

Predstavitev trianguliranih površij geometrijskih teles z izboljšanim redčenjem obnovljivih trikotnikov

Blaž Repnik¹, David Podgorelec¹, Ivana Kolingerová², Borut Žalik¹

¹Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko

²University of West Bohemia, Faculty of Applied Sciences, Pilsen, Czech Republic

E-pošta: blaz.repnik@um.si

Representation of Triangulated Surfaces of Geometric Solids with Improved Decimation of Removable Triangles

Representing computer-modelled solids by their triangulated surfaces is frequently used in practice. Such a description is stored in various open file formats. However, the number of triangles may be very large when smooth-like surfaces are desired. One approach is to decimate these triangles in a way that they can be restored uniquely. In this paper, the ideas of the decimation and restoration processes are first presented followed by an improvement. Experiments showed that the expected improvement is about 3% more decimated triangles.

1 Uvod

Aplikacije, ki uporabljajo računalniško predstavitev 3D geometrijskih objektov (v nadaljevanju: teles), danes srečamo na številnih področjih – od inženirstva, medicine, izobraževanja do digitalne kulturne dediščine in računalniških iger. Iz računalniškega vidika so zanimive podatkovne strukture, ki hranijo podatke o telesih, ter množica algoritmov, ki z njimi upravljajo. Zaradi pomembnosti te naloge se je v računalništvu razvila posebna disciplina – geometrijsko modeliranje, ki je osnova za računalniško geometrijo in grafiko, računalniško animacijo, multimedio ter navidežno resničnost.

Ključna naloga geometrijskega modeliranja je nedvoumna predstavitev teles v podatkovni strukturi. V preteklosti so za to nalogo razvili pet tako imenovanih predstavitevni metod, od katerih najpogosteje uporabljamo tri: predstavitev z ugotavljanjem zasedenosti prostora (angl. voxel representation), predstavitev s temeljnimi gradniki (angl. constructive solid geometry, CSG) in predstavitev z ovojnico (angl. boundary representation, B-rep) [1, 2, 3]. Izbor predstavitve je odvisen od področja uporabe, a predstavitev B-rep, ki, kot že ime pove, opiše površje teles, je najbolj vsestranska. Pravilno površje mora zadostiti pogojem o dobro oblikovanih površjih, in sicer [1]: biti mora zaprto (v inženirstvu pravimo, da je vodotesno), orientabilno, brez samopresečišč, omejujoče in povezano. Opis telesa, predstavljenega z B-rep, je sestavljen in dveh delov: iz opisa sosednostnih relacij med oglišči, robovi, lici, prstani in luknjami, kar v geometrijskem modeliranju krajše imenujemo kar topologija, ter iz določitve geometrijskih informacij, ki jim, spet

krajše, pravimo geometrija. Pravilno konstrukcijo topologije dosežemo z Eulerjevi operatorji [4], medtem ko je geometrija, umeščena nad topologijo, določena s koordinatami vozlišč ter implicitnimi in/ali različnimi tipi parametričnih krivulj in ploskev (na primer krivulje in ploskve NURBS). Prav ta vsestranskost predstavitve B-rep je tudi razlog, da je podatkovna struktura zapletena, implementacija algoritmov pa zahtevna. Napredek tehnologije je prinesel pomembno spremembo. S pojavom cenovno ugodnih 3D tiskalnikov je računalniško podprto izdelovanje teles pritegnilo mnogo uporabnikov, razvita pa je bila tudi množica komercialne in odprtokodne programske opreme. Sposobnosti posameznih programskih rešitev pa so praviloma različne, zato uporabniki pogosto izmenjujejo podatkovne datoteke. Žal se pri izmenjevanju pogosto zaplete, saj imajo geometrijski modelirniki različne domene teles, ki jih zmorejo predstaviti. Za izmenjevanje modelov teles so zato predlagali nevtralne podatkovne formate, kot sta specifikacija IGES in mednarodni standard STEP (ISO 10303) [5]. STEP je med najboljšežnejšimi standardi ISO in ga je zato v celoti zelo zahtevno implementirati. Zato so mnoge komercialne, predvsem pa prosto dostopne programske rešitve, raje posegle po drugi možnosti. Površje telesa interpolirajo z mnogokotniško mrežo, sestavljeno iz facet. Faceta je predstavljena z zaporedjem koplanarnih oglišč, ki tvorijo izbočen mnogokotnik. Facete so najpogostejše trikotniki, redkeje štirikotniki, facete z večjim številom oglišč pa so v geometrijskem modeliranju zelo redke. V tem prispevku se bomo omejili le na trikotnike, ki so tudi najpogostejše uporabljeni v praksi. Govorili bomo torej o trianguliranih površjih teles (TPT). Formati za njihovo shranjevanje so odprti, enostavni za razumevanje ter zato zelo pogosto v uporabi; primeri so STL (STereoLithography File Format Family), 3MF (3D Manufacturing Format), AMF (Additive Manufacturing File format), OBJ (Wavefront file format), PLY (Polygon File Format) in drugi. Za doseg primerne gladkosti ukrivljenih površij pa potrebujemo veliko število facet, kar vodi v velike datoteke, ki jih je smiselno pred arhiviranjem ali pošiljanjem preko spleta zmanjšati.

V tem članku predstavimo izboljšavo metode, objavljene v [6], ki iz trianguliranega površja teles odstrani nabor trikotnikov na način, da je odstranjene trikotnike mogoče enolično obnoviti. Posebnost te metode je, da

ne uvaja novega specifičnega formata, temveč omogoča shranjevanje v kateremkoli izmed prej omenjenih formatov.

Članek je sestavljen iz petih poglavij. V drugem poglavju na kratko predstavimo metode za stiskanje trianguliranih površij teles. V tretjem poglavju opišemo referenčno metodo in podamo njeno izboljšavo. V četrtem poglavju predstavimo rezultate ter jih komentiramo. Pri spevek zaključimo v petem poglavju, kjer podamo tudi usmeritve za nadaljnje izboljšave.

2 Predhodna dela

Prvo metodo za stiskanje TPT sta objavila Tuoma in Gotsman [7]. Metoda najprej vsakemu oglišču določi stopnjo, ki ustreza številu robov, stikajočih se v oglišču. Če naleti na nevodotesni del površja, vstavi navidezna oglišča in navidezne trikotnike. Metoda zatem ločeno obravnava topologijo in geometrijo. Topologijo TPT opiše s potovanjem skozi trikotniško mrežo, za kar potrebuje štiri ukaze. Dobljeno zaporedje ukazov in koordinate oglišč nazadnje ločeno stisne s splošnonamenskim algoritmom stiskanja. Gumhold in Straßer [8] sta predstavila metodo, temelječo na rezanju mej. Meja deli TPT v dva dela: tistega, ki je že zakodiran, in preostanek. V vsakem koraku dodamo v mejo nov trikotnik, pri čemer metoda s sedmimi ukazi razrešuje možna stanja. Rossignac je predstavil algoritem, ki sistematično obdeluje TPT s tako imenovanim lomljenjem robov [9]. Pri odstranitvi roba lahko TPT razpade v več regij. Regije se lahko stikajo v ogliščih, ne pa tudi v robovih. Algoritem nadzira obhod s petimi ukazi. Izboljšavo metode najdemo v [10]. Alliez in Desbrun [11] sta nadgradila metodo iz [7]. Uvedla sta heuristiko, ki za naslednje oglišče, ki ga bomo sprejeli, izbere oglišče z najmanjšim številom prostih robov. Zmanjšala sta tudi število navideznih oglišč v primeru, ko površje ni vodotesno. Podobno idejo so objavili v [12]. Časovno učinkovit algoritem, delujoč v linearnem času, najdemo v [13]. Vsi omenjeni pristopi so ločeno stiskali geometrijo in topologijo TPT. Coors in Rossignac [14] pa sta predlagala pristop, ki na podlagi analize notranjih kotov trikotnikov napove, kako je trikotnik povezan v mrežo, s čimer se razmerje stiskanja občutno izboljša. Algoritem, ki hkrati obdeluje tako trikotne kot štirikotne facete, najdemo v [15], pri čemer prav tako izkorišča notranje kote facet za napoved položaja koordinat naslednjih oglišč. V članku [16] ugotavljajo, da je položaje oglišč v TPT smiselno obravnavati kot oblak točk in jih stisniti z metodami za stiskanje oblakov točk, nato pa izkoristiti informacijo o položajih točk pri stiskanju topologije. Metoda za napovedovanje povezljivosti oglišča v TPT izboljša metodo iz [9]. V članku [17] je opisana metoda, ki izboljša napoved povezave oglišča v TPT z zamenjavo vnaprej definiranega obhoda z obhodom, ki temelji na vrsti s prednostjo. Na vrh vrste s prednostjo postavijo tisto oglišče, katerega verjetnost za vključitev v TPT je najvišja. Avtorji uvedejo tudi novo metriko za ocenitev kakovosti napovedi.

Učinkovitost metod se je izboljševala, a nobena ni pritegnila praktične uporabe. Razlogov za to je več [6]:

- vsaka metoda uvaja svoj nabor ukazov, zaradi česar so formati različni in med seboj nezdružljivi;
- z večjo učinkovitostjo se povečuje tudi težavnost implementacije;
- kodirnik in dekodirnik sta pri vseh metodah simetrična; če pride do napake v enem izmed njiju, je pravilna rekonstrukcija TPT nemogoča.

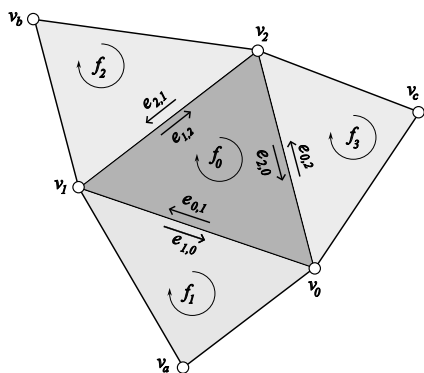
3 Izboljšano redčenje trianguliranih površij teles z odstranjevanjem obnovljivih trikotnikov

Postopek stiskanja TPT, predstavljen v [6], temelji na predpostavki, da je obravnavano telo fizično izvedljivo ter da izpolnjuje pogoje dobro oblikovanih površij, omenjene v uvodnem poglavju – natančneje pogoja vodotesnosti in orientabilnosti. Postopek, imenovan CoToS (angl. *Compression of Triangulated Solids' Surfaces by Decimating Reconstructable Triangles*), uvaja koncepte kodirnika in dekodirnika.

Kodirnik CoToS opravi postopek redčenja trikotnikov tako, da TPT postane nevodotesno, vendar na način, ki omogoča ponovno vzpostavitev vodotesnosti in pravilne orientiranosti restavriranih trikotnikov. Slika 1 prikazuje izsek iz nekega TPT. Trikotnik oziroma faceta f_0 (narisana temneje) ima tri sosedne trikotnike f_1 , f_2 in f_3 (narisane svetleje), ki si paroma delijo tri skupne robove. Posledično je ta del TPT vodotesen. Velja torej naslednje pravilo [1]: površje TPT je vodotesno, če vsak njegov rob pripada natanko dvema trikotnikoma. V konkretnem primeru bi kodirnik lahko odstranil trikotnik f_0 , saj bo dekodirnik iz obstoječe konfiguracije robov prepoznal, da je bil ta trikotnik odstranjen. Za enolično restavracijo odstranjenega trikotnika morajo trikotniki f_1 , f_2 in f_3 v razredčeni mreži zato ostati. Algoritem kodirnika CoToS je preprost in deluje v naslednjih korakih:

- Preberi podatke o TPT iz enega izmed obstoječih odprtih formatov.
- Označi vse trikotnike za neobiskane.
- Izberi poljuben neobiskan trikotnik in ga označi kot obiskanega.
- Če so vsi njegovi sosedi označeni kot neobiskani ali neodstranjeni, označi trikotnik kot odstranjenega, njegove sosede pa kot obiskane in neodstranjene; sicer označi trikotnik kot neodstranjenega.
- Postopek ponavljaj, dokler niso obiskani vsi trikotniki.
- Zapiši trikotnike, ki so označeni kot neodstranjeni, v enega izmed obstoječih odprtih formatov.

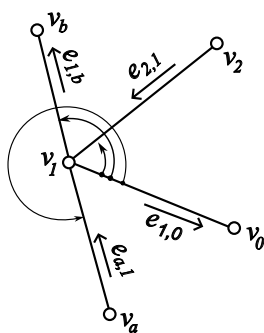
Vodotesen izsek iz TPT na sliki 1 je tudi pravilno orientiran, saj so vsi trikotniki orientirani enako. Pravimo, da ima vsak rob v TPT dva polroba, pri čemer vsak polrob pripada natanko enemu trikotniku. Na primer, trikotnika f_0 in f_2 si delita skupen rob med ogliščema v_1 in v_2 .



Slika 1: Vodotesen del trianguliranega površja telesa.

V trikotniku f_0 je ta rob predstavljen s polrobom $e_{1,2}$, v trikotniku f_2 pa s polrobom $e_{2,1}$, kot je shematsko ponazorjeno s puščico na sliki 1. Tako imenovano Möbiusovo pravilo pravi, da je površje orientabilno, če sta polroba, ki pripadata istemu robu, orientirana v nasprotnih smereh [1].

Orientabilnost in vodotesnost sta lastnosti, ki se v fazi kodiranja CoToS izgubita, zato ju mora dekodirnik ponovno vzpostaviti. Najprej iz ustreznega odprtega podatkovnega formata prebere podatke o ohranjenih trikotnikih, iz njih izlušči polrobove in jih shrani v ustrezno iskalno podatkovno strukturo. Polrobovi, ki imajo pripadajoči nasprotni polrob, pripadajo že vodotesnemu delu TPT, zato jih odstranimo. Preostale polrobove uporabimo za obnovitev odstranjenih trikotnikov. Postopek bomo razložili s sliko 2.



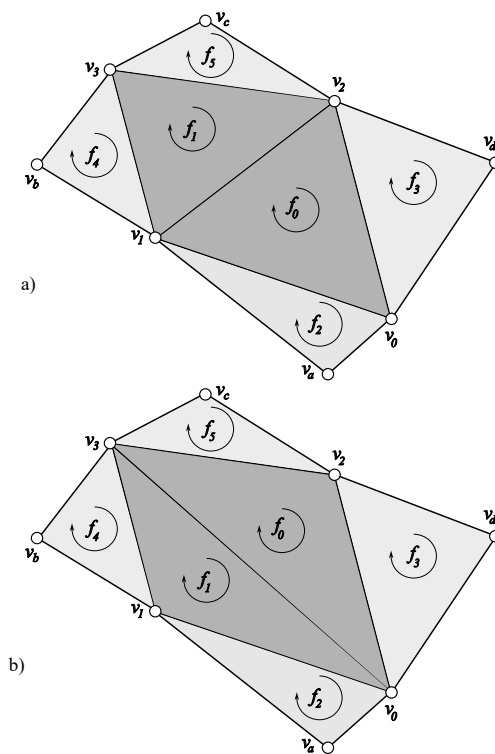
Slika 2: Postopek obnove odstranjenega trikotnika.

Dekodirnik CoToS najprej združi polrobove okrog posameznih vozlišč iz TPT. Na sliki 2 je prikazan primer, ko so polrobovi združeni okrog vozlišča v_1 . Izberemo poljuben referenčni polrob, na primer polrob $e_{1,0}$ in glede nanj izračunamo kote v matematično pozitivni smeri do vseh polrobov, ki se stikajo v v_1 . Izberemo polrob z najmanjšim kotom, to je polrob $e_{2,1}$, in se preselimo v oglišče v_2 , kjer ponovimo postopek z referenčnim polrobom $e_{2,1}$. Tudi v tem vozlišču bi izračunali kote, polrob z najmanjšim kotom v matematično pozitivni smeri pa bi bil polrob $e_{2,0}$. S tem bi prišli na začetek prvega izbranega polroba, to je dosegli bi oglišče v_0 . Zaporedje v_1 , v_2 in v_0 določa že pravilno orientiran obnovljen trikotnik f_0 . Obiskane polrobove $e_{1,0}$, $e_{0,2}$ in $e_{2,1}$ odstranimo

iz iskalne podatkovne strukture, nato pa se postopek ponovi, dokler niso obnovljeni vsi trikotniki oziroma dokler v podatkovni strukturi ostaja še kakšen polrob.

3.1 Izboljšava postopka CoToS

V nadaljevanju predstavimo izboljšavo postopka CoToS, ki jo bomo označili kot CoToS⁺. V mnogih primerih bi lahko iz TPT hkrati odstranili dva sosednja trikotnika, če bi le imeli ustrezen kriterij, kako ju nedvoumno rekonstruirati. Slika 3 kaže dve možni konfiguraciji dveh sosednjih trikotnikov f_0 in f_1 . Dva sosednja trikotnika tvorita par trikotnikov; v konkretnem primeru ta sestoji iz oglišč v_0 , v_1 , v_3 in v_2 . Takšen par trikotnikov ima lahko dve različni diagonali, kot je razvidno na slikah 3a in 3b. Kodirnik CoToS⁺ v primeru, ko je dejanska konfiguracija takšna, da je par trikotnikov razdeljen s strogo krajšo diagonalo, hkrati odstrani iz TPT dva trikotnika. Takšno konfiguracijo vidimo na sliki 3a.



Slika 3: Dve možni konfiguraciji trikotnikov.

Vsi ostali pogoji za redčenje trikotnikov so enaki kot pri osnovnem postopku CoToS. Dekodirnik CoToS⁺ deluje enako kot CoToS, le da v primeru, ko zazna, da je bil rekonstruiran par trikotnikov, vstavi vanj krajšo diagonalo in tako pravilno obnovi dva trikotnika v TPT. Programski koda je dostopna na¹.

4 Rezultati

Primerjava redčenja, dobljena s postopkoma CoToS in CoToS⁺, je prikazana na sliki 4, kjer so bila temna področja odstranjena. V tabeli 1 vidimo primerjavo učinkovitosti redčenja na telesih v formatu OBJ, dostopnih

¹<https://github.com/FeriGeMMA/CoToS>



Slika 4: Rezultat redčenja s CoToS (levo) in CoToS⁺ (desno) za isto področje na telesu.

na². V večini primerov je postopek CoToS⁺ učinkovitejši, vendar ne vedno. Kadar model vsebuje veliko ozkih in dolgih trikotnikov, kot prikazuje slika 3b, je skupna dolžina robov pogosto večja. V takšnih primerih dodatnega trikotnika ni mogoče odstraniti. Če pa do odstranitve vseeno pride, mora v okolici odstranjenega para trikotnikov ostati dovolj sosednjih trikotnikov za ohranitev topološke pravilnosti. Modeli, ustvarjeni z, na primer, Delaunayevim postopkom, vsebujejo veliko skupnih robov, kar omogoča učinkovitejše delovanje metode CoToS⁺. To je razvidno tudi pri modelu Zajec v zadnji vrstici tabele 1. Preverili smo tudi, kakšen vpliv na učinkovitost metode ima vrstni red trikotnikov. Testi (stolpec E v tabeli 1) so pokazali, da vrstni red trikotnikov nima praktičnega vpliva na učinkovitost, saj je standardni odklon manjši od enega odstotka.

Tabela 1: Število trikotnikov v: **A** - TPT, **B** - CoToS, **C** - CoToS⁺, **D** - % izboljšave, % standardnega odklona v stolpcu C glede na različen vrstni red trikotnikov.

Ime	A	B	C	D	E
Pasavec	99976	58958	57020	3.29	0.12
Ohišje	102490	58037	56284	3.02	0.13
Kocka	187482	111581	112148	-0.51	0.06
Motorj	86686	51551	49752	3.49	0.14
Stator	218816	120049	115330	3.93	0.09
Ročaj	89512	52015	50354	3.19	0.15
Obroč	18068	10722	10760	-0.35	0.21
Zajec	560968	328214	296470	9.67	0.05

5 Zaključek

V članku smo predstavili novo metodo za odstranjevanje obnovljivih trikotnikov iz trianguliranih površij teles. Metoda uvaja dodatni kriterij za odstranjevanje trikotnikov, pri čemer ne zahteva dodatnih informacij. Eksperimenti so pokazali, da lahko pričakujemo približno 3% izboljšanje uspešnosti glede na [6]. Uspešnost metode je odvisna od kakovosti triangulacije: triangulacije z velikim številom dolgih in ozkih trikotnikov dajo slabše rezultate, medtem ko so rezultati boljši pri triangulacijah s bolj enakostraničnimi trikotniki. Glede na to, da

COTOS⁺ zmanjša število robov, bi ga bilo morda smiselno uporabiti pred algoritmom Edgebreaker [9] in njegovih izboljšavah, kar bomo preverili v prihodnosti.

Zahvala

Projekt (Paradigma stiskanja podatkov z odstranjevanjem obnovljivih informacij, št. J2-4458) sta financirali Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije ter Czech Science Foundation (project No. 23-04622L) iz državnih proračunov.

Literatura

- [1] M.E. Mortenson: Geometric modeling. Wiley, 1985.
- [2] C.M. Horrmann: Geometric and solid modeling: An introduction. Morgan Kaufmann, 1989.
- [3] B. Žalik: Geometrijsko modeliranje, UM FER, 1999.
- [4] M. Mäntylä: An introduction to solid modeling. Rockville: Computer Science Press, 1988.
- [5] P. Kurylo, P. Frankovský, M. Malinowski, T. Maciejewski, J. Varga, J. Kostka, L. Adrian, S. Szufa S, S. Rusnáková: Data exchange with support for the neutral processing of formats in Computer Aided Design/Computer-Aided Manufacturing Systems. Appl Sci, 13(17):9811, 2023.
- [6] B. Repnik, B. Žalik, K. Rizman Žalik: Compression of triangulated solids' surfaces by decimating reconstructable triangles. SoftwareX, 30:102179, 2025.
- [7] C., Tuoma, C. Gotsman: Triangle mesh compression. Graphics interfaces '98, 26–34 1998.
- [8] S. Gumhold, W. Straßer: Real time compression of triangle mesh connectivity. SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, 133–140, 1989.
- [9] J. Rossignac: Edgebreaker: connectivity compression for triangle meshes. IEEE Trans. Vis. Comput. Graph, 5(1):47–61, 1999.
- [10] L. Ying, H. Zhongming, D. Mingli, D. Dagao: An Edgebreaker & code-mode based connectivity compression for triangular meshes. 2nd International Conference on Advanced Computer Control, 2, 96–101, 2010.
- [11] P. Alliez, M. Desbrun: Valence-driven connectivity encoding for 3D meshes. Comput. Graph. Forum, 20(3):480–489, 2001.
- [12] F. Kälberer, K. Polthier, U. Reitebuch, M. Wardetzky: FreeLence – Coding with free valences. Comput. Graph. Forum, 24(3):469–478, 2005.
- [13] K. Mamou, T. Zaharia, F. Preteux: TFAN: A low complexity 3D mesh compression algorithm. Comput. Animat. Virtual Worlds, 20(2-3):343–354, 2009.
- [14] V. Coors, J. Rossignac: Delphi: Geometry-based connectivity prediction in triangle mesh compression. Vis. Comput., 20(8):507–520, 2004.
- [15] H. Lee, P. Alliez, M. Debrun: Angle-analyzer: A triangle-quad mesh codec. Comput. Graph. Forum, 21(3):383–392, 2002.
- [16] P. Marais, J. Gain, D. Shreiner: Distance-ranked connectivity compression of triangle meshes. Comput. Graph. Forum, 26(4):813–823, 2007.
- [17] J. Dvořák, Z. Káčereková, P. Vaněček in L. Váša: Priority-based encoding of triangle mesh connectivity for a known geometry. Comput. Graph. Forum, 42(1):60–71, 2023.

²https://github.com/FerigeMMA/CoToS_models