

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 365

**Stabiliziranje video zapisa
pribavljenog iz lebdeće letjelice**

Mateja Čuljak

Zagreb, rujan 2012.

SADRŽAJ

1. Uvod	1
2. Srodni radovi	3
3. Opis metode	5
3.1. Praćenje značajki	6
3.1.1. Odabir značajki	6
3.1.2. Praćenje značajki	8
3.2. Estimacija i primjena homografije	10
3.2.1. Algoritam RANSAC	12
3.2.2. SVD dekompozicija matrice	13
3.2.3. Levenberg-Marquardtova gradijentna optimizacija	14
3.3. Ponovni početak praćenja	16
3.4. Bilinearna interpolacija	16
4. Skup podataka	20
5. Programska implementacija	23
5.1. Parametri algoritma KLT	23
5.2. Implementacija homografije	24
5.3. Prevođenje programskog koda	24
5.4. Pokretanje programa	25
6. Rezultati	26
6.1. Evaluacija sustava bez ponovnog praćenja značajki	26
6.2. Evaluacija sustava s ponovnim praćenjem značajki	27
7. Zaključak	37
Literatura	39

1. Uvod

Sustavi za nadzor i upravljanje prometom ovise o senzorskim informacijama kako bi se procijenili parametri toka prometa. Primjeri senzorskih informacija su gustoća, smjer i frekvencija vozila na važnim prometnim raskrižjima. Današnji sustavi koriste senzore ograničenih mogućnosti. Video nadzorom moguće je dobiti više informacija o prometu, primjerice klasifikaciju vozila, duljine kolona, relativno ubrzanje te razmak između vozila. Video nadzor je moguće ostvariti postavljanjem nepomične video kamere na raskrižja. Međutim, takva video kamera nema vertikalni pogled na raskrižje te se pojavljuju problemi zaklanjanja vozila. Ovaj problem moguće je zaobići uporabom kamere postavljene na helikopter ili bespilotnu letjelicu. Dodatna prednost uporabe letjelice je u mogućnosti premještanja kamere na različita raskrižja te video praćenje određenog skupa vozila, primjerice kolone, kroz više raskrižja kako bi se ustanovila dinamika kretanja vozila. Video nadzor letjelicom s ljudskom posadom predstavlja veliki trošak u usporedbi s bespilotnim letjelicama. Javlja se potreba za potpunom automatizacijom video nadzora uz korištenje bespilotnih letjelica.

U idealnom slučaju kamera na letjelici je posve vertikalna i nepomična u odnosu na točku promatranja (primjerice, središte prometnog raskrižja). Međutim, zbog stalnog kretanja letjelice sa šest stupnjeva slobode te utjecaja vjetra, kamera nije posve vertikalna i statična. Stoga su snimci vrlo nestabilni. Dolazi do transformacije ravnine slike video zapisa. Procjena parametara toka prometa iz tako dobivene nestabilne slike može biti otežana. Potrebno je stabilizirati video kako bi se ravnina svih slika video zapisa poklopila s ravninom početne (referentne) slike video zapisa.

Bespilotne lebdeće letjelice za nadzor prometa su često ograničene malom težinom nosivog tereta. Zato je stabilizacija kamere hardverskim postupcima, primjerice žiroskopima, često onemogućena. Stoga softverski postupci stabilizacije video zapisa dobivaju na značaju.

Fokus ovog rada je na stabilizaciji video zapisa. Na raspolaganju je skup video

zapisa pribavljenih iz letjelice koji prikazuju snimak cestovnog raskrižja iz ptičje perspektive. Letjelica se nalazi na visini od oko 100 m.

U 2. poglavlju ukratko su opisani srodni radovi koji se bave stabilizacijom video zapisa. U 3. poglavlju naveden je opis korištenih metoda računalnog vida u izgrađenom sustavu stabilizacije video zapisa. U 4. poglavlju opisan je skup podataka nad kojima je sustav testiran. Programska implementacija opisana je u 5. poglavlju. U 6. poglavlju dan je pregled i komentar rezultata. Pregled mogućnosti za budući rad na sustavu te zaključak dani su u 7. poglavlju.

2. Srodni radovi

Sim i Park [17] opisuju pristup stabilizaciji videa koji se oslanja na informacije o uzvišenju iz slika video zapisa u obliku mape uzvišenja (engl. *digital elevation map*, DEM). Dobivena mapa uzvišenja se nakon toga podudara s referentnom mapom uzvišenja dobivenom iz baze podataka. Ovaj pristup zanemaruje problem promjenjivih svjetlosnih uvjeta zbog različitih točaka gledišta referentne slike i ostalih slika video zapisa, te problem promjena vremenskih uvjeta koje su očekivane ako se referentne slike i nove video sekvence ne pribavljaju u isto vrijeme, što dovodi do razlike u osvijetljenju slika. Također, ovaj pristup je ovisan o prethodno pribavljenoj informaciji (mapi uzvišenja) te njenoj točnosti. Drugi pristup koristi telemetrijske podatke koji sadrže koordinate stvarnog svijeta, zajedno s kutevima gibanja bespilotne letjelice. Sinkronizacijom tih podataka s ulaznom video sekvencom dobiva se gruba stabilizacija.

Kombinacija telemetrijskih podataka za grubu stabilizaciju, podudaranja korištenjem vizualnih značajki, te mape uzvišenja za precizno podešavanje stabilizacije predložena je u [9].

Shekhar [15] opisuje odabir vizualnih značajki na referentnoj slici pomoću ljudskog operatera, te stabiliziranje videa praćenjem značajki. Ovaj pristup nije potpuno automatiziran, što doprinosi povećanju troška i vremena obavljanja zadatka stabilizacije.

Nedostaci navedenih pristupa su velika ovisnost o senzorskim podacima (precizni navigacijski sustavi za telemetrijske podatke su skupi), prikaz ili statične scene ili statične kamere (nema gibanja i kamere i scene), nedovoljna mogućnost stabilizacije (samo uklanjanje trešnje kamere, bez uklanjanja velike translacije i rotacije letjelice), moguća nedostupnost mape uzvišenja, nedovoljna automatiziranost (ručna stabilizacija može biti skupa te nedovoljno točna).

Shastri i Schowengerdt [14] stabiliziraju video praćenjem značajki između slika u video zapisu i primjenom projekcijske transformacije pomoću praćenih značajki na slike video zapisa. U radu nije istražena mogućnost dugotrajnije

stabilizacije nakon što se sve značajke početne slike video zapisa nađu van područja snimanja kamere. Ispitni video zapis sadrži prikaz ravnog dijela ceste bez raskrižja, pa automobili uvijek imaju jednaku orijentaciju. Automobili su položeni u smjeru osi apscise referentne slike, stoga je moguće korištenje prozora za praćenje s visinom većom od dužine (7×17 piksela). Svaki prozor za praćenje tada nužno obuhvaća i dio nepokretne pozadine, te se tako onemogućuje odabiranje pokretnih automobila kao značajki.

3. Opis metode

Metoda stabilizacije korištena u ovom radu sastoji se od dva glavna dijela:

1. algoritma za praćenje značajki na slikama ulazne video sekvence;
2. algoritma za estimaciju i primjenu projekcijske transformacije (homografije) među slikama.

U video sekvenci se na referentnoj slici odabiru značajke. Odabrane značajke se prate kroz ostale slike video sekvence. Nakon svakog uspješnog praćenja značajki, primjenjujemo projekcijsku transformaciju između trenutne i referentne slike. Rezultat je trenutna slika transformirana u ravninu referentne slike. Tijek metode se ukratko može opisati sljedećim algoritmom:

1. ulazna video sekvenca se razdvaja u niz od n slika;
2. početna slika u nizu (I_1) odabire se kao referentna slika;
3. na referentnoj slici odabiru se značajke za praćenje;
4. za svaku sliku I_k , $k = 2, \dots, n$ (bez referentne slike):
 - (a) prate se značajke između prethodne slike I_{k-1} i trenutne slike I_k te se spremaju nove koordinate značajki;
 - (b) na temelju novih koordinata praćenih značajki i korespondentnih značajki referentne slike procjenjuje se projekcijska transformacija ravnine trenutne slike I_k u ravninu referentne slike;
 - (c) ravnina trenutne slike se primjenom procijenjene transformacije transformira u ravninu referentne slike;
 - (d) pronalaze se značajke *outlieri*¹ te se izbacuju iz popisa praćenih značajki;

¹Podaci koji odudaraju od modela.

- (e) ranije izgubljene značajke koje su se ponovno našle unutar trenutne slike I_k se vraćaju u popis praćenih značajki (*outlieri* se ne vraćaju);
 - (f) sprema se izlazna transformirana slika I'_k ;
5. transformirane slike I'_k , $k = 1, \dots, n$ povezuju se u izlaznu video sekvencu.

3.1. Praćenje značajki

Za potrebe metode stabilizacije prati se velik broj značajki. Praćenje započinje referentnom slikom i vrši se kroz cijelu video sekvencu. Koristi se algoritam Kanadea, Lucasa i Tomasija temeljen na diferencijalnom praćenju [20].

3.1.1. Odabir značajki

Značajke su istaknuti dijelovi slike, tipično rubovi objekata, uglovi, točke interesa, odnosno specifične lokacije na slici koje nam koriste u daljnoj obradi slike, te koje je lako pratiti. Zbog relativno malog pomaka scene među dvjema susjednim slikama u video sekvenci, odabir značajki koje je lako pratiti ima smisla. Odabir značajki vrši se na referentnoj slici.

Postavlja se pitanje kako pronaći značajke koje se lako prate, odnosno za koje s velikom vjerojatnošću možemo pronaći korespondentne značajke u slijednim slikama. Značajke bez tekture, odnosno promjene intenziteta (primjerice, dio neba), je izuzetno teško točno locirati na novoj slici. Značajke s velikim promjenama intenziteta, tj. velikim gradijentom se lakše pronalaze. Značajke koje prikazuju segmente ravne linije imaju veliku promjenu intenziteta u jednom smjeru, no teško se pronalaze zbog “problema otvora” [16, 19]. Segment linije nam daje informaciju o položaju samo u smjeru okomitom na liniju, što nam nije dovoljno.

Najlakše se lociraju značajke s velikim promjenama intenziteta u barem dva smjera, odnosno značajke koje predstavljaju uglove.

Navedene pretpostavke su opisane u nastavku. Usporedimo dvije značajke tako da izračunamo težinsku razliku kvadrata među njima:

$$E_{WSSD}(\mathbf{u}) = \sum_i^{N_p} \omega(\mathbf{x}_i) [I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)]^2, \quad (3.1)$$

pri čemu su I_0 i I_1 dvije značajke (dva dijela slike) koje uspoređujemo, \mathbf{u} vektor pomaka, \mathbf{x}_i koordinate i -tog piksela značajke I , $\omega(\mathbf{x})$ težinska funkcija, a N_p

broj piksela u značajkama. Najčešći oblici težinske funkcije su $\omega(\mathbf{x})$ su Gaussova maska te:

$$\omega(\mathbf{x}) = \begin{cases} 1, & \text{ako je } \mathbf{x} \text{ u značajki koju pratimo} \\ 0, & \text{inače.} \end{cases} \quad (3.2)$$

Pri odabiru značajki za praćenje, nije poznato s kojim dijelovima slijednih slika ćemo uspoređivati odabranu značajku. Stoga možemo značajku usporediti sa samom sobom uz malu varijaciju u položaju $\Delta \mathbf{u}$, odnosno izračunati autokorelacijsku funkciju:

$$E_{AC}(\Delta \mathbf{u}) = \sum_i \omega(\mathbf{x}_i) [I_0(\mathbf{x}_i + \Delta \mathbf{u}) - I_0(\mathbf{x}_i)]^2, \quad (3.3)$$

E_{AC} možemo promatrati i kao različitost značajke od njenog susjedstva u ovisnosti o pomaku $\Delta \mathbf{u}$. Autokorelacijska funkcija za značajke s velikim gradijentima u barem dva smjera ima jasan lokalni minimum, što znači da se potencijalno jednostavno pronalazi unutar slike. Autokorelacijska funkcija za značajke s linearnim prijelazima (gradijent u jednom smjeru) ima više lokalnih minimuma, dok značajke bez promjena intenziteta nemaju izražene minimume autokorelacijske funkcije. Koristimo Taylorev razvoj funkcije:

$$I_0(\mathbf{x}_i + \Delta \mathbf{u}) \approx I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i) \Delta \mathbf{u}, \quad (3.4)$$

gdje je $\nabla I_0(\mathbf{x}_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y} \right) (\mathbf{x}_i) = (I_x, I_y)$, odnosno I_x i I_y su gradijenti značajke po apscisi i ordinati. Autokorelacijsku funkciju tada možemo zapisati kao:

$$\begin{aligned} E_{AC}(\Delta \mathbf{u}) &= \sum_i \omega(\mathbf{x}_i) [I_0(\mathbf{x}_i + \Delta \mathbf{u}) - I_0(\mathbf{x}_i)]^2 \\ &\approx \sum_i \omega(\mathbf{x}_i) [I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i) \Delta \mathbf{u} - I_0(\mathbf{x}_i)]^2 \\ &= \sum_i \omega(\mathbf{x}_i) [\nabla I_0(\mathbf{x}_i) \Delta \mathbf{u}]^2 \\ &= \sum_i \omega(\mathbf{x}_i) \Delta \mathbf{u}^\top \nabla I_0(\mathbf{x}_i)^\top \nabla I_0(\mathbf{x}_i) \Delta \mathbf{u} \\ &= \Delta \mathbf{u}^\top \sum_i \omega(\mathbf{x}_i) \nabla I_0(\mathbf{x}_i)^\top \nabla I_0(\mathbf{x}_i) \Delta \mathbf{u} \\ &= \Delta \mathbf{u}^\top \mathbf{A} \Delta \mathbf{u}. \end{aligned} \quad (3.5)$$

Matrica \mathbf{A} je tada:

$$\mathbf{A} = \sum_i \omega(\mathbf{x}_i) \cdot \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \quad (3.6)$$

Analizom svojstvenih vrijednosti λ_0 i λ_1 autokorelacijske matrice \mathbf{A} svih potencijalnih značajki može se pronaći lokacija pogodnih značajki za praćenje kao

lokalni maksimum svih manjih svojstvenih vrijednosti λ_0 . Nesigurnost u lokaciji praćenih značajki više ovisi o manjoj svojstvenoj vrijednosti.

Osim kao lokalni maksimum svojstvenih vrijednosti λ_0 , odabrane značajke se mogu pronaći i pomoću mjere:

$$\lambda_0\lambda_1 - k(\lambda_0 + \lambda_1)^2, \quad (3.7)$$

uz $k = 0.06$ [7]. Ova mjera je također invarijantna na rotaciju i ne odabire značajke koje prikazuju ravne linije tj. rubove (za koje je $\lambda_1 \gg \lambda_0$). Još neke mjere za odabir značajki su $\lambda_0 - k\lambda_1$ uz preporučeni $k = 0.05$ [21], te $\frac{\lambda_0\lambda_1}{\lambda_0+\lambda_1}$ [3].

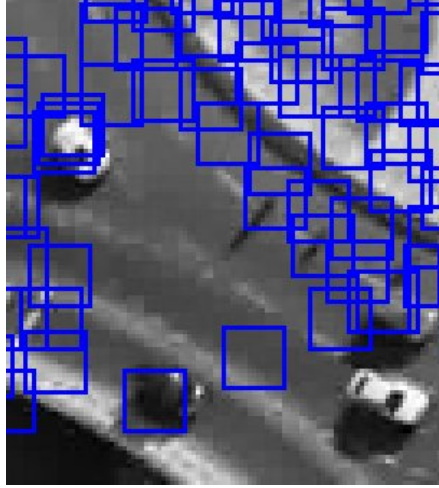
Algoritam odabira značajki može se opisati na sljedeći način:

1. Za svaku potencijalnu značajku na slici:
 - (a) Izračunaj horizontalne i vertikalne gradijente značajke I_x i I_y (ovo je moguće napraviti konvoluiranjem slike s derivacijama Gaussiana)
 - (b) Odredi autokorelacijsku matricu \mathbf{A} značajke po formuli (3.6)
 - (c) Izračunaj svojstvene vrijednosti λ_0 i λ_1 autokorelacijske matrice \mathbf{A}
 - (d) Izračunaj mjeru kvalitete značajke pomoću jedne od predloženih formula, primjerice (3.7).
2. Pronađi lokalni maksimum među svim izračunatim mjerama kvalitete značajke. Značajka kojoj pripada vrijednost lokalnog maksimuma odabire se kao pogodna za praćenje.
3. Ponovi gornje korake za željeni broj značajki N_f .

Primjer dijela odabranih značajki za praćenje na ispitnom video zapisu prikazan je na slici 3.1.

3.1.2. Praćenje značajki

Nakon odabira značajki na referentnoj slici, potrebno je odabrane značajke pratiti na svim sljedećim slikama u video sekvenci. To je moguće napraviti traženjem lokacija na slijednim slikama koje za korespondentnu značajku daju minimalnu težinsku razliku kvadrata, što je prikazano u formuli (3.1). Ako se osvjetljenje na slikama mijenja, može se umjesto težinske razlike kvadrata računati normalizirana križna korelacija.



Slika 3.1: Praćene značajke slike ispitnog video zapisa.

Kod velikog prostora pretraživanja značajki, može se koristiti hijerarhijsko pretraživanje. Podudarajuće značajke pronalaze se na segmentu slike male rezolucije. Zatim odabrani segment promatramo u većoj rezoluciji i dijelimo na manje segmente. Ponovno tražimo značajku, ali koristeći segment više rezolucije. Postupak ponavljamo dok ne dođemo do izvorne rezolucije slike. Navedeni postupak ubrzava pretraživanje značajki. Položaj značajke se tada poboljšava gradijentnom optimizacijom kako bi se dobio konačni položaj značajke. U postupku gradijentne optimizacije potrebno je naći položaj značajke na trenutnoj slici koji minimizira razliku kvadrata između značajke na trenutnoj slici I_t i značajke na referentnoj slici I_0 :

$$\hat{\Delta \mathbf{p}} = \arg \min_{\Delta \mathbf{p}} \sum_{\mathbf{x}} (I_t(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}) - I_0(\mathbf{x}))^2, \quad (3.8)$$

pri čemu je \mathbf{x} vektor koordinata slike, \mathbf{p} vektor parametara transformacije značajke I_0 na trenutnoj slici I_t , a $\hat{\Delta \mathbf{p}}$ trenutna procjena vektora \mathbf{p} u jednoj iteraciji optimizacije. Razlučivost $\Delta \mathbf{p}$ je manja od jednog piksela. $I_t(\mathbf{x}, \mathbf{p})$ preko Taylorovog razvoja funkcije zapisujemo kao:

$$I_t(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}) \approx I_t(\mathbf{x}, \mathbf{p}) + \frac{\partial I_t}{\partial \mathbf{p}}(\mathbf{x}, \mathbf{p}) \cdot \Delta \mathbf{p} \quad (3.9)$$

Izraz (3.8) tada možemo zapisati kao:

$$\hat{\Delta \mathbf{p}} = \arg \min_{\Delta \mathbf{p}} \sum_{\mathbf{x}} \left(I_t(\mathbf{x}, \mathbf{p}) + \frac{\partial I_t}{\partial \mathbf{p}}(\mathbf{x}, \mathbf{p}) \cdot \Delta \mathbf{p} - I_0(\mathbf{x}) \right)^2 \quad (3.10)$$

$$= \arg \min_{\Delta \mathbf{p}} \sum_{\mathbf{x}} \left(e + \mathbf{g}^\top \Delta \mathbf{p} \right)^2. \quad (3.11)$$

Minimum $\Delta \mathbf{p}$ se dobiva izjednačavanjem derivacije s nulom:

$$\sum_{\mathbf{x}} 2(e + \mathbf{g}^\top \Delta \mathbf{p}) \cdot \mathbf{g}^\top = \mathbf{0}^\top. \quad (3.12)$$

Rezultat iteracije je rješenje linearnog sustava:

$$\sum_{\mathbf{x}} \mathbf{g}e + \sum_{\mathbf{x}} \mathbf{g}\mathbf{g}^\top \Delta \mathbf{p} = \mathbf{0}. \quad (3.13)$$

Optimiranje prekidamo kad poboljšanje rješenja postane neznatno malo ili kad se novi položaj značajke nađe izvan slike.

U dužim video sekvencama, izgled značajki se postupno mijenja zbog velikog broja slika na kojima ih pratimo. Time se akumulira pogreška u praćenju. Potrebno je odlučiti treba li nastaviti praćenje originalne značajke ili značajku odbaciti te zamijeniti novom (ako je zamjena potrebna). Kako bismo smanjili pogrešku u praćenju, koristimo provjeru afine konzistentnosti značajke. Originalnu značajku uspoređujemo s korespondentnim dijelovima slike u svim slijednim slikama koristeći afini model gibanja. Za svaku značajku se računa sličnost između vrijednosti intenziteta u prozoru za praćenje oko značajke u trenutnoj slici te vrijednosti intenziteta u prozoru za praćenje u referentnoj slici. Ako je sličnost mala, značajka je nekonzistentna i može se odbaciti kao loša značajka.

3.2. Estimacija i primjena homografije

U drugom dijelu metode potrebno je odrediti ravninsku transformaciju, tj. homografiju, između trenutne slike i početne (referentne) slike video zapisa. Za procjenu homografije koriste se koordinate praćenih značajki obje slike. Procijenjenu homografiju primjenjujemo na trenutnu sliku kako bismo ravninu trenutne slike prebacili u ravninu referentne slike.

Homografiju možemo opisati jednadžbom:

$$\lambda \cdot \mathbf{x}' = \mathbf{H} \cdot \mathbf{x}, \quad (3.14)$$

gdje je \mathbf{x}' točka u koordinatnom sustavu transformirane slike, \mathbf{x} korespondentna točka u koordinatnom sustavu originalne slike, \mathbf{H} matrica projekcijske transformacije, a λ koeficijent, odnosno:

$$\lambda \cdot \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (3.15)$$

Prije primjene homografije, točka \mathbf{x} je proširena homogenom koordinatom. Uzevši u obzir homogenost problema, vrijedi:

$$\mathbf{x}' \times (\mathbf{H} \cdot \mathbf{x}) = \mathbf{0}. \quad (3.16)$$

To možemo napisati na sljedeći način:

$$\begin{aligned} \mathbf{x}' \times (\mathbf{H} \cdot \mathbf{x}) &= \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{x}_{H,1} \\ \mathbf{x}_{H,2} \\ \mathbf{x}_{H,3} \end{bmatrix} = \begin{bmatrix} y' \cdot \mathbf{x}_{H,3} - \mathbf{x}_{H,2} \\ \mathbf{x}_{H,1} - x' \cdot \mathbf{x}_{H,3} \\ x' \cdot \mathbf{x}_{H,2} - y' \cdot \mathbf{x}_{H,1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_3^\top & -\mathbf{x}^\top & y'\mathbf{x}^\top \\ \mathbf{x}^\top & \mathbf{0}_3^\top & -x'\mathbf{x}^\top \\ -y'\mathbf{x}^\top & x'\mathbf{x}^\top & \mathbf{0}_3^\top \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_1^\top \\ \mathbf{H}_2^\top \\ \mathbf{H}_3^\top \end{bmatrix} = \mathbf{0}_3 \end{aligned} \quad (3.17)$$

Uvedimo oznaku

$$\mathbf{A}_{3 \times 9} = \begin{bmatrix} \mathbf{0}_3^\top & -\mathbf{x}^\top & y'\mathbf{x}^\top \\ \mathbf{x}^\top & \mathbf{0}_3^\top & -x'\mathbf{x}^\top \\ -y'\mathbf{x}^\top & x'\mathbf{x}^\top & \mathbf{0}_3^\top \end{bmatrix}. \quad (3.18)$$

Za svaki par točaka dobivamo dvije nezavisne jednadžbe. Za svih n parova točaka potrebno je riješiti sustav:

$$\mathbf{A}_{2n \times 9} \cdot \begin{bmatrix} \mathbf{H}_1^\top \\ \mathbf{H}_2^\top \\ \mathbf{H}_3^\top \end{bmatrix} = \mathbf{0}_{2n}, \quad (3.19)$$

pri čemu je $\mathbf{A}_{2n \times 9}$ matrica koja sadrži korespondentne parove koordinata praćenih značajki iz trenutne i referentne slike, a n je broj korespondentnih parova. Matrica $\mathbf{H}_{3 \times 3}$ je rješenje jednadžbe i sadrži koeficijente korištene u homografiji (\mathbf{H}_i je i -ti red matrice \mathbf{H}). Sustav koji je potrebno riješiti je sustav s viškom ograničenja, jer je $2n$ obično puno veće od 9, te se stoga ne može doći do posve točnog rješenja.

Sustav se stoga može riješiti algebarski, minimizacijom L_2 norme lijeve strane jednadžbe:

$$\hat{\mathbf{h}}_{\text{alg}} = \arg \min_{\mathbf{h}_9} \|\mathbf{A}_{2n \times 9} \cdot \mathbf{h}_9\|. \quad (3.20)$$

To je moguće ostvariti korištenjem SVD dekompozicije matrice.

No budući da takvo rješenje ne maksimizira vjerodostojnost vektora \mathbf{h}_9 , dodatno je optimizirano rješenje korištenjem Levenberg-Marquardtovog algoritma (gradijentna optimizacija), čime se dobiva dobro procijenjena ravninska transformacija.

Prije procjene homografije provedeno je odbacivanje vanpopulacijskih značajki (engl. *outliers*) korištenjem algoritma RANSAC (engl. *Random sample consensus*). U slučaju snimke prometnog raskrižja, outlieri često pripadaju automobilima u pokretu, budući da oni odudaraju od željene ravninske transformacije među slikama, tj. ne pripadaju pozadini slike.

Nakon toga možemo primijeniti ravninsku transformaciju trenutne slike u odnosu na referentnu sliku. Procijenjena transformacija je primijenjena na trenutnu sliku na sljedeći način:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix}^\top = \mathbf{H}_{3 \times 3} \cdot \begin{bmatrix} x & y & 1 \end{bmatrix}^\top, \quad (3.21)$$

gdje je (x, y) točka u trenutnoj slici video zapisa, (x', y') je točka (x, y) prebačena u ravninu referentne slike video zapisa, a $\mathbf{H}_{3 \times 3}$ je matrica homografije.

Općeniti algoritam procjene i primjene homografije možemo opisati na sljedeći način:

1. za svaku sliku u slijedu slika ulazne video sekvence:
 - (a) odbaciti outlieri RANSAC metodom;
 - (b) algebarski procijeniti homografiju $\hat{\mathbf{h}}_{\text{alg}}$ između referentne i trenutne slike korištenjem SVD dekompozicije matrice;
 - (c) gradijentno optimirati rješenje korištenjem Levenberg-Marquardtovog algoritma;
 - (d) procijenjenom homografijom transformirati trenutnu sliku u ravninu referentne slike.

Postupkom praćenja u kombinaciji s estimacijom i primjenom homografije, slike u ulaznom video zapisu transformiraju se u ravninu referentne (početne) slike video zapisa. To predstavlja osnovu razvijenog sustava za stabilizaciju videa.

Na slici 3.2 prikazan je primjer korištenja homografije u stabilizaciji video zapisa.

3.2.1. Algoritam RANSAC

Algoritam RANSAC (engl. *random sample consensus*) je iterativna metoda za procjenu parametara matematičkog modela iz skupa podataka. Osnovna pretpostavka je da se podaci sastoje od populacijskih i vanpopulacijskih podataka. Populacijski podaci su podaci čija se distribucija može objasniti modelom, dok

se vanpopulacijski podaci ne mogu ukomponirati u model. Algoritam RANSAC koristi najmanji mogući početni skup podataka i povećava ga podacima koji pripadaju modelu. U razvijenom sustavu za stabilizaciju videa vanpopulacijski podaci su značajke koje ne odgovaraju estimiranoj homografiji, te moraju biti uklonjene kako bi se greška u procjeni homografije smanjila. Algoritam procjene parametara matematičkog modela RANSAC je sljedeći:

1. za svaki par referentne i trenutne slike u videu za koji želimo izračunati homografiju:
 - (a) odabrati četiri para točaka iz skupa korespondentnih praćenih značajki s referentne i trenutne slike;
 - (b) izračunati homografiju \mathbf{H} na temelju odabrana četiri para točaka;
 - (c) za svaki par točaka $(\mathbf{x}, \mathbf{x}')$ iz skupa korespondentnih praćenih značajki izračunati $d(\mathbf{H}\mathbf{x}, \mathbf{x}')$, gdje je $d(\cdot)$ Euklidska udaljenost između dvije točke;
 - (d) ako je $d(\mathbf{H}\mathbf{x}, \mathbf{x}') < t$ za prag t , par točaka se slaže s izračunatom homografijom;
 - (e) odabrati sve parove točaka koji se slažu s homografijom;
 - (f) ponavljati gornje korake dok se ne maksimizira broj parova točaka koji se slažu s homografijom;
 - (g) ostale točke su vanpopulacijski podaci tj. outlieri, te ih odbacujemo;
 - (h) ponovno izračunati homografiju \mathbf{H} korištenjem samo populacijskih podataka.

3.2.2. SVD dekompozicija matrice

Neka je \mathbf{A} pravokutna realna matrica tipa $m \times n$. Tada postoje ortogonalne kvadratne matrice \mathbf{U} reda m i \mathbf{V} reda n , te dijagonalna pravokutna matrica \mathbf{D} reda $m \times n$, takve da je:

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top. \quad (3.22)$$

Na dijagonali matrice \mathbf{D} su singularne vrijednosti $\sigma_1, \dots, \sigma_r$ matrice \mathbf{A} , gdje je r rang matrice. Matrica \mathbf{U} kao stupce ima svojstvene vektore simetrične kvadratne matrice $\mathbf{A}\mathbf{A}^\top$ reda m , a \mathbf{V} za stupce ima svojstvene vektore matrice $\mathbf{A}^\top\mathbf{A}$ reda

n . Ovakav rastav matrice se naziva singularna dekompozicija matrice (engl. *SVD decomposition*). Vrijedi:

$$\begin{aligned}\mathbf{A}\mathbf{A}^\top &= \mathbf{U}\mathbf{D}\mathbf{V}^\top\mathbf{V}\mathbf{D}\mathbf{U}^\top \\ &= \mathbf{U}\mathbf{D}^2\mathbf{U}^\top\end{aligned}\tag{3.23}$$

$$\begin{aligned}\mathbf{A}^\top\mathbf{A} &= \mathbf{V}\mathbf{D}\mathbf{U}^\top\mathbf{U}\mathbf{D}\mathbf{V}^\top \\ &= \mathbf{V}\mathbf{D}^2\mathbf{V}^\top\end{aligned}\tag{3.24}$$

Singularne vrijednosti $\sigma_1, \dots, \sigma_r$ matrice \mathbf{A} su kvadrati svojstvenih vrijednosti matrica $\mathbf{A}\mathbf{A}^\top$ i $\mathbf{A}^\top\mathbf{A}$.

U slučaju izračuna homografije, zanimaju nas svojstveni vektori matrice $\mathbf{A}^\top\mathbf{A}$, jer je vektor \mathbf{h} koji tražimo takav svojstveni vektor sa svojstvenom vrijednošću najbližom nuli. Singularna dekompozicija matrice može se izračunati na način da se singularne vrijednosti u matrici \mathbf{D} pojavljuju na dijagonali, od najveće prema najmanjoj.

3.2.3. Levenberg-Marquardtova gradijentna optimizacija

Levenberg-Marquardtov algoritam je iterativni postupak pronalaženja numeričkog rješenja problema minimizacije funkcije. Algoritam predstavlja most između Gauss-Newtonovog algoritma te postupka gradijentnog spusta. Algoritam je robustniji od Gauss-Newtonovog algoritma jer lakše pronalazi rješenja i ako se početno rješenje nalazi daleko od traženog minimuma. U tom slučaju algoritam se ponaša kao metoda gradijentnog spusta. Ako je početno rješenje relativno blizu traženog minimuma, algoritam se ponaša kao Gauss-Newtonov algoritam. Time dobijamo na brzini dolaska do rješenja, budući da Gauss-Newtonov algoritam ne zahtijeva izračun drugih derivacija funkcije. Algoritam možemo opisati kao u [10]. Definirajmo funkciju f za koju vrijedi $f(\mathbf{p}) = \hat{\mathbf{x}}$ pri čemu je $\hat{\mathbf{x}}$ procjena mjerene vrijednosti \mathbf{x} . Cilj je odrediti parametar

$$\mathbf{p}^+ = \arg \min_{\mathbf{p}} \epsilon(\mathbf{p})^\top \epsilon(\mathbf{p}),\tag{3.25}$$

pri čemu je $\epsilon(\mathbf{p}) = \mathbf{x} - f(\mathbf{p})$. Zadanu funkciju f možemo razviti Taylorevim razvojem:

$$f(\mathbf{p} + \Delta\mathbf{p}) \approx f(\mathbf{p}) + \mathbf{J} \cdot \Delta\mathbf{p},\tag{3.26}$$

gdje je $\mathbf{p} = (x_1, \dots, x_n)$ vektor parametara funkcije, $\Delta\mathbf{p}$ mali pomak, a matrica \mathbf{J} Jakobijan funkcije f . Jakobijan funkcije se može izraziti kao:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_n} & \cdots & \frac{\partial f}{\partial x_1} \end{bmatrix} \quad (3.27)$$

Algoritam je iterativan. Za početni parametar \mathbf{p}_0 , potrebno je minimizirati grešku e :

$$e = \|\mathbf{x} - f(\mathbf{p} + \Delta\mathbf{p})\|, \quad (3.28)$$

gdje je \mathbf{x} traženi minimum funkcije f . Prema razvoju funkcije, možemo zapisati:

$$e \approx \|\mathbf{x} - f(\mathbf{p}) - \mathbf{J} \cdot \Delta\mathbf{p}\| \quad (3.29)$$

$$\approx \|\epsilon - \mathbf{J} \cdot \Delta\mathbf{p}\| \quad (3.30)$$

Minimum se nalazi izjednačavanjem derivacije izraza (3.30) s nulom:

$$\mathbf{J}^\top (\mathbf{J} \cdot \Delta\mathbf{p} - \epsilon) = \mathbf{0}. \quad (3.31)$$

Vrijedi jednažba:

$$\mathbf{J}^\top \mathbf{J} \Delta\mathbf{p} = \mathbf{J}^\top \epsilon. \quad (3.32)$$

Levenberg-Marquardtov algoritam rješava modifikaciju jednažbe (3.32):

$$\mathbf{N} \Delta\mathbf{p} = \mathbf{J}^\top \epsilon, \quad (3.33)$$

gdje je \mathbf{N} matrica čiji su elementi van dijagonale jednaki elementima matrice $\mathbf{J}^\top \mathbf{J}$ na istim pozicijama ($b_{ij} = a_{ij}$), a elementi na dijagonali su jednaki $b_{ii} = \mu + a_{ii}$ za elemente a_{ii} matrice $\mathbf{J}^\top \mathbf{J}$ te neki $\mu > 0$. Dodavanje parametra μ se naziva prigušenje. Ako novoizračunati vektor $\mathbf{p} + \Delta\mathbf{p}$ za vrijednost $\Delta\mathbf{p}$ izračunatu jednažbom (3.33) dovodi do smanjenja pogreške ϵ , prigušenje μ se smanjuje. Ako ne dolazi do smanjenja pogreške, μ se povećava. Proces rješavanja jednažbe (3.33) za različite vrijednosti prigušenja μ dok se ne nađe $\Delta\mathbf{p}$ koji dovodi do smanjenja pogreške čini jednu iteraciju Levenber-Marquardtovog algoritma.

Ako je prigušenje veliko, matrica \mathbf{N} je gotovo dijagonalna i vrijednost $\Delta\mathbf{p}$ je bliska vrijednosti koja bi se dobila korištenjem gradijentnog spusta. Malo prigušenje algoritam čini bliskim Gauss-Newtonovom postupku. Algoritam završava kad je dostignut maksimalan broj iteracija ili kada vrijednost greške ili promjene prigušenja padne ispod praga t .

3.3. Ponovni početak praćenja

Zbog kretanja kamere neke se značajke na pojedinim slikama nađu van okvira slike. Već nakon kraće sekvence video zapisa može se dogoditi da sve početne značajke izađu van okvira slike. No, kako se kamera u snimkama prometnog raskrižja nastoji vratiti na istu poziciju, izgubljene značajke se mogu ponovno vratiti u okvir slike. Kako bi se omogućilo dugotrajnije praćenje, u sustav je dodana metoda za traženje izgubljenih značajki i nastavak njihovog praćenja čak i kad se one nađu van slike. Tako se značajke ponovno mogu koristiti u izračunu homografije ako se opet nađu unutar slike.

Nakon praćenja značajki između trenutne i referentne slike, pamte se koordinate svake značajke na referentnoj slici koju KLT nije uspio pratiti (izgubljene značajke). Nakon izračuna homografije, koordinate svake izgubljene značajke se transformiraju u ravninu trenutne slike te KLT-u dojavu kao pronađene značajke. Ova metoda omogućuje stabilizaciju dugotrajne video sekvence. Stabilizacija se prekida kada se gotovo sve značajke referentne slike potpuno izgube, tj. kad više ne možemo ravninu trenutne slike projicirati na ravninu referentne slike. Algoritam ponovnog praćenja značajki je sljedeći:

1. nakon praćenja značajki i izračuna homografije \mathbf{H} za trenutnu sliku, odredi sve izgubljene značajke koje nisu outlieri;
2. na svaku izgubljenu značajku primijeni homografiju \mathbf{H} ;
3. u sljedeću iteraciju praćenja uključi transformirane izgubljene značajke.

Na slikama 3.3 i 3.4 prikazan je gubitak značajki trenutne slike te ponovno vraćanje značajki unutar jedne od narednih slika.

3.4. Bilinearna interpolacija

Interpolacija je postupak izračuna novih vrijednosti funkcije iz poznatog skupa zadanih vrijednosti. U ovom sustavu interpolacija je potrebna za izračun vrijednosti piksela s realnim koordinatama iz poznatih vrijednosti susjednih piksela.

Linearna interpolacija je oblik interpolacije koji se može koristiti za procjenu vrijednosti funkcija jedne varijable. Procjena vrijednosti funkcije u proizvoljnoj točki x može se izračunati kao:

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0), \quad (3.34)$$

gdje su $f(x_0) = y_0$ i $f(x_1) = y_1$ poznate vrijednosti funkcije u točkama x_0 i x_1 , a $f(x)$ rezultat linearne interpolacije. Formulu možemo napisati i kao:

$$f(x) = \frac{x_1 - x}{x_1 - x_0} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \quad (3.35)$$

Rezultat linearne interpolacije između dvije poznate točke jest dužina između te dvije točke. Linearna interpolacija nad skupom točaka je povezivanje linearnih interpolacija između svakog od parova točaka. Rezultat je krivulja koja se prelama u poznatim točkama funkcije.

Bilinearna interpolacija je proširenje linearne interpolacije za funkcije koje ovise o dvije varijable. Potrebno je prvo provesti linearnu interpolaciju po jednoj od varijabli, a nakon toga po drugoj varijabli. Poznate su vrijednosti funkcije u četiri točke – (x_0, y_0) , (x_0, y_1) , (x_1, y_0) te (x_1, y_1) . Prvo provodimo linearnu interpolaciju po apscisi:

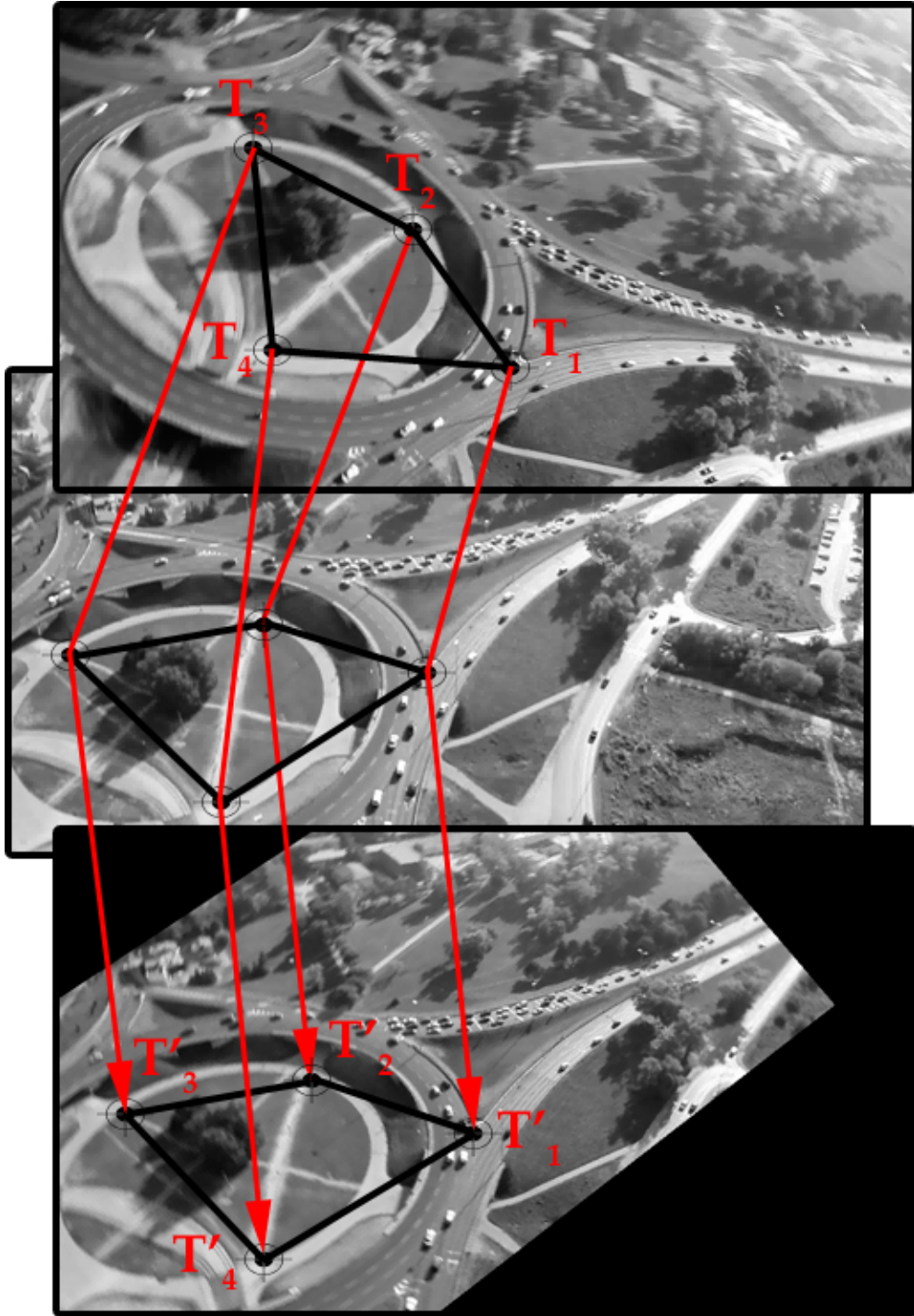
$$f(x, y_0) = \frac{x_1 - x}{x_1 - x_0} f(x_0, y_0) + \frac{x - x_0}{x_1 - x_0} f(x_1, y_0) \quad (3.36)$$

$$f(x, y_1) = \frac{x_1 - x}{x_1 - x_0} f(x_0, y_1) + \frac{x - x_0}{x_1 - x_0} f(x_1, y_1). \quad (3.37)$$

Nakon toga provodimo linearnu interpolaciju po ordinati:

$$f(x, y) = \frac{y_1 - y}{y_1 - y_0} f(x, y_0) + \frac{y - y_0}{y_1 - y_0} f(x, y_1). \quad (3.38)$$

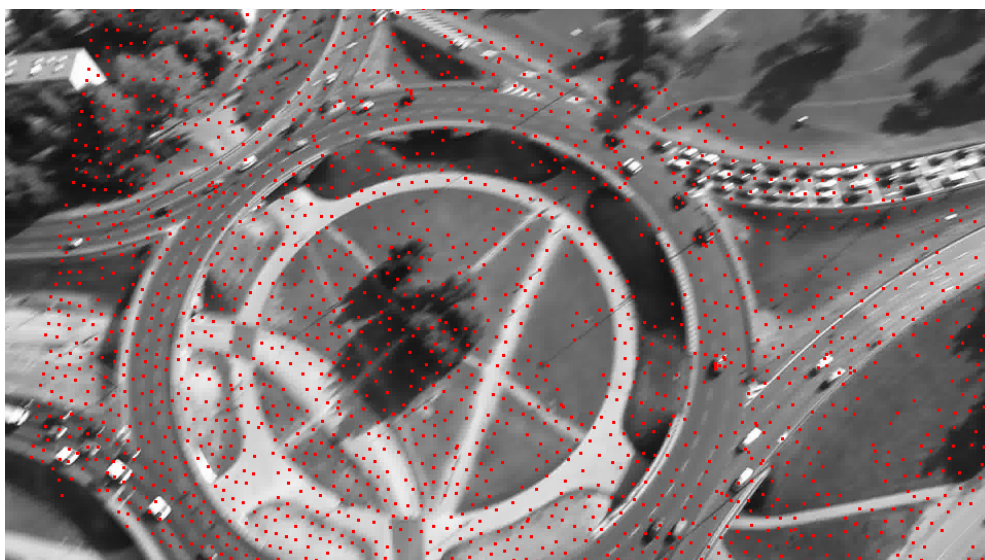
Na ulaznim slikama sustava za stabilizaciju videa, interpolacija je korištena za točke koje su susjedni pikseli, tj. $x_1 = x_0 + 1$ te $y_1 = y_0 + 1$.



Slika 3.2: Slika prikazuje trenutnu sliku u video zapisu (vrh), referentnu sliku (sredina) te transformiranu trenutnu sliku (dno). Točke T_i u trenutnoj slici su transformirane u točke T'_i korištenjem homografije između trenutne slike i referentne slike.



Slika 3.3: Slika prikazuje gubitak značajki izvan trenutne slike radi pomicanja kamere.



Slika 3.4: Na ovoj slici su neke od izgubljenih značajki ponovno pronađene i počinju se ponovno pratiti.

4. Skup podataka

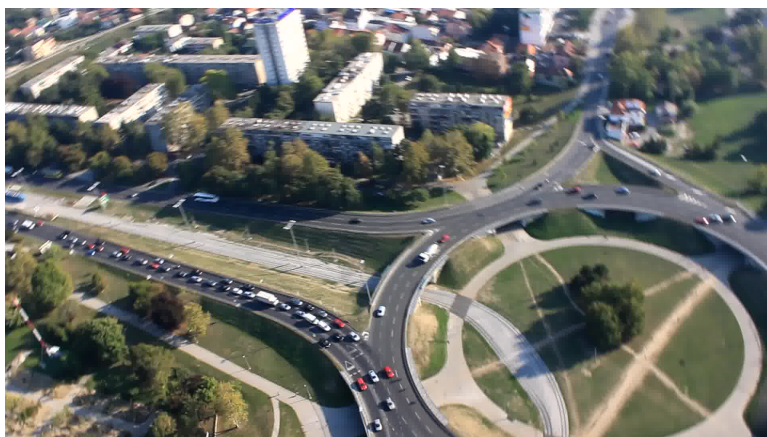
Izgrađeni sustav stabilizacije video zapisa evaluiran je na dva video zapisa snimana iz bespilotne letjelice koja prikazuju promet nad nekoliko raskrižja (kružnih tokova) te ravnih dijelova ceste u Zagrebu. Letjelica se nalazi na visini od oko 100 m. Video zapisi su u trajanju od 10 min 21 s (video A) te 13 min 53 s (video B). Veličine video zapisa su 3.2 GB te 4.3 GB. Dimenzije video zapisa su 1920×1080 piksela. Video zapisi su razdvojeni u slike korištenjem alata *ffmpeg*. Tako dobivenih slika video zapisa ima 12429 za video zapis A te 16660 za video zapis B.

Slike su skalirane na 50% dimenzija, odnosno 960×540 piksela. To je napravljeno kako bi se dobilo na brzini izvođenja stabilizacije za potrebe evaluacije sustava, budući da je brzina stabilizacije svake slike video framea reda veličine nekoliko sekundi, što za cijeli video zapis znači brzinu izvođenja u trajanju od više sati. Osim toga, slike su prebačene u sive tonove (engl. *grayscale*) jer metoda praćenja značajki KLT radi isključivo na sivim tonovima slike. To je napravljeno alatom ImageMagic. Tako dobivene ulazne slike su veličine oko 100-150 kB. Primjeri slika video zapisa za testiranje i evaluaciju su dani na slikama 4.1, 4.2, 4.4 te 4.5. Primjer slike ulaznog video zapisa nakon skaliranja i prebacivanja u sive tonove prikazan je na slici 4.3.

Problem ulaznih video zapisa je u neupotrebljivosti cijelih video zapisa. Naime, veći dio video zapisa prikazuje putovanje letjelice do određiškog raskrižja, te ravne dijelove ceste koji su nam za daljnju obradu manje zanimljivi od raskrižja. Potrebno je odbaciti takve dijelove video zapisa te odabrati samo segmente u kojima kamera nastoji prikazivati raskrižje. U dijelu putovanja letjelice do raskrižja, okolina koju kamera snima često se mijenja te se značajke nepovratno gube. Stabilizacija takvih video zapisa zahtijeva vrlo čestu promjenu referentnih slika. Nakon odbacivanja nepotrebnih segmenata, preostaju video zapisi u trajanju od oko tri minute za video A te pet minuta za video B, te je na njima testiran i evaluiran sustav. Skup slika takvog video zapisa sadrži 3261 sliku za manji video



Slika 4.1: Primjer slike ulaznog video zapisa A.



Slika 4.2: Primjer slike ulaznog video zapisa A.

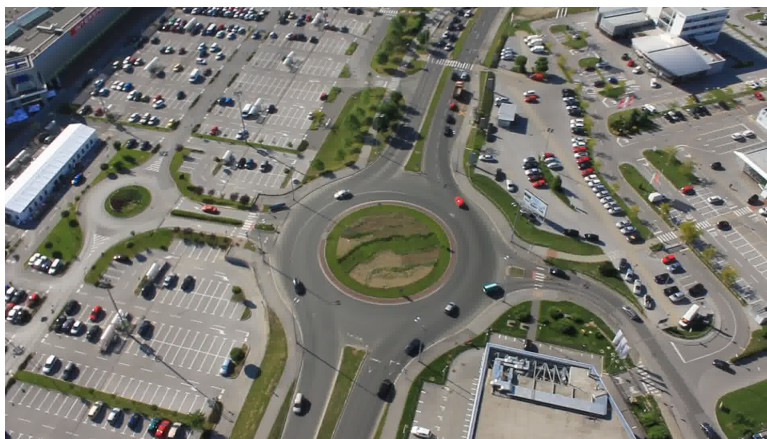
zapis te 6400 slika za veći video zapis. Drugi problem ulaznih video zapisa je i u velikom zamućenju pojedinih slika uslijed kretanja letjelica (većinom rotacije te translacije). Ovo dovodi do povećanja greške u stabilizaciji. Zamućenje možemo vidjeti na slici 4.1.



Slika 4.3: Primjer slike u sivim tonovima ulaznog video zapisa A.



Slika 4.4: Primjer slike ulaznog video zapisa B.



Slika 4.5: Primjer slike ulaznog video zapisa B.

5. Programska implementacija

Razvijeni sustav koristi biblioteku za praćenje značajki KLT, te biblioteku OpenCv “Camera Calibration and 3D Reconstruction” za estimaciju homografije.

5.1. Parametri algoritma KLT

Uspješnost stabilizacije slika uvelike ovisi o parametrima KLT algoritma. Parametri se postavljaju unutar `KLT_TrackingContext` strukture. Stvaranje strukture:

```
KLT_TrackingContext tc = KLTCreatetrackingContext();
```

Bitni parametri:

- broj značajki koje se prate
 - postavljen na 3000, mijenja se u definiciji `FEAT_NUM`
 - koristi se kod stvaranja liste značajki:
`fl = KLTCreatFeatureList(nFeatures)`
- veličina prozora za praćenje značajki – podešavanje:
`tc->window_width; tc->window_height`
- provjera affine konzistentnosti značajki
 - različite provjere konzistentnosti značajki između dvije slike s obzirom na affine transformacije provjera konzistentnosti osigurava izbor kvalitetnijih značajki tj. pouzdanost praćenja, ali smanjuje brzinu izvođenja i znatno reducira broj značajki
 - podešavanje: `tc->affineConsistencyCheck` (vrijednosti su -1, 0, 1, 2; opis: <http://www.ces.clemson.edu/~stb/klt/user/chpt8.html>)

5.2. Implementacija homografije

Podešavanje ravnine trenutne slike u ravninu početne slike izvodi se pomoću homografije. U rješenju je korištena OpenCV-ova implementacija homografije. Primjer korištenja:

```
cvFindHomography(&srcPoints, &dstPoints, H,  
                 CV_RANSAC, RANSAC_TRESHOLD, NULL);
```

Ulazi su dva niza točaka (parovi odgovarajućih točaka početnog i trenutnog framea) te matrica (`CvMat* H`) koja služi za spremanje rezultata funkcije. Dodatno je navedeno da se koristi RANSAC metoda za odbacivanje outliers. Dopuštena greška pri odbacivanju outliers se može promijeniti unutar definicije `RANSAC_TRESHOLD`¹.

Dobivena matrica `H` služi za perspektivnu projekciju slike koja se može provesti metodom `cvWarpPerspective` iz OpenCV biblioteke. Budući da metoda `cvFindHomography` vraća već invertiranu matricu `H`, bitno je metodi `cvWarpPerspective` postaviti parametar `flag` na vrijednost `CV_WARP_INVERSE_MAP+CV_WARP_FILL_OUTLIERS`².

5.3. Prevođenje programskog koda

Programsko rješenje je razvijeno koristeći programski jezik C uz biblioteke KLT (dostupna na: <http://www.ces.clemson.edu/~stb/klt/>), OpenCV (dostupan na: <http://opencv.willowgarage.com/wiki/>; korištena je verzija 2.1.0, ali moguće je da implementacija radi i uz druge verzije) te LibJpeg (dostupan na: <http://libjpeg.sourceforge.net/>).

Biblioteke OpenCV i LibJpeg mogu se naći u raznim distribucijama linuxa, tj. moguće je dohvatiti prevedene verzije iz repozitorija distribucije koristeći alate poput APT-a, YUM-a ili PACMAN-a (potrebno je dohvatiti devel verziju). Biblioteku KLT potrebno je ručno prevesti koristeći GCC prevoditelja: unutar direktorija sa kodom KLT biblioteke potrebno je izvršiti naredbu:

```
make lib
```

¹Više o navedenoj funkciji može se pročitati na: http://opencv.willowgarage.com/documentation/cpp/camera_calibration_and_3d_reconstruction.html#cv-findhomography.

²Više o metodi `cvWarpPerspective` može se pročitati na http://opencv.willowgarage.com/documentation/cpp/geometric_image_transformations.html#cv-warpperspective.

Programska logika nalazi se u datoteci `stabilizacija.c`. Prevođenje programskog koda na Linux operacijskom sustavu koristeći GCC prevoditelj:

```
gcc -o stabilizacija stabilizacija.c -L. -lklut -ljpeg \
-lcxcore -lcv -lhighgui -lcvaux -lm
```

(pretpostavlja se da je KLT biblioteka prevedena te da se datoteka `libklut.a` i sva pripadna programska zaglavlja (`.h` datoteke) nalaze u istom direktoriju kao i `stabilizacija.c`).

5.4. Pokretanje programa

Ulaz u program je putanja do videa razbijenog u frameove u grayscale jpeg formatu, putanja za izlazne datoteke (stabilizirane frameove) te broj frameova. Primjer:

```
./stabilizacija "input/img%04d.jpg" "out/trans%04d.jpg" 100
```

Navedeni primjer će učitati 100 slika počevši od “input/img0000.jpg” s krajem u “input/img0099.jpg,” stabilizirati ih te spremiti u niz slika od “out/trans0000.jpg” do “out/trans0099.jpg.” Ulazni video se može “razbiti” u slike koristeći ffmpeg alat:

```
ffmpeg -i video.avi input/img%04d.jpg
```

Prebacivanje slike u sive tonove (engl. *grayscale*) može se izvesti alatom ImageMagic, primjer (u primjeru se slike dodatno skaliraju na 50% veličine po obe dimenzije):

```
for f in *.jpg; do convert -colorspace gray -sample 50%x50% "$f"
% "small/small_$f"; done
```

Stapanje slika nazad u video može se također izvesti koristeći ffmpeg alat:

```
ffmpeg -r 10 -b 5400000 -i out/trans%04d.jpg stabilno.mp4
```

(parametar *r* određuje broj frameova po sekundi (FPS), a *b* broj bitova po sekundi (bitrate)).

6. Rezultati

Eksperimentalno vrednovanje sustava provedeno je koristeći video zapise opisane u 4. poglavlju. Karakteristike sustava koji želimo optimirati su:

- kvaliteta stabilizacije,
- trajanje stabiliziranog video zapisa,
- brzina obrade.

Parametri sustava koji utječu na karakteristike su:

- broj značajki koje se prate,
- veličina prozora za praćenje,
- provjera afine konzistentnosti značajki,
- prag algoritma RANSAC.

Prvo su predstavljeni rezultati stabilizacije bez ponovnog praćenja značajki. Sustav bez ponovnog praćenja značajki nije u mogućnosti dugotrajno stabilizirati video zapis. Zbog navedenog, metoda evaluacije sustava bez ponovnog praćenja značajki razlikuje se od metode evaluacije sustava s ponovnim praćenjem značajki. Evaluacija sustava bez ponovnog praćenja značajki navedena je u odjeljku 6.1, a evaluacija sustava s ponovnim praćenjem značajki u odjeljku 6.2.

6.1. Evaluacija sustava bez ponovnog praćenja značajki

Uspješnost rezultata može se prikazati kao prosječni pomak piksela transformirane slike u odnosu na korespondentne piksele referentne slike. Kako bi se izračunao pomak, na referentnoj slici je ručno odabrano 5 kontrolnih točaka. Kontrolne točke su nakon toga ručno pronađene na 13 transformiranih slika. Prosječni pomak u svakoj slici se računa kao prosječna razlika između koordinata kontrolnih

točaka u referentnoj slici te transformiranim slikama:

$$\bar{x} = \left[\sum_{j=1}^k \sum_{i=1}^n \frac{|xc_i - x_{i,j}|}{nk} \right], \quad (6.1)$$

$$\bar{y} = \left[\sum_{j=1}^k \sum_{i=1}^n \frac{|yc_i - y_{i,j}|}{nk} \right], \quad (6.2)$$

gdje je $k = 13$, $n = 5$, \bar{x} i \bar{y} su prosječni pomaci na apscisi i ordinati, (xc_i, yc_i) su koordinate i -te kontrolne točke na referentnoj slici, a $(x_{i,j}, y_{i,j})$ su koordinate i -te kontrolne točke na j -toj transformiranoj slici.

Uz korištenju mjere uspješnosti rezultata, na video sekvenci koja sadrži 100 slika, 3000 praćenih značajki, prozor za praćenje veličine 15×15 piksela te prag algoritma RANSAC u iznosu od 2 piksela, (\bar{x}, \bar{y}) za razvijeni sustav bez korištenja ponovnog početka praćenja izgubljenih značajki iznosi (2px, 2px). Rezultat je prikazan na slici 6.1.

Bez korištenja ponovnog početka praćenja izgubljenih značajki, sustav uspijeva stabilizirati video nekoliko sekundi (oko 100 slika video zapisa), nakon čega se gubi sposobnost stabilizacije zbog velikog broja izgubljenih značajki.

6.2. Evaluacija sustava s ponovnim praćenjem značajki

Evaluacija se provodi uz variranje pojedinih parametara. Pretpostavlja se da su parametri nezavisni, te se pri evaluaciji ovisnosti pojedine karakteristike o pojedinom parametru taj parametar varira, a ostali su zamrznuti na standardnim vrijednostima. Standardne vrijednosti parametara su:

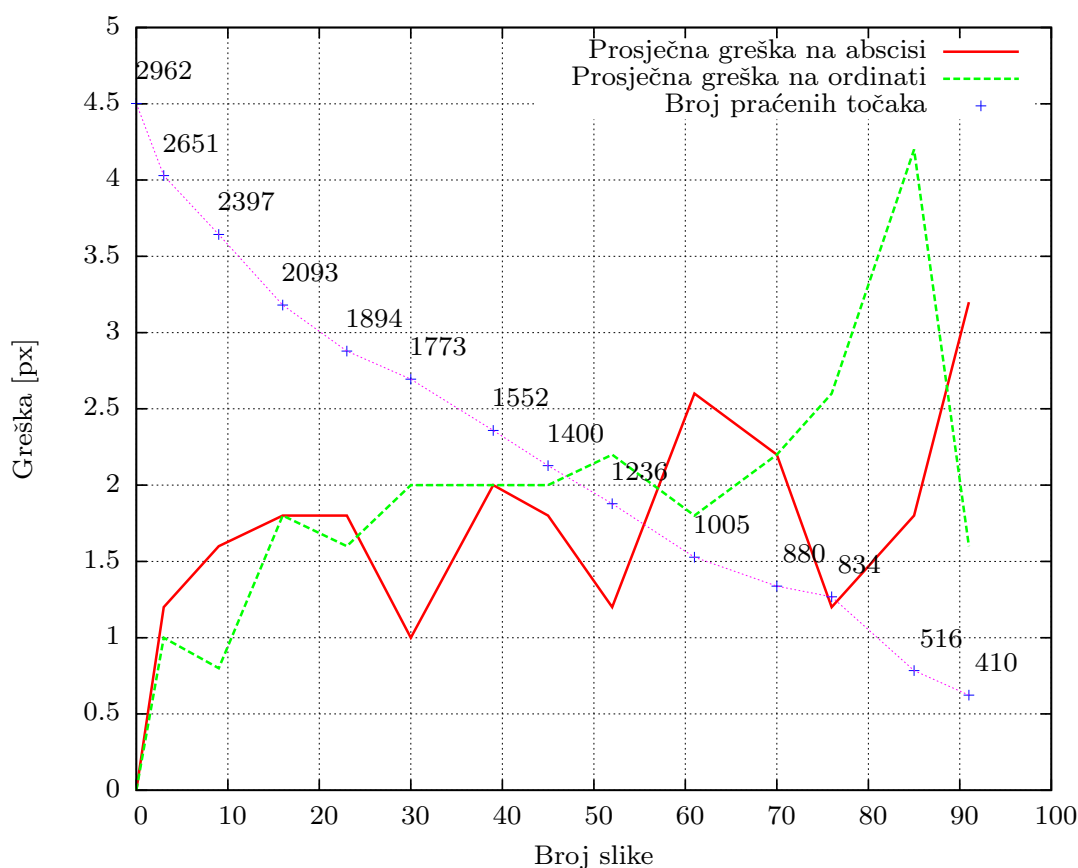
Broj praćenih značajki: 3000,

Veličina prozora za praćenje: 15×15 piksela,

Provjera afine konzistentnosti značajki: provjerava se afina konzistentnost (oznaka 2),

Prag algoritma RANSAC: 40 piksela.

Za potrebe evaluacije, odabran je skup od 300 slijednih slika video zapisa, pri čemu je korišten video zapis A. Navedeni skup je zatim kopiran te mu je obrnut



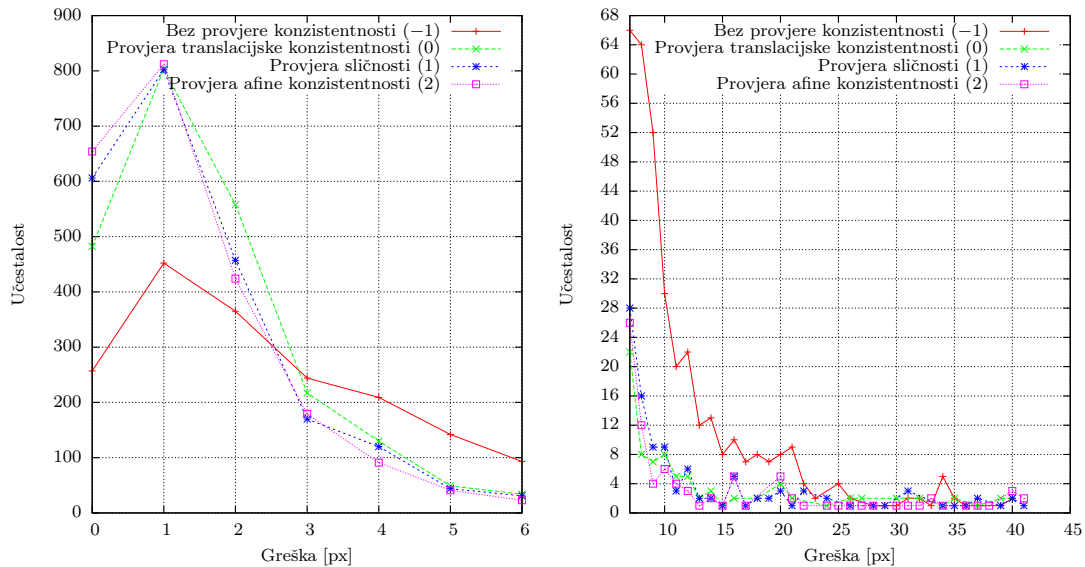
Slika 6.1: Rezultat evaluacije metode bez ponovnog praćenja značajki.

redoslijed. Tako je dobivena video sekvenca u trajanju od 599 slika (bez dupliciranja posljednje slike prvog dijela skupa), pri čemu je početna slika sekvence jednaka posljednjoj slici. Takav skup nam omogućava da na jednostavan način odredimo grešku stabilizacije. Greška se određuje računanjem udaljenosti među koordinatama značajki prve i zadnje slike testnog skupa (radi se o jednakim slikama).

Evaluacija karakteristika vršena je u ovisnosti o svakom od parametara. Rezultati su prikazani na slikama 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9 te 6.10. Slike 6.2, 6.3, 6.4 i 6.5 prikazuju grešku stabilizacije u ovisnosti o pojedinom parametru. Prikaz greške prikazan je preko dva grafa pri čemu lijevi graf prikazuje učestalost manjih grešaka (do 8 piksela), a desni učestalost većih greški. Razlog je nesrazmjer učestalosti manjih i većih greški koji nije pogodan za prikaz na linearnoj ili logaritamskoj skali.

S grafa 6.3 izbačen je slučaj praćenja 100 značajki, a s grafa 6.4 izbačen je slučaj praćenja s pragom algoritma RANSAC u vrijednosti od $2px$. U oba

izbačena slučaja nije uspjela stabilizacija svih 599 slika video zapisa, jer je došlo do gubitka svih značajki. Rezultati su tablično prikazani u tablici 6.1.



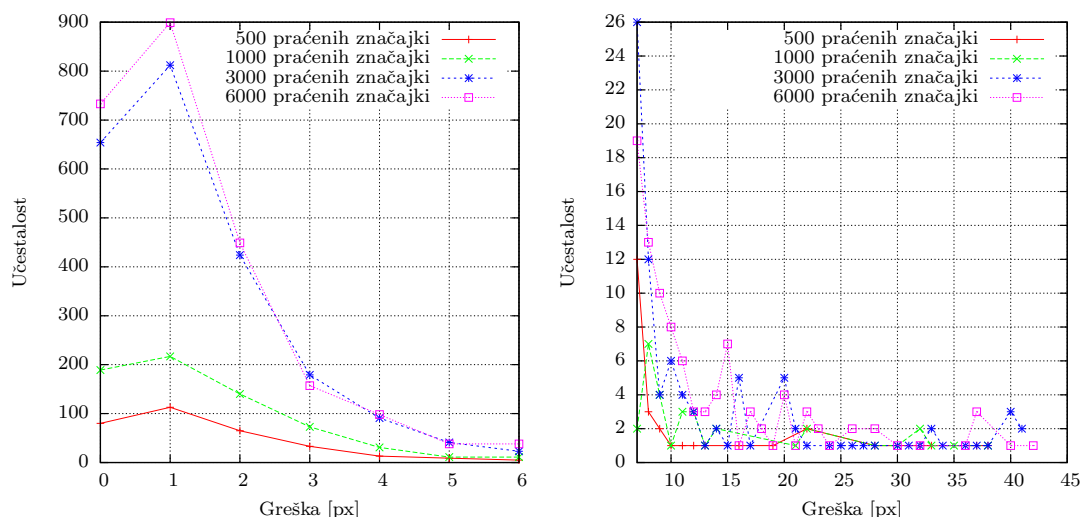
Slika 6.2: Greška u ovisnosti o provjeri affine konzistentnosti značajki. Lijevi graf prikazuje učestalost manjih grešaka (do 6 piksela), a desni učestalost većih greški.

Slike 6.6, 6.7, 6.8 i 6.9 prikazuju broj uspješno praćenih značajki u ovisnosti o pojedinom parametru, što pokazuje trajanje stabiliziranog video zapisa. Bitno je primijetiti da su grafovi simetrični s obzirom na 300-tu sliku. Uzrok tome je korištenje opisanog simetričnog video zapisa u trajanju od 599 slika. Na slikama 6.6, 6.7, 6.8 i 6.9 vidljivo je veliko smanjenje broja značajki oko 150. slike video zapisa. Razlog je veliki pomak kamere izvan okvira referentne slike, pri čemu nije moguće pronaći veći broj značajki.

Ovisnost brzine obrade o parametrima prikazana je na slici 6.10. Povećanje broja praćenih značajki, veličine prozora za praćenje, složenosti provjere konzistentnosti značajki te praga algoritma RANSAC dovodi do povećanja trajanja obrade. Povećanje praga algoritma RANSAC indirektno utječe na trajanje jer se manji broj značajki izbacuje kao *outlieri* te je potrebno pratiti veći broj značajki.

Dopuštena greška u RANSAC metodi (`RANSAC_TRESHOLD`) je postavljena na 40 piksela, što je eksperimentalno utvrđena greška koja daje najuspješnije rezultate pri dugotrajnijoj stabilizaciji na dostupnim video zapisima raskrižja.

Prilikom odbacivanja outliera interesantno je vidjeti gdje se oni na slici nalaze. Stoga je u sustav dodana metoda za označavanje outliera (outlieri su obojani zelenom bojom). Pokazuje se da su outlieri najčešće automobili u pokretu koji



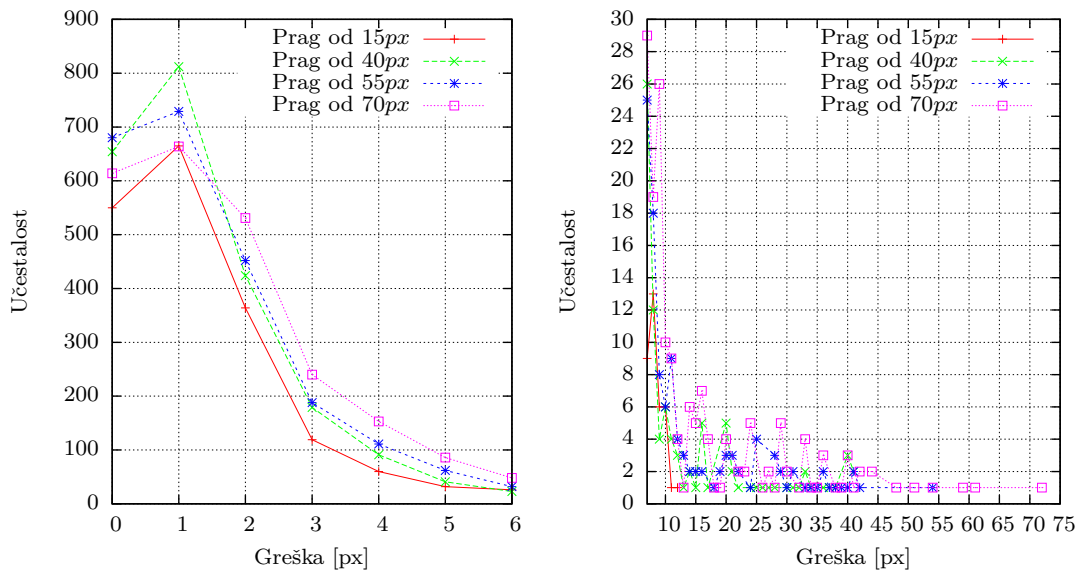
Slika 6.3: Greška u ovisnosti o broju značajki. Lijevi graf prikazuje učestalost manjih grešaka (do 6 piksela), a desni učestalost većih greški.

odudaraju od statične pozadine. Primjer outliera može se vidjeti na slici 6.11.

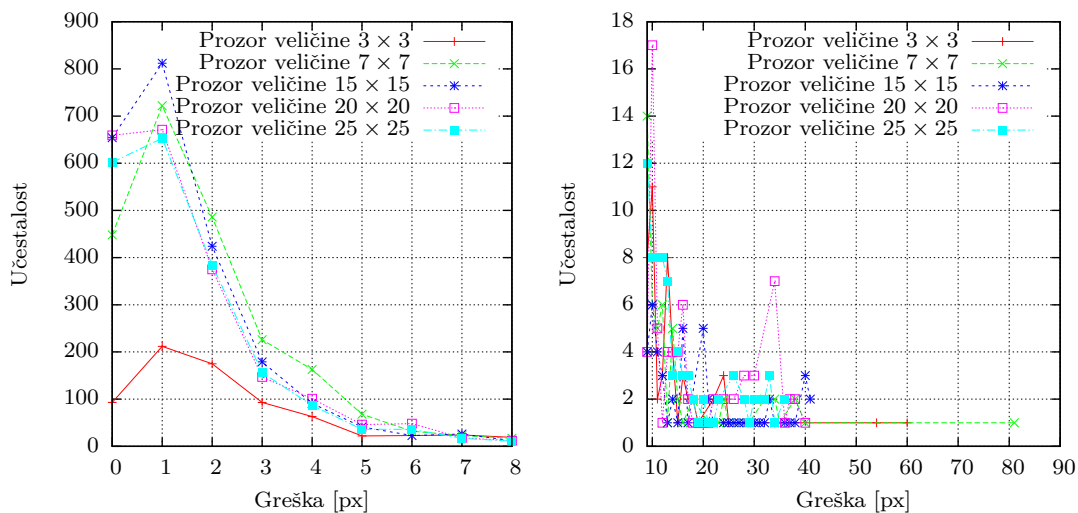
Nakon toga, dodana je metoda za izbacivanje postojećih outliera prije praćenja značajki iduće slike u videu. Nakon testirane video sekvence od 599 frameova, kao outlieri je prepoznato oko 300 značajki od ukupno 3000 praćenih značajki.

Dodatno, testirana je greška stabilizacije tako da je stabilizirana video sekvenca u trajanju od 300 frameova te 299 frameova jednake unazadne video sekvence na način da je zadnja slika video zapisa jednaka početnoj (referentnoj) slici (ukupno 599 slika). Rezultantne slike su vidljive na slici 6.12a (prva, referentna slika) i slici 6.12b (posljednja, transformirana referentna slika). Razlika piksela među dvije slike vidljiva je na slici 6.12c. Rezultati uz korišteno izbacivanje outliera se pokazuju kao nešto bolji. Razlika piksela među dvije slike bez izbacivanja outliera je vidljiva na slici 6.12d, te se uočava da izbacivanje outliera daje bolju sposobnost stabilizacije.

Razlog razlikama između referentne i transformirane referentne slike možemo naći u pogreškama pri praćenju koje nastaju uslijed značajne rotacije i translacije između slika u videu.



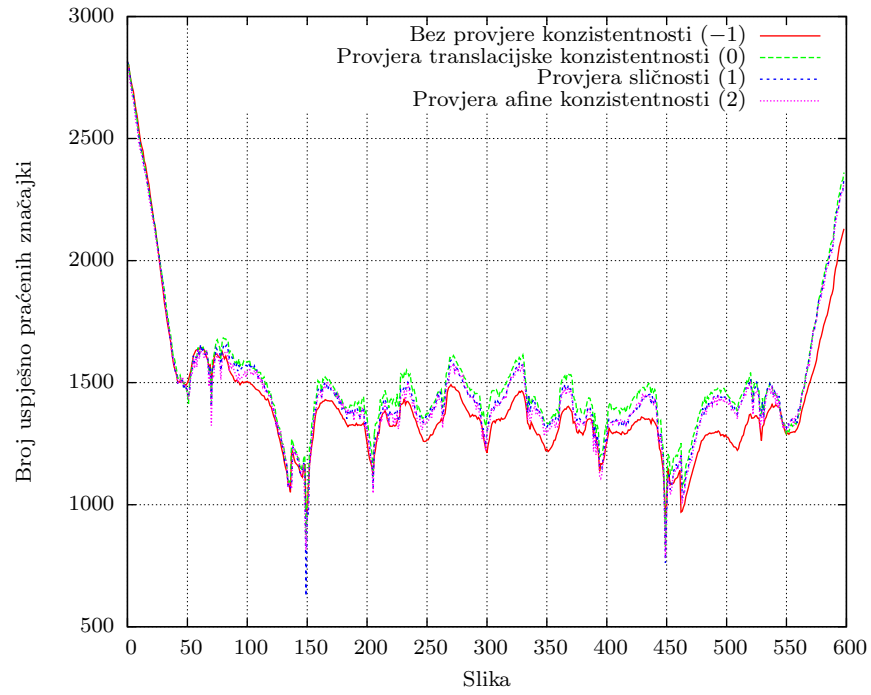
Slika 6.4: Greška u ovisnosti o pragu algoritma RANSAC. Lijevi graf prikazuje učestalost manjih grešaka (do 6 piksela), a desni učestalost većih greški.



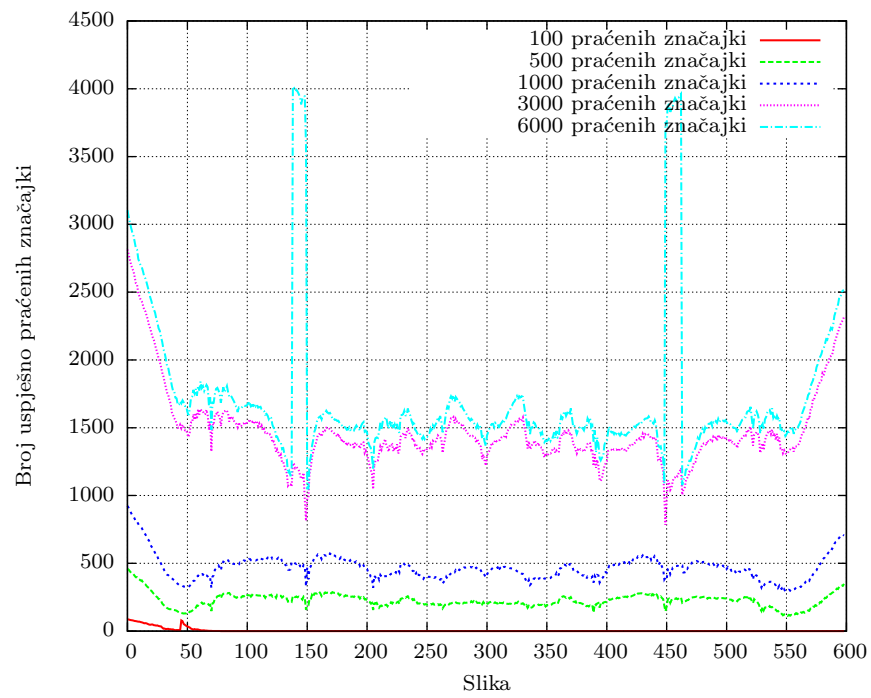
Slika 6.5: Greška u ovisnosti o veličini prozora. Lijevi graf prikazuje učestalost manjih grešaka (do 8 piksela), a desni učestalost većih greški.

Tablica 6.1: Greška stabilizacije u ovisnosti o variranju pojedinog parametra.

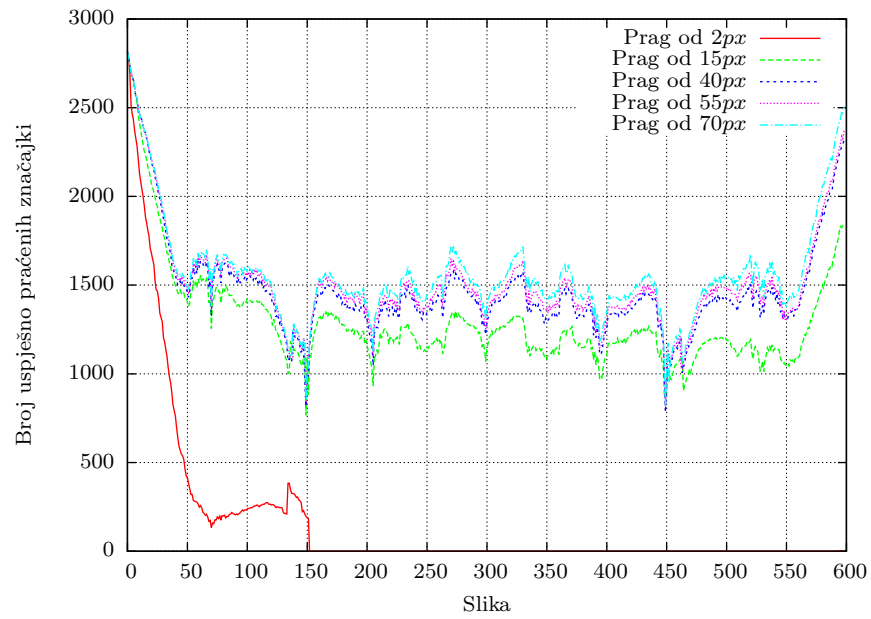
<i>Greška u ovisnosti o broju značajki</i>	
Vrijednost parametra	Prosječna greška [px]
100 praćenih značajki	-
500 praćenih značajki	2.57
1000 praćenih značajki	2.16
3000 praćenih značajki	1.90
6000 praćenih značajki	1.86
<i>Greška u ovisnosti o veličini prozora</i>	
Vrijednost parametra	Prosječna greška [px]
Prozor veličine 3×3	3.22
Prozor veličine 7×7	2.20
Prozor veličine 15×15	1.90
Prozor veličine 20×20	2.11
Prozor veličine 25×25	2.10
<i>Greška u ovisnosti o provjeri afine konzistentnosti značajki</i>	
Vrijednost parametra	Prosječna greška [px]
Bez provjere konzistentnosti (-1)	4.03
Provjera translacijske konzistentnosti (0)	2.07
Provjera sličnosti (1)	2.03
Provjera afine konzistentnosti (2)	1.90
<i>Greška u ovisnosti o pragu algoritma RANSAC</i>	
Vrijednost parametra	Prosječna greška [px]
Prag od $2px$	-
Prag od $15px$	1.44
Prag od $40px$	1.90
Prag od $55px$	2.14
Prag od $70px$	2.72



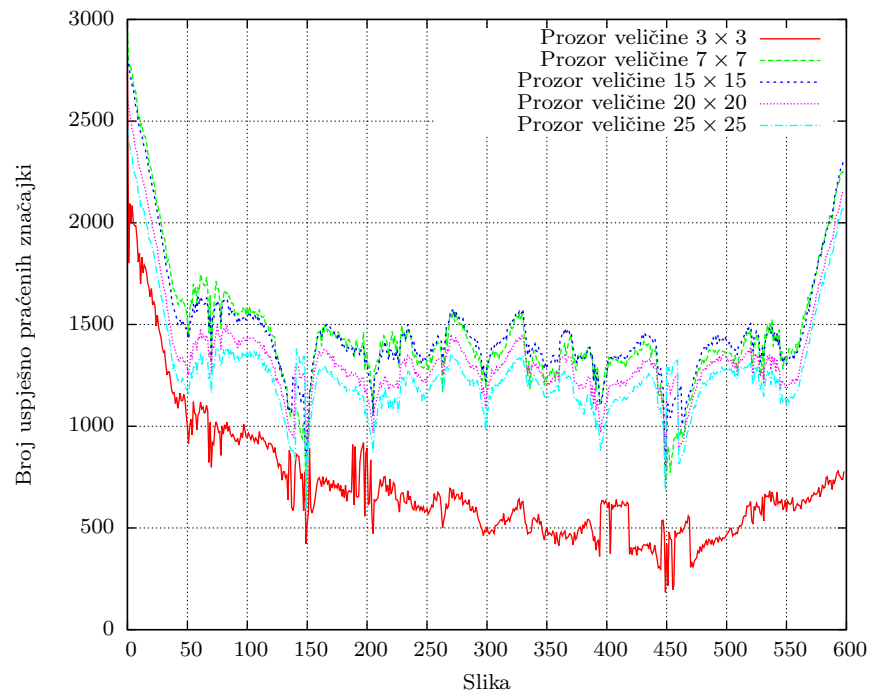
Slika 6.6: Broj uspješno praćenih značajki u ovisnosti o provjeri afine konzistentnosti.



Slika 6.7: Broj uspješno praćenih značajki u ovisnosti o početnom broju značajki.

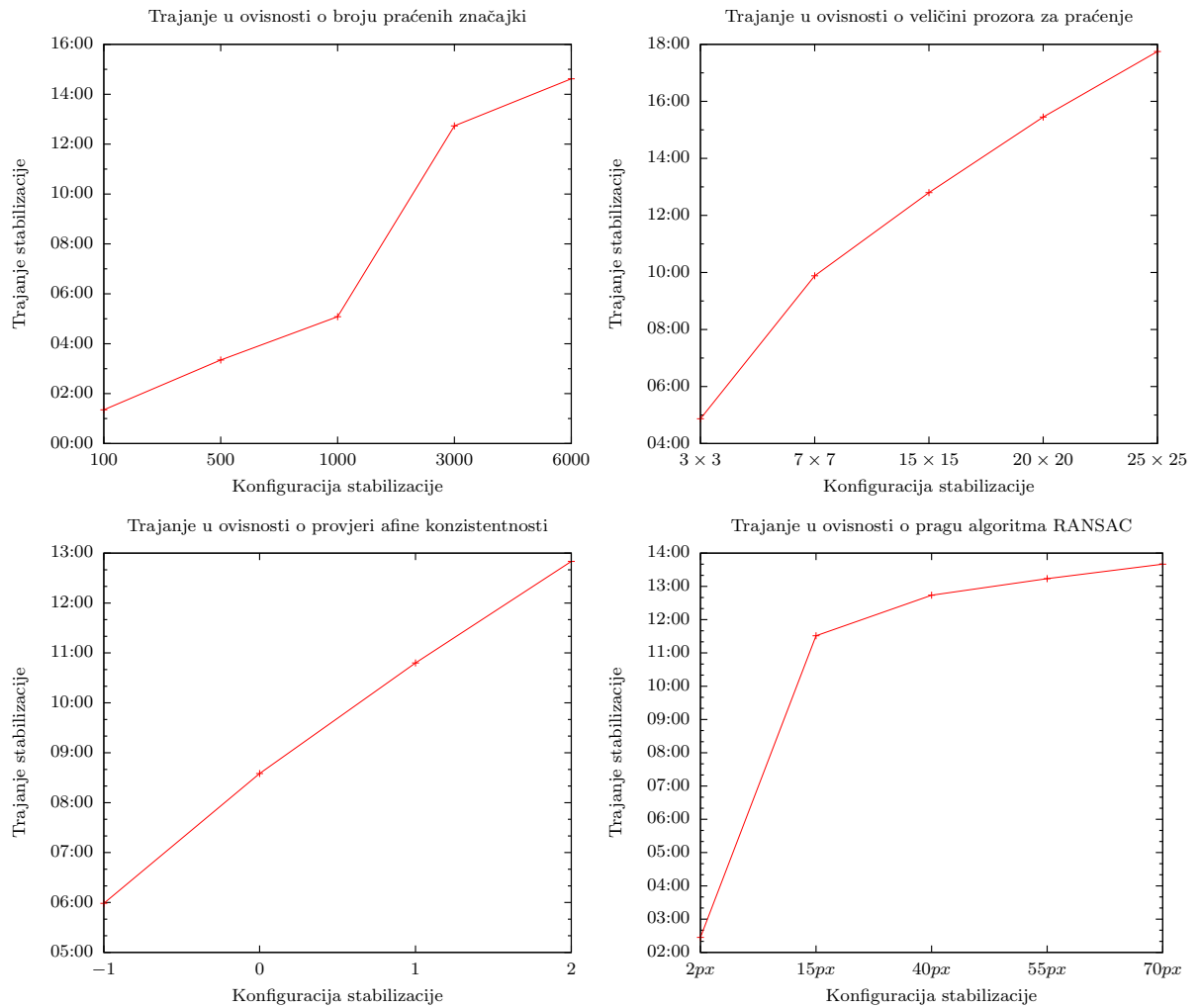


Slika 6.8: Broj uspješno praćenih značajki u ovisnosti o pragu algoritma RANSAC.



Slika 6.9: Broj uspješno praćenih značajki u ovisnosti o veličini prozora za praćenje.

Trajanje stabilizacije u ovisnosti o konfiguraciji



Slika 6.10: Brzina obrade u ovisnosti o konfiguraciji.



Slika 6.11: Na slici su prikazane točke outlieri (obojane zelenom bojom). Outlieri se nalaze u području automobila u pokretu.



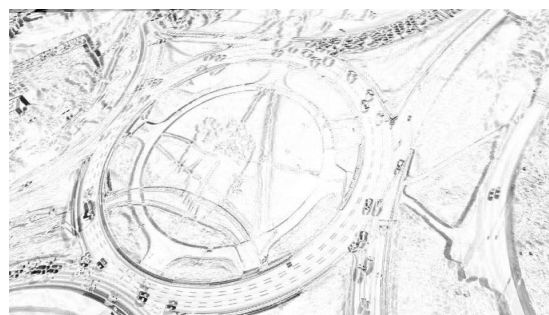
(a) Početna slika video sekvence u trajanju od 599 frameova.



(b) Posljedna, transformirana referentna slika video sekvence.



(c) Slika prikazuje razliku piksela među slikama 6.12a i 6.12b. Slici su izokrenute boje tako da najveću razliku prikazuju najtamniji pikseli.



(d) Slika prikazuje razliku piksela među referentnom i transformiranom referentnom slikom bez izbacivanja outliera. Slici su izokrenute boje tako da najveću razliku prikazuju najtamniji pikseli.

Slika 6.12: Prva i zadnja slika simetrične video sekvence od 599 slika. Zadnja slika je transformirana prva slika te su prikazane greške stabilizacije uz odbacivanje outliera i bez odbacivanja outliera.

7. Zaključak

U ovom radu predstavljen je sustav stabilizacije video zapisa prometnih raskrižja pribavljenih iz bespilotne lebdeće letjelice. Postupak stabilizacije temeljen je na algoritmu praćenja vizualnih značajki i estimaciji projekcijske transformacije ravnine s obzirom na početnu sliku video zapisa. Postignuta je dugotrajna stabilizacija video zapisa uz relativno nizak stupanj pogreške u stabilizaciji. Video zapisi su uspješno stabilizirani ukoliko kamera letjelice nastoji snimati jednu lokaciju. Postignuti rezultati su potencijalno korisni za daljnju obradu video zapisa u svrhu dobivanja informacija o toku prometa te se mogu koristiti u sustavima za nadzor prometa. Potrebno je dodatno nadograditi sustav kako bi se osigurala kontinuirana stabilizacija uz izmjenu referentnih slika video zapisa.

Rezultati ovog rada pokazuju mogućnosti za nastavak rada na sustavu. U nastavku rada potrebno je pronaći i otkloniti uzroke pogreške u stabilizaciji. U tu svrhu mogu se ispitati druge metode praćenja osim metode Kanadea, Lucasa i Tomasija.

Uz navedeno, u budućem radu bilo bi korisno povećati brzinu obrade videa te modificirati metodu da podržava rad u stvarnom vremenu. Radi smanjenja složenosti implementacije, ovisnost o biblioteci *OpenCV* mogla bi se ukloniti (budući da samo služi izračunu homografije i primjeni projekcijske transformacije). Biblioteka *OpenCV* se, u ovom slučaju, može nadopuniti npr. bibliotekom *homest* (<http://www.ics.forth.gr/~lourakis/homest/>) ili vlastitom implementacijom. Radi jednostavnosti korištenja, trebalo bi doraditi rješenje da kao ulaz prima video datoteku te kao izlaz daje video datoteku (izbacivanje korištenja *ffmpeg* alata kao međukoraka za razdvajanje video zapisa na slike).

U budućem radu treba omogućiti kontinuirano stabiliziranje na dužem videu s dinamičnom promjenom scena. Korisno je omogućiti praćenje i nakon što se izgube sve početne značajke, što uključuje izmjenu referentnih slika. Nedostatak čestih izmjena referentnih slika jest akumuliranje pogreške stabilizacije.

Omogućavanje dugotrajne stabilizacije može se pokušati uporabom istovremene

lokalizacije i mapiranja (engl. *Simultaneous localization and mapping*, SLAM). SLAM je tehnika gradnje mape nepoznatog područja s istovremenim čuvanjem informacije o trenutnom položaju automatiziranog vozila [5]. Mapiranje predstavlja problem organiziranja senzorskih informacija u konzistentni model, odnosno prikaz svijeta. S druge strane, lokalizacija predstavlja problem procjene položaja automatiziranog vozila u odnosu na mapu. Lokalizacija se može ostvariti postupcima praćenja vizualnih značajki. Kako bi se ostvarila istovremena lokalizacija i mapiranje, potrebno je poznavati gibanje vozila. Problem simultane lokalizacije i mapiranja je kumulativno povećanje pogreški u modelu svijeta. Pogreška se akumulira zbog dodavanja novih značajki u model koji već sadrži određenu razinu greške poteklu od nepreciznosti mjernih instrumenata te šuma u podacima. Greška se može kompenzirati različitim tehnikama, kao što je prepoznavanje prethodno nađenih značajki. Neke od korištenih tehnika uključuju Kalmanove filtere, filtere čestica i Monte Carlo metode.

LITERATURA

- [1] S. Baker i I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.
- [2] A. Bosco, A. Bruna, S. Battiato, G. Bella, i G. Puglisi. Digital video stabilization through curve warping techniques. *Consumer Electronics, IEEE Transactions on*, 54(2):220–224, 2008.
- [3] M. Brown, R. Szeliski, i S. Winder. Multi-image matching using multi-scale oriented patches. U *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, svezak 1, stranice 510–517. IEEE, 2005.
- [4] A. Diosi, S. Šegvić, A. Remazeilles, i F. Chaumette. Experimental evaluation of autonomous driving based on visual memory and image-based visual servoing. *Intelligent Transportation Systems, IEEE Transactions on*, (99): 1–14, 2011.
- [5] H. Durrant-Whyte i T. Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.
- [6] D.A. Forsyth i J. Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.
- [7] C. Harris i M. Stephens. A combined corner and edge detector. U *Alvey vision conference*, svezak 15, stranica 50. Manchester, UK, 1988.
- [8] R. Hartley, A. Zisserman, i Inc ebrary. *Multiple view geometry in computer vision*, svezak 2. Cambridge Univ Press, 2003.
- [9] R. Kumar, H.S. Sawhney, J.C. Asmuth, A. Pope, i S. Hsu. Registration of video to geo-referenced imagery. U *Pattern Recognition, 1998. Proceedings*.

- Fourteenth International Conference on*, svezak 2, stranice 1393–1400. IEEE, 1998.
- [10] M.I.A. Lourakis. A brief description of the levenberg-marquardt algorithm implemented by levmar. *Institute of Computer Science, Foundation for Research and Technology*, 11, 2005.
 - [11] B.D. Lucas i T. Kanade. An iterative image registration technique with an application to stereo vision. U *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.
 - [12] Y. Ma. *An invitation to 3-d vision: from images to geometric models*, svezak 26. Springer Verlag, 2004.
 - [13] S.J.D. Prince. *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.
 - [14] A.C. Shastry i R.A. Schowengerdt. Airborne video registration and traffic-flow parameter estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 6(4):391–405, 2005.
 - [15] C. Shekhar. Semi-automatic video-to-site registration for aerial monitoring. U *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, svezak 4, stranice 736–739. IEEE, 2000.
 - [16] J. Shi i C. Tomasi. Good features to track. U *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, stranice 593–600. IEEE, 1994.
 - [17] D.G. Sim i R.H. Park. Localization based on dem matching using multiple aerial image pairs. *Image Processing, IEEE Transactions on*, 11(1):52–55, 2002.
 - [18] G. Strang. Linear algebra and its applications, thomson learning. *Inc. USA*, 1988.
 - [19] R. Szeliski. *Computer vision: algorithms and applications*. Springer-Verlag New York Inc, 2010.
 - [20] C. Tomasi i T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ., 1991.

- [21] B. Triggs. Detecting keypoints with stable position, orientation, and scale under illumination changes. *Computer Vision-ECCV 2004*, stranice 100–113, 2004.
- [22] S. Šegvić i Z. Kalafatić. Dinamička analiza 3d scena.

Stabiliziranje video zapisa pribavljenog iz lebdeće letjelice

Sažetak

Video nadzorom prometnih raskrižja iz bespilotne letjelice moguće je dobiti informacije o toku prometa. Međutim, takav video zapis nije prikladan za izravnu analizu primjenom modeliranja pozadine zbog gibanja letjelice, pa je prvo potrebno ostvariti stabilizaciju pribavljenog video zapisa. U sklopu ovog rada istražen je jedan pristup stabilizaciji video zapisa pribavljenih lebdećom letjelicom.

Istraženi postupak stabilizacije temelji se na algoritmu praćenja značajki Kanadea, Lucasa i Tomasija te estimaciji homografije s obzirom na početnu sliku video zapisa. Rad opisuje korišteni postupak praćenja, estimaciju homografije, transformaciju slika video zapisa u ravninu početne slike, te postupak pronalaska izgubljenih značajki kojim osiguravamo dugotrajnu stabilizaciju.

Sustav je evaluiran nad snimkama prometnih raskrižja, te je dan prikaz rezultata. Postignuta je dugotrajna stabilizacija video zapisa s obzirom na referentnu sliku uz malo odstupanje od ravnine referentne slike. Stabilizacija je uspješna ukoliko gibanje letjelice nije veliko, odnosno kamera nastoji snimati lokaciju početne slike video zapisa.

Ključne riječi: Stabilizacija videa, KLT, praćenje značajki, homografija

Video stabilization for unmanned aerial vehicles

Abstract

Video surveillance of traffic intersections acquired by unmanned aerial vehicles (UAV) gives us valuable traffic flow information. However, such videos are not suitable for direct analysis by using background modeling due to vehicle motion, so we first need to stabilize given video. This work explores one approach to video stabilization system unmanned aerial vehicles.

The process of stabilization is based on Kanade-Lucas-Tomasi feature tracker and estimation of homography with respect to reference video image. This work describes used method of tracking, homography estimation and image warping, and also recovering lost features which enables long-term video stabilization.

The system is evaluated on traffic intersection videos, and review of results is given. Long-term video stabilization with respect to reference video image is achieved, with little deviation from reference image plane. Stabilization is successful when camera tries to film location of reference image.

Keywords: Video stabilization, KLT, feature tracking, homography