

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

**Stvaranje panorame iz više slika
bazirano na SIFT opisnicima
značajki**

Petar Palašek

Voditelj: *Doc. dr. sc. Siniša Šegvić*

Zagreb, svibanj 2011.

SADRŽAJ

1. Uvod	1
2. SIFT (Scale-Invariant Feature Transform)	2
2.1. Postupak pronalaženja SIFT značajki u slici	2
2.1.1. Detekcija ekstrema u prostoru mjerila	2
2.1.2. Lokalizacija ključnih točaka	4
2.1.3. Dodjeljivanje orijentacija ključnim točkama	5
2.1.4. Opisnici ključnih točaka	6
2.2. Pronalaženje podudarajućih značajki u slikama	7
3. Homografija	8
3.1. Preslikavanje točaka između dva različita pogleda	8
4. Spajanje slika	11
4.1. Poravnavanje slika	11
4.1.1. Biranje modela pokreta	12
4.1.2. Direktno poravnavanje slika (bazirano na pikselima)	12
4.1.3. Poravnavanje slika bazirano na značajkama	13
4.1.4. Biranje podloge za rezultatnu sliku	14
4.2. Određivanje složenih piksela rezultatne slike	14
5. Spajanje videa iz različitih pogleda	16
6. Eksperimentalni rezultati	18
6.1. Autostich program	18
7. Zaključak	20
8. Literatura	21

1. Uvod

Algoritmi za poravnavanje i spajanje slika u mozaike bez vidljivih rubova spajanja jedni su od najstarijih i najčešće korištenih algoritama u računalnom vidu [3]. Neki od primjera gdje koristimo spajanje slika su stvaranje digitalnih karata i satelitskih snimaka visoke rezolucije. Današnje moderne digitalne kamere i fotoaparati dolaze s već ugrađenom podrškom za spajanje slika, pa sada i obični korisnici mogu stvarati svoje vlastite panorame, bez potrebe za poznavanjem teorijskih osnova postupaka koji im to omogućavaju. Dostupni su komercijalni alati koji ne samo da stvaraju panorame iz više slika, već mogu prepoznati i više panorama u nesortiranom skupu slika. Primjer takvog programa je Autostitch, opisan u [1], koji je korišten za dobivanje eksperimentalnih rezultata u ovom radu. Spajanje statičkih slika može se proširiti i na spajanje videa, što je opisano u [4], gdje se spajanjem snimaka iz 3 različite kamere montirane na vozilu dobiva panoramski pogled na cestu.

U ovom radu opisano je spajanje više slika u panoramu korištenjem pristupa baziranog na SIFT značajkama koje se u posljednje vrijeme često koriste u računalnom vidu zbog svojih dobrih karakteristika [2].

Rad je strukturiran na sljedeći način. Najprije se u 2. poglavlju detaljno opisuje SIFT metoda, nakon čega se u 3. poglavlju daju osnove homografije. Zatim slijedi opis postupka spajanja slika u 4. poglavlju. U 5. poglavlju spajanje slika proširuje se na spajanje videa. U 6. poglavlju prikazani su neki eksperimentalni rezultati, nakon čega slijedi zaključak i očekivani završetak rada.

2. SIFT (Scale-Invariant Feature Transform)

SIFT (*Scale-Invariant Feature Transform*) je metoda koja služi za izlučivanje značajki invarijantnih na skaliranje i rotaciju iz slika. Pokazano je da su značajke dobivene ovom metodom vrlo robusne te da se mogu koristiti za traženje korespondentnih točaka u slikama čak i uz utjecaj afinih transformacija, promjenu 3D točke pogleda, dodatke šumova i djelomične promjene osvjetljenja. Značajke izlučene SIFT metodom vrlo su karakteristične te ih se, koristeći efikasne algoritme, u tipičnim slikama pronalazi velik broj.

SIFT je patentirana metoda, osmislio ju je David G. Lowe, a opisana je u članku [2] iz 2004. godine.

2.1. Postupak pronalaženja SIFT značajki u slici

Postupak pronalaženja SIFT značajki u slici dijelimo na 4 glavna koraka:

- Detekcija ekstrema u prostoru mjerila
- Lokalizacija ključnih točaka
- Dodjeljivanje orijentacija ključnim točkama
- Definiranje opisnika ključnih točaka

Svaki od navedenih koraka detaljno je objašnjen u nastavku ovog poglavlja.

2.1.1. Detekcija ekstrema u prostoru mjerila

Prostor mjerila slike definiramo kao funkciju $L(x, y, \sigma)$, koja predstavlja konvoluciju ulazne slike $I(x, y)$ s Gausovim filtrom $G(x, y, \sigma)$ s parametrom σ :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad (2.1)$$

gdje su x i y koordinate točke na slici, $*$ operator konvolucije, a Gaussov filtar dan izrazom

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}. \quad (2.2)$$

Kako bi se efikasno detektirale lokacije stabilnih ključnih točaka u prostoru mjerila, traže se ekstremi funkcije razlike Gaussovih funkcija (Difference-of-Gaussian, *DoG*) konvoluirane sa slikom, $D(x, y, \sigma)$, koja se može izračunati kao razlika dviju susjednih mjerila udaljenih za faktor k :

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \quad (2.3)$$

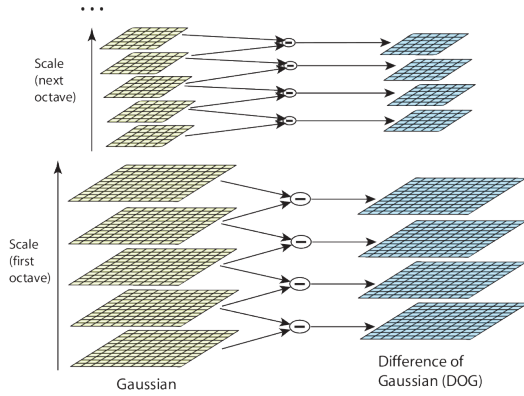
Postupak računanja $D(x, y, \sigma)$ možemo ilustrirati *DoG* piramidom, koja je prikazana na Slici 2.1. *DoG* piramida podijeljena je na tzv. oktave, od kojih se svaka sastoji od s cjelobrojnih intervala. Početna slika je uzastopce konvoluirana Gausovim filtrom s parametrom σ , pri čemu se σ u svakom koraku množi s konstantom k . Zbog takvog će postupka rezultirajuće slike u prostoru mjerila (prikazane u lijevom stupcu Slike 2.1) međusobno biti udaljene za faktor k . Dodatno želimo da početna slika u svakoj oktavi bude konvoluirana s duplo većim parametrom σ od one iz prošle oktave, a kako smo svaku oktavu podijelili na s intervala, faktor udaljenosti možemo izračunati po izrazu $k = 2^{\frac{1}{s}}$.

U svakoj oktavi se oduzimaju svake dvije susjedne slike iz lijevog stupca, te se tako dobivaju *DoG* slike prikazane u desnom stupcu. Zbog načina na koji se detektiraju ekstremi u *DoG* slikama, u svakoj oktavi mora se dobiti $s + 3$ konvoluiranih slika kako bi detekcija ekstrema pokrila cijelu oktavu.

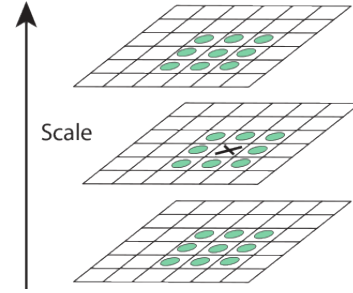
Početnu sliku u sljedećoj oktavi dobivamo na način da uzmemo 3. sliku odozgora iz trenutne oktave (ta slika imat će duplo veći parametar konvolucije σ od početne slike u trenutnoj oktavi) te je smanjimo dva puta.

Način detekcije lokalnih ekstrema $D(x, y, \sigma)$ prikazan je na Slici 2.2. Vrijednost svakog piksela označenog s X u *DoG* slici uspoređuje se s vrijednostima njegovih 8 susjeda i 18 susjeda iz susjednih slika. Ako je piksel X po vrijednosti veći od svih ili manji od svih promatranih susjednih piksela, onda on postaje potencijalna ključna točka. Filtriranje loših ključnih točaka iz skupa koji smo dobili ovim postupkom opisano je u sljedećem odjeljku.

Vrijednosti parametara s i σ koje su u eksperimentima [2] dale dobre rezultate su $s = 3$ i $\sigma = 1.6$, što znači da u svakoj oktavi računamo 6 konvoluiranih slika, uz $k = \sqrt[3]{2}$. Lowe također predlaže dodatno konvoluiranje i uvećavanje ulazne slike 2



Slika 2.1: Konstruiranje *DoG* piramide (preuzeto iz [2]).



Slika 2.2: Detektiranje ekstrema u *DoG* slikama (preuzeto iz [2]).

puta prije konstruiranja *DoG* piramide, jer se na taj način broj detektiranih stabilnih ključnih točaka povećava do 4 puta.

2.1.2. Lokalizacija ključnih točaka

Kako bi se poboljšala stabilnost odabranih točaka, iz skupa potencijalnih ključnih točaka određenih u prošlom koraku potrebno je izvaditi one lošeg kontrasta (jer su takve točke osjetljive na šumove) te one koje se nalaze na rubovima. Prije toga određujemo točne lokacije ekstrema razvijajući funkciju $D(x, y, \sigma)$ u Taylorov red (do drugog stupnja)

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}, \quad (2.4)$$

pri čemu se D i njezine derivacije računaju u trenutnoj točki iz ranije odabranog skupa, a $\mathbf{x} = (x, y, \sigma)$ označava pomak od te točke. Lokacija ekstrema $\hat{\mathbf{x}}$ određuje se tako da se funkcija D derivira po $\mathbf{x}$ i izjednači s nulom

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}}. \quad (2.5)$$

Ako je bilo koja komponenta $\hat{\mathbf{x}}$ veća od 0.5 znači da se stvarni ekstrem nalazi bliže nekoj drugoj točki iz skupa potencijalnih ključnih točaka, pa se interpolacija ponavlja oko te točke. Konačna vrijednost $\hat{\mathbf{x}}$ dodaje se točki oko koje se vršila interpolacija kako bi se dobila točna lokacija ekstrema.

Odbacivanje loših ključnih točaka

Za točke u kojima je vrijednost funkcije

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}} \quad (2.6)$$

manja od nekog praga (u eksperimentima u [2] korišteni prag je 0.03) kažemo da su lošeg kontrasta te ih odbacujemo.

Za poboljšanje stabilnosti nije dovoljno odbacivati samo ključne točke lošeg kontrasta. *DoG* funkcija ima jak odziv na rubovima, čak i kada je lokacija na rubu loše određena zbog čega su takve ključne točke nestabilne i želimo ih odbaciti. Takve ključne točke imat će veliku vrijednost gradijenta okomitog na rub i malu vrijednost gradijenta uz rub i možemo ih pronaći koristeći matricu $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}. \quad (2.7)$$

Derivacije se estimiraju računanjem razlika susjednih točaka. Označimo s α i β svojstvene vrijednosti $\mathbf{H}$, gdje je α veća vrijednost. Svojstvene vrijednosti matrice $\mathbf{H}$ proporcionalne su glavnim zakrivljenostima funkcije D . Sumu i produkt svojstvenih vrijednosti možemo izračunati iz:

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

Neka vrijedi $\alpha = r\beta$ i označimo s R omjer

$$R = \frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r}, \quad (2.8)$$

za koji vidimo da ovisi samo o omjeru svojstvenih vrijednosti, a ne i o njihovim vrijednostima. Na kraju, da bismo vidjeli trebamo li odbaciti neku ključnu točku potrebno je provjeriti vrijedi li

$$R < \frac{(r_{tr} + 1)^2}{r_{tr}}, \quad (2.9)$$

gdje r_{tr} predstavlja neku graničnu vrijednost. Ako gornja nejednakost ne vrijedi, ključna točka se odbacuje. Lowe u [2] za r_{tr} koristi vrijednost 10.

2.1.3. Dodjeljivanje orijentacija ključnim točkama

Kako bi se postigla invarijantnost značajki na rotaciju, potrebno je ključne točke opisati u ovisnosti na njihovu orijentaciju. Slijedi postupak za koji je u [2] utvrđeno da rezultira najstabilnijim značajkama.

Koristimo mjerilo ključne točke kako bi odabrali konvoluiranu sliku L u najbližem mjerilu, tako da su daljnji postupci invarijantni na skaliranje. Za svaki element slike

$L(x, y)$ izračunavaju se veličina $m(x, y)$ i orijentacija $\theta(x, y)$ gradijenta koristeći razlike susjednih elemenata:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.10)$$

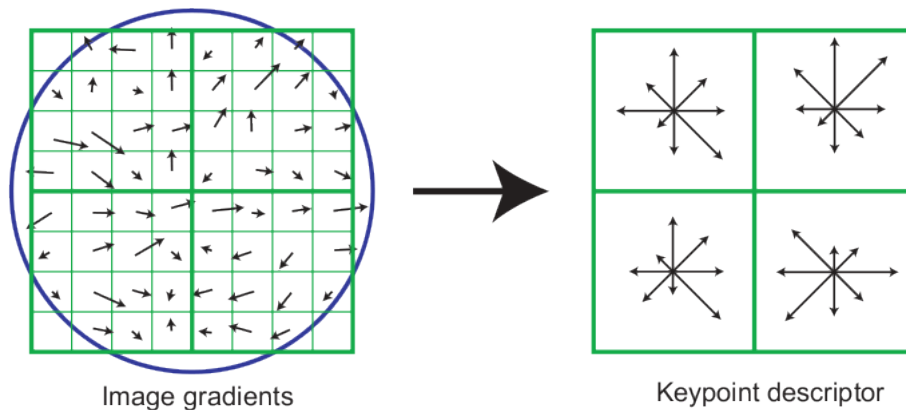
$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right). \quad (2.11)$$

Nakon toga, formulira se histogram orijentacija koristeći ranije izračunate orijentacije u određenom području oko ključne točke. Histogram je podijeljen na 36 skupova, od kojih svaki odgovara intervalu od 10° . Svaki element koji se dodaje u histogram množi se s odgovarajućom veličinom gradijenta. Uz to, svaki element skaliran je proporcionalno kružnom Gaussovom prozoru širine σ koji je 1.5 puta veći od mjerila ključne točke.

Vrhovi u histogramu odgovaraju dominantnim smjerovima lokalnih gradijenata. Najviši vrh i svaki vrh koji je u 80% vrijednosti najvišeg vrha generiraju ključnu točku s tom vrijednošću, odnosno, iz jedne ključne točke može nastati više njih na istoj lokaciji, ali s različitim orijentacijama. Kako bi se poboljšala točnost, za vrijednost orijentacije uzima se interpolirana vrijednost između odabranog i dva susjedna vrha u histogramu.

2.1.4. Opisnici ključnih točaka

U prijašnjim poglavljima opisano je kako se svakoj ključnoj točki dodjeljuju lokacija, mjerilo i orijentacija. Sljedeći korak je određivanje takvih opisnika ključnih točaka koji će biti lako razlikovani i dodatno invarijantni na promjene u osvjetljenju i promjeni 3D točke pogleda.



Slika 2.3: Računanje opisnika ključne točke.

Računanje opisnika ključne točke obavlja se tako da se najprije izračunaju vrijednosti i smjerovi gradijenata u svim točkama u određenoj okolini oko ključne točke, kao što je prikazano na slici 2.3 s lijeve strane. Nakon toga, vrijednosti svih gradijenata skaliramo pomoću kružnog Gaussovog prozora, prikazanog na slici plavom bojom, kako bi se smanjio utjecaj gradijenata koji su udaljeniji od centra. U sljedećem koraku okolina ključne točke dijeli se na više područja i za svako od tih područja radi se histogram orijentacija. Histogrami su podijeljeni na osam glavnih smjerova. U primjeru sa slike 2.3 promatrano područje oko ključne točke je dimenzija 8×8 elemenata, a podijeljeno je na 2×2 potpodručja. Kako se svaki od histograma sastoji od 8 smjerova, možemo ga predstaviti 8 dimenzionalnim vektorom. Iz toga slijedi da tako generirani opisnik možemo predstaviti 32 dimenzionalnim vektorom. Lowe u [2] promatra područje od 16×16 elemenata oko ključne točke i dijeli ih na 4×4 potpodručja. Takve deskriptore možemo predstaviti 128 dimenzionalnim vektorom kojeg normaliziramo s ciljem povećanja invarijantnosti na promjene osvjetljenja.

2.2. Pronalaženje podudarajućih značajki u slikama

Pretpostavimo da spajamo dvije slike. Nakon što smo SIFT-om izlučili značajke, potrebno je pronaći korespodencije, odnosno za svaku značajku iz prve slike provjeriti postoji li odgovarajuća značajka u drugoj slici.

Postupak za traženje korespondencija opisan u [1] i korišten u [4] iscrpnom pretragom računa euklidsku udaljenost između svakog mogućeg para opisnika iz prve i druge slike. Takav par proglašava se korespondentnim ukoliko im je udaljenost manja od 60% sljedeće najbolje udaljenosti. Drugim riječima, za svaku značajku iz prve slike pronalazi se najbliži i sljedeći najbliži susjed u drugoj slici. Ako je najbolja udaljenost manja od 60% druge najbolje udaljenosti, kažemo da se značajke podudaraju. Na taj način odbacuju se značajke koje ne čine par s nekom značajkom iz druge slike.

Kako bi se ubrzalo pronalaženje podudarajućih značajki Lowe u [2] predlaže korištenje aproksimativnog algoritma BBF (Best-Bin-First).

3. Homografija

Homografija ili ravninska projekcijska transformacija, kako se drugačije naziva, je vrsta 2D transformacije, odnosno, transformacije koja točke iz jedne ravnine preslikava u točke u drugoj ravnini.

Ako s $\tilde{x}$ označimo točku u ravnini prikazanu u homogenoj notaciji, onda ju primjenom homografije možemo preslikati u točku $\tilde{x}'$ i to zapisati kao

$$\tilde{x}' = \tilde{H}\tilde{x}. \quad (3.1)$$

Matrica $\tilde{H}$ iz gornjeg izraza je matrica dimenzija 3×3 kojom je predstavljena homografija. Dvije homografske matrice od kojih je jedna dobivena množenjem druge s nekim konstantnim faktorom su ekvivalente homografske matrice, odnosno, bitan je samo omjer elemenata u matrici. Zbog toga kažemo da je matrica $\tilde{H}$ homogena i da ima 8 stupnjeva slobode (postoji 8 različitih omjera u matrici). Kako bismo točku $\tilde{x}'$ prikazali u nehomogenom obliku, koristimo izraze

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \quad \text{i} \quad y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}. \quad (3.2)$$

Homografija čuva pravce, odnosno, pravci ostaju pravci i nakon transformacije [3].

3.1. Preslikavanje točaka između dva različita pogleda

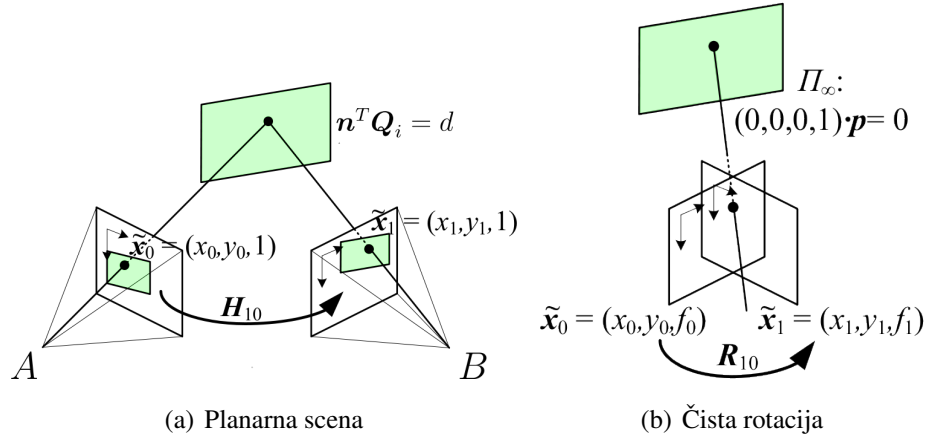
Pretpostavimo da je scena koju promatramo iz dva različita mjesta pogleda (A i B) planarna, kao što je prikazano na Slici 3.1(a). Ravnina Π u kojoj leži promatrana točka Q_i može se opisati jednadžbom $n^T Q_i = d$, odnosno $\frac{n^T Q_i}{d} = 1$.

U svakom slučaju točku Q_A , promatranu u svijetu iz mjesta pogleda kamere A , možemo preslikati u točku Q_B koristeći rotaciju i translaciju:

$$Q_B = RQ_A + t, \quad Q_A, Q_B \in \mathbb{R}^3. \quad (3.3)$$

Posebno, za točke Q_i koje pripadaju ravnini Π možemo pisati:

$$Q_{Bi} = RQ_{Ai} + t \cdot \frac{n^T Q_{Ai}}{d}, \quad (3.4)$$



Slika 3.1: Preslikavanje točka između dva različita pogleda (preuzeto iz [3]).

odnosno, nakon izlučivanja Q_{Ai} dobijemo:

$$Q_{Bi} = \left(R + \frac{t \cdot n^T}{d} \right) \cdot Q_{Ai} \quad (3.5)$$

Za točku $\hat{q}_{Bi}$ u normaliziranoj slikovnoj ravnini kamere B možemo pisati:

$$\hat{q}_{Bi} \sim Q_{Bi}. \quad (3.6)$$

Analogno za točku $\hat{q}_{Ai}$ vrijedi:

$$\hat{q}_{Ai} \sim Q_{Ai}. \quad (3.7)$$

Uvrštavanjem 3.5 u 3.6 dobijemo:

$$\hat{q}_{Bi} \sim \left(R + \frac{t \cdot n^T}{d} \right) \cdot Q_{Ai}, \quad (3.8)$$

odnosno, nakon uvršavanja 3.7:

$$\hat{q}_{Bi} \sim \left(R + \frac{t \cdot n^T}{d} \right) \cdot \hat{q}_{Ai}. \quad (3.9)$$

Izraz u zagradi u 3.9 zapravo predstavlja homografijsku matricu, pa preslikavanje točke iz jednog pogleda u drugi možemo opisati kao:

$$\hat{q}_{Bi} \sim \hat{H}_{10} \hat{q}_{Ai}, \quad (3.10)$$

ili

$$q_{Bi} \sim K_B H_{10} K_A^{-1} q_{Ai}, \quad (3.11)$$

u slučaju kada točke ne gledamo u normaliziranoj slikovnoj ravnini. Matrice K_A i K_B su kalibracijske matrice kamera.

Zanimljivo je primijetiti promjene u izrazu 3.9 kada udaljenost promatrane ravnine od točke gledišta, d , teži u beskonačnost ili kad vektor $\mathbf{t}$ teži nul vektoru, tj. kada nema translacije. U oba slučaja drugi član zagrade u izrazu teži nuli, te se izraz svodi na:

$$\hat{\mathbf{q}}_{Bi} \sim \hat{\mathbf{R}}_{10} \cdot \hat{\mathbf{q}}_{Ai}, \quad (3.12)$$

odnosno, na čistu rotaciju. Ovaj slučaj je tipičan kod stvaranja panorama i ilustriran je na Slici 3.1(b). Kada promatrane točke nisu iz normalizirane slikovne ravnine, možemo pisati:

$$\mathbf{q}_{Bi} \sim \mathbf{K}_B \mathbf{R} \mathbf{K}_A^{-1} \mathbf{q}_{Ai}. \quad (3.13)$$

4. Spajanje slika

Proces spajanja slika možemo podijeliti na dva glavna koraka: I) poravnavanje slika i II) određivanje složenih piksela rezultatne slike. Oba koraka detaljno su opisana u nastavku.



Slika 4.1: Primjer panografije nastale iz 45 fotografija¹.

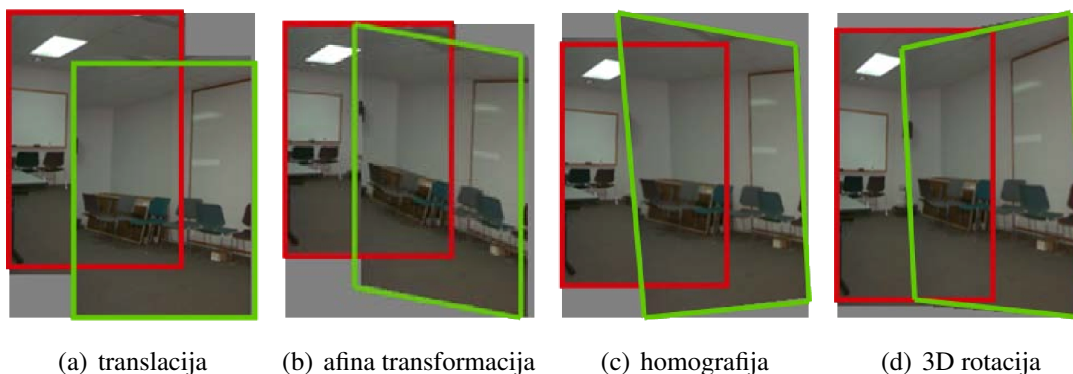
4.1. Poravnavanje slika

Prije nego što možemo krenuti s poravnavanjem slika, potrebno je odabrati prikladan model pokreta (*engl. motion model*) koji opisuje poravnavanje dviju slika koje želimo spojiti. Nakon što smo odabrali model pokreta, potrebno je na neki način odrediti parametre tog modela. Određivanje parametara može se obaviti direktnim pristupom ili pristupom baziranom na značajkama. Biranje modela pokreta i detalji oba pristupa poravnavanja opisani su u nastavku.

¹Slika preuzeta s <http://www.flickr.com/photos/25813477@N07/4579048427/>

4.1.1. Biranje modela pokreta

Biranjem modela pokreta želimo uspostaviti matematičke odnose koji preslikavaju koordinate elemenata jedne slike u koordinate u drugoj slici. Postoje razne vrste takvih parametarskih modela pokreta (*engl. parametric motion models*): jednostavne 2D transformacije, planarni perspektivni modeli, 3D rotacije kamere, izobličenje leće i preslikavanje na neplanarne (npr. cilindrične) površine [3].



Slika 4.2: 2D modeli pokreta i kako se koriste kod spajanja slika (preuzeto iz [3]).

Najjednostavniji model pokreta jest onaj koji kod poravnavanja koristi samo 2D translaciju i rotaciju slika, prikazan na Slici 4.2(a). Ovakvo poravnavanje slika jednako je onom koje bismo koristili da pokušamo poravnati razvijene fotografije postavljene na stolu. Slike nastale na taj način, s namjerno neispravljenim nedosljednostima radi umjetničkog efekta nazivaju se panografije. Primjer panografije nastale iz 45 fotografija prikazan je na Slici 4.1. Translaciju i rotaciju također možemo koristiti kao modele pokreta kad želimo stabilizirati sliku ili video ako su u pitanju manji pokreti.

U poglavlju 3.1. opisali smo preslikavanje točaka između dviju različito pozicioniranih kamera kada je scena koja se promatra planarna. Takvo preslikavanje opisano je pomoću homografije 3.10.

U istom poglavlju također smo opisali slučaj tipičan kod problema spajanja slika u panoramu, kada su pogledi međusobno rotirani, tj. kada kameru kojom slikamo samo rotiramo. Ovo preslikavanje opisali smo izrazom 3.12. Model čiste rotacije najčešće je korišten model kod spajanja slika.

4.1.2. Direktno poravnavanje slika (bazirano na pikselima)

Direktno poravnavanje radi tako da se slike koje poravnavamo preklope na onaj način koji minimizira različitosti između piksela tih slika. Kod direktnog poravnavanja

potrebno je najprije odrediti mjeru pogreške pomoću koje će se ocjenjivati odabrano preklapanje te odabrati metodu pretraživanja. Najjednostavnija metoda pretraživanja je potpuno pretraživanje, kojim se isprobavaju sva moguća poravnavanja kako bi se pronašlo najbolje. Kako takav pristup može biti vrlo spor, mogu se koristiti i druge metode kao npr. hijerarhijska *coarse-to-fine* metoda bazirana na slikovnim piramidama.

Direktan pristup se u pravilu može koristiti za poravnavanje susjednih okvira u videu, no, kod problema gdje se slike koje trebamo poravnati samo djelomično preklapaju (npr. panorame), takav pristup neće uvijek dati dobre rezultate. Kod takvih je problema bolje koristiti poravnavanje bazirano na značajkama.

4.1.3. Poravnavanje slika bazirano na značajkama

Rane metode poravnavanja bazirane na značajkama imale su probleme sa slikama koje su bile premalo ili previše teksturirane. Značajke bi često bile nejednoliko razdijeljene po slici zbog čega uparivanje slika koje je trebalo poravnati ne bi radilo.

Današnje metode su vrlo robusne te Szeliski u [3] piše da se koristeći pristupe bazirane na značajkama može pronaći dovoljan broj korespondencija potreban za spajanje slika ukoliko su značajke jednoliko raspodijeljene po slici i njihovi deskriptori dizajnirani imajući ponovljivost na umu. SIFT značajke opisane ranije u 2. poglavlju imaju baš te tražene karakteristike.

Najveća prednost poravnavanja slika baziranog na značajkama je to što omogućava automatsko prepoznavanje panorama u nesortiranom skupu slika [1], što je ponašanje kakvo običan korisnik priželjkuje.

Procjenjivanje homografije

Nakon što smo pronašli korespondentne značajke u slikama na način opisan u 2.2. poglavlju, potrebno je procijeniti parametre odabranog modela pogleda. U našem slučaju, potrebno je procijeniti homografiju. Procjenjivanje homografije opisano u [4] može se opisati u 3 koraka:

- odbacivanje *outliera* pomoću RANSAC algoritma (za generiranje hipotetne homografije potrebno je minimalno 4 korespondencije)
- reestimacija linearnim algoritmom
- iterativno poboljšanje 9 elemenata homografske matrice gradijentnom optimizacijom

4.1.4. Biranje podloge za rezultatnu sliku

Sljedeći korak kod spajanja slika jest biranje podloge za rezultatnu sliku. Ako spajamo manji broj slika možemo jednu od njih proglasiti referentnom i sve ostale slike preslikati u njezin koordinatni sustav. Takav pristup će davati dobre rezultate ukoliko vidno polje ne obuhvaća više od 90° , dok za šira vidna polja rezultatna slika postaje previše iskrivljena. Za stvaranje većih panorama potrebno je koristiti cilindričnu ili sferičnu projekciju [3]. Naravno, moguće je koristiti i druge projekcije, ali ove dvije navedene su najčešće korištene.

Nakon odabira projekcije potrebno je odrediti koja od slika će biti centrirana u panorami. Kod pristupa opisanog za mali broj slika centrirana slika bit će ona koju smo odabrali za referentu, dok npr. kod rotacijskih panorama možemo centrirati onu sliku kojoj je z koordinata najbliža prosječnoj z koordinati.

Na kraju je potrebno odrediti preslikavanje između koordinata piksela slika i panorame. Ukoliko je odabrana podloga ravnina, preslikavanje se može obaviti pomoću homografije. Ako pak se radi o cilindričnoj ili sferičnoj plohi, sve piksele u panorami pretvaramo u 3D točke te ih preslikavamo natrag u svaku sliku prema jednadžbama projekcije.

4.2. Određivanje složenih piksela rezultatne slike

Nakon što smo uspjeli poravnati slike koje želimo spojiti, moramo odrediti vrijednosti složenih piksela rezultatne slike. Jedan od načina na koji to možemo napraviti jest računanjem težinskog prosjeka piksela slika koje se preklapaju. Znači, vrijednost piksela rezultatne slike postaviti ćemo na prosjek vrijednosti piksela preklapljenih slika koji se preslikavaju u taj piksel rezultatne slike, ili matematički rečeno:

$$q_C(\mathbf{x}) = \sum_k \frac{\alpha_k(\mathbf{x}) \cdot q_k(T_k(\mathbf{x}))}{\sum_l \alpha_l(\mathbf{x})}, \forall k, \mathbf{x} \quad (4.1)$$

Želimo da pikseli koji su izvan originalne slike ne pridonose vrijednosti složenog piksela, pa u takvim slučajevima vrijednost $\alpha_k(\mathbf{x})$ postavljamo na 0:

$$\alpha_k(\mathbf{x}) = 0, \forall k, \mathbf{x} : T_k(\mathbf{x}) \notin \mathbf{I}_k. \quad (4.2)$$

Ako pak piksel leži u originalnoj slici, vrijednost $\alpha_k(\mathbf{x})$ možemo postaviti na 1 i time postići da vrijednost složenog piksela bude prosjek vrijednosti piksela na toj lokaciji u drugim slikama:

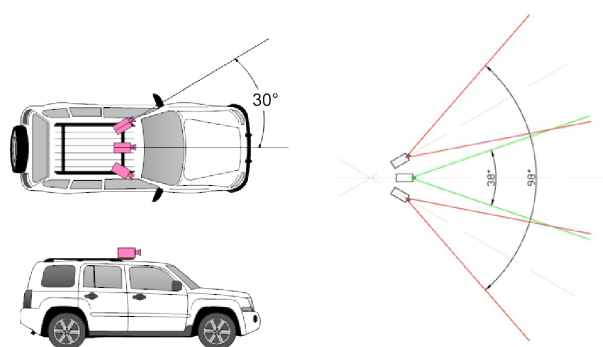
$$\alpha_k(\mathbf{x}) = 1, \forall k, \mathbf{x} : T_k(\mathbf{x}) \in \mathbf{I}_k. \quad (4.3)$$

Postoji i drugi pristup kod kojeg je vrijednost $\alpha_k(\mathbf{x})$ proporcionalna udaljenosti piksela od ruba slika, $d_B(T_k(\mathbf{x}))$:

$$\alpha_k(\mathbf{x}) = d_B(T_k(\mathbf{x})), \forall k, \mathbf{x} : T_k(\mathbf{x}) \in \mathbf{I}_k. \quad (4.4)$$

5. Spajanje videa iz različitih pogleda

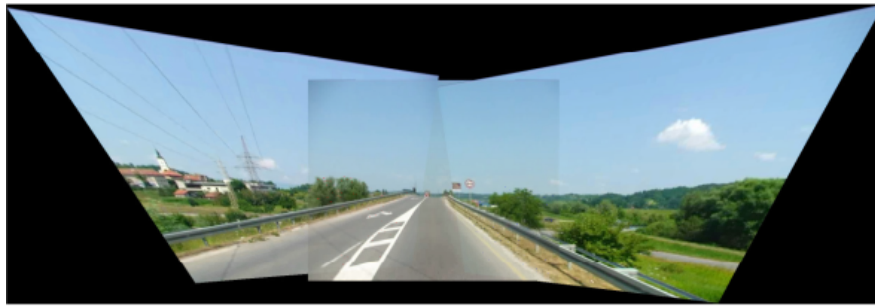
Budući da je video zapravo niz slika koje se izmjenjuju kroz vrijeme, problem spajanja videa iz različitih pogleda svodi se na spajanje slika i možemo ga riješiti koristeći postupke opisane u ovom radu. U članku [4] prikazana je primjena spajanja slika u svrhu spajanja videa iz 3 različito pozicionirane kamere. Cilj tog rada bio je olakšati očitavanje stanja ceste na način da osoba koja je za to zadužena gleda samo jedan video umjesto 3 različita kako se to obavljalo do sada. Konfiguracija kamera korištenih za pribavljanje video snimaka pomoću vozila prikazana je na Slici 5.1.



Slika 5.1: Prikaz konfiguracije kamera korištenih za pribavljanje video snimaka stanja ceste (preuzeto iz [4]).

U spomenutom radu isprobana su 2 modela pokreta: ravninska projekcijska transformacija i 3D rotacija. Također su isprobana dva pristupa određivanja složenih piksela rezultantne slike: računanje težinskog prosjeka i računanje težinskog prosjeka kod kojeg se u obzir uzimala udaljenost piksela od ruba slike. Razlike u pristupima mogu se vidjeti na Slici 5.2.

Panorame nastale korištenjem ravninske projekcijske transformacije bile su bolje geometrijski poravnate, ali su lošije izgledale. Razlog toga vjerojatno je razmak među kamerama koji ne bi trebao postojati kada se koristi taj model pokreta.



(a)



(b)

Slika 5.2: Prikaz eksperimentalnih rezultata iz [4]. Za određivanje vrijednosti složenih piksela u slikama pod a) korišteno je jednostavno računanje prosjeka vrijednosti piksela. Kod slika pod b) u obzir su uzete i udaljenosti piksela od rubova slike. Također su prikazane razlike korištenja različitih modela pokreta. Gornje slike u a) i b) dobivene su pomoću ravninske projekcijske transformacije, dok su donje dobivene korištenjem 3D rotacijskog modela. Slike su preuzete iz [4].

6. Eksperimentalni rezultati

U ovom poglavlju prikazani su eksperimentalni rezultati nastali korištenjem Autostitch programa.

6.1. Autostitch program

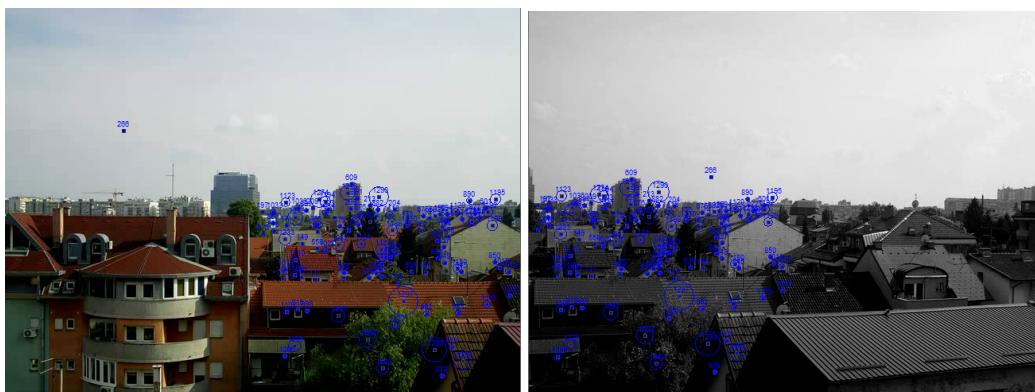
Autostitch je komercijalni program kojim je implementiran postupak automatske detekcije i stvaranja panorama opisan u [1]. Eksperimentalni rezultati prikazani u nastavku nastali su korištenjem demo verzije ovog programa, preuzete s adrese:

<http://cvlab.epfl.ch/brown/autostitch/autostitch.html>.

Također su prikazane korespondentne SIFT značajke jednog para slika, pribavljene pomoću algoritama implementiranih u *CVSH* ljusci.



Slika 6.1: Pet slika pogleda s balkona koje želimo spojiti u panoramu.



Slika 6.2: Dvije slike s označenim korespondentnim SIFT opisnicima značajki.



Slika 6.3: Panorama nastala spajanjem više slika korištenjem Autostitch programa.

7. Zaključak

U ovom su radu opisane osnovne ideje postupaka za spajanje slika u panorame bazirane na SIFT opisnicima značajki. SIFT opisnici značajki vrlo su robusni te se mogu koristiti za traženje korespondentnih točaka u slikama čak i uz utjecaj afinih transformacija, promjenu 3D točke pogleda, dodavanje šumova i djelomične promjene osvjetljenja.

Postoje različiti modeli pokreta koji se mogu koristiti kod spajanja slika u panoramu. Najčešće korišteni model pokreta jest model čiste rotacije, kada se pretpostavlja da se kamera okretala oko svoje osi kod stvaranja fotografija koje spajamo.

Prikazane su razlike između direktnog poravnavanja slika i poravnavanja slika baziranog na značajkama. Poravnavanje slika bazirano na značajkama robusnije je od direktnog poravnavanja, brže, te se može koristiti i kada se slike koje spajamo ne preklapaju velikim dijelom.

Za određivanje vrijednosti piksela rezultatne slike može se koristiti jednostavno računanje prosjeka piksela iz slika koje spajamo. Bolji rezultati mogu se postići uko-liko u obzir uzmemo i udaljenosti piksela od ruba slike.

Spajanje slika u panoramu može se proširiti i na spajanje videa snimljenih različito orijentiranim kamerama, budući da je video zapravo niz pokretnih slika.

8. Literatura

- [1] M. Brown i D. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, Vol. 74, No. 1, stranice 59–73, 2007.
- [2] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, stranice 91–110, 2004.
- [3] R. Szeliski. *Computer Vision: Algorithms and Applications, 1st Edition*. Springer, 2011.
- [4] S. Šegvić, M. Ševrović, G. Kos, V. Stanistavljević, i I. Dadić. Preliminary experiments in multi-view video stitching. Opatija, Croatia, *MIPRO*, May 2011.

Stvaranje panorame iz više slika bazirano na SIFT opisnicima značajki

Sažetak

U ovom radu opisuje se spajanje slika pribavljenih iz više pogleda u jednu panoramsku sliku. Opisuje se poravnanje slika bazirano na značajkama pribavljenim SIFT metodom. Detaljno je opisano pribavljanje SIFT opisnika korištenjem piramide razlika Gaussovih funkcija. Navode se različiti modeli pokreta koji se koriste kod spajanja slika. Ukratko je opisano spajanje videa u panoramu. Prikazuju se rezultati spajanja slika dobiveni korištenjem "Autostitch" programa.

Ključne riječi: računalni vid, SIFT, image stitching, spajanje slika, homografija, panorama