

Viziune Artificială

TEZĂ DE DOCTORAT

Metode Liniare de Estimare a Odometriei Vizuale

Student-doctorand:
Cătălin Golban

Conducător științific:
Prof. Dr. Ing. Sergiu Nedevschi

Comisia de evaluare a tezei de doctorat:

Președinte: Prof. Dr. Ing. **Prenume Nume** - Universitatea Tehnică din Cluj-Napoca;
Conducător științific: Prof. Dr. Ing. **Sergiu Nedevschi** - Universitatea Tehnică din Cluj-Napoca;

Referenți:

- Prof. Dr. Ing. **Prenume Nume** - Universitatea
.....;
- Prof. Dr. Ing. **Prenume Nume** - Universitatea
.....;
- Conf. Dr. Ing. **Prenume Nume** - Universitatea
.....

**- Cluj-Napoca -
2022**

MULȚUMIRI

Aș dori ca în primul rând să-i adresez sincere mulțumiri domnului profesor Sergiu Nedevschi pentru sfaturile valoroase, pentru îndrumarea excelentă pe care mi-a oferit-o pentru a ajunge în punctul de a susține această teză și pentru doza de motivație pe care mi-a transmis-o în mod constant și cu perseverență de-a lungul anilor. De asemenea îi mulțumesc vărului meu Petruț Cobârzan pentru numeroasele încurajări, pentru schimburile foarte bune de idei dintre care unele s-au concretizat în ceea ce numeam noi "schema secolului", și pentru ajutorul care a făcut posibilă generalizarea și extinderea metodelor propuse în prima fază și publicarea a câtorva articole științifice.

A fost o plăcere să lucrăm împreună și le mulțumesc de asemenea și celor care au contribuit la eforturile premergătoare acestei teze ca și coautori ai primelor abordări de odometrie vizuală: Cosmin Mitran, Ionuț Golban, Istvan Szakats. Doresc de asemenea să transmit sincere aprecieri pentru schimburile de idei înspre Ionel Giosan și Andrei Vătavu, colegi care m-au asigurat de valoarea și de utilitatea eforturilor pe care le depun în domeniu. Le mulțumesc de asemenea tuturor celor care au avut grija să îmi reamintească din când în când să îmi scriu teza de doctorat: prof. Vasile Pop, prof. Rodica Potolea, prof. Mihai Negru, complementând mesajul pe care îl primeam la fel de intens și din partea domnului profesor Nedevschi.

Nu în ultimul rând ţin să mulțumesc familiei: soției mele Cristina și celor doi băieți, Bogdan și Sebastian, pentru sacrificiile și compromisurile făcute mai ales când vine vorba de timp petrecut împreună și părinților mei pentru toată susținerea și, de ce să nu recunosc, pentru revizuirea riguroasă a scrisorii cu diacritice.

CUPRINS

ABREVIERI.....	9
INTRODUCERE.....	11
1. Contextul general.....	11
2. Definiția odometriei generale.....	12
3. Beneficiile odometriei vizuale.....	13
4. Obiective și contribuții.....	18
STADIUL ACTUAL AL CUNOAȘTERII.....	21
1. Evoluția sistemelor de odometrie vizuală de la bazele teoretice până la sisteme în timp real.....	22
1.1. Primele abordări și stabilirea aparatului matematic.....	22
1.2. Primele metode funcționale în timp real pe roboți mobili.....	25
1.3. Evoluția, interesul crescut din ultima decadă, și categorizarea metodelor.....	30
2. Principalele metode de robustificare a sistemelor de odometrie vizuală pe bază de trăsături.....	34
3. Tendințe noi de cercetare în domeniul odometriei vizuale.....	42
3.1. Sisteme multi-cameră.....	42
3.2. Metode pe bază de LiDAR și fuziune între video și LiDAR.....	45
3.3. Deep learning pentru odometrie vizuală.....	47
CONTRIBUȚIA PERSONALĂ.....	51
1. Modelarea matematică a problemei de estimare a odometriei vizuale.....	52
1.1. Considerații teoretice.....	52
1.1.1. Notații și considerente legate de geometria camerei.....	52
1.1.2. Notații și considerente de viziune stereo.....	58
1.1.3. Noțiuni geometrice folosite în metodele de odometrie vizuală.....	60

1.2. Modelarea matematică a metodelor stereo de odometrie vizuală pe bază de corespondențe de trăsături.....	66
1.2.1. Odometrie vizuală pe baza punctelor 3D reconstruite din stereo.....	68
1.2.2. Odometrie vizuală pe baza tensorului trifocal.....	69
1.2.3. Odometrie vizuală pe baza minimizării erorii de reproiecție în planul imagine.....	73
1.3. Metoda liniară de odometrie vizuală.....	74
1.4. Un model de eroare pentru metoda liniară de odometrie vizuală.....	81
1.5. Metode mono de odometrie vizuală pe baza de corespondențe temporale de trăsături.....	84
1.5.1. Estimarea rotației dintr-o singura camera.....	84
1.5.2. Odometrie vizuală din matricea fundamentală și din matricea esențială.....	86
2. Design-ul, implementarea și evaluarea algoritmilor de odometrie vizuală pe bază de trăsături.....	88
2.1. Principii și, metode generale de determinarea de trăsături și de asociere de trăsături.....	89
2.1.1. Determinarea de trăsături potrivite pentru a fi urmărite în imagini succesive.....	89
2.1.2. Asocierea trăsăturilor pe bază de descriptori.....	98
2.1.3. Asocierea trăsăturilor pe bază de metode diferențiale....	105
2.2. Asocierea și plauzibilizarea trăsăturilor pentru odometrie vizuală.....	109
2.2.1. Asociere circulară.....	109
2.2.2. Inițializarea metodelor de asociere pe baza de flux optic.....	111
2.2.3. Plauzibilizarea pe bază de perechi de trăsături.....	113

2.3. Metoda liniară versus metode non-liniare.....	114
2.3.1. Stereo.....	114
2.3.2. Mono.....	122
2.4. Robustificarea pe bază de descriptori invariánți la iluminare.....	127
2.5. Modelarea erorilor de odometrie vizuală.....	130
2.6. Rafinarea selecției de trăsături și re-estimarea odometriei vizuale.....	132
3. Utilizarea unor alte surse de informații de adâncime precum senzori LiDAR.....	138
3.1. Generalizarea sistemului stereo pentru LiDAR.....	138
3.2. Generalizarea modelului matematic stereo pentru LiDAR.....	139
3.3. Determinarea distanțelor trăsăturilor pe bază de LiDAR.....	141
3.4. Rezultate experimentale și concluzii.....	143
4. Reducerea erorilor acumulate la integrarea măsurătorilor de odometrie vizuală.....	148
4.1. Prezentare generală a sistemului.....	149
4.2. Determinarea de trăsături și de asociere de trăsături din LiDAR.....	151
4.3. Modelul matematic.....	153
4.4. Optimizare robustă.....	155
4.5. Rezultate și concluzii.....	156
5. Estimarea mișcării relative dintre obiecte folosind metode de odometrie vizuală.....	161
6. Concluzii finale.....	166
6.1. Concluzii generale.....	166
6.2. Originalitatea și contribuțiile inovative ale tezei.....	167
6.3. Posibile dezvoltări ulterioare.....	168

REFERINȚE.....	171
LISTA FIGURILOR.....	188
LISTA TABELELOR.....	193
ANEXE.....	195
Anexa 1 - Modelul matematic de estimare a covarianței unei măsurători rezultate din minimizarea unei funcții.....	196
Anexa 2 - Demonstrația matematica pe care se bazează detecția colturilor Harris.....	197
Anexa 3 - Invarianta ZNCC la condiții de iluminare.....	198
Anexa 4 - Interpolarea intensităților la nivel de subpixel.....	199
Anexa 5 - Demonstrația matematica a algoritmului Lukas-Kanade.....	200
Anexa 6 - Demonstrația matematica a liniarizării matricii de rotație.....	202
Anexa 8 - Corecția erorilor de odometrie pe bază de patru puncte coplanare.....	203
Anexa 9 - Corecția erorilor de odometrie pe bază de coplanaritate cu un plan dat.....	204
Anexa 10 - Transformări între imagini induse de un plan.....	205
Anexa 11 - Tensorul trifocal.....	207
LISTĂ DE PUBLICAȚII.....	211

ABREVIERI

Noțiunile și abrevierile folosite vor fi introduse parcursul lucrării. În particular modelul teoretic al lucrării va introduce un aparat matematic și o notație consistentă folosită ulterior în toată lucrarea.

INTRODUCERE

(*aprox. 2 pagini; obligatoriu pe pagină impară*)

1. Contextul general

Disponibilitatea camerelor foto digitale este în continuă creștere, acestea găsindu-se în numeroase dispozitive și accesorii care în general cresc calitatea vieții și oferă posibilități nebănuite în urmă cu două decenii. Aparate video digitale, telefoane mobile inteligente, ceasuri inteligente, pixuri cu funcție de filmare video, drone, camere de supraveghere, ochelari cu camere video incorporate sunt doar câteva exemple care puse în contextul mai larg fac parte din revoluția tehnologică și digitală la care asistăm. Dincolo de capabilitățile de fotografiere și filmare, odată ce este atașată unui dispozitiv de calcul, o cameră se transformă într-un dispozitiv de detectare și măsurare, iar viziunea artificială este disciplina care se ocupă cu interpretarea și înțelegerea prin metode algoritmice a informațiilor din imagini și video achiziționate cu ajutorul camerelor digitale de orice tip. Astfel viziunea artificială este disciplina prin care se obține o înțelegere și o reprezentare digitală abstractă și de nivel înalt, prin prelucrarea informațiilor precum imagini reprezentate ca și matrici de pixeli sau secvențe de imagini consecutive.

Camera este transformată într-un dispozitiv de măsurare datorită capabilităților de reconstrucție 3D care se pot obține folosind procesarea datelor de la camerele video. Capabilitățile de detecție și segmentare a entităților din scena care sunt în fața camerei combinate cu capabilitățile de măsurare a distanțelor până la obiecte fac din camerele video cei mai avansați senzori de percepție. Aceste metode sunt într-o continuă dezvoltare, în ultima decadă fiind puternic influențate de creșterea și ieftinirea puterii de calcul, precum și de evoluția spectaculoasă a metodelor pe baza de rețele neuronale convolutionale (deep learning) în contextul disponibilității unor volume de date din ce în ce mai mari pentru antrenarea acestora. Diversitatea metodelor de percepție pe bază de viziune artificială, precum și beneficiul acestora este ilustrat în [142] [143] pentru conducerea automată, în [144] [145] pentru automatizarea în agricultură, în [146][152] pentru percepția pe baza de camere video instalate

pe drone, în [147] pentru utilizarea viziunii artificiale în tehnologii de realitate argumentată, în [148] [149] pentru domeniul militar, în [150] pentru domeniul aerospatial, [151] pentru diverse aplicații industriale, și lista domeniilor poate să continue prin prisma faptului că percepția mediului înconjurător pe baza de viziunea artificială poate să atingă capabilități similare cu sistemul uman de viziune, deschizând posibilități de argumentare și complementare a capabilităților umane și de automatizare și înlocuire a factorului uman cu un sistem de calcul sau cu roboti.

2. Definiția odometriei generale

Odometria vizuală este capabilitatea unui sistem pe baza unei camere sau a unui sistem de camere video de a estima mișcarea camerei în timp, în spațiul tridimensional în care aceasta se află, pe baza imaginilor achiziționate succesiv la momente diferite de timp. Termenul de odometrie vizuală a fost pentru prima dată introdus în [117] și terminologia a devenit general acceptată în comunitățile de cercetare.

4

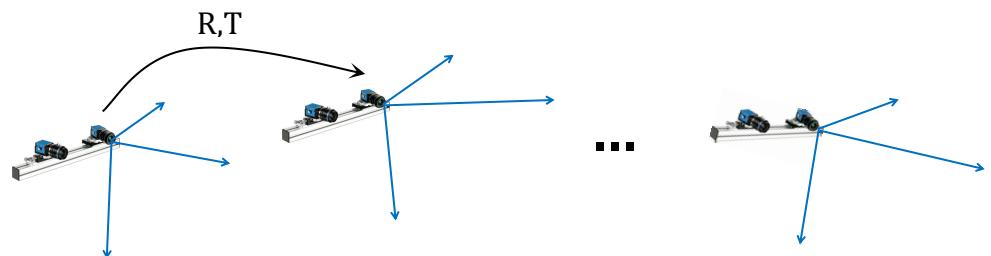


Figura 1 Problema odometriei vizuale

Din punct de vedere formal, odometria vizuală poate fi înțeleasă ca și o transformare între sisteme de coordonate. Imaginea achiziționată de o cameră, la un moment dat de timp, este proiecția spațiului tridimensional în planul imagine la acel moment de timp. Acest proces de formare a imaginii poate fi descris ca și o transformare de proiecție dintr-un sistem de coordonate tridimensional atașat camerei într-un sistem de coordonate bidimensional în planul imagine.

12

Vom numi în continuare sistemul camerei acest sistem tridimensional. Mișcarea camerei într-o poziție nouă implică mișcarea sistemului de coordonate tridimensional al camerei într-o poziție nouă, și proiecția elementelor tridimensionale din acest nou sistem de coordonate în planul imagine. Această mișcare poate fi modelată printr-un vector de translație și printr-o matrice de rotație, având în total șase grade de libertate. Odometria vizuală este estimarea transformării (rotație și translație) dintre sistemele de coordonate tridimensional atașate camerei în momente diferite de timp, pe baza imaginilor preluate de către cameră în aceste momente diferite de timp (Figura 1). Desigur că trebuie să existe entități comune între cele două imagini, deci să vizualizăm aceeași scenă din lumea reală din poziții și orientări diferite ale camerei. Se estimează astfel mișcarea camerei între momente diferite de timp, pe baza variației în aparență proiecției scenei, în imaginile produse de această cameră la momente diferite de timp. Aceste mișcări între imagini succesive preluate de o cameră video, aflate în mișcare pot fi desigur concatenate, construind o traiectorie a mișcării camerei în timp. În cazul camerelor monoculare, rotația camerei precum și două din trei grade de libertate ale translației pot fi estimate. Practic, putem estima direcția vectorului de translație, dar nu și magnitudinea exactă a acestuia. În cazul sistemelor stereo putem estima și magnitudinea translației, deci se pot estima toate cele șase grade de libertate ale mișcării, trei pentru rotație, respectiv trei pentru translație. Principalele metode explorate în cadrul acestei teze vor fi metode stereo, dar vom puncta și considerente teoretice, care vor ilustra diferențele dintre metodele monoculare și cele stereo, precum și metode de a combina aceste metode pentru a crește precizia estimării.

3. Beneficiile odometriei vizuale

Există câteva aplicații cheie în care odometria vizuală este importantă. În primul rand, mișcarea dintre poziții succesive ale camerei este folosită pentru a fuziona informația extrasă la momente diferite de timp. Fuziunea temporală a entităților detectate cu ajutorul camerelor duce la o mai bună precizie a poziționării acestora și la

eliminarea detecțiilor false prin plauzibilizare pe baza persistenței și consistenței acestora în timp. Această fuziune poate lua mai multe forme și prezintă particularitățile ei în funcție de entitățile fuzionate, iar odometria este esențială pentru a realiza această fuziune. De exemplu, în [153] odometria vizuală se obține atât pentru a construi o hartă de disparitate mai densă și mai precisă prin integrarea temporală a estimărilor de la momente de timp diferite, cât și pentru a integra în timp măsurăturile sub forma unui un grid, scopul final fiind de a estima spațiul liber din fața mașinii determinat de obiectele statice elevate. Alte exemple relevante din literatură sunt în [154] și [155] unde se construiesc hărți de înălțime pe bază de informație stereo și entități precum borduri sau în general obstacole tridimensionale care sunt plauzibilitate pe baza persistenței acestora în timp. Asocierea în timp se face pe baza informației de odometrie. O altă aplicație unde odometria vizuală poate fi folosită la modelarea dinamicii problemei este urmărirea în timp a obiectelor și detectarea obiectelor statice și a celor dinamice. În [156] este propus un model de compensare a mișcării camerei, poziționată pe o mașină în acest caz, pentru a discrimina între obiectele statice și dinamice din scenă. În [157] se propune un sistem care estimează mișcarea și traiectoriile mașinilor care se apropie de vehiculul curent. Mișcarea acestora este aproximată pe baza modelului Ackerman, iar pentru estimarea ei mișcarea camerei estimată prin metode de odometrie vizuală este compensată. Observăm, astfel, ca în comparație cu sistemele statice de camere (spre exemplu cele folosite în supraveghere video), în cazul unei camere sau a unui sistem de camere aflate în mișcare, gradul de complexitate care apare la dezvoltarea unor soluții de percepție tridimensională care să cuprindă și dinamica scenelor este unul ridicat datorită faptului că atât camera, cât și o parte din obiectele din scenă prezintă propria lor mișcare. Scopul acestei lucrări acela de a dezvolta o metodă de odometrie vizuală eficientă din punct de vedere al timpului de calcul și robustă. În plus folosirea unor metode generale de estimare a odometriei vizuale este explorată și pentru estimarea mișcării relative dintre cameră și obiecte dinamice din scenă.

Dacă în cazul camerelor stereo reconstrucția tridimensională se poate face pe baza potrivirilor la nivel de pixel între camera stângă și camera dreaptă din sistem, pentru camerele monoculare acest lucru nu este posibil. Cu toate acestea o reconstrucție tridimensională mai puțin precisă decât cea stereo este posibilă pe baza mișcării camerei în timp

(estimată prin odometrie vizuală) și potrivirea dintre trăsături în imagini preluate la momente diferite de timp. Referim [158] și [159] pentru mai multe detalii despre metodele de estimare a structurii din mișcare, unde odometria vizuală are un rol esențial. Constraințele de cost care apar atunci când introducem în producția de serie sisteme de percepție pot motiva folosirea de camere mono. În [157] este patentată o metodă monoculară de reconstrucție tridimensională densă a structurii din mișcare. De asemenea, pentru dispozitive care au deja camere încorporate, precum telefoanele inteligente, metode de estimare a structurii din mișcare pot fi utilizate pentru aplicații precum reconstrucție 3D în interiorul clădirilor, sau realitate augmentată [105]. Estimarea simultană a mișcării și a structurii scenei poate fi de asemenea o opțiune, mai ales atunci când constraințelor impuse de resursele de calcul disponibile nu sunt foarte limitative. Bundle adjustment [131] [130] este o metodă generică de estimare simultană a structurii 3D și a odometriei. Astfel de metode implică o optimizare ne-lineară a unei funcții de cost având ca și necunoscute, atât parametrii de mișcare cat și structura 3D a scenei (structura densă sau structura discretă [160]). Inițializarea parametrilor de mișcare pe baza rezultatului unei estimări independente a parametrilor de odometrie vizuală duce la o convergență mai rapidă dar și la evitarea minimelor locale în rezolvarea problemei de estimare a structurii.

O altă aplicație unde odometria este componentă esențială este localizarea și maparea (cartografierea) simultană (simultaneous localization and mapping). Aceasta cuprinde un set de metode pentru a localiza un agent mobil într-un sistem de coordonate global și în același timp pentru a construi sau îmbunătăți incremental o hartă a aceluiajū mediu pe baza observațiilor făcute de senzorii acestui agent mobil. În particular localizarea și cartografierea simultană se poate face pe bază de camere video folosite ca și senzori de măsurare și detecție [161] [162] [163] [164] [165], iar odometria vizuală este o componentă esențială în dezvoltarea unor astfel de sisteme. Intuitiv odometria vizuală estimată între imagini consecutive, se poate concatena, obținând traiectoria camerei relativ la un sistem de coordonate inițial (sau global). Măsurătorile tridimensionale obținute prin procesarea imaginilor de la cameră la fiecare moment în timp și reprezentate la diferite nivele de abstractizare (puncte 3D pentru spații închise [102] [103] [166] [137] [167] [168] [140] sau diverse obiecte sau trăsături în scene de trafic [169]) se pot transforma în acel sistem de coordonate

inițial (sau global), obținând astfel o hartă a mediului și traectoria de mișcare a camerei în acel mediu. Această traectorie este afectată de erorile de estimare acumulate succesiv, conducând la o deviere din ce în ce mai mare. Prin urmare se apelează la diverse metode de corecție, precum estimarea folosind mai mult de două imagini consecutive [137] [179], fuzionare cu senzori inerțiali [172] sau fuzionare cu senzori GPS [120]. În cadrul localizării și cartografierii simultane se folosește poziția cunoscută a obiectelor dintr-un sistem de coordonate global pentru a corecta devierea traectoriei obținute prin acumularea estimărilor dintre imagini consecutive. Estimarea simultană a structurii tridimensionale a scenei și a odometriei prin reprezentarea problemei sub forma unor structuri de tip graf [171] este intens computațională dar poate să producă rezultate mai bune, în funcție de compromisurile care se fac între puterea de calcul folosită și performanța algoritmicea necesară. O altă practică de corecție folosită mai ales pentru medii închise este aceea de a detecta dacă același loc este revizitat, iar estimarea deviației se face prin formularea unei probleme de minimizare similară cu formularea problemei de estimare a odometriei vizuale [137][139][140].

Sumarizând, din punct de vedere al importanței sale, odometria vizuală își găsește utilitatea în aplicații precum:

- ✓ Asocierea și integrarea în timp a informației extrase din imagini consecutive
- ✓ Compensarea mișcării camerei pentru aplicații de urmărire a obiectelor cu o camera dinamică
- ✓ Reconstrucția structurilor tridimensionale din scenă pe baza de poziții succesive ale aceleiași camere
- ✓ Determinarea traectoriei unei camere aflate în mișcare
- ✓ Localizarea și construcția de hărți pe bază de sisteme video folosite sub formă de dispozitive de detecție și de măsurare

Metodele de măsurare a odometriei pe bază de senzori video sunt în general atractive în numeroase aplicații unde există camere aflate în mișcare și unde există nevoie de a determina această mișcare cu precizie. Beneficiile aduse de odometria vizuală comparativ cu alti senzori sunt: costul relativ redus al camerelor video comparativ cu alți senzori precum cei inerțiali sau GPS diferențial, precizia ridicata care se

poate obține și care crește odată cu creșterea rezoluției camerelor, dispărând nevoia de sincronizare și calibrare între senzori diferiți (ex: senzori inerțiali și cameră) dacă se folosesc abordări strict vizuale de odometrie, rezolvă problemele pe care le are odometria pe bază de senzori care măsoară numărul de rotații al axului roților (*wheel speed sensors*) - acestea fiind eronate dacă apar alunecări sau derapaje pe teren noroios sau nisipos.

Printre cele mai mari provocări pe care le aduce odometria vizuală amintim: puterea de calcul necesară mai ales atunci când integrăm astfel de metode în sisteme embedded, acumularea erorilor în timp atunci când se dorește o reconstrucție a traiectoriei (provocare general valabilă și în cazul altor metode de odometrie). De asemenea există un set de provocări care sunt general valabile pentru sistemele video: condiții de iluminare scăzute, diferențe de iluminare, lipsa texturilor care poate să apară în anumite medii, condiții de ploaie, ceată sau ninsoare intense. În plus odometria vizuală poate fi afectată de calitatea calibrării camerelor (de exemplu calibrare relativă dintre cele două camere care compun un sistem stereo, sau precizia cu care modelul de distorsiune al lentilelor modelează distorsiunile reale introduse de către acestea [86]). Pentru că prin metode de odometrie vizuală se măsoară practic mișcarea camerei față de partea statică a scenei, aceasta va fi dificil de estimat în cazul scenelor acoperite predominant de obiecte aflate în mișcare.

Lucrarea de față este influențată de un set numeros de articole științifice care abordează problema odometriei vizuale pentru diverse aplicații și reflectă experiența acumulată prin cercetarea, dezvoltarea și evaluarea unor metode stereo de odometrie vizuală folosite în domeniul percepției în timp real a scenelor de trafic pentru conducerea automată. Există aplicații ale odometriei vizuale în domeniul roboților care se ghidă autonom în medii off-road [119] [120] [121] sau în medii în care nu avem metode de localizare precisă precum sisteme GPS [115], pentru inspectia halelor industriale [137], realitate virtuală [105], mașini autonome [83], obiecte zburătoare la altitudini mari [80], drone [146], telefoane inteligente [105], accesorii pentru realitate argumentată și pentru ajutorul persoanelor cu dizabilități sau a soldaților aflați în misiuni de training sau de salvare [60], industria marina și sub-marina [173], trenuri inteligente [174] [175], roboti agricoli [176] etc. Odometria vizuală și-a găsit aplicabilitatea și în domeniul medical. Metodele de localizare și cartografiere simultană sau

odometrie vizuală au fost aplicate cu succes pe fluxul de imagini primit de la un endoscop echipat cu o cameră mono [62] [64] sau stereo [61] [63]. De asemenea, capsulele endoscopice, dispozitive de dimensiunea unei pastile introduse în corpul pacientului pentru inspecția și diagnoza tubului digestiv, cuprind desigur și o cameră de luat vederi, făcând astfel odometria vizuală o alternativă plauzibilă a metodelor curente de localizare pe bază de frecvențe radio emise din exterior și triangulare [65] [66] [67].

Concluzionăm acest subcapitol introductiv și motivăm eforturile din această teză prin faptul că metodele de odometrie vizuală se dovedesc a fi benefice oriunde avem o cameră care se află în mișcare, pe baza căreia facem o percepție a mediului în care aceasta se mișcă.

4. Obiective și contribuții

Obiectivul principal al acestei lucrări este determinarea unor metode robuste și eficiente din punct de vedere computațional, de estimare a odometriei vizuale cu 6 grade de libertate, pe bază de camere stereo sau a unui sistem de senzori format dintr-o cameră mono și o hartă de adâncime asociată acesteia.

Cele mai importante contribuții sunt:

1. Dezvoltarea unui model analitic, bazat pe liniarizarea matricii de rotație și pe formularea funcției de cost pe baza erorilor de reproiecție în planul imagine și dezvoltarea pe baza modelului matematic a unui algoritm eficient de estimare a odometriei vizuale între imagini succesive.
2. Compararea și evaluarea metodei liniare de estimate a odometriei vizuale cu metode neliniare, inclusiv cu metode bazate pe tensorul trifocal.
3. Dezvoltarea unor metode de robustificare prin eliminarea trăsăturilor care nu respectă modelul de mișcare global (conservarea unghiului, potrivire circulară).

4. Stabilirea unui model analitic de propagare a erorilor de estimare a poziției trăsăturilor de intrare și folosirea acestuia pentru a estima calitatea odometriei vizuale estimate.
5. Un model de estimare a rotației pe baza corespondentelor de trăsături de puncte de la distanță și un algoritm de estimare a punctelor de la distanță pe baza variației în timp a unghiurilor dintre trăsături și centrul camerei.
6. Inițializarea metodelor de estimare a odometriei vizuale (inclusiv a celor liniare dar și a celor neliniare) cu rotația estimată din punctele de la distanță.
7. Evaluarea posibilității folosirii algoritmilor de odometrie vizuală (care reprezintă mișcarea relativă dintre cameră și scena statică) pentru estimarea transformației relative dintre cameră și obiectele din fața acesteia.
8. Propunerea unui model de paralelizare a algoritmului în cazul în care acesta se realizează pe o arhitectură paralelă de calcul.
9. Evaluarea metodei atât pe seturi de date UTCN, cat și pe seturi de date publice pentru a sublinia generalitatea metodelor propuse.
10. Generalizarea modelului stereo de estimare a odometriei vizuale cu 6 grade de libertate pentru o cameră mono combinată cu senzori LiDAR și generalizarea teoretică a modelului pentru orice metodă de estimare a adâncimii (ex: estimarea adâncimii scenei folosind rețele neuronale convenționale).
11. O metodă liniară de corecție a deviațiilor care apar prin concatenarea succesivă a măsurătorilor de odometrie vizuală pe bază de puncte 3D provenite de la un senzor LiDAR
12. Contribuții legate de estimarea de trăsături și de potriviri între trăsături folosite pentru odometria vizuală (interpretarea trăsăturilor de tip pe baza analizei componentelor principale, inițializarea metodelor diferențiale de urmărire a trăsăturilor pe baza odometriei vizuale, utilizarea de trăsături invariante la diferențe de iluminare pentru odometrie vizuală).

STADIUL ACTUAL AL CUNOAȘTERII
(obligatoriu pe pagină impară)

1. Evoluția sistemelor de odometrie vizuală de la bazele teoretice până la sisteme în timp real

1.1. Primele abordări și stabilirea aparatului matematic

Problemele estimării mișcării 3D a camerei pe baza mișcării aparente 2D din imagine au constituit subiectul unor activități constante de cercetare de-a lungul timpului. Un prim survey al metodelor de estimare a mișcării obiectelor rigide din scenă și a structurii scenei a apărut în anii '80 [200], acesta referind și articole de impact din anii '70. Metodele de estimare a mișcării sunt împărțite în două categorii: metode pe bază de trăsături și corespondențe între acestea și metode pe bază de flux optic. În particular, metoda pe bază de trăsături structurează problema estimării mișcării în patru pași: determinarea unui set de trăsături în cele două imagini, stabilirea de corespondențe între trăsături, formularea de constrângeri sub forma de ecuații matematice precum cea de obiect rigid aflat în mișcare, deplasările 2D ale trăsăturilor în imagine sunt folosite pentru a estima mișcarea 3D a scenei față de cameră, sau a unui obiect rigid față de cameră. Principalele provocări identificate sunt stabilirea corespondențelor între trăsături din planul imagine (ex: datorită ocluziunilor și a zgromotului din imagine), formularea robustă a problemei de minimizat și dezvoltarea de metode iterative stabile de optimizare numerică. Observăm astfel că încă din anii 70-80 încep să se pună bazele teoretice ale viziunii artificiale, unele aspecte teoretice fiind valabile chiar și în prezent. În [203] Ullman demonstrează o metodă prin care estimează, pe baza a trei proiecții ortografice diferite pentru patru puncte ne-coplanare, structura 3D a unei scene și transformatele (mișcarea) dintre cele trei poziții ale camerei ortografice.

În [201] și [202] este abordată problema estimării structurii 3D și a mișcării camerei pe baza unui model de proiecție perspectivă. Practic, cunoscând proiecțiile perspectiva 2D în două sau mai multe imagini, a unui set de puncte 3D aflat într-un sistem de coordonate global, sunt explorate soluții pentru problema determinării coordonatelor 3D ale fiecărui punct în sistemul de coordonate global (structura scenei), a poziției fiecărei camere în sistemul de coordonate global (mișcarea) precum și a parametrilor de proiecție (ex: distanțe

focale). Rezultă astfel o problemă de optimizare neliniară care este rezolvată folosind algoritmul Levenberg-Marquardt. Tot în [200], referind [201] și [202], este subliniat și explicitat un aspect foarte important, și anume că în cazul unei camere mono, magnitudinea translației nu poate fi determinată. Explicația oferită este ca un obiect mai mic, aflat în apropiere, va avea aceeași mișcare aparentă în imagine ca și un obiect având dimensiune dublă, situat la o distanță de două ori mai mare și care efectuează o translație de două ori mai mare comparativ cu obiectul mic și o rotație în jurul unei axe de rotație având aceeași orientare și magnitudine cu cea a obiectului mai mic. Tot din anii '80 provin și fundamentele teoretice de estimare a mișcării relativ la o suprafață planară [204] [205]. Este arătat faptul că pe baza a 8 parametri care depind de ecuația suprafeței, de proprietățile camerei (ex: distanțe focale) și de parametrii de mișcare, se poate prezice poziția unui punct în a doua imagine pe baza poziției acestuia în prima imagine [200]. Având corespondențele între puncte, cei 8 parametri se pot determina formând un sistem de ecuații liniare, urmând ca ulterior să se obțină parametrii de mișcare prin manipulare algebraică a relațiilor matematice implicate.

Problema includerii în mod explicit a unor constrângeri de rigiditate asupra scenei este abordată în [40], unde mișcarea relativ la scenă este dedusă din faptul că distanțele 3D rămân constante, independent de mișcarea scenei. Se exprimă faptul că punctele 3D se găsesc în direcția razei de proiecție, apoi se pune condiția de egalitate a distanțelor între perechi de puncte, fixând factorul de scalare, rezultând un sistem de ecuații neliniare care se rezolvă iterativ. Același grup de cercetare propune și folosirea unghiurilor dintre drepte ca și invariantă pe baza căruia se calculează deplasarea camerei [41], sau o combinație de puncte și drepte [42], argumentând faptul că localizarea liniilor în imagini se poate face cu o precizie mai mare [200].

Binecunoscuta matrice esențială a fost introdusă simultan atât în [214] cât și [219], ca fiind o matrice de 3x3 care include atât parametrii de rotație cât și cei de translație. Deși referința [219] este mai cunoscută și mai intens citată în literatura de odometrie vizuală, atât [214] cât și [219] prezintă ideea determinării matricii esențiale pe bază de 8 corespondențe, superioritatea expunerii din [214] constând în faptul că analizează în detaliu configurațiile de puncte pentru asa-zisele cazuri degenerate, în care problema nu poate fi rezolvată folosind această formulare matematică. Expresia matricii esențiale în [214],

după o deducție matematică rezultată prin manipulare strict algebrică a relațiilor matematice de proiecție în cele două imagini este prezentată în Figura 2.

Eforturile de estimare a matricii esențiale și de obținere a matricii de rotație și a direcției de translație și în general de înțelegere în detaliu a geometriei epipolare sunt continue în mod susținut în anii '90 prin lucrări cu influență precum cele ale lui Harley [18] [19] [31] sau cele avându-l pe Fagueras ca și autor [21] [22]. Metoda de calcul a parametrilor de mișcare din matricea esențială este reluată și prezentată explicit în [20]. De asemenea, în aceeași perioadă constrângerile asupra proiecțiilor în două imagini se extind la trei sau mai multe imagini, apărând noțiuni precum tensorul trifocal [23] sau quadrifocal [24] precum și noțiuni legate de relațiile existente între puncte și linii proiectate în aceste imagini [25]. Se pun astfel în mod treptat baze teoretice solide ale vizionii artificiale, elementele de geometrie proiectivă fiind descrise în articole și rapoarte științifice cu caracter de tutorial [26][27][28][29], și culminând cu una dintre cele mai importante cărți din domeniu [2].

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} E \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$E \triangleq \begin{bmatrix} \Delta z \cdot r_4 - \Delta y \cdot r_7 & \Delta z \cdot r_5 - \Delta y \cdot r_8 & \Delta z \cdot r_6 - \Delta y \cdot r_9 \\ \Delta x \cdot r_7 - \Delta z \cdot r_1 & \Delta x \cdot r_8 - \Delta z \cdot r_2 & \Delta x \cdot r_9 - \Delta z \cdot r_3 \\ \Delta y \cdot r_1 - \Delta x \cdot r_4 & \Delta y \cdot r_2 - \Delta x \cdot r_5 & \Delta y \cdot r_3 - \Delta x \cdot r_6 \end{bmatrix}$$

Figura 2 Matricea esențială. Preluare din [214]

Un alt articol de review apărut o decadă mai târziu este [32]. Sunt evaluate metode de estimare a mișcării pornind de la viteza instantanee a unui punct în planul imagine exprimată în funcție de viteza de rotație și cea de translație relativă față de cameră. Principalele metriki de evaluare folosite sunt stabilitatea estimării mișcării în condițiile în care valorile vectorilor de flux optic sunt afectate de zgromot. Sunt comparate astfel cinci metode, dintre care le punctăm pe cele mai influente. În [33] [34] prin manipulare algebrică se elimină componenta Z a originii vectorilor de flux optic, se estimează rotația în funcție de translație folosind metoda celor mai mici pătrate, apoi se estimează direcția translației folosind metode de optimizare non-

lineară. În [35] este propusă o metodă care din triplete de puncte, pe baza ecuației de mai sus, se obține o constrângere asupra parametrilor de rotație pe baza căreia se determină parametrii de rotație prin optimizare non-lineară. Metoda propusă în [36] se bazează pe observația că unghiurile dintre două puncte 3D și centrul de proiecție se pot calcula strict pe baza proiecțiilor acestor puncte.

De asemenea variația acestor unghiuri între două imagini ale aceleiași scene nu depinde de rotația dintre cele două imagini, ci doar de translație dintre acestea, făcând posibilă estimarea translației pe baza lor, independent de valoarea rotației.

1.2. Primele metode funcționale în timp real pe roboți mobili

Primele implementări real-time de metode de odometrie vizuală apar datorită necesității de poziționare precisă în medii în care metodele clasice de localizare nu funcționează. În [114] este introdusă o metodă stereo de odometrie vizuală construită cu scopul de a fi folosită pe roboți NASA de explorare a planetei Marte. Rezultate remarcabile care indică utilitatea acestor metode sunt ilustrate în [115] și [116]. În linii mari, principaliii pași sunt detecția de trăsături într-una dintre imagini, estimarea pe baza unor metode de minimizare a corelației dintre corespondențele spațiale și temporale (tracking de trăsături), reconstrucție 3D, și estimarea mișcării. Estimarea mișcării este formulată ca și o problemă de maximizare a probabilității (maximum likelihood) de aliniere a punctelor 3D corespondente, iar modelarea erorii de reconstrucție 3D este descrisă de către aceiași autori în [122] [123] (Figura 3). Tot din [114] se pot desprinde unele idei general valabile în cazul algoritmilor de odometrie vizuală. Spre exemplu se subliniază faptul că eroarea de localizare este puternic influențată de acumularea în timp a erorilor de rotație, și faptul că acest efect poate fi compensat prin introducerea unui senzor de orientare absolută (Figura 4). De asemenea, desprindem din [114] niște idei de robustificare precum: limitarea spațiului de căutare a corespondențelor în timp printr-o precalculare a unghiurilor de pitch și yaw pe baza trăsăturilor de la distanță, urmărirea de trăsături pentru mai multe imagini consecutive care are ca efect reducerea erorii acumulate în timp cu 27%,

eliminarea outlierilor pe baza constrângerii de rigiditate: distanțele 3D între perechi de trăsături trebuie să rămână la fel înainte și după mișcarea robotului. În general, prin folosirea acestor metode de robustificare, chiar dacă modelul de formulare a problemei de minimizare admite îmbunătățiri, se obțin rezultate promițătoare având erori de 1-2% din distanța parcursă. Chiar dacă formularea sub forma de maximizare a probabilității propusă în [114] este superioară metodelor de aliniere a unor seturi de puncte 3D precum cele din [124], [125] metodele care formulează funcția de cost în planul imagine se dovedesc să fie superioare.

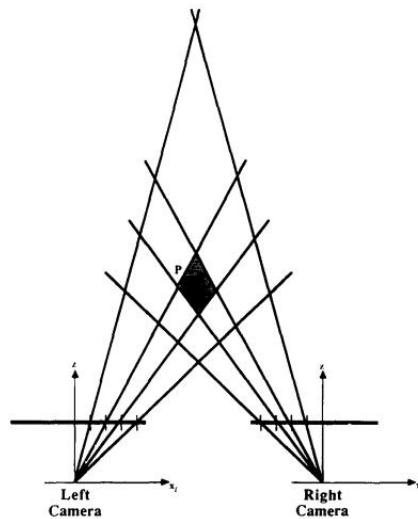


Figura 3 Geometria unei camere stereo, indicând incertitudinea de reconstrucție stereo [122]

O altă lucrare cu mare influență în domeniu este [117]. Nister, autorul căruia activități de cercetare au ca obiectiv vehiculele care navighează autonom în medii off-road, este de altfel cel care a introdus noțiunea general acceptată de odometrie vizuală (visual odometry). Aceasta propune în [117] atât un algoritm bazat pe o cameră mono cât și o metodă stereo de odometrie vizuală, ambii având ca și input un set de trăsături de tip colț [43] urmărite pentru mai multe imagini consecutive. Mișcarea relativă dintre primele trei imagini se estimează folosind algoritmul celor 5 puncte (5 points algorithm) [126] introdus de către același autor, care practic este o metodă matematică de estimare a matricii esențiale pe bază a 5 trăsături corespondente. Având mișcarea între aceste imagini se poate face o reconstrucție 3D a scenei, păstrând

un factor de scalare ales arbitrar dar consistent în timp. Având un set de puncte 3D în sistemul de coordonate al unui frame anterior, și proiecțiile acestor puncte într-un frame curent, problema de estimare a odometriei vizuale se rezumă la binecunoscuta problemă PnP, având P3P [127] [128] ca și caz particular: dat fiind un set de puncte 3D într-un sistem de coordonate, și proiecțiile perspectivă ale acestora într-o cameră poziționată în acest sistem de coordonate, se poate determina poziția camerei în acel sistem de coordonate. În cazul unei camere în mișcare, se determină practic odometria vizuală ca fiind poziția și orientarea camerei la momentul următor, în sistemul de coordonate al camerei la un moment anterior. Varianta stereo a algoritmului din [117] se bazează pe același algoritm, reconstrucția 3D a punctelor făcându-se pe bază de stereoviziune, iar corespondențele false se elimină folosind RANSAC (RANdom SAMple Consensus) [44].

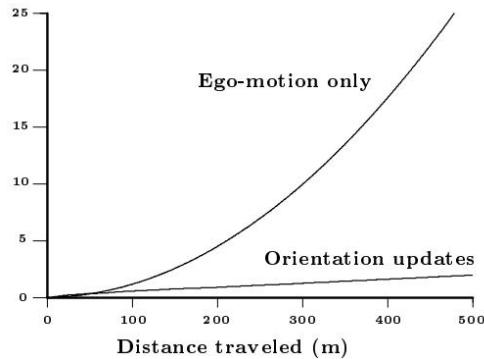


Figura 4 Creșterea super-liniară a erorilor de poziționare datorită acumulării erorilor de rotație și posibilitatea reducerii efectului cu un senzor de orientare absolută [114]



Figura 5 Exemplu de erori cauzate de alunecarea în gol a roților (stânga) pentru robotul mobil din dreapta [120]

În [120] odometria vizuală este fuzionată cu un sistem GPS standard pentru a obține o localizare precisă în medii off-road, motivat de faptul că datorită alunecării și derapării roțiilor robotului apăreau erori mari de localizare în cazul în care semnalul GPS este pierdut (Figura 5). Ipotezele de mișcare se estimează în 3D folosind grupuri de 3 puncte [124], într-un framework de tip RANSAC [44]. Verificarea punctelor care verifică aceste ipoteze de mișcare se realizează în spațiul imagine iar pe baza setului maximal de puncte se formulează o problemă liniară de optimizare a unei funcții de cost care reprezintă eroarea de reproiecție. Fuziunea cu GPS se face pe baza unui filtru Kalman extins.

O îmbunătățire a metodei din [120] este propusă de către aceiași autori în [119]. Aceasta constă în folosirea estimărilor dintre imagini consecutive într-un framework de estimare a structurii din mișcare (structure from motion, bundle adjustment) pentru a rafina mișările relative dintre pozițiile camerei folosind mai multe imagini consecutive. De asemenea în [118], aceiași autori propun fuzionarea dintre senzori inerțiali și odometrie vizuală folosind un filtru Kalman extins, argumentând faptul că există scenarii în care o estimare vizuală a odometriei este greu de realizat datorită lipsei structurilor și texturilor din scenă, și în aceste situații folosirea datelor inerțiale duce la o robustețe mai mare a sistemului.

Bundle adjustment [131][130] este o metodă generică de estimare a structurii 3D din mișcare. Chiar dacă astfel de metode sunt intens computaționale și nu fac obiectul direct al acestei teze, acestea merită prezentate pe scurt deoarece ele pot fi folosite ca și metodă de post-procesare, inițializată pe baza estimărilor incrementale între imagini succesive. Prin bundle adjustment se estimează pozițiile 3D ale unui set de puncte situate într-un sistem de coordonate comun, precum și matricile de proiecție pentru un set de N camere, pe baza proiecțiilor acelor puncte 3D în cele N camere. Deoarece cele N matrici de proiecție cuprind și transformarea din sistemul comun de coordonate în sistemul camerei corespunzătoare, odometria dintre poziții succesive ale camerei este și ea implicit determinată prin această metodă. În [113] este prezentată în detaliu o metodă stereo de odometrie vizuală pe bază de bundle adjustment, care extinde framework-ul general din [132], argumentând faptul ca introducerea mai multor imagini consecutive în procesul de minimizare duce la o scădere semnificativă a erorilor care apar la localizarea prin concatenarea mișărilor succesive.

În [8] și [9] am dezvoltat o metodă de odometrie vizuală pentru un sistem stereo video de percepție folosit pentru conducerea automată (Figura 6). Matricea de rotație este estimată folosind matricea esențială [20] estimate pe baza unui algoritm care folosește 8 puncte [19], pe baza trăsăturilor din [45]. Ulterior translație este estimată folosind reconstrucția 3D și rotația estimată anterior pe baza matricii esențiale.

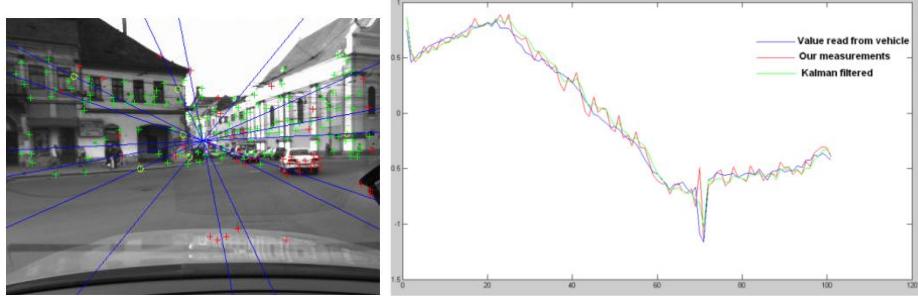


Figura 6 Trăsăturile (verde) pe baza cărora se estimează matricea esențială (liniile epipolare cu albastru). Se observă obiectele în mișcare precum și trăsăturile eronate segmentate cu roșu. Estimarea unghiului de yaw într-o succesiune de curbe prima la stânga și a doua la dreapta (albastru - valoarea citită de pe CAN, rosu - estimarea noastră, verde- valoarea filtrată cu filtru Kalman) [9]



Figura 7 Exemplu de mediu în care este evaluată metoda propusă în [121]

În [121] este prezentată o metodă stereo de odometrie vizuală care folosește trăsături de tip colț în imagini, determină corespondențele dintre acestea pe baza sumei diferențelor absolute a intensităților din jurul acestor trăsături, și estimează odometria minimizând o funcție de cost care exprimă eroarea de proiecție (la fel ca și în [120]). Elementul de noutate adus în [121] constă în faptul că se determină un set maximal de trăsături care respectă constrângerea de rigiditate: distanțele relative între perechi de trăsături se păstrează consistente de la un frame la următorul, rezultând o metodă aplicabilă

atât pe platforme mobile cu roți, utilizate în medii în care pot derapa sau aluneca (Figura 7), cât și pe roboți cu patru picioare care navighează în astfel de medii [129] (Figura 8). De asemenea, unul dintre avantajele subliniate în plus față de eliminarea trăsăturilor eronate, este acela că această constrângere de rigiditate impusă unui grup complet de trăsături elimină riscurile de a apărea erori datorate urmăririi de trăsături localizate pe umbrele roboților.

Modelul matematic propus în lucrarea curentă este dedus pe baza modelului neliniar folosit în exprimarea erorii de proiecție [120], [121], ulterior obținându-se o formă analitică liniară de exprimare a constrângerilor asupra odometriei vizuale pe baza corespondențelor 3D de trăsături între perechi de imagini stereo.

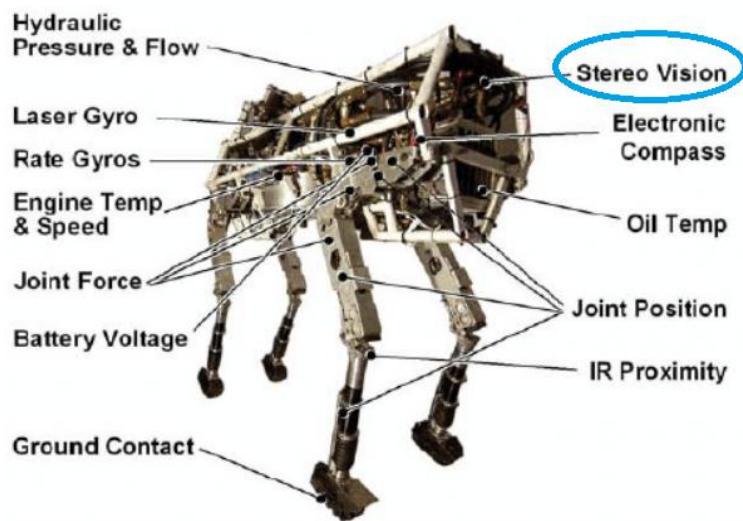


Figura 8 Robot from [129] using stereo visual odometry method from [121]

1.3. Evoluția, interesul crescut din ultima decadă, și categorizarea metodelor

Interesul și beneficiul crescut adăugat în sistemele de senzori video este însoțit și de apariția unor articole de tip tutorial [177] [178], a unor articole care sumarizează varietatea de metode existente [180] [181] [182], precum și a unui număr semnificativ de seturi de date

publice [179], subliniind varietatea de alternative disponibile atunci când facem designul unor astfel de sisteme, în special când vine vorba de sisteme de conducere automată [142] [143].

Dincolo de domeniul de aplicabilitate, sau de tipul de cameră folosit, putem clasifica metodele de odometrie vizuală în funcție de modul în care formulează și rezolvă problema estimării parametrilor de mișcare. Distingem astfel trei mari categorii: metode pe bază de trăsături, metode directe și metode pe bază de învățare, primele două categorii fiind incluse în categoria de metode geometrice iar cea de-a doua în categoria de metode non geometrice în [180].

Cele mai multe metode pe bază de trăsături se bazează pe trăsături de tip punct, având o poziție bine definită în imagine și un descriptor în general bazat pe valorile intensităților aflate într-o regiune din jurul acestor trăsături. În lucrarea de față vom folosi trăsături de tip punct chiar dacă același model matematic se poate extinde înspre trăsături de tip segment sau de tip dreaptă (segmente, linii) precum cele folosite în [95] sau o combinație de puncte și linii [96] [183]. În general folosirea trăsăturilor de tip linie sau segment poate aduce valoare în scenarii de interior unde texturile din scenă sunt mai uniforme, sau în scenarii structurare de oraș unde structurile clădirilor prezintă astfel de caracteristici, însă utilizarea lor este motivată doar atunci când nu există suficiente trăsături de tip punct în aceste scenarii.



Figura 9 Camera de parcare pentru care se pretează metodele directe [98]

Metodele directe [38] [97] nu implică existența unor trăsături pe baza cărora să se estimeze mișcarea camerei, și se bazează pe o funcție de cost compusă prin suprapunerea intensităților dintr-o imagine destinație și imaginea sursă remapată peste imaginea destinație pe baza modelului de mișcare folosit. Astfel de metode directe se pretează a fi folosite mai ales în medii în care avem texturi relativ uniforme, cum ar fi

imaginile de la o cameră de parcare, care de obicei sunt orientate spre asfalt [98] (Figura 9).

Metodele directe se combină cu metode pe bază de trăsături în [99] unde se calculează trăsături de tip punct pentru anumite imagini cheie având distanța între ele de mai mult de 12% din distanța medie a scenei. Mișcarea dintre imagini consecutive se calculează folosind metode directe, doar pe baza intensităților din imagine aflate în vecinătatea trăsăturilor de tip punct, argumentând că astfel se obține o viteză mai mare de execuție datorită faptului că nu trebuie calculate trăsăturile la fiecare frame și o precizie mai bună a mișcării datorită posibilității de interpolare la nivel de subpixel atunci când se formulează funcția de cost.

O atenție specială merită acordată abordării algoritmice stereo din [37] deoarece aceasta modifică modul standard de abordare a problemei determinării corespondențelor dintre trăsături. Astfel dacă de obicei corespondențele temporale se calculează independent pentru camera stângă și pentru camera dreaptă, în [37] autorii propun un algoritm de flux optic, care estimează simultan fluxul optic pe camera stânga, respectiv pe camera dreaptă, folosind modelul teoretic din [38], reducând astfel numărul de parametri de estimat de la patru (deplasarea trăsăturii pe direcție orizontală și verticală în imaginea stângă, respectiv în imaginea dreaptă) la trei. Acest lucru este posibil deoarece pentru trăsăturile corespondente dintre imaginea stângă și imaginea dreaptă, deplasarea pe direcția verticală în planul imagine trebuie să fie aceeași, dat fiind că cele două camere sunt rectificate. De asemenea, tot în [37] se introduce ideea că parametrii de rotație pot fi determinați pe baza trăsăturilor de la distanță și se formulează problema estimării directe a parametrilor de rotație tot pe baza aparatului matematic din [38], fără a mai estima deplasamentele trăsăturilor din planul imagine.

În funcție de camerele și de tipul lentilelor acestora putem categoriza metodele de odometrie vizuală în metode pe bază de camere mono, metode pe bază de camere stereo, și metode pe bază de camere omnidirecționale. Considerăm camerele omnidirecționale ca fiind diferite față de camerele mono datorită modelului geometric de proiecție diferit al acestora [30]. În cazul metodelor de odometrie pe bază de camere mono avem provocarea de a estima magnitudinea vectorului de translație, provocare care nu există în sistemele stereo.

Abordările din literatură folosite pentru a rezolva această provocare se bazează pe fuziunea cu senzori inertiali [172][215], cu senzori GPS [120] sau cu senzori LiDAR [70][71], sau pe cunoștințe preliminare legate de structura scenei sau de poziția camerei în scenă [216]. Lucrarea curentă își propune să abordeze problema odometriei vizuale pa bază de camere stereo și să generalizeze modelul matematic construit pentru stereo pentru a fi utilizat pentru un ansamblu compus dintr-o cameră mono și un senzor LiDAR.

În contextul fuzionării cu alți senzori, merită punctat faptul că acumularea de erori prin integrarea în timp a măsurătorilor de odometrie vizuală este redusă prin utilizarea de senzori GPS [120], prin fuziunea cu senzori de orientare absolută [221] sau prin corecția traectoriei folosind potrivirea dintre elementele detectate cu camera și elemente stocate anterior într-o hartă dacă privim în ansamblu problema localizării și cartografierii simultane [169].

2. Principalele metode de robustificare a sistemelor de odometrie vizuală pe bază de trăsături

Metoda stereo având cea mai bună performanță pe cel mai renumit set public de date de odometrie vizuală (KITTI [72] [73]) este descrisă în [136] și ulterior în [137]. Analizând în detaliu implementarea din [83] se observă faptul că performanța metodei crește dacă trăsăturile din centrul imaginii primesc o pondere mai mare decât cele de la marginile acesteia datorită imperfecțiunilor de modelare a proprietăților fizice ale lentilelor. De asemenea se observă că diferența între coordonatele verticale ale trăsăturilor corespunzătoare dintre imaginile stângă și dreapta a sistemului stereo, precum și pragul folosit pentru a valida trăsăturile care respectă modelul de mișcare pe baza erorilor de reproiecție sunt mai mari de 1-2 pixeli, în condițiile în care o calibrare perfectă ar implica praguri la nivel de sub-pixel. Îmbunătățirile aduse în [136] sunt în primul rând la nivel de selecție și urmărire de tastaturi: trăsăturile din [83] sunt urmărite pe mai multe imagini consecutive, păstrând descriptorul din prima imagine în care apare ca și descriptor de referință. Deși aceasta duce la o durată de viață mai scurtă pentru trăsături, comparativ cu cazul în care am folosit tot timpul descriptorul de la imaginea anterioară ca și referință pentru a captura schimbarea de aparență în timp, se evită în acest mod apariția de potriviri false și o deviere a trăsăturilor care ar cauza erori de odometrie. Având aceste trăsături se folosește algoritmul celor 5 puncte [126] pentru a estima rotația atât între imagini consecutive cât și între imagini aflate la distanță mai mare unele de celelalte, iar rotația finală se interpolează pentru o mai bună precizie finală. În [137] se propun câteva îmbunătățiri pentru metoda din [136]. În primul rând se rafinează modul în care se consideră combinația dintre trăsăturile vechi și trăsăturile noi apărute, ambele fiind importante pentru robustețea și precizia estimării finale. De asemenea numărul de imagini pentru care se calculează rotațiile relative este crescut, apoi se introduce și un mecanism simplu de detecție dacă se traversează o zonă prin care robotul a mai fost anterior. Acest algoritm de re-vizitare se bazează pe o căutare spațială a unor imagini cheie, apoi transformata relativă dintre poziția curentă și poziția imaginilor cheie aflate în proximitate se estimează pe baza acelorași trăsături folosite

pentru odometria vizuală și folosind același algoritm de minimizare a erorii de proiecție. Rezultă astfel o metodă de estimare a trecerii prin același loc mai simplă decât abordării standard [140] pe bază de trăsături având descriptori asociați [138] și dicționare care stochează acești descriptori [139].

În [60] este introdusă o metodă de odometrie vizuală stereo pentru o cameră pozitionată pe o cască (Figura 10), cu scopul de a fi folosită pentru nevăzători, sporind percepția în medii dinamice și complexe. Abordarea din [59] de tracking 3D de trăsături din domeniul automotive, este extinsă și folosită și în [60]. Folosind un filtru Kalman extins, poziția 3D a trăsăturilor este urmărită în timp, având ca măsurători, poziția acestora în imagine, fluxul optic precum și disparitatea. Odometria vizuală se obține minimizând erorile de reproiecție folosind o metoda neliniară robustă, care la fiecare iterație de minimizare a funcției de cost, elimină trăsăturile care au erori reziduale mai mari decât un prag.

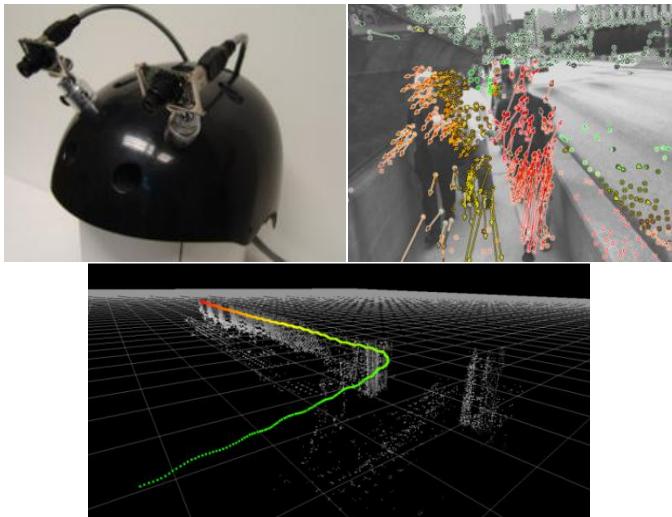


Figura 10 Stânga sus - cameră stereo folosită pentru persoane cu dizabilități [60]. Dreapta sus - trăsături folosite pentru estimarea mișcării, și vectori de mișcare pe obiecte dinamice (pietoni în acest caz). Jos - traectoria 3D și puncte 3D folosite pentru estimarea mișcării dintre imagini consecutive

Îmbunătățiri ulterioare ale metodei din [60] sunt introduse în [85]. Ca și alternativă la metode de integrare costisitoare de tipul bundle adjustment, pe lângă detecția curentă se menține și o poziție medie propagată din imaginile anterioare pentru fiecare trăsătură.

Estimarea finală se obține prin minimizarea simultană a erorilor de reproiecție în imaginea curentă atât pentru trăsătura detectată în imaginea anterioară (linii maro în Figura 11) cat și pentru media propagată până în imaginea anterioară (linii verzi în Figura 11). Pentru propagarea mediei, se propagă media din imaginea anterioară în imaginea curentă (poziția propagată este reprezentată cu triunghiuri verzi), se propagă trăsătura din imaginea anterioară în imaginea curentă (poziția trăsăturii propagată este reprezentată cu triunghiuri violet) și se mediază cele două poziții (dreptunghiul roșu).

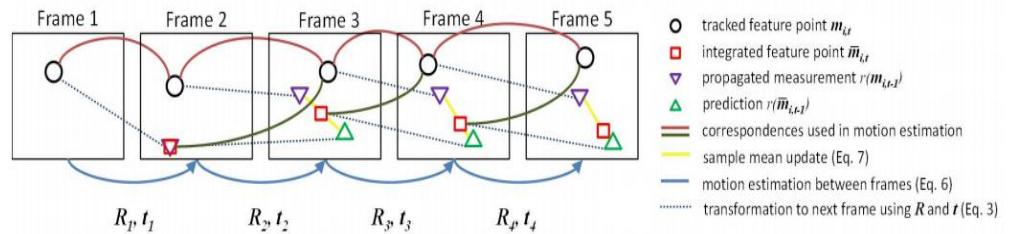


Figura 11 Ilustrarea grafică a algoritmului propus în [85] de propagare a mediei trăsăturilor din trecut pentru a îmbunătăți precizia de estimare a odometriei

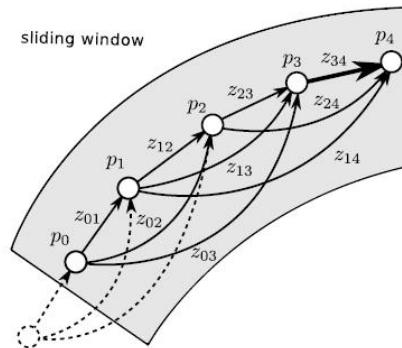


Figura 12 În [82] ultimele 4 poziții sunt rafinate formulând o funcție de cost pe baza mișcărilor relative z_{ij} dintre camere și impunând un model cu acceleratie liniară și unghiulară constantă [84]

Integrarea estimărilor dintre imagini succesive este explorată în [82], unde este propusă o alternativă pentru metode de integrare de tipul bundle adjustment, care pe lângă estimarea mișcării determină simultan și structura 3D a scenei într-un timp de calcul mai mare. Estimările între imagini succesive prin minimizarea erorilor de reproiecție propuse în [83] sunt introduse ca și intrări într-o problemă de optimizare care consideră mișcările relative dintre ultimele poziții

ale camerei (Figura 12) precum și modelul de mișcare cu acceleratie constantă și cu acceleratie unghiulară constantă evaluat în [84], producând rezultate superioare estimărilor făcute strict din imagini succesive. În general, integrarea de constrângeri adiționale care să reprezinte mai bine mișcarea obiectului pe care este montată camera este o metodă importantă de creștere a preciziei de estimare.

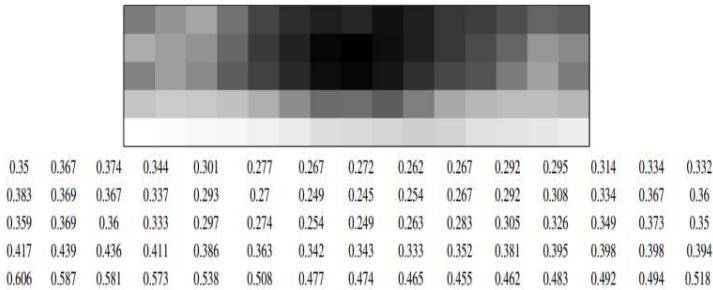


Figura 13 O reprezentare grafică a mediei erorilor de reproiecție din [86] (sus) și magnitudinea acestora (jos)

Este cunoscut faptul că pentru a folosi un model de proiecție perspectivă, camerele trebuie să fie calibrate în prealabil, iar pentru a modela distorsiunile lentilelor se folosesc modele de distorsiune precum cea tangențială sau cea radială. Așa cum este argumentat în [86], aceste model de distorsiune aproximează doar distorsiunile reale introduse de sistemul optic al camerei [87], și se propune o metodă de corectare a calibrării inițiale pe baza mișcării de referință din setul de date KITTI [72]. Observând empiric faptul că folosind odometria de referință, proiecția unor puncte din imaginea anterioară în imaginea curentă se realizează cu erori semnificativ de mari, se evaluează ipoteza că aceste erori ar putea fi provocate de un model de distorsiune imperfect. Se construiește o statistică a magnitudinii erorilor de reproiecție calculate pe baza mișcării de referință, prin acumulare pe mai multe secvențe video (Figura 13), pe regiuni distribuite peste imagine. Observăm faptul că erorile de reproiecție tind să devină mai mari înspre exteriorul imaginii și mai reduse înspre interior, motivul fiind o reprezentare corectă a distorsiunilor radiale. Având aceste observații, se pune și problema determinării unor deplasamente orizontale și verticale, pentru fiecare regiune, și folosind odometria de referință se formulează o funcție de cost care are ca scop estimarea deplasamentelor astfel încât să se atingă un optim, cunoscând trăsăturile, corespondențele temporale, corespondențele spațiale și

mișcarea de referință. Având deplasamentele, pozițiile trăsăturilor în imagine pot fi corectate, prin folosirea lor în problema de optimizare a odometriei. Figura 14 arată o traiectorie înainte și după corecția distorsiunilor folosind metoda din [86] descrisă sumar mai sus.

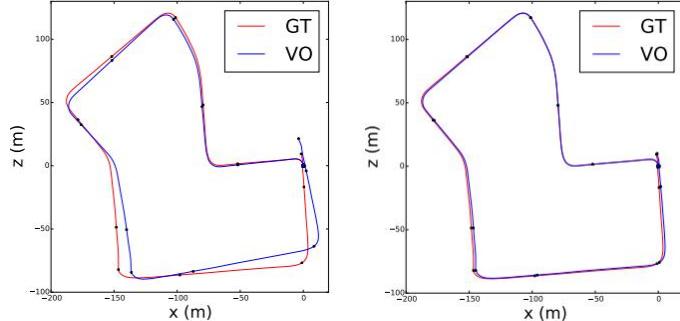


Figura 14 Traiectorii obținute cu metoda din [86] înainte (stânga) și după (dreapta) corecția erorilor de calibrare

În [39] este exploatată ideea separării trăsăturilor de la distanță de cele din apropiere, pentru un robot mobil având camera la o înălțime mică față de sol, și având aproximativ o treime din câmpul de percepție al unei camere stereo deasupra de linia orizontului. Separarea se face prin determinarea unui prag variabil în funcție de viteza pentru valorile disparităților trăsăturilor. Rotația este calculată prin optimizare non-lineară pe baza trăsăturilor din plan îndepărtat, iar translația se calculează pe baza trăsăturilor din apropiere, folosind valorile rotației calculate anterior. Chiar dacă în scenarii de automotive trăsături îndepărtate nu sunt întotdeauna disponibile (ex: înainte de a vira într-o curbă cu o clădire mare în față), în cazul robotilor mobili outdoor o astfel de separare oferă o stabilitate mai mare mai ales în situațiile în care distribuția trăsăturilor din apropiere este semnificativ mai mică decât a celor din plan îndepărtat.

O metodă îmbunătățită de distingere a trăsăturile pe baza cărora să se facă estimarea odometriei vizuale este abordată în [134]. Este subliniat faptul că în scenarii de autostradă, la viteze mari, datorită lipsei de textură, numărul trăsăturilor din apropiere este relativ mic, și acestea sunt eliminate într-un proces standard de estimare prin minimizarea erorii de reproiecție care, practic, elimină trăsături ale căror eroare de reproiecție este mai mare decât un prag fix. O analiză detaliată indică faptul că ordinul de magnitudine al erorilor de reproiecție pentru trăsăturile eronate de la distanță este același cu

ordinul de magnitudine pentru trăsăturile corecte din apropiere și un prag fix de eliminare a corespondențelor eronate de trăsături va duce la eliminarea trăsăturilor corecte din apropiere. Pe de altă parte aceste trăsături corecte din apropierea camerei au un rol important în precizia de estimare a translației, mai ales la viteze mari și este dorită folosirea lor în procesul de estimare.



Figura 15 Exemple de imagini achiziționate cu robotul mobil din [39] - Partea dreaptă indică cu verde un exemplu de scenariu în care estimarea separată a translației față de rotație crește acuratețea estimării

O analiză detaliată a magnitudinii erorilor de reproiecție arată faptul că acestea depind atât de poziția trăsăturii în imagine cât și de distanța la care acestea se află în scenă, iar soluția propusă în [134] este aceea de a normaliza erorile de reproiecție pe baza fluxului optic al fiecărei trăsături, rezultând o eroare normalizată de proiecție a cărei magnitudine devine aproximativ independentă de distanța în scenă și de poziția în imagine. Folosind această normalizare în selecția trăsăturilor folosite în procesul de optimizare neliniară va conduce la utilizarea trăsăturilor din apropierea camerei și implicit la o mai mare precizie a translației estimate (Figura 16). O îmbunătățire a abordării din [134] este propusă de către aceiași autori în [133]. Îmbunătățirea adusă este aceea de a face estimarea rotației și a translației într-o primă fază, urmând ca apoi să se eliminate efectul rotației asupra mișcării trăsăturilor. Normalizarea se face într-o a doua fază, când normalizarea erorilor de reproiecție pe baza fluxului optic cauzat strict de translație, este folosită atât la selecția trăsăturilor care respectă mișcarea, cât și la formularea funcției de minimizat. Această metodă de normalizare a erorilor de reproiecție poate fi folosită și în modelul analitic propus în această lucrare, folosind rotația obținută în mod analitic pe baza

punctelor de la distanță sau pe baza modelului de geometrie epipolară, aşa cum va fi explicitat în capitolele următoare.

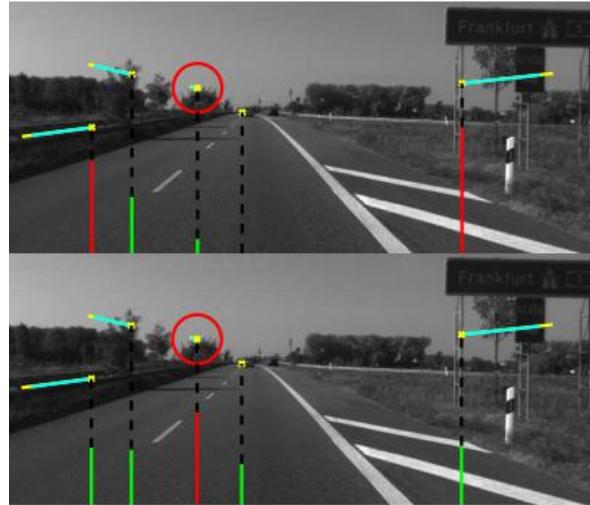


Figura 16 Exemplul din [134] cu erori standard de reproiecție (sus) comparativ cu erori normalizare (jos). Liniile verticale din partea de jos sunt proporționale în lungime cu erorile de reproiecție folosite pentru eliminarea corespondențelor eronate de trăsături. Culorile roșie este pentru trăsăturile eliminate, iar verde pentru cele folosite în procesul de minimizare liniară pentru a determina odometria. Se observă că dacă normalizăm erorile de reproiecție, trăsăturile corecte din apropiere sunt păstrate

O altă metodă de robustificare este introdusă în [77] și [135] având ca principiu reducerea numărului de parametri de mișcare și astfel, simplificarea modelului de mișcare a camerei cu scopul de a elimina corespondențele de trăsături, înainte de a fi incluse în iterațiile RANSAC. Aceasta funcționează în condițiile în care mișcarea vehiculului este una aproximativ planară (valori mici de pitch, roll și translație pe direcția verticală). Ideea de a folosi un model de mișcare simplificat am folosit-o și în [8] și [9] cu scopul de a plauzibiliza estimările de odometrie estimate pe baza matricii esențiale (rotația) și pe baza reconstrucției stereo (translația). De asemenea, această abordare ar putea fi folosită și ca un pas adițional, cu scop de prefiltrare în abordarea propusă în lucrarea curentă, deși testele curente indică faptul că metoda de conservare a unghiului și cea de potrivire circulară de trăsături, precum și estimarea în doi pași cu scopul de a avea o valoare bună de inițializare a potrivirii temporale de trăsături, oferă un grad suficient de ridicat de robustețe.

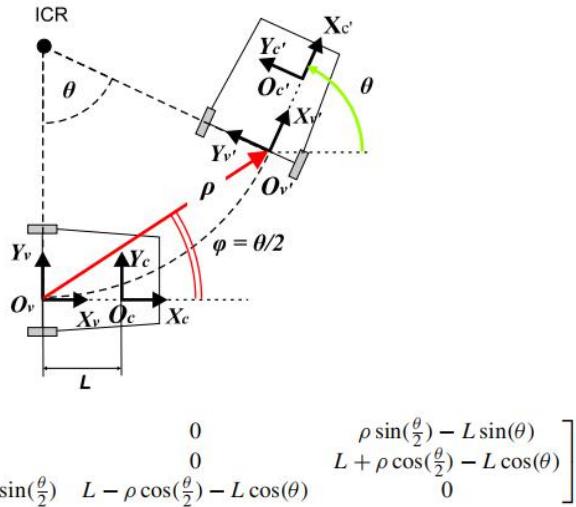


Figura 17 Varianta simplificată a matricii esențiale considerând modelul Ackerman de mișcare [135] (centrul axului spate al mașinii se mișcă pe un arc de cerc, iar sistemul de coordonate al camerei este considerat a fi la o distanță L în fața camerei)

Exploatând aceeași idee de a constrânge modelul de mișcare pe care am folosit-o în [8] și [9], în [141] este propusă o metodă prin care se estimează un factor de calitate al odometriei vizuale. Se observă că în cazul modelului Ackerman de mișcare, viteza laterală poate fi exprimată ca și o funcție de viteză longitudinală și yaw rate. Dacă această viteză laterală estimată diferă de estimarea laterală a odometriei vizuale cu 6 grade de libertate, atunci estimarea este marcată ca fiind invalidă. De asemenea, tot în [141] este propusă o metodă de exprimare a vitezei laterale, strict pe baza variației unghiului de rotație liniară, $v(\beta) = a \cdot \dot{\beta} + b$, unde valorile pentru a și b sunt estimate din date, pe acele imagini pe care odometria vizuală este corectă. Rezultatele experimentale prezentate în [141] indică faptul că această metodă este cel puțin la fel de bună pentru a plăzibiliza odometria ca și metoda de calcul a vitezei laterale pe bază de doi parametri.

Chiar dacă în lucrarea de față calitatea odometriei este indicată prin estimarea unei matrice de covarianță pentru cele 6 grade de libertate estimate, este posibilă implementarea acestieia cu metode de indicare a calității estimării, precum cele din [141], atâtă timp cât mișcarea poate fi aproximată cu o mișcare având mai puține grade de libertate.

3. Tendințe noi de cercetare în domeniul odometriei vizuale

3.1. Sisteme multi-cameră

Pe lângă sistemele cu o singură cameră mono, un sistem de camere stereo sau o cameră omnidirecțională, apare recent și problema estimării odometriei pe bază de sisteme de mai multe camere având câmpuri vizuale suprapuse sau nesuprapuse.

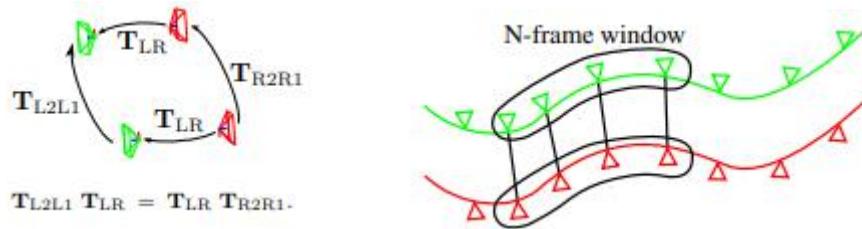


Figura 18 Estimarea magnitudinii vectorului de translație pe baza câmpurilor vizuale care nu se suprapun [74]

În [74] este prezentată o abordare prin care se estimează odometria vizuală, independent în două camere poziționate în oglinziile laterale ale unei mașini, urmând ca cele două estimări independente să se fuzioneze, identificând magnitudinea translației prin exprimarea constrângerii că transformata dintre cele două camere este una rigidă, precum este indicat în Figura 18. Pentru o mai mare stabilitate a estimării, se consideră că factorul de magnitudine al translației este constant pentru un anumit număr de imagini.

Pe baza modelului generalizat de cameră și a constrângerii epipolare generalizate din [76], un algoritm de estimare a mișcării folosind un sistem surround view de camere este introdus în [75]. Pornind de la ideea din [77] și [78] de a reduce numărul de parametri, constrângerea epipolară generalizată este simplificată, reducând numărul de parametri conform modelului Ackerman de mișcare. Unghiul de rotație și magnitudinea vectorului de translație sunt estimate folosind corespondențe între trăsături aparținând aceleiași camere sau aparținând unor camere diferite. Ulterior, este făcută o

optimizare neliniară atât pentru poziția 3D a trăsăturilor cât și pentru mișcarea 3D a camerelor pentru a rafina estimarea inițială.

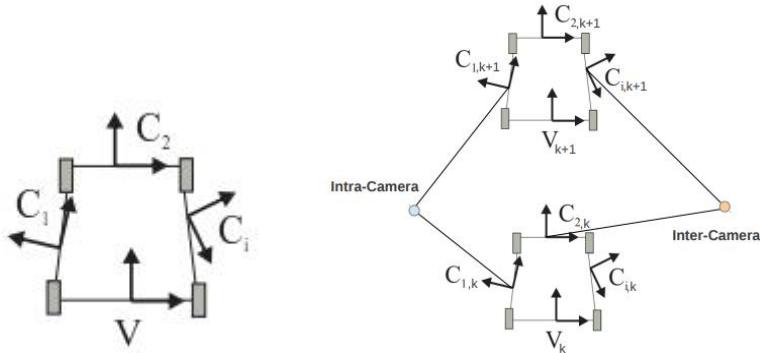


Figura 19 Stânga - sistem centralizat de cameră, prin introducerea unei camere virtuale V și generalizarea constrângerii epipolare relativ la această cameră virtuală [76]. Dreapta - tipuri de corespondențe de trăsături utilizate pentru estimarea mișcării între două achiziții succesive de imagini

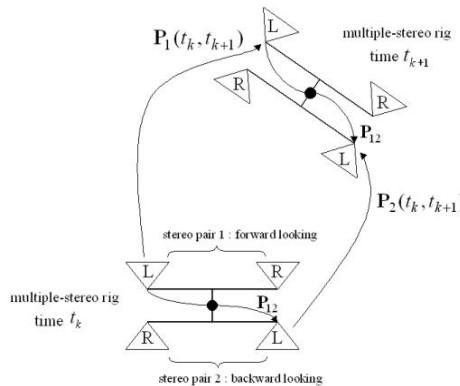


Figura 20 Sistem de două camere, orientate opus una față de celală [79]

În [79] este descris un sistem de două camere stereo, orientate opus una față de celală, pe un dispozitiv portabil. Principala contribuție este fuzionarea mișcării obținute de camera stereo frontală, cu estimarea camerei stereo orientată în sparte, precum și dezvoltarea unei metode de estimare automată a transformatei dintre cele două sisteme stereo. Calibrarea se face într-un mediu în care putem determina cu precizie odometria ambelor camere (ex: outdoor), iar ulterior pe baza celor două estimări independente se calculează transformata rigidă între cele două sisteme de coordonate. Având

această transformată, odometria oricărăia dintre cele două camere se poate transfera în camera opusă, permîțând astfel plauzibilizarea odometriei. Mișcarea estimată de camera frontală este propagată spre camera din spate și se calculează un scor total pe baza erorilor de reproiecție în cele două camere. La fel se procedează și cu camera din spate, selectând astfel estimarea mișcării camerei cu cel mai mare scor. Combinând astfel cele două camere, se crește acuratețea sistemului prin faptul că atunci când una dintre camere estimează greșit mișcarea (ex: un obiect dominant în mișcare, apropierea față de un perete netexturat), scorul mai mare și implicit estimarea finală va fi produsă de camera stereo opusă.

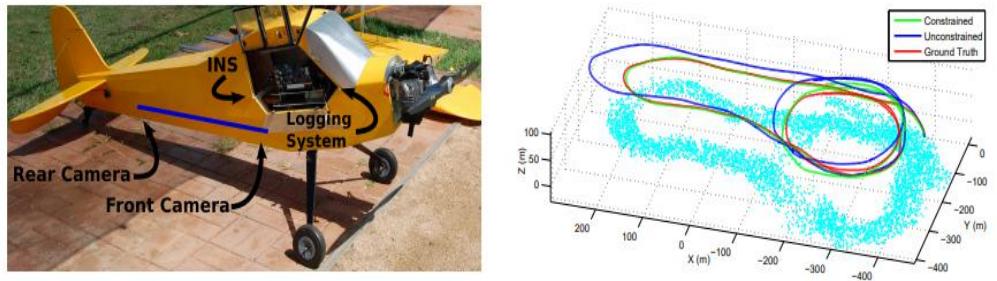


Figura 21 Sistemul folosit în [80] și o traiectorie obținută dintr-un zbor de 2.75km la altitudine de 20m/s la altitudini de 20-100m

O altă provocare a sistemelor cu mai multe camere este expusă și rezolvată în [80] unde se folosește un sistem de două camere poziționate pe un obiect zburător în cazul căruia, datorită vibrațiilor care apar în timpul funcționării, metodele clasice de stereoviziune nu se pot aplica deoarece calibrarea relativă dintre cele două camere variază în timp. Metoda propusă folosește bundle-adjustment [131] pentru estimarea odometriei vizuale, dar și a mișcării relativă dintre cele două camere, compensând în acest mod mișcările relative care apar datorită vibrațiilor. Detalii legate de metoda de calibrare sunt oferite și în [81]. De asemenea, considerând faptul că distanțele dintre cele două camere, precum și unghiurile relative pot varia doar în limite date de constrângerile mecanice (ex: câțiva milimetri pentru distanța dintre camere), se introduc aceste constrângerile în expresia funcției de cost pentru a asigura o mai bună stabilitate a problemei. Figura 21 arată o comparație între varianta constrânsă și cea neconstrânsă a problemei de optimizare.

Conceptele dezvoltate în prezenta lucrare sunt bazate pe camere stereo a căror poziționare relativă este una fixă în timp, dar pot fi generalizate pentru a încorpora vibrațiile relative dintre camere, aceasta putând constitui o problematică foarte bună, de continuare a acestei direcții de cercetare.

3.2. Metode pe bază de LiDAR și fuziune între video și LiDAR

În general pentru alinierea unui set de puncte 3D la un set de puncte 3D de referință, ceea ce implică determinarea transformației între cele două seturi de puncte, algoritmul ICP (iterative closest point) este frecvent utilizat [68]. Privit în linii mari, acesta filtrează punctele, apoi se determină potrivirile dintre punctele 3D și punctele din setul de referință, folosind metriki precum distanțele dintre puncte sau distanțele dintre puncte și planuri dominante, urmând ca pe baza acestor potriviri să se determine o transformată care minimizează eroarea de aliniere [69]. Procesul este repetat în mod iterativ. În cazul în care acuratețea de aliniere trebuie să fie ridicată, rafinările de detaliu ale metodei sunt foarte importante. Astfel, în [70] este propusă o metodă care selectează trăsăturile care intră în procesul de aliniere, precum și metricile utilizate. Se selectează astfel în prima fază doar puncte de muchie, sau puncte aparținând unor vecinătăți planare care nu sunt în apropierea limitelor de ocluzie și care nu sunt situate în plane aproximativ paralele cu cele ale razelor de proiecție. Pentru punctele de muchie se calculează muchiile corespondente în setul 3D de referință, iar pentru punctele cu vecinătăți planare se calculează planul corespunzător în setul 3D de referință, iar metrica de aliniere este considerată ca fiind distanța de la punct la dreaptă, respectiv distanța de la punct la plan. Transformata se găsește printr-un proces de optimizare neliniară. Aceste transformații relative între scanări LiDAR consecutive se acumulează în timp, rezultând o eroare de acumulare a erorii de aproximativ 0.5% din distanța parcursă și se pot construi pe baza lor hărți precise, precum cele din Figura 22.

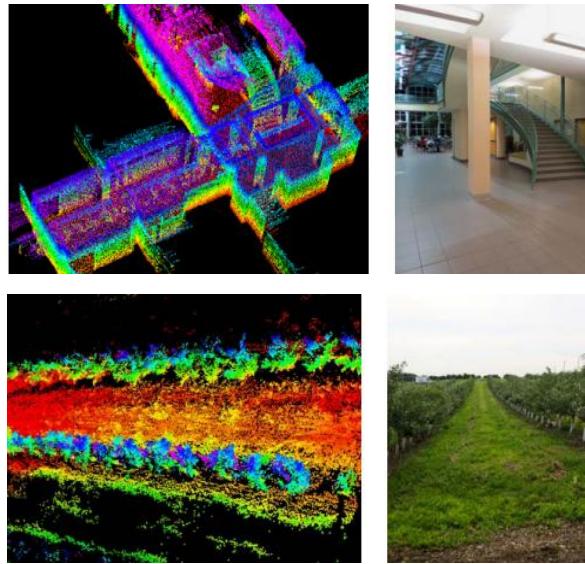


Figura 22 Hărți indoor (sus) și outdoor (jos) construite folosind metoda prezentată în [70]

Continuând eforturile din [70], în [71] este propusă o metodă care fuzionează măsurătorile de LiDAR și mono video, rezultând o precizie superioară față de metodele care folosesc un singur senzor. Metoda constă în determinarea de trăsături din video, care sunt urmărite în timp. O parte dintre aceste trăsături vor avea și informație de adâncime din LiDAR. Se formulează o funcție globală de cost care înglobează atât constrângeri 3D pentru trăsăturile care conțin informație de adâncime cât și constrângeri similare cu constrângerea epipolară [214] pentru punctele care nu conțin informație de adâncime. Prin minimizarea neliniară a acestei funcții de cost combinate se obțin transformatele între imagini succesive, iar acestea se rafinează pe baza punctelor 3D din LiDAR, folosind metoda din [70] la frecvența de 1Hz, rezultând astfel una dintre cele mai bune metode de odometrie cunoscută în prezent, pe KITTI, setul de date automotive introdus în [72] [73].

În prezenta lucrare vom evalua posibilitatea de a folosi harta de adâncime produsă de un senzor LiDAR împreună cu o camera monoculară, pentru a obține o estimare a odometriei între imagini consecutive. Corecția devierii acestei estimări pe bază de LiDAR este evaluată ca modalitate de postprocesare. Propunem o metodă eficientă

de corecție pe bază de LiDAR care nu necesită rutine costisitoare de optimizare neliniară.

3.3. Deep learning pentru odometrie vizuală

Chiar dacă încă nu au atins performanța metodelor pe bază de estimatori folosind camere stereo, metodele de estimare pe bază de învățare și pe bază de rețele neuronale convoluționale sunt foarte intens studiate în literatura de specialitate și prin urmare merită făcută o analiză a acestora pentru a observa principalele tendințe de dezvoltare precum și conexiunea dintre acestea și metodele de estimare clasice abordate în această lucrare. Punctul de pornire în cazul celor mai promițătoare metode de odometrie vizuală sunt metodele de reconstrucție 3D pe bază de învățare, ale căror performanță a crescut semnificativ și este comparabilă cu cea obținută prin reconstrucție stereo sau reconstrucție 3D din mișcare (structure from motion). Dacă primele abordări implică existența unor date de adâncime de referință, precum LiDAR sau time-of-flight [94], metodele principale de reconstrucție pe bază de învățare se bazează pe o funcție de cost care exploatează consistența spațială sau temporală a imaginilor stereo respectiv a imaginilor consecutive dintr-o secvență video. Una dintre primele abordări de estimare a reconstrucției 3D dintr-o singură imagine folosind rețele neuronale convolutionale și antrenată pe bază de imagini stereo este [88]. Funcția de penalizare este construită pe baza re-mapării pixelilor din imaginea stângă în imaginea dreaptă și viceversa, și pe baza diferenței de intensitate dintre imaginea actuală și cea sintetizată. Astfel se folosesc imagini stereo la antrenare, și inferența se face strict pe baza de imagini mono, fără a fi nevoie de un ground truth exterior. Din acest motiv acest tip de metode se consideră a fi auto supervizate (self supervised). Practic supervizarea în timpul antrenării nu necesită ground truth adnotat manual sau alți senzori de referință pe baza cărora să se formuleze funcția de penalizare a rețelei. Desigur că există varianta semi supervizată, care își propune să includă în funcția de cost o penalizare în plus acolo unde distanțele estimate de LiDAR sunt diferite față de cele rezultate în urma inferenței [89]. Pornind de la aceeași idee ca și în [88] de a formula funcția de cost pe baza diferenței de intensitate dintre imaginea sintetizată pe baza

imaginii reale și pe baza adâncimilor pixelilor din scenă, și observând că repamarea se poate face atât între imaginile stângă și dreaptă așa cum este cazul unui sistem stereo de camere [88] cât și în timp atunci când mișcarea camerei este cunoscută, în [89] este propusă o metodă de învățare supervizată care estimează atât harta de adâncime a scenei cât și mișcarea camerei strict dintr-o cameră mono folosind consistența temporală a secvențelor video. Cele două rețele se antrenă simultan, având o funcție de cost care depinde de ambele ieșiri, și se pot evalua și folosi independent la testare.

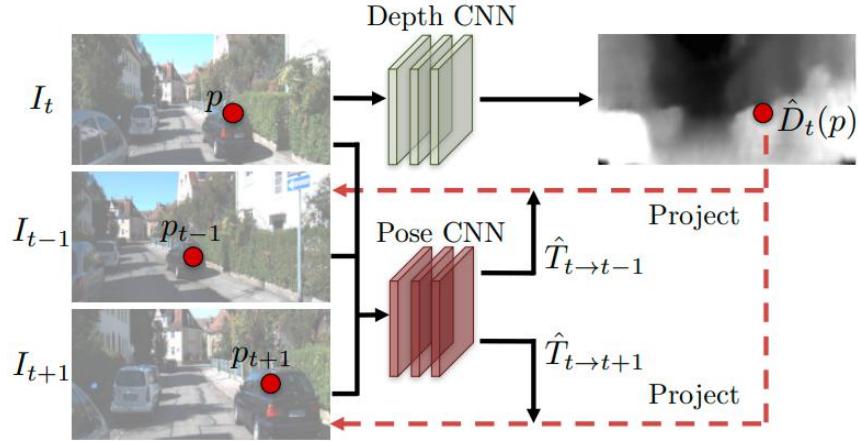


Figura 23 Ilustrarea algoritmului din [89]

Deoarece performanța acestor metode de învățare a odometriei este sub nivelul metodelor tradiționale de estimare, apar variațiuni care combină metodele de învățare a adâncimii cu metodele directe de odometrie vizuală [93] [92] sau care combină învățarea de trăsături care sunt folosite ca și valori de intrare în metodele consacrate de odometrie vizuală [91].

Bazându-se pe metode directe de odometrie vizuală investigate anterior [100], [101], [102], [103], [104], [105], în [93] este propusă o metodă care elimină neajunsurile metodelor monoculare de odometrie vizuală, încorporând predicțiile unei rețele de estimare a adâncimii, care la rândul ei este antrenată pe bază de imagini stereo. La fel ca și în [93], în [92] se construiește peste metoda de reconstrucție a hărții de adâncime pe bază de rețele neuronale convoluționale din [106]. Funcția de cost include atât elemente care să constrângă consistența spațială cât și elemente care să constrângă consistența temporală, iar pentru constrângerile temporale este estimată și odometria vizuală pe baza

unei rețele neuronale. Pe lângă harta de adâncime design-ul rețelei permite și generarea unei hărți de incertitudine, iar rafinarea finală a odometriei încorporează atât această hartă de incertitudine cât și estimările de odometrie făcute intern cu scop de auto supervizare a antrenării rețelei care generează harta de adâncime. Deși nu se bazează pe o metodă directă design-ul algoritmului prezentat în lucrarea curentă permite folosirea unei hărți de adâncime generată pe bază de rețele neuronale convoluționale ca sursă de intrare.

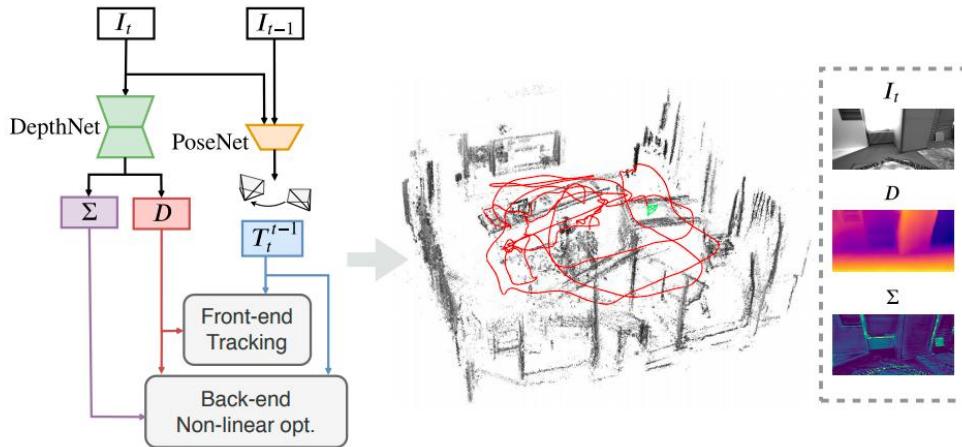


Figura 24 [92]

O altă abordare derivată din rețelele convoluționale de estimare a adâncimii este aceea de a estima simultan trăsături de tip punct, precum și descriptorii acestora. Inspirat de [107] și extinzând abordările din [108] și [109], în [110] este propusă o rețea neuronală care optimizează simultan estimarea unei hărți de adâncime și a unui set de trăsături de tip punct care optimizează simultan atât o funcție de cost specifică rețelelor de estimare a adâncimii (calculată pe baza diferențelor dintre imaginea reală și imaginea sintetizată pe baza de adâncimi), cât și o funcție de cost exprimată geometric la nivel de trăsătură similară cu eroarea de re-proiecție (Figura 25). De remarcat că modelul matematic folosit în lucrarea curentă poate fi folosit atât pentru a înlocui blocul diferențial de estimare a mișcării cât și ulterior pentru estimarea mișcării online, pe baza trăsăturilor estimate.

În aceeași ordine de idei, în [111] este propusă o metodă de odometrie vizuală care folosește atât harta de adâncime cât și o hartă de flux optic dens [112], ambele calculate folosind rețele neuronale

convoluționale. Pe baza celor mai buni vectori de flux optic, se folosește matricea esențială și reconstrucție 3D la viteze suficient de mari, respectiv minimizarea erorii de reproiecție la viteze suficient de mici.

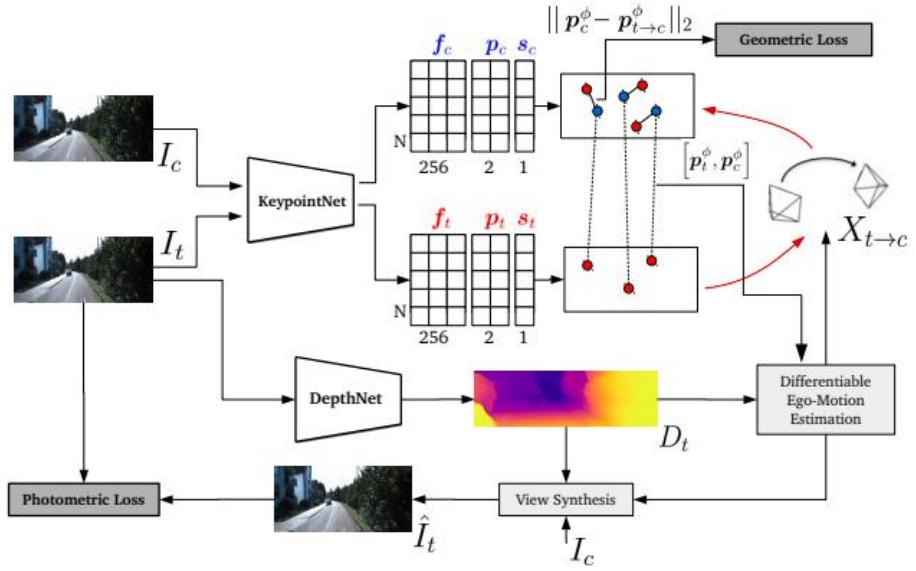


Figura 25 Ilustrarea algoritmului din [110]

Ca și concluzie, punctăm faptul că metodele prezentate în această lucrare pot fi ușor folosite împreună cu hărți de adâncime, vectori de flux optic sau trăsături estimate prin metode pe bază de rețele neuronale convoluționale și că modelele de optimizare pentru estimarea odometrie vizuale pot fi incorporate în funcțiile de penalizare (loss functions) folosite în etapele de antrenare a acestor rețele neuronale.

CONTRIBUȚIA PERSONALĂ

(obligatoriu pe pagină impară)

1. Modelarea matematică a problemei de estimare a odometriei vizuale

1.1. Considerații teoretice

1.1.1. Notații și considerente legate de geometria camerei

Din punct de vedere geometric procesul de formare al imaginii se va modela printr-o proiecție perspectivă (modelul pinhole). Scopul este de acela de a obține transformări dintre punctele 3D având coordonate metrice și punctele din planul imagine având coordonate exprimate în pixeli. În planul imagine vom nota cu u coordonatele în pixeli pe coloane (lățime) și cu v coordonatele în pixeli pe linii (înălțime). Vom folosi sistemul de coordonate 3D al camerei, având centrul în centrul de proiecție al camerei (punctul O în Figura 26). Axa X a sistemului de coordonate este paralelă cu axa u a imaginii iar axa Y este paralelă cu axa v a imaginii. Axa Z a sistemului de coordonate va fi perpendiculară pe planul imagine și poartă denumirea de axă optică (perpendiculară din centrul de proiecție pe planul imagine). Aceasta intersectează planul imagine în punctul principal având coordonatele în pixeli (u_0, v_0) . Distanța dintre centrul de proiecție și planul imagine (f este măsurat în unități metrice) se numește distanță focală.

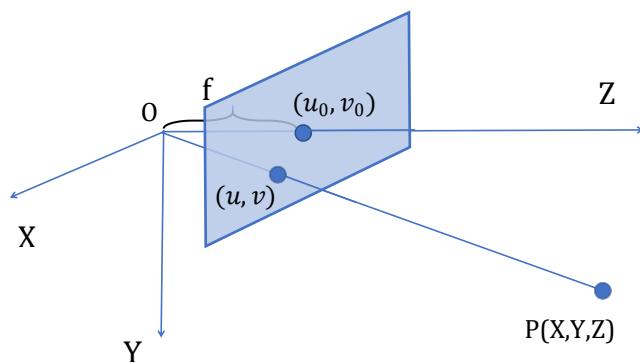


Figura 26 Modelul de proiecție perspectivă

Dându-se un punct P de coordonate metrice (X, Y, Z) și notând cu (x, y, f) coordonatele metrice ale intersecției dintre dreapta PO și planul imagine se pot deduce din asemănări de triunghiuri următoarele relații:

$$\frac{x}{X} = \frac{y}{Y} = \frac{z}{f} \Rightarrow x = \frac{fx}{Z} \text{ și } y = \frac{fy}{Z} \quad (1)$$

Transformarea din coordonate metrice în coordonate exprimate în pixeli se poate face pe baza dimensiunii metrice a unui pixel. Considerând s_x și s_y dimensiunile unui pixel pe coloane, respectiv pe linii obținem:

$$u_x = \frac{x}{s_x} = \frac{fx}{Z \cdot s_x} \quad (2)$$

$$v_y = \frac{y}{s_y} = \frac{fy}{Z \cdot s_y} \quad (3)$$

Perechea (u_x, v_y) reprezintă coordonatele în pixeli relativ la punctul principal (u_0, v_0) , punct de intersecție al axei optice cu planul imagine. Raporturile $f_x = \frac{f}{s_x}$ și $f_y = \frac{f}{s_y}$ reprezintă distanța focală exprimată în unități de pixeli orizontali respectiv verticali:

$$u_x = \frac{f_x \cdot X}{Z} \quad (4)$$

$$v_y = \frac{f_y \cdot Y}{Z} \quad (5)$$

Exprimând coordonatele în pixeli relativ la colțul stânga sus al imaginii obținem un set de ecuații pe baza cărora determinăm proiecția unui punct 3D (X, Y, Z) în imagine:

$$u = u_x + u_0 = \frac{f_x \cdot X}{Z} + u_0 \quad (6)$$

$$v = v_y + v_0 = \frac{f_y \cdot Y}{Z} + v_0 \quad (7)$$

Scrise sub formă matriceala și folosind coordonate omogene rezultă:

$$q = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z} \cdot K \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = K \cdot \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{bmatrix} \quad (8)$$

$$\text{sau echivalent, } \mathbf{q} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim K \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (9)$$

$$\text{unde } K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$v_1 \sim v_2$ reprezintă egalitate de vectori ignorând scara (vectorii au aceeași direcție dar pot avea norme diferite). În literatura de specialitate [1][2][4] matricea K poartă denumirea de matricea parametrilor intrinseci. Se observă de asemenea că prin înmulțirea unui punct din planul imagine cu inversa matricii K se obține un punct în coordonate normalize (într-un plan imaginar aflat la o unitate față de centrul de proiecție).

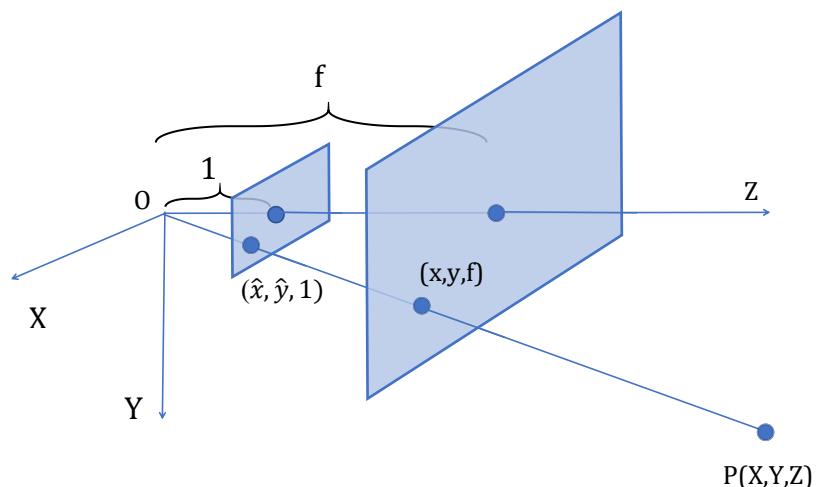


Figura 27 Coordonate normalize

Figura 27 ilustrează noțiunea coordonatelor normalize folosind notațiile din ecuațiile (11) și (12). Modelul matematic de estimare a odometriei vizuale va fi specificat prin analogie atât în coordonate imagine cât și în coordonate normalize.

$$\mathbf{p} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{u - u_0}{f_x} \\ \frac{v - v_0}{f_y} \\ 1 \end{bmatrix} \quad (11)$$

$$\text{unde } K^{-1} = \begin{bmatrix} \frac{1}{f_x} & 0 & \frac{-u_0}{f_x} \\ 0 & \frac{1}{f_y} & \frac{-v_0}{f_y} \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Acstea relații denotă faptul că adâncimea (componenta Z din ecuații) nu poate fi estimată folosind modelul de proiecție asociat unei singure camere.

Modelul pinhole ar fi valabil dacă proiecția lentilelor ar fi perfectă. Desigur că în realitate acest lucru nu se întâmplă, prin urmare distorsiunile lentilelor trebuie modelate și eliminate. Punctăm acest aspect deoarece o modelare precisă a distorsiunilor este un aspect esențial, imperfecțiunile de modelare reflectându-se direct în precizia de estimare a odometriei vizuale [86][221].

Fără distorsiuni am avea următoarea transformare din coordonate (u_x, v_x) relativ la punctul principal (u_0, v_0) în coordonate (u, v) relativ la colțul din stânga sus al imaginii:

$$u = u_x + u_0 \text{ și } v = v_y + v_0 \quad (13)$$

Transformarea ținând cont de distorsiunile introduse de lentile devine:

$$u_d = u_x + du_r + du_t + u_0 \text{ și } v_d = v_y + dv_r + dv_t + v_0 \quad (14)$$

Distorsiunea radială (du_r, dv_r) poate fi modelată folosind polinoame având ca și argument distanța r față de punctul principal ($r^2 = u_x^2 + v_y^2 = (u - u_0)^2 + (v - v_0)^2$):

$$du_r = u_x \cdot (c_1 \cdot r^2 + c_2 \cdot r^4 + ...) \quad (15)$$

$$dv_r = v_y \cdot (c_1 \cdot r^2 + c_2 \cdot r^4 + ...) \quad (16)$$

În mod similar, distorsiunea tangențială este modelată tot pe baza unei funcții polinomiale astfel:

$$du_t = 2u_x \cdot v_y \cdot c_3 + c_4 \cdot (r^2 + 2u_x^2) \quad (17)$$

$$dv_t = c_3 \cdot (r^2 + 2v_y^2) + 2u_x \cdot v_y \cdot c_4 \quad (18)$$

Observăm că pe baza ecuațiilor (14) - (18) coordonatele distorsionate (u_d, v_d) se scriu ca și o funcție de coordonatele nedistorsionate (u, v) , de centrul camerei (u_0, v_0) și de coeficienții polinoamelor care modelează distorsiunile lentilelor. Calibrarea unei camere [58] presupune determinarea distanțelor focale (f_x, f_y) a centrului camerei (u_0, v_0) precum și a coeficienților de distorsiune. Pe baza coeficienților de distorsiune imaginea distorsionată achiziționată de cameră se transformă într-o imagine nedistorsionată, urmând ca următoarele etape de prelucrare să se efectueze pe imaginea din care efectele distorsiunilor lentilelor au fost eliminate. Practic pentru fiecare pixel (u, v) din imaginea nedistorsionată se calculează coordonatele acestuia (u_d, v_d) în imaginea distorsionată. Deoarece (u_d, v_d) nu vor avea valori întregi, intensitatea pixelului de coordonate (u, v) din imaginea nedistorsionată se calculează prin interpolarea valorilor intensităților pixelilor din jurul punctului (u_d, v_d) .

Semnificația rotației	Semnificația translației	Transformare din sistemul de referință în sistemul camerei
C->R	C->R reprezentată în sistemul camerei	$P = R^{-1} \cdot P_R + T$
	R->C reprezentată în sistemul de referință	$P = R^{-1} \cdot (P_R - T)$
R->C	C->R reprezentată în sistemul camerei	$P = R \cdot P_R + T$
	R->C reprezentată în sistemul de referință	$P = R \cdot (P_R - T)$

Tabel 1 Modalități de reprezentare a rotației și translației dintre două sisteme de coordonate

Pentru a utiliza matricea parametrilor intrinseci în scopul proiecțării punctelor 3D în planul imagine trebuie ca acestea să fie reprezentate în sistemul 3D de coordonate al camerei. Astfel pentru a proiecta punctele 3D dintr-un sistem de referință oarecare în planul imagine, acestea sunt transformate în sistemul camerei urmând ca apoi să fie proiectate în planul imaginii. Există mai multe modalități de exprimare a acestei transformări folosind matrici de rotație și vectori de translație, în funcție de semnificația exactă a matricii de rotație și a

vectorului de translație. Astfel matricea de rotație poate să reprezinte o rotație de la sistemul de referință la sistemul camerei sau de la sistemul camerei la sistemul de referință. De asemenea vectorul de translație poate să reprezinte vectorul translație de la sistemul de referință la sistemul camerei sau de la sistemul camerei la sistemul de referință.

$$\text{Fie } P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \text{ și } P_R = \begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix} \text{ coordonatele aceluiași punct 3D}$$

reprezentat în sistemul camerei respectiv în sistemul de referință. O transformare de tipul $P = R \cdot P_R + T$ (rândul al treilea din Tabelul 1) se poate interpreta astfel: în prima fază facem o rotație a vectorului P_R obținând reprezentarea acestuia, $R \cdot P_R$, într-un sistem de coordinate având centrul în sistemul de referință și axele paralele cu sistemul camerei. Vectorul T reprezintă în acest caz poziția centrului sistemului de referință în sistemul de coordonate al camerei. Figura 28 ilustrează cazul $P = R \cdot P_R + T$ din Tabel 1, reprezentare pe care o vom folosi în continuare în această lucrare. Această transformare se poate scrie în mod compact ca și o matrice de 4×4 (cunoscută în literatura și sub denumirea de matricea parametrilor extrinseci [1]):

$$A = [R \quad T] \quad (19)$$

Astfel ecuația de proiecție din sistemul de referință în planul imagine devine:

$$q = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \approx K \cdot A \cdot \begin{bmatrix} X_R \\ Y_R \\ Z_R \\ 1 \end{bmatrix} = M \cdot \begin{bmatrix} X_R \\ Y_R \\ Z_R \\ 1 \end{bmatrix} \quad (20)$$

Matricea $M = K \cdot A$ este matricea de proiecție din coordonate 3D în planul imagine. În cazul în care avem imagini succesive preluate cu aceeași camera, prin considerarea dimensiunii temporale apar situații în care un set de puncte 3D corespunzător imaginii anterioare trebuie proiectat în imaginea curentă. În aceasta situație rotația și translația vor reprezenta mișcarea camerei de la momentul anterior de achiziție al imaginii la momentul curent de achiziție. Astfel toate modalitățile de reprezentare din Tabel 1 rămân valabile, cu mențiunea că ele reprezintă

transformări dintre sistemul de coordonate anterior și sistemul de coordonate curent.

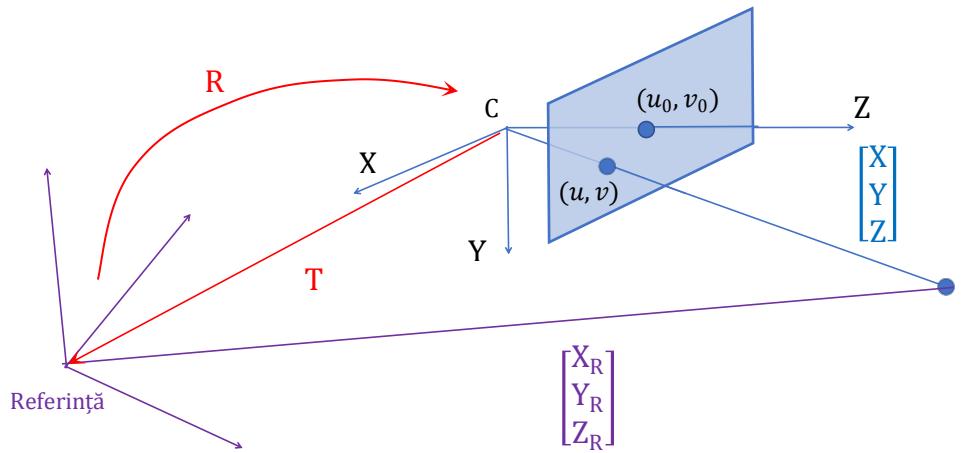


Figura 28 Parametrii extrinseci

1.1.2. Notații și considerente de viziune stereo

Având două camere calibrate (centre principale, coeficienții de distorsiune, distante focale) care vizualizează aceeași scenă și cunoscând poziția relativă dintre cele două camere se poate efectua o reconstrucție 3D a scenei. Practic din două proiecții perspective ale aceleiași scene se obține geometria scenei prin triangulare.

În particular dacă cele două camere au liniile perfect aliniate atunci corespondențele unui pixel (u, v) din imaginea stângă se va situa pe aceeași linie (aceeași coordonată v) în imaginea dreaptă la un deplasament d spre stânga numit disparitate $(u - d, v)$.

Având un punct de coordonate (X, Y, Z) în sistemul de coordonate al camerei stângi, și presupunând că sistemul de coordonate al camerei stângi este perfect aliniat cu cel al camerei drepte, acesta va avea coordonatele $(X - b, Y, Z)$ în sistemul de coordonate al camerei drepte.

Valoarea b este distanța dintre centrele de proiecție ale camerelor sistemului stereo și se numește baseline (Figura 29).

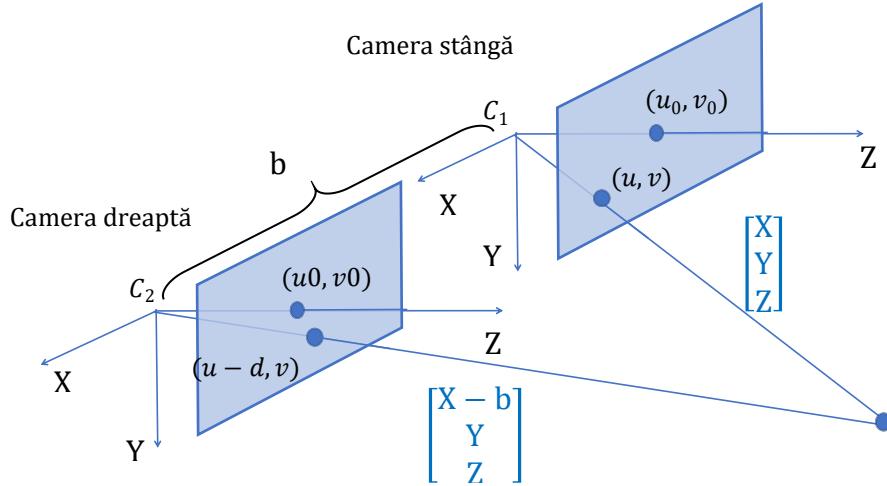


Figura 29 Geometria unui sistem stereo de camere rectificate

Proiectând punctul din spațiul 3D în cele 2 camere rezultă:

$$\begin{aligned} x_L &= \frac{f \cdot X}{Z} \text{ și } x_R = \frac{f \cdot (X-b)}{Z} \Rightarrow d_{metric} = x_L - x_R = \frac{f \cdot b}{Z} \Rightarrow \\ &\Rightarrow Z = \frac{f \cdot b}{d_{metric}} \end{aligned} \quad (21)$$

Împărțind la dimensiunea orizontală a unui pixel s_x și presupunând că centrele celor două camere au aceeași coordonată pe orizontală (lucru care se obține prin rectificare) obținem:

$$\begin{aligned} \frac{x_L}{s_x} - \frac{x_R}{s_x} &= \frac{f \cdot b}{Z \cdot s_x} \Rightarrow \left(\frac{x_L}{s_x} + u_0 \right) - \left(\frac{x_R}{s_x} + u_0 \right) = \frac{f \cdot b}{Z} \Rightarrow \\ &\Rightarrow d_{pixel} = u_L - u_R = \frac{f \cdot b}{Z} \Rightarrow \\ &\Rightarrow Z = \frac{f \cdot b}{d_{pixel}} \end{aligned} \quad (22)$$

Lucrând în coordinate normalize (obținute prin înmulțirea cu inversa matricii parametrilor intrinseci):

$$\hat{x}_L = \frac{u_L - u_0}{f_x} \text{ și } \hat{x}_R = \frac{u_R - u_0}{f_x} \Rightarrow d_{norm} = \hat{x}_L - \hat{x}_R = \frac{d_{pixel}}{f_x} \Rightarrow$$

$$\Rightarrow Z = \frac{b}{d_{norm}} \quad (23)$$

Datorită imperfecțiunilor apărute în procesul de fabricație vor exista devieri între orientarea camerei drepte față de camera stângă. Valorile acestor devieri pot fi determinate printr-un proces de calibrare [57] [58]. Pe baza acestor devieri se face o rectificare a imaginii originale nedistorsionate printr-o transformare liniară aplicată coordonatelor omogene ale pixelilor [56]. În urma procesului de rectificare corespondențele stereo ale trăsăturilor din imaginea stângă se vor găsi pe aceeași linie și în imaginea dreaptă. Pentru eficiență etapa de rectificare se combină cu etapa de eliminare a distorsiunilor introduse de lentile [55].

1.1.3. Noțiuni geometrice folosite în metodele de odometrie vizuală

1.1.3.1. Rotația unei camere

Deși în cazul odometriei vizuale sunt foarte puține cazurile în care se efectuează pur o mișcare de rotație, merită analizat acest caz particular. Motivul este acela că mișcarea trăsăturilor în cazul particular în care camera efectuează strict o rotație este similară cu mișcarea trăsăturilor de la distanță în cazul în care avem o mișcare generală a camerei (rotație plus translație).

$$\begin{aligned} q &= \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \cdot K \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \Rightarrow \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \sim K^{-1} \cdot q \\ q' &= \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \sim \cdot K \cdot \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \Rightarrow \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \sim K^{-1} \cdot q' \\ \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} &= R \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \Rightarrow K^{-1} \cdot q' \sim R \cdot K^{-1} \cdot q \Rightarrow \\ &\Rightarrow q' \sim K \cdot R \cdot K^{-1} \cdot q \end{aligned} \quad (24)$$

Observăm echivalență matematică cu variația punctelor de la infinit în cazul unei mișcări generale (homografie indușă de planul de la infinit), exprimată în ecuația (7) din Anexa 10.

1.1.3.2. Unghiul dintre perechi de raze de proiecție

O cantitate deosebit de utilă în eliminarea corespondentelor eronate de trăsături dintre imagini succesive este unghiul dintre perechi de astfel de trăsături. Având două puncte tridimensionale în sistemul de coordonate al camerei:

$$P^i = \begin{bmatrix} X^i \\ Y^i \\ Z^i \end{bmatrix} \text{ și } P^j = \begin{bmatrix} X^j \\ Y^j \\ Z^j \end{bmatrix}$$

Putem calcula unghiul $\theta_{i,j}$ format de aceste două trăsături și centrul camerei folosind produsul scalar dintre cei doi vectori:

$$\|P^i\| \cdot \|P^j\| \cdot \cos(\theta_{i,j}) = P^{iT} \cdot P^j$$

$$\cos(\theta_{i,j}) = \frac{P^{iT} \cdot P^j}{\|P^i\| \cdot \|P^j\|} \quad (25)$$

Observăm de asemenea că unghiul este invariant la variația celor două puncte pe aceeași raze de proiecție, deci va fi același pentru orice valori scalare a_i, a_j :

$$\frac{(a_i \cdot P^i)^T \cdot (a_j \cdot P^j)}{\|a_i \cdot P^i\| \cdot \|a_j \cdot P^j\|} = \frac{a_i \cdot (P^i)^T \cdot a_j \cdot (P^j)}{a_i \cdot \|P^i\| \cdot a_j \cdot \|P^j\|} = \frac{P^{iT} \cdot P^j}{\|P^i\| \cdot \|P^j\|} = \cos(\theta_{i,j})$$

Din ecuațiile (8) și (11) rezultă următoarele relații:

$$P^i = \begin{bmatrix} X^i \\ Y^i \\ Z^i \end{bmatrix} \sim \begin{bmatrix} X^i \\ \bar{Z}^i \\ Y^i \\ \bar{Z}^i \\ 1 \end{bmatrix} = p^i = \begin{bmatrix} x^i \\ y^i \\ 1 \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} u^i \\ v^i \\ 1 \end{bmatrix} = K^{-1} \cdot q^i$$

$$= \begin{bmatrix} \frac{u^i - u_0}{f_x} \\ \frac{v^i - v_0}{f_y} \\ 1 \end{bmatrix} \sim \begin{bmatrix} u^i - u_0 \\ f_x \cdot \frac{v^i - v_0}{f_y} \\ f_x \end{bmatrix} \quad (26)$$

și

$$\begin{aligned} P^j &= \begin{bmatrix} X^j \\ Y^j \\ Z^j \end{bmatrix} \sim \begin{bmatrix} X^j \\ \frac{Z^j}{f_x} \\ \frac{Y^j}{f_x} \\ 1 \end{bmatrix} = p^j = \begin{bmatrix} x^j \\ y^j \\ 1 \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} u^j \\ v^j \\ 1 \end{bmatrix} = K^{-1} \cdot q^j \\ &= \begin{bmatrix} \frac{u^j - u_0}{f_x} \\ \frac{v^j - v_0}{f_y} \\ 1 \end{bmatrix} \sim \begin{bmatrix} u^j - u_0 \\ f_x \cdot \frac{v^j - v_0}{f_y} \\ f_x \end{bmatrix} \quad (27) \end{aligned}$$

Putem calcula valoarea cosinusului unghiului dintre cele două raze de proiecție folosind vectorii p^i , p^j sau $K^{-1} \cdot q^i$, $K^{-1} \cdot q^j$:

$$\cos(\theta_{i,j}) = \frac{p^i \cdot p^j}{\|p^i\| \cdot \|p^j\|} = \frac{x^i \cdot x^j + y^i \cdot y^j + 1}{\sqrt{x^{i^2} + y^{i^2} + 1} \cdot \sqrt{x^{j^2} + y^{j^2} + 1}} \quad (28)$$

Dacă $f_x = f_y$ (lucru de obicei valabil în urma etapelor de rectificare și de eliminare distorsiunilor lentilelor) atunci:

$$\begin{aligned} \cos(\theta_{i,j}) &= \frac{(K^{-1} \cdot q^i)^T \cdot (K^{-1} \cdot q^j)}{\|K^{-1} \cdot q^i\| \cdot \|K^{-1} \cdot q^j\|} = \\ &= \frac{(u^i - u_0) \cdot (u^j - u_0) + (v^i - v_0) \cdot (v^j - v_0) + f_x^2}{\sqrt{(u^i - u_0)^2 + (v^i - v_0)^2 + f_x^2} \cdot \sqrt{(u^j - u_0)^2 + (v^j - v_0)^2 + f_x^2}} \quad (29) \end{aligned}$$

1.1.3.3. Invarianța unghiului dintre perechi de raze de proiecție la rotație

În mod intuitiv, unghiul dintre două raze de proiecție și centrul camerei, cu vârful în centrul camerei, nu depinde de poziția și de orientarea planului imagine (și implicit al sistemului de coordonate al camerei), cât timp centrul camerei rămâne fix.

Folosind în continuare relația următoare, general valabilă pentru cazul unei rotații a camerei, conform ecuației (24), precum și în particular valabilă pentru punctele de la infinit în cazul unei mișcări generale, conform ecuației (7) din Anexa 10:

$$q' \sim K \cdot R \cdot K^{-1} \cdot q \Rightarrow K^{-1} \cdot q' \sim R \cdot K^{-1} \cdot q \quad (30)$$

Deducem:

$$\begin{aligned} \cos(\theta'_{i,j}) &= \frac{(K^{-1} \cdot q^i)^T \cdot (K^{-1} \cdot q^j)}{\|K^{-1} \cdot q^i\| \cdot \|K^{-1} \cdot q^j\|} = \frac{(R \cdot K^{-1} \cdot q^i)^T \cdot (R \cdot K^{-1} \cdot q^j)}{\|R \cdot K^{-1} \cdot q^i\| \cdot \|R \cdot K^{-1} \cdot q^j\|} = \\ &= \frac{(K^{-1} \cdot q^i)^T \cdot R^T \cdot R \cdot (K^{-1} \cdot q^j)}{\|K^{-1} \cdot q^i\| \cdot \|K^{-1} \cdot q^j\|} = \frac{(K^{-1} \cdot q^i)^T \cdot (K^{-1} \cdot q^j)}{\|K^{-1} \cdot q^i\| \cdot \|K^{-1} \cdot q^j\|} = \cos(\theta_{i,j}) \end{aligned}$$

Astfel, unghiul dintre perechi de puncte va fi invariant la mișcarea generală a camerei pentru punctele de la infinit, lucru care va fi utilizat în estimarea inițială a rotației.

1.1.3.4. Matricea fundamentală și matricea esențială

Considerăm două proiecții perspectivă sau imagini ale aceleiași scene. Acestea pot să provină din două camere diferite, sau de la aceeași camera la două momente diferite de timp. Pentru problema odometriei vizuale, cel de-al doilea caz este de interes. Fie P_1^i un punct în sistemul de coordonate al camerei la momentul anterior și P_2^i , un punct în sistemul de coordonate al camerei la momentul curent.

Considerând R matricea de rotație din sistemul de coordonate al primei camere în sistemul de coordonate al celei de-a două camera, și T matricea de translație de la prima camera la cea de-a două cameră exprimată în sistemul de coordonate al primei camere. Vom avea relația:

$$P_2^i = R \cdot (P_1^i - T) \Rightarrow P_1^i - T = R^T \cdot P_2^i \quad (31)$$

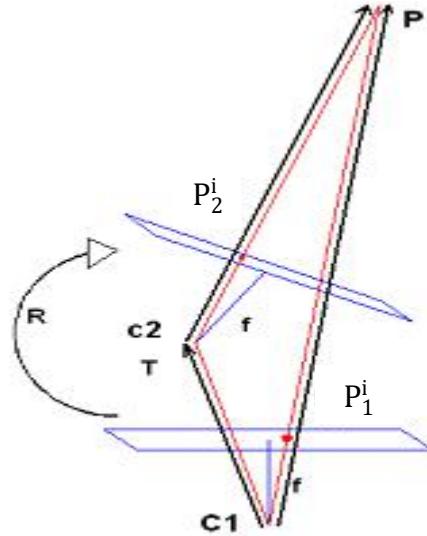


Figura 30 Matricea esențială. Figură preluată din [9]

Deoarece vectorii $P_1^i - T = R^T \cdot P_2^i$, T , și P_1^i sunt coplanari produsul scalar dintre primul vector și produsul vectorial dintre cel de-al doilea și cel de-al treilea vector este zero:

$$(R^T \cdot P_2^i)^T (T \times P_1^i) = 0 \quad (32)$$

Produsul scalar dintre vectorul $T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$ și un alt vector se poate exprima

ca și produs matriceal folosind matricea $T_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$,

rezultând astfel următoarea forma matriceala a ecuației de mai sus:

$$P_2^i{}^T \cdot R \cdot T_x \cdot P_1^i = 0 \quad (33)$$

Matricea $E_{12} = R \cdot T_x$ este matricea esențială a sistemului. În cazul în care matricea T reprezentă translația de la cea de-a două camera la prima, și este exprimată în sistemul de coordinate al celei de-a 2 camere, relația dintre vectorii implicați devine:

$$P_2^i = R \cdot P_1^i + T \quad (34)$$

La fel ca și în cazul anterior, vectorii P_2^i , $R \cdot P_1^i$ și T vor fi coplanari, rezultând:

$$P_2^i{}^T \cdot T_x \cdot R \cdot P_1^i = 0 \quad (35)$$

Astfel, cu noua semnificație a matricii de translație:

$$E_{12} = T_x \cdot R \quad (36)$$

În funcție de sistemul de coordonate în care se exprima vectorul de translație și semnificația acestuia, una dintre cele două forme de exprimare a matricii esențiale va fi folosită pentru estimarea odometriei vizuale. De remarcat este faptul că relația de mai sus este valabilă și în coordonate omogene p_2^i și p_1^i obținute prin împărțirea la coordonata Z a punctelor P_2^i respectiv P_1^i :

$$p_2^i{}^T \cdot E_{12} \cdot p_1^i = 0 \quad (37)$$

Acestea se obțin prin înmulțirea coordonatelor imagine cu inversa matricii parametrilor intrinseci:

$$p_1^i = \begin{bmatrix} x_1^i \\ y_1^i \\ 1 \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} \quad (38)$$

$$p_2^i = \begin{bmatrix} x_2^i \\ y_2^i \\ 1 \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix} \quad (39)$$

Înlocuind mai departe obținem:

$$[u_2^i \quad v_2^i \quad 1] \cdot K^{-T} \cdot E_{12} \cdot K^{-1} \cdot \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = 0 \quad (40)$$

Matricea obținută se numește matricea fundamentală a celor două camere:

$$F_{12} = K^{-T} \cdot E_{12} \cdot K^{-1} \quad (41)$$

Pe baza ei putem construi o relație de legătură între coordonatele imagine ale proiecțiilor aceluiași punct tridimensional în cele două imagini consecutive.

1.2. Modelarea matematică a metodelor stereo de odometrie vizuală pe bază de corespondențe de trăsături

În continuare se vor prezenta modelele matematice folosite pentru diverse metode de odometrie vizuala evaluate în prezența lucrare. Algoritmii evaluați vor avea ca și intrare două perechi de imagini achiziționate cu o camera stereo în două momente diferite de timp t_1 și t_2 , $t_1 < t_2$. Vom folosi denumirile de imagine anterioara stângă și imagine anterioara dreaptă pentru cele două imagini achiziționate la momentul t_1 și denumirile de imagine curentă stângă respectiv imagine curentă dreaptă pentru imaginile achiziționate la momentul t_2 . Modelele matematice se vor baza pe un set de trăsături corespondente între cele patru imagini: imaginea anterioara stângă, imaginea anterioara dreaptă, imaginea curentă stângă și imaginea curentă dreaptă. Aceste trăsături reprezintă proiecțiile perspectivă ale aceleiași primitive tridimensionale (puncte 3D de exemplu) în cele patru imagini.

Vom folosi în continuare următoarele notății (Figura 31) pentru coordonatele trăsăturii numărul i:

$$q_{L1}^i = \begin{bmatrix} u_{L1}^i \\ v_{L1}^i \\ 1 \end{bmatrix} \text{ coordonatele omogene în imaginea anterioara stângă;}$$

$$q_{R1}^i = \begin{bmatrix} u_{R1}^i \\ v_{R1}^i \\ 1 \end{bmatrix}$$

coordonatele omogene în imaginea anteroioara dreaptă;

$$q_{L2}^i = \begin{bmatrix} u_{L2}^i \\ v_{L2}^i \\ 1 \end{bmatrix}$$

coordonatele omogene în imaginea curenta stângă;

$$q_{R2}^i = \begin{bmatrix} u_{R2}^i \\ v_{R2}^i \\ 1 \end{bmatrix}$$

coordonatele omogene în imaginea curenta dreaptă;

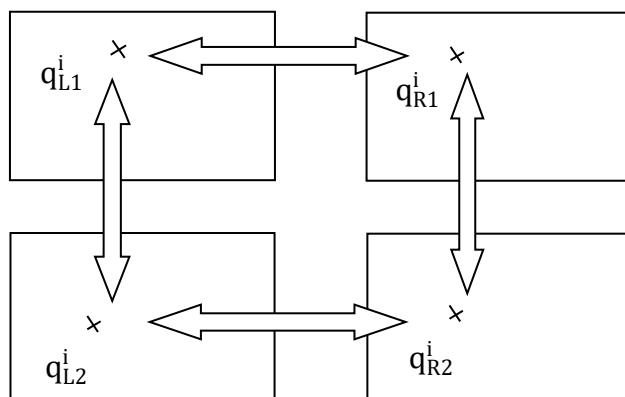


Figura 31 Corespondențe de trăsături între imagini succesive

Prin înmulțirea acestora cu inversa matricii parametrilor intrinseci obținem coordonatele normalize ale trăsăturilor. Pentru simplitatea notațiilor vom renunța la folosirea simbolului \wedge deasupra valorilor coordonatelor normalize:

$$p_{L1}^i = K^{-1} \cdot q_{L1}^i = \begin{bmatrix} x_{L1}^i \\ y_{L1}^i \\ 1 \end{bmatrix}$$

coordonate normalize în imaginea anteroiară
stângă;

$$p_{R1}^i = K^{-1} \cdot q_{R1}^i = \begin{bmatrix} x_{R1}^i \\ y_{R1}^i \\ 1 \end{bmatrix}$$

coordonate normalize în imaginea anteroiară
dreaptă;

$$p_{L2}^i = K^{-1} \cdot q_{L2}^i = \begin{bmatrix} x_{L2}^i \\ y_{L2}^i \\ 1 \end{bmatrix} \text{ coordonate normalizeaza în imaginea curentă stângă;}$$

$$p_{R2}^i = K^{-1} \cdot q_{R2}^i = \begin{bmatrix} x_{R2}^i \\ y_{R2}^i \\ 1 \end{bmatrix} \text{ coordonate normalizeaza în imaginea curentă dreaptă;}$$

1.2.1. Odometrie vizuala pe baza punctelor 3D reconstruite din stereo

Cel mai simplu model matematic se bazează pe reconstrucția 3D pe baza trăsăturilor corespondente. Pe baza punctelor 3D se determină rotația și translația dintre sistemul de coordonate anterior și cel curent.

În cazul în care nu lucrăm cu imagini rectificate punctele 3D se obțin prin triangularea [4] q_{L1}^i, q_{R1}^i pentru sistemul de coordonate al camerei anterioare și q_{L2}^i, p_{R2}^i pentru sistemul de coordonate al camerei curente. Dacă sistemul stereo este unul rectificat atunci $y_{L1}^i = y_{R1}^i, v_{L1}^i = v_{R1}^i, y_{L2}^i = y_{R2}^i$, și $v_{L2}^i = v_{R2}^i$. Reconstrucțiile 3d în sistemul de coordonate anterior respectiv în sistemul de coordonate curent vor fi:

$$P_1^i = \begin{bmatrix} \frac{b \cdot (u_{L1}^i - u_0)}{u_{L1}^i - u_{R1}^i} \\ \frac{f_x \cdot b \cdot (v_{L1}^i - v_0)}{f_y \cdot (u_{L1}^i - u_{R1}^i)} \\ \frac{f_x \cdot b}{u_{L1}^i - u_{R1}^i} \end{bmatrix} = \begin{bmatrix} \frac{b \cdot x_{L1}^i}{x_{L1}^i - x_{R1}^i} \\ \frac{b \cdot y_{L1}^i}{x_{L1}^i - x_{R1}^i} \\ \frac{b}{x_{L1}^i - x_{R1}^i} \end{bmatrix} \quad (42)$$

$$P_2^i = \begin{bmatrix} \frac{b \cdot (u_{L2}^i - u_0)}{u_{L2}^i - u_{R2}^i} \\ \frac{f_x \cdot b \cdot (v_{L2}^i - v_0)}{f_y \cdot (u_{L2}^i - u_{R2}^i)} \\ \frac{f_x \cdot b}{u_{L2}^i - u_{R2}^i} \end{bmatrix} = \begin{bmatrix} \frac{b \cdot x_{L2}^i}{x_{L2}^i - x_{R2}^i} \\ \frac{b \cdot y_{L2}^i}{x_{L2}^i - x_{R2}^i} \\ \frac{b}{x_{L2}^i - x_{R2}^i} \end{bmatrix} \quad (43)$$

Valoarea P_1^i reprezintă coordonatele punctului 3D în sistemul de coordonate al camerei anterioare stângi. Valoarea P_2^i reprezintă

coordonatele aceluiasi punct 3D în sistemul de coordonate al camerei curente stângi. Presupunem în continuare ca R este rotația folosita pentru a transforma din sistemul de coordonate al camerei anterioare stângi în sistemul de coordonate al camerei curente stângi. T este poziția centrului sistemului de coordonate al camerei anterioare stângi în sistemul de coordonate camerei curente stângi. Transformata punctului P_1^i în sistemul de coordonate al camerei curente stângi este o funcție de R și T:

$$\bar{P}_2^i(R, T) = R \cdot P_1^i + T \quad (44)$$

Construim următoarea funcție depinzând de parametrii R și T:

$$f(R, T) = \sum_{i=0}^n \| \bar{P}_2^i(R, T) - P_2^i \|^2 \quad (45)$$

unde $\left\| \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \right\| = \sqrt{v_1^2 + \dots + v_n^2}$.

Valorile rotației și translației dintre cele doua sisteme de coordonate corespund minimului funcției $f(R, T)$.

1.2.2. Odometrie vizuală pe baza tensorului trifocal

Din cunoștințele noastre, acest model a fost folosit pentru odometria vizuala pentru prima data în [52] și ulterior în [53] respectiv [54]. Așa cum se arata în [2] o linie într-una dintre cele trei proiecții determină împreună cu centrul acelei proiecții un plan în 3D. Un plan 3D oarecare induce o homografie pe baza căreia se pot transfera puncte dintr-o proiecție (imagină) în alta: $p_3 = H \cdot p_1$, unde p_1 este un punct în prima imagine iar p_3 este un punct în cea de-a treia imagine. Aceeași matrice de homografie se poate utiliza pentru a transfera linii astfel:

$$l_1 = H^T \cdot l_3 \quad (46)$$

Din relația tensorului trifocal avem:

$$l_1^i = l_2^T \cdot S_i \cdot l_3 \text{ sau } l_1 = \begin{bmatrix} l_2^T \cdot S_1 \\ l_2^T \cdot S_2 \\ l_2^T \cdot S_3 \end{bmatrix} \cdot l_3 \quad (47)$$

Comparând aceasta formula cu $l_1 = H^T \cdot l_3$ rezulta:

$$H = [S_1^T \cdot l_2 \quad S_2^T \cdot l_2 \quad S_3^T \cdot l_2] \quad (48)$$

Astfel ecuația de transfer devine:

$$p_3 = [S_1^T \cdot l_2 \quad S_2^T \cdot l_2 \quad S_3^T \cdot l_2] \cdot p_1 \quad (49)$$

Matricile S_i depind de rotațiile și de translațiile relative dintre cele trei proiecții. Pe baza acestui mecanism de transfer vom exprima o funcție neliniara, având ca și parametrii rotația și translația efectuata de camera stereo între două momente succesive de timp. Pentru ca aceasta formulare matematică să funcționeze în practică distorsiunile lentilelor trebuie eliminate, dar nu este neapărat necesar să avem imagini rectificate.

1. Definim un tensor trifocal între imaginea anteroioara stângă (prima proiecție), imaginea anteroioara dreapta (a doua proiecție) și imaginea curentă stângă (a treia proiecție). Pentru a construi tensorul trifocal pe baza modelului din Anexa 11 - Tensorul trifocal considerăm ca matricea parametrilor extrinseci pentru imaginea anteroioara stângă este $A_{L1}^L = [I \ 0]$, ceea ce înseamnă că sistemul de coordonate al lumii este identic cu sistemul de coordonate al camerei anteroioare stângi. Matricea parametrilor extrinseci pentru camera anteroioara dreapta depinde de distanța b dintre cele două camere (baseline) și va fi:

$$A_{R1}^L = \begin{bmatrix} 1 & 0 & 0 & -b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (50)$$

Matricea parametrilor extrinseci pentru camera curentă stângă va fi:

$$A_{L2}^L = [R \ T] \quad (51)$$

R este rotația folosită pentru a transforma din sistemul de coordonate al camerei anteroioare stângi (sistemul lumii) în sistemul de coordonate al camerei curente stângi. T este poziția centrului sistemului de

coordonate al lumii în sistemul de coordonate camerei curente stângi. Folosind acești parametrii extrinseci pentru cele trei imagine se construiește un tensor trifocal: $S_i^L(R, T)$, $i = \overline{1,3}$ folosind modelul matematic Anexa 11 - Tensorul trifocal. Se observă ca acesta depinde de parametrii de rotația și de translația efectuate între cele două momente de timp.

2. Considerăm o dreapta în imaginea anterioară dreapta l_{R1}^i care trece prin punctul p_{R1}^i . Un exemplu de astfel de dreapta este $l_{R1}^i = \begin{bmatrix} 1 \\ 1 \\ -x_{R1}^i - y_{R1}^i \end{bmatrix}$.

Folosind această dreapta și ecuația de transfer pe baza tensorului trifocal obținem:

$$\bar{p}_{L2}^i(R, T) = \begin{bmatrix} \bar{x}_{L2}^i(R, T) \\ \bar{y}_{L2}^i(R, T) \\ 1 \end{bmatrix} = \\ = [S_i^L(R, T) \cdot l_{R1}^i \quad S_i^L(R, T) \cdot l_{R1}^i \quad S_i^L(R, T) \cdot l_{R1}^i] \cdot p_{L1}^i \quad (52)$$

3. Definim un al doilea tensor trifocal între imaginea anterioară dreapta (prima proiecție), imaginea anterioară stângă (a doua proiecție), imaginea curentă dreapta (a treia proiecție). Vom considera că matricea parametrilor extrinseci pentru imaginea anterioară dreapta este $A_{R1}^R = [I \quad 0]$, ceea ce înseamnă că sistemul de coordonate al lumii este identic cu sistemul de coordonate al camerei anterioare drepte. Matricea parametrilor extrinseci pentru camera anterioară stângă va fi:

$$A_{L1}^R = \begin{bmatrix} 1 & 0 & 0 & b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (53)$$

Matricea parametrilor extrinseci pentru camera curentă dreapta va fi:

$$A_{R2}^R = [R \quad T] \quad (54)$$

R este matricea de rotație folosită pentru a transforma din sistemul de coordonate al camerei anterioare drepte (sistemul lumii) în sistemul de coordonate al camerei curente drepte iar T este poziția centrului

sistemului de coordonate al lumii în sistemul de coordonate camerei curente drepte. Folosind acești parametrii extrinseci pentru cele trei imagine se construiește un tensor trifocal: $S_i^R(R, T)$, $i = \overline{1,3}$ pe baza modelului matematic din Anexa 11 - Tensorul trifocal. Se observă că acesta depinde de parametrii de rotație și de translația efectuate între cele două momente de timp.

4. Considerăm o dreapta în imaginea anteroioara stângă l_{L1}^i care trece prin punctul p_{L1}^i . Un exemplu de astfel de dreapta este $l_{L1}^i = \begin{bmatrix} 1 \\ 1 \\ -x_{L1}^i - y_{L1}^i \end{bmatrix}$.

5. Folosind această dreapta și ecuația de transfer pe baza tensorului trifocal obținem:

$$\bar{p}_{R2}^i(R, T) = \begin{bmatrix} \bar{x}_{R2}^i(R, T) \\ \bar{y}_{R2}^i(R, T) \\ 1 \end{bmatrix} = \\ = [S_i^R(R, T) \cdot l_{L1}^i \quad S_i^R(R, T) \cdot l_{L1}^i \quad S_i^R(R, T) \cdot l_{L1}^i] \cdot p_{R1}^i \quad (55)$$

6. Considerăm următoarea funcție:

$$f(R, T) = \sum_{i=0}^n (\|\bar{p}_{L2}^i(R, T) - p_{L2}^i\|^2 + \|\bar{p}_{R2}^i(R, T) - p_{R2}^i\|^2) \quad (56)$$

Considerând R, T parametrii acesteia, funcția de mai sus își va atinge minimul atunci când valoarea parametrilor este egală cu valoarea rotației și translației efectuate de camera stereo de la momentul t_1 la momentul t_2 .

Pentru a îmbunătăți timpul de procesare se poate considera următoarea funcție:

$$g(R, T) = \sum_{i=0}^n \|\bar{p}_{L2}^i(R, T) - p_{L2}^i\|^2 \quad (57)$$

Practic, folosim cele două imagini anteroioare și imaginea curentă stângă pentru calculul funcției de minimizat, ignorând imaginea dreaptă pentru un timp de execuție mai mic.

1.2.3. Odometrie vizuala pe baza minimizării erorii de reproiecție în planul imagine

Continuând cu aceleași notații, avem P_1^i coordonatele punctului 3D în sistemul de coordonate al camerei anterioare stângi și P_2^i reprezintă coordonatele aceluiași punct 3D în sistemul de coordonate al camerei curente stângi. Transformarea punctului P_1^i în sistemul de coordonate al camerei curente, depinde de parametrii de translație și de parametrii de rotație astfel:

$$\bar{P}_2^i(R, T) = \begin{bmatrix} \bar{X}_2^i(R, T) \\ \bar{Y}_2^i(R, T) \\ \bar{Z}_2^i(R, T) \end{bmatrix} = R \cdot P_1^i + T \quad (58)$$

Prin împărțirea la distanța longitudinală obținem coordonatele normalize în planul imagine al camerei stângi, depinzând de parametrii de rotație și de vectorul de translație:

$$\bar{p}_{L2}^i(R, T) = \begin{bmatrix} \bar{x}_{L2}^i(R, T) \\ \bar{y}_{L2}^i(R, T) \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\bar{X}_2^i(R, T)}{\bar{Z}_2^i(R, T)} \\ \frac{\bar{Y}_2^i(R, T)}{\bar{Z}_2^i(R, T)} \\ 1 \end{bmatrix} \quad (59)$$

De asemenea se pot obține coordonatele în pixeli, în planul imagine al camerei stângi ca și o funcție având ca și parametrii matricea de rotație și vectorul de translație:

$$\bar{q}_{L2}^i(R, T) = K \cdot \bar{p}_{L2}^i(R, T) \quad (60)$$

Folosind coordonatele normalize, funcție de minimizat pentru a obține parametrii de mișcare va fi:

$$f(R, T) = \sum_{i=0}^n \| \bar{p}_{L2}^i(R, T) - p_{L2}^i \|^2 \quad (61)$$

Folosind coordonatele în pixeli funcția de minimizat va avea următoarea formă:

$$g(R, T) = \sum_{i=0}^n \| \bar{q}_{L2}^i(R, T) - q_{L2}^i \|^2 \quad (62)$$

În ambele cazuri funcția își atinge valoarea minima atunci când R și T iau valoarea rotației respectiv translației dintre sistemul de coordonate al imaginii anterioare stângi și sistemul de coordonate al imaginii curente stângi.

1.3. Metoda liniară de odometrie vizuală

În general matricea de rotație R se exprima folosind funcții trigonometrice și cele trei unghiuri Euler: roll, pitch, yaw. Dacă rotațiile care apar sunt mici atunci matricea de rotație se poate aproxima astfel:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} 1 & -\gamma & -\beta \\ \gamma & 1 & -\alpha \\ \beta & \alpha & 1 \end{bmatrix} \quad (63)$$

Semnificația unghiurilor este următoarea: α reprezintă unghiul pitch, β reprezintă unghiul yaw, γ reprezintă unghiul roll. Aproximația din (63) este motivată de liniarizarea funcțiilor trigonometrice care intră în componenta matricii de rotație conform (64) și (65), aproximând $\sin(\alpha) \approx \alpha$ și $\cos(\alpha) \approx 1$. Referim Anexa 6 pentru o demonstrație detaliată a acestei aproximări.

$$\sin(\alpha) = \alpha - \frac{\alpha^3}{3!} + \frac{\alpha^5}{5!} - \frac{\alpha^7}{7!} + \dots \quad (64)$$

$$\cos(\alpha) = 1 - \frac{\alpha^2}{2!} + \frac{\alpha^4}{4!} - \frac{\alpha^6}{6!} + \dots \quad (65)$$

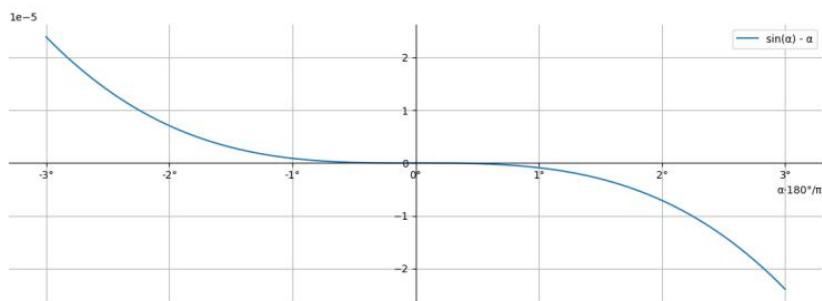


Figura 32 $\sin(\alpha) - \alpha$ for $\alpha \in [-3^\circ, 3^\circ]$

Intuitiv, Figura 32 reprezintă grafic diferența dintre $\sin(\alpha)$ și α în intervalul $[-3^\circ, 3^\circ]$, interval în care este plauzibil să avem în mod normal mișcarea dintre imagini consecutive preluate dintr-o cameră instalată într-un autovehicul. Observăm o diferență de ordinul 10^{-5} , care este suficient de mică comparativ cu precizia de estimare a odometriei vizuale.

Figura 33 arată norma Forbenius a diferenței dintre matricea de rotație și aproximarea acesteia din ecuația (63). Unghiurile pitch (α) și yaw (β) sunt variate în intervalul $[-3^\circ, 3^\circ]$ iar unghiul roll (γ) este păstrat constant la 0° respectiv 3° . Observăm o diferență de ordinul 10^{-4} (pentru $\gamma = 0^\circ$) respectiv 10^{-3} (pentru $\gamma = 3^\circ$), și de asemenea faptul că norma Forbenius este mare atunci când două sau mai multe dintre cele trei unghiuri au valori mari simultan, lucru rar întâlnit în practică.

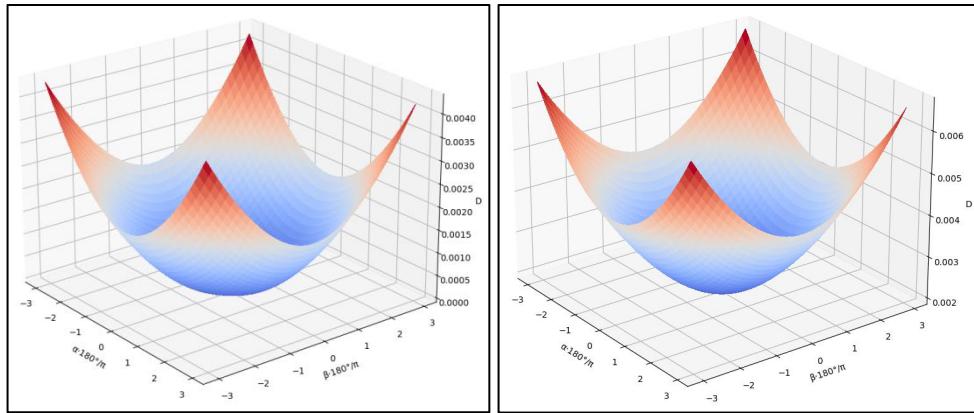


Figura 33 Norma Forbenius a diferenței dintre matricea de rotație și aproximarea acesteia. Parametrii α și β sunt variați în intervalul $[-3^\circ, 3^\circ]$ iar γ este păstrat constant la 0° (stânga) respectiv 3° (dreapta)

Matricea de translație o notam cu: $T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$. Notam cu $\delta_1^i = x_{L1}^i - x_{R1}^i$,

$\delta_2^i = x_{L2}^i - x_{R2}^i$, $d_1^i = u_{L1}^i - u_{R1}^i$, $d_2^i = u_{L2}^i - u_{R2}^i$ disparitățile în coordonate normalize și respectiv în pixeli. De asemenea presupunem ca pixelii sunt pătratici: $f_x = f_y$, presupunere validă în cardul sistemului stereo în care se va face implementarea practică a acestui model

teoretic, și în general valabila după etapa de eliminare a distorsiunilor și de rectificare a imaginilor inițial obținute de la camere.

Folosind aceste notații reconstrucțiile 3D în sistemul de coordonate al camerei anterioare și sistemul de coordonate al camerei curente vor deveni:

$$P_1^i = \begin{bmatrix} X_1^i \\ Y_1^i \\ Z_1^i \end{bmatrix} = \begin{bmatrix} \frac{b \cdot (u_{L1}^i - u_0)}{d_1^i} \\ \frac{b \cdot (v_{L1}^i - v_0)}{d_1^i} \\ \frac{f_x \cdot b}{d_1^i} \end{bmatrix} = \begin{bmatrix} \frac{b \cdot x_{L1}^i}{\delta_1^i} \\ \frac{b \cdot y_{L1}^i}{\delta_1^i} \\ \frac{f \cdot b}{\delta_1^i} \end{bmatrix} = \begin{bmatrix} \frac{Z_1^i \cdot x_{L1}^i}{f} \\ \frac{Z_1^i \cdot y_{L1}^i}{f} \\ \frac{f \cdot b}{\delta_1^i} \end{bmatrix} \quad (66)$$

$$P_2^i = \begin{bmatrix} X_2^i \\ Y_2^i \\ Z_2^i \end{bmatrix} = \begin{bmatrix} \frac{b \cdot (u_{L2}^i - u_0)}{d_2^i} \\ \frac{b \cdot (v_{L2}^i - v_0)}{d_2^i} \\ \frac{f_x \cdot b}{d_2^i} \end{bmatrix} = \begin{bmatrix} \frac{b \cdot x_{L2}^i}{\delta_2^i} \\ \frac{b \cdot y_{L2}^i}{\delta_2^i} \\ \frac{f \cdot b}{\delta_2^i} \end{bmatrix} = \begin{bmatrix} \frac{Z_2^i \cdot x_{L2}^i}{f} \\ \frac{Z_2^i \cdot y_{L2}^i}{f} \\ \frac{f \cdot b}{\delta_2^i} \end{bmatrix} \quad (67)$$

Se observă că am adăugat și o valoare $f = 1$ pentru varianta de calcul folosind coordonate normalize, astfel încât să avem o reprezentare a punctului 3D pe baza coordonatelor normalize similară cu reprezentarea pe baza coordonatelor imagine, cu scopul de a avea în continuare un set de formule matematice similar pentru cele două moduri echivalente de reprezentare. În continuare vom deduce modelul matematic pentru determinarea parametrilor de rotație și translație în coordonate normalize, urmând ca modelul pentru coordonate în pixeli să se deducă prin înlocuirea valorilor lui $f, \delta_1^i, \delta_2^i, x_{L2}^i, y_{L2}^i$ cu valorile $f_x, d_1^i, d_2^i, u_{L1}^i - u_0, v_{L1}^i - v_0$, lucru posibil datorită similarității matematice observabile în ecuațiile de mai sus. Folosind în continuare aceste notații rezultă:

$$\bar{P}_2^i(R, T) = \begin{bmatrix} \bar{X}_2^i(R, T) \\ \bar{Y}_2^i(R, T) \\ \bar{Z}_2^i(R, T) \end{bmatrix} = R \cdot P_1^i + T = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \frac{b}{\delta_1^i} \cdot \begin{bmatrix} x_{L1}^i \\ y_{L1}^i \\ f \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} =$$

$$\begin{aligned}
&= \frac{b}{\delta_1^i} \cdot \begin{bmatrix} r_{11} \cdot x_{L1}^i + r_{12} \cdot y_{L1}^i + r_{13} \cdot f \\ r_{21} \cdot x_{L1}^i + r_{22} \cdot y_{L1}^i + r_{23} \cdot f \\ r_{31} \cdot x_{L1}^i + r_{32} \cdot y_{L1}^i + r_{33} \cdot f \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \\
&= \frac{b}{\delta_1^i} \cdot \begin{bmatrix} r_{11} \cdot x_{L1}^i + r_{12} \cdot y_{L1}^i + r_{13} \cdot f + \frac{t_x \cdot \delta_1^i}{b} \\ r_{21} \cdot x_{L1}^i + r_{22} \cdot y_{L1}^i + r_{23} \cdot f + \frac{t_y \cdot \delta_1^i}{b} \\ r_{31} \cdot x_{L1}^i + r_{32} \cdot y_{L1}^i + r_{33} \cdot f + \frac{t_z \cdot \delta_1^i}{b} \end{bmatrix} = \\
&= \frac{b}{\delta_1^i} \cdot \begin{bmatrix} x_{L1}^i - \gamma \cdot y_{L1}^i - \beta \cdot f + \frac{t_x \cdot \delta_1^i}{b} \\ \gamma \cdot x_{L1}^i + y_{L1}^i - \alpha \cdot f + \frac{t_y \cdot \delta_1^i}{b} \\ \beta \cdot x_{L1}^i + \alpha \cdot y_{L1}^i + f + \frac{t_z \cdot \delta_1^i}{b} \end{bmatrix} \quad (68)
\end{aligned}$$

De asemenea avem următoarele egalități între coordonatele normalize și coordonatele 3D:

$$x_{L2}^i = \frac{f \cdot x_{L1}^i}{z_{L1}^i}, y_{L2}^i = \frac{f \cdot y_{L1}^i}{z_{L1}^i} \text{ și } \delta_2^i = \frac{f \cdot b}{z_{L1}^i} \quad (69)$$

Acestea sunt valabile desigur și de varianța parametrică:

$$\bar{x}_{L2}^i(R, T) = \frac{f \cdot \bar{x}_2^i(R, T)}{z_{L1}^i(R, T)} = f \cdot \frac{r_{11} \cdot x_{L1}^i + r_{12} \cdot y_{L1}^i + r_{13} \cdot f + \frac{t_x \cdot \delta_1^i}{b}}{r_{31} \cdot x_{L1}^i + r_{32} \cdot y_{L1}^i + r_{33} \cdot f + \frac{t_z \cdot \delta_1^i}{b}} = f \cdot \frac{\bar{x}_{L1}^i - \gamma \cdot y_{L1}^i - \beta \cdot f + \frac{t_x \cdot \delta_1^i}{b}}{\beta \cdot x_{L1}^i + \alpha \cdot y_{L1}^i + f + \frac{t_z \cdot \delta_1^i}{b}} \quad (70)$$

$$\bar{y}_{L2}^i(R, T) = \frac{f \cdot \bar{y}_2^i(R, T)}{z_{L1}^i(R, T)} = f \cdot \frac{r_{21} \cdot x_{L1}^i + r_{22} \cdot y_{L1}^i + r_{23} \cdot f + \frac{t_y \cdot \delta_1^i}{b}}{r_{31} \cdot x_{L1}^i + r_{32} \cdot y_{L1}^i + r_{33} \cdot f + \frac{t_z \cdot \delta_1^i}{b}} = f \cdot \frac{\gamma \cdot x_{L1}^i + y_{L1}^i - \alpha \cdot f + \frac{t_y \cdot \delta_1^i}{b}}{\beta \cdot x_{L1}^i + \alpha \cdot y_{L1}^i + f + \frac{t_z \cdot \delta_1^i}{b}} \quad (71)$$

$$\begin{aligned}
\bar{\delta}_2^i(R, T) &= \frac{f \cdot b}{z_{L1}^i(R, T)} = \frac{f \cdot b}{\frac{b}{\delta_1^i} \left(r_{31} \cdot x_{L1}^i + r_{32} \cdot y_{L1}^i + r_{33} \cdot f + \frac{t_z \cdot \delta_1^i}{b} \right)} = \frac{f \cdot \delta_1^i}{r_{31} \cdot x_{L1}^i + r_{32} \cdot y_{L1}^i + r_{33} \cdot f + \frac{t_z \cdot \delta_1^i}{b}} = \\
&= \frac{f \cdot \delta_1^i}{\beta \cdot x_{L1}^i + \alpha \cdot y_{L1}^i + f + \frac{t_z \cdot \delta_1^i}{b}} \quad (72)
\end{aligned}$$

Se observă că folosind aceste ecuații putem prezice poziția $(\bar{x}_{L2}^i(R, T), \bar{y}_{L2}^i(R, T))$ și disparitatea $\delta_2^i(R, T)$ în imaginea curentă stângă

a unui pixel din imaginea anterioara stângă (x_{L1}^i, y_{L1}^i) cunoscând disparitatea acestuia δ_1^i și transformata dintre sistemul de coordonate al camerei anterioare și sistemul de coordonate al camerei curente (adică mișcarea camerei). De asemenea este important de remarcat că varianta cu r_{ij} , $i = \overline{1,3}$, $j = \overline{1,3}$, a formulelor matematice este valabilă în general și nu doar pentru varianta cu rotația aproximată.

Pentru a determina valorile parametrilor de rotație și translație compunem următorul sistem de ecuații având necunoscutele $\alpha, \beta, \gamma, t_x, t_y, t_z$:

$$\begin{cases} x_{L2}^i = \bar{x}_{L2}^i(R, T) \\ y_{L2}^i = \bar{y}_{L2}^i(R, T) \\ \delta_2^i = \bar{\delta}_2^i(R, T) \end{cases} \Leftrightarrow \begin{cases} x_{L2}^i = f \cdot \frac{x_{L1}^i - \gamma \cdot y_{L1}^i - \beta \cdot f + \frac{t_x \cdot \delta_1^i}{b}}{\beta \cdot x_{L1}^i + \alpha \cdot y_{L1}^i + f + \frac{t_z \cdot \delta_1^i}{b}} \\ y_{L2}^i = f \cdot \frac{\gamma \cdot x_{L1}^i + y_{L1}^i - \alpha \cdot f + \frac{t_y \cdot \delta_1^i}{b}}{\beta \cdot x_{L1}^i + \alpha \cdot y_{L1}^i + f + \frac{t_z \cdot \delta_1^i}{b}} \\ \delta_2^i = \frac{f \cdot \delta_1^i}{\beta \cdot x_{L1}^i + \alpha \cdot y_{L1}^i + f + \frac{t_z \cdot \delta_1^i}{b}} \end{cases} \quad (73)$$

\Leftrightarrow

$$\begin{cases} \alpha \cdot y_{L1}^i \cdot x_{L2}^i + \beta \cdot (x_{L1}^i \cdot x_{L2}^i + f^2) + \gamma \cdot f \cdot y_{L1}^i + t_x \cdot \left(\frac{-f \cdot \delta_1^i}{b}\right) + t_z \cdot \frac{x_{L2}^i \cdot \delta_1^i}{b} = \\ = f \cdot (x_{L1}^i - x_{L2}^i) \\ \alpha \cdot (y_{L1}^i \cdot y_{L2}^i + f^2) + \beta \cdot x_{L1}^i \cdot y_{L2}^i + \gamma \cdot (-f \cdot x_{L1}^i) + t_y \cdot \left(\frac{-f \cdot \delta_1^i}{b}\right) + t_z \cdot \frac{y_{L2}^i \cdot \delta_1^i}{b} = \\ = f \cdot (y_{L1}^i - y_{L2}^i) \\ \alpha \cdot y_{L1}^i \cdot \delta_2^i + \beta \cdot x_{L1}^i \cdot \delta_2^i + t_z \cdot \frac{\delta_2^i \cdot \delta_1^i}{b} = f \cdot (\delta_1^i - \delta_2^i) \end{cases} \quad (74)$$

Rezolvând sistemul de mai sus pentru mai multe puncte diferite (cel puțin 2 valori diferite alături de i) se determină valorile necunoscute $\alpha, \beta, \gamma, t_x, t_y, t_z$. Folosind coordonate normalize valoarea lui f va fi 1, și ecuațiile se pot simplifica. Un set similar de ecuații se pot deduce în coordonate pixeli prin înlocuirea valorilor $f, \delta_1^i, \delta_2^i, x_{L2}^i, y_{L2}^i$ cu valorile $f_x, d_1^i, d_2^i, u_{L1}^i - u_0, v_{L1}^i - v_0$ în toate ecuațiile de mai sus. După această înlocuire vom obține următoarele ecuații de predicție:

$$\begin{aligned} \bar{u}_{L2}^i(R, T) &= f_x \cdot \frac{r_{11} \cdot (u_{L1}^i - u_0) + r_{12} \cdot (v_{L1}^i - v_0) + r_{13} \cdot f_x + \frac{t_x \cdot d_1^i}{b}}{r_{31} \cdot (u_{L1}^i - u_0) + r_{32} \cdot (v_{L1}^i - v_0) + r_{33} \cdot f_x + \frac{t_z \cdot d_1^i}{b}} + u_0 = \\ &= f_x \cdot \frac{(u_{L1}^i - u_0) - \gamma \cdot (v_{L1}^i - v_0) - \beta \cdot f_x + \frac{t_x \cdot d_1^i}{b}}{\beta \cdot (u_{L1}^i - u_0) + \alpha \cdot (v_{L1}^i - v_0) + f_x + \frac{t_z \cdot d_1^i}{b}} + u_0 \end{aligned} \quad (75)$$

$$\begin{aligned} \bar{v}_{L2}^i(R, T) &= f_x \cdot \frac{r_{21} \cdot (u_{L1}^i - u_0) + r_{22} \cdot (v_{L1}^i - v_0) + r_{23} \cdot f_x + \frac{t_y \cdot d_1^i}{b}}{r_{31} \cdot (u_{L1}^i - u_0) + r_{32} \cdot (v_{L1}^i - v_0) + r_{33} \cdot f_x + \frac{t_z \cdot d_1^i}{b}} + v_0 = \\ &= f_x \cdot \frac{\gamma \cdot (u_{L1}^i - u_0) + (v_{L1}^i - v_0) - \alpha \cdot f_x + \frac{t_y \cdot d_1^i}{b}}{\beta \cdot (u_{L1}^i - u_0) + \alpha \cdot (v_{L1}^i - v_0) + f_x + \frac{t_z \cdot d_1^i}{b}} + v_0 \end{aligned} \quad (76)$$

$$\begin{aligned} \bar{d}_2^i(R, T) &= \frac{f_x \cdot d_1^i}{r_{31} \cdot (u_{L1}^i - u_0) + r_{32} \cdot (v_{L1}^i - v_0) + r_{33} \cdot f_x + \frac{t_z \cdot d_1^i}{b}} \\ &= \frac{f_x \cdot d_1^i}{\beta \cdot (u_{L1}^i - u_0) + \alpha \cdot (v_{L1}^i - v_0) + f_x + \frac{t_z \cdot d_1^i}{b}} \end{aligned} \quad (77)$$

În mod similar, dacă lucrăm în pixeli atunci ecuațiile sistemului liniar vor arăta astfel:

$$\left\{ \begin{array}{l} \alpha \cdot (v_{L1}^i - v_0) \cdot (u_{L2}^i - u_0) + \beta \cdot ((u_{L1}^i - u_0) \cdot (u_{L2}^i - u_0) + f_x^2) + \gamma \cdot f_x \cdot (v_{L1}^i - v_0) + \\ t_x \cdot \left(\frac{-f_x \cdot d_1^i}{b} \right) + t_z \cdot \frac{(u_{L2}^i - u_0) \cdot d_1^i}{b} = f_x \cdot (u_{L1}^i - u_{L2}^i) \\ \alpha \cdot ((v_{L1}^i - v_0) \cdot (v_{L2}^i - v_0) + f_x^2) + \beta \cdot (u_{L1}^i - u_0) \cdot (v_{L2}^i - v_0) + \gamma \cdot (-f_x \cdot (u_{L1}^i - u_0)) + \\ t_y \cdot \left(\frac{-f_x \cdot d_1^i}{b} \right) + t_z \cdot \frac{(v_{L2}^i - v_0) \cdot d_1^i}{b} = f_x \cdot (v_{L1}^i - v_{L2}^i) \\ \alpha \cdot (v_{L1}^i - v_0) \cdot d_2^i + \beta \cdot (u_{L1}^i - u_0) \cdot d_2^i + t_z \cdot \frac{d_2^i \cdot d_1^i}{b} = f_x \cdot (d_1^i - d_2^i) \end{array} \right. \quad (78)$$

Pentru anumiți pași algoritmici va fi util să determinam coordonatele în pixeli în imaginea anterioara în funcție de coordonatele în pixeli din imaginea curentă. Păstrând aceleasi semnificații pentru matricea de rotație și pentru matricea de translație vom avea următoarele relații:

$$\bar{P}_2^i = \begin{bmatrix} X_2^i \\ Y_2^i \\ Z_2^i \end{bmatrix} = R \cdot P_1^i(R, T) + T \Leftrightarrow$$

$$P_1^i(R, T) = \begin{bmatrix} X_1^i(R, T) \\ Y_1^i(R, T) \\ Z_1^i(R, T) \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^{-1} \cdot \left(\frac{b}{\delta_2^i} \cdot \begin{bmatrix} x_{L2}^i \\ y_{L2}^i \\ f \end{bmatrix} - \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \right) =$$

$$= \frac{b}{\delta_2^i} \cdot \begin{bmatrix} r'_{11} \cdot x_{L2}^i + r'_{12} \cdot y_{L2}^i + r'_{13} \cdot f \\ r'_{21} \cdot x_{L2}^i + r'_{22} \cdot y_{L2}^i + r'_{23} \cdot f \\ r'_{31} \cdot x_{L2}^i + r'_{32} \cdot y_{L2}^i + r'_{33} \cdot f \end{bmatrix} - \begin{bmatrix} r'_{11} \cdot t_x + r'_{12} \cdot t_y + r'_{13} \cdot t_z \\ r'_{21} \cdot t_x + r'_{22} \cdot t_y + r'_{23} \cdot t_z \\ r'_{31} \cdot t_x + r'_{32} \cdot t_y + r'_{33} \cdot t_z \end{bmatrix} \Leftrightarrow$$

$$\bar{x}_{L1}^i(R, T) = \frac{f \cdot X_1^i(R, T)}{Z_1^i(R, T)} = f \cdot \frac{(r'_{11} \cdot x_{L2}^i + r'_{12} \cdot y_{L2}^i + r'_{13} \cdot f) - \frac{\delta_2^i}{b} (r'_{11} \cdot t_x + r'_{12} \cdot t_y + r'_{13} \cdot t_z)}{(r'_{31} \cdot x_{L2}^i + r'_{32} \cdot y_{L2}^i + r'_{33} \cdot f) - \frac{\delta_2^i}{b} (r'_{31} \cdot t_x + r'_{32} \cdot t_y + r'_{33} \cdot t_z)} \quad (79)$$

$$\bar{y}_{L1}^i(R, T) = \frac{f \cdot Y_1^i(R, T)}{Z_1^i(R, T)} = f \cdot \frac{(r'_{21} \cdot x_{L2}^i + r'_{22} \cdot y_{L2}^i + r'_{23} \cdot f) - \frac{\delta_2^i}{b} (r'_{21} \cdot t_x + r'_{22} \cdot t_y + r'_{23} \cdot t_z)}{(r'_{31} \cdot x_{L2}^i + r'_{32} \cdot y_{L2}^i + r'_{33} \cdot f) - \frac{\delta_2^i}{b} (r'_{31} \cdot t_x + r'_{32} \cdot t_y + r'_{33} \cdot t_z)} \quad (80)$$

$$\bar{\delta}_1^i(R, T) = \frac{f \cdot b}{Z_1^i(R, T)} = \frac{f \cdot \delta_2^i}{(r'_{31} \cdot x_{L2}^i + r'_{32} \cdot y_{L2}^i + r'_{33} \cdot f) - \frac{\delta_2^i}{b} (r'_{31} \cdot t_x + r'_{32} \cdot t_y + r'_{33} \cdot t_z)} \quad (81)$$

Am notat cu r'_{ij} , $i = \overline{1,3}$, $j = \overline{1,3}$ elementele matricii R^{-1} . La fel ca și în cazul transformării directe, se pot înlocui valorile f , δ_1^i , δ_2^i , x_{L2}^i , y_{L2}^i cu valorile f_x , d_1^i , d_2^i , $u_{L1}^i - u_0$, $v_{L1}^i - v_0$ obținând:

$$\bar{u}_{L1}^i(R, T) - u_0 = \frac{f_x \cdot X_1^i(R, T)}{Z_1^i(R, T)} = f_x \cdot \frac{(r'_{11} \cdot (u_{L2}^i - u_0) + r'_{12} \cdot (v_{L2}^i - v_0) + r'_{13} \cdot f_x) - \frac{d_2^i}{b} (r'_{11} \cdot t_x + r'_{12} \cdot t_y + r'_{13} \cdot t_z)}{(r'_{31} \cdot (u_{L2}^i - u_0) + r'_{32} \cdot (v_{L2}^i - v_0) + r'_{33} \cdot f_x) - \frac{d_2^i}{b} (r'_{31} \cdot t_x + r'_{32} \cdot t_y + r'_{33} \cdot t_z)} \quad (82)$$

$$\bar{v}_{L1}^i(R, T) - v_0 = \frac{f_x \cdot Y_1^i(R, T)}{Z_1^i(R, T)} = f_x \cdot \frac{(r'_{21} \cdot (u_{L2}^i - u_0) + r'_{22} \cdot (v_{L2}^i - v_0) + r'_{23} \cdot f_x) - \frac{d_2^i}{b} (r'_{21} \cdot t_x + r'_{22} \cdot t_y + r'_{23} \cdot t_z)}{(r'_{31} \cdot (u_{L2}^i - u_0) + r'_{32} \cdot (v_{L2}^i - v_0) + r'_{33} \cdot f_x) - \frac{d_2^i}{b} (r'_{31} \cdot t_x + r'_{32} \cdot t_y + r'_{33} \cdot t_z)} \quad (83)$$

$$\bar{\delta}_1^i(R, T) = \frac{f_x \cdot b}{Z_1^i(R, T)} = \frac{f_x \cdot d_2^i}{(r'_{31} \cdot (x_{L2}^i - u_0) + r'_{32} \cdot (y_{L2}^i - v_0) + r'_{33} \cdot f_x) - \frac{d_2^i}{b} (r'_{31} \cdot t_x + r'_{32} \cdot t_y + r'_{33} \cdot t_z)} \quad (84)$$

1.4. Un model de eroare pentru metoda liniară de odometrie vizuală

Rezolvarea prin metoda celor mai mici pătrate (least squares) a sistemului supradeterminat de ecuații de mai sus este echivalent cu determinarea minimului următoarei funcții:

$$F(x_{L1}^i, y_{L1}^i, x_{L2}^i, y_{L2}^i, \delta_1^i, \alpha, \beta, \gamma, t_x, t_y, t_z) = \sum_{i=0}^{N-1} \left((\alpha \cdot y_{L1}^i \cdot x_{L2}^i + \beta \cdot (x_{L1}^i \cdot x_{L2}^i + f^2) + \gamma \cdot f \cdot y_{L1}^i + t_x \cdot \left(\frac{-f \cdot \delta_1^i}{b}\right) + t_z \cdot \frac{x_{L2}^i \cdot \delta_1^i}{b} - f \cdot (x_{L1}^i - x_{L2}^i))^2 + (\alpha \cdot (y_{L1}^i \cdot y_{L2}^i + f^2) + \beta \cdot x_{L1}^i \cdot y_{L2}^i + \gamma \cdot (-f \cdot x_{L1}^i) + t_y \cdot \left(\frac{-f \cdot \delta_1^i}{b}\right) + t_z \cdot \frac{y_{L2}^i \cdot \delta_1^i}{b} - f \cdot (y_{L1}^i - y_{L2}^i))^2 \right) \quad (85)$$

Notam cu $\theta = [\theta_i, i = 0, N-1] = [\alpha \ \beta \ \gamma \ t_x \ t_y \ t_z]$, vectorul parametrilor de estimat la fiecare pereche de imagini consecutive. De asemenea, notam cu $p = [p_i, i = 0, N-1]$, unde $p_i = [p_{i,j}, j = 0, 4] = [x_{L1}^i \ x_{L2}^i \ y_{L1}^i \ y_{L2}^i \ \delta_1^i]$, vectorul parametrilor de intrare pentru fiecare pereche de imagini consecutive, variabili de la o imagine la cealaltă. Cu aceste notări problema odometriei vizuale se transformă în determinarea valorii θ care minimizează $F(\theta, p)$.

Pentru simplitatea expresiilor vom folosi armatoarele notări pentru primul termen din suma de mai sus:

$$\begin{aligned} a_0(p_i) &= y_{L1}^i \cdot x_{L2}^i, \quad a_1(p_i) = x_{L1}^i \cdot x_{L2}^i + f^2, \quad a_2(p_i) = f \cdot y_{L1}^i \\ a_3(p_i) &= \frac{-f \cdot \delta_1^i}{b}, \quad a_4(p_i) = 0, \quad a_5(p_i) = \frac{x_{L2}^i \cdot \delta_1^i}{b} \\ a_6(p_i) &= -f \cdot (x_{L1}^i - x_{L2}^i) \end{aligned} \quad (86)$$

Pentru al doilea termen vom avea:

$$\begin{aligned} b_0(p_i) &= y_{L1}^i \cdot y_{L2}^i + f^2, \quad b_1(p_i) = x_{L1}^i \cdot y_{L2}^i, \quad b_2(p_i) = -f \cdot x_{L1}^i \\ b_3(p_i) &= 0, \quad b_4(p_i) = \frac{-f \cdot \delta_1^i}{b}, \quad b_5(p_i) = \frac{y_{L2}^i \cdot \delta_1^i}{b} \end{aligned}$$

$$b_6(p_i) = -f \cdot (y_{L1}^i - y_{L2}^i) \quad (87)$$

Folosind notațiile de mai sus, o varianta mai compactă pentru $F(\theta, p)$ va fi:

$$F(\theta, p) = \sum_{i=0}^{N-1} \left(\left(\sum_{j=0}^5 \theta_j \cdot a_j(p_i) + a_7(p_i) \right)^2 + \left(\sum_{j=0}^5 \theta_j \cdot b_j(p_i) + b_7(p_i) \right)^2 \right)$$

$$\frac{\partial F(\theta, p)}{\partial \theta_k} = 2 \cdot \sum_{i=0}^{N-1} \left(a_k(p_i) \cdot \left(\sum_{j=0}^5 \theta_j \cdot a_j(p_i) + a_7(p_i) \right) + b_k(p_i) \cdot \left(\sum_{j=0}^5 \theta_j \cdot b_j(p_i) + b_7(p_i) \right) \right) \quad (88)$$

Derivatele de ordinul 2 vor fi:

$$\frac{\partial^2 F(\theta, p)}{\partial \theta_k \partial \theta_l} = 2 \cdot \sum_{i=0}^{N-1} \left(a_k(p_i) \cdot a_l(p_i) + b_k(p_i) \cdot b_l(p_i) \right) \quad (89)$$

$$\begin{aligned} \frac{\partial^2 F(\theta, p)}{\partial \theta_k \partial p_{i,l}} &= 2 \cdot \frac{\partial a_k(p_i)}{\partial p_{i,l}} \cdot \left(\sum_{j=0}^5 \theta_j \cdot a_j(p_i) + a_6(p_i) \right) + 2 \cdot a_k(p_i) \cdot \\ &\left(\sum_{j=0}^5 \theta_j \cdot \frac{\partial a_j(p_i)}{\partial p_{i,l}} + \frac{\partial a_6(p_i)}{\partial p_{i,l}} \right) + 2 \cdot \frac{\partial b_k(p_i)}{\partial p_{i,l}} \cdot \left(\sum_{j=0}^5 \theta_j \cdot b_j(p_i) + b_6(p_i) \right) + 2 \cdot \\ &b_k(p_i) \cdot \left(\sum_{j=0}^5 \theta_j \cdot \frac{\partial b_j(p_i)}{\partial p_{i,l}} + \frac{\partial b_6(p_i)}{\partial p_{i,l}} \right) \end{aligned} \quad (90)$$

	$a_0(p_i)$	$a_1(p_i)$	$a_2(p_i)$	$a_3(p_i)$	$a_4(p_i)$	$a_5(p_i)$	$a_6(p_i)$
$p_{i,0}$	0	x_{L2}^i	0	0	0	0	$-f$
$p_{i,1}$	x_{L2}^i	0	f	0	0	0	0
$p_{i,2}$	y_{L1}^i	x_{L1}^i	0	0	0	$\frac{\delta_1^i}{b}$	f
$p_{i,3}$	0	0	0	0	0	0	0
$p_{i,4}$	0	0	0	$\frac{-f}{b}$	0	$\frac{x_{L2}^i}{b}$	0

Tabel 2 Derivatele parțiale $\partial a_j(p_i) / \partial p_{i,l}$

Tabel 2 și Tabel 3 indică derivatele parțiale ale funcțiilor $a_j(p_i)$ și $\partial b_j(p_i)$, folosite pentru calculul: $\frac{\partial^2 F(\theta, p)}{\partial \theta_k \partial p_{i,l}}$.

	$b_0(p_i)$	$b_1(p_i)$	$b_2(p_i)$	$b_3(p_i)$	$b_4(p_i)$	$b_5(p_i)$	$b_6(p_i)$
$p_{i,0}$	0	y_{L2}^i	$-f$	0	0	0	0
$p_{i,1}$	y_{L2}^i	0	0	0	0	0	$-f$
$p_{i,2}$	0	0	0	0	0	0	0
$p_{i,3}$	y_{L1}^i	x_{L1}^i	0	0	0	$\frac{\delta_1^i}{b}$	f
$p_{i,4}$	0	0	0	0	$\frac{-f}{b}$	$\frac{y_{L2}^i}{b}$	0

Tabel 3 Derivatele parțiale $\partial b_j(p_i) / \partial p_{i,l}$

Consideram următoarele matrici:

$$A_{k,l} = \frac{\partial^2 F(\theta, p)}{\partial \theta_k \partial \theta_l}, k = 0,5 \text{ si } l = 0,5 \quad (91)$$

$$B_{k,m} = \frac{\partial^2 F(\theta, p)}{\partial \theta_k \partial p_{i,l}} \quad k = 0,5 \text{ si } m = 5 \cdot i + l \text{ cu } i = 0, N - 1 \text{ si } l = 0,4 \quad (92)$$

Astfel conform aparatului matematic prezentat în Anexa 1 matricea de covarianta a estivarii parametrilor de mișcare va fi următoarea matrice de dimensiune 6x6:

$$\Sigma_\theta = A^{-1} \cdot B \cdot \Sigma_p \cdot B^T \cdot A^{-1} \quad (93)$$

Se observă ca matricea de covarianta depinde de coordonatele trăsăturilor $x_{L1}^i, x_{L2}^i, y_{L1}^i, y_{L2}^i$, de disparitățile acestora în imaginea anterioară δ_1^i , de distanța focală (f) și de baseline (b), dar și de estimarea valorilor unghiurilor de rotație și a vectorului de translație $\theta = [\theta_i, i = 0,5] = [\alpha \ \beta \ \gamma \ t_x \ t_y \ t_z]$.

Din considerente practice, pentru o mai bună stabilitate numerică a rezultatului folosim constantele de normalizare ρ_1 și ρ_2 astfel:

$$\Sigma_\theta = \frac{1}{\rho_1^2} \cdot \left[\left(\frac{A}{\rho_1 \cdot \rho_2} \right)^{-1} \cdot \left(\frac{B}{\rho_2} \right) \cdot \Sigma_p \cdot \left(\frac{B}{\rho_2} \right)^T \cdot \left(\frac{A}{\rho_1 \cdot \rho_2} \right)^{-1} \right] \quad (94)$$

Fără acești factori de normalizare, există șanse ca pentru situația în care avem un număr mare de trăsături să apară pierderi de precizie atunci când facem înmulțiri intermediare de matrici.

1.5. Metode mono de odometrie vizuala pe baza de corespondente temporale de trăsături

1.5.1. Estimarea rotației dintr-o singura camera

Pornind de la ecuațiile din Capitolul 1.3 și considerând doar acele puncte care sunt la îndepărtate (i.e $\delta_1^i \rightarrow 0$) obținem:

$$\bar{x}_{L2}^i(R, T) = f \cdot \frac{r_{11} \cdot x_{L1}^i + r_{12} \cdot y_{L1}^i + r_{13} \cdot f}{r_{31} \cdot x_{L1}^i + r_{32} \cdot y_{L1}^i + r_{33} \cdot f} = f \cdot \frac{x_{L1}^i - \gamma y_{L1}^i - \beta \cdot f}{\beta \cdot x_{L1}^i + \alpha \cdot y_{L1}^i + f} \quad (95)$$

$$\bar{y}_{L2}^i(R, T) = \frac{f \cdot \bar{y}_{L2}^i(R, T)}{\bar{z}_2^i(R, T)} = f \cdot \frac{r_{21} \cdot x_{L1}^i + r_{22} \cdot y_{L1}^i + r_{23} \cdot f}{r_{31} \cdot x_{L1}^i + r_{32} \cdot y_{L1}^i + r_{33} \cdot f} = f \cdot \frac{\gamma \cdot x_{L1}^i + y_{L1}^i - \alpha \cdot f}{\beta \cdot x_{L1}^i + \alpha \cdot y_{L1}^i + f} \quad (96)$$

Sistemul de ecuații care deriva din aceste relații este:

$$\begin{cases} \alpha \cdot y_{L1}^i \cdot x_{L2}^i + \beta \cdot (x_{L1}^i \cdot x_{L2}^i + f^2) + \gamma \cdot f \cdot y_{L1}^i = f \cdot (x_{L1}^i - x_{L2}^i) \\ \alpha \cdot (y_{L1}^i \cdot y_{L2}^i + f^2) + \beta \cdot x_{L1}^i \cdot y_{L2}^i + \gamma \cdot (-f \cdot x_{L1}^i) = f \cdot (y_{L1}^i - y_{L2}^i) \end{cases} \quad (97)$$

Rezolvând acest sistem de ecuații folosind doar puncte îndepărtate obținem rotația dintre două imagini succesive. De remarcat că indicele 'L' este oarecum redundant în aceasta notație deoarece rotația se determină strict pe baza coordonatelor trăsăturilor din două imagini succesive, nemaifiind nevoie de imaginea dreaptă.

Aceleași ecuații se pot exprima în coordonate imagine astfel trecând la limita cad $d_1^i \rightarrow 0$ în ecuațiile exprimate în pixeli:

$$\begin{aligned}\bar{u}_{L2}^i(R, T) &= f_x \cdot \frac{r_{11} \cdot (u_{L1}^i - u_0) + r_{12} \cdot (v_{L1}^i - v_0) + r_{13} \cdot f_x}{r_{31} \cdot (u_{L1}^i - u_0) + r_{32} \cdot (v_{L1}^i - v_0) + r_{33} \cdot f_x} + u_0 = \\ &= f_x \cdot \frac{(u_{L1}^i - u_0) - \gamma \cdot (v_{L1}^i - v_0) - \beta \cdot f_x}{\beta \cdot (u_{L1}^i - u_0) + \alpha \cdot (v_{L1}^i - v_0) + f_x} + u_0\end{aligned}\quad (98)$$

$$\begin{aligned}\bar{v}_{L2}^i(R, T) &= f_x \cdot \frac{r_{21} \cdot (u_{L1}^i - u_0) + r_{22} \cdot (v_{L1}^i - v_0) + r_{23} \cdot f_x}{r_{31} \cdot (u_{L1}^i - u_0) + r_{32} \cdot (v_{L1}^i - v_0) + r_{33} \cdot f_x} + v_0 = \\ &= f_x \cdot \frac{\gamma \cdot (u_{L1}^i - u_0) + (v_{L1}^i - v_0) - \alpha \cdot f_x}{\beta \cdot (u_{L1}^i - u_0) + \alpha \cdot (v_{L1}^i - v_0) + f_x} + v_0\end{aligned}\quad (99)$$

În coordonate imagine sistemul de ecuații pe baza căruia se poate determina matricea de rotație devine:

$$\begin{cases} \alpha \cdot (v_{L1}^i - v_0) \cdot (u_{L2}^i - u_0) + \beta \cdot ((u_{L1}^i - u_0) \cdot (u_{L2}^i - u_0) + f_x^2) + \gamma \cdot f_x \cdot (v_{L1}^i - v_0) = \\ \qquad\qquad\qquad = f_x \cdot (u_{L1}^i - u_{L2}^i) \\ \alpha \cdot ((v_{L1}^i - v_0) \cdot (v_{L2}^i - v_0) + f_x^2) + \beta \cdot (u_{L1}^i - u_0) \cdot (v_{L2}^i - v_0) + \gamma \cdot (-f_x \cdot (u_{L1}^i - u_0)) = \\ \qquad\qquad\qquad = f_x \cdot (v_{L1}^i - v_{L2}^i) \end{cases}\quad (100)$$

Formulele de calcul de mai sus au fost deduse pe baza formulelor din stereo, și prin trecerea la limită în spațiul disparităților. Acestea se pot deduce în mod alternativ fără a considera un sistem stereo. Împărțind la $\bar{Z}_2^i(R, T)$ următoarea egalitate:

$$\bar{P}_2^i(R, T) = \begin{bmatrix} \bar{X}_2^i(R, T) \\ \bar{Y}_2^i(R, T) \\ \bar{Z}_2^i(R, T) \end{bmatrix} = R \cdot \begin{bmatrix} X_1^i \\ Y_1^i \\ Z_1^i \end{bmatrix} + T \quad (101)$$

Obținem:

$$\begin{bmatrix} \bar{X}_2^i(R, T) \\ \bar{Z}_2^i(R, T) \\ \bar{Y}_2^i(R, T) \\ \bar{Z}_2^i(R, T) \\ 1 \end{bmatrix} = R \cdot \frac{Z_1^i}{\bar{Z}_2^i(R, T)} \cdot \begin{bmatrix} X_1^i \\ Z_1^i \\ Y_1^i \\ Z_1^i \\ 1 \end{bmatrix} + \frac{1}{\bar{Z}_2^i(R, T)} \cdot T \quad (102)$$

Dacă punctul este la distanță mare atunci $\bar{Z}_2^i(R, T) \rightarrow \infty$ și $\frac{1}{\bar{Z}_2^i(R, T)} \rightarrow 0$, rezultând:

$$\begin{bmatrix} \frac{x_{L2}^i(R,T)}{f} \\ \frac{y_{L2}^i(R,T)}{f} \\ 1 \end{bmatrix} \sim R \cdot \begin{bmatrix} \frac{x_{L1}^i}{f} \\ \frac{y_{L1}^i}{f} \\ 1 \end{bmatrix} \quad (103)$$

Analizând mai atent aceasta relație de echivalentă în coordonate omogene observam ca ea este identică cu ecuația obținută prin trecere la limită (i.e $\delta_1^i \rightarrow 0$) în ecuațiile de estimare a odometriei vizuale stereo.

1.5.2. Odometrie vizuala din matricea fundamentală și din matricea esențială

Folosim în continuare aceleași notații ca și în Matricea fundamentală și matricea esențială descriind matricea fundamentală și matricea esențială. Dacă lucrăm în coordonate normalizate, parametrii de rotație și translație se vor calcula minimizând funcția:

$$f(R, T) = \sum_{i=0}^n \left(p_2^{i^T} \cdot E_{12}(R, T) \cdot p_1^i \right)^2 \quad (104)$$

Folosind coordonatele în pixeli funcția de minimizat se va exprima folosi matricea fundamentală în loc de matricea esențială:

$$g(R, T) = \sum_{i=0}^n \left([u_2^i \ v_2^i \ 1] \cdot F_{12}(R, T) \cdot \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} \right)^2 \quad (105)$$

De remarcat faptul că folosind doar două imagini consecutive fără a beneficia de informația adițională adusă de un sistem de stereoteleviziune nu se poate determina magnitudinea vectorului de translație ci doar direcția acestuia. Matematic acest lucru se poate explica analizând forma matricii esențiale:

$$E_{12}(R, T) = T_x \cdot R \quad (106)$$

Înmulțind vectorul de translație cu un scalar obținem $E_{12,s} = s \cdot T_x \cdot R$ și o funcție de minimizat:

$$f_s(R, T) = s \cdot f(R, T) \quad (107)$$

Aceasta va avea aceleași extreme ca și funcția inițială, prin urmare magnitudinea vectorului de translație nu poate fi determinată în urma minimizării.

2. Design-ul, implementarea și evaluarea algoritmilor de odometrie vizuală pe bază de trăsături

Datorită eficienței de calcul, a performantei și a robusteței acestora, propunem în continuare un set de metode eficiente pe bază de trăsături în imagine. Din punct de vedere conceptual toate acestea urmează un pattern similar, fiecare metodă fiind o variațiune de la acest pattern, prezentând avantaje și dezavantaje, în funcție de mediul exterior în care operăm, viteza de calcul disponibilă, tipul de mișcare pe care-l efectuăm. În linii mari (Figura 34), pentru fiecare perechi de imagini consecutive, într-o primă etapă se determină trăsături (pasul 1) pe baza cărora vom calcula mișcarea, apoi se face o estimare preliminară (pasul 2) a odometriei vizuale (care în multe cazuri este suficient de bună pentru a fi considerată mișcarea finală), apoi se face o rafinare a trăsăturilor și a potrivirilor de trăsături (pasul 3), urmând o estimare finală (pasul 4) și o eventuală filtrare și integrare temporală (pasul 5) a rezultatelor. Vom folosi în continuare următoarele notații pentru a descrie setul de patru imagini implicate în procesul de estimare a mișcării:

I_{L1} - imaginea anterioara stângă

I_{R1} - imaginea anterioara dreaptă

I_{L2} - imaginea curentă stângă

I_{R2} - imaginea curentă dreaptă

Astfel $I_{Sx}(u_{Sx}^i, v_{Sx}^i)$ va fi intensitatea punctului de coordinate $q_{Sx}^i = \begin{bmatrix} u_{Sx}^i \\ v_{Sx}^i \\ 1 \end{bmatrix}$

din imaginea I_{Sx} . Valoarea lui S poate fi L sau R în funcție de imaginea stângă sau drepta iar $x \in \{1,2\}$, în funcție de imaginea curentă sau imaginea anterioară (folosim aceleași notații ca și la descrierea modelului teoretic).

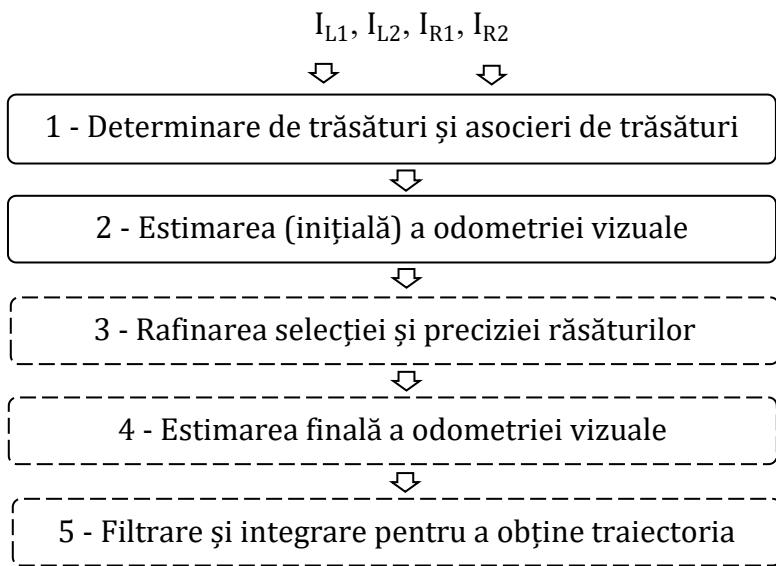


Figura 34 Etapele de estimare a odometriei vizuale pe bază de trăsături

2.1. Principii și, metode generale de determinarea de trăsături și de asociere de trăsături

2.1.1. Determinarea de trăsături potrivite pentru a fi urmărite în imagini succesive

În capitolele anterioare au fost introduse modelele matematice și geometrice folosite în odometria vizuala. Toate acestea se bazează pe existența unor trăsături de tip punct în imaginile pe baza cărora se determină parametrii de mișcare. Aceste trăsături trebuie în mod ideal să prezinte următoarele caracteristici:

- Distinctive - intensitățile imaginii dintr-o vecinătate a trăsăturii, sau vectorii de trăsături calculați pe baza acestor intensități, trebuie să fie distinctive și să reprezinte cât mai unic acea trăsătură

- Stabile în imagini consecutive - aceleași trăsături trebuie să fie determinate în imagini consecutive și să provină din proiecția în imagini a acelorași puncte fizice din scenă
- Invarianta la zgomotele care apar la achiziția imaginilor
- Invariante la deformările pixelilor cauzate de schimbarea de perspectivă între imagini consecutive
- Invariante la schimbări de iluminare (ex: camera se mută de la umbra la soare, sau își schimba orientarea față de surse de lumină).

2.1.1.1. Trăsături bazate pe derivatele imaginilor

Pentru problemele de minimizare din care obținem parametrii de mișcare avem nevoie de potriviri de trăsături de tip punct între mai multe imagini. Potrivirile se fac folosind pixelii dintr-o anumita vecinătate a trăsăturii, acești pixeli exprimând gradul de distinctivitate al trăsăturii. Pentru o trăsătură cu un grad mare de distinctivitate vom putea determina ușor potriviri în celelalte imagini. Intuitiv o trăsătură care are într-o vecinătate să doar pixeli de aproximativ aceeași intensitate nu va fi una distinctivă, o trăsătură care este situată pe o muchie nu va fi suficient de distinctivă deoarece pixelii din vecinătatea acesteia prezintă un grad mare de similaritate cu pixelii oricărei alte trăsături din lungul muchiei, și o trăsătură care se găsește la intersecția a două muchii cu orientări diferite va prezenta un grad ridicat de distinctivitate și va fi stabila în imagini consecutive dintr-un stream video.

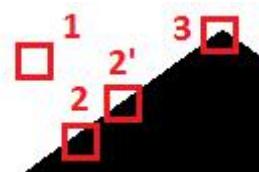


Figura 35 Trăsătura 1 are grad scăzut de distinctivitate; trăsătura 2 este pe o muchie și se asemănă cu 2' care este de asemenea în lungul muchiei; trăsătura 3 este de tip corner, fiind la intersecția dintre 2 muchii

O primă abordare propusă de Moravec în 1980 [51] se bazează pe determinarea pentru-o imagine I a variației sumei diferențelor absolute dintre o vecinătate a unui anumit punct (u, v) și aceeași vecinătate deplasată cu diverse deplasamente $(\Delta u, \Delta v)$:

$$E_{u,v}(\Delta u, \Delta v) = \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} (I(r + \Delta u, c + \Delta v) - I(r, c))^2 \quad (108)$$

Dimensiunea pe verticală a vecinătății este data de s_r iar s_c reprezintă dimensiunea pe orizontală. Dacă în toate direcțiile date de deplasamentele $(\Delta u, \Delta v)$ vom avea o valoare mică a rezultatului $E_{u,v}(\Delta u, \Delta v)$ atunci în jurul poziției (u, v) vom avea o zonă cu variații mici de intensitate. În cazul în care (u, v) este un punct de muchie, $E_{u,v}(\Delta u, \Delta v)$ are valori mari pe direcția perpendiculară pe muchie și valori mici pe direcția paralela cu muchia. Pentru colturi, $E_{u,v}(\Delta u, \Delta v)$ va avea valori mari pentru toate direcțiile. Astfel, metrica pentru un colt propusa de Moravec este:

$$T_{M,u,v} = \min_{\Delta u, \Delta v \in \{0, \pm 1, \pm 2\}; \text{Dar nu } 0 \text{ simultan}} (E_{u,v}(\Delta u, \Delta v)) \quad (109)$$

Dacă $T_{M,u,v}$ este mai mare decât un prag atunci la poziția (u, v) avem un candidat de colt.

O simplificare a abordării propuse de Moravec este făcută de Harris în 1988 [43]. Aceasta pornește de la observația că valoarea $I(r + \Delta u, c + \Delta v)$ poate fi descompusă în serie Taylor de ordinul 1 în jurul punctului (r, c) .

$$\begin{aligned} E_{u,v}(\Delta u, \Delta v) &\approx \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} (I(r, c) + I'_u(r, c) \cdot \Delta u + I'_v(r, c) \cdot \Delta v - \\ &I(r, c))^2 = \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} (I'_u(r, c) \cdot \Delta u + I'_v(r, c) \cdot \Delta v)^2 \end{aligned} \quad (110)$$

Așa cum am demonstrat în Anexa 2, aceasta valoare se reduce la:

$$E_{u,v}(\Delta u, \Delta v) \approx [\Delta u \quad \Delta v] \cdot M_{u,v} \cdot \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} \quad (111)$$

unde:

$$M_{u,v} = \begin{bmatrix} \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_u'^2(r,c) & \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_u'(r,c) \cdot I_v'(r,c) \\ \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_u'(r,c) \cdot I_v'(r,c) & \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_v'^2(r,c) \end{bmatrix} \quad (112)$$

Valorile s_r și s_c sunt dimensiunile reginii de interes pe verticală (numărul de rânduri din imagine) respectiv pe orizontală (numărul de coloane din imagine).

Astfel, variația intensității în jurul punctului (u,v) depinde strict de derivatele parțiale ale imaginii în jurul acestui punct. Forma funcției $E_{u,v}(\Delta u, \Delta v)$ în jurul punctului $(0,0)$ depinde de forma matricii M , mai precis de valorile proprii $\lambda_{1,u,v}$ și $\lambda_{2,u,v}$ ale acesteia. În 1988 Harris [43] propune aplicarea unui prag pentru următoarea metrică:

$$T_{H,u,v}(\lambda_{1,u,v}, \lambda_{2,u,v}) = \det(M_{u,v})^2 - k \cdot \text{tr}(M_{u,v}) = (\lambda_{1,u,v} \cdot \lambda_{2,u,v})^2 - k \cdot (\lambda_{1,u,v} + \lambda_{2,u,v}) \quad (113)$$

Mai târziu, în 1994, Shi & Tomasi [45] propun aplicarea unui prag pe valoarea de mai jos, simplificând astfel calculele pentru identificarea trăsăturilor:

$$T_{S,u,v}(\lambda_{1,u,v}, \lambda_{2,u,v}) = \min(\lambda_{1,u,v}, \lambda_{2,u,v}) \quad (114)$$

Aceste trăsături sunt cunoscute în literatura și sub denumirea "good features to track" deoarece acestea sunt potrivite pentru tracking în timp, și vor fi intens folosite în lucrarea prezentă.

In Figura 36 sunt prezentate asemănările dintre colturile Harris și trăsăturile propuse de Shi & Tomasi. Cu negru sunt reprezentate curbele de nivel ale funcției $T_{H,u,v}(\lambda_{1,u,v}, \lambda_{2,u,v})$. Cu cât pragul de selecție al colturilor este mai mare cu atât colțul este mai puternic adică variațiile intensităților dintr-o regiune de interes din jurul coordonatelor sale sunt mai mari. De asemenea se observă zona hașurată cu verde pentru care valoarea proprie minima a matricii M este mai mare decât un prag.

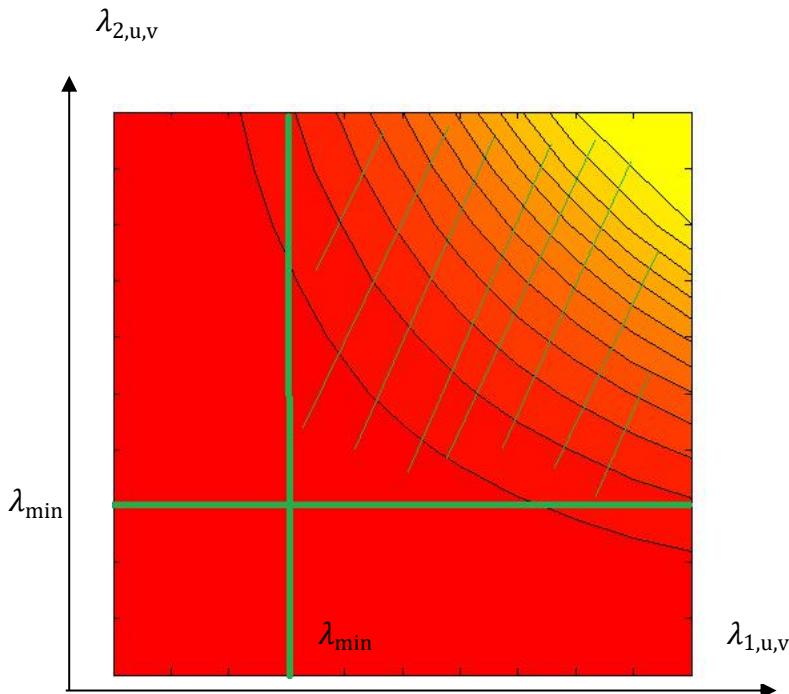


Figura 36 Colturi Harris comparate cu “good features to track”. Curbele de nivel pentru $T_{H,u,v}(\lambda_{1,u,v}, \lambda_{2,u,v})$ respectiv zona hașurată cu verde pentru un prag pe $T_{S,u,v}(\lambda_{1,u,v}, \lambda_{2,u,v}) > \lambda_{\min}$

2.1.1.2. Trăsături bazate pe derivatele imaginii interpretate pe baza analizei componentelor principale

Interpretarea intuitivă data de [43] se leagă de forma funcției $E_{u,v}(\Delta u, \Delta v)$ în jurul punctului $(0,0)$, aceasta depinzând de forma matricii M , mai precis de valorile proprii λ_1 și λ_2 ale acesteia. O interpretare alternativă pe care am dat-o se bazează pe analiza componentelor principale (PCA) [50]. Considerăm următorul set de vectori compuși din derivatele parțiale orizontale și verticale ale imaginii sursă într-o regiune de interes din jurul unui anumit punct (u, v) :

$$S_{u,v} = \left\{ \begin{bmatrix} I'_u(r, c) \\ I'_v(r, c) \end{bmatrix} \mid r = u - s_r \dots u + s_r; c = v - s_c \dots v + s_c \right\} \quad (115)$$

Dacă acest set de puncte 2D este concentrat aproksimativ într-un singur punct atunci regiunea de interes are intensități aproksimativ egale. Dacă setul este distribuit în lungul unei singure axe atunci regiunea de interes conține o muchie, iar dacă setul de interes este distribuit în lungul a două axe atunci regiunea de interes conține o trăsătură de tip colț. Notăm în continuare cu \bar{I}_u și \bar{I}_v mediile celor 2 componente ale vectorilor din setul $S_{u,v}$:

$$\bar{I}_u = \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)} \cdot \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I'_u(r, c) \quad (116)$$

$$\bar{I}_v = \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)} \cdot \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I'_v(r, c) \quad (117)$$

Notând cu $X_{u,v}$ matricea având 2 linii și $(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)$ având următoarele elemente.

$$X_{u,v} = \begin{bmatrix} \dots I'_u(r, c) - \bar{I}_u \dots \\ \dots I'_v(r, c) - \bar{I}_v \dots \end{bmatrix} \quad (118)$$

Derivatele pe direcția u sunt situate pe prima linie a matricii $X_{u,v}$ iar derivatele pe direcția v pe cea de-a doua linie, r și c luând valori între $u - s_r \dots u + s_r$ respectiv $v - s_c \dots v + s_c$. Matricea de covarianta în jurul punctului de coordonate (u, v) va avea forma următoare:

$$C_{u,v} = \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1) - 1} \cdot X_{u,v} \cdot X_{u,v}^T = \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1) - 1} \cdot \left[\begin{array}{cc} \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} (I'_u(r, c) - \bar{I}_u)^2 & \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} (I'_u(r, c) - \bar{I}_u) \cdot (I'_v(r, c) - \bar{I}_v) \\ \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} (I'_v(r, c) - \bar{I}_v) \cdot (I'_u(r, c) - \bar{I}_u) & \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} (I'_v(r, c) - \bar{I}_v)^2 \end{array} \right] \quad (119)$$

Se observă că dacă mediile derivatelor parțiale ale intensităților din regiunea de interes, \bar{I}_u respectiv \bar{I}_v sunt zero atunci forma matricii este identica cu forma matricii $M_{u,v}$, dacă ignorăm un factor de scalare care poate fi inclus și în funcția $E_{u,v}(\Delta u, \Delta v)$. Conform analizei componentelor principale (PCA) setul de vectori $X_{u,v}$ va avea cea mai mare variație în direcția vectorului propriu corespunzător celei mai mari valori proprii, și varianta este data de mărimea valorii proprii. De

asemenea, a doua cea mai mare varianta este data de a doua valoare proprie ca mărime și se realizează pe direcția vectorului propriu corespunzător celei de-a doua valori proprii. Astfel setul de vectori $X_{u,v}$ vor avea zero, una sau două componente dominante în funcție de mărimea valorilor proprii ale matricii de covarianță.

Dacă matricea $C_{u,v}$ va avea 2 valori proprii mai mari decât un anumit prag (vezi condiția folosită pentru "good features to track") atunci punctul (u,v) va fi o trăsătură de tip colț. Dacă matricea $C_{u,v}$ are o valoare proprie mare și una mică atunci regiunea de interes cu centrul în (u,v) va continua pixeli de muchie. Dacă notam cu $\lambda_{1,u,v}^c$ și $\lambda_{2,u,v}^c$ aceste valori proprii atunci cu noua formulare trăsăturile potrivite pentru tracking vor fi cele pentru care valoarea $T_{C,u,v}(\lambda_{1,u,v}^c, \lambda_{2,u,v}^c)$ depășește un anumit prag:

$$T_{C,u,v}(\lambda_{1,u,v}^c, \lambda_{2,u,v}^c) = \min(\lambda_{1,u,v}^c, \lambda_{2,u,v}^c) \quad (120)$$

In cazul în care 2 pixeli având coordonate apropriate (i.e. mai mici decât un anumit prag d_{NMS}) și respectă ambii condiția de corner, fie că aceasta este măsură Harris, Shi & Tomasi sau noua metrică propusa pe baza analizei componentelor principale (PCA), cel care are o măsură mai slabă va fi eliminat. Acest procedeu se numește suprimare punctelor non-maxime (non maxima suppression - NMS).

Revenind la notația $I_{Sx}(u_{Sx}^i, v_{Sx}^i)$ unde valoarea lui S poate fi L sau R în funcție de imaginea stângă sau dreptă, iar $x \in \{1,2\}$ în funcție de imaginea curentă sau imaginea anterioară. Vom folosi următoarele notații în restul capitolelor:

- Colturile Harris din imaginea I_{Sx}

$$\begin{aligned} H_{I_{Sx}} = \{(u_{Sx}^i, v_{Sx}^i)\} &= \{(u, v) | T_{H,u,v}(\lambda_{1,u,v}, \lambda_{2,u,v}) \\ &> p_H \text{ și } (u, v) \text{ ramane după NMS}\} \end{aligned}$$

- "Good features to track" din imaginea I_{Sx}

$$\begin{aligned} S_{I_{Sx}} = \{(u_{Sx}^i, v_{Sx}^i)\} &= \{(u, v) | T_{S,u,v}(\lambda_{1,u,v}, \lambda_{2,u,v}) \\ &> p_S \text{ și } (u, v) \text{ ramane după NMS}\} \end{aligned}$$

- Colturile obținute prin PCA din imaginea I_{Sx}

$$C_{I_{Sx}} = \{(u_{Sx}^i, v_{Sx}^i)\} = \{(u, v) | T_{C,u,v}(\lambda_{1,u,v}^c, \lambda_{2,u,v}^c) > p_c \text{ & } (u, v) \text{ ramane dupa NMS}\}$$

Valorile pragurilor p_H , p_S sau p_C alături de distanța minimă dintre 2 trăsături - d_{NMS} sunt parametrii care se pot folosi pentru reglarea densității și calității colturilor rezultante.

2.1.1.3. Trăsături pe baza relațiilor de ordine dintre intensități

Trăsăturile de tip colt prezentate anterior se bazează pe diferențierea imaginii de intensitate. O alta clasă de trăsături sunt cele care calculează scorul pe baza căruia se ia decizia dacă un anumit pixel este corner sau nu direct din imaginea de intensitate. Un avantaj al acestor trăsături este ca sunt mai robuste la zgomotele din imagine și ca se pot calcula mai repede.

O astfel de abordare clasica este detectorul SUSAN (Smallest Univalve Segment Assimilating Nucleus) propus în 1997 [48]. Acesta consideră zone circulare în jurul unui pixel central, și numără câți pixeli din jurul pixelului central au intensitate similară cu pixelul central, într-o anumita marjă de toleranță. Numărul acestor pixeli trebuie să fie sub un prag. În plus, cu cât numărul acestor pixeli este mai mic cu atât scorul de corner este mai mare (i.e un pixel central care difera de pixelii dintr-o vecinătate circulară este un corner). În mod formal:

$$i_{r,c} = \begin{cases} 1, & |I(u,v) - I(r,c)| \leq t \\ 0, & \text{altfel} \end{cases} \quad (121)$$

$$n_{u,v} = \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} i_{r,c} \quad (122)$$

Scorul de corner pentru pixelul (u,v) se calculează astfel:

$$T_{Susan,u,v} = \begin{cases} g - n_{u,v}, & n_{u,v} < g \\ 0, & \text{altfel} \end{cases} \quad (123)$$

Valoarea lui g este constantă și se alege mai mică decât jumate din numărul de pixeli din zona circulară. De asemenea se elimină și candidații pentru care pixelii care dau $n_{u,v}$ nu reprezintă o regiune

compactă. Într-un ultim pas se aplica non maxims suppression pentru rafinare.

Tot pe un cerc Bresenham, dar fără a considera și pixelii din interiorul cercului se bazează abordarea făcută în [49]. Aceștia consideră punctele diametral opuse de pe cerc și calitatea unui corer este data de minimul sumei pătratelor diferențelor de intensitate dintre pixelul central și cele două puncte diametral opuse:

$$T_{Hedley,u,v} = \min_{(r_1,c_1),(r_2,c_2) - \text{diametral opuse}} (I(u,v) - I(r_1,c_1))^2 + (I(u,v) - I(r_2,c_2))^2 \quad (124)$$

De remarcat ca valoarea aceasta este mare atunci când nu există puncte diametral opuse având aceeași intensitate ca și pixelul central (Figura 37).

Aceasta abordare pe baza analizei punctelor diametral opuse este punctul de pornire pentru trăsăturile FAST (Features from Accelerated Segment Test). O prima versiune a detectorului [46] se bazează pe observația că un punct de interes este colt dacă există pe cercul din jurul pixelului central un set de n pixel consecutivi cu intensitate mai mare decât pixelul central plus un prag sau un set de n pixeli consecutivi cu intensitate mai mică decât pixelul central minus un prag. Observând (Figura 37) că numărul total de pixeli de pe cerc este 16, pentru $n = 12$ se poate aplica următoarea logică:

- În prima fază se testează pixelii 1 și 9. Dacă aceștia sunt similari ca și intensitate cu pixelul central (u, v) atunci acesta nu poate fi de tip colt
- Dacă cel puțin unul dintre pixelii 1 și 9 difere de pixelul central atunci se testează pixelii 5 și 13. Dacă ambii sunt similari în intensitate cu pixelul central atunci trăsătura nu este de tip colt
- Dacă cel puțin 3 dintre pixelii 1,5,9,13 sunt diferiți ca și intensitate de pixelul central atunci se testează restul de pixeli intermediari până se lucrează la $n = 12$ pixeli consecutivi care trebuie să respecte condiția

Se observă că pentru a determina o trăsătură de tip colt este nevoie de testarea a 12 pixeli, în schimb pentru a decide că o trăsătură nu este de tip colt este suficient uneori să testăm doar 2 pixeli. Toată

aceasta logica este valabila doar pentru $n = 12$ deși o trăsătură poate fi considerată tot de tip colt dacă 9 din 16 pixeli consecutivi difera ca și intensitate de pixelul central. Pentru a eficientiza testele pentru valori arbitrare ale lui n aceeași autori propun în [47] o metoda de optimizare a numărului de teste pe baza de arbori de decizie optimi, obținând pentru $n = 9$ un număr mediu de 2.26 teste per pixel.

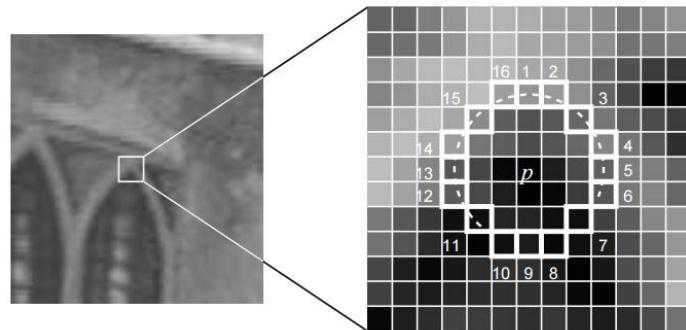


Figura 37 Un cerc Bresenham, discretizat în jurul pixelului central (p în imaginea din dreapta)

2.1.2. Asocierea trăsăturilor pe bază de descriptori

Așa cum a fost descris în capitolele care descriu modelarea matematică a problemei de estimare a odometriei vizuale este nevoie de corespondențe de trăsături între imagini consecutive dintr-un set-up de camera mono sau stereo. O astfel de corespondență este corecta atunci când trăsăturile acesteia sunt proiecții ale aceluiași punct din spațiul tridimensional. Algoritmii de detecție a acestor corespondențe dintre trăsături se bazează pe calculul unui scor de similaritate pe baza intensităților din jurul acestor corespondențe, sau pe baza unor valori caracteristice (vectori de trăsături) derive din aceste intensități. Cele mai directe metode de calcul a acestui scor de similaritate sunt suma diferențiatelor absolute (SAD - sum of absolute differences) sau suma pătratelor diferențelor (SSD - sum of squared differences) [3]. Considerând două imagini I și J, două trăsături de coordonate (u^i, v^i)

respectiv (u^i, v^j) și o regiune de interes dreptunghiulară de înălțime $2 \cdot s_r + 1$ iar $2 \cdot s_c + 1$, SSD și SAD se pot calcula astfel:

$$\begin{aligned} \text{SSD}(u^i, v^i, u^j, v^j) &= \\ &= \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} (I(u^i + \Delta r, v^i + \Delta c) - J(u^j + \Delta r, v^j + \Delta c))^2 \end{aligned} \quad (125)$$

$$\begin{aligned} \text{SAD}(u^i, v^i, u^j, v^j) &= \\ &= \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} |I(u^i + \Delta r, v^i + \Delta c) - J(u^j + \Delta r, v^j + \Delta c)| \end{aligned} \quad (126)$$

Una dintre cele mai mari probleme pe care o au aceste metriki de similaritate este aceea că nu sunt invariante la schimbări de iluminare. Ca și alternativă se pot folosi metriki precum Normalized Cross Correlation (NCC) sau Zero Mean Normalized Cross Correlation (ZNCC) [178]:

$$\text{NCC}(u^i, v^i, u^j, v^j) = \frac{\sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} I(u^i + \Delta r, v^i + \Delta c) \cdot J(u^j + \Delta r, v^j + \Delta c)}{\sqrt{\sum_{\Delta r=-s_r}^{s_r} I^2(u^i + \Delta r, v^i + \Delta c)} \cdot \sqrt{\sum_{\Delta c=-s_c}^{s_c} J^2(u^j + \Delta r, v^j + \Delta c)}} \quad (127)$$

Acestea se pot interpreta pe baza produsului scalar dintre vectori. Dacă privim regiunea de interes din jurul punctelor (u^i, v^i) și (u^j, v^j) ca și vectori coloana având $(2 \cdot s_r + 1) \times (2 \cdot s_c + 1)$ rânduri, atunci cei doi radicali de la numitor sunt magnitudinile acestor vectori, la numărător avem produsul scalar al celor doi vectori iar $\text{NCC}(u^i, v^i, u^j, v^j)$ reprezintă cosinusul unghiului dintre cei doi vectori. O valoare apropiată de $1 = \cos(0)$ pentru $\text{NCC}(u^i, v^i, u^j, v^j)$ va însemna că cei doi vectori (adică intensitățile imaginilor care compun cei doi vectori) sunt similari iar o valoare de 0 va însemna că cei 2 vectori sunt complet diferenți.

Dacă facem o normalizare a celor 2 vectori prin scăderea mediilor intensităților din zonele de interes obținem o metrică invariantă la iluminare, mai precis variații de contrast și luminozitate ale imaginii:

$$\bar{I}(u^i, v^i) = \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)} \cdot \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} I(u^i + \Delta r, v^i + \Delta c) \quad (128)$$

$$\bar{J}(u^j, v^j) = \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)} \cdot \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} J(u^j + \Delta r, v^j + \Delta c) \quad (129)$$

$$ZNCC(u^i, v^i, u^j, v^j) = \frac{\sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} (I(u^i + \Delta r, v^i + \Delta c) - \bar{I}(u^i, v^i)) \cdot (J(u^j + \Delta r, v^j + \Delta c) - \bar{J}(u^j, v^j))}{\sqrt{\sum_{\Delta r=-s_r}^{s_r} (I(u^i + \Delta r, v^i + \Delta c) - \bar{I}(u^i, v^i))^2} \cdot \sqrt{\sum_{\Delta c=-s_c}^{s_c} (J(u^j + \Delta r, v^j + \Delta c) - \bar{J}(u^j, v^j))^2}} \quad (130)$$

Aceeași interpretare pe baza de produs scalar sau teorema cosinusului se aplică și aici. Se poate verifica matematic faptul că prin transformare de tipul $I'(u, v) = a \cdot I(u, v) + b$ valoarea ZNCC rămâne neschimbata. O scurta demonstrație a acestui fapt se poate găsi în Anexa 3.

Metricile propuse mai sus se bazează toate pe utilizarea directă a intensităților din cele două imagini în funcția care calculează gradul de similaritate. Astfel, un număr mare de metode din literatura calculează un vector de caracteristici descriptiv pentru regiunea de interes din jurul punctului de interes, pe care în continuare îl vom numi descriptor.

Descriptorii propuși în [186] se pe statistici și relații de ordine între instilațiile pixelilor dintr-o anumita vecinătate a pixelului central. Pe lângă invarianta la condiții de iluminare acestea prezintă și un grad de toleranță la zone de interes care acoperă pixeli aparținând atât obiectelor din prim plan cat și celor din fundal. Metoda propusa în [186] propune două transformate: transformata rank și transformata census. Transformata rank numără câți pixeli dintr-o anumita vecinătate au intensitatea mai mică decât intensitatea pixelului central. Notând cu $|M|$ numărul de elemente din mulțimea M , putem exprima formal transformata rank astfel:

$$R_I(u, v) = |\{(u', v') \in \{u - s_r' \dots u + s_r'\} \times \{v - s_c' \dots v + s_c'\} | I'(u', v') < I(u, v)\}| \quad (131)$$

Astfel dacă dimensiunea regiunii de interes este $(2 \cdot s_r' + 1) \cdot (2 \cdot s_c' + 1)$ atunci valorile lui $R(u, v)$ vor fi numere întregi din intervalul $[0, (2 \cdot s_r' + 1) \cdot (2 \cdot s_c' + 1) - 1]$. De exemplu pentru $s_r' = 7$ și $s_c' = 7$ intervalul va fi $[0, 225]$, interval aproape la fel de mare ca și cel al intervalului unei intensități pe 8 biți, rezultând o reprezentare foarte descriptiva a conținutului imaginii, invarianta la condiții de iluminare. Aceasta indică un grad ridicat de distinctivitate al acestei transformate. Metrica folosită pentru a determina gradul de similaritate a două trăsături este SAD pe imaginea rezultată după transformata, și o vom nota cu RSAD:

$$RSAD(u^i, v^i, u^j, v^j) = \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} |R_I(u^i + \Delta r, v^i + \Delta c) - R_J(u^j + \Delta r, v^j + \Delta c)| \quad (132)$$

Tot în [186] este propusa și transformata CENSUS, un descriptor foarte des folosit pentru potriviri mai ales în stereoviziune [188], [189]. Aceasta creează un sir binar prin concatenarea valorilor $\epsilon(u', v')$ pentru toți pixelii dintr-o vecinătate a punctului de coordonate (u, v) :

$$\epsilon(u', v') = \begin{cases} 0, & I(u', v') < I(u, v) \\ 1, & \text{altfel} \end{cases} \quad (133)$$

Notam cu $C_I(u, v)$ concatenarea valorilor $\epsilon(u', v')$ pentru $(u', v') \in \{u - s_r, \dots, u + s_r\} \times \{v - s_c, \dots, v + s_c\}$ și $(u', v') \neq (u, v)$. Acești descriptori sub formă unor șiruri binare se calculează pe baza distantei Hamming [190], adică diferența dintre doi astfel de descriptori este data de numărul biților diferiți aflați pe aceeași poziție în cele două șiruri binare. Aceasta distanță se agregă pe o regiune de interes pentru o mai bună stabilitate:

$$H(u^i, v^i, u^j, v^j) = \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} \text{Hamming}(C_I(u^i + \Delta r, v^i + \Delta c), C_J(u^j + \Delta r, v^j + \Delta c)) \quad (134)$$

O extensie a transformării census [185] este propusă de către Stein în 2004. Acesta calculează pentru fiecare pixel un sir de caractere similar cu șirul binar al transformatei census, dar care poate să ia valorile 0, 1 și 2 pe baza următoarei reguli:

$$\epsilon(u', v') = \begin{cases} 0, & I(u, v) - I(u', v') > \tau \\ 1, & |I(u, v) - I(u', v')| \geq \tau \\ 2, & I(u', v') - I(u, v) > \tau \end{cases} \quad (135)$$

Prin concatenarea valorilor $\epsilon(u', v')$ se obține un sir $EC_I(u, v)$ de valori 0, 1, 2 pe baza cărora se poate face căutarea și potrivirea de trăsături între imagini succesive.

Pe baza transformatei rank și a transformărilor de tip census se pot obține potriviri robuste de trăsături chiar dacă regiunea de interes conține sub-regiuni provenind din obiecte diferite atât timp cat ordinea intensităților dintre sub-regiuni rămâne neschimbata între cele două imagini. Altfel spus, dacă în imaginea I punctul (u^i, v^i) se va situa pe

muchia unui obiect, atunci aceasta va conține o sub-regiune $R_{I,1}$ pe obiect și o regiune $R_{I,2}$ din fundal sau de pe alt obiect. După mișcarea camerei, datorită schimbării de perspectivă se obține o nouă imagine J iar regiunea de interes din jurul punctul corespunzător (u^j, v^j) va conține la rândul ei o sub-regiune $R_{J,1}$ și o sub-regiune $R_{J,2}$. Dacă ordinea dintre pixelii din $R_{I,1}$ și $R_{I,2}$ se păstrează în noua imagine între $R_{J,1}$ și $R_{J,2}$, trăsăturile de coordonate (u^i, v^i) și (u^j, v^j) pot fi potrivite cu succes.

Un alt descriptor care se bazează pe relații de ordine între intensități este BRIEF [184]. În plus fata de CENSUS, acesta nu se bazează pe o un pixel central pentru a compune sirul binar, ci se bazează pe perechi de poziții din interiorul regiunii de interes. Astfel, dacă pixelul central este afectat de zgomot acesta nu se va reflecta în întreg descriptorul. Pentru o specificare formală a descriptorului consideram o mulțime de perechi de puncte dintr-o zona rectangulară de dimensiune $(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)$:

$$\begin{aligned} & (u'_k, v'_k, u''_k, v''_k) \in \\ & \{u - s_r \dots u + s_r\} \times \{v - s_c \dots v + s_c\} \times \{u - s_r \dots u + s_r\} \times \{v - s_c \dots v + s_c\} \\ & \varepsilon_k(u'_k, v'_k, u''_k, v''_k) = \begin{cases} 0, & I(u'_k, v'_k) < I(u''_k, v''_k) \\ 1, & \text{altfel} \end{cases} \quad (136) \end{aligned}$$

Notam cu $B_l(u, v)$ concatenarea valorilor $\varepsilon_k(u'_k, v'_k, u''_k, v''_k)$. Metoda de comparație a acestor valori este similară cu cea pentru descriptori CENSUS:

$$H(u^i, v^i, u^j, v^j) = \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} \text{Hamming}(B_l(u^i + \Delta r, v^i + \Delta c), C_j(u^j + \Delta r, v^j + \Delta c)) \quad (137)$$

În cazul descriptorului BRIEF aceste perechi de puncte se iau aleator, generându-se după distribuții uniforme sau Gaussiene centrate în centrul regiunii de interes. În plus, se face o netezire a imaginii în prealabil folosind un nucleu de convoluție de tip Gaussian.

În varianta originală numărul de perechi considerate este de 128, 256 sau 512 rezultant astfel descriptori de dimensiune 16, 32 sau 64.

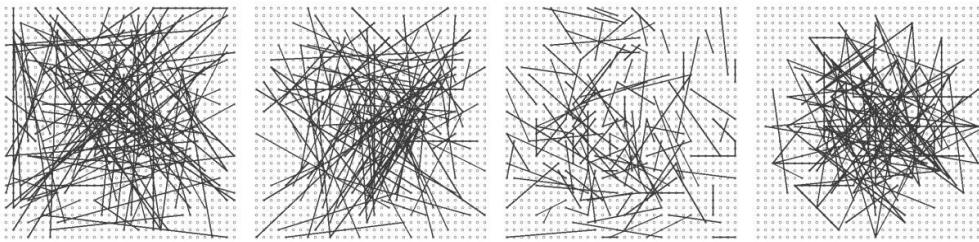


Figura 38 Linii între perechile de puncte selectate pentru BRIEF. Fiecare dintre cele 4 imagini indica 128 de perechi

2.1.2.1. Alți descriptori de interes din literatură

În literatura există o gama foarte variată de trăsători de tip punct, unele dintre acestea fiind însotite și de un set de valori reprezentative (descriptori), calculate pe baza valorilor intensităților din jurul punctului central. Amintim dintre acestea fără a intra în cele mai mici detalii SIFT - Scale Invariant Feature Transform [195], SURF - Speeded Up Robust Features [196], FREAK - Fast Retina Keypoint [193], ORB - Oriented BRIEF [197], BRISK - Binary Robust Invariant Scalable Keypoints [194], GLOH - Gradient Location and Orientation Histograms [192]. Dintre acestea cele mai renumite trăsături sunt trăsăturile SIFT. Acestea au fost propuse de către Lowe în 1999 [191], scopul utilizării acestora fiind mai larg decât determinarea de corespondențe între imagini succesive, și anume recunoașterea obiectelor. Capabilitățile pe care le oferă aceste trăsături sunt mai mari decât cele necesare pentru odometria vizuala, prin urmare utilizarea lor în odometrie vizuala nu este neapărat motivată din punct de vedere al îmbunătățirilor aduse. În plus față de localizarea unor trăsături de tip punct în imagini, este calculat și un descriptor care este folosit ulterior pentru potrivirea trăsăturilor pentru aplicații precum recunoaștere de obiecte pe baza de potriviri, compunere de imagini panoramice din imagini între care sunt diferențe mari de perspectivă. Pentru astfel de aplicații, atât detecția trăsăturilor cat și descriptorii corespunzători acestora sunt proiectați pentru a fi regăsiți în imagini diferite ale aceluiși scene (repetabili), distinctivi și invariante la condițiile de iluminare, schimbări de orientare a imaginii, schimbări de scală, și schimbări ale perspectivei din care este achiziționată o anumita scenă, precum și rezistent la zgomotele din

în general pentru odometria vizuala se acete criterii se pot relaxa. Astfel, schimbările de orientare de la un frame la altul vor fi foarte mici. La fel, în cazul mișcării care apare între două imagini consecutive preluate cu o camera, schimbările de scăla dintre imagini consecutive sunt nesemnificative pentru trăsăturile de la distanță și mici pentru trăsăturile din apropiere. Pentru a determina punctele de interes, pentru care se vor calculat descriptorii se folosesc o reprezentare piramidală a imaginii pe mai multe rezoluții. Pozițiile trăsăturilor se obțin prin determinarea punctelor de extremă în diferențele de filtre Gaussiene de diverse valori ale variantei, aplicate pe imagine la diferite rezoluții (scale). Având poziția trăsăturii la o anumita scăla în imagine se calculează o histogramă ponderată a orientărilor gradienților din jurul acestui punct. Ponderile histogramei sunt date de magnitudinea gradienților înmulțite cu o pondere data de un Gaussian centrat în punctul de interes. Maximul din această histogramă va da orientarea globală a trăsăturii. Pentru a calcula descriptorul trăsăturii, se iau un număr de 16 regiuni a către 16 pixeli fiecare și se calculează pentru fiecare regiune căte o histogramă a orientărilor gradienților din acea regiune fata de orientarea globală a trăsăturii. Fiecare din cele 16 histograme are căte 8 intrări rezultând astfel un descriptor de 128 de elemente. De remarcat că histogramele bazate pe orientările gradienților sunt o trăsătură foarte discriminatorie, care aproximativ în paralel au fost folosite și în metode de inteligență artificială de clasificare și detecție a obiectelor [198].

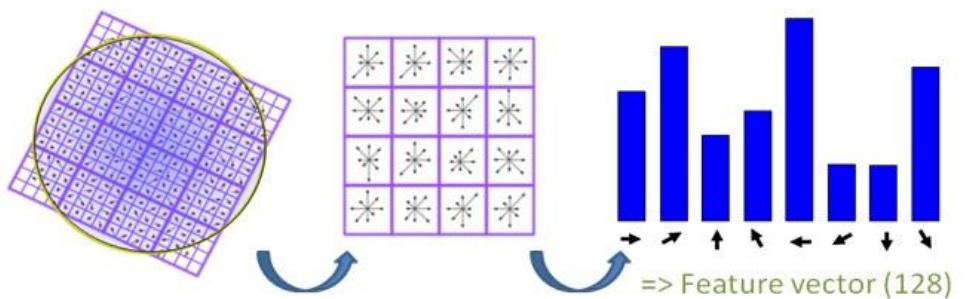


Figura 39 Calculul descriptorului SIFT relativ la orientarea globală a descriptorului

2.1.3. Asocierea trăsăturilor pe bază de metode diferențiale

O abordare alternativa de detecție a corespondențelor de trăsături în imagine se bazează pe metode de flux optic. Dacă în metodele de determinare a corespondențelor pe bază de descriptori se determinau aceste trăsături și descriptorii asociați lor în toate imaginile între care dorim să determinăm asociările, în cazul metodelor pe baza de flux optic se determină trăsături doar în una dintre imagini apoi se determină corespondențele în mod iterativ, pe baza informației de gradient din imagini. Astfel, având două imagini I și J, și o trăsătură de coordonate (u, v) în prima imagine, trebuie să determinăm deplasamentele $(\Delta u, \Delta v)$ pentru care:

$$I(u, v) = J(u + \Delta u, v + \Delta v) \quad (138)$$

Dacă privim imaginea ca o funcție de două variabile și derivabilă așa cum o face una dintre primele abordări de flow optic [199], și dacă presupunem că deplasamentele Δu și Δv sunt suficient de mici atunci putem dezvolta în serie Taylor:

$$0 = J'_u(u, v) \cdot \Delta u + J'_v(u, v) \cdot \Delta v + J(u, v) - I(u, v) \quad (139)$$

Dintr-un singur pixel se obține astfel o singură ecuație cu două necunoscute. Prin utilizarea unei vecinătăți în jurul coordonatelor în jurul cărora dorim să calculăm fluxul optic, și considerând că pixelii dintr-o vecinătate au aceleași deplasamente Δu și Δv de la un frame la celalalt obținem mai multe ecuații în Δu și Δv din care se poate extrage valoarea deplasamentelor. O astfel de abordare este folosită în renumitul algoritm de flow optic propus de Lukas & Kanade în 1981 [206]. O descriere detaliată a unei implementări a acestui algoritm se găsește în [207]. Deoarece dezvoltarea în serie Taylor a funcției $J(u + \Delta u, v + \Delta v)$ în jurul punctului (u, v) va fi posibilă doar pentru valori mici ale deplasamentelor, considerăm că se cunoaște o valoare de inițializare a acestor deplasamente $(\Delta u_0, \Delta v_0)$. Având aceste deplasamente inițiale ecuațiile devin:

$$I(u, v) = J(u + \Delta u_0 + \Delta u, v + \Delta v_0 + \Delta v) \quad (140)$$

$$0 = J'_u(u + \Delta u_0, v + \Delta v_0) \cdot \Delta u + J'_v(u + \Delta u_0, v + \Delta v_0) \cdot \Delta v +$$

$$+J(u + \Delta u_0, v + \Delta v_0) - I(u, v) \quad (141)$$

O astfel de formulare va permite inițializarea acestor deplasamente pe baza mișcării cunoscute de la alți senzori, sau prezisă din imaginile anterioare dar și o rezolvare iterativa a problemei prin lucrul la rezoluții mai scăzute initial, asa cum este descris în [206] și [207]. Această inițializare poate fi făcută și pe baza odometriei prezise din mișcările anterioare, sau a unei estimări inițiale ale odometriei, aspecte tratate în detaliu într-un capitol ulterior al acestei lucrări.

$$E_{u,v}(\Delta u, \Delta v) = \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} \left(J'_u(r + \Delta u_0, c + \Delta v_0) \cdot \Delta u + J'_u(r + \Delta u_0, c + \Delta v_0) \cdot \Delta v + J(r + \Delta u_0, c + \Delta v_0) - I(r, c) \right)^2 \quad (142)$$

Trebuie să determinăm minimele funcției $E_{u,v}(\Delta u, \Delta v)$, deci acele puncte pentru care derivatele parțiale ale funcției în raport cu Δu și Δv se anulează:

$$\frac{\partial E_{u,v}(\Delta u, \Delta v)}{\partial \Delta u} = 2 \cdot \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} J'_u(r + \Delta u_0, c + \Delta v_0) \cdot \left(J'_u(r + \Delta u_0, c + \Delta v_0) \cdot \Delta u + J'_u(r + \Delta u_0, c + \Delta v_0) \cdot \Delta v + J(r + \Delta u_0, c + \Delta v_0) - I(r, c) \right) \quad (143)$$

$$\frac{\partial E_{u,v}(\Delta u, \Delta v)}{\partial \Delta v} = 2 \cdot \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} J'_v(r + \Delta u_0, c + \Delta v_0) \cdot \left(J'_u(r + \Delta u_0, c + \Delta v_0) \cdot \Delta u + J'_u(r + \Delta u_0, c + \Delta v_0) \cdot \Delta v + J(r + \Delta u_0, c + \Delta v_0) - I(r, c) \right) \quad (144)$$

Formula de calcul al deplasamentelor Δu și Δv este propusă în Anexa 5. Un detaliu foarte subtil și care aduce optimizări de performanță semnificative este propus în [207]. Aceasta este optimizare este că și aproximatie intuitivă în [207], dar o analiză atenta a particularităților matematice ne indică faptul că poate fi explicata și riguros din punct de vedere matematic. În loc de a folosi derivatele imaginii destinație J putem folosi derivatele imaginii sursă I , și adăugăm în plus observația că derivatele celor două imaginii în jurul punctelor de potrivire sunt egale:

$$I'_u(u, v) = J'_u(u + \Delta u_0 + \Delta u, v + \Delta v_0 + \Delta v) \quad (145)$$

$$I'_v(u, v) = J'_v(u + \Delta u_0 + \Delta u, v + \Delta v_0 + \Delta v) \quad (146)$$

Dezvoltând în serie Taylor în jurul punctului destinație $(u + \Delta u_0 + \Delta u, v + \Delta v_0 + \Delta v)$ în loc de a dezvolta în jurul punctului sursă $(u + \Delta u_0, v + \Delta v_0)$ vom avea:

$$J(u + \Delta u_0, v + \Delta v_0) = J(u + \Delta u_0 + \Delta u, v + \Delta v_0 + \Delta v) + J'_u(u + \Delta u_0 + \Delta u, v + \Delta v_0 + \Delta v) \cdot (-\Delta u) + J'_v(u + \Delta u_0 + \Delta u, v + \Delta v_0 + \Delta v) \cdot (-\Delta v) \quad (147)$$

Rezultă:

$$J(u + \Delta u_0, v + \Delta v_0) = I(u, v) + J'_u(u + \Delta u_0 + \Delta u, v + \Delta v_0 + \Delta v) \cdot (-\Delta u) + J'_v(u + \Delta u_0 + \Delta u, v + \Delta v_0 + \Delta v) \cdot (-\Delta v) \quad (148)$$

In continuare, prin înlocuirea derivatelor parțiale ale imaginii J cu cele ale imaginii I și rearanjând termenii obținem:

$$0 \approx I'_u(u, v) \cdot \Delta u + I'_v(u, v) \cdot \Delta v + J(u + \Delta u_0, v + \Delta v_0) - I(u, v) \quad (149)$$

Reluând aceeași demonstrație ca și în Anexa 5, se obține:

$$\begin{aligned} & \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \\ & \begin{bmatrix} \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_u'^2(r, c) & \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_u'(r, c) \cdot I_v'(r, c) \\ \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_u'(r, c) \cdot I_v'(r, c) & \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_v'^2(r, c) \end{bmatrix}^{-1} \cdot \\ & \begin{bmatrix} -\sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_u'(r, c) \cdot (J(r + \Delta u_0, c + \Delta v_0) - I(r, c)) \\ -\sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} I_v'(r, c) \cdot (J(r + \Delta u_0, c + \Delta v_0) - I(r, c)) \end{bmatrix} = \\ & = M_{u,v}^{-1} \cdot T_{u,v} \end{aligned} \quad (150)$$

Observam că matricea $M_{u,v}$ este aceeași ca și cea folosită pentru a decide dacă un anumit pixel conține o trăsătură de tip corner. Un aspect important al acestei formulări este faptul că pentru a funcționa bine este nevoie de două valori de initializare Δu_0 și Δv_0 . Astfel pornind de la două valori inițiale pentru aceste deplasamente, acestea se rafinează iterativ. În plus, o abordare standard pentru a folosi deplasamentele mai mari este de a folosi o reprezentare piramidală a unei imagini (Figura 40). Baza piramidei va conține imaginea originală iar fiecare dintre nivelele superioare se construiește printr-un filtru Gaussian aplicat pe nivelul anterior urmat de o scalare a imaginii filtrate Gaussian

cu un factor de scalare $\rho < 1$. În cazul în care ρ nu este un număr rational de forma $1/n$, aceasta scalare este însotită de o interpolare (vezi Anexa 4 pentru interpolari biliniare). Algoritmul se va rula astfel pornind de la nivelul din vârful piramidei și mergând înspre baza acesteia. Iterația corespunzătoare vârfului piramidei va lua valorile $\Delta u_0 = 0$ și $\Delta v_0 = 0$. Iterațiile corespunzătoare nivelelor inferioare se vor inițializa cu valorile deplasamentului estimat la nivelul superior înmulțite cu inversa factorului de scalara, $1/\rho$.

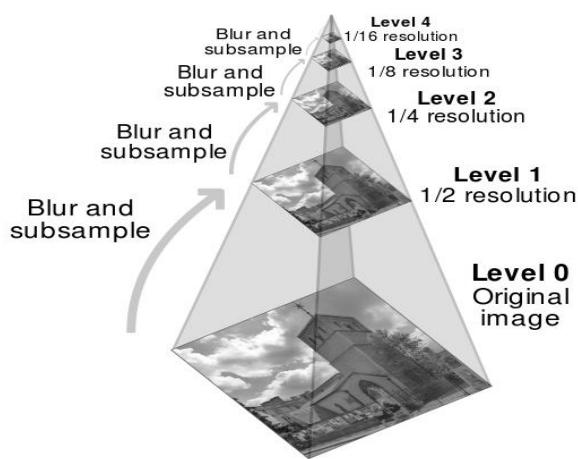


Figura 40 Reprezentare piramidală a unei imagini. Sursa:
[https://en.wikipedia.org/wiki/Pyramid_\(image_processing\)](https://en.wikipedia.org/wiki/Pyramid_(image_processing))

Avantajul acestei abordări este ca inversa matricii poate fi calculată o singură dată la fiecare nivel al piramidei, chiar dacă deplasarea este rafinată în mai multe iterații pe același nivel. De asemenea, aceste derive sunt calculate în valorile întregi ale punctelor (r, c) dacă valorile inițiale (u, v) nu sunt rafinate la nivel de subpixel. Un ultim amănunt este că prin aceste rafinări iterative valorile inițiale Δu_0 și Δv_0 pot să fie reale, iar în acest caz valorile imaginii destinație $J(r + \Delta u_0, c + \Delta v_0)$ se obțin prin interpolare (vezi Anexa 4).

2.2. Asocierea și plauzibilizarea trăsăturilor pentru odometrie vizuală

Calitatea odometriei vizuale este puternic determinată de calitatea trăsăturilor și a corespondențelor de trăsături folosite ulterior în procesul de estimare. Vom explica în continuare metodele folosite pentru robustificarea acestor potriviri de trăsături între perechi de imagini successive făcând referire la metodele de bază prezentate în mod intuitiv în capitolul 2.1. Fie ca este vorba de algoritmi de determinare a potrivirilor de trăsături pe bază de descriptori, fie că este vorba de algoritmi pe bază de metode diferențiale (flux optic) aceștia prezintă erori (potriviri eronate între trăsături) și din punct de vedere al robusteștii sistemului este bine ca aceste potriviri eronate să fie eliminate din pașii de procesare cat mai devreme.

2.2.1. Asociere circulară

Un mod de este eliminare a corespondențelor eronate de trăsături este potrivirea circulară (Figura 41). Pentru metodele diferențiale, determinăm trăsăturile q_{L1}^i în imaginea anterioară stângă pe baza derivatelor (folosind metodele explicate în capitolele 2.1.1.1 și 2.1.1.2) și determinăm apoi punctele q_{L2}^i corespondente acestora în imaginea curentă stângă folosind metoda de flux optic descrisă în Capitolul 2.1.3. Folosind punctele q_{L2}^i din imaginea curentă stângă, calculăm mai departe corespondențele acestora în imaginea curentă dreaptă q_{R2}^i , iar pe baza acestora din urmă, continuând procesul calculăm q_{R1}^i în imaginea anterioară dreaptă. În cadrul unui ultim pas, pentru a închide bucla, calculăm corespondențele punctelor din imaginea anterioară dreaptă în imaginea anterioară stângă q_{R1}^i , iar dacă rezultatul este la o distanță mai mică decât un anumit prag față de punctul de pornire q_{L1}^i atunci trăsătura se va considera în următorii pași ai algoritmului de estimare a odometriei vizuale.

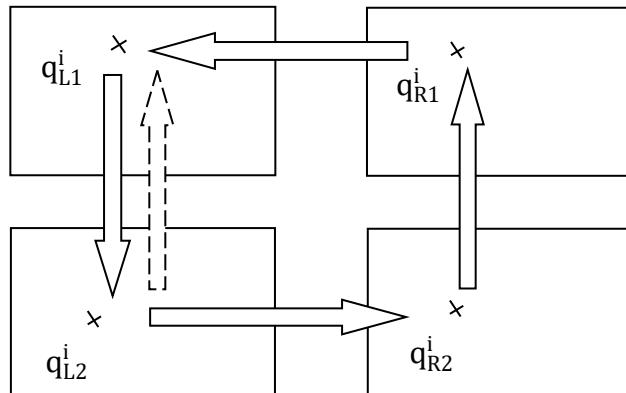


Figura 41 Estimarea circulară a corespondențelor de trăsături între două perechi de imagini stereo succesive

O abordare similară de eliminare a corespondențelor eronate de trăsături este adoptată și în cazul metodelor pe bază de descriptori, diferența fiind că determinăm trăsăturile și descriptorii acestora în toate două perechi de imagini consecutive, apoi calculăm potrivirile circular pe baza metodelor sugerate în capitolul 2.1.2. Dacă notăm trăsătura q_{L1}^i în imaginea anterioară stângă din care pornește algoritmul de determinare a potrivirilor, q_{L2}^i potrivirea acesteia în imaginea curentă stângă, q_{R2}^i în imaginea curentă dreaptă, și q_{R1}^i în imaginea anterioară dreaptă atunci condiția ca trăsătura să fie utilizată mai departe în algoritmul de estimare este ca distanța dintre descriptorii trăsăturilor q_{R1}^i și q_{L1}^i să fie sub un prag de similaritate.

În cazul metodelor mono de odometrie vizuală, un principiu similar se poate aplica. Fiind o cameră monoculară, imaginea dreaptă nu există prin urmare se calculează corespondențe de la imaginea anterioară la imaginea curentă apoi invers (săgeata punctată din Figura 41).

Atât în cadrul potrivirii circulare pe bază de descriptori cât și în cazul celor pe bază de flux optic, potrivirile dintre imaginile stângă și dreaptă pot fi la fel de bine făcute pe baza hărților de disparitate în cazul în care sistemul stereo este rectificat, reducând eventual în acest mod timpul de calcul. De asemenea, dacă imaginile sunt rectificate și totuși nu există o hartă densă de disparitate în sistem, atunci când

calculăm corespondențe stânga-dreapta de trăsături putem să reducem spațiul de căutare sau să verificăm ca potrivirile să fie pe aceeași linie (de exemplu când facem asocierea dintre q_{L2}^i și q_{R2}^i impunem $v_{L2}^i = v_{R2}^i$).

2.2.2. Inițializarea metodelor de asociere pe baza de flux optic

Observăm ca metodele de asociere pe bază de flux optic (sau metode diferențiale aşa cum le-am numit în capitolul 2.1.3) se bazează pe o rafinare iterativă a deplasamentului dintre poziția unei trăsături în imaginea anterioară și poziția acesteia în imaginea curentă. Ecuația (150) ilustrează acest proces iterativ de rafinare deplasamentului $\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}$ pe baza derivelor imaginii sursă, pe baza intensităților celor două imagini și pe baza deplasamentelor inițiale $\begin{bmatrix} \Delta u_0 \\ \Delta v_0 \end{bmatrix}$.

Considerând în continuare ca folosim metodele de flux optic în contextul odometriei vizuale, putem remarca faptul că putem să prezicem poziția unei trăsături în imaginea curentă pe baza poziției acesteia în imaginea anterioară dacă parametrii de mișcare și disparitatea (sau adâncimea) acesteia sunt cunoscute. Acest lucru este ilustrat matematic în ecuațiile (75) și (76).

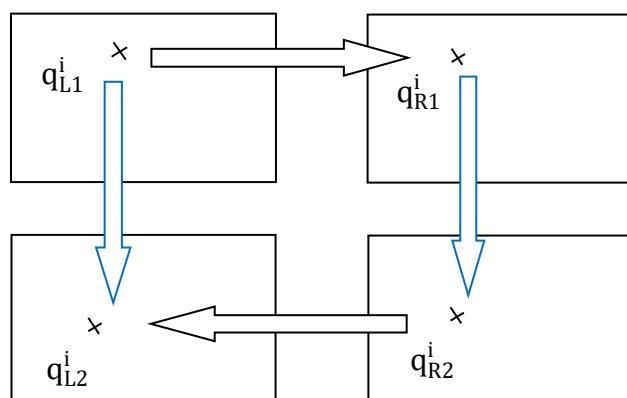


Figura 42 Modificarea asocierii circulare în cazul în care folosim predicția pentru a inițializa metodele diferențiale sau pentru a reduce spațiul de căutare. Cu albastru sunt reprezentate asocierile care beneficiază de predicție

Astfel, avem nevoie de asociere între trăsături pentru a estima odometria vizuală, și în același timp având parametrii de mișcare putem prezice aceste asociere. Soluția aleasă este de a folosi estimarea de anterioară a odometriei (de la imaginile anterioare) pentru primul pas algoritmic (Figura 34). Alternativ, la acest pas putem să folosim și parametrii de mișcare provenind de la senzori externi, în cazul sistemelor multi-senzor. De asemenea, pentru un nivel mai ridicat de precizie, putem să folosim odometria vizuală estimată pentru perechea curentă de imagini, pentru o rafinare a asocierilor de trăsături urmată de o re-estimare a parametrilor de mișcare (pașii 3 și 4 din Figura 34). În contextul asocierilor circulare prezentate în capitolul 2.2.1, ordinea în care vom determina potrivirile se schimbă (vezi Figura 42), verificările închiderii corecte a ciclului făcându-se în imaginea curentă stângă.

Cu cât inițializarea (bazată pe predicția din imaginea anterioară în cea curentă în cazul de fată) metodelor de flux optic este mai bună, cu atât numărul necesar de nivele de piramidă este mai mic. Astfel datorită faptului că folosim parametrii de mișcare estimati pentru perechea curentă de imagini, la pasul 3 (Figura 34) se vor folosi număr mai mic de nivele de piramidă, scopul acestuia fiind de a reduce numărul de potriviri false (regini de interes diferite cu intensități similare), dar și de obține o precizie mai bună la nivel de sub-pixel. De remarcat că mecanismele de robustificare introduse în procesul de estimare a odometriei vizuale de pasul 2 (ex: RANSAC [44]) vor avea rolul de a elimina la rândul lor o parte dintre asocierile false, care ulterior la pasul 3 pot fi transformate în asociere corecte datorită unei bune inițializări a algoritmilor de potrivire. Dacă din motive de eficiență computațională la pasul 1 estimăm doar rotația atunci pentru rafinarea de la pasul 3 putem să folosim aceasta rotație împreună cu translația estimată anterior.

Această idee de predicție pe baza odometriei vizuale poate fi utilizată și în cazul metodelor pe bază de descriptori, pentru reducerea spațiului de căutare. Căutând într-o vecinătate a poziției prezise, vom reduce probabilitatea de asociere false și vom îmbunătăți timpul de calcul necesar.

2.2.3. Plauzibilizarea pe bază de perechi de trăsături

În urma procesului de potrivire circulară de trăsături și a îmbunătățirii acestuia pe baza unei bune inițializări în cazul metodelor de asociere pe bază de metode diferențiale sau a reducerii spațiului de căutare în cazul metodelor pe bază de descriptori, obținem un set de trăsături candidate pentru procesul de estimare al mișcării (realizat inițial la pasul 2, și opțional la pasul 4, Figura 34). Deoarece există șanse ca anumite trăsături să fie în continuare eronate, sau ca anumite trăsături să se găsească pe obiecte aflate în mișcare definim noi măsuri de evaluare a corectitudinii acestora. Observăm că distanțele dintre coordonatele 3D ale perechilor de trăsături sunt invariante în timp dacă acestea se găsesc pe obiecte aflate în mișcare, observație pe care o putem exprima matematic astfel:

$$\left| \sqrt{(X_1^i - X_1^j)^2 + (Y_1^i - Y_1^j)^2 + (Z_1^i - Z_1^j)^2} - \sqrt{(X_2^i - X_2^j)^2 + (Y_2^i - Y_2^j)^2 + (Z_2^i - Z_2^j)^2} \right| < \varepsilon_d^{i,j} \quad (151)$$

Datorită erorilor de reconstrucție stereo, vom exprima această constrângere doar pentru acele puncte care sunt aproape de camera, $\varepsilon_d^{i,j}$ alergându-se în funcție de pragul de apropiere.

O altă constrângere similară poate fi exprimată pe baza unghiurilor dintre perechi de trăsături și centrul camerei. Așa cum am arătat în 1.1.3.3 unghiul dintre perechi de puncte este invariant la mișcarea camerei dacă aceste puncte sunt la distanță suficient de mare față de cameră. Pe baza ecuației (29) vom avea:

$$\cos(\theta_{i,j}^1) = \frac{(K^{-1} \cdot q_{L1}^i)^T \cdot (K^{-1} \cdot q_{L1}^j)}{\|K^{-1} \cdot q_{L1}^i\| \cdot \|K^{-1} \cdot q_{L1}^j\|} \quad (152)$$

$$\cos(\theta_{i,j}^2) = \frac{(K^{-1} \cdot q_{L2}^i)^T \cdot (K^{-1} \cdot q_{L2}^j)}{\|K^{-1} \cdot q_{L2}^i\| \cdot \|K^{-1} \cdot q_{L2}^j\|} \quad (153)$$

$$|\cos(\theta_{i,j}^1) - \cos(\theta_{i,j}^2)| < \varepsilon_\theta^{i,j} \quad (154)$$

Ecuația (154) este valabilă pentru punctele suficient de îndepărtate de cameră, valoarea lui $\varepsilon_{\theta}^{i,j}$ depinzând de cât de apropiat alegem acest prag. În [10] am introdus o metodă de estimare a unui prag adaptiv în funcție de viteza de deplasare (preluată de la senzori externi sau de la estimarea anterioară a vitezei) și distanțeze până la pozițiile 3D ale celor două trăsături. Cu cât viteza este mai mare cu atât pragul va deveni mai permisiv, cu cât viteza este mai mică cu atât pragul va deveni mai restrictiv. Cu cât distanța până la cea mai apropiată trăsătură este mai mare, cu atât pragul va fi mai restrictiv, odată cu scăderea acesteia pragul adaptiv devine mai permisiv. Valoarea lui Φ este determinată experimental.

$$\varepsilon_{\theta}^{i,j}(v, Z_1^i, Z_1^j) = \Phi \cdot \frac{v \cdot \Delta t}{\min(Z_1^i, Z_1^j)} \quad (155)$$

2.3. Metoda liniară versus metode non-liniare

2.3.1. Stereo

Capitolul 2.1 detaliază principiile de bază ale celor două alternative de determinare (2.1.1) și de asociere de trăsături, pe bază de descriptori (2.1.2) respectiv pe bază de metode diferențiale (2.1.3). În plus, Capitolul 2.2. descrie utilizarea metodelor din capitolul 2.1 în contextul specific al odometriei vizuale, detaliind aspecte precum asocierea circulară, reducerea spațiului de căutare al trăsăturilor și plauzibilizarea acestora, conducând în final la o mai bună precizie a estimării odometriei vizuale. Asocierea de trăsături se efectuează practic în pasul 1 al schemei bloc din Figura 34.

Având asocieri de trăsături în perechi de imagini consecutive se poate face o estimare a mișcării (odometrie vizuală), lucru efectuat în pasul 2 al diagramei conceptuale din Figura 34.

Majoritatea metodelor pe bază de asocieri de trăsături din literatură se bazează pe formularea unei funcții de cost care include parametrii de rotație și translație ca și cantități nerecunoscute. Aceste

sunt estimate prin metode de optimizare ne-liniară. Capitolul 1.2 descrie trei modele matematice prin care se poate formula funcția de cost. Capitolul 1.2.1 descrie o formulare pe baza alinierii de puncte 3D, abordare oarecum naivă și care nu este folosită în practica datorită gradului mare de influență a erorilor de reconstrucție 3D, dar pe care o includem în această lucrare pentru a oferi o mai bună intuiție asupra problemei. Capitolele 1.2.2 și 1.2.3 prezintă două modele de formulare a funcției de cost pe baza unor cantități reziduale exprimate în planul imagine, pe baza proprietăților de transfer ale tensorului trifocal (55)(52)(56) respectiv pe baza minimizării erorii de reproiecție (62).

În [5] am propus o metodă de estimare a odometriei vizuale prin estimare de metode neliniare. Accentul este pus pe exprimarea funcției de cost pe baza erorilor de reproiecție dar în cadrul rezultatelor experimentale este făcută o comparație și cu o abordare pe bază de tensor trifocal, inspirată de [54]. Pentru a elimina corespondențele de trăsături eronate sau cele aflate pe obiectele în mișcare am folosit RANSAC [44]:

A - selectăm subseturi de trăsături care rezultă în urma procesului de asociere circulară (2.2.1, 2.2.2) și plauzibilizare (2.2.3)

B - Formulăm o funcție de cost conform ecuațiilor (56) pentru metoda pe bază de tensor trifocal, respectiv (61) sau (62) pentru metoda pe bază de erori de reproiecție. Estimarea parametrilor de mișcare (rotație, translație) se face apoi folosind algoritmul Levenberg-Marquardt [212][213].

C - determinăm numărul de trăsături care verifică parametrii estimări la pasul B (eroarea reziduală sub un prag).

Etapele A-B-C se execută de un număr suficient de mare de ori, iar parametrii de mișcare care sunt verificate de cel mai mare set de asociere de trăsături la C vor reprezenta estimarea mișcării. Conceptual, aceștia fac parte din pasul 2 în diagrama din figura Figura 34. Tot în [5] am propus o metodă de plauzibilizare a estimării pe baza observației că mișcarea laterală este corelată cu unghiul yaw, eliminând estimările pentru care această constrângere nu este respectată. Figura

43 arată cum sunt filtrate trăsăturile pentru estimarea odometriei pe bază de trăsături.

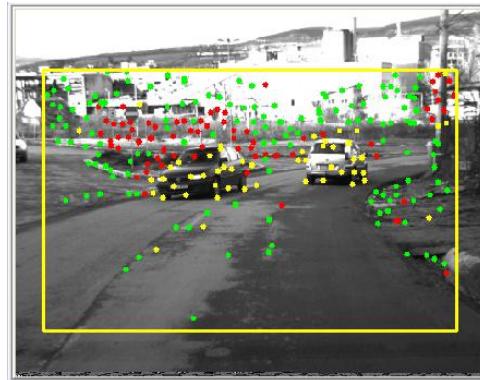


Figura 43 Filtrarea trăsăturilor pentru estimarea odometriei vizuale. Cu roșu sunt reprezentate trăsăturile care sunt eliminate prin asociere circulară, cu galben sunt reprezentate trăsăturile eliminate prin procesul de RANSAC, cu verde sunt reprezentate trăsăturile care verifică mișcarea estimată

Figurile următoare ilustrează o parte dintre rezultatele obținute în [5]. Figura 44 ilustrează faptul că estimarea parametrilor de odometrie vizuală poate fi făcută și o dată la fiecare două imagini stereo consecutive, deci din două în două flame-uri, aceasta fiind practic echivalentul unei capabilități de estimare a odometriei la viteze mai mari.

Figura 45 prezintă estimarea parametrilor de translație cu RANSAC, deci prin pașii A-B-C descriși mai sus, comparată cu o estimare fără RANSAC, deci formulând problema de optimizare pe tot setul de trăsături rezultat în urma procesului de asociere circulară și plauzibilizare.

Figura 46 reprezintă traectoria mișcării obținute prin integrarea valorilor reprezentate pe graficele din stânga în Figura 44 și Figura 45, conducând de două ori în jurul parcului reprezentat în partea stângă a figurii. Observăm diferențele foarte mici dintre prima parcurgere și cea de-a doua, diferențe care sunt influențate de doi factori: acumularea succesivă de erori de estimare și faptul că în a doua parcurgere nu s-a condus exact prin același loc ca și la prima parcurgere (drumul având

mai multe benzi pe alocuri, și achiziția fiind făcută în condiții standard, și cu prezența altor participanți la trafic).

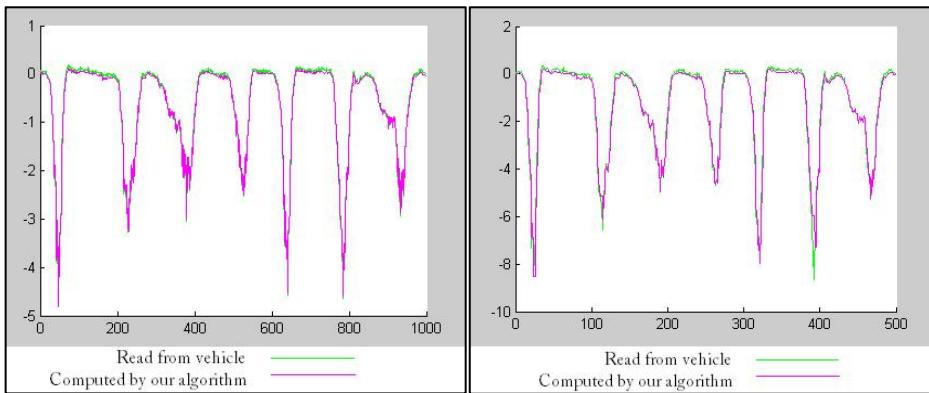


Figura 44 Estimarea unghiului yaw (stânga) pentru imagini stereo consecutive, respectiv sărind 1 din 2 imagini.

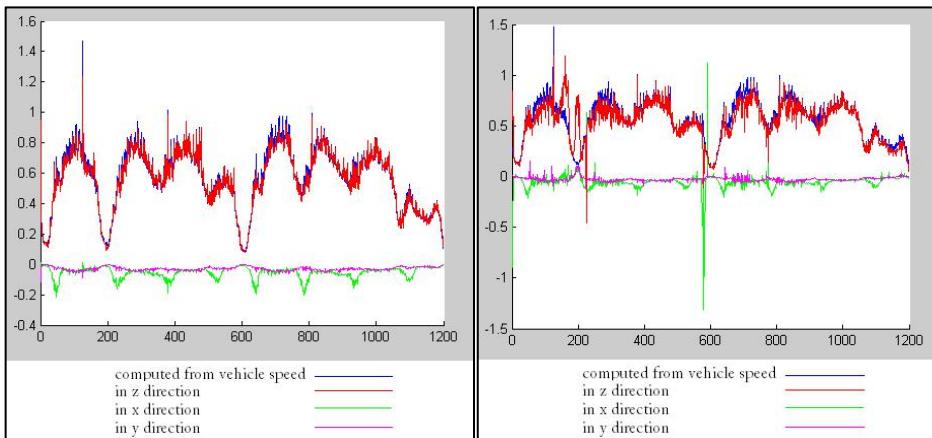


Figura 45 Estimarea translației cu metode ne-lineare (stânga). În dreapta este ilustrată estimarea făcută pe tot setul de trăsături, fătă a folosi RANSAC

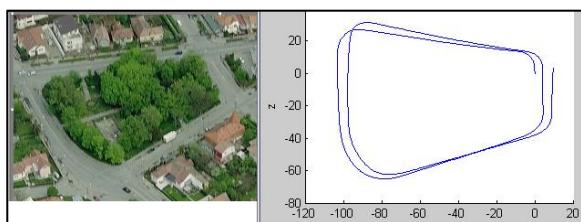


Figura 46 O reprezentare top view a trajectoriei obținute prin integrarea valorilor odometrie vizuale estimate pe un set de 1200 de imagini consecutive (dreapta) la parcurgerea de două ori a ariei geografie din stânga

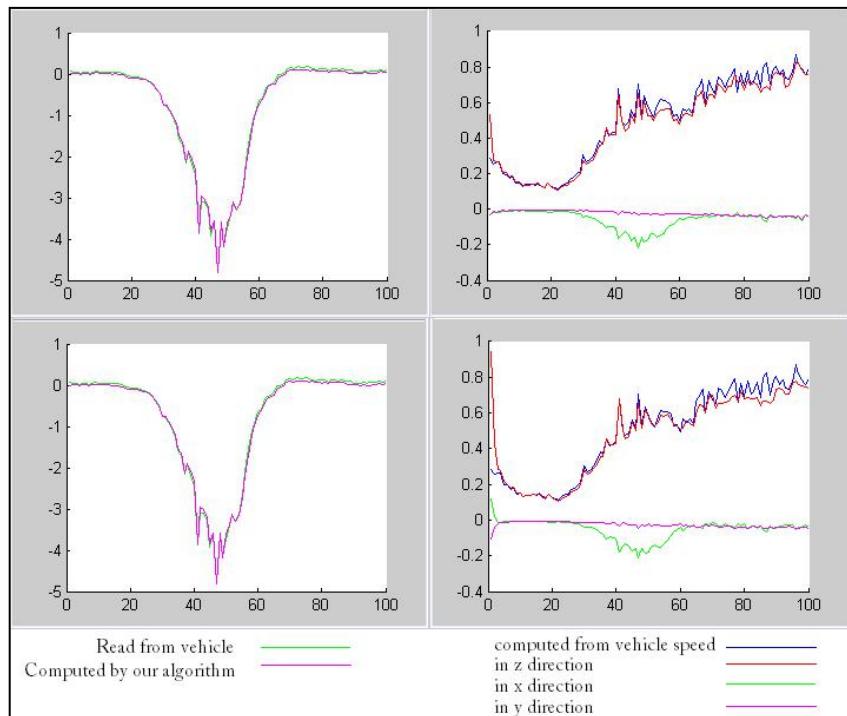


Figura 47 Comparație cantitativă între estimarea pe bază de tensorul trifocal (graficele de jos) și metoda ne-liniară de minimizare a erorii de reproiecție (graficele de sus). Pe coloana din stânga este unghiul yaw, exprimat în grade inclusiv valorile citite de pe magistrala CAN a mașinii, iar e coloana din dreapta valorile translației exprimate în metri

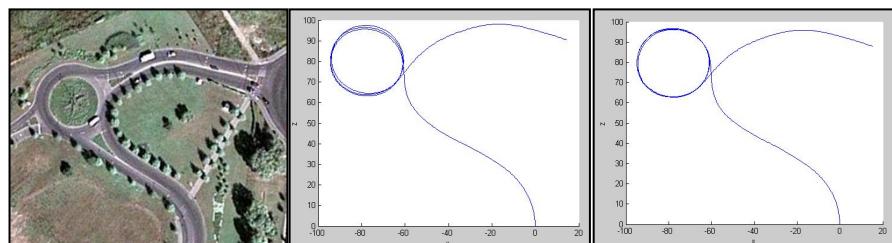


Figura 48 Traiectoria obținută prin metode ne-liniare pe baza erorii de re-proiecție (mijloc) respectiv pe baza tensorului trifocal (drepta) la parcurgerea de trei ori al unui sens giratoriu (stânga)

O comparație calitativă dintre formularea funcției de cost pe baza proprietăților de transfer ale tensorului trifocal (55)(52)(56) respectiv pe baza minimizării erorii de reproiecție (62) este ilustrată în Figura 47, ilustrând practic faptul ca, deși mai complexă din punct de vedere al

modelului matematic aceasta nu produce rezultate superioare. Aceeași concluzie este ilustrată în Figura 48.

În [6] am propus o metodă de simplificare a modelului non-linear de estimare din [5]. Practic etapa B din succesiunea A-B-C de mai sus este înlocuită cu formularea unui sistem de ecuații și rezolvarea acestuia. Modelul matematic propus în [6] este prezentat pe larg în Capitolul 1.3. Pornind de la reprezentarea simplificată a matricii de rotație din ecuația (63), determinarea odometriei vizuale se reduce la rezolvarea unui sistem supradeterminat de ecuații. Pentru fiecare asociere de trăsături între perechi de imagini stereo succesive obținem un set de trei ecuații precum cel din (74) sau (78). Dacă la pasul A vom selecta un număr de K trăsături, atunci la pasul B vom obține un sistem liniar de $3K$ ecuații, care se rezolvă prin metode consacrate pentru a determina parametrii de mișcare. Determinarea trăsăturilor care verifică parametrii de estimații, specificată la pasul C, se face prin comparația pozițiilor trăsăturilor din imaginea curentă cu poziția prezisă a acestora pe baza parametrilor de mișcare conform ecuațiilor (70)(71)(75)(76).

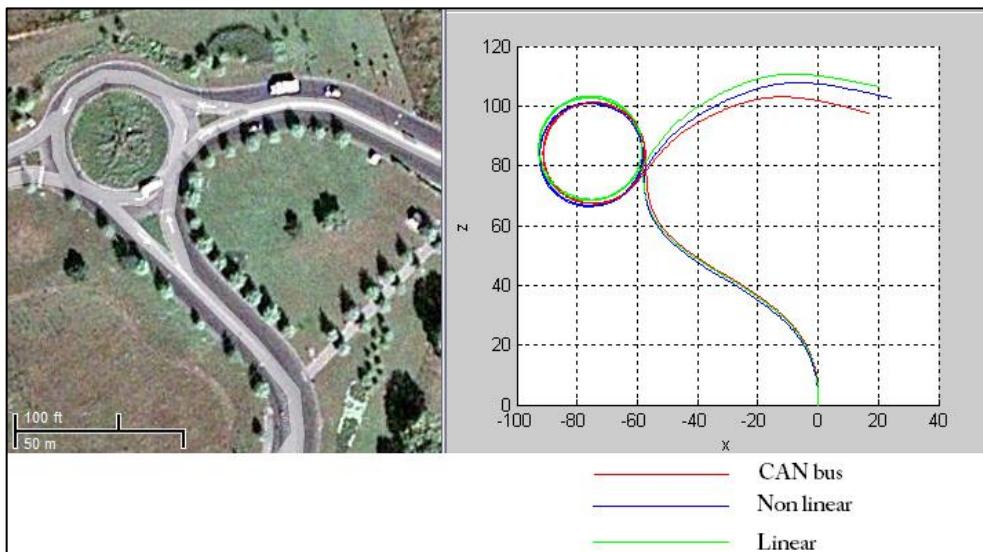


Figura 49 O comparație a traiectoriilor generate cu metoda liniară (verde) și metoda ne-lineară (albastru). Includem ca și referință traiectoria obținută pe baza unor valori de translație și yaw rate citite de pe magistrala CAN a mașinii

Figura 49 prezintă o comparație între traectoria obținută prin compunerea rotațiilor și translațiilor dintre imagini constitutive obținute prin metoda liniară (verde) și metode non-linare (albastru). Observăm diferențe minore între cele două traекторii. Ca și referință am inclus și o traectorie (desenată cu roșu) generată pe baza unghiului yaw și a translației cîte de magistrală CAN am mașinii. Figura 50 prezintă o comparație a unghiurilor pentru primele 100 de imagini de pe traectoria prezentată în Figura 49. Observam ce în general există o suprapunere foarte bună între cele două estimări.

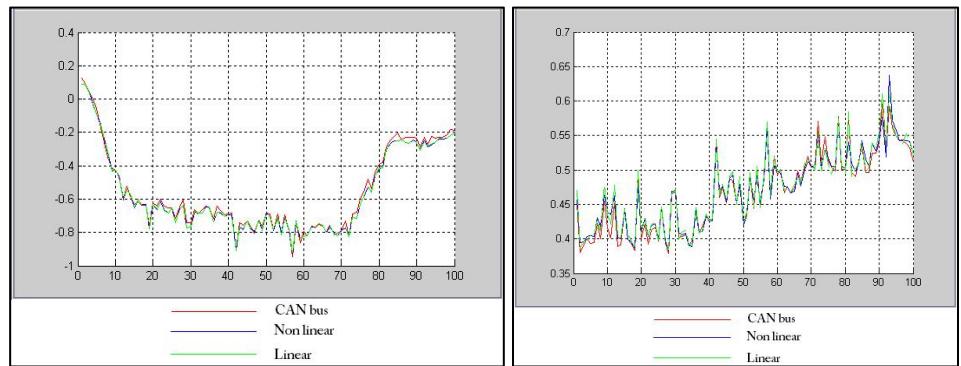


Figura 50 Comparație între unghiul yaw în grade (stânga) și translația longitudinală în metrii (dreapta) pentru primele 100 de imagini de pe traectoria din Figura 49

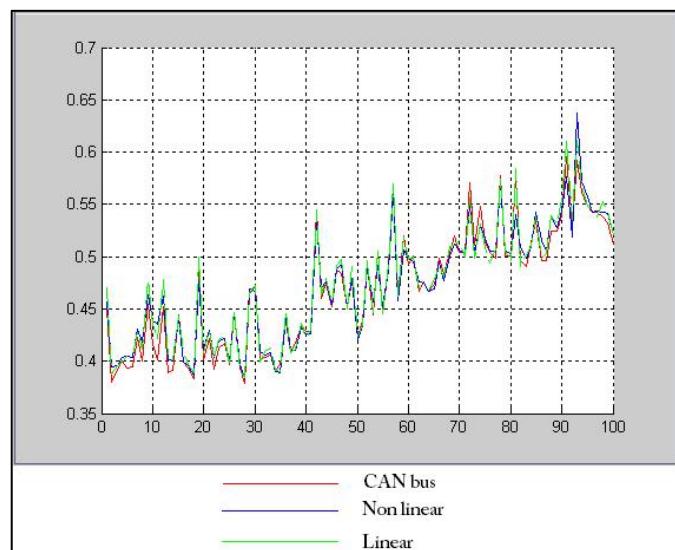


Figura 51 Estimarea ughiului yaw exprimă în grade din 3 în 3 imagini pentru o secvență de 300 din 900 imagini succesive

Un alt experiment propus în [6], și care merită readus în lucrarea prezentă este ilustrat în Figura 51, unde ilustrăm faptul a putem obține o estimare consistentă chiar dacă sărim 2 din 3 imagini succesive. Acest lucru arată faptul că astfel de metode funcționează și la viteze mai mari ale mașinii echipate cu senzori video. Tot în [6] am făcut analiză a gradului de robustețe la precizia estimare a corespondențelor dintre trăsături. Pornind de la setul maximal de trăsături pe baza căruia se face estimarea mișcării acesta poate fi alterat prin introducerea de zgomot. Facem apoi o nouă estimare pe baza setului de corespondențe afectate de zgomot și observăm gradul de stabilitate al metodei.

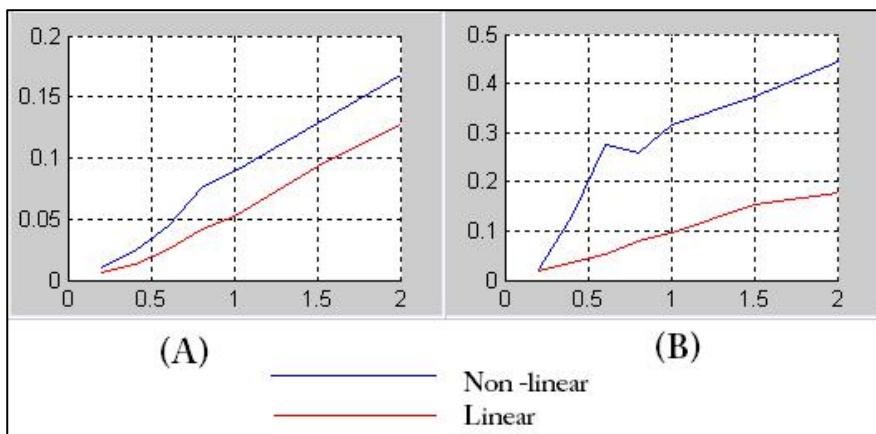


Figura 52 Deviația standard (axa verticală) a estimării relativ la estimarea originală (neafectată de zgomot de asociere). Deviația standard a zgomotului introdus în asocierile spațiale (disparitate) este pe axa orizontală. (A) - deviația standard a unghiului yaw (exprimată în grade). (B) - deviația standard a translației în lungul axei longitudinale (exprimată în metri)

Figura 52 și Figura 53 prezintă deviațiile standard pentru unghiul yaw și pentru translația longitudinală pentru zgomot cu deviații standard cuprinse între 0.25px și 2px. Experimentul e efectuat pentru un set de 100 de imagini succesive. Figura 52 reprezintă cazul în care zgomotul este introdus în asocierile spațiale la fiecare dintre cele 100 de imagini, iar Figura 53 reprezintă comportamentul rezultat în urma introducerii zgomotului în asocierile temporale. Concluzionăm ca metoda liniară este mai robustă la erori de asociere prin faptul că erorile cauzate de zgomotele de asociere sunt în general mai mici decât

în cazul metodei ne-liniare (graficul u roșu este tot timpul situat sub graficul cu albastru).

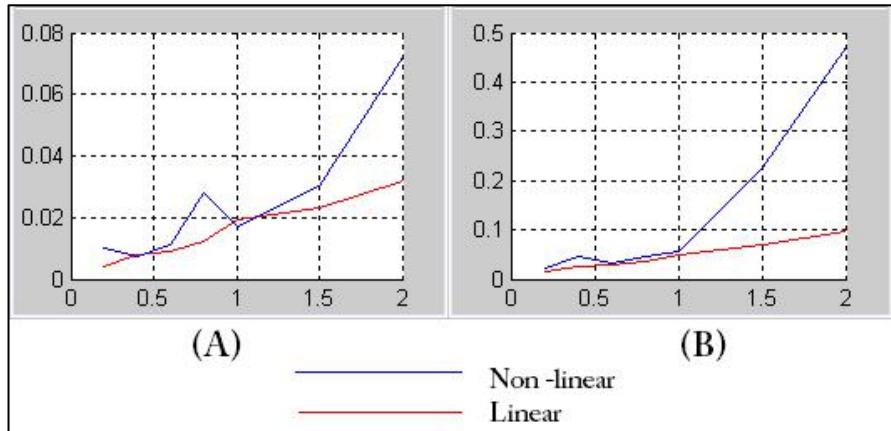


Figura 53 Deviația standard (axa verticală) a estimării relativ la estimarea originală (neafectată de zgomot de asociere). Deviația standard a zgomotului introdus în asocierile temporale (flux optic) este pe axa orizontală. (A) - deviația standard a unghiului yaw (exprimată în grade). (B) - deviația standard a translației în lungul axei longitudinale (exprimată în metrii)

Fără a intra în detalii referim [7] unde am prezentat o metodă de paralelizare a metodelor liniare de odometrie vizuală, atât la nivel de determinare de trăsături și asocieri între trăsături cât și la nivel de execuție de iterații RANSAC. Concluzie prezentată și argumentată în [7] este că având la dispoziție o arhitectură paralelă de calcul, timpul necesar strict pentru a efectua partea de optimizare pentru estimarea odometriei vizuală este neglijabil comparativ cu timpul de detecție și asociere de trăsături. Referim [7] pentru mai multe detalii leg de paralelizarea algoritmului le care l-am introdus inițial în [6].

2.3.2. Mono

Utilizarea matricii esențiale (vezi capitolul 1.1.3.4 și capitolul 1.5.2) este varianta standard de a determina rotația și direcția de translație a unei camere. Aceasta este o metoda neliniară care presupune optimizarea funcției de cost descrise în capitolul 1.5.2.. Deoarece această optimizare este consumatoare de resurse

computaționale am investigat în [12] posibilitatea de a face o estimare a matricii de rotație pe baza punctelor de la distanță. Pentru filtrarea punctelor de la distanță constatăm că unghiurile formate de perechi de trăsături și centrul camerei (vezi 1.1.3.3) se păstrează aproximativ constante de la un frame la următorul. Intuitiv, cu cât distanța la care se află cele două trăsături este mai mare cu atât valoarea lui $|\theta_{i,j}^1 - \theta_{i,j}^2|$ este mai mică (Figura 54).

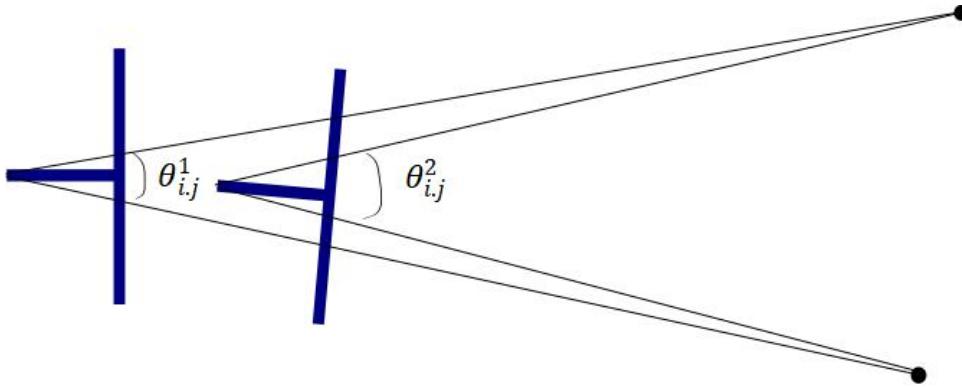


Figura 54 Unghiul dintre perechi de trăsături pentru imagini succesive

Prezentăm în continuare un experiment făcut în [12] pentru o mai bună intuiție asupra valorilor $|\theta_{i,j}^1 - \theta_{i,j}^2|$. Considerând un set de trăsături corespondente în imagini succesive putem calcula valorile $|\theta_{i,j}^1 - \theta_{i,j}^2|$ pentru fiecare pereche de trăsături. Figura 55 indică prin segmente albastre perechile de trăsături având cele mai mici valori, respectiv cele mai mari valori pentru $|\theta_{i,j}^1 - \theta_{i,j}^2|$. Se observă că se poate face o filtrare a punctelor de la distanță pe baza acestor diferențe de unghiuri dintre perechi de trăsături succesive. După determinarea unui set de potriviri de trăsături folosind metodele generale descrise în 2.1, astăzi cum este descris în [12], algoritmul propus implică următorii pași pentru fiecare iterație de tip RANSAC:

A - Selectam o pereche aleatoare de trăsături, pentru care valoarea $|\theta_{i,j}^1 - \theta_{i,j}^2|$ este suficient de mică

B - Se calculează valoarea parametrilor de rotație conform modelului matematic propus în [12] și dedus în detaliu în capitolul 1.5.1. Folosim ecuațiile (97) sau (100)

C - Pentru rotația obținută se evaluează câte puncte respectă rotația estimată, folosind ecuațiile (95)(96) sau (98)(99)

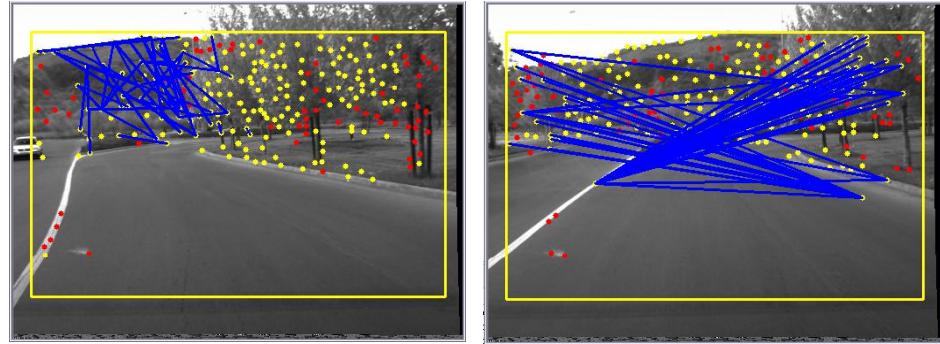


Figura 55 Primele 50 de perechile de trăsături pentru care valoarea lui $|\theta_{i,j}^1 - \hat{\theta}_{i,j}^1|$ este cea mai mică (stânga) respectiv cea mai mare (dreapta). Punctele galbene sunt cele luate în considerare, punctele roșii fiind eliminate pe baza metodelor descrise în 2.1

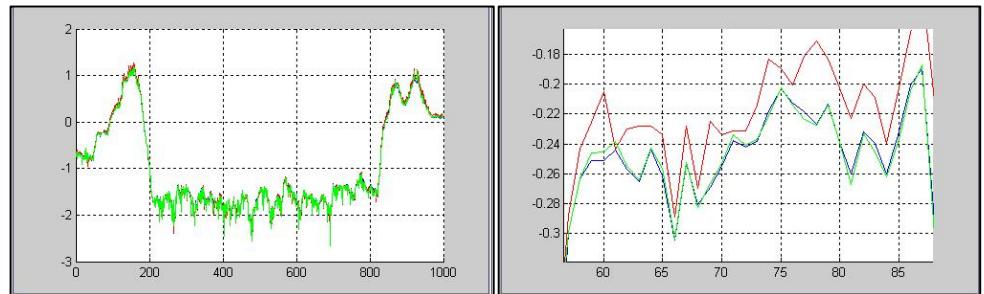


Figura 56 Estimarea unghiului yaw (reprezentat în grade pe axa verticală) pentru un set de 1000 de imagini succesive. Cu roșu este reprezentată o valoare de referință obținută de pe magistrala CAN a mașinii de test, cu albastru este estimarea obținută cu o metodă stereo iar cu verde este estimarea obținută cu metoda mono descrisă în capitolul curent. Figura din dreapta prezintă o variantă mărită a figurii din stânga

Matricea estimată la pasul B, din care rezultă cel mai mare număr de trăsături la pasul C va reprezenta cea mai bună estimare a matricii de rotație. Observăm ca algoritmul propus este o variantă modificată de RANSAC [44] prin faptul că pentru selecția aleatoare de perechi de trăsături pună condiția suplimentară ca valoarea lui $|\theta_{i,j}^1 - \hat{\theta}_{i,j}^1|$

$\theta_{i,j}^1$ să fie mică pentru a selecta trăsături aflate la distanță. Figura 57 prezintă exemple de trăsături selectate la punctul C al algoritmului.

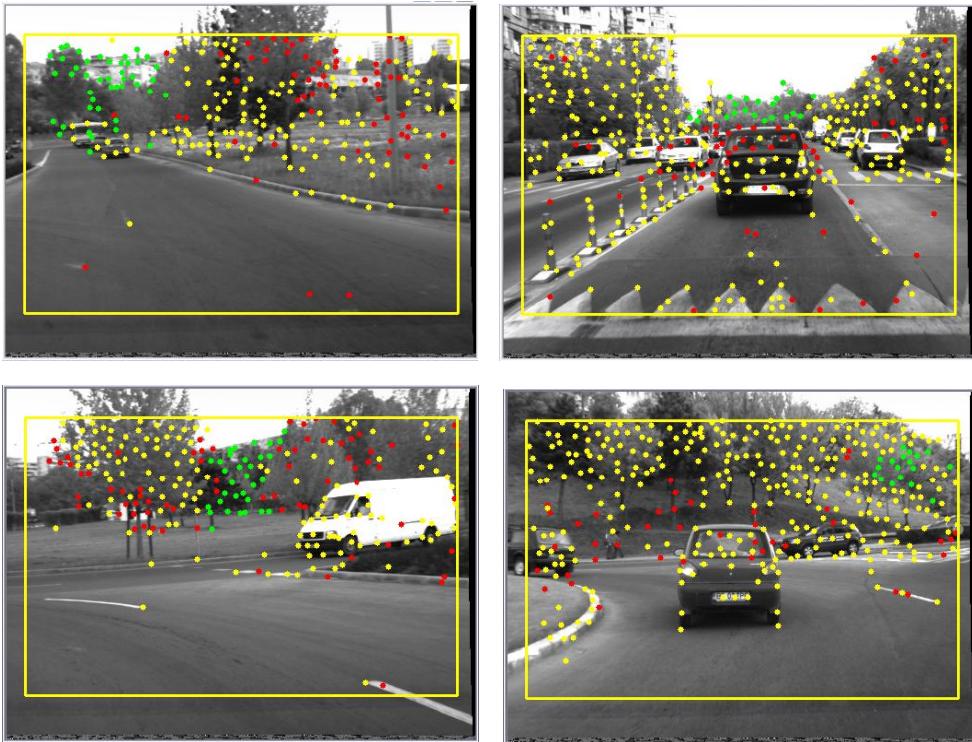


Figura 57 Trăsăturile de la distanță marcate cu verde sunt folosite pentru estimarea rotației

În Figura 56 sunt prezentate rezultatele obținute pe o secvență de 100 de imagini succesive. Un alt experiment pe care l-am efectuat și prezentat în [12] este trecerea peste un limitator de viteză. Rezultatele sunt ilustrate în Figura 58.

Deoarece experimentele sunt efectuate într-un sistem stereo, avem posibilitatea de a evalua estimarea rotației cu diverse condiții de filtrare a trăsăturilor pe baza distanțelor acestora. În Figura 59 prezentăm estimarea unghiului yaw pentru 150 de imagini succesive, în condițiile în care consideram în algoritm doar trăsăturile mai apropiate decât un anumit prag. Observăm astfel comportamentul algoritmului și erorile care pot să apară în scene în care nu avem puncte la distanțe foarte mari față de cameră. De remarcat faptul că erorile din Tabel 4 sunt

dependente de parametrii camerei (rezoluție, distanțe focale, calitatea lentilelor și precizia cu care modelul matematic folosit pentru eliminarea distorsiunilor poate modela realitatea). Parametrii diferenți ai camerei vor duce la erori diferite de estimare atunci când nu avem puncte suficiente de îndepărtate. Așa cum vom detalia ulterior în cadrul acestei lucrări aceasta estimare a rotației poate fi folosită ca și inițializare a unor altor metode de odometrie, crescând precizia acestora, dar și reducând timpul de estimare (sau calculele care trebuie efectuate pentru fiecare estimare).

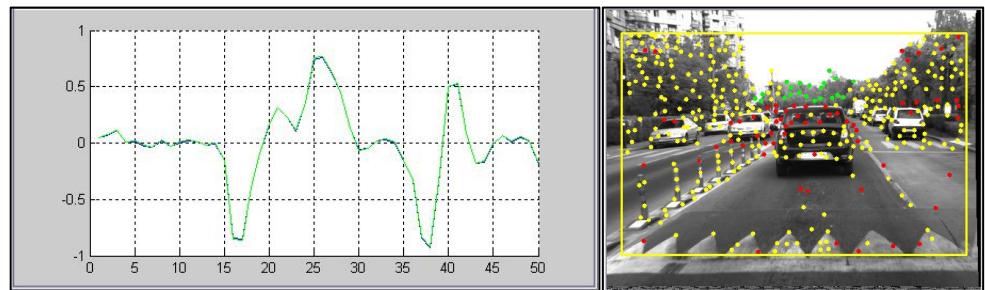


Figura 58 Estimarea unghiului pitch (stânga) la trecerea peste limitatorul de viteză din imaginea dreaptă. Cu albastru este estimarea obținută cu o metodă stereo iar cu verde este estimarea obținută cu metoda mono descrisă în capitolul curent

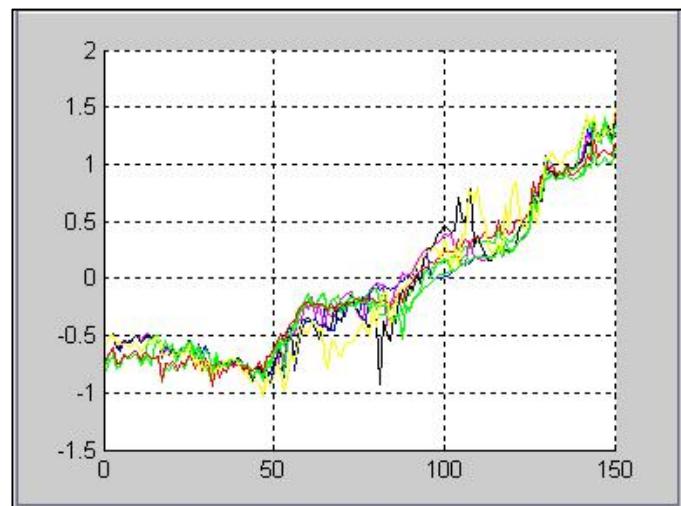


Figura 59 Estimarea unghiului yaw (în grade) pentru diverse praguri de adâncime maximă a trăsăturilor. Culoarea roșie indică valoarea unghiului yaw citit de la magistrala CAN a mașinii. Fiecare culoare corespunde unei valori din Tabel 4

Distanța (metrii)	Eroare (grade)	Culoare în Figura 59
Infinite	0.0503	verde
120m	0.1139	verde
80m	0.1369	verde
60m	0.1466	albastru
50m	0.1519	violet
40m	0.1666	negră
30m	0.2017	galben

Tabel 4 Erorile de estimare a unghiului yaw relativ la valoarea citită de pe magistrala CAN a mașinii. Estimarea se face considerând doar trăsăturile aflate la o distanță mai mică decât pragul indicat pe prima coloană a tabelului. Culoarea indicată pe ultima coloană este folosită pentru graficul din Figura 59

2.4. Robustificarea pe bază de descriptori invariante la iluminare



Figura 60 Exemple de scenarii dificile pentru odometria vizuală

Estimarea odometriei vizuale prezintă anumite provocări atunci când imaginile pe baza cărora aceasta este efectuată sunt influențate de factori externi (Figura 60) precum condițiile de iluminare care afectează senzorii video (ex: suprasaturare), condiții meteorologice (ceată, ploaie, ninsoare) sau obiecte mișcătoare care ocluzionează parti predominante din scenă.

În [10] am propus o metodă de robustificare pe baza transformatei rank inițial propusă în [186] pentru a îmbunătăți reconstrucția stereo atunci când apar condiții diferite de iluminare între camera stângă și camera dreaptă dintr-un sistem stereo. Propunerea noastră din [10] calculează trăsături pe baza derivatelor imaginii (vezi 2.1.1.1, 2.1.1.2) în perechi de imagini consecutive. În același timp se calculează transformata rank a acestor imagini conform descrierii din capitolul 2.1.2, ecuația (131), folosind o fereastră de dimensiune 9 ($s_r = 4$ și $s_c = 4$). Figura 61 ilustrează vizual cum arată această reprezentare, scalată pentru a lua valori între 0 și 255. Imaginea rezultată are proprietatea de a fi invariantă la schimbări de iluminare între imagini succesive.

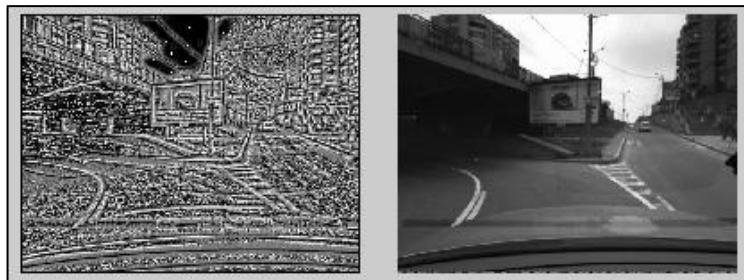


Figura 61 Transformata rank (stânga) pentru o imagine de trafic (dreapta)

Pentru a face potriviri între trăsături vom folosi suma diferențelor absolute $RSAD(u^i, v^i, u^j, v^j)$ calculate pe o fereastră de dimensiune 7px în jurul trăsăturilor determinate în cele două perechi de imagini succesive conform ecuației (132). Deoarece imaginile cu care lucrăm sunt rectificate determinarea de corespondențe între imaginea stânga și dreapta se face reducând spațiul de căutare la aceeași linie din imagine. În cazul în care nu avem o calibrare perfectă se poate permite un prag de toleranță de 1-2px între coordonatele u_{Lx}^i și u_{Rx}^i $x \in \{1,2\}$. Optimizarea spațiului de căutare pentru corespondențele în timp

(indicate cu albastru în Figura 42) se face prin reducerea spațiului de căutare în jurul poziției prezise conform ecuațiilor (70)(71)sau (75)(76). descrierii din capitolul 1.3.

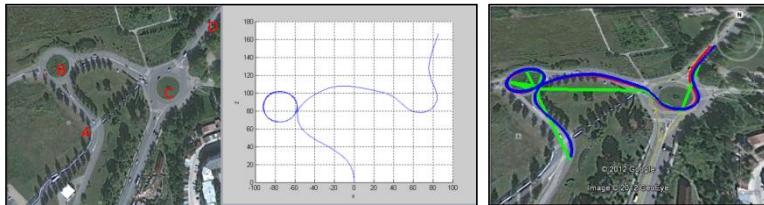


Figura 62 Comparație între traiectoria generată pe baza odometriei vizuale obținute din asocieri de trăsături obținute prin metode diferențiale (albastru în dreapta) și asocieri pe baza de rank transform (roșu în dreapta). Graficul din mijloc este echivalentul liniei roșii din imaginea dreapta. Cu verde este reprezentată o linie poligonala care unește coordonatele GPS obținute pe ruta A-B-C-D, și făcând mai multe rotații în sensul giratoriu B

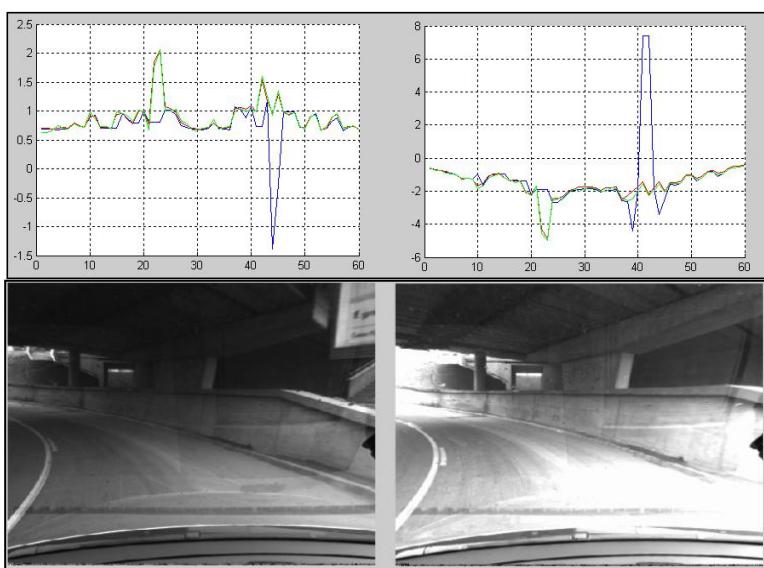


Figura 64 Partea de sus arata evoluția unghiului yaw la intrarea sub un pod. Imaginile în rândul de jos sunt două imagini succesive având condiții de iluminare diferite datorită trecerii de la soare la umbră. Cu roșu este reprezentat unghiul citit de la magistrala CAN a mașinii, cu verde este reprezentată estimarea din metode de asociere diferențiale, cu verde este estimarea din asocieri de trăsături pe bază de rank transform

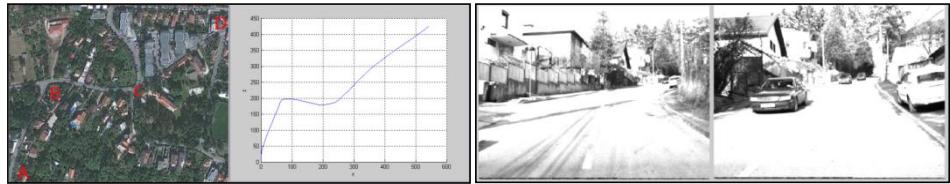


Figura 63 Traекторia obținută prin integrarea odometriei vizuale obținute pe traseul A-B-C-D din asociere de trăsături pe bază de rank transform. Traseul prezintă zone cu probleme de iluminare precum cele din dreapta

Figurile de mai sus ilustrează beneficiul adus de trăsăturile invariante la iluminare. Pentru mai multe evaluări experimentale și concluzii referim articolul nostru anterior [10].

2.5. Modelarea erorilor de odometrie vizuală

Sumarizăm în acest capitol rezultatele pe care le-am publicat anterior legat de modelarea explicită a erorilor de odometrie vizuală [10] Modelul matematic propus în [10] constă în determinarea unor formule directe pentru a calcula matricea de covarianță a estimării odometriei vizuale, cunoscând erorile pe care le avem în datele de intrare, și anume precizia de localizare a trăsăturilor pe baza cărora facem estimarea odometriei. Acest model matematic este descris pe larg în capitolul 1.4, și practic se poate rezuma la o formulă directă prin care calculăm matricea de covarianță pentru modelul liniar de estimare propus în capitolul 1.3, formula care este explicit punctată în (93) și pe care din considerente de stabilitate numerică o regăsim înglobând numiști factori de normalizare în (94).

În Figura 65 din stânga am reprezentat translația obținută pe baza modelului liniar de odometrie vizuală (vezi capitolul 1.3), considerând atât o estimare care folosește toate trăsăturile din scenă (verde) cât și două versiuni care consideră doar trăsături aflate la o distanță mai mare decât un anumit prag (15m albastru, 30m violet). Observăm în Figura 65 din dreapta că deviația standard crește odată cu creșterea pragului de filtrare al trăsăturilor, lucru așteptat deoarece erorile de reconstrucție 3D sunt mai mari pentru trăsăturile de la

distanță. Cu alte cuvinte, metoda propusă va estima o deviație standard mai mare a translației atunci când nu avem trăsături în apropierea camerei.

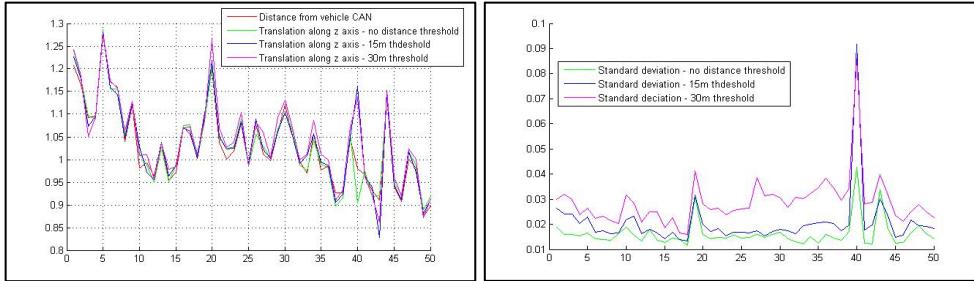


Figura 65 Stânga - translația estimată în lungul axei longitudinale în metrii (axa verticală) pentru 50 de imagini consecutive (axa orizontală) cu diverse praguri de distanță pentru filtrarea trăsăturilor. Dreapta - deviația standard pentru translația longitudinală exprimată în metrii (axa verticală) pentru cele 50 de imagini consecutive din stânga (axa orizontală) și pentru diverse praguri de distanță

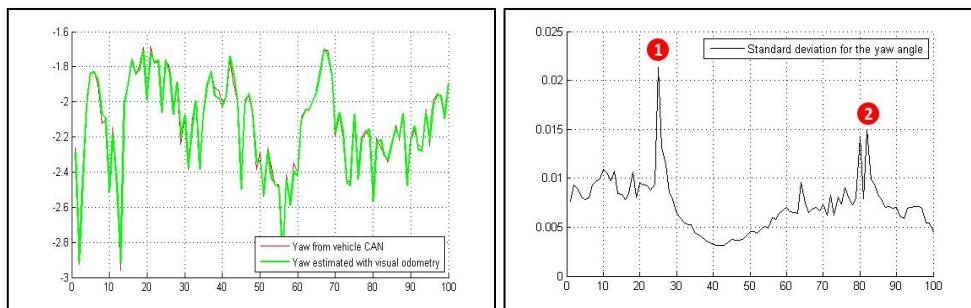


Figura 66 Stânga - unghiul yaw (verde, în grade) calculat pentru aceleasi imagini ca și Figura 65 și unghiul yaw citit de la mașină (roșu). Dreapta - deviația standard a unghiului yaw (vertical, exprimată în grade). Cu 1 și 2 sunt marcate imaginile unde avem valori mari ale deviației standard a unghiului yaw datorită schimbărilor bruște de iluminare. Punctul 1 corespunde celor două imagini din stânga în Figura 67 iar punctul 2 corespunde celor două imagini din dreapta ale aceleiasi figuri

Figura 66 din stânga reprezintă cu verde unghiul yaw pentru aceleasi 100 de imagini consecutive ca și cele din Figura 65, în care mașina echipată cu o cameră video se deplasează printr-un sens giratoriu. În grafic din dreapta am reprezentat deviația standard a unghiului yaw, calculată conform ecuației (94). Observăm cele două puncte de pe grafic care sunt încercuite cu roșu. Acestea corespund unor imagini consecutive pentru care variația de iluminare de la una la alta este ridicată (Figura 67). Conform așteptărilor din punct de vedere

teoretic, imprecizia de localizare a trăsăturilor cauzată de schimbarea condițiilor de iluminare este reflectată în modelul propus de propagare a erorilor de la nivel de trăsături și asociere de trăsături la nivel de estimare a parametrilor de odometrie.



Figura 67 Perechi de imagini consecutive unde avem schimbări de condiții de iluminare. Cu verde sun colorate trăsăturile folosite pentru estimarea odometriei vizuale. C galben sunt desenare trăsăturile eliminate în procesul de RANSAC. Vectorii de flux optic reprezentați cu roșu indică corespondențele temporale

Concluzionam prin a preciza faptul că metoda de propagare a erorilor de detecție și localizare de trăsături vine să completeze metodele care presupun descriptori invariante la condiții de iluminare (capitolul 2.4) precum și metodele de robustificare a asocierilor de trăsături (capitolul 2.2) pentru a obține algoritmi de odometrie vizuală cu un grad ridicat de robustețe.

2.6. Rafinarea selecției de trăsături și re-estimarea odometriei vizuale

Prezentăm în acest capitol principalele concluzii din lucrarea pe care am publicat-o anterior [16]. În cazul aplicațiilor automotive când sistemul de senzori video pe baza căruia este estimată odometria din trăsături video este poziționat pe o mașină și prin urmare aproximarea din (63) este validă și odometria poate fi estimată eficient folosind ecuațiile (78) sau (74). Se pune întrebarea în ce măsură putem să reutilizăm modelul matematic din capitolul 1.3 în cazul în care avem mișcări de o amplitudine mult mai mare între imagini consecutive (de exemplu în situația în care senzorii video sunt poziționați pe roboti zburători, avioane, drone, sau în cazul în care avem platforme mobile care se deplasează pe teren accidentat). În aceste situații propunerea noastră din [16] este să efectuăm o estimare inițială a rotației (fie prin metoda stereo al cărei model matematic este prezentat în capitolul 1.3,

fie prin metodele mono prezentate matematic în capitolul 1.5 și descrise algoritmice în 2.3.2.). Utilizăm apoi această rotație pentru a transforma punctul (u_1^i, v_1^i, d_1^i) într-un sistem de coordonate intermedie apoi folosim o metodă stereo pentru a determina transformata dintre sistemul de coordonate intermedie și poziția finală a camerei.

Presupunând că $f_x = f_y$:

$$P_1^i = \begin{bmatrix} X_1^i \\ Y_1^i \\ Z_1^i \end{bmatrix} = R * \begin{bmatrix} X_1^i \\ Y_1^i \\ Z_1^i \end{bmatrix} = \frac{b}{d_1^i} \cdot R \cdot \begin{bmatrix} (u_1^i - u_0) \\ (v_1^i - v_0) \\ f_x \end{bmatrix} \quad (156)$$

și

$$\begin{bmatrix} u_1^i \\ v_1^i \\ d_1^i \end{bmatrix} = \begin{bmatrix} \frac{f_x \cdot X_1^i}{Z_1^i} + u_0 \\ \frac{f_y \cdot Y_1^i}{Z_1^i} + v_0 \\ \frac{f_x \cdot b}{Z_1^i} \end{bmatrix} \quad (157)$$

O analiză mai atentă a modelului matematic din capitolul 1.3 arată că elementele vectorului:

$$q^i = R \cdot \begin{bmatrix} (u_1^i - u_0) \\ (v_1^i - v_0) \\ f_x \end{bmatrix} \quad (158)$$

sunt de fapt valorile:

$$\begin{aligned} q_1^i &= (u_1^i - u_0) - \gamma \cdot (v_1^i - v_0) - \beta \cdot f_x \\ q_2^i &= \gamma \cdot (u_1^i - u_0) + (v_1^i - v_0) - \alpha \cdot f_x \\ q_3^i &= \beta \cdot (u_1^i - u_0) + \alpha \cdot (v_1^i - v_0) + f_x \end{aligned}$$

Acestea sunt deja calculate în algoritmul de estimare a odometriei vizuale deoarece sunt folosite în etapa de verificare a trăsăturilor care respectă modelul de odometrie estimat (ca parte a procesului RANSAC), folosind ecuațiile (75), i (76) și (77) pentru predicție, și pot fi refolosite.

Valorile rotite (u_1^i, v_1^i, d_1^i) , sunt folosite în continuare pentru re-estimarea odometriei vizuale. De asemenea, având o estimare intermedie a rotației putem să recalculăm potrivirile de trăsături conform metodei descrise în capitolul 2.2.2, deoarece o mai bună

inițializare a metodelor diferențiale sau o mai bună delimitare a spațiului de căutare pentru metodele pe bază de descriptori reduce şansele de asocieri eronate. Folosind valorile intermediare ale rotației pentru a obține reprezentarea virtuală (u'_1, v'_1, d'_1) rezultă următorul algoritm:

Pasul 1. Detectam set de trăsături corespondente între imagini succesive $(u_1^i, v_1^i, d_1^i) \leftrightarrow (u_2^i, v_2^i, d_2^i), i = \overline{1, N}$ (folosind metodele din capitolele 2.1 și 2.2)

Pasul 2. Estimăm matricea de rotație R având unghiurile (α, β, γ) și opțional translația $T = [t_x \ t_y \ t_z]$ (folosim metodele descrise matematic în capitolul 1.3 sau 1.5.1 cu una din implementările descrise în capitolul 2.3 și cu îmbunătățirile propuse în capitolele 2.4 și 2.5)

Pasul 3. Optional re-estimarea corespondențelor de trăsături și rafinarea selecției acestora (folosim metodele descrise în capitolul 2.2.2)

Pasul 4.

Pasul 4.1. Transformăm (u_1^i, v_1^i, d_1^i) în (u'_1, v'_1, d'_1) folosind ecuațiile (156) și (157)

Pasul 4.2. Estimam rotația parțial R' având unghiurile $(\alpha', \beta', \gamma')$ și translația $T = [t_x \ t_y \ t_z]$ folosind modelul matematic din capitolul 1.3, dar pe baza valorilor (u'_1, v'_1, d'_1) și (u_2^i, v_2^i, d_2^i)

Pasul 4.3. Compunem rotația totală astfel: $R = R' \cdot R$

Pașii 3 și 4 pot fi repetați de câteva ori, dar în practică acest lucru nu este necesar datorită convergenței rapide în două iterații. Cu toate acestea, considerăm că metoda poate fi folosită și pentru estimarea orientării și poziției relative dintre camere aflate la distanțe mai mari unele de altele, sau pentru a estima transformarea dintre imagini care nu sunt neapărat achiziționate succesiv, caz în care efectuarea mai multor iterații are sens.

Pentru evaluare am generat un set de 400 de trăsături (u_1^i, v_1^i) distribuite aleatoriu în spațiul unei imagini având rezoluția de 1024 x 768 pixeli. Punctul principal (u_0, v_0) este setat în centrul imaginii iar

distanța focală a acesteia este $f_x = f_y = 1000$ px. Pentru fiecare punct (u_1^i, v_1^i) generăm aleatoriu o distanță Z_1^i pe baza căreia calculăm valoarea corespunzătoare a disparității d_1^i folosind ecuația (22). Din (u_1^i, v_1^i, d_1^i) putem reconstrui P_1^i , poziția 3D a punctului folosind ecuația (66). Generăm apoi o valoare de referință pentru matricea de rotație și pentru vectorul de translație, și le folosim pentru a determina poziția punctului 3D în sistemul de coordonat al celei de-a doua camere:

$$P_2^i = R_{\text{ref}} \cdot P_1^i + T_{\text{ref}}$$

Proiectând punctul P_2^i în planul celei de-a doua camere folosind ecuația (67) obținem poziția (u_2^i, v_2^i) și disparitatea d_2^i corespunzătoare acesteia. Avem astfel un set de trăsături corespondente $(u_1^i, v_1^i, d_1^i) \leftrightarrow (u_2^i, v_2^i, d_2^i)$, generate sintetic pe baza unei mișcări de referință $R_{\text{ref}}, T_{\text{ref}}$.

Vom genera în continuare mai multe seturi de astfel de puncte pentru a evalua beneficiul reestimării rotației asa cum am propus în pasul 4 al algoritmului de mai sus.

Figura 68 prezintă erorile de estimare pentru o configurație în care T_{ref} este fixat la o mișcare de translație în direcția axei Z, unghiul γ este de asemenea fixat la 0° iar valorile lui α, β sunt variate în intervalul $[-5^\circ, 5^\circ]$. Pentru fiecare combinație de α și β , folosind valorile fixe pentru γ și T_{ref} generăm un set de corespondențe $(u_1^i, v_1^i, d_1^i) \leftrightarrow (u_2^i, v_2^i, d_2^i)$. Matricea R_{ref} compusă din α, β, γ folosește reprezentarea exactă a matricii (vezi Anexa 6). Pe baza corespondențelor $(u_1^i, v_1^i, d_1^i) \leftrightarrow (u_2^i, v_2^i, d_2^i)$ folosim metodele liniare de estimare a mișcării și comparăm aceasta mișcare cu mișcarea de referință, rezultând erorile de estimare prezentat în Figura 68. Observăm un ordin de magnitudine de 10^{-3} pentru o singură estimare, respectiv un ordin de magnitudine de 10^{-7} pentru o reiterare a estimării, maximele fiind atunci când ambele valori α, β se apropiu de extretele intervalului $[-5^\circ, 5^\circ]$.

Tabel 5 ne indică ordinul de magnitudine al erorilor de estimare al odometriei vizuale prin metode liniare. Valoarea translației T_{ref} este păstrată constant la valoarea [0 0 – 1] (parcurgea unei distanțe de 1m în față) și unghiurile de rotație sunt variate în intervalul de valori descris pe fiecare rând al tabelului. Observăm că ordinul de

magnitudine scade semnificativ la efectuarea celei de-a doua iterații și că erorile cresc odată cu creșterea unghiurilor de rotație.

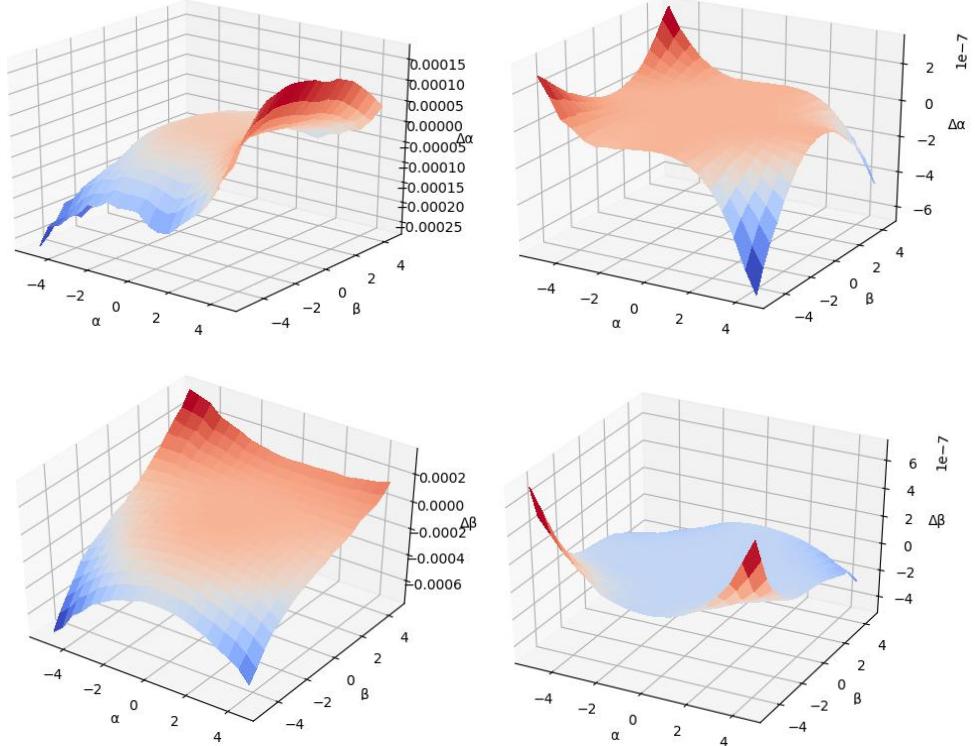


Figura 68 Erorile de estimare ale unghiurilor pitch (α , rândul de sus) și yaw (β , rândul de jos). Coloana stângă prezintă rezultatele după o singură estimare pe baza metodei liniare (pasul 2 al algoritmului). Coloana dreaptă prezintă rezultatele după reestimarea efectuată la pasul 4 al algoritmului

Experimentele prezentare în Tabel 5 se desfășoară în condiții ideale. Pentru a confirma beneficiul adus de reestimarea liniară a odometriei atunci când avem unghiuri mari de mișcare, am repetat experimentul din Tabel 5 cu adăugare de zgomot Gaussian cu o deviație standard de 0.5px punctelor (u_1^i, v_1^i, d_1^i) și (u_2^i, v_2^i, d_2^i) pe baza cărora efectuăm estimarea. Ordinul de magnitudine al erorilor relativ la valorile de referință $R_{\text{ref}}, T_{\text{ref}}$ sunt prezentare în Tabel 6. Observând ultimele patru coloane ale Tabel 6 remarcăm că erorile de rotație sunt de ordinul 10^{-6} radiani, iar cele de translație de ordinul 10^{-2} metrii chiar și mișcări de $\pm 10^\circ$ între imagini constitutiv și concluzionăm astfel că

metodele liniare de odometrie vizuală sunt generalizabile pentru orice tip de mișcare efectuată de o cameră stereo.

Fără zgomote adăugate la trăsăturile generate sintetic	Metoda liniară, o singură iterație				Metoda liniară, o iterație urmată de reestimare			
	$\Delta\alpha$ (rad)	$\Delta\beta$ (rad)	$\Delta\gamma$ (rad)	ΔT (m)	$\Delta\alpha$ (rad)	$\Delta\beta$ (rad)	$\Delta\gamma$ (rad)	ΔT (m)
$\alpha, \beta, \gamma \in [-3, 3]$	1e-4	1e-4	1e-3	1e-2	1e-8	1e-8	1e-8	1e-5
$\alpha, \beta, \gamma \in [-5, 5]$	1e-3	1e-3	1e-2	1e-1	1e-7	1e-7	1e-7	1e-4
$\alpha, \beta, \gamma \in [-10, 10]$	1e-2	1e-2	1e-1	1e-0	1e-5	1e-5	1e-5	1e-3

Tabel 5 Evaluarea ordinului de magnitudine al erorilor pentru o translație longitudinală $T_{ref} = [0 \ 0 \ -1]$ și pentru unghiiurile de rotație variind în intervalul indicat pe fiecare rând al tabelului

Cu zgomote adăugate la trăsăturile generate sintetic	Metoda liniară, o singură iterație				Metoda liniară, o iterație urmată de reestimare			
	$\Delta\alpha$ (rad)	$\Delta\beta$ (rad)	$\Delta\gamma$ (rad)	ΔT (m)	$\Delta\alpha$ (rad)	$\Delta\beta$ (rad)	$\Delta\gamma$ (rad)	ΔT (m)
$\alpha, \beta, \gamma \in [-3^\circ, 3^\circ]$	1e-4	1e-4	1e-3	1e-2	1e-4	1e-4	1e-4	1e-3
$\alpha, \beta, \gamma \in [-5^\circ, 5^\circ]$	1e-3	1e-3	1e-2	1e-1	1e-4	1e-4	1e-4	1e-2
$\alpha, \beta, \gamma \in [-10^\circ, 10^\circ]$	1e-2	1e-2	1e-2	1e-0	1e-4	1e-4	1e-4	1e-2

Tabel 6 Evaluarea ordinului de magnitudine al erorilor pentru o translație longitudinală $T_{ref} = [0 \ 0 \ -1]$ și pentru unghiiurile de rotație variind în intervalul indicat pe fiecare rând al tabelului, și adăugând zgomot Gaussian trăsăturilor generate sintetic

3. Utilizarea unor alte surse de informații de adâncime precum senzori LiDAR

Metodele prezentate în capitolul anterior se bazează pe o cameră stereo pentru informația de adâncime, și pentru a determina magnitudinea vectorului de translație. Alternativ sau complementar camerelor stereo, sistemele de percepție din mașinile cu funcții de conducere asistată și automată tind să includă și senzori LiDAR în fuziune cu senzorii video. Datorită preciziei superioare, LiDAR-ul este uneori considerat ca și o alternativă mai bună în ciuda costurilor mai ridicate. Acest lucru a motivat studiul nostru din [11] unde am propus o integrare a informației LiDAR cu o cameră mono, obținând rezultate superioare metodei stereo din [5] și [6]. Văzut dintr-o perspectivă mai largă cinci din șase grade de libertate ale mișcării unei camere între imagini succesive pot fi obținute cu metode de viziune artificială [4][3][126], rămânând un singur grad de libertate nedeterminat. Alternativ metodelor stereo, există și posibilitatea determinării celui de-al șaselea grad de libertate prin fuziune cu senzori precum GPS [120] sau cu senzori inerțiali [172][215]. De asemenea, cunoscând anumite informații preliminare despre scenă sau despre poziționarea camerei relativ la scenă se pot estima cele șase grade de libertate ale mișcării. Spre exemplu în [216] odometria vizuală este estimată presupunând că avem un model de drum planar și că înălțimea camerei relativ la planul drumului este cunoscută cu precizie. Prezentăm în continuare cele mai relevante detalii publicate anterior în [11].

3.1. Generalizarea sistemului stereo pentru LiDAR

Pornind de la metoda stereo descrisă în capitolul 2.3.1 și al cărei model matematic este detaliat în capitolul 1.3 vom dezvolta o generalizare care să funcționeze cu orice informații de adâncime, în particular cu un senzor LiDAR. De asemenea, vom utiliza aceeași abordare de optimizare numerică precum cea pentru stereo. Presupunerea generală este că avem disponibile date de intrare

calibrate (imagini rectificate în cazul unui sistem stereo, respectiv calibrare spațială și temporală între sistemul de coordonate al dispozitivului LiDAR și sistemul de coordonate al camerei. Din punct de vedere al asocierii de trăsături, metoda mono descrisă în 2.2.1 rămâne complet valabilă. Figura 69 Overview al metodelor pe bază de stereo (sus) comparativ cu metoda pe bază de LiDAR și o cameră monoculară (jos) prezintă comparație din punct de vedere al design-ului de sistem între metoda de odometrie vizuală stereo și cea de odometrie vizuală mono combinat cu LiDAR. Observăm ca modulul de asociere de trăsături precum și cel de optimizare robustă (RANSAC) sunt reutilizabile cu mențiunea că acesta se face între două perechi de imagini stereo, respectiv între două imagini consecutive provenite de la o cameră monoculară. Apare însă un modul de asociere a măsurătorilor de LiDAR la trăsăturile din planul imagine. Acestei abordări ii vom dedica un subcapitol capitol separat. Folosirea aceluiși model matematic și prin urmare a aceluiși algoritm de optimizare robustă face posibilă reutilizarea versiunii de implementare paralelă pe care am propus-o în [7].

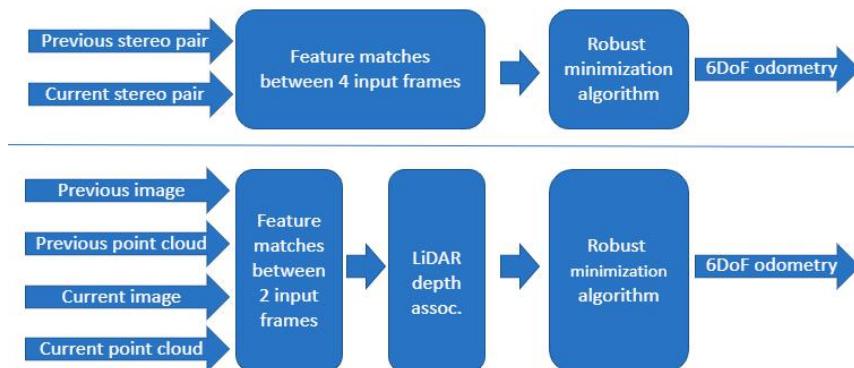


Figura 69 Overview al metodelor pe bază de stereo (sus) comparativ cu metoda pe bază de LiDAR și o cameră monoculară (jos) [11]

3.2. Generalizarea modelului matematic stereo pentru LiDAR

Pornim de la modelul matematic descris anterior în capitolul 1.3. Vom lucra în coordonate imagine, precizând că raționamentul este

același dacă lucrăm în coordonate normalize. Considerăm ecuația (78) și împărțim ultima ecuație la b:

$$\begin{cases} \alpha \cdot (v_{L1}^i - v_0) \cdot (u_{L2}^i - u_0) + \beta \cdot ((u_{L1}^i - u_0) \cdot (u_{L2}^i - u_0) + f_x^2) + \gamma \cdot f_x \cdot (v_{L1}^i - v_0) + \\ t_x \cdot \left(\frac{-f_x \cdot d_1^i}{b} \right) + t_z \cdot \frac{(u_{L2}^i - u_0) \cdot d_1^i}{b} = f_x \cdot (u_{L1}^i - u_{L2}^i) \\ \alpha \cdot ((v_{L1}^i - v_0) \cdot (v_{L2}^i - v_0) + f_x^2) + \beta \cdot (u_{L1}^i - u_0) \cdot (v_{L2}^i - v_0) + \gamma \cdot (-f_x \cdot (u_{L1}^i - u_0)) + \\ t_y \cdot \left(\frac{-f_x \cdot d_1^i}{b} \right) + t_z \cdot \frac{(v_{L2}^i - v_0) \cdot d_1^i}{b} = f_x \cdot (v_{L1}^i - v_{L2}^i) \\ \alpha \cdot (v_{L1}^i - v_0) \cdot \frac{d_2^i}{b} + \beta \cdot (u_{L1}^i - u_0) \cdot \frac{d_2^i}{b} + t_z \cdot \frac{d_2^i \cdot d_1^i}{b \cdot b} = f_x \cdot \frac{(d_1^i - d_2^i)}{b} \end{cases} \quad (159)$$

Am subliniat cu albastru cantitățile care sunt specifice unui sistem stereo de camere. Restul valorilor nu depind de a doua cameră din sistemul stereo. Din ecuația (22) obținem:

$$d_1^i = \frac{f_x \cdot b}{Z_1^i} \quad \text{și} \quad d_2^i = \frac{f_x \cdot b}{Z_2^i} \quad (160)$$

$$\frac{d_1^i}{b} = \frac{f_x}{Z_1^i} \quad \text{și} \quad \frac{d_2^i}{b} = \frac{f_x}{Z_2^i} \quad (161)$$

Putem deduce că aplicând ecuația (161) $\frac{d_1^i}{b}$ și $\frac{d_2^i}{b}$ pot fi înlocuite cu $\frac{f_x}{Z_1^i}$ respectiv cu $\frac{f_x}{Z_2^i}$, valori care depind doar de distanța focală a camerei și de distanțele la care se află trăsăturile în prima respectiv în a doua imagine. Astfel spus, dacă avem distanță calculată din LiDAR sau alte surse de măsurare a profunzimii obținem un sistem de ecuații identic cu cel utilizat pentru metoda prezentată în capitolul 1.3. În [11] am definit un aşanumit pseudo-baseline, pe baza căruia am calculat o pseudo-disparitate pe baza distantei până la trăsăturile din scenă folosind ecuația (160). Referim cititorul înspre [11] pentru mai multe detalii. Datorită densității mai mici a măsurătorilor de LiDAR comparativ cu cele de video, determinarea distanțelor Z_1^i și Z_2^i nu este trivială, și o vom explica în subcapitolul următor. Având această nouă formulare pe bază de pseudo-baseline și pseudo-disparitate sistemele de ecuații de la capitolul 1.3. își păstrează aceeași formă ca și în cazul metodelor stereo.

3.3. Determinarea distanțelor trăsăturilor pe bază de LiDAR

În [11] am folosit date de la un senzor LiDAR care este aliniat cu camera. Mediul de achiziție este cel descris în [72]. Dispozitivul LiDAR se rotește în jurul acei sale verticale și achiziționează măsurători 3D. Preluarea de imagine este inițiată atunci când LiDAR-ul preia măsurători în direcția axei optice a camerei, urmând un pas de post-procesare prin care se compensează mișcarea [72]. Setul de puncte 3D se transformă în sistemul de coordonate al camerei pe baza transformării obținute în procesul de calibrare. Acesta va corespunde elementelor statice din spațiul imagine. Elementele dinamice vor fi distorsionate, dar pentru estimarea odometriei vizuale acestea nu sunt neapărat de interes. Setul de puncte 3D obținut este mai puțin dens decât pixelii din imagine, și se vrea a fi folosit pentru a asocia informații de adâncime trăsăturilor din planul imagine. Vom prezenta în continuare procesul de interpolare pe care l-am propus în [11].

În prima fază proiectăm în imagine punctele 3D transformate anterior în sistemul de coordonate al camerei, și reținem în continuare poziția 3D a acestora (Figura 70).

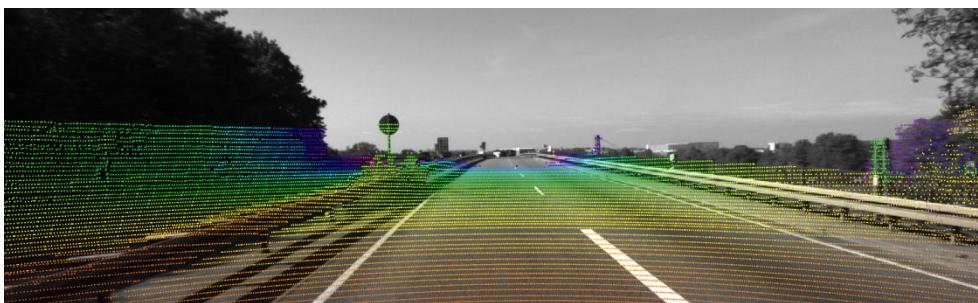


Figura 70 Puncte 3D provenite de la un senzor LiDAR, transformate în sistemul camerei, și proiectate în planul imagine

Vom construi apoi o triangulare Delaunay [217] pe baza acestor proiecții. Figura 71 ilustrează o reprezentare grafică a acestei triangulări. Pentru fiecare dintre trăsătură candidată pentru a fi folosită pentru estimarea odometriei vizuale vom determina triunghiul în care aceasta este situată apoi vom estima distanța acesteia printr-o interpolare liniară a distanțelor corespunzătoare coordonatelor 3D a

vârfurilor acestui triunghi. De asemenea, efectuăm o filtrare a trăsăturilor pe care le folosim în continuare în cadrul procesului de estimare a odometriei, pe baza proprietăților geometrice triunghiurilor în care acestea se găsesc. Motivația acestei filtrări este dată de faptul că atunci când facem interpolarea pe baza distanțelor corespunzătoare vârfurilor triunghiului presupunem ca acel triunghi corespunde unei suprafețe 3D planare, și prin urmare încercăm să filtrez acele triunghiuri care nu au astfel de proprietăți. Constrângerile pe care le impunem reflectă faptul că dorim ca vârfurile triunghiurilor considerate să fie apropiate atât în planul imagine cât și în 3D. În primul rând o parte dintre trăsăturile din imagine nu vor fi acoperite de triunghiuri construite din proiecția punctelor de LiDAR deoarece acestea acoperă doar partea de jos a imaginii. Acele trăsături care nu se găsesc în interiorul învelitorii convexe a proiecțiilor punctelor de LiDAR sunt eliminate. În al doilea rând eliminăm trăsăturile care se găsesc în triunghiuri care nu sunt suficient de mici în planul imagine. În [11] am folosit un prag de 30px pentru perimetru triunghiului.

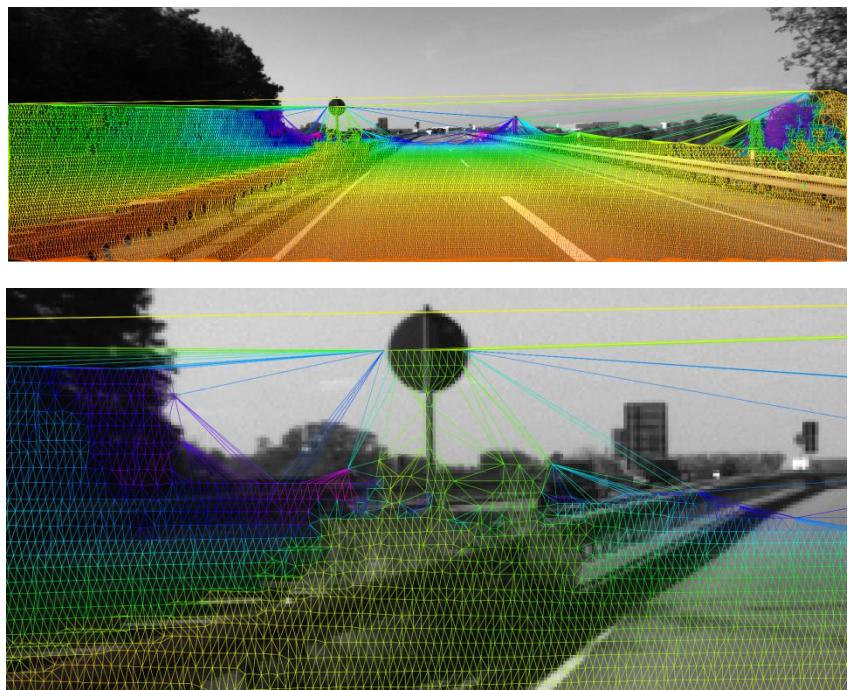


Figura 71 Triangulare Delaunay construită pe baza punctelor LiDAR proiectate în imagine (sus) și o versiune mărită (jos) pentru o mai bună vizualizare a triangulării

O a treia filtrare se bazează pe distanțele longitudinale a vârfurilor triunghiului. O trăsătură este eliminată dacă triunghiul în care aceasta este situat are următoarea proprietate: diferența dintre distanța vârfului aflat la distanță maximă și distanța vârfului aflat la distanță minimă este semnificativ mai mare decât distanța care ar fi între acestea dacă cele două puncte s-ar situa pe o suprafață planară orizontală în fața mașinii. Pentru un triunghi având vârful de distanță minimă minD , distanța maxim acceptabilă pentru celelalte vârfuri este $L = \tan\left(\arctan\left(\frac{\text{minD}}{H}\right) + \delta\right) \cdot H$, unde H este înălțimea camerei deasupra drumului, iar $\delta = 1.5^\circ$ este o distanță unghiulară maxim acceptată în condițiile în care LiDAR-ul folosit pentru achiziția setului de date pe care am lucrat [72] are o rezoluție de 0.4° . Figura 72 ilustrează geometric motivația formulei de calcul pentru distanța L . Trăsăturile situate în triunghiuri pentru care $\text{maxD} \geq L$ sunt considerate pentru calculul odometriei deoarece există o probabilitate mare ca triunghiul corespunzător acestora să nu corespundă unei suprafețe planare și prin urmare interpolarea liniară pentru determinarea distanței până la trăsătură nu este validă.

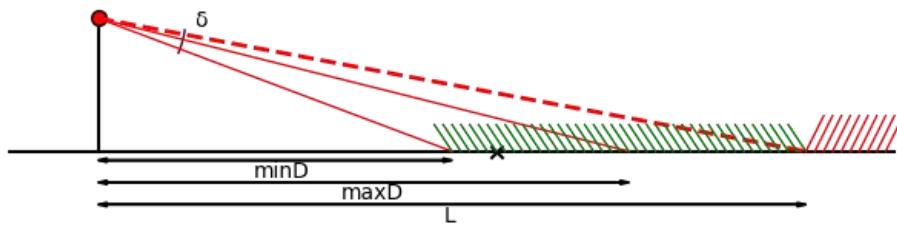


Figura 72 Filtrare adaptivă a triunghiurilor în funcție de distanța minD a celui mai apropiat vârf. Distanța maxim acceptată pentru celelalte vârfuri este L

3.4. Rezultate experimentale și concluzii

Expunem în continuare o parte dintre rezultatele experimentale din [11] pentru a ilustra superioritatea utilizării distanțelor din LiDAR în condiții bune de calibrare, precum și efectele decalibrării dintre senzori asupra metodei. Evaluăm comparativ 3 metode:

A - metoda liniară descrisă în capitolul 2.3.1 și al cărei model matematic este detaliat în capitolul 1.3.

B - o metodă hibrid care preia distanțele LiDAR acolo unde aceasta este posibil conform capitolului 3.3 și păstrează trăsăturile neacoperite de LiDAR cu disparitățile corespunzătoare acestora

C - metoda care preia distanțele LiDAR acolo unde aceasta este posibil conform capitolului 3.3 și ignoră trăsăturile neacoperite de măsurători LiDAR

Tabel 7 ilustrează performanța celor trei configurații pe setul 10 de secvențe KITTI [72]. Observăm performanța superioară obținută pentru estimarea translației, configurația A atingând o eroare de 1.31% din distanța parcursă în comparație cu C care are o eroare de 1.1% din distanța parcursă. Pentru a avea și o impresie calitativă prezentăm în Figura 73 traiectoriile obținute prin integrarea estimărilor de odometrie vizuală între imagini succesive pe secvențele 1 și 2 din setul de date KITTI [72]. Secvența 1 este cunoscută datorită vitezelor de autostradă iar secvența 2 este o secvență mai lungă care prezintă situații complexe de scenarii urbane.

	Eroarea de translație	Eroarea de rotație
A	1.31%	0.00441 [deg/m]
B	1.24%	0.00441 [deg/m]
C	1.1%	0.00412 [deg/m]

Tabel 7 Erorile de translație și rotație pentru configurațiile A (stereo), B(LiDAR combinat cu stereo) și C (LiDAR)

Un alt experiment pe care l-am făcut în [11] se referă la evaluarea impactului decalibrărilor care pot apărea între LiDAR și camera video asupra odometriei vizuale. Rezultatele vor fi analizate pe secvența KITTI 0 care are o lungime de 4500 de imagini. Observăm în tabelele de mai jos o coloană centrală pe care evaluăm erorile de rotație și cele de translație cu calibrarea originală a setului public de date. Apoi introducem un offset și re-evaluăm comportamentul algoritmului. Concluzionăm că erori de translație +/- 0.2m în direcția longitudinală

au un efect mic asupra estimării, iar cu erori mai mari de 0.2m precizia odometriei pe bază de LiDAR și video devine mai puțin precisă decât o metodă stereo.

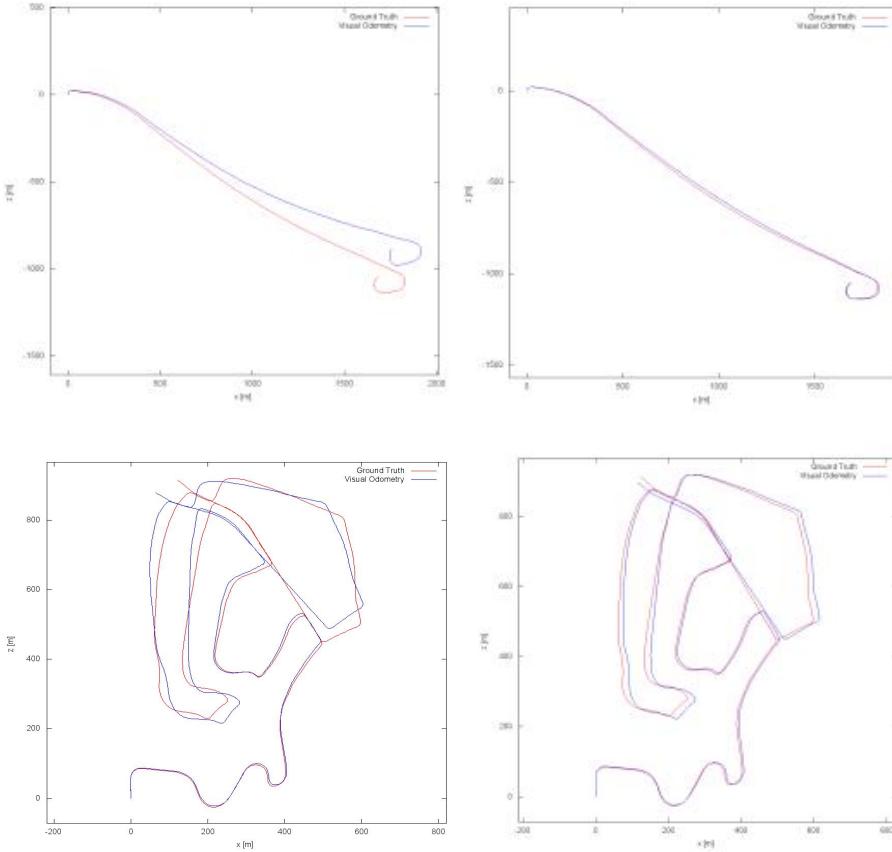


Figura 73 Evaluare calitativă între (A) metoda stereo (stânga) și (C) metoda bazată pe distanțele trăsăturilor estimate din LiDAR (dreapta) pe secvențele KITTI 1 (sus) și 2 (jos). Cu albastru este estimarea noastră iar cu roșu estimarea de referință

Valorile decalibrărilor de rotație sunt exprimate în pixeli pentru o mai bună intuiție a gradului de aliniere necesar între cei doi senzori. Practic unghiul ed decalibrare se poate calcula ca fiind $\varphi = \text{atan}(\Delta p / f_x)$, unde Δp este decalibrarea exprimată în pixeli, și invers $\Delta p = f_x \cdot \tan(\varphi)$. Observăm că valori de decalibrare de ordinul +/-2px conduc la erori de estimare ale odometriei vizuale similare cu cele ale metodei stereo, în timp ce erori ordinul 5-10px degradează semnificativ precizia de măsurare a odometriei.

	Longitudinal translation offset (m)					
	Stereo only	-0.4	-0.2	0	0.2	0.4
Rotation error (rad/m x1e-6)	83	83	73	66	67	74
Translation error (%)	1.16	1.19	1.01	1.00	1.19	1.55

Tabel 8 Erorile de estimare a odometriei la introducerea de erori de decalibrare pe direcția longitudinală a translației [11]

	Yaw offset (px)							
	Stereo only	-10	-5	-2	0	2	5	10
Rotation error (rad/m x1e-6)	83	176	102	68	66	84	121	177
Translation error (%)	1.16	2.26	1.41	1.03	1.00	1.21	1.61	2.23

Tabel 9 Erorile de estimare a odometriei la introducerea de erori de decalibrare dintre cameră și dispozitivul LiDAR pe unghiul yaw (exprimate în pixeli) [11]

	Pitch offset (px)							
	Stereo only	-10	-5	-2	0	2	5	10
Rotation error (rad/m x1e-6)	83	225	131	77	66	76	111	171
Translation error (%)	1.16	4.29	2.45	1.45	1.00	1.02	1.82	3.48

Tabel 10 Erorile de estimare a odometriei la introducerea de erori de decalibrare dintre cameră și dispozitivul LiDAR pe unghiul yaw (exprimate în pixeli) [11]

Observăm de asemenea un component asimetric, atât în cazul erorilor de translație cât și în cazul erorilor de rotație. Spre exemplu degradarea la adăugarea unui offset negativ pentru yaw este mai mică decât degradarea la adăugarea unui offset pozitiv. O explicație a acestui

fenomen ar putea fi faptul că calibrările din setul public de dat prezintă la rândul lor erori și această ipoteză poate constitui un punct de pornire pentru o metodă de estimare automată a calibrării dintre cei doi senzori.

Concluzionăm prin a puncta faptul că generalizarea metodei stereo de odometrie vizuală înspre alți senzori de măsurare a distanțelor a fost evaluată pe bază de LiDAR demonstrând rezultate superioare, dar și reutilizând în același model matematic, aceleași module de asociere de trăsături și același algoritm de optimizare robustă. De asemenea, considerăm că metoda poate funcționa și cu alte surse de estimare a distanțelor precum metode din ce în ce mai promițătoare pe bază de deep learning [89][110].

4. Reducerea erorilor acumulate la integrarea măsurătorilor de odometrie vizuală

Pentru a rezolva probleme de localizare si cartografiere estimarea odometriei vizuale între imagini succesive se compun succesiv obținând traiectoria camerei. Prin această compunere temporale a estimărilor succesive, erorile care apar la fiecare estimare dintre imagini succesive se compun la rândul lor, rezultând într-o deviere de la traiectoria reală care se amplifică în timp. Cu cât compunem mai multe estimări succesive cu atât devierea este mai mare deoarece fiecare mișcare relativă dintre imagini consecutive pe care o compunem pentru a obține traiectoria globală contribuie cu erorile sale de estimare.

În [17] am propus o metodă de atenuare a erorilor acumulate la integrarea măsurătorilor succesive de odometrie vizuală pe bază de măsurători LiDAR. Eforturile sunt motivate de faptul ca senzorii LiDAR sunt nelipsiți din activitățile de cercetare legată de percepție 3D pentru mașini autonome, urmărind astfel să creștem beneficiul adus de aceștia și pentru aplicații de cartografiere și localizare. Vom prezenta în continuare detalii relevante din [17] unde am propus o metodă de reducere a erorilor care apar la acumularea de măsurători succesive de odometrie vizuală prin alinierea de puncte 3D de LiDAR achiziționate din poziții cheie aflate la o anumită distanță între ele. Deoarece aceste erori sunt suficient de mici ca și ordin de magnitudine, am propus o metodă de formulare a problemei sub forma unui sistem linear de ecuații, eliminând în acest mod nevoie de a folosi metode de optimizare nelineară care sunt costisitoare din punct de vedere computațional. Aceasta conduce la o metodă mai eficientă computațional decât alte abordări din literatură. Problema reducerii erorilor de acumulare la integrarea în timp a odometriei vizuale este tratată și în alte moduri în literatura de specialitate. În [114] și [116] este subliniat faptul că urmărirea de trăsături video pentru mai multe imagini succesive și utilizarea acestor trăsături stabile reduce eroarea totală de integrare. Aceiași idee de a urmări trăsături pentru mai multe imagini se aplică în [60] unde urmărirea se face pe bază de filtre Kalman. În [136] și [137]

urmărirea trăsăturilor pentru mai multe imagini successive este realizată folosind descriptorii din prima imagine ca și referință, și se argumentează ce deși în acest mod trăsăturile urmărite sunt pierdute mai repede decât atunci când folosim descriptori calculați din imagini successive, acestea produc estimări mai stabile de odometrie vizuală. O alta clasă de metode sunt cele pe bază de bundle adjustment [130][131] care includ într-o problema de optimizare trăsături și parametrii de mișcare pentru un număr mai mare de imagini consecutive [113]. Fuzionarea cu senzori inertiali [172] sau cu senzori GPS [120] ajută de asemenea la reducerea erorii de integrare. De asemenea folosirea de descriptori invariante calculate pe baza unor descriptori video [140] sau LiDAR [218] poate ajuta la identificarea locurilor pe care le re-vizităm, metodă cunoscută sub denumirea de loop closure în literatură.

Lucrarea noastră din [17] este influențată mai ales de [70][71]. În [70] este propusă o metodă de odometrie vizuală strict pe bază de LiDAR, iar aceeași autori propun o fuziune între LiDAR și video în [71]. Vom prelua din [70] ideea de a detecta puncte de muchie și puncte planare într-un moment de timp, de a le asocia muchiilor corespunzătoare respectiv planelor corespunzătoare achiziționate într-un alt moment de timp și de a determina transformata dintre cele două momente diferite de timp. Folosind aceste trăsături, dezvoltăm un model matematic care presupune rezolvarea unui sistem liniar de ecuații pentru a determina și corecta erorile de integrarea mișcărilor successive.

4.1. Prezentare generală a sistemului

Figura 74 prezintă o diagramă block a sistemului de corecție a erorilor de integrare temporală a odometriei vizuale și contextul mai larg de estimare a odometriei vizuale între imagini successive. Odometria între imagini succesive, reprezentată prin blocuri de culoare gri, este realizată folosind metodele din capitolul 2.3.1, sau varianta îmbunătățită a acestora din capitolul 3. În general această componentă poate fi înlocuită cu orice altă metodă de odometrie care estimează matricea de rotație și vectorul de translație reprezentând mișcarea dintre poziții successive al sistemului.

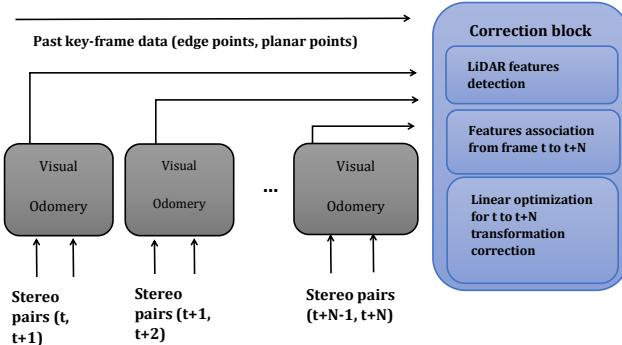


Figura 74 Diagrama bloc a sistemului de corecție a odometriei vizuale [17]

Imaginiile cheie între care se execută rutina de corecție sunt selectate pe baza distanței parcuse de la ultima corecție (sau de la inițializarea sistemului). Rutina de corecție se bazează pe măsurătorile LiDAR asociate cu imaginea cheie trecută și respectiv cu imaginea cheie curentă. Într-o prima etapă detectăm puncte de muchie și puncte planare aşa cum este descris în subcapitolul 4.2 (vezi Figura 75). Punctele de muchie asociate cu imaginea cheie anterioară sunt asociate cu muchii 3D din imaginea cheie curentă. La fel, puncte planare din imaginea cheie anterioară sunt asociate cu plane 3D din imaginea cheie curentă. Transformatele cunoscute dintre imagini consecutive pot fi compuse, obținând o transformată totală dintre imaginea cheie anterioară (t) și imaginea cheie curentă ($t+N$). Această transformată dintre imaginea (t) și imaginea ($t+N$) este folosită pentru a reduce spațiul de căutare la asocierea dintre puncte de muchie și de plan din imaginea cheie anterioară (t) la muchii și plane 3D în imaginea cheie curentă ($t+N$).

Odată ce corespondențele sunt disponibile formulăm o problemă de optimizare care estimează o transformată de corecție prin rezolvarea unui sistem liniar de ecuații specificând următoarele constrângeri:

- punctele de muchie propagate de la imaginea cheie anterioară (t) trebuie să aparțină muchiilor cu care acestea sunt asociate în imaginea cheie curentă ($t+N$)
- punctele planare propagate de la imaginea cheie anterioară (t) trebuie să aparțină planelor cu care acestea sunt asociate în imaginea cheie curentă ($t+N$)

Calcularea acestor transformări de corecție între imagini cheie dintr-un stream video și utilizarea lor atunci când facem integrarea temporală a estimărilor de odometrie vizuală conduce la o acuratețe mai bună a traiectoriei estimate. Practic erorile acumulate la integrarea în timp a odometriei vizuale sunt corectate de acest mecanism de corecții care intervine după parcurgerea unei anumite distanțe prin selecția acelor imagini cheie între care se face rutina de corecție.

4.2. Determinarea de trăsături și de asociere de trăsături din LiDAR

În prima fază clasificăm punctele de LiDAR în trei categorii: puncte de muchie, puncte planare și o a treia categorie de puncte care nu sunt de interes pentru algoritmul de corecție. Clasificarea se face pe baza unui scor de confidență similar cu cel descris în [70]. Pentru fiecare punct P_i transformat în sistemul de coordonate al camerei calculăm următorul scor:

$$c_i = \frac{1}{|\Gamma_i| \cdot \|P_i\|} \cdot \left\| \sum_{j \in \Gamma_i} P_j - P_i \right\| \quad (162)$$

unde Γ_i reprezintă vecinătatea punctului P_i în lista ordonată a punctelor achiziționate de LiDAR sortate după timpul de achiziție a fiecărui punct. Γ_i include toate punctele dintr-o vecinătate a punctului P_i , în lista sortată de primire a acestora de la dispozitivul LiDAR, dar excludând punctele care au o distanță unghiulară față de P_i mai mare decât un prag dat (de exemplu 2°). Pentru a discrimina între puncte de muchie și între puncte de plan am ales vecinătăți de raze diferite în lista de intrare (2 points respectiv 30 points). Punctele care au scorul din ecuația (162) mai mare decât un prag sunt clasificate ca fiind puncte de muchie, în timp ce punctele care au scorul sub un prag (calculat cu o vecinătate am mare) sunt considerate ca fiind puncte planare (Figura 75).

Procesul de selecție a punctelor de muchie include încă două etape suplimentare de filtrare. În primul rând executăm un pas de eliminare a punctelor aflate la limita de ocluziune, la fel ca și în [70]. Aceasta se face prin analiza unei vecinătăți a fiecărui punct P_i reprezentat

în coordonate polare. Fiind reprezentat în coordonate polare, o vecinătate este considerată ca fiind setul de puncte care au cele două unghiuri (azimuth, elevation) din reprezentarea acestora în coordonate polare apropiate de reprezentarea lui P_i . Vom elimina din această vecinătate punctele care sunt la distanță semnificativ mai mare decât P_i față de centrul camerei, eliminând astfel punctele de background care sunt aproape de muchii 3D și care pot fi ulterior acoperite dacă deplasăm sistemul de senzori într-o altă poziție. După această etapă o procedură de non-maxima-suppression este aplicată pe setul de puncte deja clasificate ca și puncte de muchie. Subțiem practic muchiile și le păstrăm doar pe cele puternice conform scorului din (162).

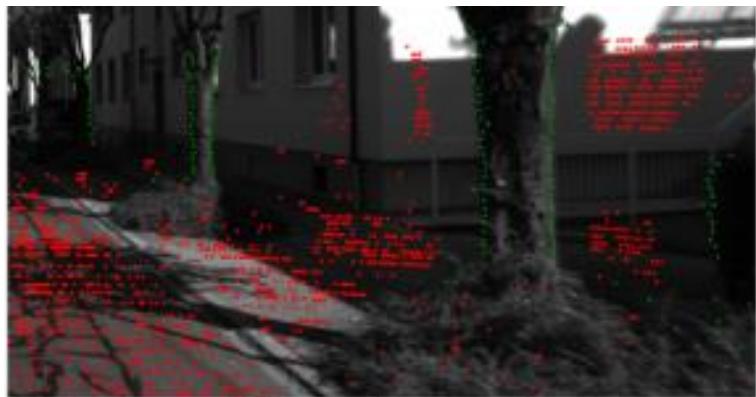


Figura 75 Clasificarea punctelor LiDAR în puncte planare (roșu) și puncte de muchie (verde)

Punctele clasificate ca fiind puncte planare sau puncte de muchie se propagă din sistemul de coordonate corespunzător imaginii cheie anterioare în sistemul de coordonate al imaginii cheie curente pe baza unei transformate compuse din estimările de odometrie vizuală între cele două imagini cheie.

Folosim în continuare aceeași abordare ca și cea descrisă în [24] pentru a asocia punctele din trecut transformate pe baza odometriei vizuale la muchiile și la planele corespunzătoare din imaginea curentă. Pentru fiecare punct de muchie transformat din trecut căutăm cele mai apropiate puncte de muchie din prezent situate pe linii de achiziție diferite. Similar pentru fiecare punct planar proiectat din trecut căutăm cele mai apropiate trei puncte planare din prezent situate pe cel

puțin două linii de achiziție diferite are dispozitivului LiDAR. Pentru a permite căutări spațiale rapide punctele planare și cele de muchie din imaginea cheie curentă sunt adăugate într-un arbore k-d [220].

Pentru puncte de muchie vom face căutări spațiale pe o rază de 0.25m. Pentru punctele planare vom face căutări de corespondențe pe o rază de 0.5m și adăugăm condiția de a avea cel puțin 10 puncte în această vecinătate. Toate aceste puncte trebuie să aparțină unui plan. Testăm acest lucru făcând o analiză a componentelor principale pe setul de puncte rezultat în urma căutării spațiale și verificăm că o componentă principală (cea din direcția normalei la plan) are o valoare principală mică și două componente principale (situate practic în plan) au valori principale mari. Folosirea unui plan de regresie prin toate punctele returnate de căutarea spațială pe o rază de 0.5m este o alternativă la utilizarea unui plan compus din cele mai apropiate 3 puncte situate pe linii de achiziție diferite. Această alternativă a fost de asemenea evaluată în lucrarea noastră din [17].

4.3. Modelul matematic

Fie $\{P_i^t\}$ setul de trăsături LiDAR corespunzătoare imaginii numărul t și R^t, T^t odometria vizuală estimată din stereo între imaginile t și t + 1. Folosind această estimare punctul 3D P_i^t poate fi transformat din sistemul de coordonate al imaginii t în sistemul de coordonate al imaginii t + 1 astfel: $R^t \cdot P_i^t + T^t$. Aceste transformări pot fi concatenate în mod succesiv pentru a propaga puncte 3D din trecut în prezent astfel:

$$\begin{aligned} & R^t \cdot P_i^t + T^t \\ & R^{t+1} \cdot R^t \cdot P_i^t + (R^{t+1} \cdot T^t + T^{t+1}) \\ & R^{t+2} \cdot R^{t+1} \cdot R^t \cdot P_i^t + (R^{t+2} \cdot R^{t+1} \cdot T^t + R^{t+2} \cdot T^{t+1} + T^{t+2}) \end{aligned}$$

În general transformarea dintre sistemul de coordonate al imaginii t și cel al imaginii t + N poate fi calculată recursiv astfel:

$$R^{t+N,t} = R^{t+N} \cdot R^{t+N-1,t} \quad (163)$$

$$T^{t+N,t} = R^{t+N} \cdot T^{t+N-1,t} + T^{t+N} \quad (164)$$

Pentru o trăsătură de muchie $P_i^{t_1}$ din sistemul de coordonate asocial imaginii numărul t_1 asociem cele mai apropiate două puncte de muchie

aflate pe linii de achiziție diferite ale LiDAR-ului $P_j^{t_2}, P_k^{t_2}$. Punctul $P_i^{t_1}$ este propagat în sistemul de coordonate al imaginii t_2 astfel: $\bar{P}_i^{t_2} = R^{t_2, t_1} \cdot P_i^{t_1} + T^{t_2, t_1}$. După transformarea lui $P_i^{t_1}$ setul de puncte $\bar{P}_i^{t_2}, P_j^{t_2}, P_k^{t_2}$ sunt în același sistem de coordonate și aparțin aceleiași muchii (deci sunt colineare). Cu toate acestea, datorită erorilor de estimare a odometriei vizuale care sunt incluse în valorile parametrilor de transformare $R^{t_2, t_1}, T^{t_2, t_1}$ cele trei puncte nu vor fi perfect colineare, și scopul nostru va fi acela de a determina o transformată de corecție R_c, T_c care să le facă colineare. Făcând presupunerea ca unghiurile de rotație sunt mici, la fel ca și în capitolul 1.3 (vezi și Anexa 6), putem considera:

$$R_c = \begin{bmatrix} 1 & -\gamma & -\beta \\ \gamma & 1 & -\alpha \\ \beta & \alpha & 1 \end{bmatrix} \quad T_c = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (165)$$

Pentru simplitatea notațiilor, în restul acestui capitol vom renunța la indexul t_2 când reprezentăm explicit coordonatele 3D ale punctelor:

$$\bar{P}_i^{t_2} = \begin{bmatrix} X^i \\ Y^i \\ Z^i \end{bmatrix}, P_j^{t_2} = \begin{bmatrix} X^j \\ Y^j \\ Z^j \end{bmatrix}, P_k^{t_2} = \begin{bmatrix} X^k \\ Y^k \\ Z^k \end{bmatrix} \quad (166)$$

Matricea de corecție este aplicată pentru $\bar{P}_i^{t_2}$. Vom scrie condiția de coliniaritate pentru $R_c \cdot \bar{P}_i^{t_2} + T_c, P_j^{t_2}$ și $P_k^{t_2}$ astfel:

$$\frac{X^i - \gamma \cdot Y^i - \beta \cdot Z^i + t_x - X^k}{X^j - X^k} = \frac{\gamma \cdot X^i + Y^i - \alpha \cdot Z^i + t_y - Y^k}{Y^j - Y^k} \quad (167)$$

$$\frac{X^i - \gamma \cdot Y^i - \beta \cdot Z^i + t_x - X^k}{X^j - X^k} = \frac{\beta \cdot X^i + \alpha \cdot Y^i + Z^i + t_z - Z^k}{Z^j - Z^k} \quad (168)$$

Aceasta va conduce la un sistem de ecuații având următoarea formă:

$$A \cdot [\alpha \quad \beta \quad \gamma \quad t_x \quad t_y \quad t_z]' = b \quad (169)$$

unde simbolul ' reprezintă operatorul de transpunere a unei matrici sau a unui vector.

Se poate demonstra ușor că valoarea lui 'A' este următoarea:

$$\left[\begin{array}{cc} (X^j - X^k) \cdot Z^i & -(X^j - X^k) \cdot Y^i \\ -(Y^j - Y^k) \cdot Z^i & -(Z^j - Z^k) \cdot Z^i - (X^j - X^k) \cdot X^i \\ -(Y^j - Y^k) \cdot Y^i - (X^j - X^k) \cdot X^i & -(Z^j - Z^k) \cdot Y^i \\ Y^j - Y^k & Z^j - Z^k \\ -(X^j - X^k) & 0 \\ 0 & -(X^j - X^k) \end{array} \right] \quad (170)$$

Valoarea vectorului b este:

$$\begin{bmatrix} (X^j - X^k) \cdot (Y^i - Y^k) - (X^i - X^k) \cdot (Y^j - Y^k) \\ (X^j - X^k) \cdot (Z^i - Z^k) - (X^i - X^k) \cdot (Z^j - Z^k) \end{bmatrix} \quad (171)$$

Rezultă că dintr-o asociere de punct de muchie obținem două ecuații care constrâng vectorul necunoscut de parametrii ai transformației de corecție: $[\alpha \ \beta \ \gamma \ t_x \ t_y \ t_z]$.

Similar cu modul de gestionare a trăsăturilor de muchie vom gestiona și trăsăturile planare. Pentru o trăsătură planară $P_i^{t_1}$ din sistemul de coordonate asocial imaginii numărul t_1 asociem cele mai apropiate trei puncte de planare aflate pe linii de achiziție diferite ale LiDAR-ului $P_j^{t_2}$, $P_k^{t_2}$, $P_l^{t_2}$. Punctul $P_i^{t_1}$ este propagat în sistemul de coordonate al imaginii t_2 astfel: $\bar{P}_i^{t_2} = R^{t_2, t_1} \cdot P_i^{t_1} + T^{t_2, t_1}$. Apoi scopul este de a găsi o transformată de corecție R_c, T_c care face punctele $R_c \cdot \bar{P}_i^{t_2} + T_c, P_j^{t_2}, P_k^{t_2}, P_l^{t_2}$ coplanare. Aceasta poate fi exprimată sub forma unui determinant, rezultând o nouă ecuație liniară care constrâng transformata de corecție R_c, T_c reprezentată ca si vector de șase parametrii:

$$\begin{vmatrix} X^i - \gamma \cdot Y^i - \beta \cdot Z^i + t_x - X^k & X^j - X^k & X^l - X^k \\ \gamma \cdot X^i + Y^i - \alpha \cdot Z^i + t_y - Y^k & Y^j - Y^k & Y^l - Y^k \\ \beta \cdot X^i + \alpha \cdot Y^i + Z^i + t_z - Z^k & Z^j - Z^k & Z^l - Z^k \end{vmatrix} = 0 \quad (172)$$

Pentru mai multe detalii referim Anexa 8. Alternativa descrisă și în capitolul anterior este de a face o regresie de plan într-o vecinătate a trăsăturii planare propagate de la t_1 . Modelul matematic pentru această opțiune de modelare a problemei este descris în Anexa 9.

4.4. Optimizare robustă

Punând împreună constrângerile date de toate corespondențele de trăsături de muchie (170)(171) și cele date de toate corespondențele de trăsături planare (172) obținem un sistem liniar supradeterminat care poate fi rezolvat direct, prin metode consacrate de analiză numerică, pentru parametrii săi necunoscuți $[\alpha \ \beta \ \gamma \ t_x \ t_y \ t_z]$. Robustea este asigurată prin efectuarea unor iterații prin intermediul căror să eliminăm punctele care în continuare au o distanță mare până la dreapta sau până la planul corespunzător. Cu alte cuvinte recalcăm

coordonatele punctelor de muchie și punctelor planare în sistemul curent de coordonate considerând transformata de corecție estimată: $R_c \cdot \bar{P}_i^{t_2} + T_c$, apoi calculăm distanța de la aceste puncte la muchiile și la planele asociate. Corespondențele care prezintă o distanță mai mare decât un prag adaptiv sunt eliminate la fiecare iterație. Referim [17] pentru mai multe detalii.

4.5. Rezultate și concluzii

Abordarea noastră este dezvoltată pentru a corecta erorile de integrare a metodelor propuse anterior care folosesc strict informația din imagini consecutive (capitolul 2.3.1 și capitolul 3). Experimentele le-am făcut pe setul de date KITTI [72] care dispune atât de date stereo calibrate cât și de măsurători achiziționate cu un LiDAR și care sunt transformate în sistemul de coordonate al camerei stângi, permitând astfel evaluarea acestor metode de fuziune.

Corecția se bazează pe selecția de trăsături de muchie și de trăsături planare estimate în ordinea de achiziție a măsurătorilor LiDAR (linii de măsurare orizontale). Aceasta poate să conducă la o capacitate mai redusă de a constrânge anumite grade de libertate, prin urmare evaluăm potențialul metodei de a corecta diferite componente ale parametrilor de mișcare $[\alpha \ \beta \ \gamma \ t_x \ t_y \ t_z]$. Considerăm patru subseturi de parametrii pe care încercăm să-i corectăm:

- corectăm setul întreg de parametrii $[\alpha \ \beta \ \gamma \ t_x \ t_y \ t_z]$
- excludem unghiul roll și corectăm restul $[\alpha \ \beta \ t_x \ t_y \ t_z]$
- corectăm unghiul yaw și translația $[\beta \ t_x \ t_y \ t_z]$
- corectăm yaw și translația în planul orizontal $[\beta \ t_x \ t_z]$

Pentru dimensiunile vectorului de parametrii pe care le excludem din procesul de corecție setăm coeficienții corespunzători din ecuațiile (170),(171) și (172) la zero. Tabel 11 sumarizează analiza pe setul de secvențe de la 00 la 10 și subliniază faptul că varianta optimă este să corectăm unghiul yaw și vectorul de translație.

Parameter subset	Translation error (% from traveled distance)	Rotation error (radians/m)
Baseline (no correction)	1.10	0.000063
$[\alpha \ \beta \ \gamma \ t_x \ t_y \ t_z]$	1.21	0.000101
$[\alpha \ \beta \ t_x \ t_y \ t_z]$	1.07	0.000081
$[\beta \ t_x \ t_y \ t_z]$	0.87	0.000064
$[\beta \ t_x \ t_z]$	0.89	0.000064

Tabel 11 Evaluarea performanței de corecție a odometriei vizuale pe bază de LiDAR.
Liniile tabelului indică diferite configurații de parametrii pe care-i corectăm

KITTI sequence	Translation error (% from traveled distance)		Rotation error (radians/m)	
	Frame-to-frame [2], [5]	With correction	Frame-to-frame [2], [5]	With correction
00	1.00	0.87	0.000066	0.000078
01	1.16	1.42	0.000043	0.000091
02	1.15	0.87	0.000059	0.000057
03	1.55	0.73	0.000040	0.000055
04	1.18	0.58	0.000062	0.000064
05	0.89	0.57	0.000058	0.000049
06	1.12	0.47	0.000070	0.000046
07	1.14	0.56	0.000113	0.000058
08	1.38	1.09	0.000077	0.000074
09	0.85	0.71	0.000046	0.000035
10	0.86	1.11	0.000049	0.000070
Overall	1.10	0.87	0.000063	0.000064

Tabel 12 Evaluarea performanței algoritmului de corecție pe setul de date KITTI [72]

Dacă alegem să corectăm toși cei șase parametrii de performanță va scădea ușor datorită lipsei muchiilor orizontale care constrâng în general unghiurile pitch și roll.

Tabel 12 ilustrează performanța crescută când aplicăm metodele de corecție din [17], cu o distanță între imaginile cheie între care se face corecția de 4m. Figura 76 arată exemple de traекторii corectate comparativ cu cele necorectate pentru aceeași distanță de 4m între imagini cheie.

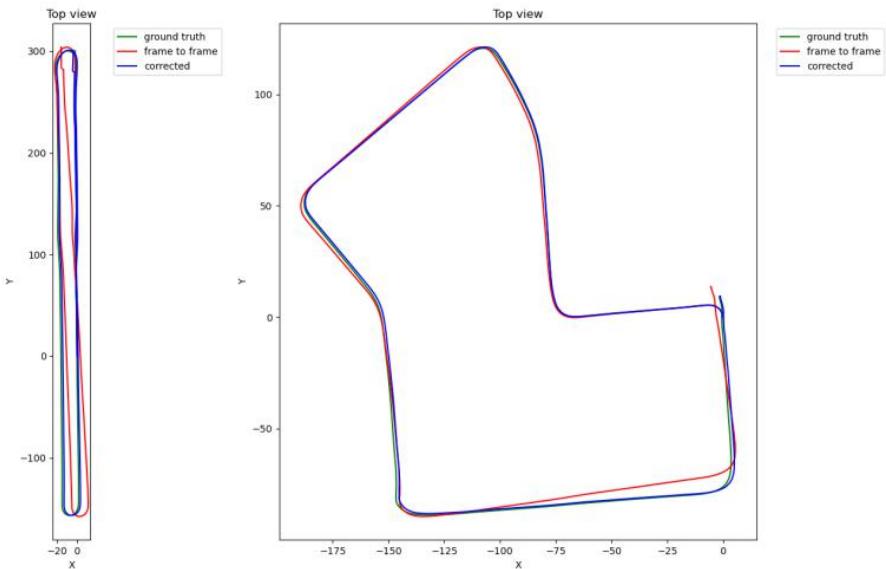


Figura 76 Exemple de traекторii corectate (secvența KITTI 06 în stânga și secvența 07 în dreapta). Cu verde este reprezentată traectoria de referință (în foarte mare mărime acoperită de traectorii estimate și corectate), cu albastru este traectoria corectată prin metodele propuse în [17], cu roșu este traectoria generată cu o metodă care face estimarea strict din imagini succesive

Am studiat de asemenea efectul pe care îl are asupra corecției pragul de distanță între imagini cheie. Rezultatele sunt prezentate în Tabel 12.

Folosirea a două puncte de muchie pentru a reprezenta o muchie și a trei puncte planare pentru a reprezenta un plan conduce la o salvare semnificativă a resurselor de calcul. Am evaluat care ar fi performanța dacă am folosi mai multe puncte pentru a reprezenta un plan (Anexa 9) comparativ cu utilizarea celor mai apropiate trei puncte de pe minim

două linii de achiziție a dispozitivului LiDAR (Anexa 8). Tabel 14 arată faptul ca aceasta din urmă conduce la o performanță mai bună de corecție a traiectoriei. Referim [17] pentru mai multe detalii.

Threshold for traveled distance between key-frames	Translation error (% from the traveled distance)	Rotation error (radians/m)
1	0.91	0.000067
2	0.88	0.000066
3	0.87	0.000065
4	0.87	0.000064
5	0.97	0.000069
6	0.95	0.000067
7	0.97	0.000066
8	1.00	0.000065
9	1.02	0.000064
10	1.01	0.000063

Tabel 13 Evaluarea efectului pe care îl are pragul de distanță între imagini cheie asupra performanței de corecție a traiectoriei pe cele 10 secvențe KITTI

Translation error (% from traveled distance)		Rotation error (radians/m)	
Corrected (planes with 3 points)	Corrected (fitting of planes)	Corrected (planes with 3 points)	Corrected (fitting of planes)
0.87	0.89	0.000064	0.000065

Tabel 14 Evaluarea unor opțiuni alternative de reprezentare a planelor [17]

Concluzionăm prin a puncta că am propus o metodă de corecție a erorilor care se acumulează la compunerea traiectoriei din estimări din imagini successive ale odometriei vizuale. Aceasta se bazează pe trăsături de tip muchie și pe trăsături planare din imagini dintr-o imagine cheie anterioară, asociate la muchiile și respectiv la planele corespunzătoare în imaginile cheie curente. Pragul optim a distanței

dintre imaginile cheie este de 4m. Deoarece erorile de acumulare sunt mici pe, am propus o metodă de liniarizare a problemei de corecție, formulând-o sub forma unui sistem liniar de ecuații care se poate rezolva eficient din punct de vedere computațional. Rezultatele experimentale arată o îmbunătățire a precizia de estimare a traectoriei pe setul de date KITTI [72] comparativ cu abordările care se bazează doar pe compunerea estimărilor dintre imagini succesive.

Ca și direcții viitoare de cercetare intenționăm să evaluăm în ce măsură formularea neliniară din [70] aduce beneficii față de formularea curentă propusă în [17]. Considerăm că printr-o rafinare a parametrilor algoritmului se pot obține în continuare performanțe ușor îmbunătățite (o reducere de 2-3% din distanța parcursă a erorii de translație), mai ales pe secvențele care prezintă în continuare erori mai mari (spre exemplu 01. și 10). De asemenea vom experimenta în ce măsură este posibil să folosim o categorizare semantică a punctelor din LiDAR pentru o clasificare mai robustă a punctelor de muchie și a celor planare. Nu în ultimul rând includerea unor muchii orizontale care să constrângă mai bine unghiurile de roll și de pitch poate să fie benefică și să aducă un plus de performanță în reconstrucția traectoriei.

5. Estimarea mișcării relative dintre obiecte folosind metode de odometrie vizuală

Capitolul curent se bazează pe publicațiile noastre anterioare [14][15] și își propune să lanseze ideea estimării precise a mișcării relative dintre obiectele dinamice din scenă și o cameră stereo care la rândul ei este în mișcare. Pornim de la observația că metodele de odometrie vizuală stereo descrise în capitolul 2 determină un set maximal de trăsături care au aceeași mișcare relativ la o cameră stereo. Deoarece majoritatea trăsăturilor sunt situate pe obiecte statice, mișcarea respectivă va fi echivalentă cu mișcarea camerei. Dacă reducem regiunea de interes din spațiul imagine pe baza căreia facem estimarea și rulăm același algoritm de odometrie vizuală observăm ca setul maximal de trăsături care duce la convergența algoritmului este situat pe un obiect în mișcare, și ca mișcarea estimată este de fapt mișcarea relativă dintre cameră și acel obiect (Figura 77).



Figura 77 Trăsăturile cu verde reprezintă setul maximal de trăsături pe baza cărora se face estimarea mișcării relative dintre camera și background (stânga) respectiv dintre cameră și un obiect aflat în mișcare (dreapta)

Pornind de la aceste observații am propus în [13] următorul algoritm:

Pasul 1 Detectam un set S de trăsături corespondente între două perechi stereo de imagini succesive (folosind metodele descrise în capitolele 2.1 și 2.2)

Pasul 2 Executăm un algoritm de odometrie vizuală care va determina în set de trăsături statice și de background B ca și set maximal al algoritmului RANSAC (folosind metodele descrise în 2.3 și 2.4), precum și

parametrii de mișcare ai camerei față de lumea statică în care aceasta se mișcă

Pasul 3 $S = S \setminus B$

Pasul 4 RANSAC pentru detecția mișcării relative dintre obiecte:

Pasul 4.1 Selectăm 3 trăsături aleatorii din S care sunt consistente geometric (δ_{road} , δ_{image} , δ_{height})

Pasul 4.2 Calculăm parametrii de mișcare pe baza sistemelor de ecuații de forma (74) sau (78) scrise pentru trăsăturile selectate la pasul 4.1

Pasul 4.3 Determinăm setul de trăsături care respectă parametrii de mișcare calculați la pasul 4.2. Dacă este mai mare ca și setul anterior atunci îl stocăm în variabila I

Pasul 4.4 Dacă numărul maxim de iterații nu este atins atunci revenim la 4.1

Pasul 5 Dacă I este suficient de mare:

Pasul 5.1 Calculăm parametrii de mișcare pe baza ecuațiilor (74) sau (78) scrise pentru trăsăturile din I . Viteza absolută este calculată cunoscând viteza calculată la pasul 2 precum și mișcarea calculată pe baza trăsăturilor din I

Pasul 5.2 $S = S \setminus I$

Pasul 5.3 Dacă S mai are elemente neprocesate revenim la pasul 4

În urma pașilor 1 și 2 vom obține odometria vizuală și trăsăturile pe baza cărora a fost calculată aceasta (Figura 77). Trăsăturile rămase sunt fie obiecte dinamice, fie trăsături având asocieri eronate. Următorii pași au ca și scop delimitarea de grupuri de trăsături care se mișcă la fel relativ la cameră, deci apartinând aceluiași obiect. Consistența geometrică punctată la pasul 4.1 este impusă prin următoarele condiții:

- Distanța dintre puncte în planul drumului este mai mică decât δ_{road}
- Distanța dintre puncte în planul imagine este mai mică decât δ_{image}
- Diferența dintre înălțimile punctelor este de asemenea mai mică decât δ_{height}

De remarcat ca aceste condiții nu se aplică dimensiunii obiectelor aflate în mișcare ci doar setului de trăsături inițial considerat pentru a crește probabilitatea ca acestea să se afle pe același obiect.

Prezentăm în continuare câteva rezultate preluate din [13] pentru a ilustra capabilitățile algoritmului pe seturi de imagini de rezoluție 512x383 achiziționate în scene de trafic din Cluj-Napoca. Parametrii folosiți sunt $\delta_{\text{road}} = 4\text{m}$, $\delta_{\text{image}} = 80\text{px}$, $\delta_{\text{height}} = 3\text{m}$.



Figura 78 Trăsăturile folosite pentru a calcula mișcarea generală a camerei (verde) și cele rezultate pentru calculul mișcării obiectelor din scenă

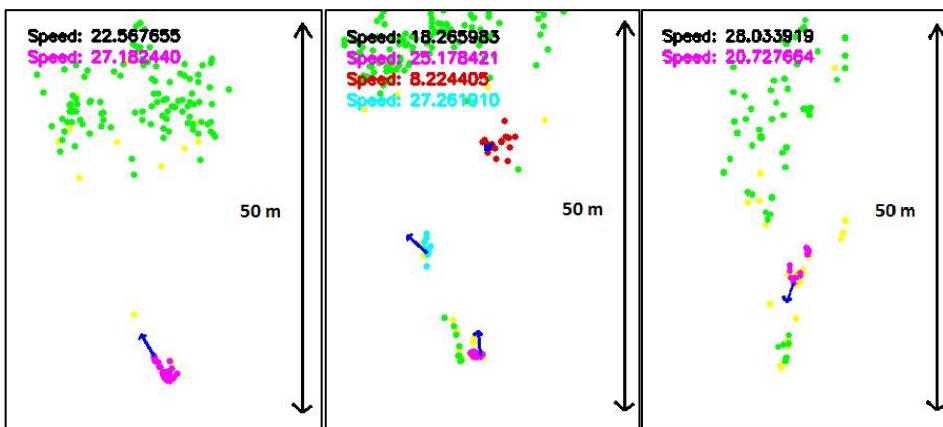


Figura 79 Reprezentare de sus a scenelor din Figura 78. Observăm vitezele de deplasare estimate pentru obiectele dinamice, precum și orientarea vectorului de viteză al acestor obiecte (albastru)

Figura 78 ilustrează cu verde punctele folosite pentru a estima mișcarea față de background (odometria vizuală) și cu alte culori punctele folosite pentru calcul mișcările celorlalte obiecte din scenă. Figura 79 este o reprezentare de sus (in planul drumului) a scenelor din

Figura 78. Observăm în plus orientarea și magnitudinea vitezelor celorlalte obiecte.

De remarcat faptul ca metoda este promițătoare pentru a identifica orice grupuri de puncte care efectuează aceeași mișcare rigidă relativ la cameră. Spre exemplu Figura 80 ilustrează o scenă de trafic de foarte mare relevanță pentru sistemele de siguranță și asistență. Observăm că algoritmul nostru estimează viteza relativă a unei mașini care intră în aria vizuală a camerei înainte de a avea obiectul complet în scenă, permitând astfel un timp mai mare pentru a lua o decizie automată (frânare, virare la stânga) de evitare a unui posibil accident. Metoda prezentată în [13] nu este dependentă de o pre-segmentare a obiectelor din scenă. Desigur că dacă am avea un detector de obiecte preselecția trăsăturilor devine mai robustă. În [14] am propus o metodă care combină segmentarea scenelor 3D pe baza de informații de adâncime și super-pixeli (Figura 81). Referim cititorul înspre [14] pentru mai multe detalii legat de gruparea super-pixelilor în obiecte și combinarea unor astfel de metode de grupare și segmentare cu algoritmi de estimare a mișcării relative între obiecte.

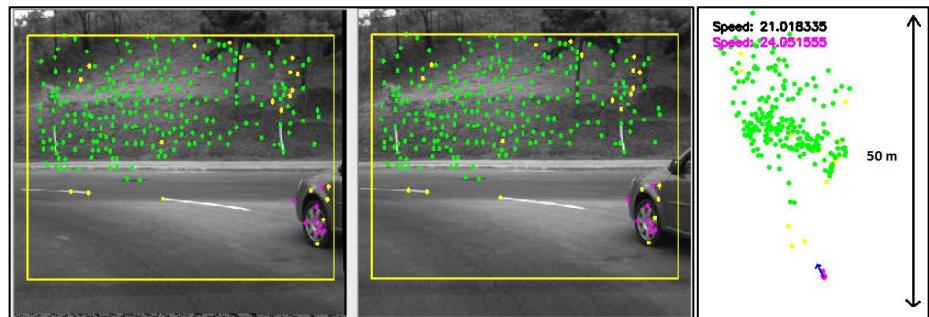


Figura 80 Estimarea mișcării relative a unei mașini care intră în scenă (primele două imagini din stânga)

Ca și concluzii consideram ca astfel de metode de estimare precisa a atributelor obiectelor pot fi foarte promițătoare dacă sunt folosite împreună cu metodele standard de detecție a obiectelor. De obicei metodele consacrate de detecție a obiectelor precum [209], [210], [211] pierd din rezoluție în procesul de grupare a informațiilor 3D în obiecte, și implicit din precizia pe care o obțin ulterior la estimarea atributelor acestor obiecte. Combinând aceste metode de detecție cu metode de

estimare a atributelor precum cele prezentate în [13] și [14] vor duce la o mai bună precizie de estimare, și implicit la o mai bună performanță a funcțiilor implicate. Evaluarea rezoluțiilor senzorilor optici și a configurațiilor stereo (distanța și orientarea relativă dintre camere) necesare pentru a atinge performanța de estimare a mișcării relative a obiectelor la diferite distanțe poate fi o direcție promițătoare de cercetare pentru sisteme 360° de camere ale căror arii de percepție se suprapun.



Figura 81 Folosirea de super-pixeli și metode de odometrie vizuală pentru percepția scenelor dinamice [14]

6. Concluzii finale

6.1. Concluzii generale

Lucrarea de față abordează problema estimării odometriei pe baza unei camere video stereo sau a unei camere video mono complementată de un senzor de măsurare a distanțierelor.

Odometria vizuală este o problemă consacrată și intens studiată în literatura de specialitate aşa cum am arătat în paginile care descriu stadiul actual al cunoașterii, punctând atât primele abordări care au pus bazele teoretice ale odometriei vizuale și în general ale viziunii artificiale cât și primele metode care au funcționat în timp real. De asemenea am făcut un studiu al principalelor metode de robustificare a algoritmilor de odometrie vizuală pe bază de trăsături, metode care stau la baza contribuțiilor din această lucrare. Considerăm ca metodele de odometrie vizuală bazate pe o cameră mono, stereo sau omnidirecțională vor evoluă în trei direcții: sisteme multi-cameră, fuziune cu alți senzori, utilizarea rețelelor neuronale convoluționale. Din acest motiv pentru care am dedicat cate un subcapitol de analiză al stadiului actual al cunoașterii în aceste trei direcții, punctând și posibilele dezvoltări ulterioare ale contribuțiilor din această teză în cele trei direcții menționate.

Lucrarea construiește o formulare matematică coerentă pentru modelele stereo de odometrie vizuală și detaliază ulterior aspectele specifice legat de formularea problemei de optimizare pentru estimarea odometriei vizuale mono și stereo ca și un sistem liniar de ecuații.

Din punct de vedere al implementării practice explicăm în detaliu trăsăturile folosite pentru odometria vizuală și prezentăm apoi rezultatele experimentale obținute folosind metode liniare de rezolvare a problemei comparativ cu metode de optimizare non-liniară, concluzia fiind că metodele liniare propuse ating aceeași performanță ca și metodele liniare din literatură și necesită resurse de calcul mai puține.

Punctăm de asemenea că generalizarea metodelor stereo pentru a fi utilizate pentru alți senzori de adâncime (de exemplu LiDAR) este o

contribuție semnificată a acestei lucrări. De asemenea metodele de postprocesare a traiectoriei obținute pe bază de senzori LiDAR pentru a reduce erorile de acumulate la concatenarea măsurătorilor succesive ale odometriei vin să completeze metodele de estimare dintre imagini succesive, scăzând eroarea de poziționare pe bază de odometrie vizuală.

Estimarea vitezei relative a obiectelor în mișcare folosind metode de odometrie vizuală este o contribuție care merită exploatață în continuare ca și aplicație a metodelor de odometrie vizuală.

6.2. Originalitatea și contribuțiile inovative ale tezei

Din punct de vedere formal, putem grupa contribuțiile și inovațiile acestei lucrări în aspecte teoretice și aspecte practice.

Contribuțiile teoretice sunt următoarele:

- ✓ Dezvoltarea unui model matematic de estimare a odometriei vizuale prin rezolvarea unui sistem de ecuații [6]
- ✓ Particularizarea modelului liniar de estimare a odometriei vizuale pentru a estima rotația camerei folosind puncte de la distanță [12]
- ✓ Dezvoltarea unui model analitic de propagare a erorilor pe care le prezintă trăsăturile folosite în odometria vizuală pentru a estima erorile de estimare ale mișcării [13]
- ✓ Dezvoltarea unei generalizări a metodei liniare de odometrie vizuală care permite utilizarea acesteia și în scenarii în care rotația relativă dintre imagini succesive este mare, eventual folosind rotația estimată din puncte de la distanță [16]
- ✓ Dezvoltarea unui model matematic pentru a corecta erorile cumulate la concatenarea măsurătorilor succeseive de odometrie vizuală pe bază de LiDAR [17]

Din punct de vedere practic, punctăm următoarele contribuții:

- ✓ Evaluarea metodelor liniare de odometrie vizuala comparativ cu metode non-lineare [5] si cu metode bazate pe tensorul trifocal
- ✓ Paralelizarea algoritmilor de odometrie vizuala pe arhitecturi paralele de calcul [7]
- ✓ Studiul trăsăturilor și a descriptorilor din literatura de specialitate și evaluarea unor metode de robustificare a algoritmilor de odometrie vizuala pe baza de descriptori invariante la schimbarea condițiilor de iluminare [10]
- ✓ Preprocesarea și verificarea trăsăturilor de odometrie vizuala pentru a elimina trăsăturile eronate: consistența unghiurilor dintre perechi de trăsături, verificare circulară, inițializarea metodelor diferențiale și reducerea spațiului de căutare pe baza mișcării estimate anterior [8][9][10]
- ✓ Generalizarea metodei liniare de odometrie vizuala propusă în [6] pentru a folosi alte surse de informație de adâncime precum LiDAR [11]
- ✓ Folosirea metodelor de odometrie vizuala pentru a estima vitezele relative dintre obiecte [14][15]

6.3. Posibile dezvoltări ulterioare

Punctăm câteva direcții noi care considerăm merită explorare în continuare:

- ✓ Folosirea de trăsături dedicate pentru odometrie vizuala, calculate folosind rețele neuronale convoluționale
- ✓ Folosirea de hărți de adâncime pe bază de rețele neuronale convoluționale
- ✓ Evaluarea impactului imperfecțiunilor de modelare a distorsiunilor
- ✓ Robustificarea metodei liniare de odometrie vizuala prin ponderarea diferită a ecuațiilor din sistem

- ✓ Folosirea odometrie vizuale în contextul localizării și cartografierii simultane
- ✓ Generalizarea modelului matematic liniar de estimare a odometrie vizuală pentru a considera mai multe imagini consecutive (sliding window)
- ✓ Generalizarea modelului matematic liniar de estimare a odometrie vizuală pentru sisteme cu mai multe camere

Încheiem prin a menționa faptul că deși metodele de odometria vizuală au ajuns la o maturitate suficient de mare pentru a fi incluse în produse disponibile pentru publicul larg, oportunitățile de cercetare în domeniu rămân în continuare numeroase mai ales datorită unor factori precum: creșterea rezoluțiilor camerelor, nevoia de sisteme multi-cameră pentru o percepție surround view, disponibilitatea puterii de calcul crescute și a metodelor de percepție pe bază de rețele neuronale convoluționale, sisteme multi-senzor în care măsurările video sunt fuzionate cu informații de la senzori de localizare (ex; GPS), senzori inertial sau alți senzori de percepție (LiDAR, camere infraroșii etc).

REFERINȚE

(obligatoriu pe pagină impară)

- [1] David A. Forsyth, Jean Ponce, Computer Vision: A Modern Approach, 2nd Edition, November 5, 2011
- [2] Richard Hartley, Andrew Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2nd edition, April 19, 2004
- [3] Richard Szeliski, "Computer Vision: Algorithms and Applications", Springer, November 24, 2010
- [4] Trucco, Emanuele, and Alessandro Verri. Introductory techniques for 3-D computer vision. Vol. 201. Englewood Cliffs: Prentice Hall, 1998.
- [5] Golban, Cătălin, Ionuț Golban, and Sergiu Nedevschi. "Vision based three-dimensional vehicle motion detection by minimizing nonlinear functions." Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing. IEEE, 2010.
- [6] Golban, Cătălin, and Sergiu Nedevschi. "Linear vs. non linear minimization in stereo visual odometry." 2011 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2011.
- [7] Istvan, Szakats, Cătălin Golban, and Sergiu Nedevschi. "Fast vision based ego-motion estimation from stereo sequences—A GPU approach." 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2011.
- [8] Nedevschi, Sergiu, Cătălin Golban, and Cosmin Mitran. "Improving accuracy for Ego vehicle motion estimation using epipolar geometry." 2009 12th International IEEE Conference on Intelligent Transportation Systems. IEEE, 2009.
- [9] Golban, Cătălin, Cosmin Mitran, and Sergiu Nedevschi. "A practical method for ego vehicle motion estimation from video." 2009 5th International Conference on Intelligent Computer Communication and Processing. IEEE, 2009.
- [10] Golban, Cătălin, Szakats Istvan, and Sergiu Nedevschi. "Stereo based visual odometry in difficult traffic scenes." 2012 IEEE Intelligent Vehicles Symposium. IEEE, 2012.
- [11] Golban, Catalin, Petrut Cobarzan, and Sergiu Nedevschi. "A Comparison Study on Replacing Stereo Disparity with LiDAR in Visual Odometry Methods." 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2020.
- [12] Golban, Catalin, and Sergiu Nedevschi. "An experiment on relative rotation estimation from distant points with monocular vision." 2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2013.

- [13] Golban, Catalin, Petrut Cobarzan, and Sergiu Nedevschi. "Direct formulas for stereo-based visual odometry error modeling." 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2015.
- [14] Golban, Catalin, and Sergiu Nedevschi. "Speed estimation for scene objects using stereo visual odometry methods." 2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2013.
- [15] Golban, Catalin, and Sergiu Nedevschi. "Moving rigid objects segmentation in 3D dynamic traffic scenes using a stereovision system." 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2014.
- [16] Golban, Catalin, Petrut Cobarzan, and Sergiu Nedevschi. "On visual odometry estimation using linear methods - a mathematical framework" 2020 Automation, Computers, Applied Mathematics (ACAM).
- [17] Golban, Catalin, Petrut Cobarzan, and Sergiu Nedevschi. "Visual Odometry Drift Reduction Based on LiDAR Point Clouds Alignment" 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2021.
- [18] Hartley, Richard I. "Estimation of relative camera positions for uncalibrated cameras." European conference on computer vision. Springer, Berlin, Heidelberg, 1992.
- [19] Hartley, Richard I. "In defense of the eight-point algorithm." IEEE Transactions on pattern analysis and machine intelligence 19.6 (1997): 580-593.
- [20] Horn, Berthold KP. "Recovering baseline and orientation from essential matrix." J. Opt. Soc. Am 110 (1990).
- [21] Faugeras, Olivier, and Bernard Mourrain. "On the geometry and algebra of the point and line correspondences between n images." Proceedings of IEEE International Conference on Computer Vision. IEEE, 1995.
- [22] Faugeras, Olivier, and Luc Robert. "What can two images tell us about a third one?." International Journal of Computer Vision 18.1 (1996): 5-19.
- [23] Hartley, Richard I. "Lines and points in three views and the trifocal tensor." International Journal of Computer Vision 22.2 (1997): 125-140.
- [24] Hartley, Richard I. "Computation of the quadrifocal tensor." European Conference on Computer Vision. Springer, Berlin, Heidelberg, 1998.
- [25] Hartley, R. "Multilinear relationships between coordinates of corresponding image points and lines." Proc. International Workshop on Computer Vision and Applied Geometry. 1995.
- [26] Birchfield, Stan. "An introduction to projective geometry (for computer vision)." Unpublished note, Stanford university (1998).
- [27] Mohr, Roger, and Bill Triggs. "Projective geometry for image analysis." 1996.

- [28] Elias, Rimon, and Robert Laganiere. "Projective geometry for three-dimensional computer vision." Seventh World Multiconference on Systemics, Cybernetics and Informatics. Vol. 5. 2003.
- [29] Pollefeys, Marc. "Visual 3D Modeling from Images." VMV. 2004.
- [30] Scaramuzza, Davide & Siegwart, Roland. (2008). Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles. *Robotics, IEEE Transactions on*. 24. 1015 - 1026. 10.1109/TRO.2008.2004490.
- [31] Hartley, Richard I. "An investigation of the essential matrix." GE CRD, Schenectady, NY, Tech. Rep (1995).
- [32] Tian, Tina Yu, Carlo Tomasi, and David J. Heeger. "Comparison of approaches to egomotion computation." *Proceedings cvpr ieee computer society conference on computer vision and pattern recognition*. IEEE, 1996.
- [33] Bruss, Anna R., and Berthold KP Horn. "Passive navigation." *Computer Vision, Graphics, and Image Processing* 21.1 (1983): 3-20.
- [34] MacLean, W. James, Allan D. Jepson, and Richard C. Frecker. "Recovery of Egomotion and Segmentation of Independent Object Motion Using the EM Algorithm." *BMVC*. 1994.
- [35] Prazdny, K. "Egomotion and relative depth map from optical flow." *Biological cybernetics* 36.2 (1980): 87-102.
- [36] Tomasi, Carlo, and Jianbo Shi. "Direction of heading from image deformations." *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1993.
- [37] Ni, Kai and Frank Dellaert. "Stereo Tracking and Three-Point/One-Point Algorithms - A Robust Approach in Visual Odometry." *2006 International Conference on Image Processing* (2006): 2777-2780.
- [38] Baker, Simon, and Iain Matthews. "Lucas-kanade 20 years on: A unifying framework." *International journal of computer vision* 56.3 (2004): 221-255.
- [39] Kaess, Michael, Kai Ni, and Frank Dellaert. "Flow separation for fast and robust stereo odometry." *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009.
- [40] Mitiche, Amar, Steve Seida, and J. K. Aggarwal. "Determining position and displacement in space from images." in *Proceedings of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, (Sar. Francisco, CA), pp. 504-509, June 19 23, 1985.
- [41] Mitiche, Amar, Steven Seida, and J. K. Aggarwal. "Line-based computation of structure and motion using angular invariance." in *Proceedings of IEEE Computer Society Workshop on Motion*, pp. 173-180, May 1986.

- [42] Aggarwal, J., and Y. Wang. "Analysis of a sequence of images using point and line correspondences." Proceedings. 1987 IEEE International Conference on Robotics and Automation. Vol. 4. IEEE, 1987.
- [43] C. Harris and M. Stephens, A Combined Corner and Edge Detector, Proc. Fourth Alvey Vision Conference, pp.147-151,1988.
- [44] M. Fischler and R. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography, Commun. Assoc. Comp. Mach., 24:381-395, 1981
- [45] Jianbo Shi and Carlo Tomasi, "Good features to track", Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognition., pp 593-600, 1999
- [46] E. Rosten, G. Reitmayr, T. Drummond, "Real-time video annotations for augmented reality", International Symposium on Visual Computing, 2005
- [47] E. Rosten, T. Drummond, "Machine learning for high-speed corner detection", Eunice on Computer Vision, 2006, pp. 430-443
- [48] S. M. Smith and J. M. Brady, "SUSAN – a new approach to low level image processing", International Journal of Computer Vision, May 1997, pp. 45-78
- [49] M. Trajkovic, M. Hedley, "Fast corner detection. Image and Vision Computing", 1998, pp.75-87
- [50] Jon Shlens, A tutorial on principal component analysis, Derivation, Discussion and Singular Value Decomposition, 25 March 2003, Version 1, available online https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf
- [51] H. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover", Tech Report CMU-RI-TR-3 Carnegie-Mellon University - Robotics Institute, 1980
- [52] Comport, Andrew I., Ezio Malis, and Patrick Rives. "Accurate quadrifocal tracking for robust 3d visual odometry." Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007.
- [53] Comport, Andrew I., Ezio Malis, and Patrick Rives. "Real-time quadrifocal visual odometry." The International Journal of Robotics Research 29.2-3 (2010): 245-266.
- [54] Alberto, S., Florez, R., Fremont, V., & Bonnifait, P. (2009, October). An experiment of a 3d real-time robust visual odometry for intelligent vehicles. In 2009 12th International IEEE Conference on Intelligent Transportation Systems (pp. 1-6). IEEE.
- [55] Oniga, Florin, Alexandra Trif, and Sergiu Nedevschi. "Stereovision for obstacle detection on smart mobile devices: First results." 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, 2013.
- [56] Vancea, Cristian, and Sergiu Nedevschi. "Analysis of different image rectification approaches for binocular stereovision systems." Proceedings of IEEE 2nd

International Conference on Intelligent Computer Communication and Processing (ICCP 2006). Vol. 1. 2006.

- [57] Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. Proceedings of the Seventh IEEE International Conference on Computer Vision, 1, 666-673 vol.1.
- [58] 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.
- [59] Franke, U., Rabe, C., Badino, H., & Gehrig, S. (2005, August). 6d-vision: Fusion of stereo and motion for robust environment perception. In Joint Pattern Recognition Symposium (pp. 216-223). Springer, Berlin, Heidelberg.
- [60] Badino, Hernán, and Takeo Kanade. "A Head-Wearable Short-Baseline Stereo System for the Simultaneous Estimation of Structure and Motion." MVA. 2011.
- [61] Chang, Ping-Lin, Ankur Handa, Andrew J. Davison, Danail Stoyanov and Philip J. Edwards. "Robust Real-Time Visual Odometry for Stereo Endoscopy Using Dense Quadrifocal Tracking." IPCAI (2014).
- [62] Grasa, Óscar G., Ernesto Bernal, Santiago Casado, Ismael Gil and J. M. M. Montiel. "Visual SLAM for Hand-Held Monocular Endoscope." (2013).
- [63] Lourenço, Miguel & Stoyanov, Danail & Barreto, Joao P. (2014). Visual Odometry in Stereo Endoscopy by Using PEaRL to Handle Partial Scene Deformation. 33-40. 10.1007/978-3-319-10437-9_4.
- [64] Stap, Nanda van der, Ferdinand van der Heijden and Ivo Amj Broeders. "Towards automated visual flexible endoscope navigation." Surgical Endoscopy 27 (2013): 3539-3547.
- [65] Turan, Mehmet, Yasin Almalioğlu, Helder Araújo, Ender Konukoglu and Metin Sitti. "Deep EndoVO: A recurrent convolutional neural network (RCNN) based visual odometry approach for endoscopic capsule robots." Neurocomputing 275 (2018): 1861-1870.
- [66] Dimas, George & Iakovidis, Dimitris & Karargyris, Alexandros & Ciuti, Gastone & Koulaouzidis, Anastasios. (2017). An artificial neural network architecture for non-parametric visual odometry in wireless capsule endoscopy. Measurement Science and Technology. 28. 10.1088/1361-6501/aa7ebf.
- [67] Turan, Mehmet & Ornek, Evin & Ibrahimli, Nail & Giracoglu, Can & Almalioğlu, Yasin & Yanik, Mehmet & Sitti, Metin. (2018). Unsupervised Odometry and Depth Learning for Endoscopic Capsule Robots. 1801-1807. 10.1109/IROS.2018.8593623.
- [68] Rusu, Radu Bogdan, and Steve Cousins. "3d is here: Point cloud library (pcl)." 2011 IEEE international conference on robotics and automation. IEEE, 2011.
- [69] Pomerleau, F., Colas, F., Siegwart, R., & Magnenat, S. (2013). Comparing ICP variants on real-world data sets. Autonomous Robots, 34(3), 133-148.

- [70] Zhang, Ji, and Sanjiv Singh. "LOAM: Lidar Odometry and Mapping in Real-time." *Robotics: Science and Systems*. Vol. 2. 2014.
- [71] Zhang, Ji, and Sanjiv Singh. "Visual-lidar odometry and mapping: Low-drift, robust, and fast." 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015.
- [72] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012.
- [73] Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11), 1231-1237.
- [74] Kazik, T., Kneip, L., Nikolic, J., Pollefeys, M., & Siegwart, R. (2012, June). Real-time 6d stereo visual odometry with non-overlapping fields of view. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1529-1536). IEEE.
- [75] Hee Lee, Gim, Friedrich Fraundorfer, and Marc Pollefeys. "Motion estimation for self-driving cars with a generalized camera." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.
- [76] Pless, Robert. "Using many cameras as one." 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.. Vol. 2. IEEE, 2003.
- [77] Scaramuzza, Davide, Friedrich Fraundorfer, and Roland Siegwart. "Real-time monocular visual odometry for on-road vehicles with 1-point ransac." 2009 IEEE International Conference on Robotics and Automation. IEEE, 2009.
- [78] Scaramuzza, D., Fraundorfer, F., Pollefeys, M., & Siegwart, R. (2009, October). Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In 2009 IEEE 12th International Conference on Computer Vision (pp. 1413-1419). IEEE.
- [79] Oskiper, T., Zhu, Z., Samarakone, S., & Kumar, R. (2007, June). Visual odometry system using multiple stereo cameras and inertial measurement unit. In 2007 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.
- [80] Warren, Michael, and Ben Upcroft. "High altitude stereo visual odometry." Proceedings of Robotics: Science and Systems IX (2013).
- [81] Warren, Michael, David McKinnon, and Ben Upcroft. "Online calibration of stereo rigs for long-term autonomy." 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013.
- [82] Lategahn, H., Geiger, A., Kitt, B., & Stiller, C. (2012, June). Motion-without-structure: Real-time multipose optimization for accurate visual odometry. In 2012 IEEE Intelligent Vehicles Symposium (pp. 649-654). IEEE.

- [83] Geiger, A., Ziegler, J., & Stiller, C. (2011, June). Stereoscan: Dense 3d reconstruction in real-time. In 2011 IEEE Intelligent Vehicles Symposium (IV) (pp. 963-968). IEEE.
- [84] Schubert, Robin, Eric Richter, and Gerd Wanielik. "Comparison and evaluation of advanced motion models for vehicle tracking." 2008 11th International Conference on Information Fusion. IEEE, 2008.
- [85] Badino, H., Yamamoto, A., & Kanade, T. (2013). Visual odometry by multi-frame feature integration. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 222-229).
- [86] Krešo, Ivan, and Siniša Šegvić. "Improving the egomotion estimation by correcting the calibration bias." 10th International Conference on Computer Vision Theory and Applications. 2015.
- [87] Hartley, Richard, and Sing Bing Kang. "Parameter-free radial distortion correction with center of distortion estimation." IEEE Transactions on Pattern Analysis and Machine Intelligence 29.8 (2007): 1309-1321.
- [88] Godard, Clément, Oisin Mac Aodha, and Gabriel J. Brostow. "Unsupervised monocular depth estimation with left-right consistency." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017
- [89] Amiri, Ali Jahani, Shing Yan Loo, and Hong Zhang. "Semi-supervised monocular depth estimation with left-right consistency using deep neural network." 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2019
- [90] Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1851-1858).
- [91] Tang, J., Ambrus, R., Guizilini, V., Pillai, S., Kim, H., & Gaidon, A. (2019). Self-Supervised 3D Keypoint Learning for Ego-motion Estimation. arXiv preprint arXiv:1912.03426.
- [92] Yang, N., Stumberg, L. V., Wang, R., & Cremers, D. (2020). D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1281-1292).
- [93] Yang, N., Wang, R., Stuckler, J., & Cremers, D. (2018). Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 817-833).
- [94] Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In Advances in neural information processing systems (pp. 2366-2374).

- [95] Witt, Jonas, and Uwe Weltin. "Robust stereo visual odometry using iterative closest multiple lines." 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2013
- [96] Irani, Michal, and P. Anandan. "About direct methods." International Workshop on Vision Algorithms. Springer, Berlin, Heidelberg, 1999.
- [97] R. Gomez-Ojeda and J. Gonzalez-Jimenez, "Robust stereo visual odometry through a probabilistic combination of points and line segments," in Robotics and Automation (ICRA), 2016 IEEE International Conference on, 2016.
- [98] Lovegrove, Steven, Andrew J. Davison, and Javier Ibanez-Guzmán. "Accurate visual odometry from a rear parking camera." 2011 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2011
- [99] Forster, Christian, Matia Pizzoli, and Davide Scaramuzza. "SVO: Fast semi-direct monocular visual odometry." 2014 IEEE international conference on robotics and automation (ICRA). IEEE, 2014.
- [100] Engel, Jakob, Vladlen Koltun, and Daniel Cremers. "Direct sparse odometry." IEEE transactions on pattern analysis and machine intelligence 40.3 (2017): 611-625.
- [101] Wang, Rui, Martin Schworer, and Daniel Cremers. "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras." Proceedings of the IEEE International Conference on Computer Vision. 2017.
- [102] Engel, Jakob, Thomas Schöps, and Daniel Cremers. "LSD-SLAM: Large-scale direct monocular SLAM." European conference on computer vision. Springer, Cham, 2014.
- [103] Engel, Jakob, Jörg Stückler, and Daniel Cremers. "Large-scale direct SLAM with stereo cameras." 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).IEEE, 2015.
- [104] Engel, Jakob, Jürgen Sturm, and Daniel Cremers. "Semi-dense visual odometry for a monocular camera." Proceedings of the IEEE international conference on computer vision. 2013.
- [105] Schöps, Thomas, Jakob Engel, and Daniel Cremers. "Semi-dense visual odometry for AR on a smartphone." 2014 IEEE international symposium on mixed and augmented reality (ISMAR). IEEE, 2014.
- [106] Godard, C., Mac Aodha, O., Firman, M., & Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In Proceedings of the IEEE international conference on computer vision (pp. 3828-3838).

- [107] DeTone, Daniel, Tomasz Malisiewicz, and Andrew Rabinovich. "Superpoint: Self-supervised interest point detection and description." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018.
- [108] Christiansen, Peter Hviid, et al. "Unsuperpoint: End-to-end unsupervised interest point detector and descriptor." arXiv preprint arXiv:1907.04011 (2019).
- [109] Tang, J., Kim, H., Guizilini, V., Pillai, S., & Ambrus, R. (2019). Neural Outlier Rejection for Self-Supervised Keypoint Learning. arXiv preprint arXiv:1912.10615.
- [110] Tang, J., Ambrus, R., Guizilini, V., Pillai, S., Kim, H., & Gaidon, A. (2019). Self-Supervised 3D Keypoint Learning for Ego-motion Estimation. arXiv preprint arXiv:1912.03426.
- [111] Zhan, H., Weerasekera, C. S., Bian, J. W., & Reid, I. (2020, May). Visual odometry revisited: What should be learnt?. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4203-4210). IEEE.
- [112] Hui, Tak-Wai, Xiaoou Tang, and Chen Change Loy. "Liteflownet: A lightweight convolutional neural network for optical flow estimation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- [113] Sünderhauf, N., Konolige, K., Lacroix, S., & Protzel, P. (2006). Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle. In Autonome Mobile Systeme 2005 (pp. 157-163). Springer, Berlin, Heidelberg.
- [114] Olson, C. F., Matthies, L. H., Schoppers, H., & Maimone, M. W. (2000, June). Robust stereo ego-motion for long distance navigation. In Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Vol. 2, pp. 453-458). IEEE.
- [115] Cheng, Y., Maimone, M., & Matthies, L. (2005, October). Visual odometry on the Mars exploration rovers. In 2005 IEEE International Conference on Systems, Man and Cybernetics (Vol. 1, pp. 903-910). IEEE.
- [116] Maimone, M., Cheng, Y., & Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. Journal of Field Robotics, 24(3), 169-186.
- [117] Nistér, David, Oleg Naroditsky, and James Bergen. "Visual odometry." Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.. CVPR 2004.. Vol. 1. IEEE, 2004.
- [118] Konolige, K., Agrawal, M., & Sola, J. (2007). Large-scale visual odometry for rough terrain. In International symposium on robotics research (Vol. 19, pp. 23-43).
- [119] Agrawal, M., & Konolige, K. (2007, August). Rough terrain visual odometry. In Proceedings of the International Conference on Advanced Robotics (ICAR) (Vol. 25, pp. 28-30).

- [120] Agrawal, M., & Konolige, K. (2006, August). Real-time localization in outdoor environments using stereo vision and inexpensive gps. In 18th International conference on pattern recognition (ICPR'06) (Vol. 3, pp. 1063-1068). IEEE.
- [121] Howard, A. (2008, September). Real-time stereo visual odometry for autonomous ground vehicles. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3946-3952). IEEE.
- [122] Matthies, Larry, and Stevena Shafer. "Error modeling in stereo navigation." IEEE Journal on Robotics and Automation 3.3 (1987): 239-248.
- [123] Matthies, Larry. Dynamic Stereo Vision. PhD thesis, Carnegie Mellon University, October 1989
- [124] Umeyama, Shinji. "Least-squares estimation of transformation parameters between two point patterns." IEEE Transactions on Pattern Analysis & Machine Intelligence 4 (1991): 376-380.
- [125] Arun, K. Soman, Thomas S. Huang, and Steven D. Blostein. "Least-squares fitting of two 3-D point sets." IEEE Transactions on pattern analysis and machine intelligence 5 (1987): 698-700.
- [126] Nistér, David. "An efficient solution to the five-point relative pose problem." IEEE transactions on pattern analysis and machine intelligence 26.6 (2004): 756-770.
- [127] Nistér, David, and Henrik Stewénius. "A minimal solution to the generalised 3-point pose problem." Journal of Mathematical Imaging and Vision 27.1 (2007): 67-79.
- [128] Haralick, B. M., Lee, C. N., Ottenberg, K., & Nölle, M. (1994). Review and analysis of solutions of the three point perspective pose estimation problem. International journal of computer vision, 13(3), 331-356.
- [129] M. Buehler, R. Playter, and M. Raibert. Robots step outside. In International Symposium of Adaptive Motion of Animals and Machines (AMAM), Ilmenau, Germany, Sept 2005
- [130] Engels, Chris, Henrik Stewénius, and David Nistér. "Bundle adjustment rules." Photogrammetric computer vision 2.32 (2006).
- [131] Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (1999, September). Bundle adjustment—a modern synthesis. In International workshop on vision algorithms (pp. 298-372). Springer, Berlin, Heidelberg.
- [132] Lourakis, M., & Argyros, A. (2004). The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt-algorithm (Vol. 1, No. 2, p. 5). Technical Report 340, Institute of Computer Science-FORTH, Heraklion, Crete, Greece.
- [133] Buczko, Martin, and Volker Willert. "Flow-decoupled normalized reprojection error for visual odometry." 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016.

- [134] Buczko, Martin, and Volker Willert. "How to distinguish inliers from outliers in visual odometry for high-speed automotive applications." 2016 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2016.
- [135] Scaramuzza, Davide. "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints." International journal of computer vision 95.1 (2011): 74-85.
- [136] Cvišić, Igor, and Ivan Petrović. "Stereo odometry based on careful feature selection and tracking." 2015 European Conference on Mobile Robots (ECMR). IEEE, 2015.
- [137] Cvišić, Igor, et al. "SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles." Journal of field robotics 35.4 (2018): 578-595.
- [138] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in IEEE Int. Conf. Comput. Vision (ICCV), 2011, pp. 2564–2571.
- [139] Gálvez-López, Dorian, and Juan D. Tardos. "Bags of binary words for fast place recognition in image sequences." IEEE Transactions on Robotics 28.5 (2012): 1188-1197.
- [140] Mur-Artal, Raul, and Juan D. Tardós. "ORB-SLAM2: An open-source slam system for monocular, stereo, and rgbd cameras." IEEE Transactions on Robotics 33.5 (2017): 1255-1262.
- [141] Buczko, M., Willert, V., Schwehr, J., & Adamy, J. (2018, June). Self-validation for automotive visual odometry. In 2018 IEEE Intelligent Vehicles Symposium (IV) (pp. 1-6). IEEE
- [142] Janai, J., Güney, F., Behl, A., & Geiger, A. (2020). Computer vision for autonomous vehicles: Problems, datasets and state of the art. Foundations and Trends® in Computer Graphics and Vision, 12(1-3), 1-308.
- [143] Hussain, R., & Zeadally, S. (2018). Autonomous cars: Research results, issues, and future challenges. IEEE Communications Surveys & Tutorials, 21(2), 1275-1313.
- [144] Tian, H., Wang, T., Liu, Y., Qiao, X., & Li, Y. (2020). Computer vision technology in agricultural automation—A review. Information Processing in Agriculture, 7(1), 1-19.
- [145] Kapach, K., Barnea, E., Mairon, R., Edan, Y., & Ben-Shahar, O. (2012). Computer vision for fruit harvesting robots—state of the art and challenges ahead. International Journal of Computational Vision and Robotics, 3(1-2), 4-34.
- [146] Kanellakis, C., & Nikolakopoulos, G. (2017). Survey on computer vision for UAVs: Current developments and trends. Journal of Intelligent & Robotic Systems, 87(1), 141-168.
- [147] Lepetit, Vincent. "On computer vision for augmented reality." 2008 international symposium on ubiquitous virtual reality. IEEE, 2008.

- [148] Miller, Ian. "Computer Vision on the Battlefield: Can Machines Distinguish Between Enemy and Civilian in Military Urban Operations?." Available at SSRN 3310951 (2017).
- [149] Svenmarck, P., Luotsinen, L., Nilsson, M., & Schubert, J. (2018, May). Possibilities and challenges for artificial intelligence in military applications. In Proceedings of the NATO Big Data and Artificial Intelligence for Military Decision Making Specialists' Meeting.
- [150] Matthies, Larry, et al. "Computer vision on Mars." International Journal of Computer Vision 75.1 (2007): 67-92.
- [151] Malamas, E. N., Petrakis, E. G., Zervakis, M., Petit, L., & Legat, J. D. (2003). A survey on industrial vision systems, applications and tools. *Image and vision computing*, 21(2), 171-188.
- [152] Al-Kaff, A., Martin, D., Garcia, F., de la Escalera, A., & Armingol, J. M. (2018). Survey of computer vision algorithms and applications for unmanned aerial vehicles. *Expert Systems with Applications*, 92, 447-463.
- [153] Badino, Hernán, Uwe Franke, and Rudolf Mester. "Free space computation using stochastic occupancy grids and dynamic programming." Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil. Vol. 20. Citeseer, 2007.
- [154] Oniga, Florin, and Sergiu Nedevschi. "Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection." *IEEE Transactions on Vehicular Technology* 59.3 (2009): 1172-1182.
- [155] Oniga, Florin, Sergiu Nedevschi, and Marc Michael Meinecke. "Curb detection based on a multi-frame persistence map for urban driving scenarios." 2008 11th International IEEE Conference on Intelligent Transportation Systems. IEEE, 2008.
- [156] Rabe, Clemens, Uwe Franke, and Stefan Gehrig. "Fast detection of moving objects in complex scenarios." 2007 IEEE Intelligent Vehicles Symposium. IEEE, 2007.
- [157] Livyatan, H., Berberian, O., Cohen, B., & Stein, G. (2018). U.S. Patent No. 9,959,595. Washington, DC: U.S. Patent and Trademark Office.
- [158] Ozyesil, O., Voroninski, V., Basri, R., & Singer, A. (2017). A survey of structure from motion. *arXiv preprint arXiv:1701.08493*.
- [159] Schonberger, Johannes L., and Jan-Michael Frahm. "Structure-from-motion revisited." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
- [160] Chellappa, Rama, Gang Qian, and Sridhar Srinivasan. "Structure from motion: Sparse versus dense correspondence methods." Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348). Vol. 2. IEEE, 1999.
- [161] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I. The essential algorithms," *Robot. Automat. Mag.*, vol. 13, no. 2, pp. 99-110, 2006.

- [162] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (SLAM): Part II. State of the art," *Robot. Automat. Mag.*, vol. 13, no. 3, pp. 108–117, 2006.
- [163] Taketomi, Takafumi, Hideaki Uchiyama, and Sei Ikeda. "Visual SLAM algorithms: a survey from 2010 to 2016." *IPSJ Transactions on Computer Vision and Applications* 9.1 (2017): 16.
- [164]]Yousif, Khalid, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. "An overview to visual odometry and visual SLAM: Applications to mobile robotics." *Intelligent Industrial Systems* 1.4 (2015): 289-311.
- [165] Fuentes-Pacheco, Jorge, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. "Visual simultaneous localization and mapping: a survey." *Artificial intelligence review* 43.1 (2015): 55-81.
- [166] Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D. Tardos. "ORB-SLAM: a versatile and accurate monocular SLAM system." *IEEE transactions on robotics* 31.5 (2015): 1147-1163.
- [167] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proc. Int. Symp. Mixed and Augmented Reality*, 2007, pp. 225–234.
- [168] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. Int. Conf. Computer Vision*, 2003, pp. 1403–1410.
- [169] "Road Experience Management", Mobileye, <https://www.mobileye.com/our-technology/rem/>
- [170] Dellaert, Frank. "Visual SLAM Tutorial: Bundle Adjustment." (2014)
- [171] Thrun, Sebastian, and Michael Montemerlo. "The graph SLAM algorithm with applications to large-scale mapping of urban structures." *The International Journal of Robotics Research* 25.5-6 (2006): 403-429.
- [172] Scaramuzza, Davide, and Zichao Zhang. "Visual-inertial odometry of aerial robots." *arXiv preprint arXiv:1906.03289* (2019).
- [173] Wirth, Stephan, Pep Lluis Negre Carrasco, and Gabriel Oliver Codina. "Visual odometry for autonomous underwater vehicles." *2013 MTS/IEEE OCEANS-Bergen*. IEEE, 2013.
- [174] Etxeberria-Garcia, M., Labayen, M., Zamalloa, M., & Arana-Arexolaleiba, N. (2020, January). Application of Computer Vision and Deep Learning in the railway domain for autonomous train stop operation. In *2020 IEEE/SICE International Symposium on System Integration (SII)* (pp. 943-948). IEEE.
- [175] Tschopp, F., Schneider, T., Palmer, A. W., Nourani-Vatani, N., Cadena, C., Siegwart, R., & Nieto, J. (2019). Experimental comparison of visual-aided odometry methods for rail vehicles. *IEEE Robotics and Automation Letters*, 4(2), 1815-1822.

- [176] Zaman, Shahzad, et al. "Cost-effective visual odometry system for vehicle motion control in agricultural environments." *Computers and Electronics in Agriculture* 162 (2019): 82-94.
- [177] Scaramuzza, Davide and Friedrich Fraundorfer. "Visual Odometry [Tutorial]." *IEEE Robotics & Automation Magazine* 18 (2011): 80-92.
- [178] Fraundorfer, Friedrich and Davide Scaramuzza. "Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications." *IEEE Robotics & Automation Magazine* 19 (2012): 78-90.
- [179] Martin Rebert, David Monnin, Stéphane Bazeille, and Christophe Cudel "A review of the dataset available for visual odometry", Proc. SPIE 11172, Fourteenth International Conference on Quality Control by Artificial Vision, 111720W (16 July 2019)
- [180] Poddar, S., Kottath, R., & Karar, V. (2018). Evolution of Visual Odometry Techniques. arXiv preprint arXiv:1804.11142.
- [181] Aqel, Mohammad & Marhaban, Mohammad Hamiruce & Saripan, M Iqbal & Ismail, Napsiah. (2016). Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*. 5. 10.1186/s40064-016-3573-7.
- [182] B. Zhao, T. Hu and L. Shen, "Visual odometry - A review of approaches," 2015 IEEE International Conference on Information and Automation, Lijiang, 2015, pp. 2569-2573
- [183] T. Koletschka, L. Puig and K. Daniilidis, "MEVO: Multi-environment stereo visual odometry," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 4981-4988, doi: 10.1109/IROS.2014.6943270.
- [184] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, "BRIEF: Binary Robust Independent Elementary Features", 11th European Conference on Computer Vision (ECCV), Heraklion, Crete. LNCS Springer, September 2010
- [185] Fridtjof Stein, "Efficient Computation of Optical Flow Using the Census Transform", Proceedings of the 26th DAGM Symposium, pp. 79-86, 2004
- [186] Ramin Zabih, John Woodfill, "Non-parametric local transforms for computing visual correspondence", ECCV 1994, Proceedings of the Third European Conference on Computer Vision - Volume II, pp. 151-158
- [187] Jerome Martin and James L Crowley, "Comparison of Correlation Techniques", Conference on Intelligent Autonomous Systems, Karlsruhe, March 1995
- [188] C. Pantilie, S. Nedevschi, "SORT-SGM: Subpixel Optimized Real-Time Semiglobal Matching for Intelligent Vehicles", *IEEE Transactions on Vehicular Technology*, Volume 61, Issue 3, pp. 1032 - 1042, 2012
- [189] I. Haller , C. Pantilie , F. Oniga and S. Nedevschi , "Real-time semiglobal dense stereo solution with improved subpixel accuracy" , Proc. IEEE IV Symp. , pp.369 -376 , 2010

- [190] Richard W. Hamming, "Error detecting and error correcting codes", Bell System Technical Journal, pp. 147–160, 1950
- [191] David G. Lowe, "Object recognition from local scale-invariant features", Proceedings of the International Conference on Computer Vision, pp. 1150–1157, 1999
- [192] Mikolajczyk, Krystian, and Cordelia Schmid. "A performance evaluation of local descriptors" IEEE Transactions on Pattern Analysis and Machine Intelligence, pp/ 1615-1630, 2005
- [193] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast Retina Keypoint. In IEEE Conference on Computer Vision and Pattern Recognition, 2012. CVPR 2012 Open Source Award Winner.
- [194] Stefan Leutenegger, Margarita Chli and Roland Siegwart: BRISK: Binary Robust Invariant Scalable Keypoints. ICCV 2011: 2548-2555.
- [195] Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.
- [196] Bay, H. and Tuytelaars, T. and Van Gool, L. "SURF: Speeded Up Robust Features", 9th European Conference on Computer Vision, 2006
- [197] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary R. Bradski, "ORB: An efficient alternative to SIFT or SURF", ICCV 2011,pp. 2564-2571
- [198] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 886–893, June 2005
- [199] Berthold K.P. Horn and Brian G. Schunck, "Determining Optical Flow", Massachusetts Institute of Technology, Artificial Intelligence Laboratory, No. 1-3, pp. 185-203, August 1981
- [200] Aggarwal, J.K. & Nandhakumar, Nagaraj. (1988). On the Computation of Motion from Sequences of Images – a Review. Proceedings of the IEEE. 76. 917 - 935. 10.1109/5.5965.
- [201] Roach, J. W., & Aggarwal, J. K. (1979). Computer tracking of objects moving in space. IEEE Transactions on Pattern Analysis and Machine Intelligence, (2), 127-135.
- [202] Roach, J. W., & Aggarwal, J. K. (1980). Determining the movement of objects from a sequence of images. IEEE Transactions on Pattern Analysis and Machine Intelligence, (6), 554-562.
- [203] Ullman, S. (1979). The interpretation of structure from motion. Proceedings of the Royal Society of London. Series B. Biological Sciences, 203(1153), 405-426.
- [204] Tsai, Ryan & HUANG, TS. (1982). Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch. Acoustics, Speech and Signal Processing, IEEE Transactions on. 29. 1147 - 1152. 10.1109/TASSP.1981.1163710.

- [205] Tsai, R., Huang, T., & Zhu, W. L. (1982). Estimating three-dimensional motion parameters of a rigid planar patch, II: singular value decomposition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(4), 525-534.
- [206] Bruce D. Lucas, Takeo Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", Proceedings of Imaging Understanding Workshop, 1981, pp. 121-130.
- [207] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker. Description of the algorithm", Intel Corporation, 2000.
- [208] D. Demirdjian, T. Darrell, "Motion estimation from disparity images", ICCV 2001, pp: 213 - 218.
- [209] D. Pfeiffer, U. Franke: "Efficient Representation of Traffic Scenes by means of Dynamic Stixels", IEEE Intelligent Vehicles Symposium IV 2010, San Diego, CA, June 2010
- [210] A. Vatavu, R. Danescu, S. Nedevschi, "Real-time dynamic environment perception in driving scenarios using difference fronts", IEEE Intelligent Vehicles Symposium, 2012, pp. 717-722.
- [211] R. Danescu, C. Pantilie, F. Oniga, S. Nedevschi, "Particle Grid Tracking System for Stereovision Based Obstacle Perception in Driving Environments", IEEE Intelligent Transportation Systems Magazine, vol. 4, No. 1, March 2012, pp. 6-20.
- [212] D. Marquardt, "An algorithm for the least-squares estimation of nonlinear parameters," SIAM Journal of Applied Mathematics, vol. 11, pp. 431-441, 1963.
- [213] M. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in c/c++. <http://www.ics.forth.gr/lourakis/levmar/>.
- [214] Tsai, Roger Y., and Thomas S. Huang. "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces." *IEEE Transactions on pattern analysis and machine intelligence* 1 (1984): 13-27.
- [215] Delmerico, J., & Scaramuzza, D. (2018, May). A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2502-2509). IEEE.
- [216] Zhou, D., Dai, Y., & Li, H. (2019). Ground-plane-based absolute scale estimation for monocular visual odometry. *IEEE Transactions on Intelligent Transportation Systems*, 21(2), 791-802.
- [217] B. Delaunay, 'Sur la sphère vide. A la mémoire de Georges Voronoï, Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na, 1934, no. 6, 793-800.
- [218] F. Kallasi, D. L. Rizzini and S. Caselli, "Fast Keypoint Features From Laser Scanner for Robot Localization and Mapping," in IEEE Robotics and Automation Letters, vol. 1, no. 1, pp. 176-183, Jan. 2016, doi: 10.1109/LRA.2016.2517210.

- [219] Longuet-Higgins, H. Christopher. "A computer algorithm for reconstructing a scene from two projections." *Nature* 293.5828 (1981): 133.
- [220] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. *Introduction to Algorithms*. MIT Press and McGraw-Hill.
- [221] Cvišić, Igor, Ivan Marković, and Ivan Petrović. "Recalibrating the KITTI Dataset Camera Setup for Improved Odometry Accuracy." *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021.
- [221] R. Gonzalez, F. Rodriguez, J. L. Guzman, C. Pradalier and R. Siegwart, "Combined visual odometry and visual compass for offroad mobile robots localization," *Robotica*, vol. 30, pp. 865-878, 2012.

LISTA FIGURILOR

Figura 1 Problema odometriei vizuale.....	12
Figura 2 Matricea esențială. Preluare din [214].....	24
Figura 3 Geometria unei camere stereo, indicând incertitudinea de reconstrucție stereo [122].....	26
Figura 4 Creșterea super-liniară a erorilor de poziționare datorită acumulării erorilor de rotație și posibilitatea reducerii efectului cu un senzor de orientare absolută [114]	27
Figura 5 Exemplu de erori cauzate de alunecarea în gol a roților (stânga) pentru robotul mobil din dreapta [120].....	27
Figura 6 Trăsăturile (verde) pe baza cărora se estimează matricea esențială (liniile epipolare cu albastru). Se observă obiectele în mișcare precum și trăsăturile eronate segmentate cu roșu. Estimarea unghiului de yaw într-o succesiune de curbe prima la stânga și a doua la dreapta (albastru - valoarea citită de pe CAN, rosu - estimarea noastră, verde - valoarea filtrată cu filtre Kalman) [9].....	29
Figura 7 Exemplu de mediu în care este evaluată metoda propusă în [121].....	29
Figura 8 Robot from [129] using stereo visual odometry method from [121].....	30
Figura 9 Camera de parcare pentru care se pretează metodele directe [98].....	31
Figura 10 Stânga sus - cameră stereo folosită pentru persoane cu dizabilități [60]. Dreapta sus - trăsături folosite pentru estimarea mișcării, și vectori de mișcare pe obiecte dinamice (pietoni în acest caz). Jos - traекторia 3D și puncte 3D folosite pentru estimarea mișcării dintre imagini consecutive.....	35
Figura 11 Ilustrarea grafică a algoritmului propus în [85] de propagare a mediei trăsăturilor din trecut pentru a îmbunătăți precizia de estimare a odometriei.....	36
Figura 12 În [82] ultimele 4 poziții sunt rafinate formulând o funcție de cost pe baza mișcărilor relative z_{ij} dintre camere și impunând un model cu accelerare liniară și unghiulară constantă [84].....	36
Figura 13 O reprezentare grafică a mediei erorilor de reproiecție din [86] (sus) și magnitudinea acestora (jos).....	37
Figura 14 Traекторii obținute cu metoda din [86] înainte (stânga) și după (dreapta) corecția erorilor de calibrare	38
Figura 15 Exemple de imagini achiziționate cu robotul mobil din [39] - Partea dreaptă indică cu verde un exemplu de scenariu în care estimarea separată a translației față de rotație crește acuratețea estimării	39
Figura 16 Exemplul din [134] cu erori standard de reproiecție (sus) comparativ cu erori normalizare (jos). Liniile verticale din partea de jos sunt proporționale în lungime cu erorile de reproiecție folosite pentru eliminarea corespondențelor eronate de trăsături. Culoarea roșie este pentru trăsăturile eliminate, iar verde pentru cele folosite în procesul de minimizare liniară pentru a determina odometria. Se observă că dacă normalizează erorile de reproiecție, trăsăturile corecte din apropiere sunt păstrate	40
Figura 17 Varianta simplificată a matricii esențiale considerând modelul Ackerman de mișcare [135] (centrul axului spate al mașinii se mișcă pe un arc de cerc, iar sistemul de coordonate al camerei este considerat a fi la o distanță L în fața camerei)	41

Figura 18 Estimarea magnitudinii vectorului de translație pe baza câmpurilor vizuale care nu se suprapun [74].....	42
Figura 19 Stânga - sistem centralizat de cameră, prin introducerea unei camere virtuale V și generalizarea constrângerii epipolare relativ la această cameră virtuală [76]. Dreapta - tipuri de corespondențe de trăsături utilizate pentru estimarea mișcării între două achiziții succesive de imagini.....	43
Figura 20 Sistem de două camere, orientate opus una față de celalătă [79].....	43
Figura 21 Sistemul folosit în [80] și o traiectorie obținută dintr-un zbor de 2.75km la altitudine de 20m/s la altitudini de 20-100m.....	44
Figura 22 Hărți indoor (sus) și outdoor (jos) construite folosind metoda prezentată în [70].....	46
Figura 23 Ilustrarea algoritmului din [89].....	48
Figura 24 [92].....	49
Figura 25 Ilustrarea algoritmului din [110].....	50
Figura 26 Modelul de proiecție perspectivă.....	52
Figura 27 Cordonate normalize.....	54
Figura 28 Parametrii extrinseci	58
Figura 29 Geometria unui sistem stereo de camere rectificate.....	59
Figura 30 Matricea esențială. Figură preluată din [9].....	64
Figura 31 Corespondențe de trăsături între imagini succesive.....	67
Figura 32 $\sin(\alpha) - \alpha$ for $\alpha \in [-3^\circ, 3^\circ]$	74
Figura 33 Norma Forbenius a diferenței dintre matricea de rotație și aproximarea acesteia. Parametrii α și β sunt variați în intervalul $[-3^\circ, 3^\circ]$ iar γ este păstrat constant la 0° (stânga) respectiv 3° (dreapta).....	75
Figura 34 Etapele de estimare a odometriei vizuale pe bază de trăsături.....	89
Figura 35 Trăsătura 1 are grad scăzut de distinctivitate; trăsătura 2 este pe o muchie și se aseamănă cu 2' care este de asemenea în lungul muchiei; trăsătura 3 este de tip corner, fiind la intersecția dintre 2 muchii.....	90
Figura 36 Colturi Harris comparate cu "good features to track". Curbele de nivel pentru $T_{H,u,v}(\lambda_{1,u,v}, \lambda_{2,u,v})$ respectiv zona hașurată cu verde pentru un prag pe $T_{S,u,v}(\lambda_{1,u,v}, \lambda_{2,u,v}) > \lambda_{min}$	93
Figura 37 Un cerc Bresenham, discretizat în jurul pixelului central (p în imaginea din dreapta).....	98
Figura 38 Linii între perechile de puncte selectate pentru BRIEF. Fiecare dintre cele 4 imagini indică 128 de perechi.....	103
Figura 39 Calculul descriptorului SIFT relativ la orientarea globală a descriptorului	104
Figura 40 Reprezentare piramidală a unei imagini. Sursa: https://en.wikipedia.org/wiki/Pyramid_(image_processing)	108
Figura 41 Estimarea circulară a corespondențelor de trăsături între două perechi de imagini stereo succesive.....	110
Figura 42 Modificarea asocierii circulare în cazul în care folosim predicția pentru a inițializa metodele diferențiale sau pentru a reduce spațiul de căutare. Cu albastru sunt reprezentate asocierile care beneficiază de predicție.....	111

Figura 43 Filtrarea trăsăturilor pentru estimarea odometriei vizuale. Cu roșu sunt reprezentate trăsăturile care sunt eliminate prin asociere circulară, cu galben sunt reprezentate trăsăturile eliminate prin procesul de RANSAC, cu verde sunt reprezentate trăsăturile care verifică mișcarea estimată.....	116
Figura 44 Estimarea unghiului yaw (stânga) pentru imagini stereo consecutive, respectiv sărind 1 din 2 imagini.....	117
Figura 45 Estimarea translației cu metode ne-lineare (stânga). În dreapta este ilustrată estimarea făcută pe tot setul de trăsături, fără a folosi RANSAC.....	117
Figura 46 O reprezentare top view a traectoriei obținute prin integrarea valorilor odometriei vizuale estimate pe un set de 1200 de imagini consecutive (dreapta) la parcurgerea de două ori a ariei geografie din stânga.....	117
Figura 47 Comparație cantitativă între estimarea pe bază de tensorul trifocal (graficele de jos) și metoda ne-liniară de minimizare a erorii de reproiecție (graficele de sus). Pe coloana din stânga este unghiul yaw, exprimat în grade inclusiv valorile citite de pe magistrala CAN a mașinii, iar pe coloana din dreapta valorile translației exprimate în metrii.....	118
Figura 48 Traекторia obținută prin metode ne-liniare pe baza erorii de re-proiecție (mijloc) respectiv pe baza tensorului trifocal (dreapta) la parcurgerea de trei ori al unui sens giratoriu (stânga).....	118
Figura 49 O comparație a traectoriilor generate cu metoda liniară (verde) și metoda ne-lineare (albastru). Includem ca și referință traectoria obținută pe baza unor valori de translație și yaw rate citite de pe magistrala CAN a mașinii.....	119
Figura 50 Comparație între unghiul yaw în grade (stânga) și translația longitudinală în metrii(dreapta) pentru primele 100 de imagini de pe traectoria din Figura 49....	120
Figura 51 Estimarea ughiului yaw exprima în grade din 3 în 3 imagini pentru o secvență de 300 din 900 imagini succesive.....	120
Figura 52 Deviația standard (axa verticală) a estimării relativ la estimarea originală (neafectată de zgromot de asociere). Deviația standard a zgromotului introdus în asocierile spațiale (disparitate) este pe axa orizontală. (A) - deviația standard a unghiului yaw (exprimată în grade). (B) - deviația standard a translației în lungul axei longitudinale (exprimată în metrii).....	121
Figura 53 Deviația standard (axa verticală) a estimării relativ la estimarea originală (neafectată de zgromot de asociere). Deviația standard a zgromotului introdus în asocierile temporale (flux optic) este pe axa orizontală. (A) - deviația standard a unghiului yaw (exprimată în grade). (B) - deviația standard a translației în lungul axei longitudinale (exprimată în metrii).....	122
Figura 54 Unghiul dintre perechi de trăsături pentru imagini succesive.....	123
Figura 55 Primele 50 de perechile de trăsături pentru care valoarea lui $ \theta_{i,j}^1 - \theta_{i,j}^1 $ este cea mai mică (stânga) respectiv cea mai mare (dreapta). Punctele galbene sunt cele luate în considerare, punctele roșii fiind eliminate pe baza metodelor descrise în 2.1.....	124
Figura 56 Estimarea unghiului yaw (reprezentat în grade pe axa verticală) pentru un set de 1000 de imagini succesive. Cu roșu este reprezentată o valoare de referință obținută de pe magistrala CAN a mașinii de test, cu albastru este estimarea obținută cu	

o metodă stereo iar cu verde este estimarea obținută cu metoda mono descrisă în capitolul curent. Figura din dreapta prezintă o variantă mărită a figurii din stânga..	124
Figura 57 Trăsăturile de la distanță marcate cu verde sunt folosite pentru estimarea rotației.....	125
Figura 58 Estimarea unghiului pitch (stânga) la trecerea peste limitatorul de viteză din imaginea dreaptă. Cu albastru este estimarea obținută cu o metodă stereo iar cu verde este estimarea obținută cu metoda mono descrisă în capitolul curent.....	126
Figura 59 Estimarea unghiului yaw (în grade) pentru diverse praguri de adâncime maximă a trăsăturilor. Culoarea roșie indică valoarea unghiului yaw citit de la magistrala CAN a mașinii. Fiecare culoare corespunde unei valori din Tabel 4.....	126
Figura 60 Exemple de scenarii dificile pentru odometria vizuală.....	127
Figura 61 Transformata rank (stânga) pentru o imagine de trafic (dreapta).....	128
Figura 62 Comparație între traiectoria generată pe baza odometriei vizuale obținute din asocieri de trăsături obținute prin metode diferențiale (albastru în dreapta) și asocieri pe baza de rank transform (roșu în dreapta). Graficul din mijloc este echivalentul liniei roșii din imaginea dreapta. Cu verde este reprezentată o linie poligonală care unește coordonatele GPS obținute pe ruta A-B-C-D, și făcând mai multe rotații în sensul giratoriu B.....	129
Figura 64 Partea de sus arată evoluția unghiului yaw la intrarea sub un pod. Imaginele În rândul de jos sunt două imagini succesive având condiții de iluminare diferite datorită trecerii de la soare la umbră. Cu roșu este reprezentat unghiul citit de la magistrala CAN a mașinii, cu verde este reprezentată estimarea din metode de asociere diferențiale, cu verde este estimarea din asocieri de trăsături pe bază de rank transform.....	129
Figura 63 Traiectoria obținută prin integrarea odometriei vizuale obținute pe traseul A-B-C-D din asociere de trăsături pe bază de rank transform. Traseul prezintă zone cu probleme de iluminare precum cele din dreapta.....	130
Figura 65 Stânga - translația estimată în lungul axei longitudinale în metrii (axa verticală) pentru 50 de imagini consecutive (axa orizontală) cu diverse praguri de distanță pentru filtrarea trăsăturilor. Dreapta - deviația standard pentru translația longitudinală exprimată în metrii (axa verticală) pentru cele 50 de imagini consecutive din stânga (axa orizontală) și pentru diverse praguri de distanță.....	131
Figura 66 Stânga - unghiul yaw (verde, în grade) calculat pentru aceleași imagini ca și Figura 65 și unghiul yaw citit de la mașină (roșu). Dreapta - deviația standard a unghiului yaw (vertical, exprimată în grade). Cu 1 și 2 sunt marcate imaginile unde avem valori mari ale deviației standard a unghiului yaw datorită schimbărilor brusete de iluminare. Punctul 1 corespunde celor două imagini din stânga în Figura 67 iar punctul 2 corespunde celor două imagini din dreapta ale aceleiași figuri.....	131
Figura 67 Perechi de imagini consecutive unde avem schimbări de condiții de iluminare. Cu verde sunt colorate trăsăturile folosite pentru estimarea odometriei vizuale. C galben sunt desenare trăsăturile eliminate în procesul de RANSAC. Vectorii de flux optic reprezentați cu roșu indică corespondențele temporale.....	132
Figura 68 Erorile de estimare ale unghiurilor pitch (α , rândul de sus) și yaw (β , rândul de jos). Coloana stângă prezintă rezultatele după o singură estimare pe baza metodei	

liniare (pasul 2 al algoritmului). Coloana dreaptă prezintă rezultatele după reestimarea efectuată la pasul 4 al algoritmului.....	136
Figura 69 Overview al metodelor pe bază de stereo (sus) comparativ cu metoda pe bază de LiDAR și o cameră monoculară (jos) [11].....	139
Figura 70 Puncte 3D provenite de la un senzor LiDAR, transformate în sistemul camerei, și proiectate în planul imagine.....	141
Figura 71 Triangular Delaunay construită pe baza punctelor LiDAR proiectate în imagine (sus) și o versiune mărită (jos) pentru o mai bună vizualizare a triangulării	142
Figura 72 Filtrare adaptivă a triunghiurilor în funcție de distanța $minD$ a celui mai apropiat vârf. Distanța maxim acceptată pentru celelalte vârfuri este L.....	143
Figura 73 Evaluare calitativă între (A) metoda stereo (stânga) și (C) metoda bazată pe distanțele trăsăturilor estimate din LiDAR (dreapta) pe seventele KITTI 1 (sus) și 2 (jos). Cu albastru este estimarea noastră iar cu roșu estimarea de referință.....	145
Figura 74 Diagrama bloc a sistemului de corecție a odometriei vizuale [17]	150
Figura 75 Clasificarea punctelor LiDAR în puncte planare (roșu) și puncte de muchie (verde).....	152
Figura 76 Exemple de traекторii corectate (sevena KITTI 06 în stânga și sevena 07 în dreapta). Cu verde este reprezentată traectoria de referință (în foarte mare parte acoperită de traectorii estimate și corectate), cu albastru este traectoria corectată prin metodele propuse în [17], cu roșu este traectoria generată cu o metodă care face estimarea strict din imagini succesive.....	158
Figura 77 Trăsăturile cu verde reprezintă setul maximal de trăsături pe baza cărora se face estimarea mișcării relative dintre camera și background (stânga) respectiv dintre cameră și un obiect aflat în mișcare (dreapta).....	161
Figura 78 Trăsăturile folosite pentru a calcula mișcarea generală a camerei (verde) și cele rezultate pentru calculul mișcării obiectelor din scenă.....	163
Figura 79 Reprezentare de sus a scenelor din Figura 78. Observăm vitezele de deplasare estimate pentru obiectele dinamice, precum și orientarea vectorului de viteză al acestor obiecte (albastru).....	163
Figura 80 Estimarea mișcării relative a unei mașini care intră în scenă (primele două imagini din stânga).....	164
Figura 81 Folosirea de super-pixeli și metode de odometrie vizuala pentru percepția scenelor dinamice [14].....	165
Figura 82 Transfer folosind un plan.....	205
Figura 83 Tensorul trifocal, imagine preluată din [2].....	208

LISTA TABELELOR

Tabel 1 Modalități de reprezentare a rotației și translației dintre două sisteme de coordonate.....	56
Tabel 2 Derivatele parțiale $\partial a_j(p_i)/\partial p_{i,l}$	82
Tabel 3 Derivatele parțiale $\partial b_j(p_i)/\partial p_{i,l}S$	83
Tabel 4 Erorile de estimare a unghiului yaw relativ la valoarea citită de pe magistrala CAN a mașinii. Estimarea se face considerând doar trăsăturile aflate la o distanță mai mică decât pragul indicat pe prima coloană a tabelului. Culoarea indicată pe ultima coloană este folosită pentru graficul din Figura 59.....	127
Tabel 5 Evaluarea ordinului de magnitudine al erorilor pentru o translație longitudinală $T_{ref} = [0 \ 0 \ -1]$ și pentru unghiiurile de rotație variind în intervalul indicat pe fiecare rând al tabelului.....	137
Tabel 6 Evaluarea ordinului de magnitudine al erorilor pentru o translație longitudinală $T_{ref} = [0 \ 0 \ -1]$ și pentru unghiiurile de rotație variind în intervalul indicat pe fiecare rând al tabelului, și adăugând zgromot Gaussian trăsăturilor generate sintetic.....	137
Tabel 7 Erorile de translație și rotație pentru configurațiile A (stereo), B(LiDAR combinat cu stereo) și C (LiDAR).....	144
Tabel 8 Erorile de estimare a odometriei la introducerea de erori de decalibrare pe direcția longitudinală a translației [11].....	146
Tabel 9 Erorile de estimare a odometriei la introducerea de erori de decalibrare dintre cameră și dispozitivul LiDAR pe unghiul yaw (exprimate în pixeli) [11].....	146
Tabel 10 Erorile de estimare a odometriei la introducerea de erori de decalibrare dintre cameră și dispozitivul LiDAR pe unghiul yaw (exprimate în pixeli) [11].....	146
Tabel 11 Evaluarea performanței de corecție a odometriei vizuale pe bază de LiDAR. Liniile tabelului indică diferențe configurații de parametrii pe care-i corectăm.....	157
Tabel 12 Evaluarea performanței algoritmului de corecție pe setul de date KITTI [72].....	157
Tabel 13 Evaluarea efectului pe care îl are pragul de distanță între imagini cheie asupra performanței de corecție a traiectoriei pe cele 10 secvențe KITTI.....	159
Tabel 14 Evaluarea unor opțiuni alternative de reprezentare a planelor [17].....	159

ANEXE

(*obligatoriu pe pagină impară*)

Anexa 1 - Modelul matematic de estimare a covarianței unei măsurători rezultate din minimizarea unei funcții

Preluăm conținutul acestei anexe din [13] pentru a servi ca și referință pentru dezvoltarea modelului matematic din Capitolul 1.5

We measure a column vector p perturbed by some random noise Δp . The actual measured value is thus $\bar{p} = p + \Delta p$. Through this set of measurements we need to fit a model having parameters vector θ , θ being also a column vector. The random perturbation vector Δp that affects the measurements will cause a perturbation vector $\Delta\theta$ in the model parameter vector. The model fitting involves finding the parameters vector θ that minimize the function $F(p, \theta)$. However because the vector p is measured with noise, we actually compute a vector $\bar{\theta} = \theta + \Delta\theta$ that minimizes $F(p + \Delta p, \theta + \Delta\theta)$. Because θ minimizes the function $F(p, \theta)$ the partial derivatives of F with respect to each entry of the vector θ will be 0 at (p, θ) :

$$g(p, \theta) = \frac{\partial F(p, \theta)}{\partial \theta} = 0$$

The function $g(p, \theta)$ is represented as a column vector having the same size as the parameters vector. Similarly, because $\theta + \Delta\theta$ minimizes the function $F(p + \Delta p, \theta + \Delta\theta)$, we have $g(p + \Delta p, \theta + \Delta\theta) = 0$. By doing a first order Taylor expansion of $g(p, \theta)$ around $g(p + \Delta p, \theta + \Delta\theta)$, we obtain:

$$g(p, \theta) \approx g(p + \Delta p, \theta + \Delta\theta) - \frac{\partial g(p + \Delta p, \theta + \Delta\theta)}{\partial p} \cdot \Delta p - \frac{\partial g(p + \Delta p, \theta + \Delta\theta)}{\partial \theta} \cdot \Delta\theta$$

This means that:

$$0 \approx \frac{\partial g(p + \Delta p, \theta + \Delta\theta)}{\partial p} \cdot \Delta p + \frac{\partial g(p + \Delta p, \theta + \Delta\theta)}{\partial \theta} \cdot \Delta\theta$$

To compute $\frac{\partial g(p, \theta)}{\partial p}$, the partial derivatives of a given element of $g(p, \theta)$ with respect to the entries of p are computed and stored in the corresponding row of the matrix $\frac{\partial g(p, \theta)}{\partial p}$. Then is a positive definite symmetric matrix

because:

$$\frac{\partial g(p, \theta)}{\partial \theta} = \frac{\partial F^2(p, \theta)}{\partial^2 \theta}$$

Using its inverse we obtain:

$$\Delta\theta \approx - \left(\frac{\partial g(p+\Delta p, \theta+\Delta\theta)}{\partial \theta} \right)^{-1} \cdot \frac{\partial g(p+\Delta p, \theta+\Delta\theta)}{\partial p} \cdot \Delta p = - \left(\frac{\partial g(\bar{p}, \bar{\theta})}{\partial \theta} \right)^{-1} \cdot \frac{\partial g(\bar{p}, \bar{\theta})}{\partial p} \cdot \Delta p$$

This indicates how a the measurement error Δp is reflected in the parameters estimation error $\Delta\theta$. Treating $\Delta\theta$ and Δp as random variables we infer the following relation:

$$\begin{aligned} \Sigma_{\Delta\theta} &= E[\Delta\theta \cdot \Delta\theta^T] = \\ &E \left[\left(\frac{\partial g(\bar{p}, \bar{\theta})}{\partial \theta} \right)^{-1} \cdot \frac{\partial g(\bar{p}, \bar{\theta})}{\partial p} \cdot \Delta p \cdot \Delta p^T \cdot \left(\frac{\partial g(\bar{p}, \bar{\theta})}{\partial p} \right)^T \cdot \left(\frac{\partial g(\bar{p}, \bar{\theta})}{\partial \theta} \right)^{-1} \right] \\ &= \left(\frac{\partial g(\bar{p}, \bar{\theta})}{\partial \theta} \right)^{-1} \cdot \frac{\partial g(\bar{p}, \bar{\theta})}{\partial p} \cdot E[\Delta p \cdot \Delta p^T] \cdot \left(\frac{\partial g(\bar{p}, \bar{\theta})}{\partial p} \right)^T \cdot \left(\frac{\partial g(\bar{p}, \bar{\theta})}{\partial \theta} \right)^{-1} \\ &= \left(\frac{\partial F^2(\bar{p}, \bar{\theta})}{\partial^2 \theta} \right)^{-1} \cdot \frac{\partial F^2(\bar{p}, \bar{\theta})}{\partial \theta \partial p} \cdot \Sigma_{\Delta p} \cdot \left(\frac{\partial F^2(\bar{p}, \bar{\theta})}{\partial \theta \partial p} \right)^T \cdot \left(\frac{\partial F^2(\bar{p}, \bar{\theta})}{\partial^2 \theta} \right)^{-1} \end{aligned}$$

Anexa 2 - Demonstrația matematică pe care se bazează detecția colturilor Harris

$$\begin{aligned} E(\Delta u, \Delta v) &\approx \sum_{r=u-s_r}^{r=u+s_r} \sum_{c=v-s_c}^{c=v+s_c} (I_u(r, c) \cdot \Delta u + I_v(r, c) \cdot \Delta v)^2 = \\ &= \sum_{r=u-s_r}^{r=u+s_r} \sum_{c=v-s_c}^{c=v+s_c} (I_u^2(r, c) \cdot \Delta u^2 + I_v^2(r, c) \cdot \Delta v^2 + 2 \cdot I_u(r, c) \cdot I_v(r, c) \cdot \Delta u \cdot \Delta v) \end{aligned}$$

$$= \Delta u^2 \sum_{r=u-s_r}^{r=u+s_r} \sum_{c=v-s_c}^{c=v+s_c} I_u'^2(r, c) + 2 \cdot \Delta u \cdot \Delta v \sum_{r=u-s_r}^{r=u+s_r} \sum_{c=v-s_c}^{c=v+s_c} I_u'(r, c) \cdot I_v'(r, c) \\ + \Delta v^2 \sum_{r=u-s_r}^{r=u+s_r} \sum_{c=v-s_c}^{c=v+s_c} I_v'^2(r, c)$$

Considerând o forma biliniară $B(x, y) = x^2 \cdot A + 2 \cdot x \cdot y \cdot C + y^2 \cdot B$, se poate verifica prin înmulțire directă de matrici că aceasta se poate scrie sub formă matriceală astfel:

$$B(x, y) = [x \ y] \cdot \begin{bmatrix} A & C \\ C & B \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Folosind acest rezultat deducem următoarea expresie pentru $E(\Delta u, \Delta v)$:

$$E(\Delta u, \Delta v) \approx [\Delta u \ \Delta v] \cdot \begin{bmatrix} \sum_{r=u-s_r}^{r=u+s_r} \sum_{c=v-s_c}^{c=v+s_c} I_u'^2(r, c) & \sum_{r=u-s_r}^{r=u+s_r} \sum_{c=v-s_c}^{c=v+s_c} I_u'(r, c) \cdot I_v'(r, c) \\ \sum_{r=u-s_r}^{r=u+s_r} \sum_{c=v-s_c}^{c=v+s_c} I_u'(r, c) \cdot I_v'(r, c) & \sum_{r=u-s_r}^{r=u+s_r} \sum_{c=v-s_c}^{c=v+s_c} I_v'^2(r, c) \end{bmatrix} \cdot \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}$$

Anexa 3 - Invarianta ZNCC la condiții de iluminare

Avem:

$$\bar{I}(u^i, v^i) = \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)} \cdot \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} I(u^i + \Delta r, v^i + \Delta c)$$

$$\bar{J}(u^j, v^j) = \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)} \cdot \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} J(u^j + \Delta r, v^j + \Delta c)$$

$$ZNCC(u^i, v^i, u^j, v^j) = \frac{\sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} (I(u^i + \Delta r, v^i + \Delta c) - \bar{I}(u^i, v^i)) \cdot (J(u^j + \Delta r, v^j + \Delta c) - \bar{J}(u^j, v^j))}{\sqrt{\sum_{\Delta r=-s_r}^{s_r} (I(u^i + \Delta r, v^i + \Delta c) - \bar{I}(u^i, v^i))^2} \cdot \sqrt{\sum_{\Delta c=-s_c}^{s_c} (J(u^j + \Delta r, v^j + \Delta c) - \bar{J}(u^j, v^j))^2}}$$

Pentru $I'(u, v) = a \cdot I(u, v) + b$ valoarea medie devine:

$$\begin{aligned}
\bar{I}(u^i, v^i) &= \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)} \cdot \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} I'(u^i + \Delta r, v^i + \Delta c) = \\
\bar{I}(u^i, v^i) &= \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)} \cdot \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} (a \cdot I(u^i + \Delta r, v^i + \Delta c) + b) \\
&= \\
&= \frac{1}{(2 \cdot s_r + 1) \cdot (2 \cdot s_c + 1)} \\
&\quad \cdot \left(a \cdot \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} I(u^i + \Delta r, v^i + \Delta c) + \sum_{\Delta r=-s_r}^{s_r} \sum_{\Delta c=-s_c}^{s_c} b \right) = \\
&= a \cdot \bar{I}(u^i, v^i) + b
\end{aligned}$$

Analizând valoarea $I'(u^i + \Delta r, v^i + \Delta c) - \bar{I}(u^i, v^i)$ care apare atât la numărător cat și la numitor în expresia ZNCC:

$$\begin{aligned}
I'(u^i + \Delta r, v^i + \Delta c) - \bar{I}(u^i, v^i) &= a \cdot I(u^i + \Delta r, v^i + \Delta c) + b - a \cdot \bar{I}(u^i, v^i) - b \\
&= a \cdot (I(u^i + \Delta r, v^i + \Delta c) - \bar{I}(u^i, v^i))
\end{aligned}$$

Valoarea lui a se poate da factor comun atât în suma de produse la numărător cat și în suma de sub radical de la numitor, urmând să se simplifice.

Anexa 4 - Interpolarea intensităților la nivel de subpixel

Considerând o imagine I se pune problema estimării intensităților $I(u, v)$ din imagine pentru valori ale indicilor u și v care nu sunt numere întregi. Notând cu u_0 și cu v_0 valorile partilor integri ale numerelor u și v , și cu $u_1 = u_0 + 1$ și $v_1 = v_0 + 1$, dorim să exprimăm $I(u, v)$ ca și o medie ponderată a valorilor $I(u_0, v_0)$, $I(u_1, v_0)$, $I(u_0, v_1)$, $I(u_1, v_1)$. Fiecare intensitate va fi ponderată cu aria dreptunghiului opus astfel:

$$I(u, v) = (u_1 - u) \cdot (v_1 - v) \cdot I(u_0, v_0) + (u - u_0) \cdot (v_1 - v) \cdot I(u_1, v_0) + \\ (u_1 - u) \cdot (v - v_0) \cdot I(u_0, v_1) + (u - u_0) \cdot (v - v_0) \cdot I(u_1, v_1)$$

Alternativ problema se poate formula prin încercarea de a determina funcția:

$$I(u, v) = a + b \cdot u + c \cdot v + d \cdot u \cdot v$$

Cunoscând $I(u_0, v_0), I(u_1, v_0), I(u_0, v_1), I(u_1, v_1)$ putem formula următorul sistem de ecuații:

$$\begin{bmatrix} 1 & u_0 & v_0 & u_0 \cdot v_0 \\ 1 & u_1 & v_0 & u_1 \cdot v_0 \\ 1 & u_0 & v_1 & u_0 \cdot v_1 \\ 1 & u_1 & v_1 & u_1 \cdot v_1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} I(u_0, v_0) \\ I(u_1, v_0) \\ I(u_0, v_1) \\ I(u_1, v_1) \end{bmatrix}$$

Pe baza coeficienților a,b,c,d se vor aproxima valorile imaginii în orice puncte reale intermediare:

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 & u_0 & v_0 & u_0 \cdot v_0 \\ 1 & u_1 & v_0 & u_1 \cdot v_0 \\ 1 & u_0 & v_1 & u_0 \cdot v_1 \\ 1 & u_1 & v_1 & u_1 \cdot v_1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} I(u_0, v_0) \\ I(u_1, v_0) \\ I(u_0, v_1) \\ I(u_1, v_1) \end{bmatrix}$$

$$I(u, v) = [1 \ u \ v \ u \cdot v] \cdot \begin{bmatrix} 1 & u_0 & v_0 & u_0 \cdot v_0 \\ 1 & u_1 & v_0 & u_1 \cdot v_0 \\ 1 & u_0 & v_1 & u_0 \cdot v_1 \\ 1 & u_1 & v_1 & u_1 \cdot v_1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} I(u_0, v_0) \\ I(u_1, v_0) \\ I(u_0, v_1) \\ I(u_1, v_1) \end{bmatrix}$$

Prin calcule se va ajunge la aceeași formula ca și cea rezultată din ponderarea cu suprafete.

Anexa 5 - Demonstrația matematică a algoritmului Lukas-Kanade

$$\frac{\partial E_{u,v}(\Delta u, \Delta v)}{\partial \Delta u} = 2 \cdot \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} J'_u(r + \Delta u_0, c + \Delta v_0) \\ \cdot (J'_u(r + \Delta u_0, c + \Delta v_0) \cdot \Delta u + J'_v(r + \Delta u_0, c + \Delta v_0) \cdot \Delta v + J(r + \Delta u_0, c + \Delta v_0) \\ - I(r, c)))$$

$$= 2 \cdot \Delta u \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u^2(r, c) + 2 \cdot \Delta v \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u(r, c) \cdot J_v(r, c) + 2 \\ \cdot \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u(r, c) \cdot (J(r, c) - I(r - \Delta v_0, c - \Delta v_0)) = 0$$

$$\frac{\partial E_{u,v}(\Delta u, \Delta v)}{\partial \Delta u} = 2 \cdot \sum_{r=u-s_r}^{u+s_r} \sum_{c=v-s_c}^{v+s_c} J_v(r + \Delta u_0, c + \Delta v_0) \\ \cdot (J_u(r + \Delta u_0, c + \Delta v_0) \cdot \Delta u + J_v(r + \Delta u_0, c + \Delta v_0) \cdot \Delta v + J(r + \Delta u_0, c + \Delta v_0) \\ - I(r, c))) \\ = 2 \cdot \Delta u \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u(r, c) \cdot J_v(r, c) + 2 \cdot \Delta v \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_v^2(r, c) + 2 \\ \cdot \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_v(r, c) \cdot (J(r, c) - I(r - \Delta u_0, c - \Delta v_0)) = 0$$

Egalând cu zero cele două derivate parțiale în raport cu Δu și Δv putem scrie cele două ecuații în forma matriceală astfel:

$$\begin{bmatrix} \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u^2(r, c) & \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u(r, c) \cdot J_v(r, c) \\ \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u(r, c) \cdot J_v(r, c) & \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_v^2(r, c) \end{bmatrix} \cdot \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} \\ = \begin{bmatrix} - \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u(r, c) \cdot (J(r, c) - I(r - \Delta u_0, c - \Delta v_0)) \\ - \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_v(r, c) \cdot (J(r, c) - I(r - \Delta u_0, c - \Delta v_0)) \end{bmatrix}$$

Astfel:

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \left[\begin{array}{cc} \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u^{'2}(r, c) & \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u^{'}(r, c) \cdot J_v^{'}(r, c) \\ \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u^{'}(r, c) \cdot J_v^{'}(r, c) & \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_v^{'2}(r, c) \\ - \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_u^{'}(r, c) \cdot (J(r, c) - I(r - \Delta u_0, c - \Delta v_0)) \\ - \sum_{r=u-s_r+\Delta u_0}^{u+s_r+\Delta u_0} \sum_{c=v-s_c+\Delta v_0}^{v+s_c+\Delta v_0} J_v^{'}(r, c) \cdot (J(r, c) - I(r - \Delta u_0, c - \Delta v_0)) \end{array} \right]^{-1}$$

Anexa 6 - Demonstrația matematică a liniarizării matricii de rotație

O matrice de rotație poate fi descompusă într-un produs de trei matrici de rotație în jurul axelor X, Y și respectiv Z:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \cdot \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ = \begin{bmatrix} \cos(\beta)\cos(\gamma) & -\cos(\beta)\sin(\gamma) & -\sin(\beta) \\ -\sin(\alpha)\sin(\beta)\cos(\gamma) + \cos(\alpha)\sin(\gamma) & \sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & -\sin(\alpha)\cos(\beta) \\ \cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\sin(\gamma) & -\cos(\alpha)\sin(\beta)\sin(\gamma) + \sin(\alpha)\cos(\gamma) & \cos(\alpha)\cos(\beta) \end{bmatrix}$$

De asemenea, funcțiile sin și cos pot fi descompuse în serie Taylor pentru fiecare dintre unghiurile α, β, γ astfel:

$$\sin(\alpha) = \alpha - \frac{\alpha^3}{3!} + \frac{\alpha^5}{5!} - \frac{\alpha^7}{7!} + \dots$$

$$\cos(\alpha) = 1 - \frac{\alpha^2}{2!} + \frac{\alpha^4}{4!} - \frac{\alpha^6}{6!} + \dots$$

Folosind primul termen din aproximarea Taylor a acestora, o aproximare a lui R devine:

$$R \approx \begin{bmatrix} 1 & -\gamma & -\beta \\ -\alpha\beta + \gamma & \alpha\beta\gamma + 1 & -\alpha \\ \beta + \alpha\gamma & -\beta\gamma + \alpha & 1 \end{bmatrix}$$

Produsul a două sau mai multe dintre valorile unghiurilor α, β, γ exprimate în radiani poate fi aproimat la zero deoarece valoarea acestora în radiani este mică, rezultând:

$$R \approx \begin{bmatrix} 1 & -\gamma & -\beta \\ \gamma & 1 & -\alpha \\ \beta & \alpha & 1 \end{bmatrix}$$

Anexa 8 - Corecția erorilor de odometrie pe bază de patru puncte coplanare

Preluăm conținutul acestei anexe din [17] pentru a servi ca și referință a modelului matematic din capitolul 4.

We start from equation:

$$\begin{vmatrix} X^i - \gamma \cdot Y^i - \beta \cdot Z^i + t_x - X^k & X^j - X^k & X^l - X^k \\ \gamma \cdot X^i + Y^i - \alpha \cdot Z^i + t_y - Y^k & Y^j - Y^k & Y^l - Y^k \\ \beta \cdot X^i + \alpha \cdot Y^i + Z^i + t_z - Z^k & Z^j - Z^k & Z^l - Z^k \end{vmatrix} = 0$$

We define the following auxiliary terms:

$$\begin{aligned} A &= (Y^j - Y^k) \cdot (Z^l - Z^k) - (Y^l - Y^k) \cdot (Z^j - Z^k) \\ B &= -(X^j - X^k) \cdot (Z^l - Z^k) + (X^l - X^k) \cdot (Z^j - Z^k) \\ C &= (X^j - X^k) \cdot (Y^l - Y^k) - (X^l - X^k) \cdot (Y^j - Y^k) \end{aligned}$$

By using those terms we can further write the determinant as:

$$\begin{aligned} A \cdot (X^i - \gamma \cdot Y^i - \beta \cdot Z^i + t_x - X^k) + \\ + B \cdot (\gamma \cdot X^i + Y^i - \alpha \cdot Z^i + t_y - Y^k) \\ + C \cdot (\beta \cdot X^i + \alpha \cdot Y^i + Z^i + t_z - Z^k) = 0 \end{aligned}$$

Thus the linear equation can be written as:

$$\alpha \cdot (C \cdot Y^i - B \cdot Z^i) + \beta \cdot (C \cdot X^i - A \cdot Z^i) + \gamma \cdot (B \cdot X^i - A \cdot Y^i) +$$

$$+ t_x \cdot A + t_y \cdot B + t_z \cdot C = \\ = - A \cdot (X^i - X^k) - B \cdot (Y^i - Y^k) - C \cdot (Z^i - Z^k)$$

Anexa 9 - Corecția erorilor de odometrie pe bază de coplanaritate cu un plan dat

Preluăm conținutul acestei anexe din [17] pentru a servi ca și referință a modelului matematic din capitolul 4.

Considering the plane equation:

$$A \cdot X + B \cdot Y + C \cdot Z = -D$$

by replacing the rotated and translated point we get:

$$A \cdot (X^i - \gamma \cdot Y^i - \beta \cdot Z^i + t_x) + \\ B \cdot (\gamma \cdot X^i + Y^i - \alpha \cdot Z^i + t_y) + \\ + C \cdot (\beta \cdot X^i + \alpha \cdot Y^i + Z^i + t_z) = -D$$

Thus the linear equation can be written as:

$$\alpha \cdot (C \cdot Y^i - B \cdot Z^i) + \beta \cdot (C \cdot X^i - A \cdot Z^i) + \gamma \cdot (B \cdot X^i - A \cdot Y^i) + \\ + t_x \cdot A + t_y \cdot B + t_z \cdot C = \\ = -A \cdot X^i - B \cdot Y^i - C \cdot Z^i - D$$

Anexa 10 - Transformări între imagini induse de un plan

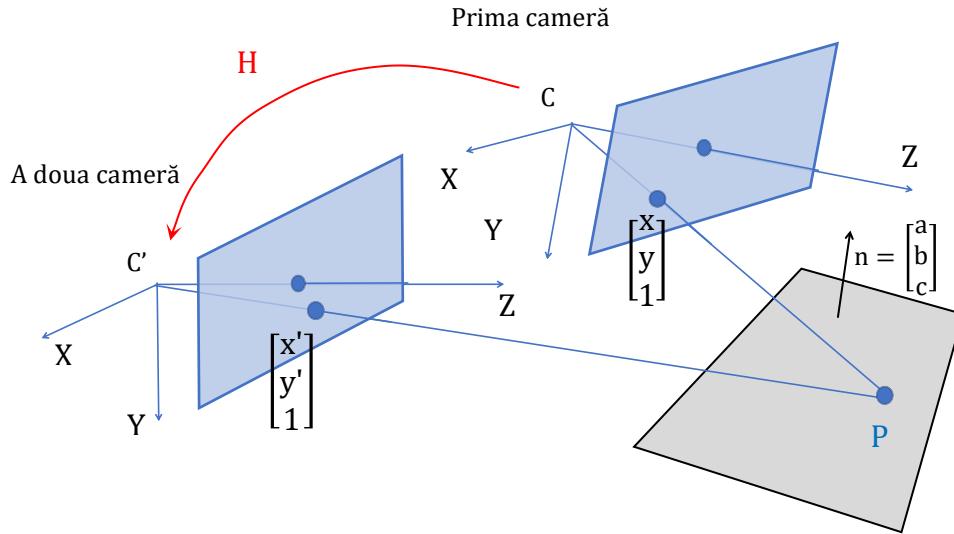


Figura 82 Transfer folosind un plan

Având un plan în sistemul de coordonate al lumii se poate demonstra că acesta induce o transformată liniară (lucrând în coordonate omogene) dintr-o imagine în celalătă: $p' \approx H \cdot p$ unde $p' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ și $p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ iar H este o matrice inversabilă de dimensiune 3×3 .

Aceasta matrice se numește matrice de homografie. Fie $n = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ și normala planului și $n^T \cdot P = 1$ ecuația planului exprimată într-un sistem de coordonate de referință (Figura 82). Pentru simplitate considerăm că sistemul de referință este aliniat cu sistemul primei camere și R, T sunt matricea de rotație și vectorul de translație folosite pentru a transforma din sistemul primei camere în sistemul celei de-a două:

$$\begin{aligned} p \approx I \cdot P + 0 = P &\Rightarrow p = k \cdot P \Rightarrow n^T \cdot p = k \cdot n^T \cdot P \Rightarrow \\ &\Rightarrow n^T \cdot p = k \text{ și } P = \frac{p}{n^T \cdot p} \end{aligned}$$

$$\begin{aligned}
p' &\approx R \cdot P + T = R \cdot \frac{p}{n^T \cdot p} + T \Rightarrow p' \approx R \cdot p + T \cdot n^T \cdot p \Rightarrow \\
&\Rightarrow H = R + T \cdot n^T
\end{aligned} \tag{1}$$

Dacă punctele în planele celor două imagini sunt exprimate în pixeli atunci:

$$\begin{aligned}
q = K \cdot p \text{ și } q' = K' \cdot p' \Rightarrow K'^{-1} q' &\approx H \cdot K^{-1} \cdot q \Rightarrow \\
\Rightarrow q' &\approx K' \cdot H \cdot K^{-1} \cdot q
\end{aligned} \tag{2}$$

Matricea $K' \cdot H \cdot K^{-1}$ este de asemenea o matrice inversabilă.

Un caz particular este atunci când planul prin intermediul căruia se face transferul este planul de la infinit. Ecuația planului în coordonate omogene este:

$$[a \ b \ c \ 1] \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0 \tag{3}$$

Punctele de la infinit sunt reprezentate în coordinate omogene având cea de-a patra coordonată zero, iar planul pe care sunt situate aceste puncte are vectorul normal zero:

$$n = \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

Intuitiv faptul ca punctele de la infinit sunt situate pe planul de la infinit este exprimat astfel:

$$[0 \ 0 \ 0 \ 1] \cdot \begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix} = 0 \tag{5}$$

Înlocuind în relația de mai sus, obținem pentru punctele de la infinit o variantă simplificată a matricii de homografie, care va fi egală cu matricea de rotație:

$$H = R + T \cdot n^T = R + T \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = R \quad (6)$$

Dacă punctele de la infinit proiectate în planele celor două imagini sunt exprimate în pixeli atunci:

$$q' \approx K' \cdot R \cdot K^{-1} \cdot q \quad (7)$$

Observăm astfel că relația dintre proiecțiile punctelor aflate la infinit față de cameră depinde doar de matricea de rotație și nu depinde deloc de matricea de translație dintre pozițiile diferite ale camerei.

O relație similară există pentru linii. Considerăm l' și l două linii corespondente (proiecțiile aceleiași linii 3D) în planele celor două imagini exprimate în coordonate omogene. Fie p' un punct oarecare situat în planul celei de-a doua imaginii pe l' și p punctul corespunzător din planul primei imagini:

$$l'^T p' = 0$$

$$\Rightarrow l'^T \cdot H \cdot p = 0, \forall p', p \text{ puncte corespondente de pe cele două proiecții}$$

$$\Rightarrow l'^T \cdot H = l^T \quad (8)$$

sau echivalent dacă transpunem întreaga relație:

$$l = H^T \cdot l' \quad (9)$$

Anexa 11 - Tensorul trifocal

Tensorul trifocal joacă în cazul a 3 proiecții ale aceleiași scene are același rol pe care matricea fundamentală sau matricea esențială îl joacă în cazul a 2 proiecții [2]. Având 3 proiecții perspectivă ale aceleiași scene 3D dorim să avem relații între coordonatele proiecțiilor acelorași structuri 3D (linii, puncte). De exemplu având o dreapta în 3D aceasta se va proiecta în 3 drepte diferite în cele 3 proiecții (imagini) și dorim să determinăm relația dintre ecuațiile celor 3 drepte proiectate (exemplu în Figura 83). Un alt exemplu util în contextul lucrării de față

este problema transferului enunțată astfel: dându-se un punct în prima imagine și un punct în cea de-a două imagine dorim să determinăm punctul corespunzător în cea de-a treia imagine, acest lucru fiind posibil folosind noțiunile geometrice introduse de tensorul trifocal.

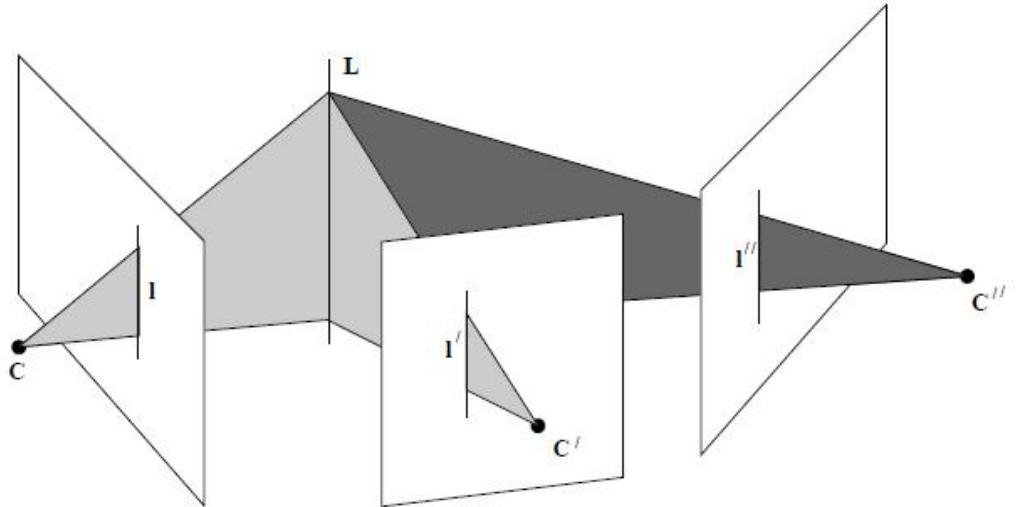


Figura 83 Tensorul trifocal, imagine preluată din [2]

Considerând o dreapta l exprimată în coordonate omogene în planul imagine, punctele p aparținând dreptei vor respecta ecuația:

$$l^T \cdot p = 0 \quad (1)$$

Punctele 3D care se proiectează pe dreapta l sunt situate într-un plan care conține atât dreapta cât și centrul de proiecție al camerei. Pentru un punct 3D P exprimat în coordonate omogene și o matrice de proiecție A vom avea :

$$p \approx A \cdot P \Rightarrow l^T \cdot p = 0 = l^T \cdot A \cdot P \quad (2)$$

Deducem astfel ca $A^T \cdot l$ este ecuația planului 3D care trece prin dreapta l .

Consideram cele 3 proiecții perspectiva din Figura 83 prima proiecție perspectiva având matricea de proiecție $A_1 = [I \ 0]$, a doua proiecție perspectivă având matricea de proiecție $A_2 = [R_2 \ T_2]$, iar a treia $A_3 = [R_3 \ T_3]$. Având o dreapta în 3D, aceasta se va proiecta în l_1 , l_2 respectiv l_3 și va determina câte un plan cu centrul fiecareia dintre ele

trei camere: $\pi_1 = A_1^T \cdot l_1$, $\pi_2 = A_2^T \cdot l_2$ respectiv $\pi_3 = A_3^T \cdot l_3$. Pentru că cele trei plane se intersectează într-o dreaptă, rezultă că sistemul de ecuații are o infinitate de soluții, adică matricea sistemului va avea rangul mai mic decât 3:

$$\begin{cases} \pi_1^T \cdot X = 0 \\ \pi_2^T \cdot X = 0 \\ \pi_3^T \cdot X = 0 \end{cases} \quad (3)$$

Astfel vectorii π_1 , π_2 și π_3 sunt liniar dependenți:

$$\pi_1 = \alpha \cdot \pi_2 + \beta \cdot \pi_3 \quad (4)$$

Ținând cont de forma matricilor A_1 , A_2 , A_3 avem:

$$[\pi_1 \quad \pi_2 \quad \pi_3] = \begin{bmatrix} l_1 & R_2^T \cdot l_2 & R_3^T \cdot l_3 \\ 0 & T_2^T \cdot l_2 & T_3^T \cdot l_3 \end{bmatrix} \quad (5)$$

Din relația de dependință liniara, pe baza ultimului rand din matricea de mai sus rezultă:

$$\begin{aligned} 0 &= \alpha \cdot T_2^T \cdot l_2 + \beta \cdot T_3^T \cdot l_3 \Rightarrow \\ &\Rightarrow \alpha = c \cdot T_3^T \cdot l_3 \text{ și } \beta = -c \cdot T_2^T \cdot l_2 \text{ unde } c \in \mathbb{R}^+ \end{aligned}$$

Din relația de dependință liniara scrisă pentru primele 3 linii ale matricii rezulta:

$$\begin{aligned} l_1 &= \alpha \cdot R_2^T \cdot l_2 + \beta \cdot R_3^T \cdot l_3 = \\ &= c \cdot (T_3^T \cdot l_3) \cdot R_2^T \cdot l_2 - c \cdot (T_2^T \cdot l_2) \cdot R_3^T \cdot l_3 \quad (6) \end{aligned}$$

Lucrând în coordinate omogene, constanta c din relația de mai sus poate fi neglijată. Rescriind relația de mai sus pentru poziția numărul i din l_1 obținem:

$$\begin{aligned} l_1^i &= (T_3^T \cdot l_3) \cdot R_2^{iT} \cdot l_2 - (T_2^T \cdot l_2) \cdot R_3^{iT} \cdot l_3 = \\ &= (l_3^T \cdot T_3) \cdot R_2^{iT} \cdot l_2 - (l_2^T \cdot T_2) \cdot R_3^{iT} \cdot l_3 = \\ &= l_3^T \cdot (T_3 \cdot R_2^{iT}) \cdot l_2 - l_2^T \cdot (T_2 \cdot R_3^{iT}) \cdot l_3 = l_2^T \cdot (R_2^i \cdot T_3^T) \cdot l_3 - l_2^T \cdot \\ &\quad (T_2 \cdot R_3^{iT}) \cdot l_3 = l_2^T \cdot (R_2^i \cdot T_3^T - T_2 \cdot R_3^{iT}) \cdot l_3 \quad (7) \end{aligned}$$

Cele trei matrici de dimensiune 3x3 compun tensorul trifocal:

$$S_i = R_2^i \cdot T_3^T - T_2 \cdot R_3^{i^T} \quad (8)$$

Acestea verifică relațiile:

$$l_1^i = l_2^T \cdot S_i \cdot l_3, \forall i = \overline{1,3} \quad (9)$$

Acum matrici vor avea un rol important în problema transferului: dându-se două proiecții ale aceluiași punct 3D, se poate determina proiecția punctului 3D în cea de-a treia imagine. Pentru mai multe detalii legate de tensorul trifocal referim cititorul spre [2].

LISTĂ DE PUBLICAȚII

(obligatoriu pe pagină impară)

Golban, Cătălin, Ionuț Golban, and Sergiu Nedevschi. "Vision based three-dimensional vehicle motion detection by minimizing nonlinear functions." Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing. IEEE, 2010.

Golban, Cătălin, and Sergiu Nedevschi. "Linear vs. non linear minimization in stereo visual odometry." 2011 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2011.

Istvan, Szakats, Cătălin Golban, and Sergiu Nedevschi. "Fast vision based ego-motion estimation from stereo sequences—A GPU approach." 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2011.

Nedevschi, Sergiu, Cătălin Golban, and Cosmin Mitran. "Improving accuracy for Ego vehicle motion estimation using epipolar geometry." 2009 12th International IEEE Conference on Intelligent Transportation Systems. IEEE, 2009.

Golban, Cătălin, Cosmin Mitran, and Sergiu Nedevschi. "A practical method for ego vehicle motion estimation from video." 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing. IEEE, 2009.

Golban, Cătălin, Szakats Istvan, and Sergiu Nedevschi. "Stereo based visual odometry in difficult traffic scenes." 2012 IEEE Intelligent Vehicles Symposium. IEEE, 2012.

Golban, Catalin, Petrut Cobarzan, and Sergiu Nedevschi. "A Comparison Study on Replacing Stereo Disparity with LiDAR in Visual Odometry Methods." 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2020.

Golban, Catalin, and Sergiu Nedevschi. "An experiment on relative rotation estimation from distant points with monocular vision." 2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2013.

Golban, Catalin, Petrut Cobarzan, and Sergiu Nedevschi. "Direct formulas for stereo-based visual odometry error modeling." 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2015.

Golban, Catalin, and Sergiu Nedevschi. "Speed estimation for scene objects using stereo visual odometry methods." 2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2013.

Golban, Catalin, and Sergiu Nedevschi. "Moving rigid objects segmentation in 3D dynamic traffic scenes using a stereovision system." 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2014.

Golban, Catalin, Petrut Cobarzan, and Sergiu Nedevschi. "On visual odometry estimation using linear methods - a mathematical framework" 2020 Automation,

Computers, Applied Mathematics (ACAM).

Golban, Catalin, Petrut Cobarzan, and Sergiu Nedevschi. "Visual Odometry Drift Reduction Based on LiDAR Point Clouds Alignment" 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2021.