Miskolc, Hungary*, pp. 51-56.
- [4] Trucco E., Verri A, **Introductory techniques for 3D Computer Vision**, New Jersey: Prentice Hall, 1998.
- [5] J. Ramesh, R. Kasturi, B. G. Schunk, **Machine Vision**, New York: McGraw-Hill Inc., 1996.
- [6] Todd A. Williamson, **A High-Performance Stereo Vision System for Obstacle Detection**, Ph.D. Thesis. Carnegie Mellon Technical Report, September 1998

Al doilea articol prezintă întregul sistem de stereo viziune și rolul detecției de obstacole în cadrul sistemului, și a fost publicat în Proc. Intelligent Vehicles 2007 (Istambul, Turkey):

## A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision

Sergiu Nedevschi, Radu Danescu, Tiberiu Marita, Florin Oniga, Ciprian Pocol, Stefan Sobol, Corneliu Tomiuc, Cristian Vancea

Technical University of Cluj-Napoca, ROMANIA  
e-mail: Sergiu.Nedevschi@cs.utcluj.ro

Marc Michael Meinecke, Thorsten Graf,  
Thanh Binh To, Marian Andrzej Obojski

Volkswagen A.G. GERMANY  
email: marc-michael.meinecke@volkswagen.de

**Abstract**—The urban driving environment is a complex and demanding one, requiring increasingly complex sensors for the driving assistance systems. These sensors must be able to analyze the complex scene and extract all the relevant information, while keeping the response time as low as possible. The sensor presented in this paper answers to the requirements of the urban scenario through a multitude of detection modules, built on top of a hybrid (hardware plus software) dense stereo reconstruction engine. The sensor is able to detect and track clothoid and non-clothoid lanes, cars, pedestrians (classified as such), and drivable areas in the absence of lane markings. The hybrid stereovision engine and the proposed detection algorithms allow accurate sensing of the demanding urban scenario at a high frame rate.

### I. INTRODUCTION

Urban Driving Assistance Systems have little in common with the systems targeted for highways. Not only is the problem itself more complex, with difficult and atypical road geometries, crowded traffic, the presence of pedestrians and other type of traffic participants which are not normally found on the highway, but the expectations from such a system are considerably higher. The detection errors are not tolerated, the measurement errors need to be considerably smaller, and the field of view must be considerably increased, to account for the lateral objects. The thematic network ADASE (Advanced Driving Assistance Systems in Europe, [www.adase2.org](http://www.adase2.org)) helps define the driving assistance requirements by harmonizing and communicating active safety functions, identifying technological needs and focusing on essentials and preparing architectures, roadmaps and standards.

A realistic analysis of the requirements and of the possibilities of the urban traffic environment leads to the establishment of several goals for urban traffic assistance, to be implemented in the near future:

- Follow-to-Stop
- Stop to fixed/ non-moving Obstacles/ Vehicles
- Go Inhibit if Objects are in front of the own Vehicle
- Lateral Support in narrow road conditions
- Set of Max-Velocity depending on lanes width

Manuscript received 15.01.2007. The work was funded by Volkswagen A.G. in the framework of the DESBOR (Dense Stereo-Based Object Recognition) project.

- Go (automatically/ driver initiated)
- In order to achieve these urban ACC applications, a vision sensor must provide the following functions:
- Lane Detection / Lane Parameters Estimation
  - Navigable channel detection and channel parameters estimation in crowded environments
  - Vehicle Detection and tracking
  - Detection of fixed (non-moving) Obstacles
  - Classification of Pedestrians

The sensorial systems for driving assistance (highway and urban) are today the focus of large, joint research projects, which combine active and passive sensors, GPS navigation, and telematics. Projects such as INVENT ([www.invent-online.de](http://www.invent-online.de)), PREVENT ([www.prevent-ip.org](http://www.prevent-ip.org)), CARSENSE ([www.carsense.org](http://www.carsense.org)) bring together car manufacturers and research partners for the common goal of solving the driving assistance problem. Sensing in urban environments is also a long-time effort of the DaimlerChrysler research department [15].

Many new research papers deal with problems that are present in the urban driving environments. The researchers at Toyota [16] present a stereovision-based system combined with a near infrared projector for road and obstacle detection in any environment. The unstructured scenario driving is approached in [12], [1], and a combined, structured plus non-structured system is presented in [13]. A dense stereo system for obstacle detection for go inhibit, with possible applications in urban environments is presented in [7]. New methods for lane detection, suitable for urban environments, are presented in [2] and [6].

The research team of the Technical University of Cluj Napoca, in cooperation with Volkswagen AG, has already implemented a stereovision-based sensor for the highway environment [10]. This sensor was able to detect the road geometry and the obstacle position, size and speed, from a pair of synchronized grayscale images, using edge-based software stereo reconstruction with general geometry.

The urban scenario required important changes in the detection algorithms, which in turn required more stereo information. Thus, the edge-based stereo engine was discarded, and replaced with a dense stereo system. A software dense stereo system being time consuming, a hybrid solution was chosen: software rectification and down sampling, followed by hardware correspondence search. The



time gained by the hardware part compensated the increase in complexity of the new algorithms.

The dense stereo information is vital for the new obstacle reconstruction module, which extracts oriented objects even in serious clutter, and also allows better shape segmentation for recognition of pedestrians. Dense stereo information allows us to compute and track an unstructured elevation map, which provides drivable areas in the case when no lane markings or any other road delimiting features are present or visible.

Lane detection requires edges, but the vertical profile is better computed from dense stereo information. The edge based lane detection algorithms are completely changed, adapted to the limited and variable viewing distance of the urban environment. A freeform lane detection module was added, in order to solve the problem of the non-standard geometry roads.

The dense stereovision based sensor presented in this paper provides complex and accurate functionality on a conventional PC architecture, covering many of the problems presented by the urban traffic environment, and promising to be a valuable addition to a driving assistance system.

## II. STEREOVISION SYSTEM ARCHITECTURE

### A. Image Acquisition

The hardware acquisition system (fig. 1) includes two JAI CV-M4+CL cameras with 2/3" (1380x1030) CCD sensors and 6.5 mm fixed focal length lenses, allowing a horizontal field of view (HFOV) of 72 [deg]. The cameras are mounted on a rigid rig with a baseline of 320 [mm] (fig. 2). The images are acquired at full resolution with a microEnable3-XXL acquisition board with a maximum frame rate of 24 fps.

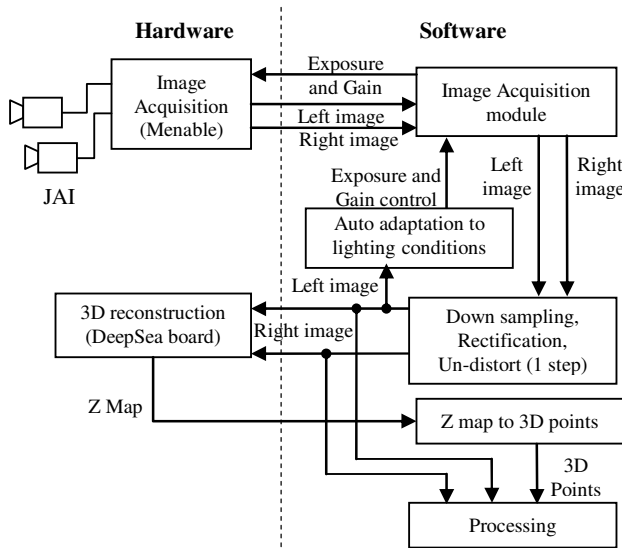


Fig. 1. The stereovision system architecture.

The camera parameters are calibrated using a dedicated method optimized for high accuracy stereovision [14] using the full resolution images.

The images are further enhanced by lens distortion

correction and rectified in order to fulfill the dense stereo reconstruction requirements (canonical images). A down-sampling step is used to adapt the image size to the DeepSea board parameters (512 pixels width) and to minimize the noise introduced by the digital rectification and image correction. The whole process is reduced to an image warping approach performed in a single step (fig. 1) using reverse mapping and bilinear interpolation [3]. An optimized implementation using MMX instructions and lookup tables was used in order to minimize the processing time.

### B. 3D Reconstruction

The 3D reconstruction of the scene is performed using the DeepSea hardware board provided by TYZX [5]. The input of the board consists in two rectified images and the output can be either a disparity or a Z map (left camera coordinate system). Our system uses 3D points set for scene representation; therefore the preferred output is the Z map. Using the Z coordinate value, the X and Y coordinate can be computed and then transformed into the car coordinate system.

### C. Detection Range

With the current system setup a detection range optimally suited for the urban environments is obtained (fig. 2):

- minimum distance: 0.5 m in front of the ego car;
- delimiters of the current lane are visible at 1.0 m;
- reliable detection range: 0.5 ... 35 m;

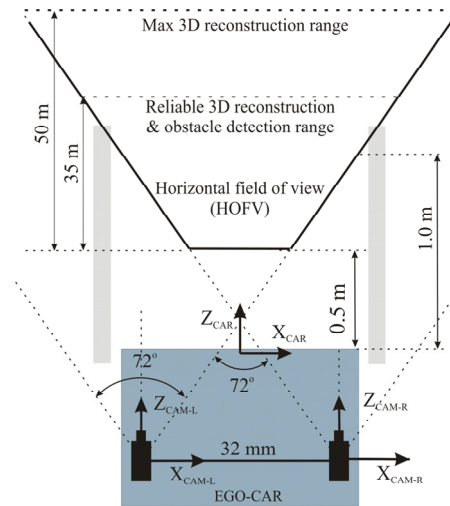


Fig. 2. Detection range of the current stereo system setup.

## III. LANE DETECTION

The urban environment brings new requirements for the lane detection system. The scenes are more complex, the presence of nearby vehicles and other obstacles shorten the visibility range, and the geometry of the lanes does not always comply with the clothoid model. Therefore, a serious

method overhaul was needed.

The first step is to detect the vertical profile and to separate the road obstacle points. The pitch angle and the vertical curvature are detected using histograms, in a manner similar to the Hough transform [11].

The highway lane detection approach required little information about lane markings, because it could rely greatly on the lane model. For the urban environment, however, we require a fast and robust lane marking extraction algorithm.

The lane marking extraction method relies on the well-known dark-light-dark transition detection [4]. We have to search for pairs of gradients of opposing sign and equal magnitude. We have improved the method by using a variable filter for computing the horizontal gradient. The size of the filter is the size of a standard width lane marking projected in the image space, and varies because of the perspective effect. Applying the variable width filter we preserve the level of detail in the distance while filtering the noise in the near areas.

The gradient maxima and minima are paired and the DLD pairs are extracted as lane markings. The complete technique is described in [8].

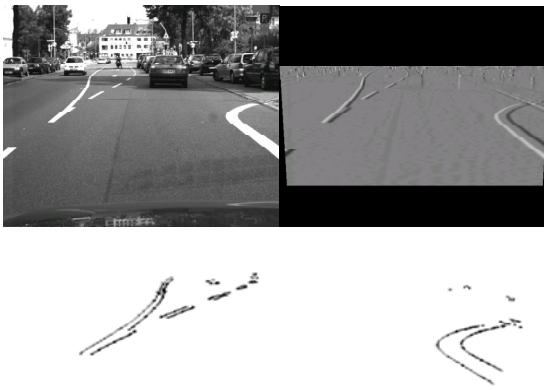


Fig. 3. Lane marking detection – top left, original image; top right, results of the adaptive gradient filter; bottom, lane marking results

Although the clothoid model is not always accurate for the urban scenario, it has several benefits, such as good results when the lane is delimited by simple edges (unmarked roads). Due to the short visibility range, we have decided to avoid matching the whole clothoid model on the image data, but to match pairs of line segments instead, in two zones: near and far.

First, we make an attempt for the near zone (2m to 5 m). Hough transform is performed on the image edges corresponding to the road points, and line segments are extracted. Lane markings will have a higher weight in the Hough bins, and therefore they will have a higher priority.

We divide then the line segments in two sets – left and right. The segments on the left are paired with the segments on the right, and the best pair is selected as our lane measurement. The linear measurement will update the clothoidal model parameters using the Extended Kalman Filter. If the linear fit for the near zone is successful, the same is done for the far zone, and the model parameters are updated again.



Fig. 4. Updating the clothoidal lane profile from linear segments using the Extended Kalman Filter

Sometimes the clothoid lane model is not suited for the road we wish to observe, and the detection will be incorrect. For these cases, a freeform lane detection system has been implemented. Because we don't have strong models to guide our search, we have to discard the non-marking delimiters, and work with lane markings only. The markings are projected onto a top view image, and then distance transform is performed, to facilitate the fitting of a lane border. The left and the right lane borders are represented as Catmull-Rom splines with four control points. The lateral coordinates of the four control points are the model parameters, and they are found using a simulated annealing search strategy in the model space.

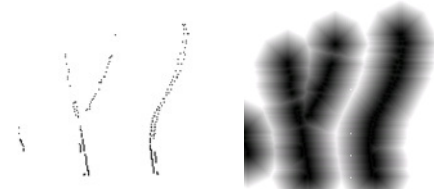


Fig. 5. Top view of the lane markings and the distance transform image used for freeform lane matching

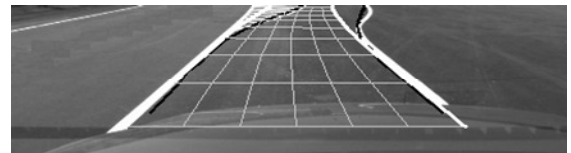


Fig. 6. Freeform lane detection succeeds in situations where the clothoid model fails

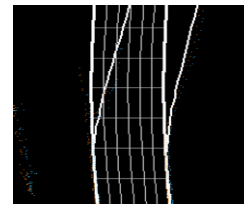


Fig. 7. Freeform detection versus model-based detection, bird-eye view comparison

#### IV. DRIVABLE AND NON-DRIVABLE AREAS DETECTION

There are some urban scenarios where the 3D lane cannot be detected, especially when not enough lane delimiters exist (ex. road crossing). An alternative method must be used to detect elevated areas (obstacles), regions where the ego vehicle cannot be driven. Complementary, the obstacle-free road areas can be considered as drivable.

The dense stereo engine usually reconstructs most of the road surface points even if lane markings are not present. Thus, the surface of the road can be computed by fitting a geometric model to the 3D data. The fitting must be performed in a least-square fashion (LSQ), or, more robustly, using a statistical approach (ex. RANSAC). The model used for the road is a planar one, allowing for non-zero pitch and roll angles of the ego car. The algorithm can be extended to fit complex surfaces, such as quadratic or cubic.

The 3D data available is a set of 3D points (80,000 to 120,000). Fitting the road surface to this set, in real-time, is not possible because it has a high computational complexity. A (bird-eye rectangular, 12x35 meters) region of interest of the 3D space can be represented similar to a digital elevation map. An image of elevations is formed, with each pixel (cell) having the intensity proportional to the 3D height. If a cell has more than one 3D point, then the greatest height is used. Morphological dilation is used to fill voids and compensate for the perspective effect (the 3D space gets sparser with the depth).

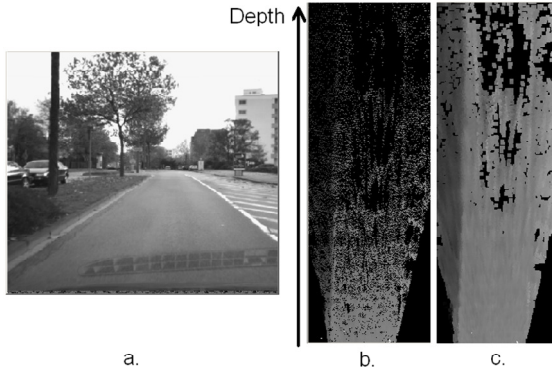


Fig. 8. The urban scenario (a) and the elevation image (b. initial, c. with morpho-dilation). Darker means more elevated.

Since the LSQ fitting is sensitive to noise in the data set, a pre-selection of candidate road points is required. Most of the 3D points in front of the ego car are road points if no nearby large obstacle is occluding the road. A histogram of intensities from the lower part of the elevation image is computed (fig. 9) and the dominant intensity is selected. Candidate road points are those having the intensity close to the dominant one (in the range depicted by arrows in figure 9.b).

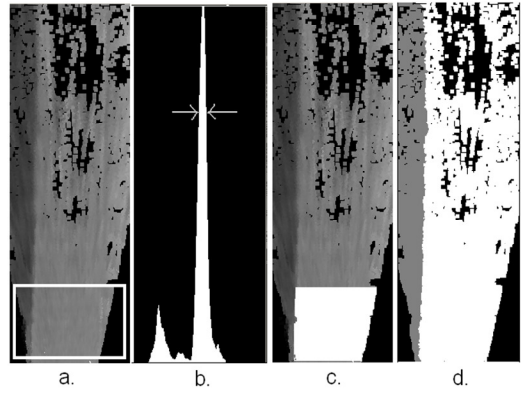


Fig. 9. The road level is selected from the patch in a, using the histogram in b. Candidate points are shown in c. The final result is presented in d (white means road and gray means elevated).

The road model is fitted to the selected road points in a LSQ fashion by minimizing the error along the vertical direction (vertical axis in the 3D space). A better approach can be used, the RANSAC algorithm, if the image quality is poor and a lot of noise exists in the 3D data.

For the final classification into drivable/non-drivable areas, the depth uncertainty model from [9] was extended to a height uncertainty model (1). The expected uncertainty  $Y_{err}$  is a function of the height  $Y$  and the depth  $Z$  of the 3D point, height of the camera  $H_{cam}$ , and the estimated depth uncertainty  $Z_{err}$ .  $Z_{err}$  is also a function of the system parameters and the expected disparity uncertainty. The disparity uncertainty was chosen experimentally as 1.5 pixels, although a more complex model for estimating the correlation's accuracy can be developed.

$$Y_{err} = \left| \frac{(Y - H_{cam}) * Z_{err}}{Z} \right| \quad (1)$$

Each point is labeled as road if it is closer to the road surface than its estimated height uncertainty. The result for the scenario in figure 8.a is shown in figure 9.d. Another result is presented in figure 10, re-projected as a grid onto the left image.

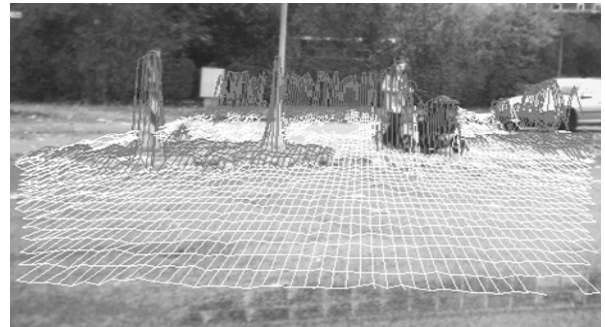


Fig. 10. The result re-projected back as a grid onto the left image: pedestrians, sidewalks and vehicles (dark gray) are separated from the road surface (white).



## V. OBSTACLE DETECTION

The road/obstacle separation is done by using the lane's vertical profile. The 3D points situated above the road, but not above a convenient height relative to the road (let's say 1.5 meters), are used by the obstacle detection algorithm (fig. 11.b).

Ideally, an obstacle detection algorithm should not divide a real obstacle into smaller detected obstacles and not merge more real obstacles into one detected obstacle; two opposite requirements.

Previous experience has shown that these requirements are hard to be fulfilled due to reconstruction errors and limitations.

The current approach prefers to merge more real obstacles into one detected obstacle, named **occupied area**, and then, to fragment it.

It is supposed that the obstacles do not overlap each other; no obstacle is hanged up in the air above other one. In other words, in a top view (fig. 11.c) the obstacles are disjoint.

Consequently, in what follows, the Y coordinate (the elevation) of the 3D points will be ignored and all the processing is done using only the X and Z coordinates (top view).

To counteract the problem of the points' density (less points for higher distances), a schema to divide the cartesian top view space into tiles with constant density is proposed. (fig. 11.c). The horizontal field of view of the camera is divided into slices with constant aperture – counteracts the sparseness on the X-axis. The depth range is divided into intervals, the length of each interval being bigger and bigger as the distance grows – counteracts the sparseness on the Z-axis.

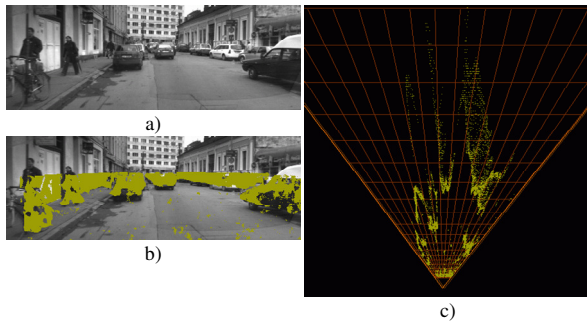


Fig. 11. Tiles division. a) Gray scale image, b) 3D points – perspective view, c) 3D points – top view; the tiles are here considerably larger for visibility purpose. Wrong reconstructed points can be observed in random places. Reconstruction error is visible as well (spread points for further obstacles).

A specially **compressed space** is created (fig. 12.a). The cells of the compressed space correspond to the

trapezoidal tiles of the cartesian space. The compressed space is, in fact, a bi-dimensional histogram, each cell counting the number of 3D points found in the corresponding trapezoidal tile (fig. 12.a). The cells having no points represent free space. The cells having just a few points are also considered free, most probably that those points were wrong reconstructed. The other cells, having many points, denote the existence of obstacles.

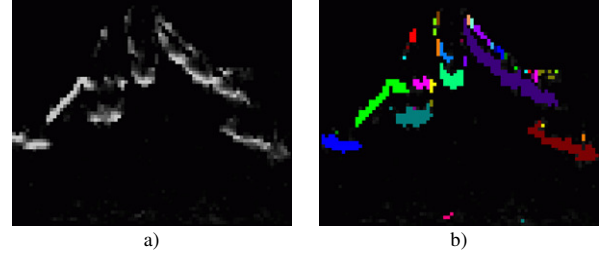


Fig. 12. The compressed space (for scene in figure 11) – a bi-dimensional histogram counting 3D points.

On the cells (having many points) of the compressed space, a labeling algorithm is applied: it groups adjacent high density cells (fig. 12.b). Each group of cells, obtained by the labeling algorithm, represents an **occupied area**. The small groups are filtered out.

### Fragmentation of occupied areas into primitive obstacles

A **primitive obstacle** is defined as an obstacle without concavities. An obstacle with concavities must be fragmented into primitive obstacles.

To ease the understanding of the process, a very simple example is shown in figure 13: an inward corner of a building.

The idea here is to determine the envelope of the cells of an occupied area, and then for each side of the envelope, the concavity between the side and the occupied cells is determined. If it is big, there are in fact two primitive obstacles (or occupied areas) and the deepest point of the concavity gives the column where the division will be done. The two sub-parts are subject to be divided again and again as long as concavities are found.

Example. In figure 13.c the bottom side of the envelope for the cells in figure 13.b delimits a big concavity. For each new sub-part, the envelope of the cells has been calculated again (and painted as well), but without revealing big concavities for new divisions to be performed.

By reconsidering the coordinates (including Y) of the 3D points that have filled the cells of an obstacle, the limits of the circumscribing box are determined. Boxes are shown in figure 13.d (perspective view) and figure 13.e (top view).

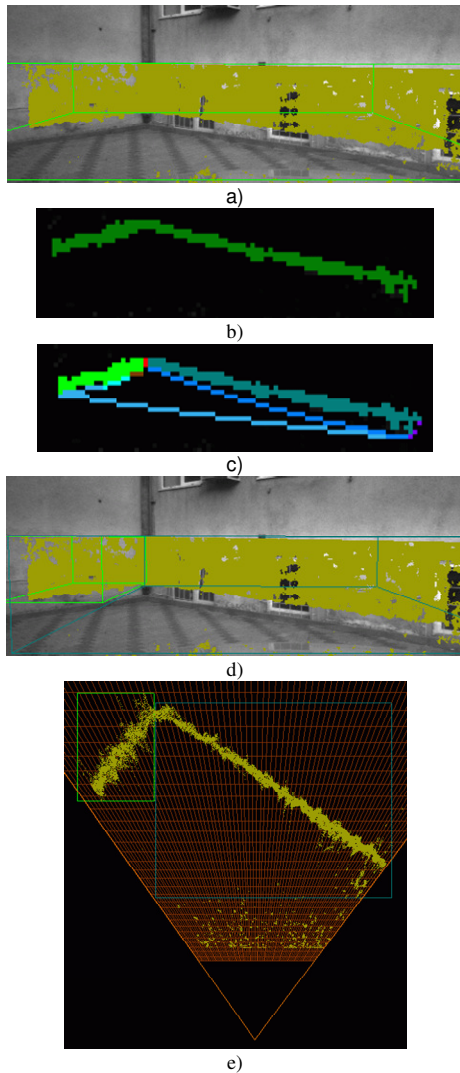


Fig. 13. Fragmentation of occupied areas into primitive obstacles. a) an occupied area, b) the labeling in the compressed space, c) sides of the envelope and the two primitive obstacles of the occupied area – compressed space, d) the two primitive obstacles – perspective view, e) the two primitive obstacles – top view

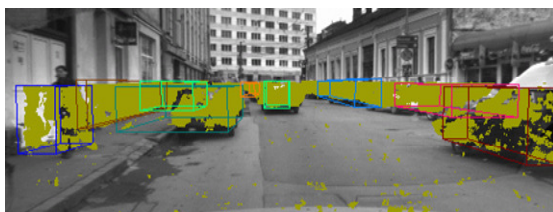


Fig. 14. Obstacle detection results

## VI. OBJECT CLASSIFICATION

The object classification is performed using a new approach to pattern matching exploiting both 2D image information and 3D dense stereo information. Because the 3D information accuracy does not allow the direct

classification of the 3D shape, a combined 3D-2D method is used. The 3D object information consists of 3D position, size and aspect ratio, obtained from dense stereo data through a grouping process. The 3D data is used for model selection and scale estimation. Based on the 3D orientations of the object, the appropriate 2D views of the model are generated from a database of 3D models. The corresponding 2D image window is selected based also on the 3D object information. The scaled models are matched against the features found in the selected window using an elastic high speed matching based on chamfer distances.

An important advantage in the use of dense 3D information is the possibility to make a better separation between the object and its background. This separation is done based on the difference in depth between the object of interest and the background behind it. A clear separation of the features belonging to the object allows the use of pattern matching classification techniques with a higher success rate since no additional noise derived from the object's background will be considered in the feature set for the classification process.

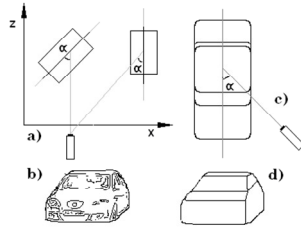


Fig 15. Object Segmentation

To be able to handle a wide variety of objects the model database can contain a mixture of 2D and 3D models. For simple objects with a rigid structure, the model database can contain 3D models. The orientation of the object relative to the camera is used in the classification process to transform the 3D models in 2D models using a projection according to the specific viewing angle. This allows the classification of objects that have a general orientation related to the observer. For objects that have a more complex shape or do not have a rigid structure, 2D models representing different points of view and different states are stored directly in the database. The orientation of the object can still be used to select the correct 2D model for these complex objects. The direction of movement associated with the 3D object is used to determine if the object is seen from the front or from the rear, or for pedestrians to determine which way they are facing.

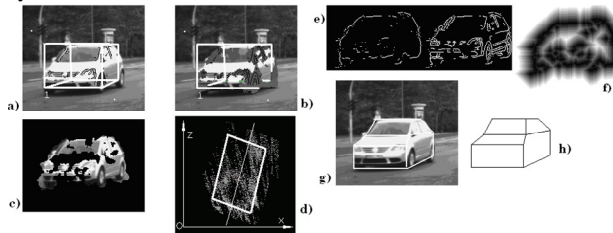
The model selection is done based on the 3D size and aspect ratio of the object (car models, truck models, pedestrian models, etc).

The 3D sensorial information provides the exact positions and dimensions of the objects in the scene. By projection the size and position of the objects in the 2D image can be evaluated and the model scaling factor can be deduced.



a) Top view of 3D object with orientation b) Object features c) 3D model with point of view d) Model projection  
Fig. 16. 2D Model Determination

The use of dense stereo information in the classification process provides significant improvements both in speed and success rate for the classification process. The 3D information allows the direct determination of the parameters for the model (type, scale factor, approximate position) which leads to a reduction in computational time. The selection of relevant features for the object (background separation) improves the success rate of the classification system.



a) Object Selection (ROI Selection) b) Used 3D Information c) Image Feature Selection d) Object Orientation e) Edge Selection f) Distance Transform g) Result of Pattern Matching h) Model Used  
Fig. 17. Results

The same approach has been successfully used in the case of pedestrian classification. The only modification in the process is the use of a hierarchical model database to reduce the number of pedestrian models used in the pattern matching stage.



Fig. 18 Pedestrian Classification

## VII. EXPERIMENTAL RESULTS

The dense stereovision based sensorial system was subjected to extensive test situations, targeted for each algorithm. The tests included recorded urban images from Wolfsburg (Germany) and Cluj Napoca (Romania), in light, medium and heavy traffic, in different times of the day and different weather conditions. The system was also tested online (on board of the vehicle, while driving) in the same

situations. The lane detection, object detection and pedestrian detection algorithms performed correctly under a vast majority of the test situations, and the failed test results provided valuable information for future algorithm improvement. The detection range was limited to the stereo reconstruction capability of a 6.5 mm focal distance camera, 35 m, a tradeoff for an increase in the field of view. The time performance of the whole processing cycle on Pentium Core 2 Duo E6600 architecture (2.4 GHz) is 20 frames per second.

## VIII. CONCLUSIONS

We have proposed and implemented a sensor system dedicated to urban driving assistance applications, built on the infrastructure of a dense stereovision engine. The dense stereo information provides the foundation for a new object reconstruction algorithm, dedicated to identify objects in the city clutter, and also allows the identification of drivable/non drivable areas in the case when no lane delimiting information is visible. Lane detection takes advantage of a better vertical profile provided by dense stereo, but its model matching algorithms were also changed to cope with the urban situations. Freeform lane border estimation was introduced to cope with the non-clothoidal road geometries.

The results give reasons to believe that dense stereovision is a valuable tool for the driving assistance sensorial systems, and draw the directions of the future work: the lane detection system will be enhanced by adding a curb extraction algorithm, and by combining the freeform and model based results using a fusion module; the object detection will benefit from orientation-based tracking; the drivable/non drivable area results will be fused with the obstacle detection and with the lane detection results.

## REFERENCES

- [1] Britta Hummel, Soeren Kammel, Thao Dang, Christian Duchow, Christoph Stiller, Vision-based Path Planning in Unstructured Environments, in Proceedings of IEEE Intelligent Vehicles Symposium 2006, Tokyo, Japan, June 13-15, 2006, pp. 176-181
- [2] C. Duchow, "A Novel, Signal Model Based Approach to Lane Detection for Use in Intersection Assistance", IEEE ITSC 2006, Toronto, Canada, September, 2006, pp. 1162-1167
- [3] C. Vancea, S. Nedevschi. Analysis of different image rectification approaches for binocular stereovision systems. *IEEE 2<sup>nd</sup> International Conference on Intelligent Computer Communication and Processing (ICCP 2006)*, vol. 1, 1-2 September 2006, Cluj-Napoca, Romania, pp. 135-142
- [4] J. Goldbeck, B. Huertgen, "Lane Detection and Tracking by Video Sensors", In *Proc. IEEE International Conference on Intelligent Transportation Systems, October 5-8, 1999, Tokyo Japan*, pp. 74-79
- [5] J.I. Woodfill, G. Gorden, R. Buck, Tyx DeepSea High Speed Stereo Vision System, IEEE Conference on Computer Vision and Pattern Recognition, Washington, D.C., 2004, pp. 41-45
- [6] Labayrade, Raphaël, Aubert, Didier, Douret, Jérôme, "A Multi-Model Lane Detector That Handles Road Singularities", IEEE ITSC 2006, Toronto, Canada, September, 2006, pp. 1143-1148
- [7] Massimo Bertozzi, Alberto Broggi, Paolo Medici, Pier Paolo Porta Agneta Sjogren, "Stereo Vision-Based Start-Inhibit for Heavy Goods Vehicles", IEEE Intelligent Vehicles Symposium 2006, Tokyo, Japan, June 13-15, 2006, pp. 350-355

- [1] R. Danescu, S. Nedevschi, "Robust Real-Time Lane Delimiting Features Extraction", in proc. of *International Conference on Intelligent Computer Communication and Processing, (ICCP) 2006, Cluj Napoca, Romania*
- [2] S. Nedevschi, F. Oniga, R. Danescu, T. Graf, R. Schmidt, Increased Accuracy Stereo Approach for 3D Lane Detection, *Proceedings of IEEE Intelligent Vehicles Symposium, (IV2006)*, June 13-15, 2006, Tokyo, Japan, pp. 42-49.
- [3] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, T. Graf, R. Schmidt, "Driving Environment Perception Using Stereovision", *Proceedings of IEEE Intelligent Vehicles Symposium, (IV2005), June 2005, Las Vegas, USA*, pp.331-336
- [4] S. Nedevschi, R. Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, "3D Lane Detection System Based on Stereovision", *IEEE Intelligent Transportation Systems Conference (ITSC), 2004, Washington, USA*
- [5] S. Thrun, M. Montemerlo, A. Aron, "Probabilistic Terrain Analysis for High-Speed Desert Driving", Robotics Science and Systems Online Proceedings, [www.roboticsproceedings.org](http://www.roboticsproceedings.org)
- [6] Sascha Kolski, Dave Ferguson, Mario Bellino and Roland Siegwart, Autonomous Driving in Structured and Unstructured Environments, IEEE Intelligent Vehicles Symposium 2006, Tokyo, Japan, June 13-15, 2006, pp. 558-563
- [7] T. Marita, F. Oniga, S. Nedevschi, T. Graf, R. Schmidt, Camera Calibration Method for Far Range Stereovision Sensors Used in Vehicles, *Proceedings of IEEE Intelligent Vehicles Symposium, (IV2006)*, June 13-15, 2006, Tokyo, Japan, pp. 356-363
- [8] U. Franke, D. M. Gavrila, S. Görzig, F. Lindner, F. Paetzold and C. Wöhler, "Autonomous Driving Approaches Downtown", *IEEE Intelligent Systems*, vol.13, nr.6, pp. 40-48, 1998.
- [9] Yasuhiro Takagi, Setsuo Tokoro, Masayuki Usami, Tatsuya Shiraishi, "The Development of Stereo Image Obstacle Recognition System", World Automotive Congress 2006 (FISITA 2006), Yokohama, Japan



# Obstacle Detection Based on Dense Stereovision for Urban ACC Systems

Ciprian Pocol<sup>\*</sup>, Sergiu Nedevschi<sup>\*</sup>, Marc-Michael Meinecke<sup>\*\*</sup>

<sup>(\*)</sup> Technical University of Cluj-Napoca  
Computer Science Department, Gh. Baritiu Street 26, Romania  
{ciprian.pocol, sergiu.nedevschi}@cs.utcluj.ro

<sup>(\*\*)</sup> Volkswagen AG, Electronic Research, P.O.Box 1776, 38436 Wolfsburg, Germany  
marc-michael.meinecke@volkswagen.de

**Abstract**—The dense (all pixels in image) stereo reconstruction, real-time computable nowadays, brings 3D reconstructed points even for less textured image regions, and has a lower percentage of wrong reconstructed points. This article presents novel algorithms for obstacle detection, using dense stereo reconstruction. They analyze, on the top view of the scene, the local density and vicinity of the 3D points, and determine the occupied areas which are then fragmented into obstacles with cuboidal shape: without concavities and only with 90° convexities. The orientation of the obstacles is determined in order to get a very good fitting of the cuboidal model to the obstacles in the scene ahead and, consequently, to minimize the free space which is encompassed by the cuboids. The main abilities of the approach are: generic obstacle detection, determination of obstacles' orientation, confident fitting of the cuboidal model.

**Index terms**—dense stereo vision, obstacle detection, obstacle modeling, urban traffic, automatic cruise control

## I. INTRODUCTION

Although the nowadays automatic cruise control (ACC) systems are a great achievement, they have limitations even in highway traffic where the scenario is quite simple. In the crowded city traffic solutions are not available yet. Currently the ACC systems detect the obstacles ahead by using either a radar or laser setup.

The stereovision is the most promising technology so far: it is based on a passive sensor, it accurately measures the 3D position of thousands of 3D points in the scene ahead and it allows algorithms working on 3D data and intensity images as well. The stereovision has the disadvantage of requesting high complexity algorithms and processing large amount of data.

The most known stereovision based approaches are: Inverse Perspective Mapping [1]; V-Disparity [2, 3], Warping [4]. They try to avoid the full 3D

reconstruction in order to reduce the processing time. Unfortunately these methods have intrinsic limitations due to the methods themselves, due to the abandon of the disambiguation of the obstacles close to each other, and due to the inaccurate modeling of the 3D obstacles.

The strongest point of our approach, based on the full 3D reconstruction of the scene [5], is the possibility to carry out geometrical reasoning for generic obstacle detection regardless the 2D appearance in images. The obstacles are confidently described by using the cuboidal model.

The algorithms presented in this paper work on 3D points provided by a dense stereo reconstruction engine. It provides 3D reconstructed points even for less textured image regions (figure 1.b).

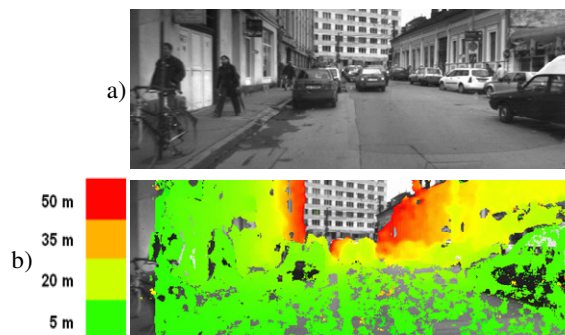


Figure 1. a) Left grayscale image, b) Pixels reconstructed as 3D points. The leftmost part is not reconstructed being not visible in the right image.

The target is to detect the obstacles in terms of 3D position, orientation and size, as boxes (cuboids) circumscribing the obstacles. The confident fitting of cuboids on obstacles is achieved in several steps. By analyzing the vicinity and the density of the 3D points, the occupied areas are localized (section IV). An occupied area

consists of one or more cuboidal obstacles that are close to each other. By applying criteria regarding the shape, the occupied areas may get fragmented into parts that obey the cuboidal shape (section V). The orientation of the obstacles (on the road surface) is of high importance as well (section VI).

The environment is considered as having a planar or a second-degree road surface, with obstacles above it. As guessed, an algorithm for obstacle/road separation is involved.

## II. DENSE STEREO RECONSTRUCTION

A calibration process estimates the camera's intrinsic parameters (which are related to its internal optical and geometrical characteristics) and extrinsic ones (which are related to the 3D position and orientation of the camera relative to a global world coordinate system) [6].

The most computational expensive task, the stereo correlation, is performed by hardware, a specialized PCI board (www.tyzx.com). A software module rectifies the left and the right images, the hardware computes the disparities, and then, the 3D points are computed, by software. Both rectification and 3D points computing use the cameras' parameters obtained through the calibration.

Textureless image regions, regions with repetitive texture, regions of the faraway scene and regions that are not visible in both left and right images cannot be reconstructed (figure 1).

The obstacle detection algorithms work on the 3D reconstructed pixels and on the left image space (figure 1).

A 3D coordinate system is defined. The origin of the coordinates system is the left camera. The Z axis is oriented ahead, representing the depth, the X-axis represents the lateral displacement and the Y-axis is used to express the elevation.

## III. OBSTACLE/ROAD SEPARATION

The obstacle/road separation algorithm ([5], [7]), is able to detect the surface of the road, by modeling it as a second-degree surface.

The only 3D points used by the obstacle detection algorithms are those situated above the road and below the height of the ego car (figures 2, and 3.b).

In figure 3.b one can observe some road points marked as above the road, due to the reconstruction errors.

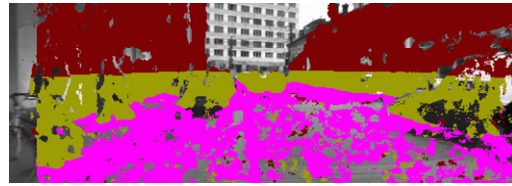


Figure 2. 3D points classified as: road, above road and too high points

## IV. LOCALIZATION OF OCCUPIED AREAS

Ideally, the obstacle detection algorithms should neither divide a real obstacle into smaller detected obstacles nor merge more real obstacles into one detected obstacle; two opposite requirements.

Previous experience has shown that these requirements are hard to be fulfilled due to reconstruction errors and limitations. The presented approach favors the merging of more real obstacles into one detected obstacle, named **occupied area**, and then applies fragmentation criteria.

It is supposed that the obstacles do not overlap each other on the vertical direction. In other words, on a top view (figure 3.c) the obstacles are disjoint. Consequently, in what follows, the Y coordinate (the elevation) of the 3D points will be ignored and **all the processing is done using only the X and Z coordinates (top view)**.

Due to the perspective effect of the camera, further obstacles appear smaller in our images, providing fewer pixels, and therefore, less, sparser 3D reconstructed points in the 3D space. On the other hand, the error of the depth reconstruction increases with the distance too, which, contributes to the 3D points sparseness as well.

The obstacle detection algorithms, processing the 3D points, would work easier if they would receive constant density of the points, regardless the distance. One idea would be to artificially introduce 3D points in between the reconstructed ones in order to keep a constant density, but this approach neither enriches the real 3D data nor reduces the depth error, and it slows down the processing.

To counteract the problem of the points' density, a schema to divide the Cartesian top view space into tiles with constant density is proposed (figure 3). The horizontal field of view of the camera is divided into polar slices with constant aperture, trying to keep a constant density on the X-axis. The

depth range is divided into intervals, the length of each interval being bigger and bigger as the distance grows, trying to keep a constant density on the Z-axis.

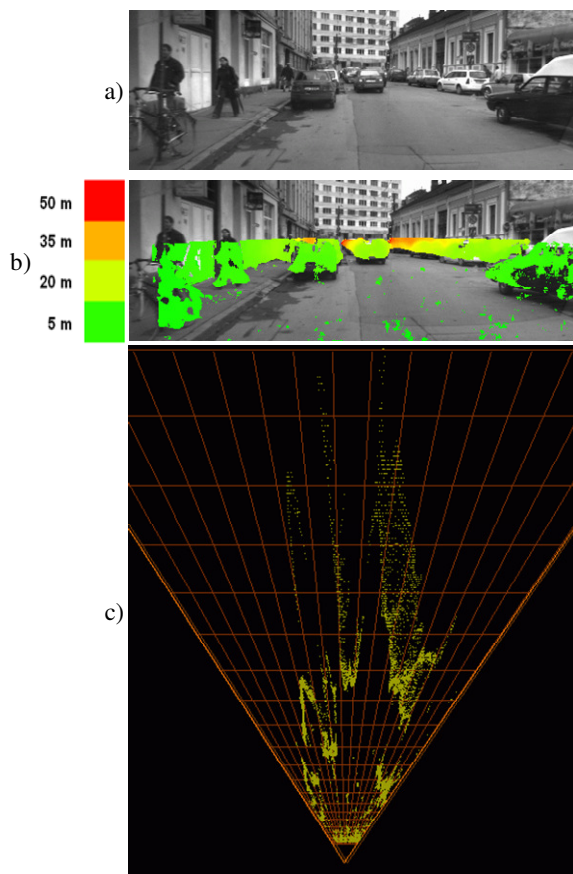


Figure 3. Division into tiles. a) Gray scale image, b) 3D points – perspective view, c) 3D points – top view; the tiles are here considerably larger for visibility purpose; wrong reconstructed points can be seen in random places; reconstruction error is visible as well.

A specially **compressed space** is created, as a matrix (figure 4.a). The cells in the compressed space correspond to the trapezoidal tiles of the Cartesian space. The compressed space is, in fact, a bi-dimensional histogram, each cell counting the number of 3D points found in the corresponding trapezoidal tile. For each 3D point, its cell in the compressed space is computed  $C(Row, Column)$ , and the cell value is incremented.

The column formula is:

$$Column = ImageColumn / c$$

where  $ImageColumn$  is the left image column of the 3D point and  $c$  is the number of adjacent image columns grouped into a polar slice as shown in figure 3.c ( $c = 6$ ).

The row of the compressed space, for a 3D point, has a formula obtained through a longer way:

The Cartesian interval corresponding to the first row of the compressed spaces is:

$$[Z_0 \dots Z_0 + IntervalLength(Z_0)] = [Z_0 \dots Z_1]$$

where  $Z_0 = Z_{min}$ , a minimum detection depth and  $IntervalLength(Z) = k * Z$  is the length of the interval beginning at a certain  $Z$  (further intervals are longer,  $k$  is empirically chosen). Thus

$$Z_0 + IntervalLength(Z_0) = Z_0 + k * Z_0 = Z_0 * (1 + k) = Z_1.$$

The Cartesian interval corresponding to the second row of the compressed spaces is:

$$[Z_1 \dots Z_1 + IntervalLength(Z_1)] = [Z_1 \dots Z_2]$$

where  $Z_1 = Z_0 + IntervalLength(Z_0)$ , in other words the second interval begins where the first one ends:

$$Z_1 = Z_0 * (1 + k) \text{ and}$$

$$Z_1 + IntervalLength(Z_1) = Z_1 + k * Z_1 = Z_1 * (1 + k) = Z_0 * (1 + k) * (1 + k) = Z_0 * (1 + k)^2 = Z_2.$$

The ending of the second interval is the beginning of the third one:

$$Z_2 = Z_0 * (1 + k)^2$$

The Cartesian interval corresponding to the  $n^{th}$  row of the compressed spaces is:

$$[Z_n \dots Z_n + IntervalLength(Z_n)], \text{ where}$$

$$Z_n = Z_0 * (1 + k)^n \text{ (provable by mathematical induction)}$$

For a certain 3D point, having depth  $Z$ , the  $i^{th}$  interval it belongs to is  $[Z_i \dots Z_i + IntervalLength(Z_i)] = [Z_i \dots Z_{i+1}]$ . From the formula  $Z = Z_0 * (1 + k)^i$ ,

the  $i$ , as an integer number, is

$$Row = i = \lceil \log_{1+k} \frac{Z}{Z_0} \rceil$$

As a bottom line, each 3D point is transformed, from the Cartesian space to the compressed space, into the cell  $C(Row, Column)$ .

There is one more remark for the building process of the compressed space: due to the fact that the used points are situated in a 3D band with a constant 3D height (along the Y-axis), as described in section III, that band, on the image space, has a decreasing height as the depth increases (see figure 3.b), having less 3D points respectively; to counteract this, the density of the cells is proportionally amplified with the depth.

The cells having no points represent free space. The cells having just a few points (less than 20 – empiric value) are also considered free, most probably that those points were wrong reconstructed. The other cells, having many points, reveal the existence of obstacles.

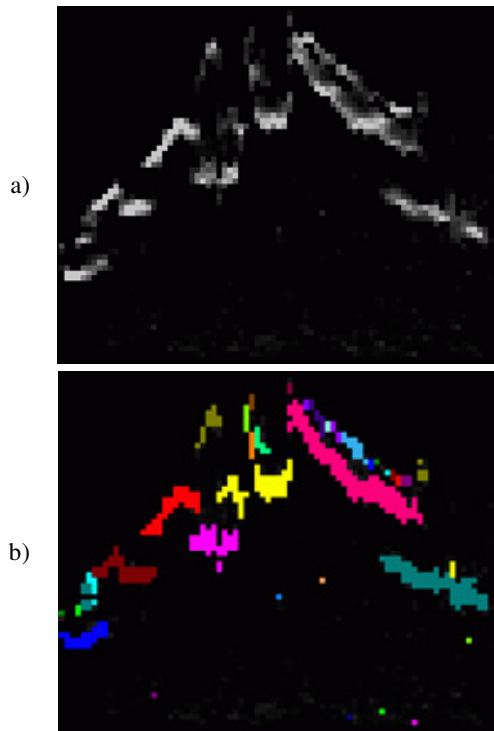


Figure 4. The compressed space (for scene in figure 3) – a bi-dimensional histogram counting 3D points (a). Groups of cells obtained by labeling (b)

The compressed space is used for occupied area detection. The detection is based on a labeling algorithm applied only on the cells having many points. The small groups are rejected. Most probably that the rejected groups come from 3D reconstruction errors generated by repetitive patterns or less textured image regions.

## V. FRAGMENTATION OF OCCUPIED AREAS INTO CUBOIDAL OBSTACLES

The occupied areas may contain more obstacles and by consequence may have miscellaneous shapes (figure 4.b). The obstacle tracking algorithm [8] as well as the ADAS applications need the individual cuboidal obstacles. Therefore the fragmentation of the occupied areas into the individual cuboidal obstacles is required. Two fragmentation criteria are proposed:

### A. CONCAVITY FREE SHAPES

The shape of a cuboidal obstacle has no concavities. The shape of the occupied area (figure 5.b) of a concave corner of a building (figure 5.a) is a relevant example (although it is particular, having

right angle) and must be fragmented into two parts, one for each wall.

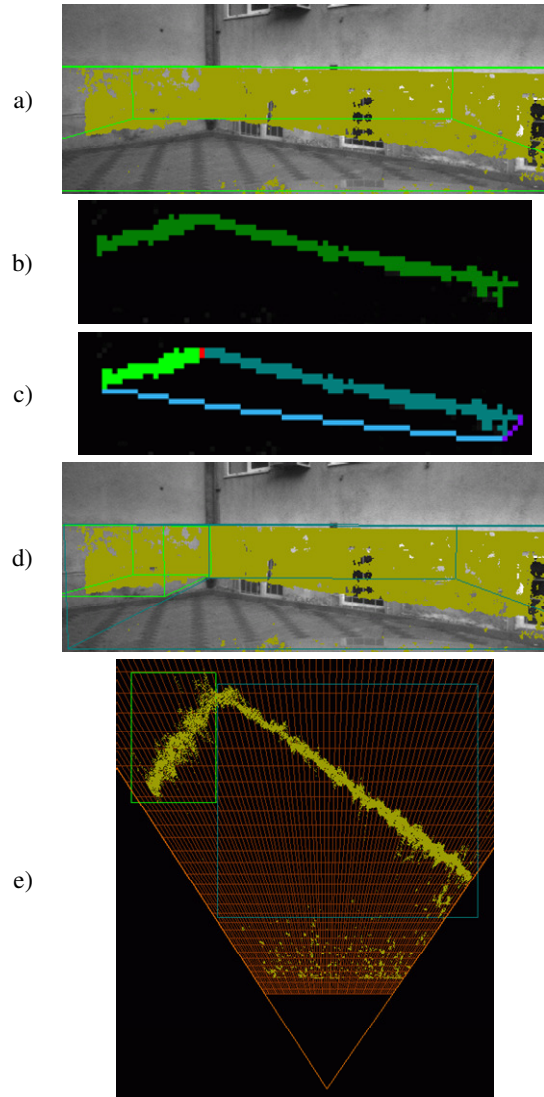


Figure 5. Fragmentation of a concave occupied area: a) free space included in the circumscribed cuboid, b) labeling in the compressed space, c) visible sides of the envelope of occupied area and the two subparts – compressed space, d) and e) the two circumscribed cuboids – perspective view and top view respectively

The solution is to determine the envelope of the cells of each occupied area, and then for each visible side (towards the camera) of the envelope (figure 5.c), the concavity between the side and the occupied cells is determined. If the concavity is large and deep, its deepest point gives the column (painted in red in figure 5.c) where the fragmentation will be done. The two subparts are subject to be fragmented again and again as long as they have significant concavities.



By reconsidering the 3D coordinates (including Y) of the points that have filled the cells of an occupied area, the limits of the circumscribing cuboid are determined (figure 5.d – perspective view, and figure 5.e – top view). At this point, the cuboids are parallel with coordinate axes and the occupied areas may be named “obstacles” even though the criterion *B* might fragment them again.

### B. NO EMPTY CORNERS OF THE CUBOIDS

A real obstacle is confidently detected whether the space between the sides of its oriented cuboid (its determination is presented in the section VI) and the visible shape of the cloud of its 3D points (on the top view) has a small area. In other words the shape must have only quasi-90° convexities. The visible free area in each corner of the cuboid is calculated in square meters, and, in order to be relevant (the relevance of the metric measurements decreases with the distance), it is transformed into the number of cells of the compressed top view space. When one of the corners has a free space of more than  $k=10$  cells (an empirically chosen threshold), there must be performed a fragmentation into two or more sub-obstacles that have confident cuboids.

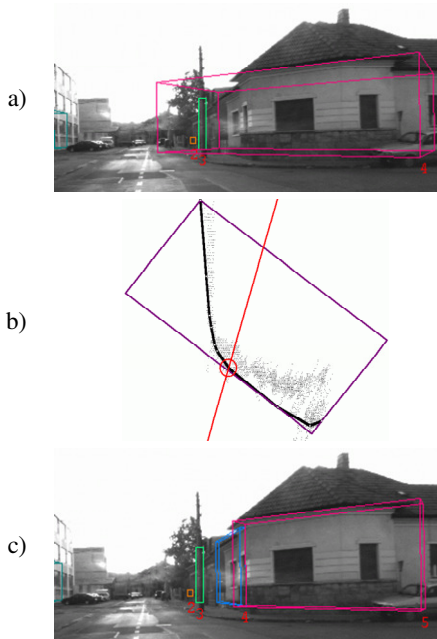


Figure 6. Fragmentation of an obstacle that doesn't have a cuboidal shape: a) initial cuboid, b) top view of the initial cuboid and the optical ray of the fragmentation, c) the two sub-obstacles obtained by fragmentation

An example is shown in figure 6.a and b, where no oriented cuboid would be confident. The

fragmentation is done by using the optical ray that passes through one of the vertices of the free corner. The chosen vertex is the one that is the most interior in the obstacle. The procedure is recursively applied on the newly obtained (sub)obstacles as needed. The fragmentation of the large cuboid in figure 6.a is shown in figure 6.c.

### VI. DETERMINING THE ORIENTATION OF OBSTACLES

In fig 7.a, it can be observed that, even though the real obstacle is oblique oriented, the cuboid encompasses some free space because the cuboid is parallel with the coordinate system. The shape of the cloud of 3D points is modeled by their envelope

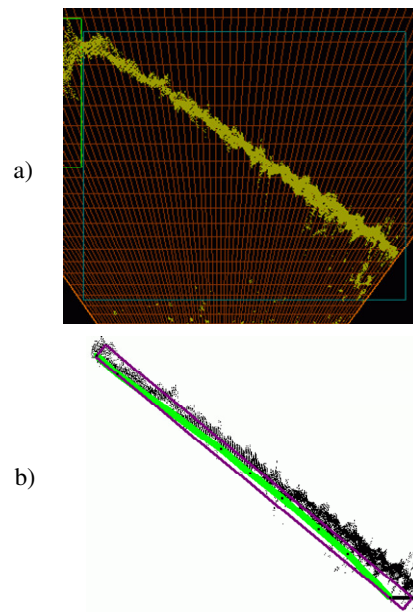


Figure 7. Obstacle orientation (top view): a) un-oriented box, b) the longest chain of visible envelope sides (green) and its surrounding rectangle

and an analysis of its visible sides (towards the camera) can determine the orientation of the obstacle. If the analysis cannot determine a preponderant orientation of these sides, the box remains parallel with the axes of the coordinates system; it is un-oriented.

In figure 7.b such an envelope is shown. The algorithm searches chains of consecutive sides having a low standard deviation of the slope of its components. If the length of the longest chain (the thick green one) is at least 70% from the length of the visible envelope, the obstacle orientation is the weighted average of the slope of the chain's sides. As weights we use here the lengths of the sides.

A rectangle with the found orientation is circumscribed on the visible sides and gives the base (top view – figure 7.b) of the oriented cuboid of the obstacle. The perspective result is shown in figure 8.b. Note: the envelope is not quite convex here, due to some approximations for speed up.

## VII. RESULTS

The horizontal field of view of the cameras is about  $70^\circ$  and the resolution is of  $512 \times 383$  allowing the stereo reconstruction to work fine up to 50 meters. The tests on thousand of images have shown that the approach is a robust one. The real-time performance is of 20 frames/second, for the whole application, when a P4, 2.6 GHz computer is used and the dense stereo correlation is done by hardware.

The figures used in this paper represent only the region of interest consisting of a centered horizontal band.

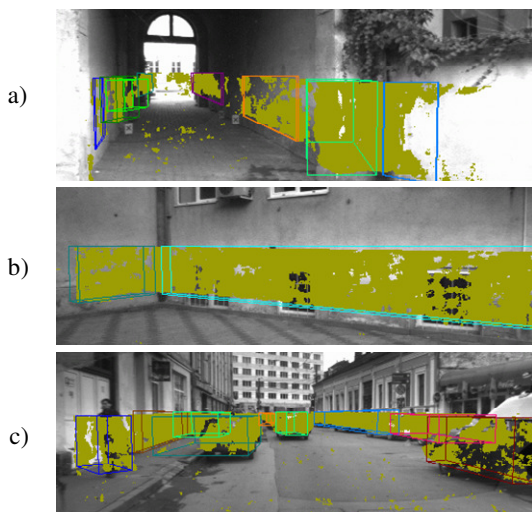


Figure 8. Results. a) the orientation of the second obstacle from the right couldn't be determined, b) an ideal case, c) due to merging of the pedestrian with the gate behind (left), the orientation is strange.

## VIII. CONCLUSIONS

The proposed obstacle detection approach is original due to the direct use of the 3D reconstructed space and due to the divers geometric reasoning carried out on this space.

The local density and vicinity of the 3D points is analyzed in a special compressed top view space. The compressed space keeps a constant density of the 3D points and neutralizes the error of the

reconstructed depth. On the compressed space, a labeling algorithm is applied, on the cells with high density of points, determining the occupied areas.

The occupied areas are fragmented into obstacles that are suitable for the cuboidal model. For this fragmentation, the visible shape (towards the camera) of the 3D points is analyzed and two criteria are used: the shape must have no concavities and the cuboid fitted on the shape must not contain significant free (drivable) space.

The orientation of the obstacles is determined, when possible.

Most of the processing is done on the compressed space which concentrates the useful information of the set of 3D points and leads to a fast computation.

Due to stereo reconstruction errors and limitations, the instantaneous results can differ in successive frames. Thus, a tracking algorithm is needed in order to filter the noise, to reject bad detection and to fill in the detection gaps.

## IX. REFERENCES

- [1] M. Bertozzi, A. Broggi, "Real-time lane and obstacle detection on the GOLD system", *Intelligent Vehicles Symposium 1996*, pp. 213-218.
- [2] R. Labayrade, D. Aubert, J.-P. Tarel, "V-disparity on Non Flat Roads", *Intelligent Vehicles Symposium 2002*, pp. 646-651.
- [3] Z. Hu, K. Uchimura, "U-V-Disparity - An efficient algorithm for Stereovision Based Scene Analysis", *Intelligent Vehicles Symposium 2005*, pp. 48-56.
- [4] C. Rossig, R. Hermann, M. Hotter, "Continuous Estimation of the road plane for mesuring obstacles", *Sixth Intern. Conf. on Image Processing and its Applications 1997*, vol. 1, pp. 394-398.
- [5] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, T. Graf "High accuracy stereo vision system for far distance obstacle detection", *IEEE Intelligent Vehicles Symposium, Parma (IV'04), Italy, June 14-17, 2004*, pp. 292-297.
- [7] Jean-Yves Bouguet (2003), "Camera Calibration Toolbox for Matlab", MRL – Intel Corp., [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/).
- [7] F. Oniga, S. Nedevschi, M-M. Meinecke, T-B. To (2007), "Road Surface and Obstacle Detection Based on Elevation Maps from Dense Stereo", *International IEEE Conference on Intelligent Transportation Systems 2007*, Seattle, USA, Sept. 30 – October 3, 2007, pp. 401-404.
- [8] R. Danescu, S. Nedevschi, M.M. Meinecke, T. Graf, "Stereovision Based Vehicle Tracking in Urban Traffic Environments", *International IEEE Conference on Intelligent Transportation Systems 2007*, Seattle, USA, Sept. 30 – October 3, 2007, pp. 271-276.

# G. Curriculum Vitae



**UNIVERSITATEA  
TEHNICĂ DIN CLUJ-  
NAPOCA**  
FACULTATEA DE  
AUTOMATICĂ ȘI  
CALCULATOARE

## INFORMAȚII PERSONALE

Numele și prenumele

Telefon / Fax

E-mail

**POCOL, CIPRIAN**

0264-401457

**Ciprian.Pocol@cs.utcluj.ro**

## PROFESIA / OCUPAȚIA ACTUALĂ

- Data 2002 – prezent (cercetare) și 2005 – prezent (didactic)
- Loc de muncă Facultatea de Automatică și Calculatoare; Catedra de Calculatoare
- Profesia Inginer, Specializarea Calculatoare
- Ocupația Preparator universitar
- Activitatea principală Activități didactice și de cercetare
- Conducător de doctorat Nu
- Domeniul –

## EDUCAȚIE ȘI STUDII DE CALIFICARE

- Anul 2002
- Numele și tipul organizației Universitatea Tehnică din Cluj-Napoca, Facultatea de Automatică și Calculatoare
- Titlul obținut Inginer
- Specializarea Calculatoare
- Anul 2003
- Numele și tipul organizației Universitatea Tehnică din Cluj-Napoca, Facultatea de Automatică și Calculatoare
- Titlul obținut Master
- Specializarea Noua generație de Calculatoare

## ACTIVITATE DIDACTICĂ

EXPERIENȚĂ PROFESIONALĂ	IN CADRUL DISCIPLINELOR	PROGRAMUL DE STUDII	ANUL
	Computer Programming	Calculatoare și Tehnologia Informatiei	I
	Object Oriented Programming	Calculatoare și Tehnologia Informatiei	II

## ACTIVITATE ȘTIINȚIFICĂ

### TEME DE CERCETARE

- Image Processing Techniques
- Computer Vision
- Vision-based sensors for automotive industry
- Environment modeling and understanding
- Intelligent Robots

### PROECȚE DE CERCETARE

- Cooperative Intersection Safety ([www.intersafe-2.eu](http://www.intersafe-2.eu)), Beneficiar: The European Commission, (2008-2011)
- Stereo System Modular Design (MODULAR), Beneficiar: Volkswagen AG, Germania, (2008-

	2009)
	- Dense STEREO Based Object Tracking and CLASSification for Pre-Crash applications (STEREOCLASS), Beneficiar: Volkswagen AG, Germania, (2008)
	- Dense STEREO Based Object Tracking and PEDESTrian Recognition for Pre-Crash Applications (STEREOPED), Beneficiar: Volkswagen AG, Germania, (2007-2008)
	- STEREO SENSor for City Automatic Cruise Control (STEREOSENS), Beneficiar: Volkswagen AG, Germania, (2007)
	- DENSE Stereo Based Object Detection, Tracking and Pedestrian Classification in Traffic Environments (DESPED), Beneficiar: Volkswagen AG, Germania, (2006 - 2007)
	- Dense Stereo Based Object Recognition System for Automatic Cruise Control in Urban Traffic Environments (DESBOR), Beneficiar: Volkswagen AG, Germania, (2005 - 2006)
	- Stereo Camera Based Object Recognition System for Vehicle Application - 3 Extension (SCABOR3Ex), Beneficiar: Volkswagen AG, Germania, (2004)
	- Stereo Camera Based Object Recognition System for Vehicle Application - 3 (SCABOR3), Beneficiar: Volkswagen AG, Germania, (2003 - 2004)
	- Stereo Camera Based Object Recognition System for Vehicle Application - 2 (SCABOR2), Beneficiar: Volkswagen AG, Germania, (2002 - 2003)
	- Stereo Camera Based Object Recognition System for Vehicle Application (SCABOR), Beneficiar: Volkswagen AG, Germania, (2001 - 2002)
EXPERIENȚĂ INTERNAȚIONALĂ	- 2002, Iulie: Volkswagen A.G. – Research Centre, Wolfsburg, Germania (4 săptămâni, în cadrul proiectului SCABOR1)
	- 2003, Aprilie-Mai: Volkswagen A.G. – Research Centre, Wolfsburg, Germania (6 săptămâni, în cadrul proiectului SCABOR2)
	- 2003, Octombrie-Decembrie: Volkswagen A.G. – Research Centre, Wolfsburg, Germany (11 săptămâni, în cadrul proiectului SCABOR3)
	- 2004, Noiembrie-Decembrie: Volkswagen A.G. – Research Centre, Wolfsburg, Germany (4 săptămâni, în cadrul proiectului SCABOR3E)
PARTICIPARE/ASISTARE LA CONFERINȚE	- IEEE International Conference on Intelligent Computer Communication and Processing ICCP, Cluj-Napoca, Romania, September 2014
	- IEEE International Conference on Intelligent Computer Communication and Processing ICCP, Cluj-Napoca, Romania, September 2013
	- IEEE International Conference on Intelligent Computer Communication and Processing ICCP, Cluj-Napoca, Romania, September 2012
	- IEEE International Conference on Intelligent Computer Communication and Processing ICCP, Cluj-Napoca, Romania, September 2010
	- IEEE International Conference on Intelligent Computer Communication and Processing ICCP, Cluj-Napoca, Romania, September 2009
	- IEEE International Conference on Intelligent Computer Communication and Processing ICCP, Cluj-Napoca, Romania, September 2008
	- International Workshop on Intelligent Transportation WIT, Hamburg, Germany, March 2008
	- IEEE International Conference on Intelligent Computer Communication and Processing ICCP, Cluj-Napoca, Romania, September 2007
	- International Conference on Intelligent Computer Communication and Processing ICCP, Cluj-Napoca, Romania, September 2006
	- International Conference on Intelligent Computer Communication and Processing ICCP, Cluj-Napoca, Romania, September 2005
	- International Conference on Automation, Quality and Testing, Robotics AQTR, Cluj-Napoca, Romania, May 2004