

Rekonstrukcija predmetov iz barvnih slik z načrtovanjem naslednjega pogleda

Kristian Žarn, Danijel Skočaj

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko
E-pošta: kristian.zarn@gmail.com, danijel.skocaj@fri.uni-lj.si

Object reconstruction from color images with next best view planning

The quality of the reconstructed 3D model heavily depends on the input images. A user that has no feedback about the suitability of the images during the acquisition process can have difficulties complying with assumptions of the algorithms, which can result in an unsuccessful reconstruction. In this work, we focus on development of a software system that supports the entire reconstruction process. The user gets online information about the adequacy of every captured image and an estimate of quality for the current 3D model. We also present a novel method for next best view planning that systematically improves the quality of reconstruction.

1 Uvod

Kvaliteta pridobljenega 3D modela pri rekonstrukciji je močno odvisna od zajetih slik. Ročno zajemanje je lahko dolgotrajno opravilo, pri katerem želimo s slikami doseči ustrezno natančnost in pokritost modela. Uporabnik, ki med zajemanjem nima nobene povratne informacije o ustreznosti slik, s težavo upošteva vse predpostavke algoritmov, kar je lahko vzrok za neuspešno rekonstrukcijo. Omenjene težave so pogoste v klasičnih pristopih, kjer je gradnja 3D modela izvedena ločeno, po zajemanju slik.

Priljubljeni obstoječi rešitvi za gradnjo 3D modelov iz množice slik sta na primer VisualSFM [4] (odprtokodna) in Reality Capture [5] (komercialna). V omenjenih rešitvah so slike naložene z diska, zato uporabnik nima informacij o stanju rekonstrukcije med zajemanjem slik. To težavo smo v našem delu skušali odpraviti z načrtovanjem naslednjega pogleda.

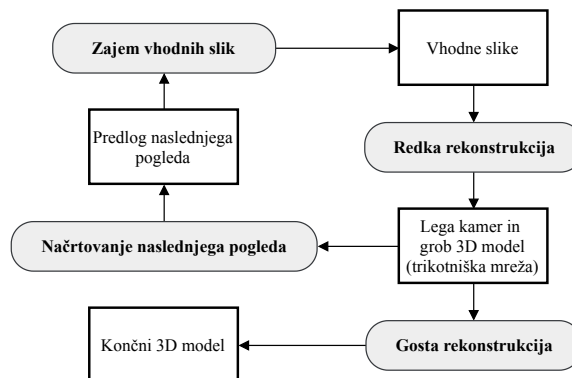
V literaturi so sicer s pojmom načrtovanje naslednjega pogleda označeni zelo različni pristopi. V tem delu problem obravnavamo kot inkrementalni proces sistematične izboljšave natančnosti in pokritosti 3D modela. Kriegl in ostali [2] so za gradnjo 3D modela uporabili industrijski robotski manipulator z laserskim skenerjem. Naslednji pogledi so izbrani na podlagi robov v modelu in cenovne funkcije, ki strmi k raziskovanju. Na področju rekonstrukcije iz barvnih slik, sta problem naslovila Dunn in Frahm [3]. Predlagala sta cenovno funkcijo, ki združuje negotovost v oceni strukture, projekcijo modela in videz teksture. Naslednji pogled je določen z optimizacijo te funkcije. Sproten odziv pri rekonstrukciji so na-

slovili tudi Hoppe in ostali [1]. V svojem delu so razvili rešitev, ki s sprotno informacijo o ustreznosti slik in kvaliteti modela pomaga uporabniku pri rekonstrukciji.

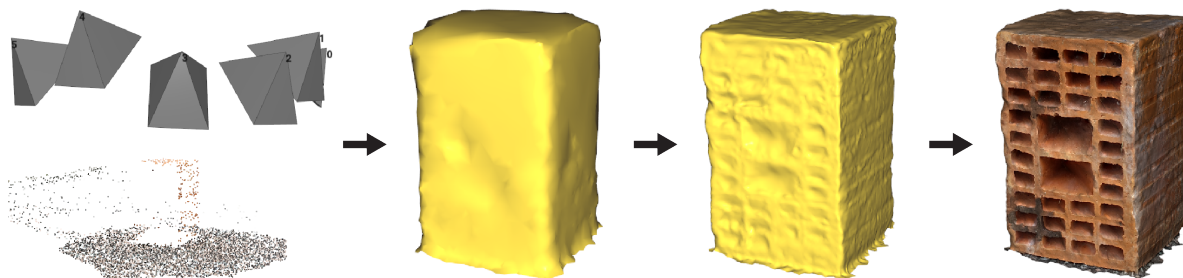
Podoben cilj smo si zadali tudi v tem delu. V njem se osredotočamo na razvoj postopka in programske opreme, ki podpira celoten proces rekonstrukcije. Za vsako zajeto sliko dobi uporabnik sprotno informacijo o njeni ustreznosti ter oceni kvalitete trenutnega 3D modela. Tako lahko uporabnik bolje presodi in načrtuje zajemanje nadaljnjih slik. Pri tem mu pomaga tudi predlagana nova metoda za načrtovanje naslednjih pogledov, ki sistematično izboljšajo kvaliteto rekonstrukcije.

Oris naše programske rešitve je prikazan na sliki 1. Diagram prikazuje odvisnost med funkcionalnostmi programa ter vhodnimi in izhodnimi podatki. Postopek se prične z zajemanjem slik, ki so pridobljene z IP kamero. V našem primeru je kot IP kamera uporabljen kar mobilni telefon, kar je mogoče s pomočjo dodatne mobilne aplikacije. Iz zajetih slik se postopoma gradi redka rekonstrukcija, ki vrača lego kamer in grob 3D model predstavljen s trikotniško mrežo. Na podlagi te predstavitev so predlagani tudi naslednji pogledi. Ko je zajetih dovolj slik, lahko uporabnik z algoritmi goste rekonstrukcije pridobi končni 3D model.

Podrobnejši opis posameznih sklopov naše rešitve je podan v poglavjih 2 in 3. V poglavju 4 predstavimo nekaj rezultatov rekonstrukcije, prispevek pa zaključimo s sklenimi ugotovitvami ter smernicami za nadaljnje delo.



Slika 1: Oris programske rešitve. Sivi okvirji predstavljajo funkcionalnosti programa, beli pa njihove vhode in izhode.



Slika 2: Prikazani so rezultati korakov gradnje 3D modela. Od leve proti desni si sledijo, redka rekonstrukcija z metodo Struktura iz gibanja, rekonstrukcija površine, izboljšava ločljivosti in natančnosti ter teksturiranje.

2 Gradnja 3D modela

Kot prikazuje slika 1 sta najboljšežnejši komponenti postopka 3D rekonstrukcije predmetov redka in gosta rekonstrukcija. Ti dve komponenti sta opisani v tem poglavju.

2.1 Redka rekonstrukcija

Za redko rekonstrukcijo smo uporabili postopek imenovan inkrementalna struktura iz gibanja [6]. Rekonstrukcija je inicializirana na podlagi prvih dveh slik. Na slikah poiščemo značilnice SIFT [7] in njihova ujemanja. Na podlagi ujemanj izračunamo tako imenovano osnovno matriko. Iz nje pridobimo relativno lego prvih dveh kamer [8]. Magnituda translacije iz osnovne matrike ne more biti določena, zato pravimo, da je rekonstrukcija določena do velikosti natančno. Oblak 3D točk inicializiramo s triangulacijo [9] ujemajočih značilnic. Rekonstrukcijo nato postopoma nadgrajujemo z dodajanjem novih slik. Iz nove vhodne slike najprej pridobimo značilnice. Naslednji korak je kritičen za hitro razširitev. Namesto iskanja ujemanj z vsemi predhodnimi slikami, iščemo ujemanja samo s petimi najbolj podobnimi slikami. Te kandidate za ujemanje pridobimo z uporabo tako imenovanega vizualnega slovarja [11]. Značilnice nove slike, za katere smo našli ujemanje, lahko sedaj razdelimo v dve množici. V prvi množici so tiste, ki so že del rekonstrukcije. Uporabimo jih za določanje absolutne lege kamere [10]. V drugi množici so tiste, ki še niso del rekonstrukcije. Dodamo jih s triangulacijo. Pomembno je, da po vsaki dodani sliki minimiziramo reprojekcijsko napako. S tem izboljšamo natančnost in omilimo akumulacijo napake. Na podlagi oblaka točk in lege kamer lahko rekonstruiramo površino predmeta [12]. Ker gre za hitro operacijo, jo lahko izvedemo po vsaki dodani sliki. Tako pridobimo grob 3D model predstavljen s trikotniško mrežo. Na sliki 2 so prikazani vmesni rezultati gradnje 3D modela. Opisani postopek zajema prva dva koraka.

Kot osnovo pri implementaciji redke rekonstrukcije smo uporabili knjižnico TheiaSfM [15]. Ta vsebuje številne algoritme s področja strukture iz gibanja. Za hitrejše iskanje značilnic smo uporabili implementacijo na grafični kartici iz knjižnice SiftGPU [16]. Implementacijo vizualnega slovarja smo vzeli iz knjižnice COLMAP [17].

2.2 Gosta rekonstrukcija

Za ceno dodatnega procesiranja lahko redko rekonstrukcijo izboljšamo z algoritmi goste rekonstrukcije. Implementacije algoritmov iz knjižnice OpenMVS [18] smo integrirali v našo rešitev. V nadaljevanju na kratko predstavimo njihove glavne ideje.

Izboljšava ločljivosti 3D modela [13] je časovno zahtevna operacija, ki jo izvedemo, ko so zajete vse slike. Ločljivost grobega modela je z delitvijo trikotnikov prilagojena ločljivosti vhodnih slik. Nad vozlišči gostejšega modela je definirana energijska funkcija. Z minimizacijo te funkcije določimo natančnejšo pozicijo vozlišč.

Končni model pridobimo s teksturiranjem [14], ki je ključnega pomena za doseg realističnega izgleda. Algoritem deluje v dveh korakih. V prvem koraku vsakemu trikotniku pripišemo oznako, ki določa en pogled, ki bo uporabljen za njegovo teksturiranje. Drugi korak algoritma pa je globalna uskladitev barv, tako da med trikotniki ni vidnih prehodov. Opisana algoritma zajemata zadnja dva koraka slike 2.

3 Načrtovanje naslednjega pogleda

Vhod v rekonstrukcijo 3D modela, predstavljeno v prejšnjem poglavju, je množica slik. Običajno so te slike posnete vnaprej. V tem delu pa predlagamo postopek, pri katerem uporabnik sproti dobi odziv o primernosti zajetih slik, oz. namig s katerih pogledov naj zajame slike, da zagotovi smiselno gradnjo 3D modela.

3.1 Ocena kvalitete 3D modela

V ta namen vpeljemo oceno za kvaliteto 3D modela in jo definiramo s funkcijo, ki vsakemu trikotniku priredi realno vrednost. Želimo si, da prirejena vrednost čim boljše predstavlja kvaliteto končnega modela. Na ta način ima uporabnik že med rekonstrukcijo na voljo informacije o tem, kateri deli modela so bolj oz. manj natančni. Tako lahko sprejme informirano odločitev o postavitvi naslednje kamere, poleg tega pa smo oceno kvalitete uporabili tudi kot osnovo za načrtovanje naslednjega pogleda. V tem delu predlagamo novo mero imenovano PPA (angl. pixels per area). Mero zapišemo z enačbo

$$Q_{PPA}(T_i) = \sqrt{\frac{\sum C_j P(T_i, C_j)}{A(T_i)}}. \quad (1)$$

Trikotnik T_i preslikamo na vse kamere C_j na katerih je viden in seštejemo s funkcijo P dobljeno število slikovnih elementov, ki jih preslikave zasedajo. Dobljeno vrednost delimo s ploščino trikotnika $A(T_i)$ v 3D prostoru in rezultat korenimo. Večja vrednost funkcije Q_{PPA} predstavlja višjo kvaliteto modela.

3.2 Izbira najboljšega naslednjega pogleda

Ustreznost naslednjega pogleda ocenimo s cenovno funkcijo, ki za podano lego kamere oceni kako dobra je njena postavitev. Zapišemo jo z enačbo

$$f_{NBV}(C) = \alpha * \mu(Q_{PPA}(v(C))) - \beta * \sigma(Q_{PPA}(v(C))). \quad (2)$$

Funkcija v za postavitev kamere C vrne vse vidne trikotnike. Za vsak viden trikotnik pridobimo njegovo oceno kvalitete Q_{PPA} . Na podlagi pridobljenih vrednosti izračunamo dva podatka in sicer njihovo povprečje μ (prvi člen enačbe) in standardni odklon σ (drugi člen enačbe). Prispevek posameznega člena uravnavamo s parametroma $\alpha = 1$ in $\beta = 5$. Utemeljitev za takšno definicijo je, da si za novo kamero želimo, da izboljša območja slabe kvalitete (nizka vrednost μ), po drugi strani, pa je kamera lahko pravilno lokalizirana samo, če so vidni tudi bolj kvalitetni deli modela (visoka vrednost σ).

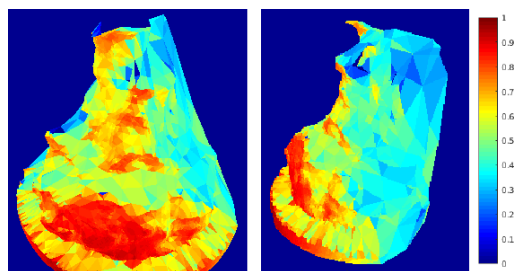
Načrtovanje naslednjega pogleda poteka tako, da generiramo nekaj kandidatov za lego kamere in nato s funkcijo f_{NBV} ovrednotimo njihovo ustreznost. Kandidati so generirani na naslednji način. Model je naključno razdeljen na povezane gruče, ki vsebujejo med $c_{min} = 100$ in $c_{max} = 300$ trikotnikov. Normala posameznega trikotnika v gruči od povprečja ne sme odstopati za več kot $\phi = 100$ stopinj. Za vsako gručo generiramo eno postavitev kamere. Kamera je postavljena od središča gruče v smeri povprečja normal trikotnikov znotraj gruče. Usmerjena je proti centru gruče, njena oddaljenost pa je določena z enačbo $d = \gamma \sqrt[3]{A(G)}$, kjer funkcija A vrne ploščino gruče, korenska funkcija pa prepreči preveliko oddaljenost kamere. Dodaten parameter $\gamma = 3$ je odvisen od goriščne razdalje kamere. Izračun cenovne funkcije f_{NBV} za konkreten primer je prikazan na sliki 3.

Izboljšava kvalitete modela je postopna. Na začetku postavimo ciljno kvaliteto $q_t = 1500$. Vsakič ko $q_p = 85\%$ trikotnikov doseže ciljno kvaliteto, parameter q_t povečamo za $q_{inc} = 500$. Parameter q_t se torej postopoma povečuje in zasede vrednosti 1500, 2000, 2500 itd. Na ta način še dodatno spodbudimo raziskovanje in preprečimo morebitno kopičenje pogledov na enem območju. Vrednosti parametrov v tem opisu smo določili empirično.

4 Rezultati

Naš sistem smo evalvirali z rekonstrukcijo resničnih predmetov. Ker za te predmete nimamo poznanega referenčnega 3D modela, predstavljamo v tem poglavju samo kvalitativne rezultate.

Za inicializacijo smo po lastni presoji izbrali nekaj pogledov, pri dodajanju slik pa smo upoštevali predlagane poglede. Lokalizacija predlaganega pogleda je včasih neuspešna, ali pa natančna postavitev kamere zaradi



Slika 3: Upodobitev modela z mero PPA za dva predlagana pogleda. Rdeča barva predstavlja dobro natančnost, modra pa slabo. Vrednost cenovne funkcije f_{NBV} za levi pogled znaša 30,3, za desni primer pa $-111,9$. Desni pogled je bolje ocenjen zaradi manjšega povprečja in večjega standardnega odklona mere PPA na vidnih trikotnikih.

fizičnih omejitev prostora ni mogoča. V takšnih primerih smo izbrali naslednji predlagani pogled, v rezultatih pa poročamo o povprečnem indeksu izbranega pogleda (indeks 0 pomeni, da so bile ustrezne vse slike, indeks 1 pa, da je bila za vsako izbrano sliko ena izpuščena).

Prvi primer rekonstrukcije je prikazan na sliki 4. Za gradnjo 3D modela smo uporabili 57 slik. Povprečni indeks izbranega naslednjega pogleda znaša 0,8. Geometrija debla se dobro ujema z njegovim resničnim videzom.



Slika 4: Nekaj primerov slik (levo zgoraj), postavitev kamer (desno zgoraj) in rezultat goste rekonstrukcije debla (spodaj).

Rekonstrukcija naslednjega primera je prikazana na sliki 5. Zajetih je bilo 31 slik, povprečni indeks naslednjega pogleda pa znaša 0,48. Na slabše teksturiranih predelih čevlja sicer lahko opazimo nekaj nepravilnosti, v splošnem pa se geometrija vizualno dobro ujema z resničnim predmetom.

Zadnji primer rekonstrukcije je prikazan na sliki 6. Čeprav je oblika predmeta navidez preprosta, je zaradi ostrih robov in nekoliko manj ugodne teksture žoge ta primer bolj težaven za rekonstrukcijo. Zajetih je bilo 43 slik,



Slika 5: Nekaj primerov slik (levo zgoraj), postavitev kamer (desno zgoraj) in rezultat goste rekonstrukcije gorskih čevljev (spodaj).

povprečni indeks naslednjega izbranega pogleda pa znaša 1,1. Rekonstrukcija spodnjega dela (lonca) je zelo uspešna, nekoliko več nepravilnosti pa lahko opazimo na žogi, predvsem zaradi homogene površine, ki ne omogoča zanesljive detekcije ponovljivih značilnic. Homogene površine torej niso najbolj primerne za 3D rekonstrukcijo z opisanim pristopom.



Slika 6: Primer slike (levo) ter rezultat goste rekonstrukcije lonca in žoge (desno).

5 Zaključek

V tem delu smo z uporabo odprtokodnih knjižnic in sodobnih algoritmov implementirali celoten postopek rekonstrukcije predmetov iz barvnih slik. Poleg programske opreme je prispevek našega dela tudi zasnova in implementacija nove metode za načrtovanje najboljšega naslednjega pogleda, ki temelji na novi meri za oceno kvalitete 3D modela. Uspešno delovanje naše metode smo prikazali z rekonstrukcijo resničnih predmetov. V delu [19] je na voljo podrobnejši opis uporabljenih metod in obsežnejša evalvacija našega sistema.

V nadaljnjem delu se bomo posvetili še nekaterim izboljšavam. Oceno kvalitete bi lahko izboljšali s pomočjo strojnega učenja. Pri načrtovanju naslednjih pogledov

pa bi v cenovno funkcijo bilo smiselno vpeljati dodatne informacije o videzu teksture in vidnosti rekonstruiranih značilnic. Na ta način bi bil proces gradnje 3D modelov še bolj zanesljiv.

Literatura

- [1] C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, G. Reitmayr, Online feedback for structure-from-motion image acquisition, in: *BMVC*, Vol. 2, 2012, p. 6.
- [2] S. Kriegel, C. Rink, T. Bodenmüller, M. Suppa, Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects, *Journal of Real-Time Image Processing* 10 (4) (2015) 611-631.
- [3] E. Dunn, J.-M. Frahm, Next best view planning for active model improvement, in: *BMVC*, 2009, pp. 1-11.
- [4] C. Wu, *Visualsfm: A visual structure from motion system*, <http://ccwu.me/vsfm/>.
- [5] RealityCapture, <https://www.capturingreality.com/>.
- [6] C. Wu, Towards linear-time incremental structure from motion, in: *3D Vision-3DV 2013*, 2013 International Conference on, IEEE, 2013, pp. 127-134.
- [7] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International journal of computer vision* 60 (2) (2004) 91-110.
- [8] H. Stewenius, C. Engels, D. Nistér, Recent developments on direct relative orientation, *ISPRS Journal of Photogrammetry and Remote Sensing* 60 (4) (2006) 284-294.
- [9] R. Szeliski, *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.
- [10] L. Kneip, D. Scaramuzza, R. Siegwart, A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation, *CVPR 2011*, pp. 2969-2976.
- [11] R. Arandjelović, A. Zisserman, Dislocation: Scalable descriptor distinctiveness for location recognition, in: *ACCV*, Springer, 2014, pp. 188-204.
- [12] M. Jancosek, T. Pajdla, Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces, *International scholarly research notices* 2014.
- [13] H.-H. Vu, P. Labatut, J.-P. Pons, R. Keriven, High accuracy and visibility-consistent dense multiview stereo, *IEEE transactions on pattern analysis and machine intelligence* 34 (5) (2012) 889-901.
- [14] M. Waechter, N. Moehrle, M. Goesele, Let there be color! large-scale texturing of 3D reconstructions, in: *ECCV*, Springer, 2014, pp. 836-850.
- [15] C. Sweeney, Theia multiview geometry library: Tutorial & reference, <http://theia-sfm.org>.
- [16] W. Changchang, SiftGPU: a GPU implementation of scale invariant feature transform (SIFT), <https://github.com/pitZER/SiftGPU> (2007).
- [17] J. L. Schönberger, J.-M. Frahm, Structure-from-motion revisited, in: *CVPR*, 2016.
- [18] Openmvs: open multi-view stereo reconstruction library, <https://github.com/cdseacave/openMVS>.
- [19] Kristian Žarn, *Gradnja 3D modelov predmetov iz barvnih slik z načrtovanjem najboljšega naslednjega pogleda*, Magistrsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2019.