

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 683

**POSTUPCI SEGMENTACIJE  
OBJEKATA ZADANIH  
POLIGONALNOM MREŽOM**

Damjan Križaić

Zagreb, lipanj 2014.

*Umjesto ove stranice umetnite izvornik Vašeg rada.  
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*



# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Postupci segmentacije CAD modela</b>	<b>2</b>
2.1. Segmentacija . . . . .	2
2.2. CAD modeli . . . . .	3
2.2.1. Krivulje NURBS . . . . .	3
2.2.2. Volumni elementi . . . . .	4
2.2.3. Poligonalna mreža . . . . .	4
2.3. Interaktivna segmentacija poligonalnih mreža . . . . .	10
2.4. Automatska segmentacija poligonalnih mreža . . . . .	10
2.4.1. Podjela i opisi postupaka . . . . .	11
2.4.2. Hijerarhijska neizrazita segmentacija poligonalnih mreža . . . . .	17
2.4.3. Usporedba postupaka . . . . .	25
<b>3. Primjena segmentacije objekata zadanih poligonalnom mrežom</b>	<b>27</b>
3.1. Određivanje skeletne strukture . . . . .	27
3.2. Uklanjanje šuma i popravak modela . . . . .	29
3.3. Pojednostavljenje poligonalne mreže i kompresija . . . . .	30
<b>4. Programska implementacija</b>	<b>31</b>
4.1. Programski alati . . . . .	31
4.2. Rezultati . . . . .	35
4.3. Analiza . . . . .	37
<b>5. Zaključak</b>	<b>40</b>
<b>Literatura</b>	<b>41</b>

# 1. Uvod

Vrijeme je najvrijedniji resurs kojim ljudi raspolažu. Iz tog razloga poželjno je koristiti ga čim efikasnije.

Računala nam pomažu automatizirati neke poslove čime ostvarujemo iste rezultate u kraćem vremenu.

U području računalne grafike mnogi poslovi se obično obavljaju ručno. Tu spadaju modeliranje, popravak modela, određivanje skeletne strukture, animiranje modela itd. Za neke od spomenutih poslova potrebna je segmentacija modela koja se također obično obavlja ručno. Sve te postupke je poželjno automatizirati ili barem olakšati njihovo provođenje.

Ovaj rad proučava automatizirane postupke segmentacije i njihovu primjenjivost u kontekstu CAD modela. Posebna pažnja je posvećena modelima koji su zadani poligonalnom mrežom.

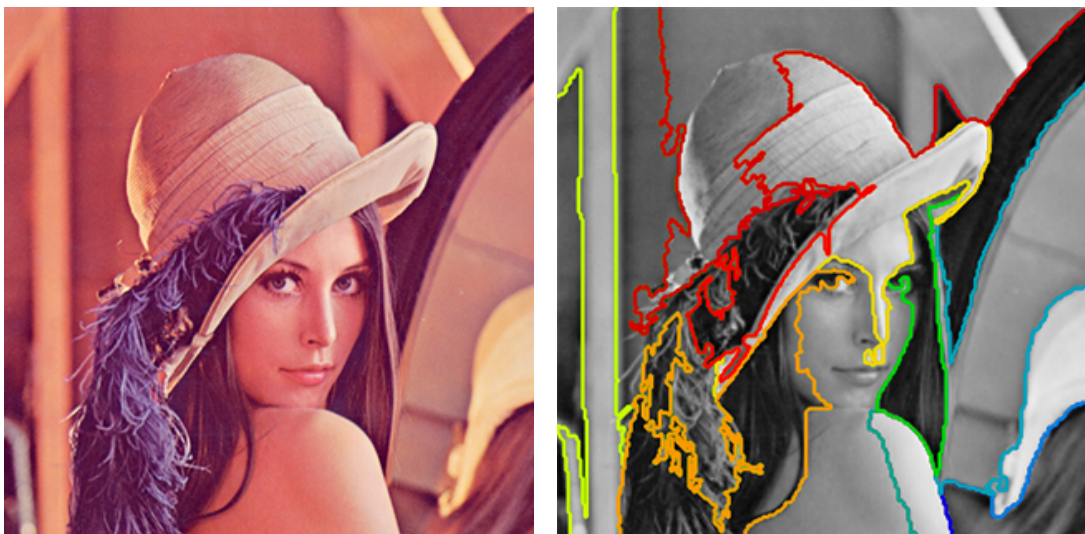
## 2. Postupci segmentacije CAD modela

### 2.1. Segmentacija

Pojam **segmentacije** ili **dekompozicije** u računarskoj znanosti predstavlja proces rastavljanja kompleksnog sustava ili problema na više manjih, međusobno nepreklapajućih cjelina koje je lakše pojmiti, shvatiti i analizirati. Prikaz modela koji predstavlja neki sustav ili problem se pojednostavljuje u više manjih, smislenih prikaza.

Problem kojeg segmentacija rješava je reverzan problemu stvaranja većeg sustava pomoću jednostavnih elemenata.

Rastavljanje kompleksnih modela u manje cjeline je temeljni problem u raznim disciplinama. Kod obrade slika, slike se segmentiraju u regije, što prikazuje slika 2.1. To je važan korak u pretprocesiranju slika za algoritme iz područja računalnog vida. U računalnoj grafici se CAD modeli segmentiraju u manje skupove elemenata od kojih su građeni.



Slika 2.1: Segmentacija slike

## 2.2. CAD modeli

CAD je akronim za *Computer-aided design* tj. za oblikovanje pomoću računala. To su postupci računalne izrade 3D modela, po potrebi dinamičnih (animiranih), sa svim podacima potrebnima za daljnju obradu.

Postoje tri glavna tipa CAD modela. Prvi tip su objekti sastavljeni od površina tzv. primitiva (ravnine, sfere, valjci, itd.). Drugi tip su modeli koji su sastavljeni od površina slobodnog oblika. Treći tip su kombinacija prethodna dva.

Zapis CAD modela ovisi o primjeni. Najčešći su:

- krivulje NURBS
- volumni elementi
- poligonalna mreža

### 2.2.1. Krivulje NURBS

**Krivulje NURBS** (engl. *non-uniform rational B-spline*) koriste B-krivulje za definiciju i zapis krivulja i površina. Krivulja je definirana skupom težinskih kontrolnih točaka, redom i vektorom uzlova.

**Težinske kontrolne točke** određuju oblik krivulje. Svaka kontrolna točka utječe samo na intervale krivulje na kojima je aktivna. To omogućava da se prilikom modeliranja mijenja samo jedan dio površine dok ostali ostaju nepromijenjeni.

**Red krivulje** označava broj susjednih kontrolnih točaka koje utječu na točke krivulje. U praksi se najčešće koriste krivulje trećeg reda.

**Vektor uzlova** sadrži parametre koji određuju kako i gdje kontrolne točke utječu na NURBS krivulju.

NURBS krivulje i površine se koriste iz više razloga:

- invarijantne su na operacije poput translacije i rotacije (operacije utječu na cijelu NURBS krivulju tako da se primjene samo na kontrolne točke NURBS krivulje)
- pružaju veliku slobodu pri dizajniranju proizvoljnih oblika
- koriste manje memorije od jednostavnijih metoda za zapis kompleksnih modela
- mogu se analizirati relativno brzo pomoću numerički stabilnih i preciznih algoritama

Neki od postupaka segmentacije koji koriste krivulje NURBS su (Chen i Liu, 1997), (Richtsfeld et al., 2012) i (Suinesiaputra et al., 2009).

## 2.2.2. Volumni elementi

**Volumni element** ili *voxel* predstavlja neki podatak ili uzorak u pravilno razmaknutoj trodimenzionalnoj mreži. Podaci obično sadrže prozirnost i boju. Također mogu, ovisno o primjeni, sadržavati vektore i tenzore podataka. Sami volumni elementi ne sadrže koordinate svojih pozicija. Njihove pozicije se zaključuju prema susjednim volumnim elementima. Volumen kojeg mreža predstavlja nije nigdje zapisan, već se mora rekonstruirati ili aproksimirati raznim metodama poput interpolacije.

*Voxeli* se koriste u računalnim igrama (najčešće za prikaz terena) te u medicini za CT, MRI i ultrazvučne snimke.

Neki od postupaka segmentacije koji koriste volumne elemente su (Sherbondy et al., 2003), (Pražni et al., 2010) i (Shen et al., 2011).

## 2.2.3. Poligonalna mreža

**Poligonalna mreža** (engl. *polygon mesh*) je skup elemenata koji definiraju oblik nekog 3D modela. Slika 2.2 ilustrira elemente poligonalne mreže. Elementi mogu biti:

- **vrh** (engl. *vertex*)
- **brid** (engl. *edge*)
- **poligon** (engl. *face*)

Podaci koji opisuju elemente mreže su povezanost i geometrija mreže.

**Povezanost** ili **topologija** mreže opisuju odnose između elemenata mreže (npr. koji vrhovi su susjedni, koji bridovi čine poligon, itd.).

**Geometrija** mreže određuje poziciju i ostale geometrijske karakteristike vrhova.

Mreža je tzv. *manifold* ako za nju vrijedi:

- svaki brid je vezan samo za jedan ili dva poligona
- poligoni koji dijele jedan vrh stvaraju zatvorenu ili otvorenu lepezu poligona

**Orijentaciju** poligona određuje ciklički redoslijed povezanih vrhova.

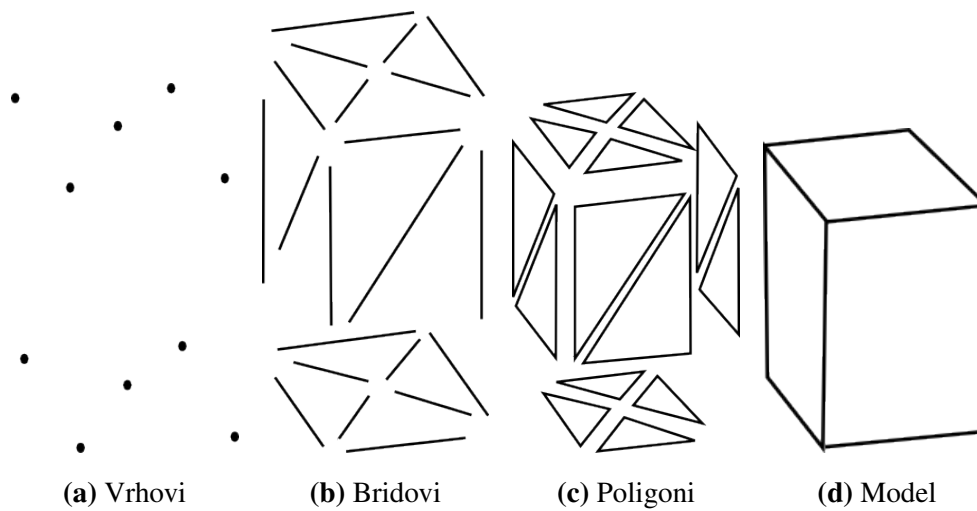
Orijentacija dvaju susjednih poligona je **kompatibilna** ako su dva vrha na zajedničkom bridu obrnutog redoslijeda.

*Manifold* mreža je **orijentabilna** ako bilo koja dva susjedna poligona imaju kompatibilnu orijentaciju.

Poligonalna mreža je **pravilna** ako svi poligoni imaju jednak broj bridova i svi vrhovi su vezani za jednak broj bridova.

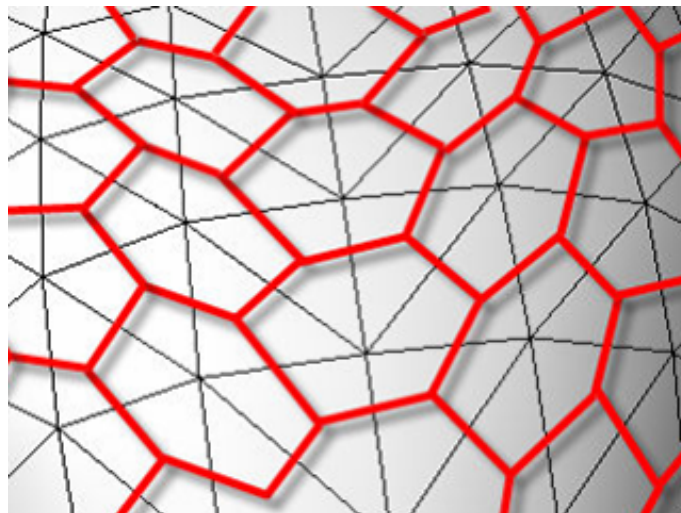
**Konveksna** poligonalna mreža je ona unutar koje se može povući pravac koji će uvijek presijecati mrežu u najviše dvije točke.





**Slika 2.2:** Elementi poligonalne mreže

**Dualni graf** poligonalne mreže je graf čiji čvorovi predstavljaju poligone. Čvorovi su povezani ako poligoni koje predstavljaju dijele brid. Slika 2.3 crvenom bojom prikazuje dualni graf.



**Slika 2.3:** Dualni graf poligonalne mreže

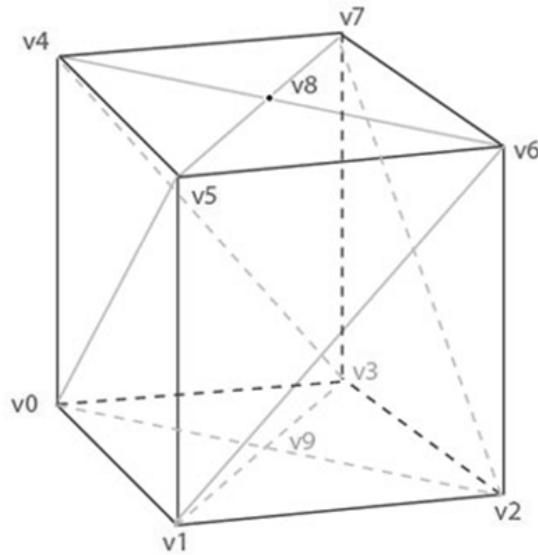
### Zapisi poligonalnih mreža

Odnose između vrhova, bridova i poligona unutar poligonalne mreže moguće je zapisati na razne načine. Svaki od zapisa ima svoje prednosti i nedostatke.

U nastavku su opisani najpoznatiji načini zapisa poligonalnih mreža.

**Vertex-vertex mreža** Zapis koji sadrži samo vrhove koji pokazuju na druge vrhove. Informacije o bridovima i poligonima su implicitno zadane. Ovaj zapis je ilustriran na slici 2.4.

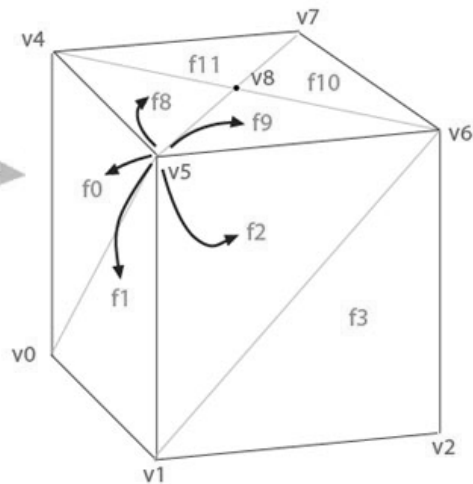
v0	0,0,0	v1 v5 v4 v3 v9
v1	1,0,0	v2 v6 v5 v0 v9
v2	1,1,0	v3 v7 v6 v1 v9
v3	0,1,0	v2 v6 v7 v4 v9
v4	0,0,1	v5 v0 v3 v7 v8
v5	1,0,1	v6 v1 v0 v4 v8
v6	1,1,1	v7 v2 v1 v5 v8
v7	0,1,1	v4 v3 v2 v6 v8
v8	.5,.5,1	v4 v5 v6 v7
v9	.5,.5,0	v0 v1 v2 v3



Slika 2.4: Vertex-vertex mreža

**Face-vertex mreža** Lista vrhova s koordinatama te poligona koji pokazuju na vrhove od kojih su sastavljeni. Slika 2.5 ilustrira ovaj zapis.

f0	v0 v4 v5	v0	0,0,0	f0 f1 f12 f15 f7
f1	v0 v5 v1	v1	1,0,0	f2 f3 f13 f12 f1
f2	v1 v5 v6	v2	1,1,0	f4 f5 f14 f13 f3
f3	v1 v6 v2	v3	0,1,0	f6 f7 f15 f14 f5
f4	v2 v6 v7	v4	0,0,1	f6 f7 f0 f8 f11
f5	v2 v7 v3	v5	1,0,1	f0 f1 f2 f9 f8
f6	v3 v7 v4	v6	1,1,1	f2 f3 f4 f10 f9
f7	v3 v4 v0	v7	0,1,1	f4 f5 f6 f11 f10
f8	v8 v5 v4	v8	.5,.5,1	f8 f9 f10 f11
f9	v8 v6 v5	v9	.5,.5,0	f12 f13 f14 f15
f10	v8 v7 v6			
f11	v8 v4 v7			
f12	v9 v5 v4			
f13	v9 v6 v5			
f14	v9 v7 v6			
f15	v9 v4 v7			



Slika 2.5: Face-vertex mreža

**Winged-edge mreža** Zapis kod kojeg bridovi pokazuju na dva vrha, dva poligona i četiri brida koja dodiruju. Pogodni su za prolazak mrežom i promjenu geometrije u konstantnom vremenu. Nedostatak su im veći memorijski zahtjevi.

**Half-edge mreža** Slično *winged-edge* mreži, uz razliku da se koristi samo pola informacija za prolazak bridovima.

**Quad-edge mreža** Mreža zapisana vrhovima, polu-bridovima i bridovima. Poligoni su zadani implicitno i dobivaju se prolaskom kroz strukturu.

**Corner-table** Tablica koja pohranjuje vrhove na način da prolaz tablicom implicitno definira poligone. Koriste se za efikasan dohvat poligona. Nedostaci su spore promjene u strukturi i činjenica da jedna tablica ne predstavlja poligonalnu mrežu u potpunosti. Za predstavljanje poligonalne mreže u potpunosti potrebno je koristiti više tablica.

**Render dynamic** Kombinacija *face-vertex* mreže i *winged-edge* mreže. Eksplicitno se zapisuju vrhovi poligona i poligoni vrhova kao u slučaju *face-vertex* mreže te vrhovi i poligoni bridova kao kod *winged-edge* mreže. Svaki prolaz i promjena geometrije se odvija u konstantnom vremenu.

### **Datotečni formati poligonalnih mreža**

Za pohranu poligonalnih mreža u stalnu memoriju služe datotečni formati. Svaki datotečni format je najbolje koristiti u kontekstu za koji je zamišljen od strane autora. Najpoznatiji datotečni formati su:

- Wavefront OBJ (.obj)
- OFF Geometry Format (.off)
- Polygon File Format (.ply)
- STereoLithography (.stl)
- Digital Asset Exchange (COLLADA) (.dae)
- Autodesk FBX Format (.fbx)

U nastavku su detaljnije opisani Wavefront OBJ i OFF Geometry Format koji su korišteni u programskoj implementaciji segmentacijskog postupka.

**Wavefront OBJ** OBJ je jednostavan datotečni format koji opisuje samo 3D geometriju modela u `.obj` datoteci. Svi vrhovi poligona su navedeni redoslijedom obrnutim od kazaljke na satu što ujedno i određuje orijentacije normala poligona. Slika 2.6 prikazuje strukturu `.obj` datoteke.

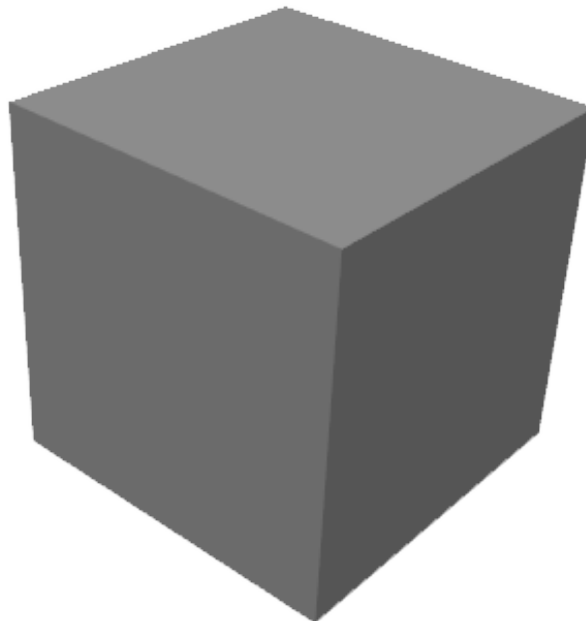
**OFF geometry format** OFF je datotečni format definicije geometrije koji sadrži opis poligona koji čine 3D model. Prvi redak zapisa definira broj vrhova, poligona i bridova. Nakon toga slijede redci s  $x$ ,  $y$ , i  $z$  koordinatama vrhova poligona. Kraj datoteke čini popis poligona koji su definirani brojem vrhova i slijedom indeksa vrhova od kojih su sastavljeni. Slika 2.7 sadrži primjer zapisa modela u `.off` datoteci koji je učitani i iscrtan na slici 2.8.

```
# vrhovi s definiranim koordinatama
v 0.111 0.222 0.333 1.0
v ...
# koordinate tekstura
vt 0.500 1 [0]
vt ...
# normale
vn 0.707 0.000 0.707
vn ...
# vrhovi zadani u parametarskom prostoru
vp 0.110000 3.300000 2.200000
vp ...
# definicije poligona
# niz vrhova koji definiraju poligon
f 1 2 3
# niz vrh/koordinata teksture
f 3/1 4/2 5/3
# niz vrh/koordinata teksture /normala
f 6/4/1 3/5/3 7/6/5
f ...
```

**Slika 2.6:** Primjer OBJ datoteke

```
OFF
8 6 0
-0.500000 -0.500000 0.500000
0.500000 -0.500000 0.500000
-0.500000 0.500000 0.500000
0.500000 0.500000 0.500000
-0.500000 0.500000 -0.500000
0.500000 0.500000 -0.500000
-0.500000 -0.500000 -0.500000
0.500000 -0.500000 -0.500000
4 0 1 3 2
4 2 3 5 4
4 4 5 7 6
4 6 7 1 0
4 1 7 5 3
4 6 0 2 4
```

**Slika 2.7:** Primjer OFF datoteke (kocka.off)



**Slika 2.8:** Prikaz učitano modela iz datoteke kocka.off

## 2.3. Interaktivna segmentacija poligonalnih mreža

Pod pojmom interaktivne segmentacije poligonalne mreže se zapravo smatra ručna segmentacija od strane čovjeka. Čovjek sam procjenjuje broj i granice segmenata. 3D modeli se ručno segmentiraju u raznim programskim alatima koji dopuštaju kreiranje proizvoljnih selekcija. Tu spadaju skoro svi alati razvijeni za 3D modeliranje.

Prednost ovog postupka je velika točnost segmentacije s finim granicama i bez neželjenih segmenata. Čovjek lako procijeni u koje segmente je prikladno segmentirati neki model. Očiti nedostatak ovog pristupa je vrijeme potrebno da se segmentira neki složeni model.

Još jedan način interaktivne segmentacije je modifikacija struktura podataka koje opisuju dijelove poligonalne mreže u za to prikladnim datotečnim formatima. Najpoznatiji za to prikladni formati su VRML (engl. *Virtual Reality Modeling Language*) i njegov nasljednik X3D. Ovaj način je dosta neintuitivan i vremenski još zahtjevniji od ručne segmentacije pomoću CAD alata.

## 2.4. Automatska segmentacija poligonalnih mreža

Proteklih godina je razvijeno mnoštvo segmentacijskih postupaka koji su uz minimalnu intervenciju korisnika u stanju *dovoljno dobro* segmentirati poligonalne mreže. Ovdje *dovoljno dobro* znači da je automatski dobivena segmentacija jako slična onoj dobivenoj ručno. U (Chen et al., 2009) je provedeno istraživanje u kojem se usporedilo ručno i automatsko segmentiranje 3D modela. Preko 4300 ručnih segmentacija je dobiveno na 380 modela zadanih poligonalnom mrežom. Rezultati pokazuju da su ručno dobivene segmentacije od različitih ljudi međusobno jako slične. Također, pokazuju se neke pravilnosti u granicama dobivenim ručnom segmentacijom. Te granice se obično nalaze na konkavnih granicama poligonalne mreže koje neki segmentacijski postupci i traže.

Prednost segmentacijskih postupaka je najprije ušteda vremena. Osim toga, moguća su proširenja postupaka za određene primjene. Primjene segmentacijskih postupaka su opisane u poglavlju 3. Neke primjene bi teško ili skoro nemoguće bilo provoditi ručno.

Najveći problem segmentacijskih postupaka je rastaviti model u *smislene* cjeline. *Smislene* cjeline su one koje se perceptivno mogu razlikovati od ostatka modela. Ostali problemi koji se obično javljaju su presegmentiranost i zupčaste granice segmentiranih cjelina.

### 2.4.1. Podjela i opisi postupaka

Segmentacijske postupke moguće je svrstati u kategorije prema različitim kriterijima.

Jedna od mogućih podjela segmentacijskih postupaka je na čisto geometrijske i na semantički orijentirane.

**Geometrijski orijentirani postupci** bi bili oni koji daju poligonalne mreže segmentirane u zakrpe koje su uniformne u odnosu na neko svojstvo, npr. na zakrivljenost ili udaljenost do neke ravnine. Ovi postupci se mogu koristiti za pretprocesiranje modela koji bi u nastavku bili segmentirani semantički orijentiranim postupcima.

**Semantički orijentirani postupci** pokušavaju identificirati dijelove modela koji predstavljaju njegove značajke. Oni se lako proširuju da bi se koristili i za druge primjene. Prikladni su za određivanje skeletne strukture, metamorfozu modela, dohvat sličnih oblika, itd.

Druga moguća podjela je prema načinu aproksimiranja segmenata modela. Postoje postupci koji aproksimiraju regije standardnim CAD površinama i postupci koji to čine volumetrijskim primitivima.

**Postupci temeljeni na površinama** segmentiraju poligonalnu mrežu u segmente koji se mogu aproksimirati primitivima poput ravnina, sfera, valjaka itd.

**Postupci temeljeni na dijelovima** segmentiraju poligonalnu mrežu u segmente koji se mogu aproksimirati volumetrijskim primitivima poput super-elipsoida.

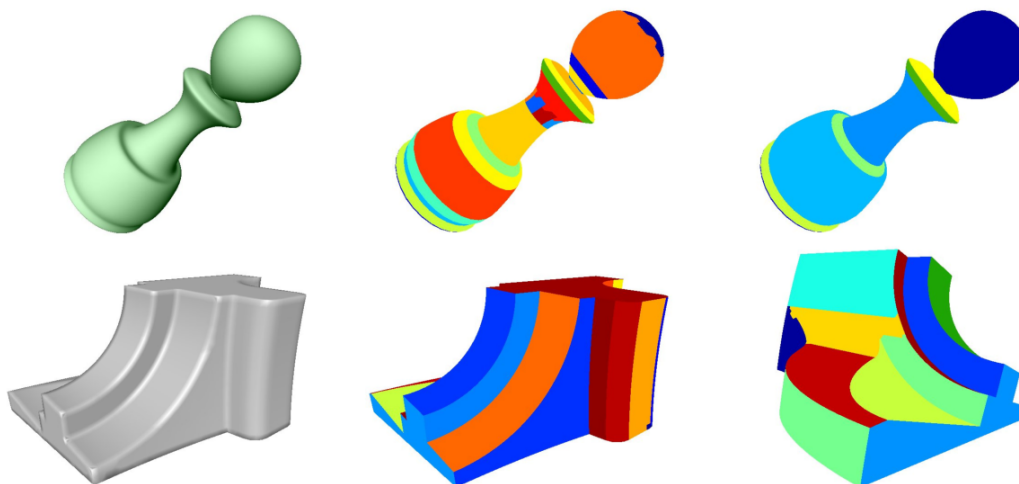
U (Agathos et al., 2007) segmentacijski postupci su podijeljeni prema temeljnim metodama. Prema toj podjeli dobivene su kategorije postupaka koje su u nastavku detaljnije opisane.

#### **Postupci temeljeni na rastućima regijama**

Postupci koji se služe rastućim regijama najprije odrede početne vrhove, poligone ili skupove poligona. Ti početni elementi predstavljaju regije koje se iterativno povećavaju poštujući određena pravila. Regije rastu dok ne zadovolje kriterije zaustavljanja.

Ovi postupci se mogu podijeliti na postupke temeljene na površinama i na one temeljene na dijelovima.

Primjer postupaka temeljenih na površinama je (Guillaume et al., 2004) koji klasificira vrhove poligona u  $k$  grozdova prema apsolutnim vrijednostima vlastitih vektora operatora ploha. Ovaj postupak pronalazi oštre rubove i pripadajuće oštre vrhove. Regije rastu dodavanjem susjednih trokutastih poligona koji ne dijele oštre rubove i svi njihovi vrhovi su ili oštri ili pripadaju istom grozdu. Rezultati ovog postupka su vidljivi na slici 2.9.

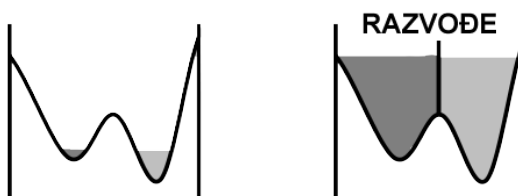


**Slika 2.9:** Rezultati iz (Guillaume et al., 2004)

Postupak iz (Zuckerberger et al., 2002) određuje segmente postepenim popunjavanjem dualnog grafa poligonalne mreže. Popunjavanje kreće iz čvorova koji se dobiju algoritmima za obilazak grafa u širinu (engl. *Breadth First Search*; BFS) ili u dubinu (engl. *Depth First Search*; DFS). Segmenti rastu dok njihova konveksnost nije narušena. Nakon narušavanja konveksnosti stvara se novi segment iz neobiđenog čvora te se obilazak dualnog grafa nastavlja. Ovaj postupak je sklon presegmentaciji.

### Postupci temeljeni na razvođu

Ova kategorija postupaka segmentira modele analogno punjenju udubina neravnih površina vodom. Kako voda puni udubine dolazi do stvaranja granica u dodirnim točkama regija oko udubina. Te granice čine razvođe i predstavljaju granice segmenata modela. Nastanak razvođa je ilustriran na slici 2.10.



**Slika 2.10:** Razvođe nastalo postupnim popunjavanjem udubina

Ovi postupci koriste istu transformaciju temeljenu na razvođu koja se koristi i za segmentaciju slika. Da bi se to omogućilo, potrebno je najprije transformirati 3D model nekom funkcijom  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  koja radi preslikavanje iz trodimenzionalnog pros-



tora u jednodimenzionalni prostor. Funkcija  $f$  očuva minimume udubina 3D modela.

Glavni nedostatak ovih postupaka je česta pojava presegmentacije. Ovaj problem je moguće ublažiti spajanjem razvodnih regija prema određenim kriterijima.

Većina postupaka koristi zakrivljenost kao funkciju razvodnjavanja. Nedostaci ovog pristupa su zupčaste granice segmenata te ograničenost na stvaranje segmenata samo iz regija veće zakrivljenosti.

### **Postupci temeljeni na modeliranju**

Postupci ove kategorije grade model površine mreže.

(Wu i Levine, 1997) simuliraju distribuciju električnih naboja na površini da bi segmentirali model. Točnije, grade segmentacijske granice na mjestima jako visoke i jako niske gustoće naboja. Prednost ovog postupka je manja osjetljivost na šum. Njegov nedostatak je pronalazak samo segmentacijskih regija koje su konkavno omeđene.

### **Postupci temeljeni na skeletnoj strukturi**

U ovu kategoriju spadaju postupci koji koriste skeletnu strukturu modela kao vodilju za segmentaciju.

U (Li et al., 2001) se najprije odredi skeletna struktura pojednostavljenjem površine modela pomoću postupka kontrakcije bridova opisanog u (Hoppe, 1996). Segmenti modela se određuju prolaskom ravnine kroz bridove skeletne strukture. Granice se određuju kritičnim točkama parametarske geometrijske i topološke funkcije koje se dobiju u presjecištima ravnine i poligonalne mreže. Rezultati ovog postupka su vidljivi na slici 2.11.

### **Postupci temeljeni na grupiranju u grozdove**

Segmentacija se ovim postupcima dobiva iterativnim grupiranjem elemenata poligonalne mreže u grozdove.

(Garland et al., 2001) su razvili hijerarhijski postupak grupiranja u grozdove za 3D modele. Postupak sažima bridove dualnog grafa poligonalne mreže. Bridovi se sažimaju ovisno o planarnosti (smjestivosti u ravninu), orijentaciji i obliku.

Postupak kojeg su razvili (Shlafman et al., 2002) grupira trokutaste poligone. Najprije izračuna udaljenosti između parova poligona. Zatim određuje reprezentativne poligone segmenata. Na kraju pripaja poligone njihovim najbližim reprezentativnim poligonima i pritom ažurira reprezentativne poligone minimizirajući zbroj udaljenosti



**Slika 2.11:** Rezultati dobiveni u (Li et al., 2001)

između poligona grupe i odgovarajućeg reprezentativnog poligona. Funkcija udaljenosti je pritom težinska suma geodetske udaljenosti centara masa i kutova između normala poligona.

(Katz i Tal, 2003) su razvili hijerarhijski segmentacijski postupak temeljen na neizrastivosti. Ovaj postupak je detaljno opisan u pododjeljku 2.4.2 i programski je implementiran. Programska implementacija je opisana u poglavlju 4. Neke rezultate koje daje ovaj postupak prikazuje slika 2.12.

Može se primjetiti da su postupci temeljeni na grupiranju u grozdove i oni temeljeni na rastućim regijama vrlo slični. Osnovna razlika je u tome što se polazne točke u postupcima temeljenima na grupiranju u grozdove ažuriraju tijekom segmentacije dok se kod postupaka temeljenima na rastućim regijama odrede u početku i do kraja rada postupka ostaju nepromijenjene.

### **Spektralna analiza**

Spektralna analiza za segmentaciju koristi svojstvene vektore pravilno definiranih matrica koje opisuju povezanost dualnog grafa poligonalne mreže.

(Liu i Zhang, 2004) definiraju matricu afiniteta  $\mathbf{W}$  s elementima  $0 < w_{ij} < 1$  koji određuju vjerojatnost pripadanja poligona  $i$  i  $j$  istom segmentu. Ovom matricom je moguće dobiti *smislene* segmente. U nastavku se određuje matrica  $\mathbf{V}$  čije stupce čine  $k$  najvećih svojstvenih vektora normalizirane matrice afiniteta  $\mathbf{W}_n$ . Nad redcima matrice  $\mathbf{V}$  se provodi  $k$ -struko grupiranje u grozdove. Svaki od grupiranih vektora predstavlja skupinu poligona. Time se poligonalna mreža segmentira.



Slika 2.12: Rezultati dobiveni u (Katz i Tal, 2003)

### Određivanje izričitih granica

Ovim postupcima se određuju konture granica segmentacije.

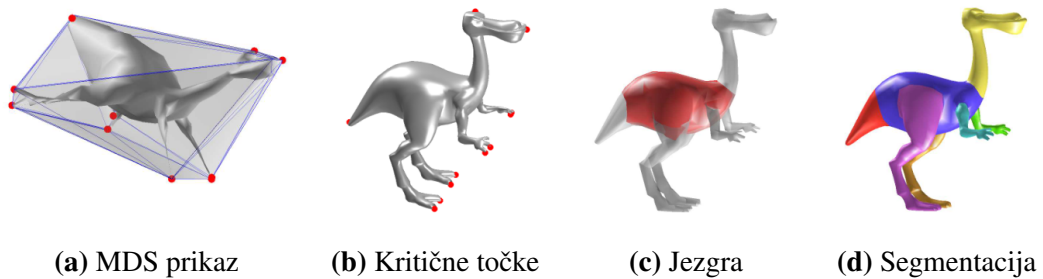
(Lee et al., 2004) i (Lee et al., 2005) traže konture na temelju minimuma zakrivljenja površine modela. Iz tih početnih kontura se odabire njihov podskup. Odabrane konture se zatvaraju čime se stvaraju granice segmenata. Zatvaranje kontura se dobiva traženjem najkraćeg puta analizom udaljenosti, normala, kutova i sl. svojstava poligona. Dobivene granice se na kraju dodatno pročišćuju. Krajnji rezultat su glatke granice između segmenata. Nedostatak ovih postupaka je dobivanje segmenata koji nisu uvijek *smisleni*.

### Postupci temeljeni na kritičnim točkama

Istaknuta svojstva 3D modela koja predstavljaju razna isturenja nazivaju se kritične točke. One služe postupcima ove kategorije kao smjernice za segmentaciju modela.

(Katz et al., 2005) kritične točke pronalaze u prikazu modela neovisnom o pozi. Do tog prikaza se dolazi višedimenzionalnom metodom skaliranja (engl. *multi-dimensional scaling method*; MDS). Točke se nalaze u lokalnim maksimumima funkcije koja opisuje isturenja. Novi prikaz poligonalne mreže se zajedno s kritičnim točkama projicira na okolnu sferu. Projekcijom se izdvoji središnji dio modela bez kritičnih točaka kojeg oni nazivaju jezgra. Dijelovi poligonalne mreže koji nisu u jezgri čine segmente

koji se dobiju obilaskom mreže iz smjera jezgre prema kritičnim točkama. Slika 2.13 prikazuje glavne faze ovog postupka.



**Slika 2.13:** Faze postupka iz (Katz et al., 2005)

### Postupci temeljeni na opisnicima višeskalnih oblika

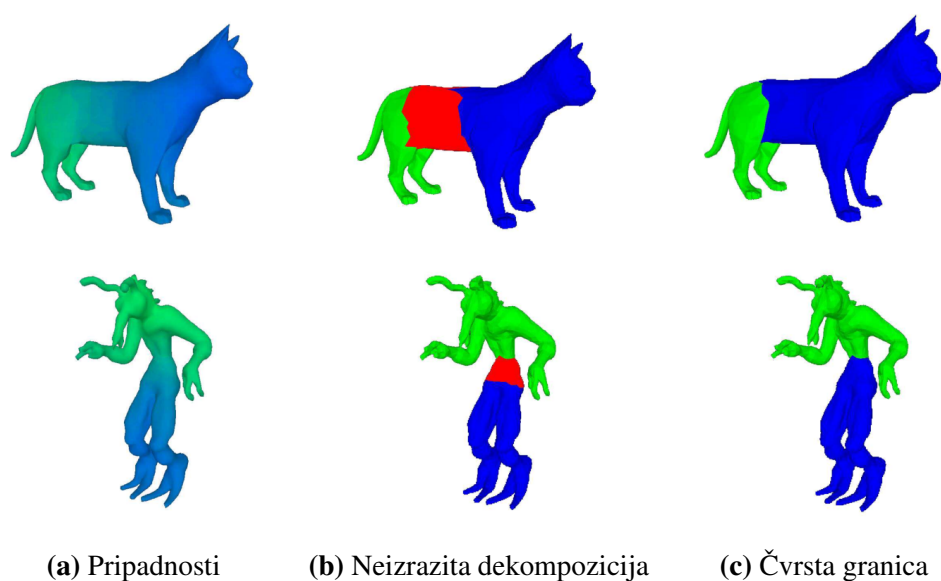
Ovi postupci koriste opisnike višeskalnih oblika za opis i određivanje komponenti poligonalne mreže.

(Mortara et al., 2003) za segmentaciju koriste opisnike svojstava oblika na različitim skalama. U svakoj točki koristi se skup polumjera  $R_i$  koji sijeku poligonalnu mrežu  $M$ . Broj sjecišta dobivenih iz neke točke na više skala kategorizira oblik skupine poligona i susjednih poligona. Jedno sjecište odgovara vrhovima i udubinama modela. Dva sjecišta označavaju udove, spojeve i sl. Mjesta s tri ili više sjecišta se nazivaju rascjepi. Nakon kategorizacije sjecišta poligonalna mreža se postepeno popunjava slično kao kod postupaka temeljenih na razvođu. Time se dobivaju segmentacijske regije.

## 2.4.2. Hijerarhijska neizrazita segmentacija poligonalnih mreža

Segmentacijski postupak opisan u (Katz i Tal, 2003) spada u postupke temeljene na grupiranje u grozdove.

Postupak kreće od grublje segmentacije koju razvija u sve finiju. Svaki čvor u stablu hijerarhije predstavlja mrežu poligona koja čini određeni segment, a korijen tog stabla predstavlja cijeli model. Postupak kod svakog čvora određuje prikladan broj  $k$  segmenata u koje segmentira dio modela zapisan u tom čvoru. Ideja postupka je naprije pronaći smislene cjeline uz neizrazite granice između njih. Nakon toga neizrazite regije se obrađuju i grade se čvrste granice. Faze postupka su ilustrirane na slici 2.14.



**Slika 2.14:** Faze postupka iz (Katz i Tal, 2003)

Prednosti postupka su:

- postupak je hijerarhijski
- segmentira model u *smislene* cjeline
- izbjegava presegmentaciju i zupčaste granice

Postupak se odvija u četiri faze:

1. Određivanje međusobnih udaljenosti poligona
2. Računanje pripadnosti pojedinog poligona nekoj zakrpi
3. Neizrazita dekompozicija
4. Određivanje čvrstih granica

Neka je  $S$  skup kojeg čini orijentabilna poligonalna mreža. Poželjno je da mreža bude i *manifold*. Ne mora biti trokutasta niti zatvorena.

Definiramo:

**$k$ -struka dekompozicija**  $S$  je segmentiran u  $S_1, S_2, \dots, S_k$  ako i samo ako vrijedi:

1.  $\forall i, 0 \leq i \leq k$  vrijedi  $S_i \subseteq S$
2.  $\forall S_i$  vrijedi da je povezan
3.  $\forall i \neq j, 0 \leq i, j \leq k$  vrijedi da su  $S_i$  i  $S_j$  disjunktni skupovi poligona
4.  $\cup_{i=1}^k S_i = S$

**Binarna dekompozicija**  $S_1$  i  $S_2$  su binarna dekompozicija od  $S$  ako su  $k$ -struka dekompozicija za  $k = 2$ .

**Zakrpa** Svaki  $S_i$  iz  $k$ -struke dekompozicije skupa  $S$  se zove zakrpa od  $S$ .

U nastavku su detaljno opisane faze postupka.

## 1. Određivanje udaljenosti

Najprije se izračunaju Euklidske udaljenosti između centara masa parova poligona. Vjerojatnost da poligon pripada nekoj zakrpi ovisi o njegovoj udaljenosti do ostalih poligona te zakrpe. Pretpostavlja se da se šansa da dva poligona pripadaju istoj zakrpi smanjuje kako im međusobna udaljenost raste. Isto tako, šanse da pripadaju istoj zakrpi im se povećavaju što su bliži jedan drugom.

Za dva susjedna poligona  $f_i$  i  $f_j$  definiramo kutnu udaljenost:

$$d_{kut}(\alpha_{ij}) = \eta(1 - \cos(\alpha_{ij})) \quad (2.1)$$

gdje je:

$\alpha_{ij}$  - kut između normala poligona  $f_i$  i  $f_j$

Konkavna svojstva površine modela su bolji kandidati za granice pa se za konkavne kutove uzima  $\eta = 1$ , a za konveksne kutove mali pozitivni  $\eta$ . Time konkavni kutovi imaju veći utjecaj na kutnu udaljenost između susjednih poligona.

Nakon računanja kutnih udaljenosti slijedi određivanje dualnog grafa poligonalne mreže. Definiramo težinu brida između dva vrha dualnog grafa:

$$tezina(dual(f_i), dual(f_j)) = \delta \cdot \frac{d_{geo}(f_i, f_j)}{avg(d_{geo})} + (1 - \delta) \cdot \frac{d_{kut}(\alpha_{ij})}{avg(d_{kut})} \quad (2.2)$$

Prvi izraz ovisi o Euklidskoj udaljenosti. Drugi izraz ovisi o kutnoj udaljenosti i iznosi 0 kad su poligoni koplanarni.  $\delta$  određuje utjecaj pojedinog izraza na konačnu težinu brida.

Za kraj prve faze je još potrebno pronaći duljine najkraćih puteva između parova vrhova dualnog grafa. Udaljenosti  $d$  između parova vrhova grafa se računaju samo jednom. Za to se koristi neki od algoritama koji su opisani u (Cormen et al., 2001). U ovom slučaju je odabran WFI algoritam.

**WFI algoritam** Warshall-Floyd-Ingermanov algoritam, složenosti  $O(V^3)$ , računa najmanju udaljenost dvaju vrhova u grafu.

Neka je  $D$  polje  $|V| \times |V|$  najmanjih udaljenosti parova vrhova.

Pseudokod algoritma glasi:

inicijalizacija: vrijednosti  $D$  na  $\infty$ ;

**za svaki vrh  $v$  čini**

    |  $D[v][v] = 0$ ;

**kraj**

**za svaki brid  $(u, v)$  čini**

    |  $D[u][v] = tezina(u, v)$ ;

**kraj**

**za  $k$  od 1 do  $|V|$  čini**

    | **za  $i$  od 1 do  $|V|$  čini**

        | **za  $j$  od 1 do  $|V|$  čini**

            | **ako  $D[i][j] > D[i][k] + D[k][j]$  onda**

                |  $D[i][j] = D[i][k] + D[k][j]$ ;

            | **kraj**

        | **kraj**

    | **kraj**

**kraj**

## 2. Računanje pripadnosti

Nakon računanja svih udaljenosti bira se  $k$  reprezentativnih poligona  $REP_i$  koji predstavljaju određenu zakrpu. Za slučaj  $k = 2$  biraju se  $REP_A$  i  $REP_B$  tako da udaljenost između njih bude najveća moguća.

Prvi problem koji se javlja za slučaj  $k > 2$  je određivanje broja zakrpa u koje će se čvor iz stabla hijerarhije segmentirati. Rješenje je iterativno biranje uz poznat kriterij zaustavljanja. Prvi reprezentativni poligon je onaj s najmanjom sumom udaljenosti do svih ostalih poligona. Ostali reprezentativni poligoni se dodaju iterativno tako da se maksimizira njihova najmanja udaljenost od svih do tada određenih reprezentativnih poligona. Tim postupkom dobivamo funkciju 2.3 najmanje udaljenosti  $k$ -tog reprezentativnog poligona od prethodno izabranih reprezentativnih poligona.

$$G(k) = \min_{i < k} (d(REP_k, REP_i)) \quad (2.3)$$

Funkcija se smanjuje dodavanjem novih reprezentativnih poligona. Empirijski se pokazalo da nakon određenog dodanog  $k$ -tog reprezentativnog poligona dođe do naglog pada funkcije 2.3. To je znak da nije potrebno dodavati više od  $k$  reprezentativnih poligona jer svaki novi dovodi do presegmentiranosti. Iz toga slijedi da je potrebno odabrati  $k$  koji maksimizira prvu derivaciju funkcije 2.3.

Nakon određivanja broja reprezentativnih poligona slijedi dodjeljivanje vjerojatnosti pripadanja pojedinog poligona određenoj zakrpi.

Za slučaj  $k = 2$  cilj je svakom od poligona pridodati vjerojatnost  $P_B$  pripadanju zakrpi  $B$ .

Neka su:

$$a_{f_i} = d(f_i, REP_A) \quad (2.4)$$

$$b_{f_i} = d(f_i, REP_B) \quad (2.5)$$

Definiramo funkciju pripadnosti zakrpi  $REP_B$ :

$$P_B(f_i) = \frac{a_{f_i}}{a_{f_i} + b_{f_i}} = \frac{d(f_i, REP_A)}{d(f_i, REP_A) + d(f_i, REP_B)} \quad (2.6)$$



Vrijede svojstva:

1.  $P_B(f_i) = 1 - P_A(f_i)$
2.  $P_B(REP_B) = 1$
3.  $P_B(REP_A) = 0$
4.  $0 < P_B(f) < 1$

Za slučaj  $k > 2$  vrijedi općenita funkcija 2.7.

$$P_{p_j}(f_i) = \frac{\frac{1}{d(f_i, REP(p_j))}}{\sum_l \frac{1}{d(f_i, REP(p_l))}} \quad (2.7)$$

Lako se provjeri da je ova funkcija proširenje binarnog slučaja. Funkcija ima sljedeća svojstva:

1.  $0 \leq P(f_i, p_j) \leq 1$
2. Suma svih vjerojatnosti iznosi 1
3. Vjerojatnost pripadnosti zakrpi se smanjuje povećanjem udaljenosti od reprezentativnog poligona zakrpe

### 3. Neizrazita dekompozicija

Do neizrazite dekompozicije se dolazi neizrazitim grupiranjem poligona u grozdove koji u ovom slučaju čine zakrpe. To se postiže minimiziranjem funkcije 2.8, gdje je  $p$  neki reprezentativni poligon.

$$F = \sum_p \sum_f vjerojatnost(f \in zakrpa(p)) \cdot d(f, p) \quad (2.8)$$

Bira se početni skup  $V_k$  od  $k$  reprezentativnih poligona, nakon čega slijedi iterativni algoritam:

1. Računanje vjerojatnosti pripadnosti pojedinoj zakrpi
2. Biranje novog skupa  $V'_k$  tako da se minimizira funkcija  $F$
3. Ako se  $V_k$  i  $V'_k$  razlikuju, zamijeni  $V_k$  s  $V'_k$  i vrati se na korak 1.

Drugi korak se provodi računanjem novih reprezentativnih poligona preko 2.9 i 2.10.

$$REP_A = \min_f \sum_{f_i} (1 - P_B(f_i)) \cdot d(f, f_i) \quad (2.9)$$

$$REP_B = \min_f \sum_{f_i} P_B(f_i) \cdot d(f, f_i) \quad (2.10)$$

Nakon prethodnih koraka slijedi dekompozicija poligona u zakrpe.

Ako vjerojatnost pripadanja poligona nekoj zakrpi premaši određenu granicu, pridaje se toj zakrpi. Od ostalih poligona nastaje neizraziti skup poligona koji mogu pripadati većem broju zakrpi.

U binarnom slučaju imamo tri zakrpe:  $A$ ,  $B$  i  $C$ .  $C$  je zakrpa koja predstavlja neizrazitu regiju poligona.

Vrijede sljedeće definicije skupova  $A$ ,  $B$  i  $C$ :

$$A = \{f_i | P_B(f_i) < 0.5 - \epsilon\} \quad (2.11)$$

$$B = \{f_i | P_B(f_i) > 0.5 + \epsilon\} \quad (2.12)$$

$$C = \{f_i | 0.5 - \epsilon \leq P_B(f_i) \leq 0.5 + \epsilon\} \quad (2.13)$$

Vidljivo je da  $\epsilon$  određuje veličinu neizrazite regije.

Za slučaj  $k > 2$  izračun neizrazite dekompozicije se također provodi revizijom neizrazitih pripadnosti pomoću iterativnog postupka grupiranja u grozdove. Biraju se susjedni parovi zakrpa i algoritam nastavlja kao u binarnom slučaju.

#### 4. Izračun čvrstih granica

Neizrazitom dekompozicijom su nađene *smislene* cjeline ali ne i čvrste granice između njih.

U binarnom slučaju treba naći tu čvrstu granicu u regiji  $C$ . Poligone iz zakrpe  $C$  treba svrstati u zakrpe  $A$  i  $B$ . To se svodi na problem particioniranja grafa.

Najprije se odrede dualni graf poligonalne mreže  $G = (V, E)$  i skupovi vrhova koji pripadaju određenoj zakrpi ( $V_A$  i  $V_B$ ).

Potrebno je razdvojiti skup  $V$  u  $V_{A'}$  i  $V_{B'}$  tako da razdruživanje između njih bude minimalno.

Točnije, traži se presjek grafa  $G$  tako da vrijedi:

1.  $V = V_{A'} \cup V_{B'}$
2.  $V_{A'} \cap V_{B'} = \emptyset$
3.  $V_A \subseteq V_{A'}$ ,  $V_B \subseteq V_{B'}$
4.  $tezina(\text{presjek}(V_{A'}, V_{B'})) = \sum_{u \in V_{A'}, v \in V_{B'}} \omega(u, v)$  je minimalna

Da bi se to postiglo, definira se graf  $G' = (V', E')$  za kojeg vrijedi:

$$V' = V_C \cup V_{CA} \cup V_{CB} \cup \{S, T\}$$

$$E' = E_C \cup \{(S, v), \forall v \in V_{CA}\} \cup \{(T, v), \forall v \in V_{CB}\} \cup \{e_{ij} \in E \mid i \in V_C, j \in \{V_{CA} \cup V_{CB}\}\}$$

gdje su:

$V_C$  - skup vrhova dualnoga grafa čiji odgovarajući poligoni pripadaju zakrpi  $C$

$V_{CA}$  - skup vrhova dualnoga grafa čiji odgovarajući poligoni iz zakrpe  $A$  dijele bridove s poligonima zakrpe  $C$

$V_{CB}$  - skup vrhova dualnoga grafa čiji odgovarajući poligoni iz zakrpe  $B$  dijele bridove s poligonima zakrpe  $C$

$S$  - vrh grafa povezan s  $V_{CA}$

$T$  - vrh grafa povezan s  $V_{CB}$

$E_C$  - skup bridova dualnog grafa zakrpe  $C$

Slika 2.15 prikazuje zakrpe binarne dekompozicije i graf  $G'$ . Na temelju grafa  $G'$  gradimo neusmjerenu mrežu u kojoj kapacitet brida između vrhova  $i$  i  $j$  iznosi:

$$\text{kapacitet}(i, j) = \begin{cases} \frac{1}{1 + \frac{d\_kut(\alpha_{ij})}{\text{avg}(d\_kut)}} & , \text{ ako } \{i, j \neq S, T\} \\ \infty & , \text{ inače} \end{cases} \quad (2.14)$$

Na novoj mreži se koristi postupak za traženje najvećeg protoka od  $S$  do  $T$  čime se dobiva minimalni presjek mreže. Za to je prikladan Ford-Fulkersonov postupak opisan u (Cormen et al., 2001). Dobiveni presjek je konačna granica koja segmentira poligone zakrpe  $C$  u  $A$  i  $B$ .

**Ford-Fulkersonov postupak** Ovaj postupak rješava problem protoka u mrežama i izvediv je raznim algoritmima različitih složenosti.

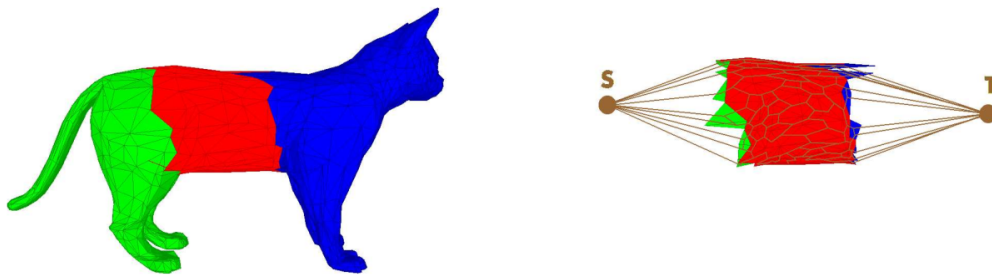
Ideja Ford-Fulkersonovog postupka glasi:

inicijalizacija: tokovi svih bridova  $f(u, v) = 0$ ;

**dok** u mreži postoji dopunski put  $p$  **čini**

    | uvećati sve tokove na putu  $p$  za  $f_p$ ;

**kraj**



**Slika 2.15:** Zakrpe binarne dekompozicije i pripadni graf  $G'$

### Uvjeti zaustavljanja postupka

Svaki čvor iz stabla hijerarhije se rekurzivno segmentira tako dugo dok se ne ispuni neki od sljedećih uvjeta:

- udaljenost između reprezentativnih poligona je manja od neke granice
- razlika između najvećeg dihedalnog kuta i najmanjeg dihedalnog kuta je manja od neke granice - time se izbjegava segmentiranje zakrpi koje imaju relativno konstantnu zakrivljenost
- omjer između prosječne udaljenosti unutar zakrpe i one unutar cijelog modela je manji od neke granice - pošto prosječna udaljenost uzima u obzir veličinu segmenta i informaciju o kutevima (zakrivljenost), dodatna segmentacija je nepotrebna kad su obje veličine male u odnosu na početni model

### 2.4.3. Usporedba postupaka

Ne postoji savršeni segmentacijski postupak. Svaki postupak ima svoje prednosti i nedostatke. Uspješnost nekog segmentacijskog postupka ovisi o njegovoj primjeni i dobri rezultati se dobivaju samo u slučaju kad su zadovoljeni svi kriteriji koje postupak zahtijeva.

U (Attene et al., 2006a) su predloženi kriteriji po kojima se ocjenjuje pojedini segmentacijski postupak.

#### Vrsta segmentacije

Segmentacijski postupci se mogu kategorizirati u one koji grade *smislene* segmente i one koji segmente temelje na jednostavnim geometrijskim oblicima.

Postupci koji grade *smislene* segmente su prikladniji za primjene koje koriste *prirodne* podatke. Primjer toga bi bilo određivanje skeletne strukture na temelju modela neke životinje.

Druga spomenuta vrsta postupaka je prikladnija za CAD primjene u industriji.

#### Određivanje *ispravnih* segmenata

*Ispravnost* segmenata je nemoguće procijeniti nekim algoritmom. Nije jednostavno formalno definirati *smislene* cjeline kompleksnih modela jer nema kvantitativne usporedbe. Ostaje samo kvalitativna usporedba gledanjem slika konačnih segmentacija.

Za odabir prikladnog postupka potrebno je znati kontekst primjene i prirodu segmentacijskog postupka.

#### Određivanje granica

Za pronađene granice je također nemoguće procijeniti *ispravnost*. Moguće je doduše analizirati duljinu, glatkoću i poziciju u odnosu na druga svojstva površine modela.

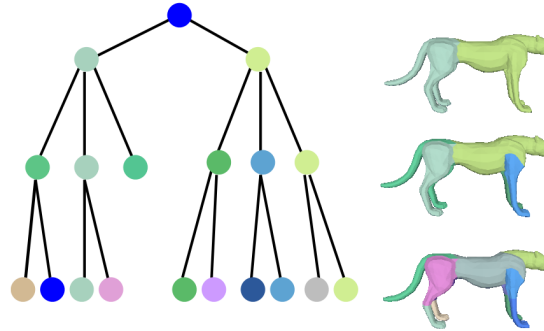
Za segmentacijske postupke koji stvaraju zupčaste granice postoji poboljšanje opisano u (Kaplansky i Tal, 2009). Valja napomenuti i to da pri nekim primjenama nisu potrebne gratke granice.

#### Hijerarhijska i višeskalna segmentacija

Rijetko kad je jednokratna segmentacija dovoljna za *dovoljno dobru* segmentaciju cijelog objekta. Zato su razvijeni hijerarhijski i višeskalni postupci.

Hijerarhijska segmentacija je prikladna kad postoji potreba za konzistentnom segmentacijom na više razina. Ovo se najviše odnosi na modele koji imaju zglobove. Primjer stabla hijerarhije ilustrira slika 2.16.

Višeskalna segmentacija je korisna za segmentaciju detalja na nekom većem i složenijem modelu.



Slika 2.16: Stablo hijerarhije generirano postupkom iz (Katz i Tal, 2003)

### Osjetljivost na pozu

Za neke primjene je potrebno da se isti objekt u različitim pozama segmentira podjednako.

Ne postoji postupak koji je otporan na promjenu poze, ali ima onih koji nisu jako osjetljivi na to. Primjer toga je (Katz et al., 2005) koji pretvara model u prikaz neovisan o pozici. On može služiti i kao korak predprocesiranja modela za druge postupke.

### Asimptotska složenost

Asimptotska složenost utječe na vrijeme izvođenja postupka. Za većinu postupaka ju nije lako odrediti jer ovisi o odabiru algoritama za određene korake poput onih za traženje najkraćih puteva u grafu, za traženje najvećeg protoka mreže i sl. Asimptotska složenost donekle ovisi i o odabiru zapisa poligonalne mreže te o platformi na kojoj se izvodi.

### Kontrolni parametri

Broj kontrolnih parametara govori o količini interakcije korisnika potrebne da bi se proizvela kvalitetna segmentacija. Njihov broj se kreće od samo jednog do desetak parametara. Više parametara daje veću kontrolu nad procesom segmentacije. Kod nekih primjena je poželjno da se segmentacija obavlja samostalno bez intervencije korisnika.

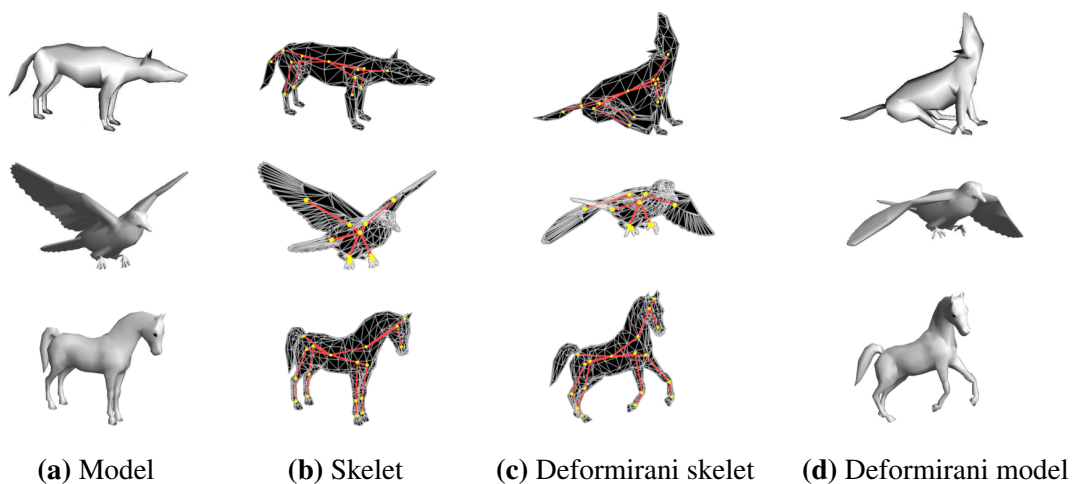
## 3. Primjena segmentacije objekata zadanih poligonalnom mrežom

Segmentacija modela sama po sebi nije pretjerano korisna. Njezina važnost dolazi do izražaja tek kada se koristi u raznim kontekstima kao pomoć ili predprocesni korak. Postupci segmentacije objekata se mogu primjenjivati za određivanje skeletne strukture, metamorfozu modela, dohvat sličnih oblika, pojednostavljenje poligonalne mreže, kompresiju poligonalne mreže, detekciju sudara, uklanjanje šuma i popravak modela, mapiranje tekstura, reverzni inženjering modela i u razne druge svrhe.

U nastavku su opisane neke od primjena.

### 3.1. Određivanje skeletne strukture

**Skeletna struktura** ili **kontrolni skelet** je skup zglobova koji su hijerarhijski povezani kostima. Kontrolni skelet se koristi za animaciju modela tako da se najprije poveže s poligonalnom mrežom modela.



Slika 3.1: Deformiranje modela kontrolnim skeletom

U (Katz i Tal, 2003) se za određivanje kontrolnog skeleta koriste granice između segmentiranih regija. Kontrolni skelet može se izravno dobiti iz hijerarhijske strukture nastale segmentacijom. Nužno je da se dijelovi modela u hijerarhiju dodaju kao potomci dijelova o kojima ovise. Npr. ljudski lakat mora biti potomak ramena jer se miče zajedno s ramenom. Jednostavan način za postizanje ovoga je da se svaki čvor u hijerarhiji segmentira u neku centralnu zakrpu povezanu sa svim ostalim zakrpama nastalima iz tog čvora. Cilj tog postupka je dobiti zvjezdoliku strukturu. Za ovo je potrebna mala modifikacija segmentacijskog postupka. Prvi kandidat za centralnu zakrpu je zakrpa nastala iz prvog reprezentativnog poligona. Nakon njegove segmentacije provjerava se je li nastala zvjezdolika struktura. Ako nije, zakrpa koja nije susjedna centralnoj zakrpi se spaja sa susjednom zakrpom s kojom granični poligoni zakrpa u prosjeku zatvaraju najmanji kut. Nakon  $k$ -struke segmentacije, iz stabla zakrpa se stvara stablo zglobova. Na svakoj razini hijerarhije se stvaraju zglobovi između centralne zakrpe i njezinih susjednih zakrpa. Svaki spoj se pozicionira u centar mase granice između zakrpa i poveže se s mrežom poligona. Korijen stabla zglobova je povezan s cijelim modelom. Segmentacijom modela iz grubljeg u finije novonastale zakrpe se povezuju s djecom čvorovima stabla zglobova.

Kontrolni skelet se povezuje s modelom na način da se svakom vrhu  $v_i$  poligonalne mreže pridoda težina  $w_{ij}$  kojom pripada određenom zglobu  $j$ .  $w_{ij}$  se računa kao postotak susjednih poligona vrha  $v_i$  koji pripadaju zglobu  $j$ . Time se dobiva sljedeće:

- pojedini vrh je povezan samo sa zakrpom kojoj pripada
- svaki vrh pripada barem jednoj zakrpi
- $\sum_j w_{ij} = 1$

Modeli se deformiraju mijenjanjem kutova između zglobova i kostiju. Deformirane modele ilustrira slika 3.1. Novi položaj vrha poligona računa se prema formuli:

$$y_i = \sum_{j=0}^{J-1} (w_{ij} x_{ij} M_j) \quad (3.1)$$

gdje je:

$J$  - broj zglobova

$x_{ij}$  - vektor relativnog položaja vrha  $v_i$  prema koordinatnom sustavu zgloba  $j$

$M_j$  - transformacijska matrica zgloba  $j$

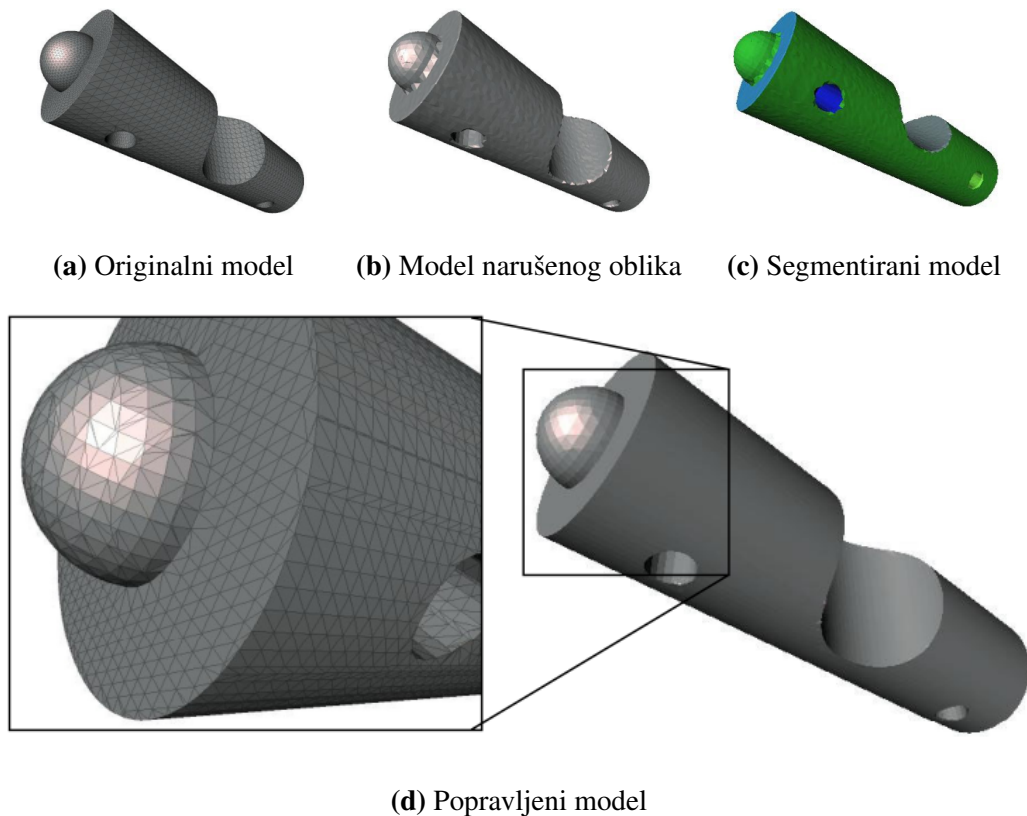
Relativni položaj vrha prema zglobu je konstantan. Položaj vrha koji pripada k više zglobova se kreće između krajnosti u kojima bi bio da pripada isključivo svakom od tih zglobova.



## 3.2. Uklanjanje šuma i popravak modela

Segmentacijski postupci koji grade segmente geometrijskim primitivima mogu poslužiti za poboljšanje kvalitete CAD modela koji se dobivaju laserskim skeniranjem stvarnih objekata. Poligonalne mreže tih modela najčešće sadrže šum kojeg je poželjno ispraviti. Ako se skenirani model sastoji od dobro definiranih geometrijskih primitiva, sve vrhove poligonalne mreže koja ga opisuje je moguće poravnati tako da tvore pravilne oblike. Grupiranje u grozdove pronalazi ispravne rubove modela. Time se uklanja šum i kvaliteta poligonalne mreže se povećava.

Postupak opisan u (Attene et al., 2006b) je prikladan za ovu primjenu. Njihovi rezultati su vidljivi na slici 3.2.



**Slika 3.2:** Uklanjanje šuma i popravak modela pomoću segmentacije

### 3.3. Pojednostavljenje poligonalne mreže i kompresija

Pojednostavljenjem poligonalne mreže modela dobiva se ubrzanje iscrtavanja tog modela.

Jedan od postupaka pojednostavljenja modela je decimacija. Decimacijom se iterativno uklanjaju vrhovi poligonalne mreže i iz novonastalih rupa se stvaraju novi, jednostavniji poligoni. Na ovaj način se i smanjuje broj poligona mreže. Postupak se ponavlja dok se ne dobije željeni broj vrhova i poligona mreže. Decimacija je brza i ovisna o topologiji poligonalne mreže. Problemi na koje nailazi ovaj postupak su način odabira vrhova za brisanje i način stvaranja novih poligona. Već samo nakon nekoliko iteracija decimacije može doći do gubitka značajki oblika modela.

Segmentacija u ovom slučaju pomaže na način da izolira decimaciju u pojedine segmente modela. Time značajke oblika modela ostaju uglavnom očuvane.

U (Zuckerberger et al., 2002) je postignuto pojednostavljenje modela od 39697 poligona na samo 7936 poligona, pri čemu se segmentacijom dobilo 104 segmenata a značajke oblika modela su ostale očuvane. Slika 3.3 prikazuje dobiveni rezultat.



(a) 39697 poligona (100%)

(b) 7936 poligona (20 %)

**Slika 3.3:** Pojednostavljenje modela iz (Zuckerberger et al., 2002)

## 4. Programska implementacija

Računarstvo nije samo sebi svrha. Postupci opisani u prethodnim poglavljima postaju korisni tek kad ih se implementira na računalu.

Postupak opisan u 2.4.2 programski je implementiran i proširen za određivanje skeletne strukture kako je opisano u 3.1. Rezultati su prikazani na različitim primjerima. Napravljen je osvrt na utjecaj parametara algoritama postupka, vrijeme izvođenja i memorijske zahtjeve. Na kraju su predložena moguća poboljšanja ostvarene implementacije.

### 4.1. Programski alati

Za ostvarenje programske implementacije segmentacijskih postupaka potrebno je moći baratati poligonalnim mrežama, kontrolirati tijek postupka različitim parametrima i iscrutati dobivene rezultate.

Programski proizvod je razvijen pomoću programskog jezika C++ i odgovarajućih biblioteka.

Za učitavanje i obradu poligonalnih mreža korištena je biblioteka OpenMesh.

Grafičko korisničko sučelje i iscrtavanje poligonalnih mreža ostvareno je pomoću razvojne okoline Qt i grafičkog programskog sučelja OpenGL.

3D modeli su preuzeti iz AIM@SHAPE javnog repozitorija digitalnih oblika.

#### OpenMesh

OpenMesh je biblioteka razvijena u svrhu implementacije istoimene strukture podataka opisane u (Botsch et al., 2002).

Biblioteka nudi strukture podataka i alate za prikaz i manipulaciju poligonalnih mreža.

Od poznatih datotečnih formata, učitavanje i spremanje poligonalnih mreža je podržano za `.obj`, `.off` i `.stl` datoteke.

Kretanje po poligonalnoj mreži je izuzetno jednostavno i vremeski efikasno. To omogućuje *half-edge* struktura podataka opisana u 2.2.3.

Nad učitanim mrežom se najprije definiraju *iteratori*. To su pokazivači na odgovarajuće elemente poligonalne mreže čiji operatori omogućuju kretanju po istovjetnim elementima. Uz *iteratore* postoje i *cirkulatori*. Oni su također pokazivači na odgovarajuće elemente ali njihova kretanja je ograničena po susjednim elementima elementa na kojeg pokazuje *iterator*. Njihova kombinacija omogućuje da se u samo dvije napisane programske petlje prođe po svim elementima poligonalne mreže i svim njihovim susjednim elementima.

U nastavku je naveden dio programskog koda koji definira *iterator* po poligonima i odgovarajući *cirkulator* po svim vrhovima poligona te ih koristi za izračun centra mase pojedinog poligona.

```
typedef OpenMesh::PolyMesh_ArrayKernelT<> MyMesh;
MyMesh my_mesh;
MyMesh::Point cog;
MyMesh::Scalar valence;
std::vector<MyMesh::Point> cogs;
...
MyMesh::FaceIter f_it, f_end(my_mesh.faces_end());
MyMesh::FaceVertexIter fv_it;
for (f_it = my_mesh.faces_begin(); f_it != f_end; ++f_it)
{
    cog[0] = cog[1] = cog[2] = valence = 0.0;
    fv_it = my_mesh.fv_iter(*f_it);
    for (fv_it; fv_it.is_valid(); ++fv_it)
    {
        cog += my_mesh.point(*fv_it);
        ++valence;
    }
    cogs.push_back(cog / valence);
}
```

**Slika 4.1:** Primjer kretanja po OpenMesh strukturi podataka

Struktura podataka koja predstavlja poligonalnu mrežu je generična. To znači da je proširiva na razne načine da bi se koristila u specifične svrhe. Svakome elementu poligonalne mreže je moguće dodijeliti temeljne tipove podataka.

U nastavku je naveden dio programskog koda koji svakom poligonu učitane poligonalne mreže pridodaje mogućnost pamćenja cijelog broja i kretanjem po mreži svakome pridodaje jedinstveni broj.

```
OpenMesh::FPropHandleT<int> fprop_id;
my_mesh.add_property(fprop_id, "id");
my_mesh.property(fprop_id).set_persistent(true);
...
MyMesh::FaceIter f_it, f_end(my_mesh.faces_end());
int i = 0;
for (f_it = my_mesh.faces_begin(); f_it != f_end; ++f_it)
{
    my_mesh.property(fprop_id, *f_it) = i;
    i++;
}
```

**Slika 4.2:** Primjer dodavanja svojstava OpenMesh strukturi podataka

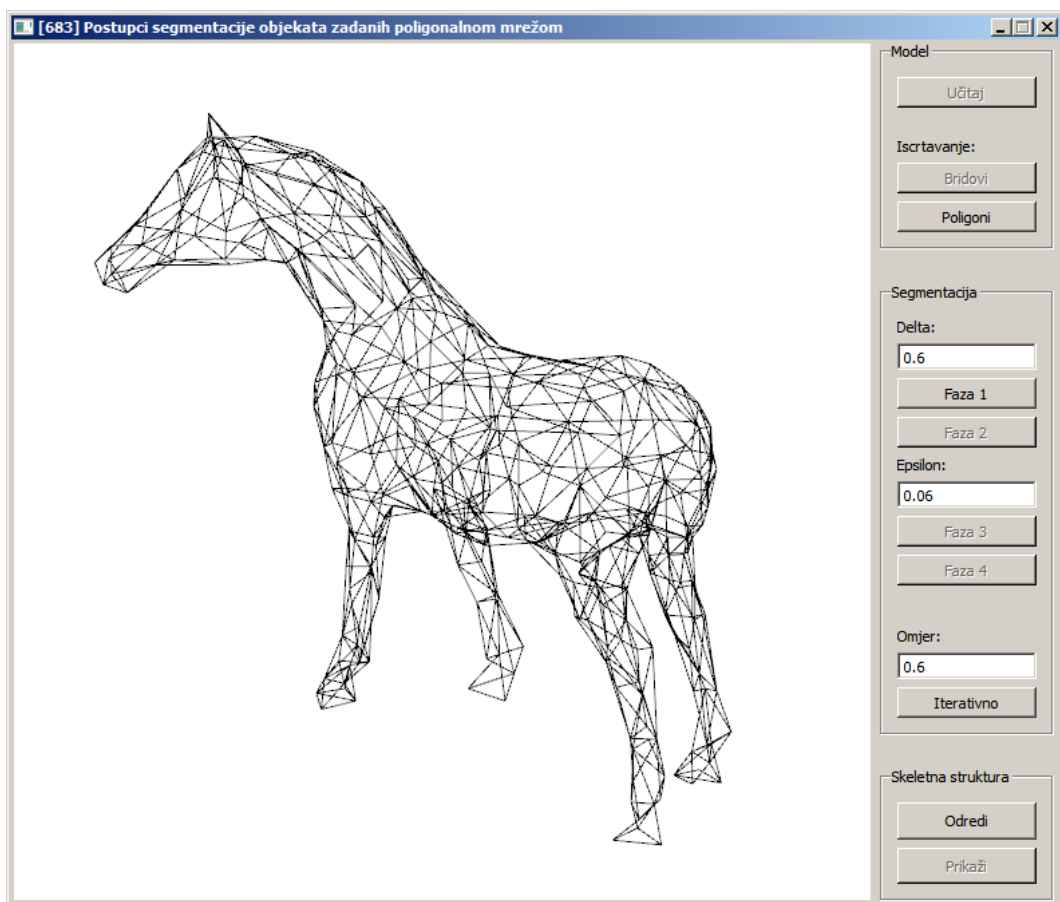
## Grafičko korisničko sučelje

Slika 4.3 prikazuje grafičko korisničko sučelje programa.

Korisničko sučelje nudi sljedeće mogućnosti:

- **Učitaj** - učitava poligonalnu mrežu iz podržane datoteke
- **Bridovi** - iscrtava bridove modela
- **Poligoni** - iscrtava poligone modela
- **Delta** - vidi formulu 2.2
- **Faza 1** - pokreće prvu fazu binarnog segmentacijskog postupka
- **Faza 2** - pokreće drugu fazu binarnog segmentacijskog postupka
- **Epsilon** - vidi formule 2.11, 2.12 i 2.13

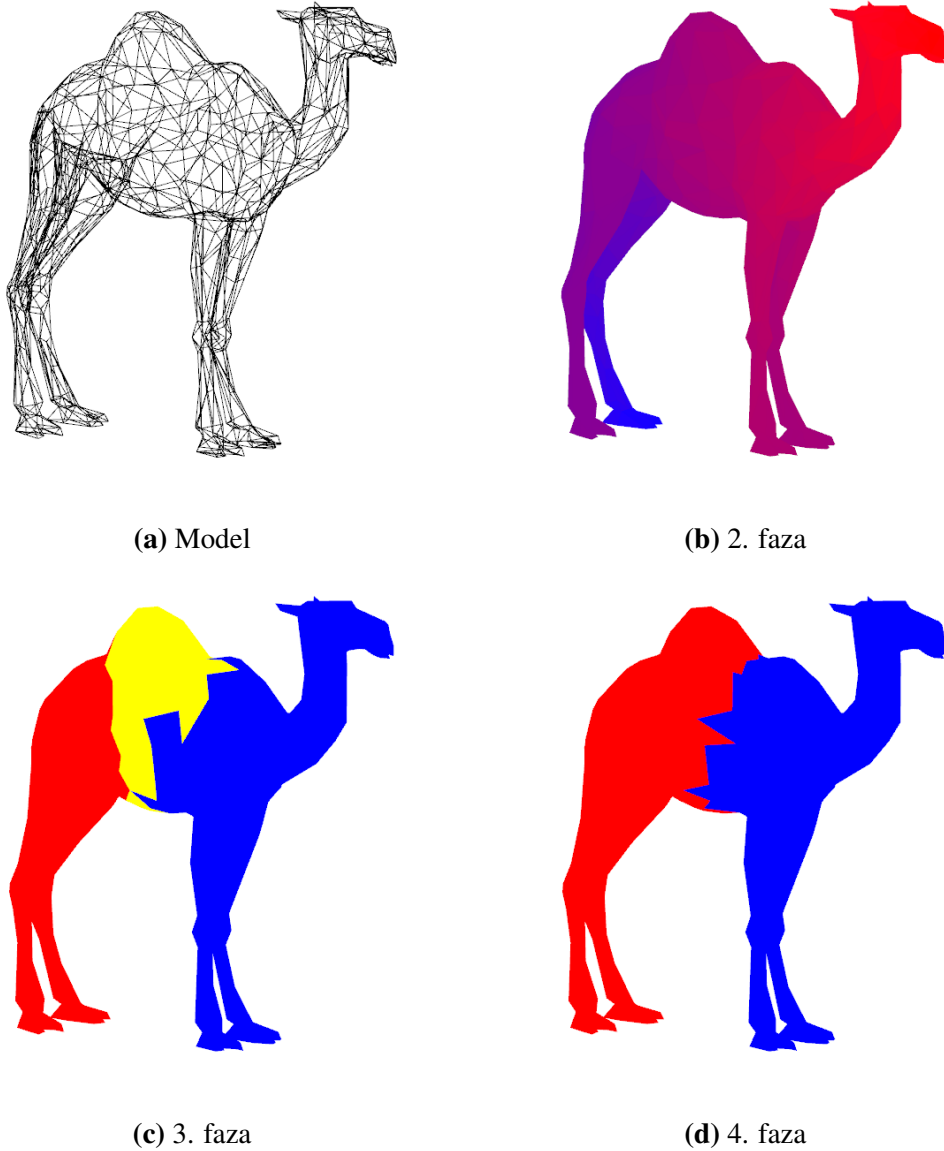
- **Faza 3** - pokreće treću fazu binarnog segmentacijskog postupka
- **Faza 4** - pokreće četvrtu fazu binarnog segmentacijskog postupka
- **Omjer** - uvjet zaustavljanja iterativnog postupka (omjer između prosječne udaljenosti unutar zakrpe i one unutar cijelog modela)
- **Iterativno** - pokreće iterativni binarni segmentacijski postupak
- **Odredi** - određuje skeletnu strukturu modela
- **Prikaži** - iscrtava skeletnu strukturu modela



**Slika 4.3:** Grafičko korisničko sučelje

## 4.2. Rezultati

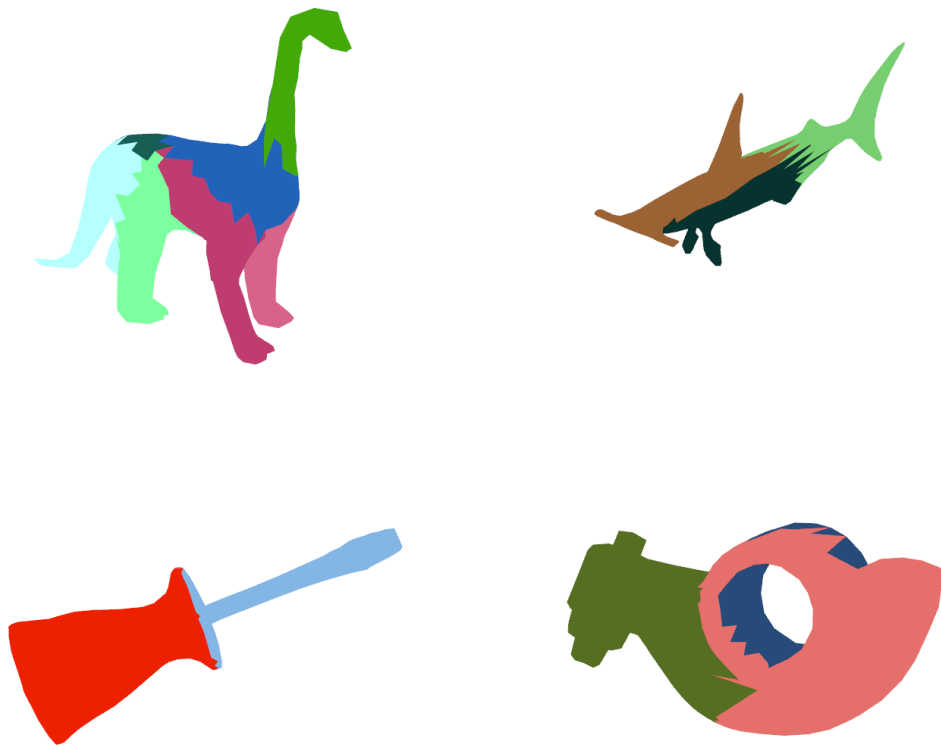
Programski je implementirana binarna inačica segmentacijskog postupka iz 2.4.2. Rezultati pojedinih faza postupka vidljivi su na slici 4.4.



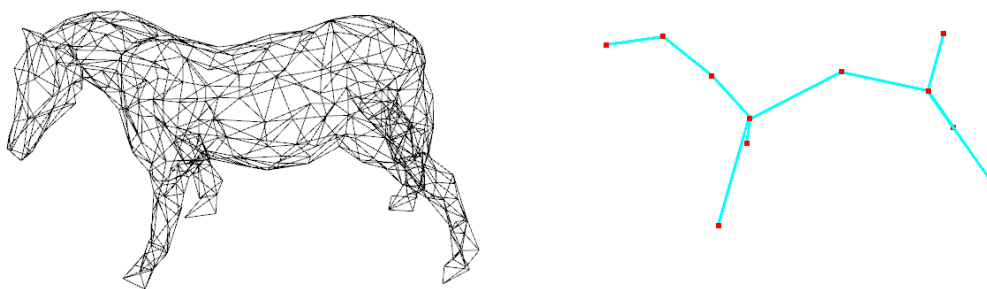
**Slika 4.4:** Faze implementiranog segmentacijskog postupka

Primjeri rezultata iterativnog postupka vidljivi su na slici 4.5.

Skeletna struktura je na kraju povezana s modelom kojeg je moguće deformirati pomicanjem zglobova. Primjer dobivene skeletne strukture prikazuje slika 4.6.



**Slika 4.5:** Rezultati iterativnog postupka



**(a)** Model

**(b)** Kontrolni skelet

**Slika 4.6:** Model i pripadni kontrolni skelet



### 4.3. Analiza

Implementacija postupka uspješno pronalazi smislene cjeline na modelima ako se dobro podese parametri, posebice uvjet zaustavljanja iterativnog postupka.

Zupčaste granice su posljedica malog broja poligona od kojih su sastavljeni korišteni modeli.

**Sklopovlje računala** Na vrijeme i mogućnost izvođenja implementiranog postupka utječu isključivo procesor i radna memorija računala. U nastavku su navedeni CPU i RAM koji su korišteni prilikom programske implementacije.

CPU: Intel Core 2 Quad Q6600 @ 2.40GHz

RAM: DDR2 - 8192 MBytes

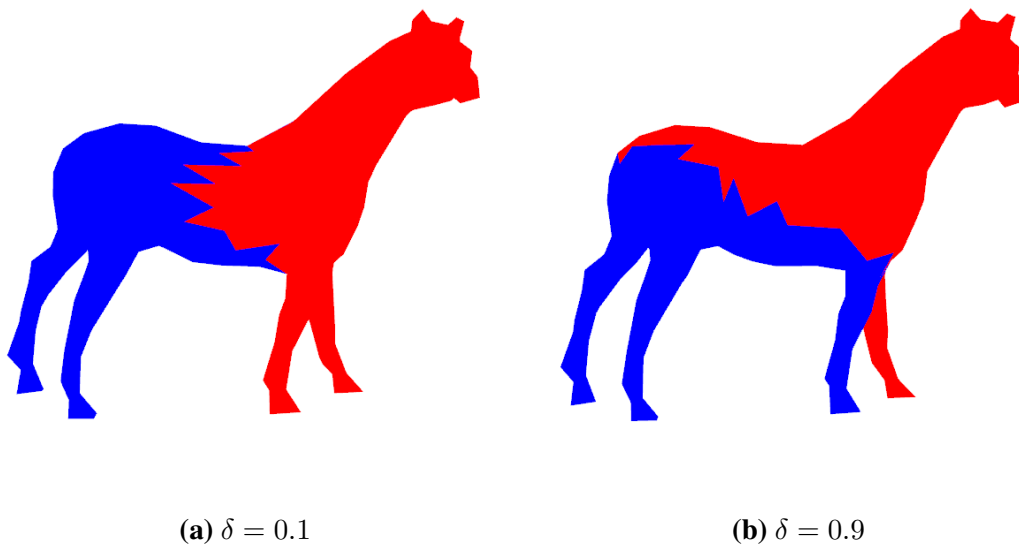
#### Utjecaj parametara

Preko grafičkog korisničkog sučelja moguće je podesiti tri parametra postupka.  $\delta$  iz formule 2.2,  $\epsilon$  iz formula 2.11, 2.12 i 2.13 i omjer između prosječne udaljenosti unutar zakrpe i one unutar cijelog modela, koji predstavlja uvjet zaustavljanja iterativnog postupka.

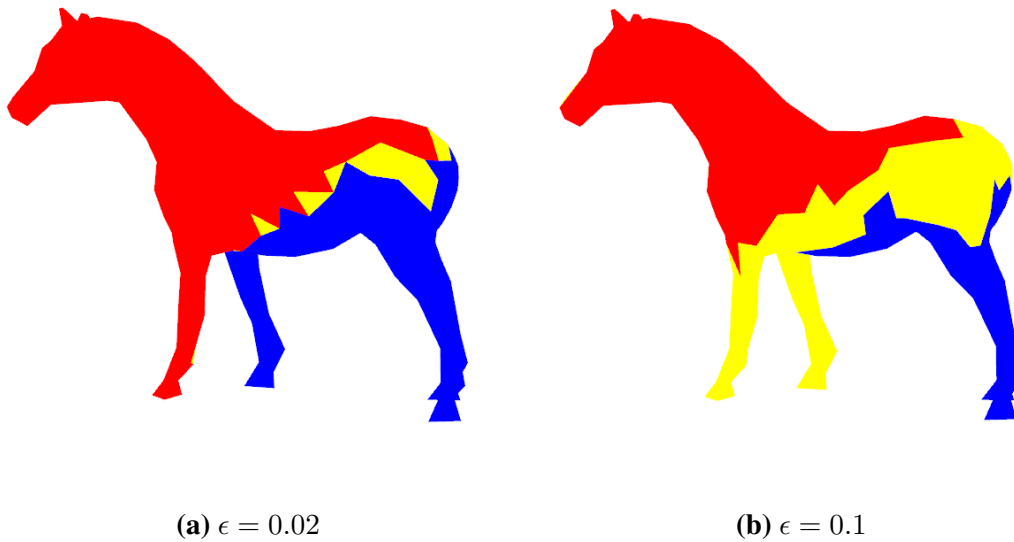
Slika 4.7 sadrži usporedbu rezultata različitih  $\delta$ .

Slika 4.8 sadrži usporedbu rezultata različitih  $\epsilon$ .

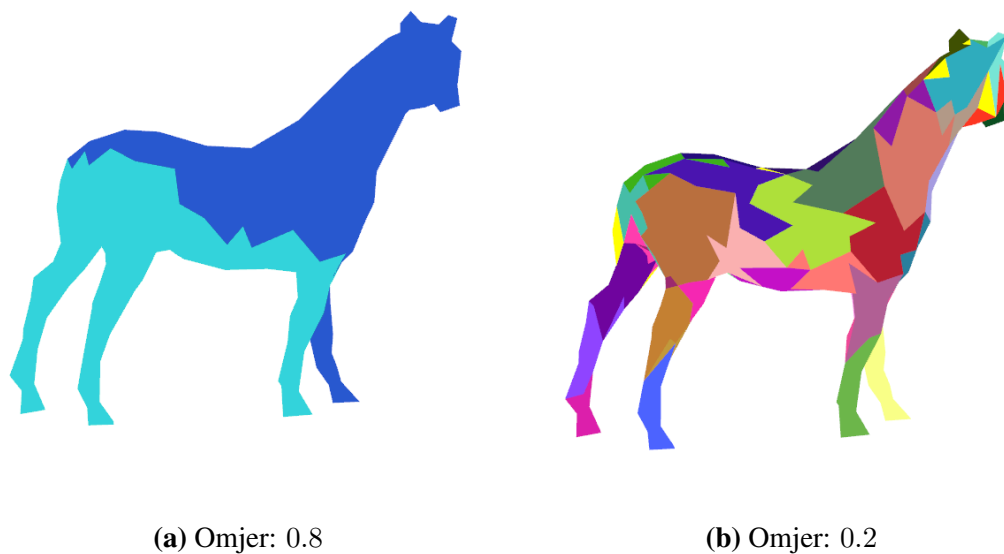
Slika 4.9 sadrži usporedbu rezultata različitih uvjeta zaustavljanja.



**Slika 4.7:** Utjecaj parametra  $\delta$



**Slika 4.8:** Utjecaj parametra  $\epsilon$



**Slika 4.9:** Utjecaj različitih uvjeta zaustavljanja

### Vrijeme izvođenja

Ukupno trajanje segmentacijskog postupka mjerljivo je u sekundama, što je indikator da su segmentacijski postupci vremenski jako složeni.

Prva faza traje uvjerljivo najduže je nekoliko puta računa udaljenosti između svih parova poligona. Tome pridonosi i WFI algoritam složenosti  $O(V^3)$ . Druga i treća faza traju najkraće. Trajanje četvrte faze ovisi o odabiru parametra  $\epsilon$  iz formula 2.11, 2.12 i 2.13 koji utječe na veličinu grafa sa slike 2.15.

## Memorijski zahtjevi

Tijekom programske implementacije nije se posvetila posebna pažnja optimizaciji korištenja radne memorije računala. Za pamćenje udaljenosti između svih parova poligona koriste se polja od  $|V| \times |V|$  elemenata, gdje je  $|V|$  broj vrhova dualnog grafa poligonalne mreže.

Rezultat toga je ograničenost implementiranog postupka na rad sa modelima sastavljenima od otprilike najviše 1500 poligona, a posljedica toga je korištenje pojednostavljenih modela za testiranje koji su izgubili svojstva simetričnosti.

Veliki  $\epsilon$  iz formula 2.11, 2.12 i 2.13 dodatno povećava memorijske zahtjeve jer se gradi veća mreža sa slike 2.15 za koju je potrebno računati najveći protok tj. minimalni rez.

## Moguća poboljšanja

Implementirani postupak je bez sumnje moguće poboljšati da bolje koristi memoriju računala. Autori postupka u (Katz i Tal, 2003) su uspjeli segmentirati modele od 67170 i 654666 poligona, iako im je za to trebalo 244 i 1654 sekundi.

U vidu ubrzanja postupka moglo bi se paralelizirati algoritme koji to dozvoljavaju. Dodatno ubrzanje bi se moglo postići prilagodbom postupka za pokretanje na programibilnim grafičkim karticama.

Sam postupak bi se mogao kombinirati s drugim segmentacijskim postupcima čime bi se možda dobili bolji rezultati. Ako bi primjena postupka to zahtijevala, ovaj postupak bi se mogao i zanemariti i odabrati neki prikladniji. Ima ih, na sreću, veliki broj.

## 5. Zaključak

Segmentacija objekata zadanih poligonalnom mrežom važan je i koristan postupak u području primjene računalne grafike, a i šire. Proteklih godina razvijeni su razni postupci koji na specifične načine segmentiraju CAD modele.

Iz velikog broja literature vidljivo je da je područje razvoja segmentacijskih postupaka vrlo aktivno. Pošto nema savršenog postupka segmentacije, još uvijek su moguća brojna poboljšanja postojećih postupaka. Također ima mjesta i za neke nove ideje.

# LITERATURA

- Alexander Agathos, Ioannis Pratikakis, Stavros Perantonis, Nikolaos Sapidis, i Philip Azariadis. 3D Mesh Segmentation Methodologies for CAD applications. *Computer-Aided Design Applications*, 4:827–841, 2007.
- M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, i A. Tal. Mesh Segmentation - A Comparative Study. *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, 2006a. doi: 10.1109/SMI.2006.24.
- Marco Attene, Bianca Falcidieno, i Michela Spagnuolo. Hierarchical mesh segmentation based on fitting primitives, 2006b. ISSN 0178-2789.
- M. Botsch, S. Steinberg, S. Bischoff, i L. Kobbelt. OpenMesh – a generic and efficient polygon mesh data structure. U *OpenSG Symposium*, 2002.
- Xiaobai Chen, Aleksey Golovinskiy, i Thomas Funkhouser. A benchmark for 3D mesh segmentation, 2009. ISSN 07300301.
- Y. H. Chen i C. Y. Liu. Robust segmentation of CMM data based on NURBS, 1997. ISSN 0268-3768.
- Thomas H. Cormen, Charles E. Leiserson, i Ronald L. Rivest. *Introduction to Algorithms*, *Second Edition*, svezak 7. 2001. ISBN 0262032937. doi: 10.2307/2583667.
- M. Garland, Andrew Willmott, i P. S. Heckbert. Hierarchical face clustering on polygonal surfaces. U *Proceedings of the 2001 symposium on Interactive 3D graphics*, stranice 49–58, 2001. ISBN 1581132921. doi: 10.1145/364338.364345.
- Lavoué Guillaume, Dupont Florent, i Baskurt Atilla. Curvature tensor based triangle mesh segmentation with boundary rectification. U *Proceedings of Computer Graphics International Conference, CGI*, stranice 10–17, 2004. ISBN 0769521711. doi: 10.1109/CGI.2004.1309187.

- Hugues Hoppe. Progressive meshes. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, stranice 99–108, 1996. ISSN 00978930. doi: 10.1145/237170.237216.
- Lotan Kaplansky i Ayellet Tal. Mesh Segmentation Refinement. *Computer Graphics Forum*, 28:1995–2003, 2009. ISSN 01677055. doi: 10.1111/j.1467-8659.2009.01578.x.
- Sagi Katz i Ayellet Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts, 2003. ISSN 07300301.
- Sagi Katz, George Leifman, i Ayellet Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer (Pacific Graphics)*, 21:649–658, 2005. ISSN 00104485. doi: 10.1007/s00371-005-0344-9.
- Yunjin Lee, Seungyong Lee, Ariel Shamir, Daniel Cohen-Or, i Hans Peter Seidel. Intelligent mesh scissoring using 3D snakes. U *Proceedings - Pacific Conference on Computer Graphics and Applications*, stranice 279–287, 2004. ISBN 0-7695-2234-3. doi: 10.1016/B0-12-227410-5/00419-1.
- Yunjin Lee, Seungyong Lee, Ariel Shamir, Daniel Cohen-Or, i Hans Peter Seidel. Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design*, 22:444–465, 2005. ISSN 01678396. doi: 10.1016/j.cagd.2005.04.002.
- Xuetao Li, T. W. Woon, T. S. Tan, i Zhiyong Huang. Decomposing polygon meshes for interactive applications. *ACM Symp on Intractive 3D Graphics*, stranice 35–42, 2001.
- Rong Liu i Hao Zhang. Segmentation of 3D meshes through spectral clustering. U *Proceedings - Pacific Conference on Computer Graphics and Applications*, stranice 298–305, 2004. ISBN 0769522343. doi: 10.1109/PCCGA.2004.1348360.
- Michela Mortara, Giuseppe Patané, Michela Spagnuolo, Bianca Falcidieno, i Jarek Rossignac. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica (New York)*, 38:227–248, 2003. ISSN 01784617. doi: 10.1007/s00453-003-1051-4.
- Jörg Stefan Praßni, Timo Ropinski, i Klaus Hinrichs. Uncertainty-aware guided volume segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 16:1358–1365, 2010. ISSN 10772626. doi: 10.1109/TVCG.2010.208.

- Andreas Richtsfeld, Thomas Morwald, Johann Prankl, Michael Zillich, i Markus Vincze. Segmentation of unknown objects in indoor environments. U *IEEE International Conference on Intelligent Robots and Systems*, stranice 4791–4796, 2012. ISBN 9781467317375. doi: 10.1109/IROS.2012.6385661.
- Tian Shen, Hongsheng Li, i Xiaolei Huang. Active volume models for medical image segmentation. *IEEE Transactions on Medical Imaging*, 30:774–791, 2011. ISSN 02780062. doi: 10.1109/TMI.2010.2094623.
- Anthony Sherbondy, Mike Houston, i Sandy Napel. Fast Volume Segmentation with Simultaneous Visualization Using Programmable Graphics Hardware. U *Proceedings of the IEEE Visualization Conference*, stranice 171–176, 2003. ISBN 0-7803-8120-3. doi: 10.1109/VISUAL.2003.1250369.
- Shymon Shlafman, Ayellet Tal, i Sagi Katz. Metamorphosis of Polyhedral Surfaces using Decomposition. *Computer Graphics Forum*, 21:219–228, 2002. ISSN 01677055. doi: 10.1111/1467-8659.00581.
- A. Suinesiaputra, P. de Koning, E. Zudilova-Seinstra, J. Reiber, i R. van der Geest. A 3D MRA Segmentation Method Based on Tubular NURBS Model. U *Proceedings 17th Scientific Meeting, International Society for Magnetic Resonance in Medicine*, svezak Honolulu, stranica 3829, 2009. ISBN 1545-4428.
- K. Wu i M. D. Levine. 3D part segmentation using simulated electrical charge distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 1997. ISSN 0162-8828. doi: 10.1109/34.632982.
- Emanoil Zuckerberger, Ayellet Tal, i Shymon Shlafman. Polyhedral surface decomposition with applications. U *Computers and Graphics (Pergamon)*, svezak 26, stranice 733–743, 2002. doi: 10.1016/S0097-8493(02)00128-0.

## POSTUPCI SEGMENTACIJE OBJEKATA ZADANIH POLIGONALNOM MREŽOM

### Sažetak

Ovaj rad proučava postupke segmentacije modela primjenjive u računalnoj grafici. Posebna pažnja je posvećena postupcima koji hijerarhijski analiziraju i segmentiraju modele zadane poligonalnom mrežom. Dodatno se proučavaju konteksti u kojima su ti postupci korisni i primjenjivi, s naglaskom na određivanje skeletne strukture i povezivanje segmentiranog modela sa skeletnom strukturom. Jedan od postupaka je programski implementiran i dobiveni rezultati su ocijenjeni.

**Ključne riječi:** segmentacija poligonalnih mreža, neizrazita dekompozicija, određivanje skeletne strukture

## SEGMENTATION OF POLYGONAL OBJECTS

### Abstract

Master's thesis presented here analyzes mesh segmentation methods used in computer graphics. Special emphasis is given to methods which hierarchically analyze and segment polygon mesh models. Applications which benefit from mesh segmentation are also analyzed, with emphasis on control skeleton extraction. One of the methods is implemented and its results are reviewed.

**Keywords:** polygon mesh segmentation, fuzzy decomposition, control skeleton extraction