

Rekonstrukcija objekta in poravnava oblakov točk

Martin Knap, Janez Lapajne

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: janez.puhan@fe.uni-lj.si

Povzetek. Izdelovalci integriranih vezij za svoje procese podajajo knjižnice osnovnih digitalnih gradnikov, ki se uporabljajo pri načrtovanju digitalnih integriranih vezij. Izvedbe posameznih gradnikov na tranzistorskem nivoju so s tem določene in se med načrtovanjem vezja ne spreminjajo več. Gradniki, kot npr. medpomnilniki, logična vrata, seštevvalniki, flip-flopi itd., se uporabljajo brez sprememb, kjerkoli jih načrtovalec potrebuje. Zaradi tega posamezen gradnik ni povsem prilagojen na specifično okolje v vezju, kjer se nahaja. To odpira prostor optimizaciji gradnika točno na zahteve v katerih deluje. članek opisuje primer optimizacije digitalnega gradnika na tranzistorskem nivoju. Zaradi šumnosti kriterijske funkcije je bila uporabljena robustna globalna optimizacijska metoda. Rezultati pokažejo znatno izboljšanje želenih lastnosti glede na splošni gradnik izdelovalca integriranih vezij.

Ključne besede: rekonstrukcija 3d objekta, kalibracija kamere, poravnava oblakov točk

1 UVOD

V nalogi je obravnavan postopek 3D rekonstrukcije objektov na podlagi večih zajetih slik. Za rekonstrukcijo je uporabljena FBP (angl. *Filtered back projection*) metoda. Temelji na rekonstrukciji 3D oblika iz slik, ki so zajete pri znanih kotih zasuka objekta. Pri postopku rekonstrukcije ima zelo veliko vlogo sistem s katerim zajemamo slike. Tega je potrebno ustrezno skalibrirati in merjene točke pretvoriti v metrični koordinatni sistem. Bolje kot zastavimo zajem slik, ustrežnejši so končni rezultati. Dobro zajete slike nam olajšajo delo pri kasnejši obdelavi.

2 TEORETIČNO OZADJE

2.1 Kalibracija

Pred začetkom izvedbe rekonstrukcije je potrebna geometrijska kalibracija sistema. Namen kalibracije je pridobitev parametrov toge preslikave, ki jih bomo potrebovali kasneje v algoritmu 3D rekonstrukcije s filtrirano povratno projekcijo. Cilj je projekcija iz slikovnega v metrični koordinatni sistem.

Referenčne točke kalibracijskega objekta (3D) najprej togo premaknemo na mesto rotirajoče mize. Nato sledi DLT (angl. *Direct Linear Transformation*) kalibracija med 3D točkami referenčnega kalibra $T_{ref,i} = (X_i, Y_i, Z_i)$ ter 2D naborom točk kalibra iz slikovnega koordinatnega sistema $t_{slik,i} = (x_i, y_i)$. Pri tem je število točk kalibracijskega objekta $i = 8$. Pri DLT kalibraciji z razcepom na singularne vrednosti rešujemo sledeči sistem:

$$a = \begin{bmatrix} X_i & Y_i & Z_i & 0 & 0 & 0 & -x_i X_i & -x_i Y_i & -x_i Z_i & 1 & 0 & -x_i \\ 0 & 0 & 0 & X_i & Y_i & Z_i & -y_i X_i & -y_i Y_i & -y_i Z_i & 0 & 1 & -y_i \end{bmatrix} \quad (1)$$

$$p = [r_{11} \ r_{12} \ r_{13} \ r_{21} \ r_{22} \ r_{23} \ r_{31} \ r_{32} \ r_{33} \ t_x \ t_y \ t_z] \quad (2)$$

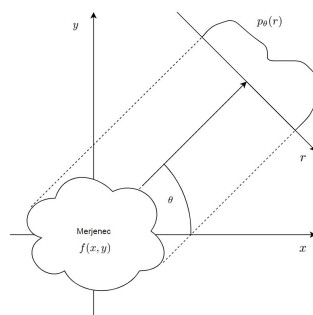
$$ap^T = 0 \quad (3)$$

Rešitev sistema predstavlja 9 parametrov toge preslikave, ki predstavljajo rotacijo r_{ij} ter translacijo t_k .

2.2 Filtrirana povratna projekcija

Jedro naloge predstavlja postopek FBP (angl. *Filtered Backprojection*), ki je eden od temeljnih načinov rekonstrukcije pri slikanju z računalniško tomografijo na področju zdravstva.

Idejo koncepta filtrirane povratne projekcije je najenostavneje predstaviti na 2D primeru, kjer 2D merjencu $f(x, y)$ za več kotov θ zajamemo 1D prečne projekcije $p_\theta(r)$.



Slika 1:: Filtrirana povratna projekcija.

Zajete projekcije so nato pretvorjene v frekvenčno domeno s pomočjo Fourier-ove transformacije $P_\theta(\omega)$ za vsak kot. Tu nato sledi filtriranje projekcij z visokoprepustnim filtrom s frekvenčnim odzivom $|\omega|$ tako, da dobimo $g_\theta(r) = \int P_\theta(\omega)|\omega|e^{i2\pi\omega r}d\omega$. S filtriranjem se želimo znebiti visokih frekvenc v projekcijah in posledično pridobiti kvalitetnejšo rekonstrukcijo. Po filtriranju nato sledi še rekonstrukcija v $f'(x, y)$ in sicer tako, da filtrirane inverze Fourier-ove transformacije seštejemo.

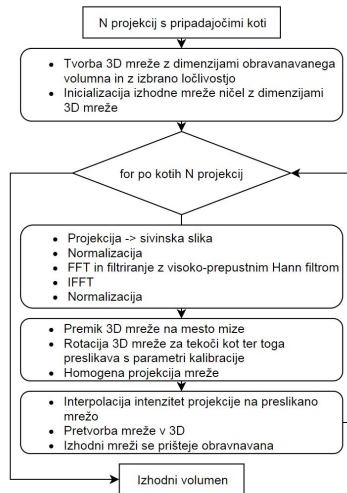
$$f'(x, y) = \int g_\theta(r)(x \cos \theta + y \sin \theta)d\theta \quad (4)$$

Ali v diskretni obliki, kjer je $\Delta\theta$ korak diskretizacije:

$$f'(x, y) = \frac{1}{2\pi} \sum_{i=0}^{N-1} \Delta\theta_i g_{\theta_i}(x \cos \theta_i + y \sin \theta_i) \quad (5)$$

Za potrebe naloge je postopek razširjen v tretjo dimenzijo, implementirani algoritem pa je predstavljen v spodnjem podpoglavju.

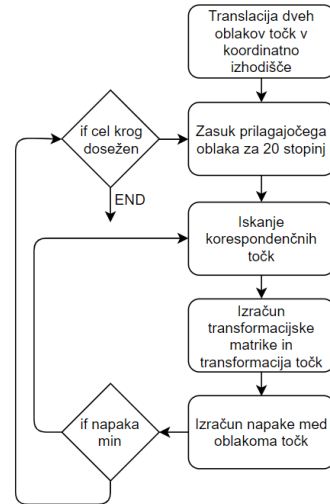
2.2.1 FBP algoritem: Na spodnji sliki je prikazan diagram poteka uporabljenega FBP algoritma.



Slika 2:: Diagram poteka FBP algoritma.

2.3 Poravnava oblakov točk

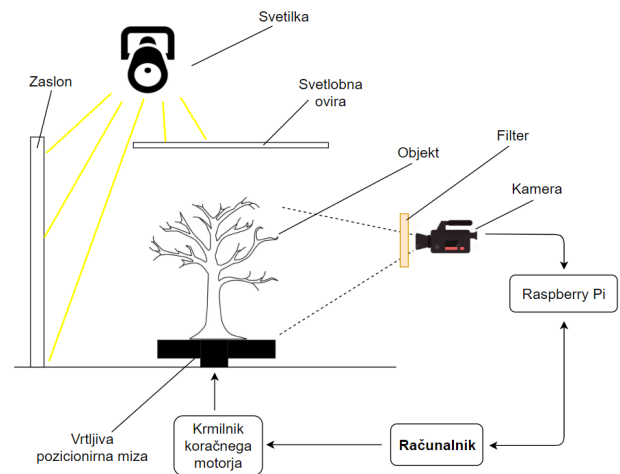
Potek algoritma poravnave točk je prikazan na sliki 3. Poravnavo izvajamo med dvema oblakoma točk tako, da enega prilagajamo na drugega. Na začetku oba oblaka točk postavimo v koordinatno izhodišče. Nato izvajamo optimizacijo prileganja na različnih rotacijah prilagajajočega oblaka. Slednje je izvedeno tako, da prilagajajoč oblak rotiramo po 20 stopinj in v vsaki legi izvajamo optimizacijo. To ponavljamo za 360 stopinj. Na koncu med vsemi izberemo poravnano lego, ki nam zagotavlja najmanjšo povprečno napako.



Slika 3:: Potek algoritma poravnave.

3 ZAJEM SLIK

Sistem s katerim smo izvajali rekonstrukcijo je blokovno prikazan na sliki 4. Objekt, ki ga želimo skonstruirati postavimo na vrtljivo pozicionirno mizo. Iz zgornje strani navzdol svetimo s homogeno, difuzno osvetlitvijo. Objekta direktno ne osvetljujemo, saj je namen osvetlitve predvsem osvetlitev ozadja z enakomerno intenziteto. Cilj je doseči čimvečjo razliko v intenziteti med opazovanim objektom in zaslonom. Za ta namen je med objekt in svetilko postavljena svetlobna ovira. Slike smo zajemali z Raspberry Pi kamero v osi na objekt in na osvetljeno ozadje. Pred kamero je dodan filter za namen filtriranja okoliške svetlobe.



Slika 4:: Semioperacijska shema sistema.

Celoten proces zajemanja krmilimo z računalnikom. Ta je žično povezan na krmilnik s katerim upravljamo rotirajočo pozicionirno mizo. Prav tako je računalnik brezžično povezan na Raspberry Pi. Ta zajete slike pošilja na računalnik. Objekt rekonstruiramo tako, da ga

postavimo na predvideno mesto. Nato lahko pričnemo z zajemom slik. Najprej inicializiramo kamero na začetne vrednosti. Nastavimo ISO na 200, svetlost na 45 in resolucijo na 900x1000. Objekt obračamo po začrtanih zasukih. Po vsakem izvedenem zasuku posnamemo eno sliko, ki se nato brezžično prenese iz Raspberry Pi-ja na računalnik. Postopek ponavljamo dokler objekta ne poslikamo v rangu celotnega kroga. Zajete slike ob znanih kotih zasuka se kasneje uporabijo za 3d rekonstrukcijo. V tabeli 1 so zapisane vse glavne komponente, ki so zajete v postavljenem sistemu. Manjše komponente kot so vijaki in nosilci v tabeli niso zajeti.

Tabela 1:: Komponente sistema.

Komponenta	Podrobnosti
Raspberry Pi	Raspberry Pi 3 Model B
Svetilka	Halogenska svetilka (12V)
Zaslon	Bela ravna plošča
Krmilnik	Fischertechnik stage control
Kamera	RP Cam V2-8 MP,1080p
Svetlobni filter	Ozkopasovni filter
Svetlobna ovira	Kartonasta zavesa
Pozicionirna miza	Fischertechnik rotary stage

Primer zajete slike obravnavanega objekta z opisanim sistemom je prikazan na sliki 5. S slike lahko vidimo dober kontrast objekta in precejšnjo razliko v barvi in svetlosti objekta proti ozadju.



Slika 5:: Zajeta slika.

4 REZULTATI

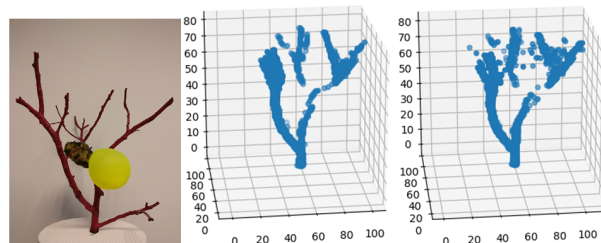
4.1 Vpliv kontrasta

- primerjava kompleksnosti modela - Primerjaj slab in dober kontrast slike (primer slusalk na 30 slikah)

4.2 Število zajetih slik

Dvakrat smo rekonstruirali isti predmet. Enkrat na podlagi 30 slik in drugič na podlagi 90 slik. Opazovali smo kako število zajetih slik vpliva na rekonstrukcijo. Na sliki 6 je čisto na levi strani prikazan skeniran objekt, na sredini objekt rekonstruiran na podlagi 90 slik in na desni objekt rekonstruiran na podlagi 30 slik. Vidimo, da je dobimo boljše obliko na podlagi večih

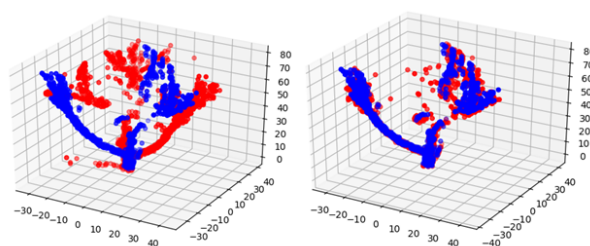
slik. Za prikaz celotnega objekta v obeh primerih bi bilo potrebno prilagoditi meje upravanja.



Slika 6:: a) objekt, b) skan - 90 slik, c) skan - 30 slik.

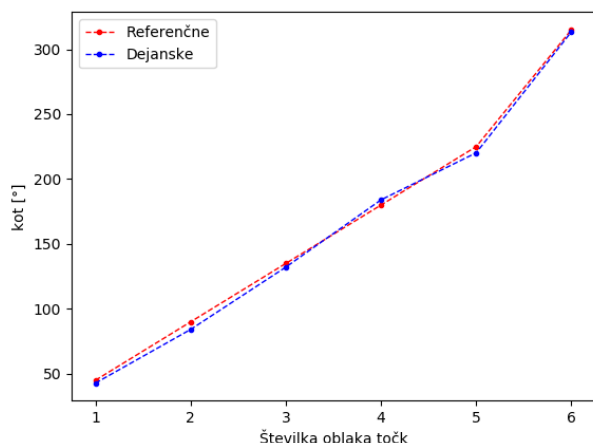
4.3 Poravnava oblakov točk

Za dodaten eksperiment smo izvedli medsebojno poravnavo oblakov točk. Referenčni oblak točk smo zajeli z 90 slikami in si zapolnili referenčno lego objekta na pozicionirni mizi. Nato smo izvedli še šest zajemov po 30 slik. Ob vsakem zajemu smo skeniran objekt zasukali za predpisan kot glede na referenčno lego okoli vertikalne osi. Tako smo zajeli oblake točk zasukane za 45, 90, 135, 18, 225 in 315 stopinj. Na sliki 7 sta zaradi preglednosti prikazana le referenčni oblak in oblak zasukan za 90 stopinj. Na levi strani sta oblaka prikazana pred poravnavo in na desni po poravnavi.



Slika 7:: a) neporavnana, b) poravnana.

Za vse, pod različnimi koti zajete oblake točk, smo izvedli poravnavo. Nato smo iz pridobljene transformacijske matrike izluščili rotacijo okoli vertikalne osi. Ker smo poznali kote zasuka objekta glede na referenčno lego, smo jih primerjali s temi pridobljenimi iz transformacijske matrike. Na sliki 8 so z rdečo barvo prikazani ročno določeni zasuki, z modro pa zasuki pridobljeni iz transformacijske matrike.



Slika 8:: Razlika med referenčnimi in dejanskimi koti.

4.4 Vzvratno inženirstvo

- mesh v blenderju in 3d print (reverse engineering)

5 ZAKLJUČEK

LITERATURA

- [1] H. Kaeslin, *Digital integrated circuit design: from VLSI architectures to CMOS fabrication*, Cambridge University Press, 2008.
- [2] *HSPICE® simulation and analysis user guide*, Synopsys®, 2005.
- [3] K.S. Kunderth, *The designer's guide to SPICE and Spectre*, Kluwer Academic Publishers, 1995.
- [4] *Virtuoso® Spectre® circuit simulator user guide*, Cadence Design Systems, Inc., 2008.
- [5] T. Tuma, Á. Bűrmen, *Circuit simulation with SPICE Opus, theory and practice*, Birkhäuser, 2009.
- [6] D.E. Thomas, P.R. Moorby, *The Verilog hardware description language*, 5th ed. Kluwer Academic Publishers, 2003.
- [7] V.A. Pedroni, *Circuit design with VHDL*, Massachusetts Institute of Technology, 2004.
- [8] P. Kurup, T. Abbasi, *Logic synthesis using Synopsys®*, 2nd ed. Kluwer Academic Publishers, 1997.
- [9] H. Bhatnagar, *Advanced ASIC chip synthesis using Synopsys® Design Compiler™ Physical Compiler™ and PrimeTime®*, 2nd ed. Kluwer Academic Publishers, 2002.
- [10] Á. Bűrmen et al., "Automated robust design and optimization of integrated circuits by means of penalty functions," *AEÜ, International Journal of Electronics and Communications*, vol. 57, no. 1, pp. 47–56, 2003.
- [11] J. Olenšek, Á. Bűrmen, J. Puhon, T. Tuma, "DESA: a new hybrid global optimization method and its application to analog integrated circuit sizing," *Journal of Global Optimization*, vol. 44, no. 1, pp. 1–25, 2008.
- [12] R. Hooke, T. Jeeves, "Direct search solutions of numerical and statistical problems," *Journal of the Association for Computing Machinery*, vol. 8, no. 2, pp. 212–229, 1961.
- [13] Á. Bűrmen, I. Fajfar, T. Tuma, "Combined Simplex-Trust-Region optimization algorithm for automated IC design," *Proceedings of ECCTD07 European Conference on Circuit Theory and Design*, pp. 543–546, 2007.