

Kreiranje dubinske mape korišćenjem stereo kamere

U ovom radu je vršeno kreiranje dubinske mape, korišćenjem stereo kamere. Upoređivane su dve metode, block-matching i PatchMatch, koje su testirane na standardnoj bazi (Scharstein et al. 2014). Dobijeni rezultati pokazuju da PatchMatch metod daje bolje rezultate, što se i očekivalo, budući da ovaj algoritam uvodi mogućnost da se ravan scene nalazi pod određenim uglom. Primena PatchMatch algoritma na slici Teddy iz referentne baze ostvaruje 22.8% pogrešnih piksela (piksela sa greškom većom od jedan u odnosu na referentnu mapu) i prosečnu grešku (prosek apsolutnog odstupanja od referentne mape) od 2.55 piksela. Primena block-matching algoritma na istoj slici ostvaruje 26.7% pogrešnih piksela i prosečnu grešku od 3.70 piksela. Algoritmi su upoređeni i na slikama Motorcycle, Piano i Playtable iz standardne baze. PatchMatch daje bolje rezultate po kriterijumu prosečne greške na svim slikama, dok po kriterijumu procenta pogrešnih piksela daje bolju metriku jedino na slici Playtable. Variranjem maksimalne razlike ispravnog piksela i piksela referentne mape, za vrednosti maksimalne razlike od 5 i 0.5 piksela, PatchMatch daje bolju metriku po kriterijumu procenta pogrešnih piksela na svim slikama. PatchMatch algoritam se u odnosu na referentne algoritme (49 algoritama) standardne baze pozicionira na 20. mestu po kriterijumu procenta pogrešnih piksela i na 7. mestu po kriterijumu prosečne greške.

Uvod

Stereo vizija predstavlja proces rekonstrukcije informacija o dubini iz para slika koje su snimljene u istom trenutku pomoću dve horizontalno razmaknute kamere. Stereo vizija se bazira na određivanju pomeraja (disparity) u položaju projekcija objekata scene na dve slike. Informacije o dubini mogu se direktno odrediti iz pomeraja (Szeliski 2010).

Postoji više načina za određivanje pomeraja. Mogu se podeliti u dve velike grupe – lokalne i globalne metode. Principi koje podržavaju lokalne metode svode se na određivanje pomeraja svakog piksela slike na osnovu piksela u okolini. Globalne metode formulišu problem u zavisnosti od funkcije energije koja se pritom optimizuje. Pomeraji svih piksela se tokom optimizacije energije paralelno procenjuju. Neki od primera globalnih metoda su graph cuts (Boykov et al. 2001) i belief propagation (Boykov et al. 2001).

U ovom radu poredi se različite lokalne metode – block-matching (Je i Park 2013) i PatchMatch (Bleyer et al. 2011). Lokalne metode se zasnivaju na poređenju prozora na levoj i desnoj slici i određivanju njihove sličnosti. Kriterijum sličnosti je cost funkcija čijom se minimizacijom dolazi do najbližnjih prozora. Razlika u pozicijama dva najbližnja prozora na levoj i desnoj slici predstavlja pomeraj.

Block-matching je algoritam koji poredi kvadratne prozore istih dimenzija na levoj i desnoj slici. Korišćenje ovakvih prozora uvodi aproksimaciju da se svi pikseli prozora nalaze na ravni paralelnoj kameri. Nasuprot navedenom algoritmu, PatchMatch vrši transformacije prozora, tako da se svi pikseli prozora nalaze na ravni koja

Danilo Tonić (1999), Kraljevo, Dušana Karapandžića 1k, učenik 2. razreda Gimnazije Kraljevo

Marko Skakun (1997), Sremska Mitrovica, Lačarak, 1. novembra 350, učenik 4. razreda Mitrovačke gimnazije

MENTOR: Damjan Dakić, student Elektrotehničkog fakulteta Univerziteta u Beogradu

ne mora biti paralelna kameri, već može stajati pod određenim uglom. Prilikom minimizacije cost funkcije, za razliku od block-matching algoritma kod kog se optimizuje samo pomeraj, u PatchMatch algoritmu optimizuju se i parametri ravni.

Krajnji rezultat je mapa pomeraja (disparity map) koja je predstavljena crno-belom slikom na kojoj je osvetljenje svakog piksela proporcionalno pomeraju. Po generisanju inicijalne mape pomeraja, vrši se naknadna obrada (post-processing) koja podrazumeva proveru leve i desne mape pomeraja sa markiranjem loše procenjenih piksela (consistency check), linearnu interpolaciju markiranih piksela i njihovo filtriranje.

Cilj našeg rada je poređenje dva opisana algoritma. Prvi kriterijum po kome se porede jeste procenat loših piksela, pri čemu se pod lošim pikselom podrazumeva piksel sa pomerajem čije je odstupanje od referentnih podataka priloženih u bazi veće od određene vrednosti. Drugi kriterijum za poređenje jeste prosečno apsolutno odstupanje mape pomeraja od referentne mape. Pomenuti kriterijumi su deo sistema za evaluaciju algoritama i priloženi su uz standardnu bazu (Scharstein *et al.* 2014) koja je korišćena za testiranje.

Može se pretpostaviti da će prema definisanim kriterijumima PatchMatch dati bolje rezultate, jer koristi prozore koji mogu da modeluju i površine koje se nalaze pod uglom u odnosu na kameru, za razliku od block-matching algoritma gde se uvodi aproksimacija da je sve što je unutar prozora paralelno kameri.

Metod

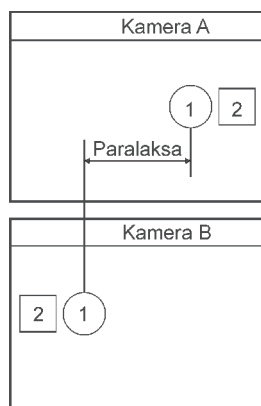
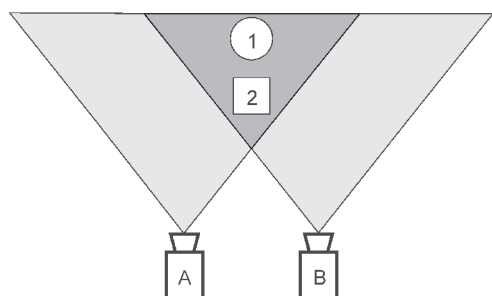
Kreiranje dubinske mape se zasniva na paralaksi (slika 1). Paralaksa predstavlja prividnu promenu položaja nekog objekta u odnosu na pozadinu. Što je objekat bliži posmatraču paralaksa, tj. pomeraj je veći. Prava koja prolazi kroz fokus i kroz bilo koju tačku na slici jedne kamere, na istoj toj kameri videće se samo kao jedna tačka, dok će se na drugoj kameri videti kao epipolarna linija. Samim tim, ako se neki objekat pomera po istoj toj pravoj, ostaće stacionaran na prvoj kameri, dok će se na drugoj kameri pomerati po epipolarnoj liniji. Najjednostavnije je vršiti estimaciju ako su sve epipolarne linije horizontalne, jer je paralaksa po vertikalnoj osi tada jednaka nuli.

Da bi epipolarne linije bile horizontalne, za početak potrebno je postaviti kamere sa samo horizontalnim pomakom, a zatim se kamere moraju kalibrisati, a slike potom ispraviti i poravnati na osnovu kalibracije. Kalibracija podrazumeva određivanje parametara sočiva i relativnog položaja kamera, na osnovu čega se ispravlja zakrivljenje, pomak sočiva i rotacija slike.

Jedan od problema koji se javlja u stereo viziji je okluzija. Okluzija je pojava u kojoj objekti koji su bliži kameri svojim pomerajem sakrivaju delove objekata u pozadini, i samim tim, neki objekti ostaju vidljivi samo na jednoj kameri.

Block-matching

Block-matching algoritam bazira se na upoređivanju kvadratnih prozora istih dimenzija

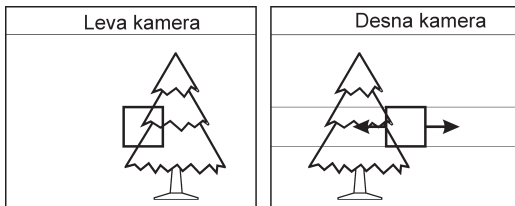


Slika 1.
Epipolarna geometrija

Figure 1.
Epipolar geometry

(slika 2). Za određeni piksel p jedne slike, uzima se prozor predefinisanih dimenzija tako da se piksel p nalazi u njegovom centru. Na drugoj slici, odabrani prozor poredi se sa prozorima čiji se centralni pikseli q nalaze na istoj horizontalnoj liniji (epipolarnoj liniji) kao odabrani piksel p . Smer u kome je dovoljno vršiti proveru određen je time da li proveru vršimo na desnoj ili levoj slici. Odnosno, ako se proveru vrši na levoj slici, dovoljno je proveriti prozore koji se nalaze desno od odabranog, i obrnuto. Za pomeraj piksela p biće uzeta razlika njegove pozicije i pozicije piksela q , tj. centralnog piksela čiji se prozor ispostavio za najbliži. Sličnost među prozorima definiše se vrednošću cost funkcije, pa tako najbliži prozori imaju minimalnu vrednost cost funkcije.

Kako su prozori koje de obrađuju kvadratni i istih dimenzija, svaki piksel prozora u cost funkciji utiče sa istom vrednošću pomeraja. Ista vrednost pomeraja svakog piksela u prozoru

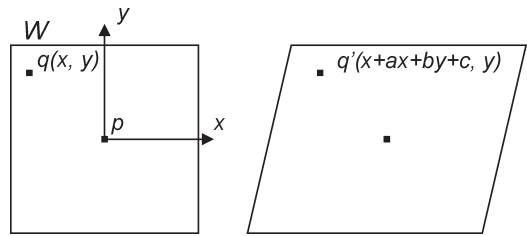


Slika 2. Upoređivanje blokova

Figure 2. Block-matching (left and right camera)

unosni aproksimaciju da se pikseli obuhvaćeni prozorom nalaze na istoj dubini, što podrazumeva da je površina koju obrazuju paralelna ravni objektivna kamere.

Za centralni piksel p prozora W koji se posmatra vezuje se lokalni koordinatni sistem takav da se početak nalazi u pikselu p (slika 3). Osa x je horizontalna, a osa y je vertikalna. Pikseli prozora W predstavljaju projekciju neke površi i u ovom radu te površi se aproksimiraju ravnima oblika $f(x, y) = ax + by + c$. Parametri a i b su određeni uglom pod kojim se ravan nalazi u odnosu na horizontalnu i vertikalnu osu, a c je pomeraj centralnog piksela, odnosno pomeraj kompletnog lokalnog koordinatnog sistema. Za



Slika 3. Prozor i njegov koordinatni sistem (levo) i projekcija prozora (desno)

Figure 3. Window and its coordinate system (left) and window projection (right)

određeni piksel $q(x, y)$ prozora W , na osnovu ravni f_p koja je vezana za centralni piksel p , određena je njegova projekcija q' u tački $(x + f_p(x, y), y)$ (slika 3).

Formula (1) definiše pomeraj sa kojim piksel q prozora sa koordinatama (x, y) utiče u cost funkciji, a na osnovu parametara ravni f_p za koju je vezan centralni piksel prozora. U block-matching metodi ravan f_p koju obrazuju pikseli prozora paralelna je objektivu kamere, odnosno definisana je parametrima $a_{fp} = b_{fp} = 0$:

$$\text{disp}_q = a_{fp} q_x + b_{fp} q_y + c_{fp} \quad (1)$$

Opisani metod usled svoje prirode nudi isključivo rezultate u celim brojevima (vrednosti pomeraja su celi brojevi).

Cost funkcija

Uloga cost funkcija jeste u tome da izmeri i brojno izrazi sličnost među prozorima. Manja vrednost cost funkcije znači da su dva prozora sličnija. Neki od primera cost funkcija jesu SAD (sum of absolut difference), SSD (sum of squared difference) i NCC (normalized cross correlation) (Patil *et al.* 2013). U ovom radu korišćena je cost funkcija predložena u PatchMatch (Bleyer *et al.* 2011).

Za određeni piksel p , procenjuje se da je na ravni f_p ako je vrednost cost funkcije za p i f_p minimalna:

$$f_p = \arg \min(\text{cost}(p, f)) \quad (2)$$

Cost funkciju za određeni piksel p i ravan f čine dva faktora:

$$\text{cost}(p, f) = \sum_{q \in W_p} w(p, q) \cdot u(q, q') \quad (3)$$

Prvi faktor predstavlja težina piksela koja definiše verovatnoću da se određeni piksel prozora q i centralni piksel p nalaze na istoj ravni:

$$w(p, q) = \exp \left(-\frac{\|I_p - I_q\|}{\gamma} \right) \quad (4)$$

Pretpostavlja se da će objekti na različitim dubinama imati različite boje, tj. da će $\|I_p - I_q\|$ biti veliko, gde je I_p boja piksela p , a I_q boja piksela q . Samim tim težina piksela zavisi od udaljenosti boja dva piksela u RGB prostoru. Formula težine data je u eksponencijalnom obliku, što zbog minusa u eksponentu znači da će za veću razliku u boji vrednost težine biti manja. Koeficijentom γ definiše se uticaj razlike u boji na težinu određenog piksela. Pikseli prozora q sa manjom težinom w će imati manji uticaj na vrednost cost funkcije. Opisani princip dovodi do boljeg izdvajanja objekata različite udaljenosti, odnosno do boljeg izdvajanja ivica na mapi pomeraja (Bleyer *et al.* 2011).

Drugi faktor ove cost funkcije svodi se na upoređivanje boje i gradijenata piksela q prozora jedne slike i piksela q' na drugoj slici sa kojim je svojim pomerajem uparen (5). Piksel q je uparen sa pikselom q' ako je $q' = q + \text{disp}_q$:

$$u(q, q') = (1 - \alpha) \cdot \min(\|I_q - I_{q'}\|, \tau_{\text{col}}) + \alpha \min(\|\nabla I_q - \nabla I_{q'}\|, \tau_{\text{grad}}) \quad (5)$$

Određeni piksel q prozora na jednoj slici poredi se sa odgovarajućim pikselom q' iz prozora druge slike. Razlika u boji $\|I_q - I_{q'}\|$ definisana je kao udaljenost boja u RGB prostoru, a razlika u gradijentu $\|\nabla I_q - \nabla I_{q'}\|$ kao apsolutna razlika horizontalnih gradijenata crno-bele slike. Radi postavljanja maksimuma cost funkcije, definišu se parametri τ_{col} i τ_{grad} kao maksimalne razlike među vrednostima boja i gradijenata koje se uzi-

maju u obzir. Faktor $u(q, q')$ računa se kao zbir razlike u boji i gradijentu, a uticaj vrednosti pomenutih razlika na zbir određen je koeficijentom α (5).

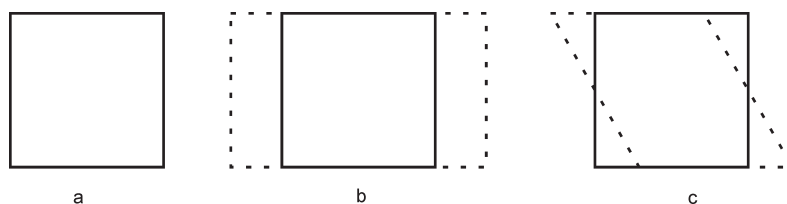
Kako pomeraj uzima vrednost iz skupa realnih brojeva, piksel q biće uparen sa tačkom koja se nalazi između dva susedna piksela, pa je potrebno interpolirati taj prostor radi definisanja vrednosti boje i gradijenta te tačke.

Vrednost cost funkcije, za piksel p i parametre ravni f_p , jednaka je sumi proizvoda dva faktora za svaki piksel prozora (3).

PatchMatch

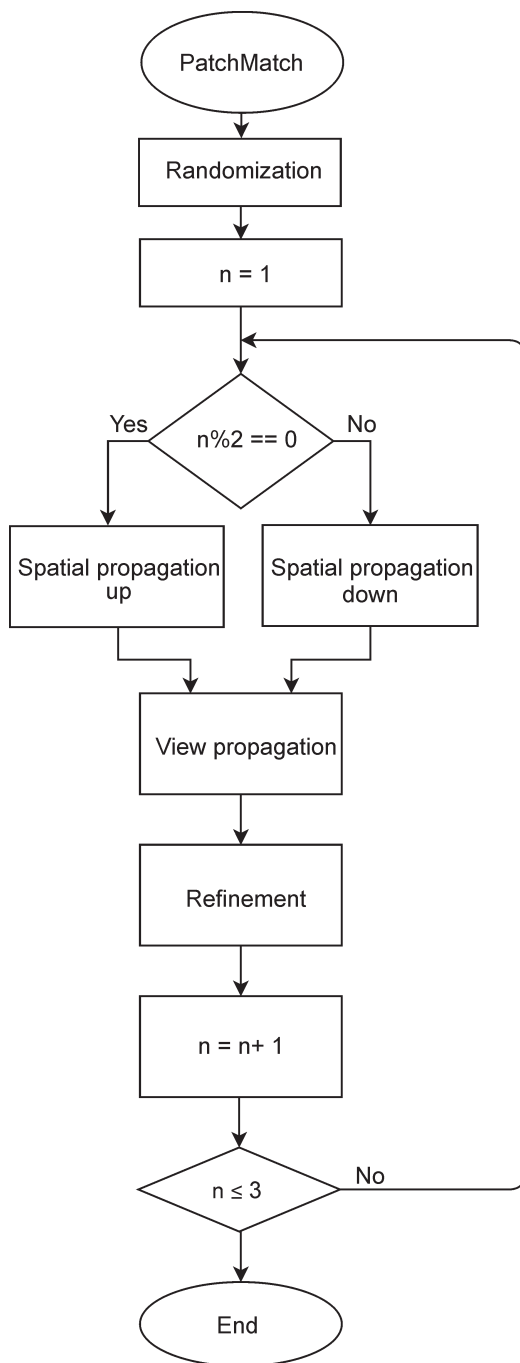
PatchMatch algoritam uvodi aproksimaciju da se svi pikseli istog prozora nalaze na istoj ravni, proizvoljnog položaja, za razliku od block matchinga koji uzima u obzir isključivo ravni paralelne objektivu kamere. Ovaj algoritam procenjuje parametre ravni kojoj piksel sa slike pripada. Menjanje parametara ravni se manifestuje promenom oblika prozora koja podrazumeva njegovo širenje i smicanje (slika 4). Širenje prozora odražava zakrenutost ravni po horizontalnoj osi i definiše se uglom čiji tangens predstavlja koeficijent širenja. Smaknutost odražava ugao ravni u odnosu na vertikalnu osu. Parametri ravni definišu parametre u formuli (1).

PatchMatch algoritam uvodi i aproksimaciju da se svi pikseli slike sigurno vide na drugoj slici, što podrazumeva da se ne nalaze u okluziji ili van slike. Nakon generisanja ravni, moramo biti sigurni da se deo slike za koji se generiše ravan, neće nalaziti u okluziji pri prelasku na drugu sliku, pa se tako pri estimaciji biraju one ravni koje su po horizontalnoj osi zakrenute tako da ih druga kamera sigurno može videti (nezavisno od udaljenosti kamera). Takođe, parametri ravni se biraju tako da nijedan piksel prozora pri prelasku na drugu sliku ne izlazi iz okvira slike.



Slika 4. Prozor (a), širenje prozora (b) i smicanje prozora (c)

Figure 4. Window (a), Window scaling (b) and Window shearing (c)



Slika 5. Blok šema PatchMatch algoritma

Figure 5. Block diagram of PatchMatch algorithm

Parametri ravni koji se optimizuju su pomeraj centralnog piksela i dva ugla ravni, u odnosu na horizontalnu odnosno vertikalnu osu, koji služe da bolje predstavljaju deo prostora koji obuhvata prozor, pa tako i pomognu boljem određivanju pomeraja centralnog piksela za koji se ravan generiše.

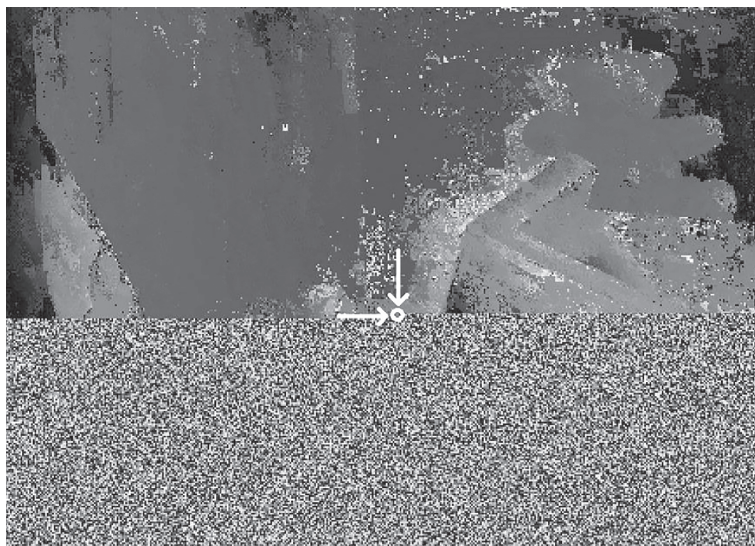
Pomeraj koji se formulom (1) izračunava jeste realan broj, pa tako pozicija piksela jedne slike na drugoj slici leži u kontinualnom prostoru. Ovaj koncept omogućava da preciznost ovog algoritma bude u okvirima manjim od jednog piksela (subpixel preciznost).

PatchMatch algoritam (slika 5) čine početna randomizacija parametara prozora i iteracije koje se sastoje iz tri faze: prostorna propagacija (spatial propagation), propagacija koja poredi mapu pomeraja koja odgovara levoj kameri i mapu pomeraja koja odgovara desnoj kameri (view propagation) i prečišćavanje (refinement). Ispostavlja se da su tri iteracije dovoljne za postizanje optimalnih rezultata (Bleyer *et al.* 2011), pa se zato u ovom radu vrše tri iteracije.

Početna randomizacija. Svaki piksel odlikuju tri parametra, uglovi koji definišu ravan na kojoj se piksel nalazi i njegov pomeraj. Na samom početku svi pikseli uzimaju nasumičnu vrednost za svaki parametar u njegovom dozvoljenom opsegu.

Prostorna propagacija. Faza prostorne propagacije upoređuje parametre ravni piksela sa parametrima ravni njemu susjednih piksela. Ukoliko parametri ravni susjednog piksela bolje odgovaraju pikselu za koji se poređenje vrši, onda se oni uzimaju za parametre ravni tog piksela. Ukoliko je q susjedni piksel piksela p i ukoliko važi $\text{cost}(p, f_q) < \text{cost}(p, f_p)$, ravan f_q postaće nova ravan piksela p .

Ideja ove faze oslanja se na činjenicu da je verovatno da će susjedni pikseli imati približno iste parametre. Zavisno od toga da li je iteracija parna ili ne, procesiranje se odvija na dva načina. U neparnim iteracijama, kroz sliku se prolazi u smeru desno dole (slika 6), počevši od gornjeg levog piksela i pritom isprobavajući parametre prvog gornjeg i levog piksela, dok se u parnim iteracijama odvija u smeru gore levo gore, sa početkom u donjem desnom pikselu, proveravajući pritom parametre prvog donjeg i desnog piksela.



Slika 6. Neparna iteracija

Figure 6. Even iteration

Propagacija koja poredi mape pomeraja sa obe kamere. Uloga ove faze je održavanje konzistentnosti između dve slike. Pretpostavlja se da trenutni piksel i pikseli druge slike, koji su svojim pomerajima sa njim upareni, imati slične parametre ravni. Za parametre trenutnog piksela uzimaju se parametri piksela koji na njega pokazuje, ukoliko daje manju vrednost cost funkcije za trenutni piksel. Ukoliko je piksel p' uparen sa pikselom p i ukoliko važi $\text{cost}(p, f_{p'}) < \text{cost}(p, f_p)$, ravan $f_{p'}$ postaje nova ravan piksela p .

Prečišćavanje. Za dalje redukovanje vrednosti cost funkcije koristi se sistem nasumičnog generisanja promene svakog parametara. Za svaki parametar generiše se vrednost promene u maksimalnom dozvoljenom opsegu. Početna faza podrazumeva veliki opseg u kome se promena može naći, što omogućuje velike promene vrednosti parametara. U narednom koraku, dozvoljeni opseg u kome se promena generiše deli se sa dva, ponovo iterira, i tako do trenutka kada dostigne zanemarljivu vrednost. Završna faza obraća pažnju na detalje i vrši blaga podešavanja parametara.

Maksimalni dozvoljeni opseg predstavlja polovinu opsega vrednosti ugla, odnosno pomeraja:

$$\max(\Delta a) = \frac{\max(\Delta a)}{2}, \max(\Delta b) = \frac{\max(\Delta b)}{2},$$

$$\max \Delta(\text{disp}) = \frac{\max(\Delta \text{disp})}{2} \quad (6)$$

Ako je ravan f'_p ravan sa parametrima:

$$a_{fp} + \Delta a, b_{fp} + \Delta b \text{ i } c_{fp} + \Delta \text{disp}$$

i ako je $\text{cost}(p, f'_p) < \text{cost}(p, f_p)$, ravan f'_p biće uzeta za novu ravan piksela p . Iteracije se vrše dok važi da je $\max(\Delta \text{disp}) \geq 1$.

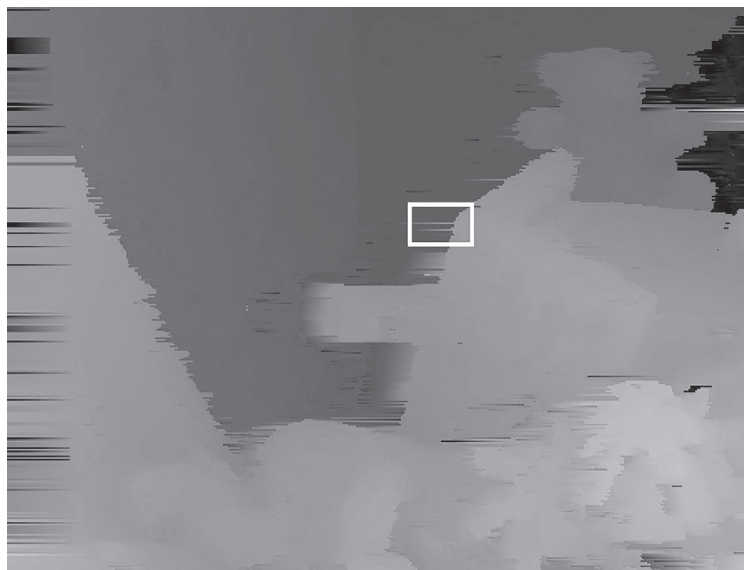
Naknadna obrada

Naknadna obrada (post-processing) ima za cilj poboljšavanje rezultata, oslanjajući se na poznate veze među podacima. Jedan primer poboljšanja rezultata se oslanja na pretpostavku da bi leva i desna slika trebalo da imaju slične mape pomeraja, u smislu da parovi elemenata mapa svojim pomerajima pokazuju jedno na drugo.

U ovom radu, naknadna obrada se sastoji iz tri faze, faze provere održivosti pomeraja, faze linearne interpolacije i faze filtriranja.

Provera održivosti pomeraja. Tokom prve faze upoređuju se leva i desna mapa pomeraja. Obeležavaju se oni pikseli koji su svojim pomerajima upareni sa pikselima koji svojim pomerajem nisu upareni sa njima, tj. proverava se uslov $|\text{disp}_p - \text{disp}_p| \leq 1$. Pretpostavlja se da će se pomenutim obeležavanjem izvršiti izdvajanje piksela čiji pomeraj nije tačan i sa njima pikseli koji se nalaze u delovima pod okluzijom.

Linearna interpolacija. Linearna interpolacija ima za cilj generisanje novih pomeraja na osnovu parametara piksela koji su ostali neobe-



Slika 7. Neusaglašene horizontalne linije

Figure 7. Horizontal streaks

ženi u fazi provere održivosti pomeraja. Za obeleženi piksel, generiše se pomeraj na osnovu parametara ravni najbližih neobeležanih piksela, sa leve i desne strane. Pomeraj se generiše po formuli (1).

U ovom delu interpolacije uvodi se aproksimacija da se proverom održivosti pronalaze samo pikseli zahvaćeni okluzijom. Parametri ravni na osnovu kojih je generisan manji pomeraj (objekat je dalje) biće uzeti za nove parametre obeleženog piksela koji se obrađuje. Ovaj odabir parametara oslanja se na činjenicu da je objekat pod okluzijom vidljiv na samo jednoj slici, pa je tako na drugoj slici sakriven iza, odnosno nalazi se dalje u odnosu na ostale objekte koji ga okružuju na slici na kojoj je vidljiv. Značajna primena PatchMatch algoritma oslikava se u ovoj interpolaciji, jer je određivanjem parametara ravni većine piksela moguće proceniti položaj piksela koji su pod okluzijom, odnosno za određivanje čijeg položaja nema dovoljno informacija.

Filtriranje. Nakon faze linearne interpolacije, dolazi do pojave neusaglašanih horizontalnih linija (slika 7). Male varijanse u parametrima ravni piksela koji ostaju neobeleženi u fazi provere održivosti pomeraja, prilikom interpolacije na obeležene piksele postaju velike. Ovo produkuje da pikseli koji su susedni po vertikalnoj osi, a koji su prethodno prošli fazu interpolacije, imaju neprirodno različite pomeraje.

Ovaj problem rešava se težinskim medijan filtrom. Filtracija se vrši samo nad obeleženim pikselima u fazi provere održivosti pomeraja. Način rada ovog filtra svodi se na obradu vrednosti pomeraja u određenom prozoru slike, u kome je piksel koji se obrađuje centralni piksel, a koji je iste veličine kao i u algoritmu. Svi pomeraji obuhvaćeni prozorom se izdvajaju i sortiraju. Tako sortiranim pomerajima dodaje se težinski faktor, koji se dobija formulom (4), isto kao i u cost funkciji. Medijan ovog težinskog niza biće nova vrednost piksela koji se obrađuje.

Rezultati i diskusija

Testiranje algoritama vršeno je na standardnoj bazi (Scharstein *et al.* 2014). Standardna baza sadrži skupove od po petnaest slika, namenjenih za validaciju i testiranje, u tri različite rezolucije. Testiranje je vršeno na test skupu slika, na četvrtini rezolucije (quarter resolution). U sklopu baze, postoji i referentna mapa pomeraja (ground truth), koja služi za upoređivanje rezultata. Prilikom evaluacije, koristi se odgovarajuća maska, koja definiše koji delovi mape nisu za upoređivanje. Ona takođe dolazi u sklopu baze za svaku sliku, i maskira kritične delove pod okluzijom.

Rezultati su upoređivani po procentu piksela čiji se pomeraji razlikuju za više od jedan (bad-

Tabela 1. 20 najboljih rezultata block-matching algoritma gledano po procentu piksela čija je apsolutna greška pomeraja veća od jednog piksela

window (pix)	alfa	gama	%mask	%badPix	avgError (pix)
15	0.5	20	98.0	26.7	3.70
15	0.5	15	98.0	26.8	3.72
15	0.3	20	98.0	26.9	3.76
15	0.7	15	98.0	26.9	3.70
15	0.7	20	98.0	27.0	3.74
15	0.5	10	98.0	27.1	3.73
15	0.7	10	98.0	27.1	3.79
15	0.3	15	98.0	27.2	3.78
15	0.9	10	98.0	27.2	3.98
15	0.1	20	98.0	27.2	3.80
15	0.9	15	98.0	27.3	3.95
15	0.1	15	98.0	27.4	3.78
15	0.9	20	98.0	27.4	3.91
15	0.3	10	98.0	27.4	3.75
15	0.7	5	98.0	27.6	3.82
15	0.1	10	98.0	27.6	3.77
15	0.5	5	98.0	27.8	3.84
25	0.5	5	98.0	27.9	4.07
25	0.3	10	98.0	27.9	4.06
25	0.9	10	98.0	27.9	4.11

Pix) od pomeraja referentne mape, kao i po prosečnom odstupanju pomeraja od referentne mape (avgError) izraženom u pikselima.

Prilikom testiranja varirani su parametri veličine prozora (W), gama (γ) i alfa (α) koeficijenti. Po opisanoj metrici, na slici Teddy iz standardne baze, dobijeni rezultati su prikazani u tabeli 1 za block-matching algoritam i u tabeli 2 za Patch-Match algoritam.

Najbolji rezultati dobijeni su sa prozorom veličine 15×15 piksela. Veličina prozora definiše koliki je deo scene za koji se uvodi aproksimaciju da se nalazi na istoj ravni, pa se tako može reći da direktno zavisi od rezolucije slike koja se obrađuje. Veličina prozora takođe određuje broj korišćenih piksela, da bi se odredio pomeraj centralnog piksela. Pikseli u prozoru mogu se nalaziti u okluziji, pa tako njihovo korišćenje pri upoređivanju prozora i njihovih piksela loše utiče na rezultate. Sa druge strane, veći broj piksela u prozoru, ukoliko se oni ne nalaze u okluziji,

znači i veći broj korisnih informacija što može povoljno uticati na rezultate.

Faktor gama (γ) definiše koliko razlika u boji utiče na težinski faktor određenog piksela po formulama (3) i (4). Najbolji rezultati su dobijeni sa vrednošću 20. Sa povećanjem prozora, razlika u rezultatima je sve manja, odnosno sve je manji uticaj ovog faktora.

Koeficijent alfa (α) definiše uticaj razlike u boji u odnosu na razliku gradijenta u cost funkciji po formulama (3) i (5). Sa PatchMatch algoritmom najbolji rezultat je postignut sa vrednošću 0.1, pa se tako zaključuje da pridavanje većeg značaja razlici u boji pozitivno utiče na rezultate kod ovog algoritma. Sa block-matching algoritmom najbolji rezultat je postignut sa vrednošću 0.5, što znači da je podjednak uticaj boje i gradijenta.

Analizom rezultata block-matching algoritma (slika 8c) i PatchMatch algoritma (slika 8e) na slici Teddy (slika 8a, slika 8b) i utvrđivanjem pozicija pogrešnih piksela dva algoritma (slika

Tabela 2. 20 najboljih rezultata PatchMatch algoritma gledano po procentu piksela čija je apsolutna greška pomeraja veća od jednog piksela

window (pix)	alfa	gama	%mask	%badPix	avgError (pix)
15	0.1	20	98.0	22.8	2.55
15	0.3	20	98.0	22.8	2.59
15	0.5	20	98.0	23.0	2.47
15	0.3	15	98.0	23.4	2.56
15	0.7	20	98.0	23.5	2.56
15	0.5	15	98.0	23.6	2.56
15	0.7	15	98.0	23.6	2.57
15	0.9	15	98.0	23.6	2.59
15	0.1	15	98.0	23.7	2.64
15	0.9	20	98.0	23.9	2.61
25	0.3	20	98.0	24.0	3.02
25	0.5	20	98.0	24.1	3.05
25	0.3	15	98.0	24.2	3.03
15	0.3	10	98.0	24.4	2.72
25	0.7	20	98.0	24.9	3.06
25	0.1	20	98.0	24.9	3.04
15	0.7	10	98.0	25.0	2.70
15	0.5	10	98.0	25.2	2.69
25	0.1	15	98.0	25.3	3.10
15	0.9	10	98.0	25.3	2.75

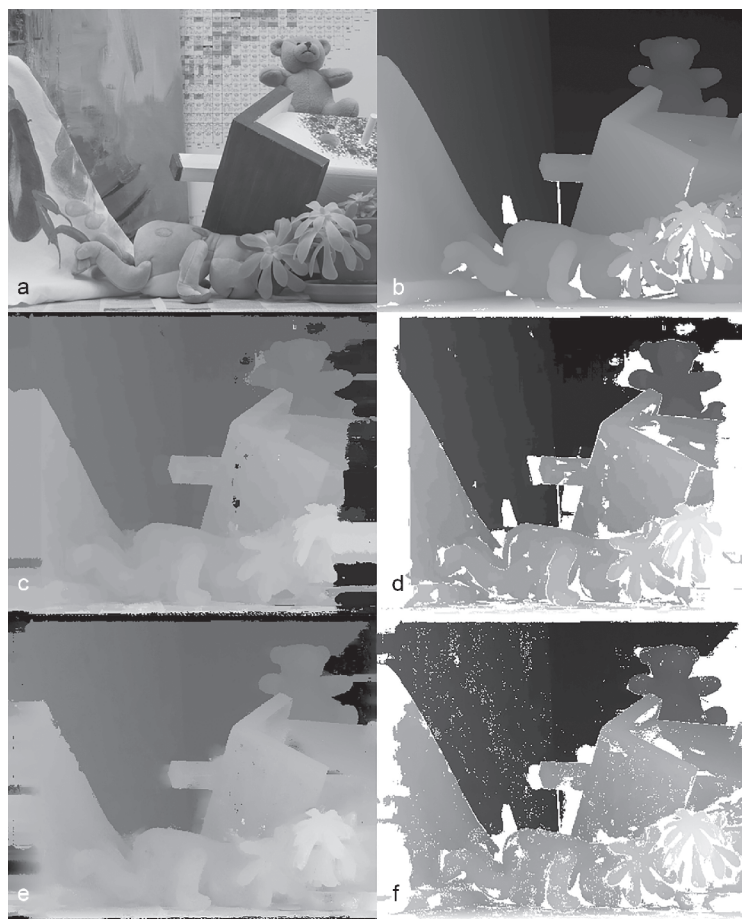
Tabela 3. Rezultati na slikama Motorcycle, Piano i Playtable

Slika	Algoritam	window (pix)	alfa	gama	%mask	%badPix	avgError (pix)
Motorcycle	BM	15	0.1	20	97.9	31.1	4.73
Motorcycle	PM	15	0.1	20	97.9	34.7	3.49
Piano	BM	15	0.1	20	94.6	39.5	4.12
Piano	PM	15	0.1	20	94.6	41.0	3.08
Playtable	BM	15	0.1	20	97.3	42.1	6.66
Playtable	PM	15	0.1	20	97.3	36.3	3.07

8d, slika 8f), može se zaključiti koje su oblasti kritične kojem algoritmu. Problem oba algoritma jesu slabo teksturisane površine, što je posledica forme cost funkcije, koja zavisi od razlika u boji i gradijentu, koje su veoma male na slabo teksturisanim površinama, a što se primećuje na slikama 8d i 8f (desno od medvedića i između lutke i odžaka kućice). Primećuje se takođe da PatchMatch algoritam daje bolje rezultate u većim delovima

slike pod okluzijom (levi deo slike i predeo levo od medvedića), što je posledica toga što PatchMatch određuje položaj ravni scene. Međutim PatchMatch algoritam generiše puno pogrešnih piksela rasutih po celoj slici (bele tačke na slici 8f). Ova pojava može se neutralisati povećanjem broja iteracija.

Algoritmi su nakon podešavanja parametara koji su određeni na osnovu slike Teddy testirani i



Slika 8. (a) Slika Teddy, (b) Referentna mapa slike Teddy, (c) Rezultat block-matching algoritma, (d) Ispravni pikseli generisani block-matching algoritmom, (e) Rezultat PatchMatch algoritma i (f) Ispravni pikseli generisani Patch Match algoritmom

Figure 8. (a) Teddy picture, (b) Ground-truth Teddy, (c) Result of block-matching algorithm, (d) Regular pixels generated by block-matching algorithm, (e) Result of PatchMatch algorithm and (f) Regular pixels generated by PatchMatch algorithm

na slikama Piano, Playtable i Motorcycle, dobijeni su rezultati koji se nalaze u tabeli 3. Slika Teddy je odabrana kao pogodna za određivanje optimalnih parametara jer ima velike ravne površi, slabo teksturisane površi i kompleksne površi.

Rezultati na slikama iz tabele 3 značajno su lošiji od rezultata oba algoritma na slici Teddy, za šta smatramo da je uzrok razlika u rezoluciji, zbog koje optimalni parametri na slici Teddy nisu optimalni i na ostalim slikama.

Na sve tri slike PatchMatch algoritam daje bolje rezultate po kriterijumu prosečne greške, što ukazuje na to da se određivanjem položaja ravni uspešnije određuju pomeraji u delovima slike pod okluzijom, koji predstavljaju njen značajan deo.

Odnos procenata pogrešnih piksela dva algoritma varira zavisno od karakteristika scene koja se nalazi na slici koja se obrađuje, pa se tako ispostavlja da block-matching algoritam na određenim slikama daje bolje rezultate po kriterijumu procenta pogrešnih piksela. Usled pojave ovakvih rezultata urađeno je dodatno testiranje, u kome je varirana maksimalna razlika pomeraja piksela od pomeraja piksela referentne mape za koju se piksel prihvata kao ispravan (tabela 4).

Analizom rezultata iz tabele 4, primećuje se da je za maksimalne razlike veće od 1, metrika PatchMatch algoritma u odnosu na block-matching algoritam bolja sa povećanjem vrednosti maksimalne razlike. Ovo znači da veći procenat pomeraja piksela generisanih PatchMatch algoritmom u odnosu na procenat pomeraja piksela

Tabela 4. Rezultati na slikama Piano, Playtable, Motorcycle i Teddy, za vrednosti 0.5, 1, 2 i 5 piksela maksimalne razlike pomeraja ispravnog piksela od pomeraja piksela referentne mape

Slika	Algoritam	%badPix 0.5	%badPix 1	%badPix 2	%badPix 5
Piano	PM	51.1	41.0	32.2	19.4
Piano	BM	53.7	39.5	31.8	21.3
Playtable	PM	50.4	36.3	25.8	15.1
Playtable	BM	60.3	42.1	31.9	23.1
Motorcycle	PM	48.2	34.7	25.8	17.7
Motorcycle	BM	50.1	31.1	22.6	17.8
Teddy	PM	36.3	22.8	15.7	10.9
Teddy	BM	44.3	27.2	21.1	14.1

Tabela 5. Rezultati iz standardne baze i pozicije u odnosu na druge algoritme u bazi po kriterijumu procenta pogrešnih piksela sa maksimalnom razlikom 1 i po kriterijumu prosečne greške

Slika	best %badPix	best avgError	worst %badPix	worst avgError	PM position (%badPix)	PM position (avgError)
Motorcycle	16.3	2.98	76.9	20.6	27	3
Piano	23.9	2.29	85.7	18.8	29	8
Playtable	20.5	1.81	90.2	45.6	11	3
Teddy	15.0	1.36	62.4	14.3	13	11

generisanih block-matching algoritmom ima manju apsolutnu grešku, pa tako za vrednost maksimalne razlike od 5 piksela PatchMatch algoritam daje bolju metriku po kriterijumu pogrešnih piksela na svim slikama. To je posledica toga što PatchMatch algoritam, zbog svoje mogućnosti da odredi položaj ravni scene, daje približnije rezultate u odnosu na block-matching algoritam u kritičnim delovima slike.

Specijalno, za vrednost maksimalne razlike od 0.5 piksela, metrika PatchMatch algoritma je bolja na svim slikama. To se dešava zbog sub-pixel preciznosti PatchMatch algoritma, odnosno zbog toga što PatchMatch algoritam za razliku od block-matching algoritma nudi rezultate u realnim brojevima što pri ovoj vrednosti maksimalne razlike predstavlja značajniju prednost.

Rezultati PatchMatch algoritma upoređeni su sa ostalih 49 algoritama standardne baze (tabela 5). PatchMatch algoritam je bolje plasiran u odnosu na ostale algoritme po kriterijumu prosečne greške (u proseku 7. pozicija) nego po kriterijumu udela pogrešnih piksela (u proseku 20. po-

zicija). Smatra se da bi se povećanjem maksimalne razlike pomeraja ispravnih piksela od pomeraja piksela referentne mape PatchMatch algoritam pozicionirao bolje po kriterijumu procenta pogrešnih piksela.

Zaključak

Najbolji rezultati na slici Teddy su postignuti sa prozorom veličine 15×15 piksela i sa gama 20. Optimalna vrednost koeficijenta alfa (α) za PatchMatch algoritam jeste 0.1, dok je vrednost α od 0.5 optimalna za block-matching algoritam. Koristeći optimalne parametre generisane na slici Teddy, testovi su vršeni i na slikama Motorcycle, Piano i Playtable standardne baze.

PatchMatch daje bolju metriku u odnosu na block-matching algoritam po kriterijumu prosečne greške, pa se može zaključiti da menjanje položaja ravni pozitivno utiče na rezultate po tom kriterijumu.

Metrika po kriterijumu procenta pogrešnih piksela PatchMatch algoritma bolja je od metrike

block-matching algoritma samo na slikama Playtable i Teddy. Stoga, izvršen je dodatni test u kome je varirana maksimalna razlika pomeraja ispravnog piksela od pomeraja piksela referentne mape. Rezultati tog testa pokazuju da za povećanje maksimalne razlike metrika PatchMatch algoritma u odnosu na block-matching algoritam postaje sve bolja. Za maksimalnu razliku od 5 piksela pomeraja ispravnog piksela od pomeraja piksela referentne mape, PatchMatch daje bolju metriku na svim slikama. Testirana je i metrika algoritma za vrednost maksimalne razlike od 0.5 piksela, na kojoj PatchMatch takođe postiže bolje rezultate na svim slikama.

Poredeći PatchMatch algoritam sa 49 algoritama iz standardne baze, pokazuje se da se, uzimajući u obzir testirane slike, algoritam pozicionira na 20. mestu po kriterijumu procenta pogrešnih piksela i na 7. mestu po kriterijumu prosečne greške.

Generalni problem PatchMatch algoritma jeste u tome što linearizuje scenu koja se vidi u delu slike u okviru prozora, pritom ostajući zavistan od parametra veličine prozora. Rešenje ovog problema nazire se u generisanju krivih površina, dodavanjem jednog ili više parametara po osi. Restrikcija parametara prozora, da se svi pikseli u okviru prozora moraju videti i na drugoj slici, može se rešiti uvođenjem normalizovane cost funkcije. Normalizovana cost funkcija ne bi razmatrala piksele za koji su parametri takvi da se ne vide na drugoj slici, dajući im vrednost aritmetičke sredine vrednosti funkcije svih vidljivih piksela. Normalizovana cost funkcija bi omogućila generisanje svih mogućih oblika, što bi značajno smanjilo uticaj okluzije na rezultate. Smatramo da je uticaj okluzije presudan u dobijenim rezultatima i da bi navedene ideje mogle značajno uticati na poboljšanje samih rezultata.

Literatura

Bleyer M., Rhemann C., Rother C. 2011. PatchMatch Stereo – Stereo Matching with Slanted Support Windows. U *British Machine Vision Conference BMVC 2011*, str. 1-11.

Boykov Y., Veksler O., Zabih R. 2001. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence*, **23** (11): 1222.

Je Ch., Park H-M. 2013. Optimized hierarchical block matching for fast and accurate image registration. *Signal Processing: Image Communication*, **28** (7): 779.

Patil S., Nadar J. S., Gada J., Motghare S., Nair S. S. 2013. Comparison of various stereo vision cost aggregation methods. *International Journal of Engineering and Innovative Technology*, **2** (8): 222.

Scharstein D., Hirschmüller H., Kitajima Y., Krathwohl G., Nesci N., Wang X., Westling P.. 2014. High-resolution stereo datasets with subpixel-accurate ground truth. U *36th German Conference on Pattern Recognition* (ur. X. Jiang *et al.*). Springer, str. 31-42.

Sun J., Shum H-Y., Zheng N-N. 2002. Stereo Matching Using Belief Propagation. *Computer Vision. U ECCV 2002 Lecture Notes in Computer Science*, str. 510-524.

Szeliski R. 2010. *Computer vision: Algorithms and applications*. Springer

Danilo Tonić and Marko Skakun

Creating Depth Map Using Stereo Camera

In this paper, a depth map was created using a stereo camera. A comparison of block-matching (Je and Park 2013) and PatchMatch (Bleyer *et al.* 2011) methods was conducted. PatchMatch introduces the possibility for a plane in the scene to be at a certain angle, and so, based on existing work of Bleyer, Rhemann and Rother (2011), it is expected that the mentioned algorithm yields better result. The methods were tested on a standard base (Scharstein *et al.* 2014) and the results confirmed the hypothesis. Results of PatchMatch on a Teddy picture are 22.8% bad pixels (pixels with error greater than one, compared to reference map) and average error (average of absolute deviation of reference map) of 2.43 pixels, while the results of block-matching for the same picture are 26.7% bad pixels and an average error of 3.70 pixels. The algorithms were also compared on Motorcycle, Piano and Playtable pictures from a standard base. PatchMatch shows better

performance for the criterion of average error on all pictures, while for the criterion of bad pixels percentage it gives better metrics only for Playtable. By varying the maximum difference of correct pixel and reference map pixel, for a maximum difference value of 0.5 and 5 pixels, PatchMatch shows better metrics for the bad pixel percentage criteria for all pictures. Relative to standard base algorithms (49 algorithms), the PatchMatch algorithm is positioned at place 20 based on the bad pixels criteria and place 7 for the average error criteria.

