

**UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA**

Autori:

Dr Zoran Milojević, vanredni profesor

Dr Slobodan Navalušić, redovni profesor

Dr Milan Zeljković, redovni profesor

Dr Slobodan Tabaković, vanredni profesor

Dr Marija Vićević, docent

Dr Aleksandar Živković, docent

**PROGRAMSKO REŠENJE ZA SIMULACIJU
PROCESA VIŠEOSNE OBRADE ZASNOVANO NA
TRI-DEKSEL MODELU**

- TEHNIČKO REŠENJE -

Novi Sad, 2013. godine

Podaci o tehničkom rešenju

Vrsta tehničkog rešenja	M-85: Prototip, nova metoda, softver, standardizovan ili atestiran instrument, nova genetska proba, mikroorganizmi
Autori tehničkog rešenja	Autori: Dr Zoran Milojević, vanredni profesor Dr Slobodan Navalusić, redovni profesor Dr Milan Zeljković, redovni profesor Dr Slobodan Tabaković, vanredni profesor Dr Marija Vićević, docent Dr Aleksandar Živković, docent
Naziv tehničkog rešenja	PROGRAMSKO REŠENJE ZA SIMULACIJU PROCESA VIŠEOSNE OBRADE ZASNOVANO NA TRI- DEKSEL MODELU
Za koga je rađeno tehničko rešenje	Programsko rešenje za simulaciju procesa višeosne obrade zasnovano na tri-deksel modelu je razvijeno za potrebe istraživanja koja se sprovode na Fakultetu tehničkih nauka u Novom Sadu u okviru projekta „Savremeni prilaz u razvoju specijalnih rešenja uležištenja u mašinstvu i medicinskoj protetici” ev. broj TR 35025
Ko koristi tehničko rešenje	Programsko rešenje se koristi za potrebe istraživanja i nastave na Fakultetu tehničkih nauka u Novom Sadu
Godina izrade tehničkog rešenja	2013.
Verifikacija rezultata	Od strane recenzenata: Prof. dr Miloš Glavonjić Prof. dr Ratko Obradović
Ko je prihvatio tehničko rešenje	Nastavno-naučno veće Fakulteta tehničkih nauka iz Novog Sada
Primena rezultata	Istraživačko – razvojno sredstvo

SADRŽAJ

1.0 OBLAST PRIMENE TEHNIČKOG REŠENJA	1
2.0 TEHNIČKI PROBLEM	1
3.0 DETALJAN OPIS TEHNIČKOG REŠENJA	4
3.1. Generisanje tri-deksel tekstura priprema	5
3.2. Generisanje tri-deksel struktura priprema	8
3.3. Generisanje matrica transformacija instanci alata	9
3.4 Transformacija alata i generisanje tri deksel tekstura alata	11
3.5 Oduzimanje alata od obradka	12
3.6 Prikaz tri deksel modela obradka u procesu simulacije	15
3.7 Eksport tri deksel struktura obradka	15
4.0 SOFTVERSKA IMPLEMENTACIJA PREDLOŽENOG REŠENJA	16
5.0 REZULTATI RAZVIJENOG PROGRAMSKOG REŠENJA	17
6.0 ZAKLJUČAK	18
7.0 LITERATURA	18

1.0 OBLAST PRIMENE TEHNIČKOG REŠENJA

Tehničko rešenje pripada oblasti simulacije višeosne obrade u realnom vremenu. Svoju primenu može naći kako u mašinstvu za simulaciju mašinske obrade, tako i u programskim sistemima gde se zahteva interaktivno oduzimanje 3D modela jednog od drugog. U širem smislu, zbog svoje primenljivosti razvijeno tehničko rešenje može se posmatrati kao deo oblasti proizvodnog mašinstva a zbog principa na kome je zasnovano može se posmatrati i kao deo oblasti kompjuterske grafike .

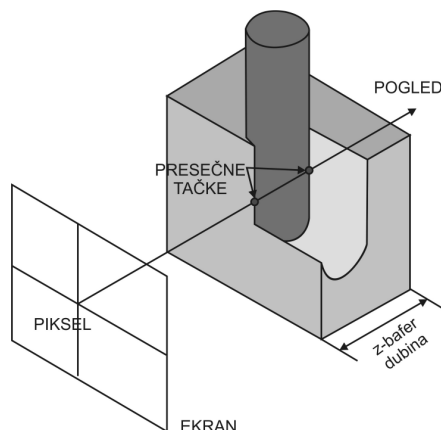
2.0 TEHNIČKI PROBLEM

Prikaz simulacije NC upravljačkog programa, vrlo je značajan sa stanovišta provere i verifikacije tačnosti izradka, kao i eventualnih kolizija u radnom prostoru mašine alatke. Na taj način je moguće, u fazi projektovanja tehnologije izrade obradka na radnom mestu tehnologa - programera, uočiti i korigovati eventualne greške nastale u procesu generisanja NC upravljačkog programa.

U toku simulacije procesa obrade na mašini alatki, najčešća grafička prezentacija prikazuje obradak i alat, pri čemu se alat pomera duž putanje definisane generisanim NC upravljačkom programom. Pri procesu skidanja materijala, vrši se proračun koji uklanja delove materijala sa obradka. Prikaz obradka i alata ostvaruje se tehnikama vezanim za kompjutersku grafiku, najčešće primenom neke od grafičkih biblioteka, koje omogućavaju 3D prikaz alata i obradka.

Kod simulacije procesa skidanjem materijala višeosnom obradom, razvijeni programski sistemi zasnovani su na dva prilaza: egzaktnom i aproksimativnom. Kod egzaktnog prilaza simulaciji NC upravljačkog programa, vrši se primena Bool-ovog logičkog operatora za razliku (oduzimanje) omotača alata generisanog između početnog i krajnjeg položaja u procesu rezanja od modela radnog predmeta. Verifikacija se ostvaruje primenom logičkog operatora za oduzimanje modela izradka od modela zahtevanog dela. Prednost predstavlja tačan prikaz radnog predmeta nakon obrade, pošto su i alat i radni predmet definisani egzaktno, najčešće CSG ili BRep prezentacijom. Hardverska zahtevnost kod ovog metoda može se predstaviti u obliku $O(N^4)$ što predstavlja funkciju broja operacija u zavisnosti od četvrtog stepena broja kretanja alata. U radovima [1-3] primenjen je egzaktni prilaz simulaciji procesa skidanja materijala. Cilj simulacije procesa skidanja materijala je da se obradak prikaže što detaljnije uz što kraće vreme proračuna. Upravo zbog velike hardverske zahtevnosti ovog metoda što se tiče vremena proračuna (iako je obradak definisan izuzetno detaljno), istraživanja su se razvijala u pravcu aproksimativnih pristupa.

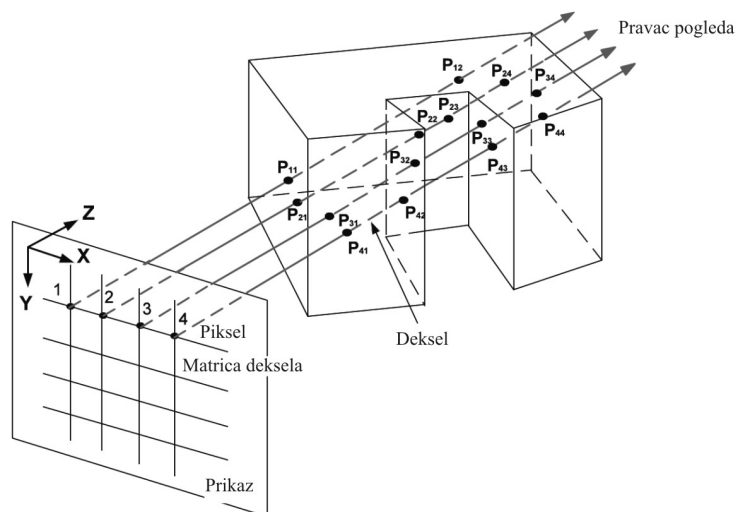
Za početak razvoja aproksimativnih pristupa vezuje se rad [4], gde je obradak predstavljen kao 3D histogram (svaka x,y koordinata ima svoju visinu u z pravcu), tako da je ovaj pristup nazvan i z-map metod. Pristupi [5,6], aproksimiraju površinu izradka sa tačkama u kojima su definisane normale. Proračunom se određivalo podsecanje normala sa omotačem alata u procesu rezanja. Konačna dužina vektora predstavlja višak ili manjak materijala na izradku posle završene obrade. Funkcija zavisnosti broja operacija od broja kretanja alata je $O(N)$, dakle linearna. Aproksimativni pristupi su se uglavnom razvijali u pravcu takozvanih *view-based* metoda (metoda zasnovanih na pogledu ili projekciji) [7-10]. Princip ovog metoda prikazan je na slici 2.1.



Slika 2.1. Princip view-based metode simulacije uklanjanja materijala

Kod view-based metode korišćen je standardni z-bafer algoritam za određivanje skrivenih površina, i za oduzimanje alata od obradka iskorišćene su mogućnosti grafičkih kartica tj. z-bafer grafičke kartice. Obradak je predstavljen slikom u memoriji grafičke kartice. Proračun ovim metodom je bio izuzetno brz, međutim greške koje mogu nastati pri obradi koje nisu vidljive u datoj projekciji, ne mogu da se detektuju.

Proširenje view-based metode prikazano je u radu [8]. U radu je prvi put uveden pojam deksel (depth element), u skladu sa oznakama piksel i voksel. Deksel se može definisati kao solid oblika kvadra, koji odgovara delu obradka koji se nalazi iza piksela u odgovarajućoj projekciji (slika 2.2). Bitna karakteristika ovog pristupa je u tome što svakom piksel-u na ekranu odgovara sortirana lista deksel-a, što omogućava izradu i prikaz šupljih oblika. Struktura svakog deksel-a sadrži podatke: z-bliže, z-dalje, boju i pokazivač na sledeći dexel, ako postoji na obradku (što znači da postoji otvor na obradku). Oduzimanje alata od obradka je svedeno na jednodimenzionalno oduzimanje deksel-a, tako da je ostvarena simulacija procesa rezanja u realnom vremenu. Ovaj metod je idealan za simulaciju 3-osnog glodanja, pošto se struktura dexela alata ne menja pri rezanju. Obradak je kao i kod osnovnog view-based metoda bio zavisao od definisane projekcije, pri čemu je za promenu pogleda bilo potrebno ponoviti proces simulacije.

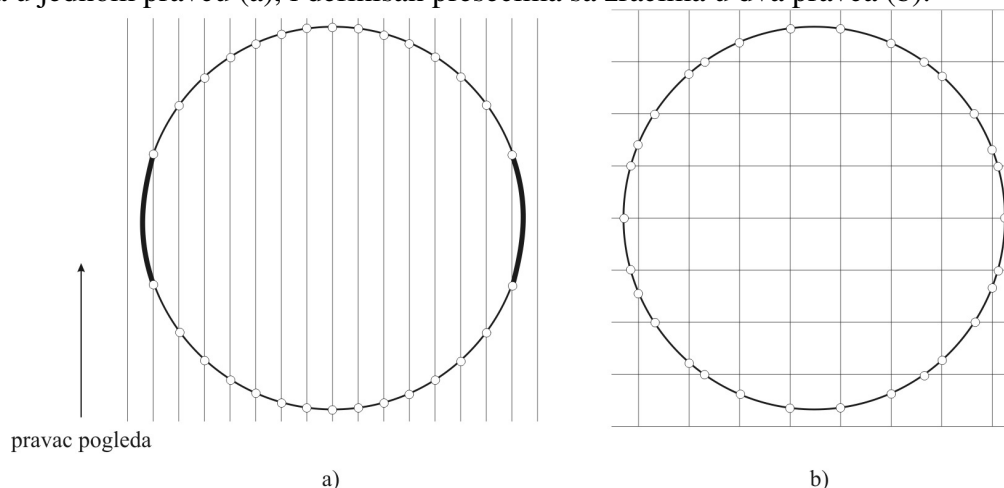


Slika 2.2. Princip opisa modela dekselima

U radu [9], predstavljen je takozvani prošireni deksel metod (Extended dexel). Autori su predstavili novi algoritam pri čemu metod nije ograničen definisanim pogledom na

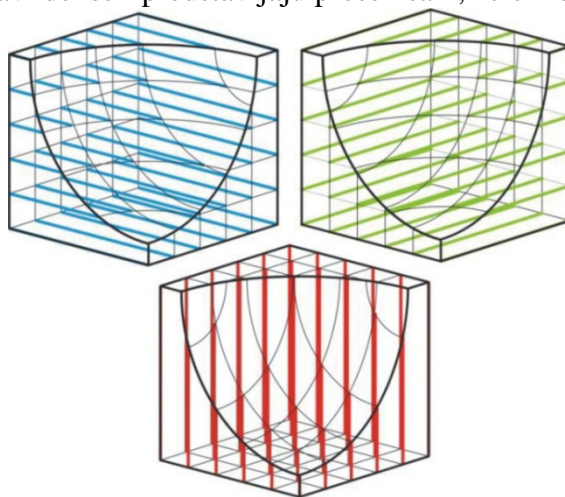
pripremak. Za prikaz su odabrali konturno linijsko prikazivanje, pa je bilo moguće pri simulaciji NC programa, dinamički menjati pogled na obradak.

Jedan od nedostataka deksel pristupa predstavlja činjenica da su površine koje su paralelne pravcu pogleda loše opisane. Takođe, za zahtevanu tačnost opisa obradka, potrebno je definisati dosta veliki broj podele mreže. Zbog prethodno navedenih nedostataka opisa modela deksel pristupom, došlo se na ideju da se umesto jednog pravca pogleda, model opiše dekselima u tri pravca (x,y i z pravac) [11]. Ovakav pristup daje ravnomerniji opis površina modela. Na slici 2.3, radi jednostavnosti prikaza, dat je opis kružnice definisan presecima sa zracima u jednom pravcu (a), i definisan presecima sa zracima u dva pravca (b).



Slika 2.3. Kružnica opisana zracima u jednom pravcu (a) i zracima u dva pravca (b)

Sa slike 2.3 se vidi da je opis kružnice ravnomerniji za dva pravca, pri čemu je za oba slučaja upotrebljen isti broj zraka. Takođe slab opis površina paralelnih sa pravcem pogleda prikazan je na slici 2.3. a (podebljani delovi kružnice). Prikaz opisa 3D modela u tri pravca dat je na slici 2.4, gde plavi dekseli predstavljaju presek sa x, zeleni sa y i crveni sa z zracima.



Slika 2.4. Opis modela sa tri dekselima

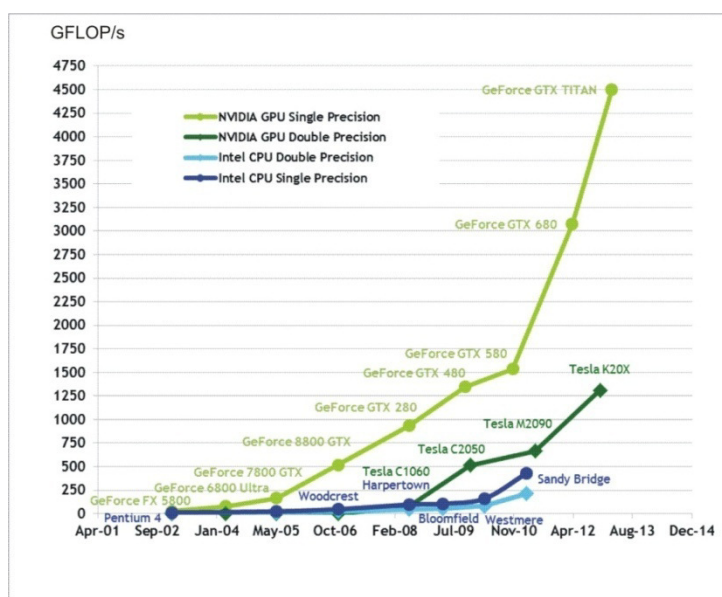
Tri deksel opis modela, svoju primenu nalazi u svim oblastima gde je potrebno interaktivno oduzimanje 3D modela jednog od drugog, kao što su simulacija uklanjanja alata od obradka [11][12][13], ili slobodoručno modelovanje proizvoda pokretima ruke u virtualnom okruženju [14].

Koncepcija predloženog tehničkog rešenja, predstavlja programski sistem za simulaciju višeosne obrade primenom tri deksel opisa obradka i alata. U skladu sa savremenim

trendovima, kompletan proračun uklanjanja materijala izvršava se u grafičkoj kartici, što značajno skraćuje vreme proračuna i omogućava interaktivan prikaz. Pripremak i alat su definisani u STL (STereo Litography) formatu. Alat je u procesu uklanjanja materijala između dve pozicije predstavljen sa instancama (interpolirane pozicije između početnog i krajnjeg položaja). Ovaj metod je izabran za razliku od generisanja omotača alata između dve pozicije zbog jednostavnosti implementacije i zbog mogućnosti primene alata slobodnog oblika, što omogućava da se razvijeni programski sistem osim primene u simulaciji višeosne mašinske obrade, primeni i u oblasti slobodnoručnog modelovanja proizvoda. Generisani izradak je moguće eksportovati kao skup tačaka sa normalama ili kao STL fajl, što omogućava import u neki od komercijalnih CAD sistema radi naknadne analize. Takođe programski sistem je otvoren za nadogradnju, tako da je moguće proširenje izvršiti u cilju podrške heptičkih uređaja gde bi se umesto iz NC upravljačkog programa, pozicija alata u sistemu definisala matricom pozicije definisane pozicijom heptičkog uređaja u prostoru.

3.0 DETALJAN OPIS TEHNIČKOG REŠENJA

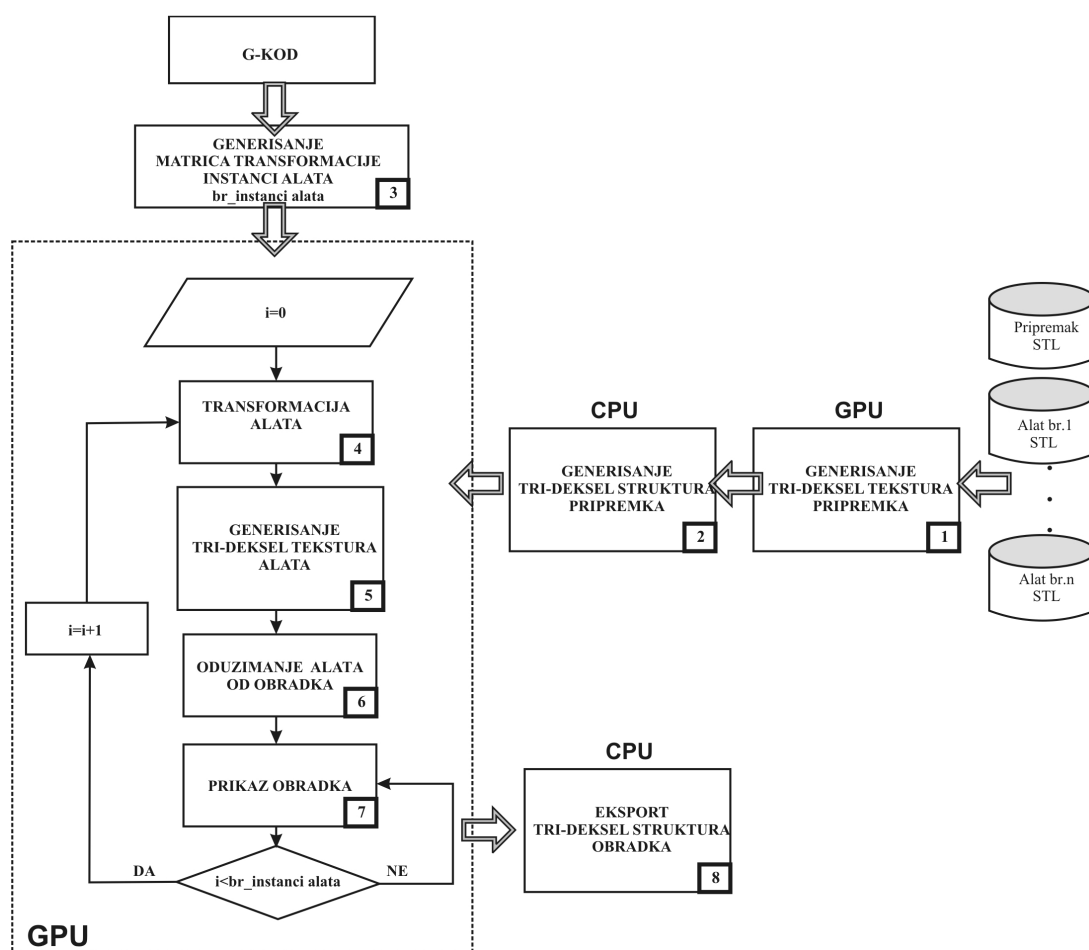
U poslednjih desetak godina GPU (Graphics Processing Unit – Jedinica za procesiranje grafike), se dosta brže razvijala od CPU (Central Processing Unit – Centralne procesne jedinice). Na slici 3.1 [15] dato je uporedno poređenje procesorske snage (GFLOP/s Giga FLOating Points per second – milijardu operacija u pokretnom zarezu u sekundi) GPU i CPU jedinica.



Slika 3.1. Poređenje procesorske snage GPU i CPU jedinica

Jasno se vidi da su moderni GPU u velikoj prenosti u odnosu na CPU što se tiče procesorske snage. Razlog za ovakav odnos, leži u tome da su GPU optimizovani za paralelno izvršavanje proračuna za razliku od CPU-a. Tako u današnje vreme dok moderni CPU sadrže 4 jedinice, GPU sadrže po nekoliko stotina manjih procesorskih jedinica, koje su optimizovane za paralelno procesiranje. Porast procesorske snage GPU, doveo je do razvoja GPGPU (General-Purpose computing on Graphics Processor Units ili do proračuna opšte namene na jedinicama za procesiranje grafike), tako da se osim osnovnog zadatka grafičke kartice (da generiše sliku na monitoru), grafička kartica može upotrebiti za složene proračune u svim poljima tehnike.

Model razvijenog programskog rešenja za simulaciju višeosne obrade primenom tri-deksel modela prikazan je na slici 3.2. Kao što je prikazano na slici 3.2, modeli priprema i alata importuju se u sistem u obliku STL fajlova. Sledeći korak predstavlja generisanje tri deksel tekstura priprema u GPU jedinici. Zatim se na osnovu generisanih tekstura, formiraju strukture za svaki deksel pomoću CPU jedinice. Na osnovu G-koda, formiraju se matrice instanci alata. Generisane strukture deksela, se prebacuju u memoriju GPU jedinice, i kompletan proračun oduzimanja alata od obradka se vrši u GPU jedinici. Prvi korak predstavlja transformisanje alata i generisanje tekstura alata, nakon toga se vrši oduzimanje alata od obradka i prikaz modifikovanog obradka nakon oduzimanja. Isti postupak se ponavlja za svaku instancu alata. Nakon završene simulacije, deksel strukture generisanog izradka, mogu se prebaciti iz GPU jedinice, na dalju obradu pomoću CPU jedinice. U nastavku se daje detaljan prikaz karakterističnih koraka razvijenog programskog rešenja.

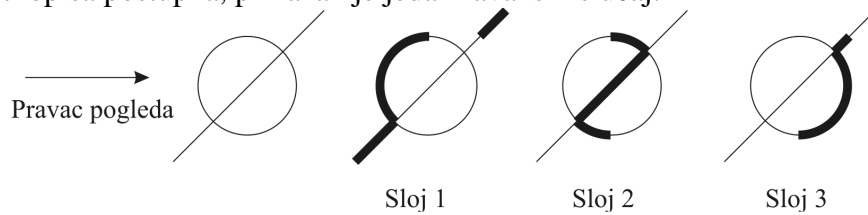


Slika 3.2. Model razvijenog programskog rešenja za simulaciju višeosne obrade primenom tri-deksel modela

3.1 Generisanje tri-deksel tekstura priprema

Model priprema se u programski sistem učitava u STL formatu (kao skup trougaonih poligona sa definisanom normalom za svaki poligon). Pošto priprema može da bude kompleksnog oblika (na zraku u jednom pravcu može da bude više prodora sa pripremkom), da bi se odredili svi prodori x, y i z zraka, sa pripremkom, primenjena je Depth peeling ("ljuštenje" po dubini) metoda. Metoda je prvi put uvedena u računarsku grafiku u cilju bržeg određivanja transparentnosti modela [16]. Model se senči (renderuje) u onoliko prolaza kolika

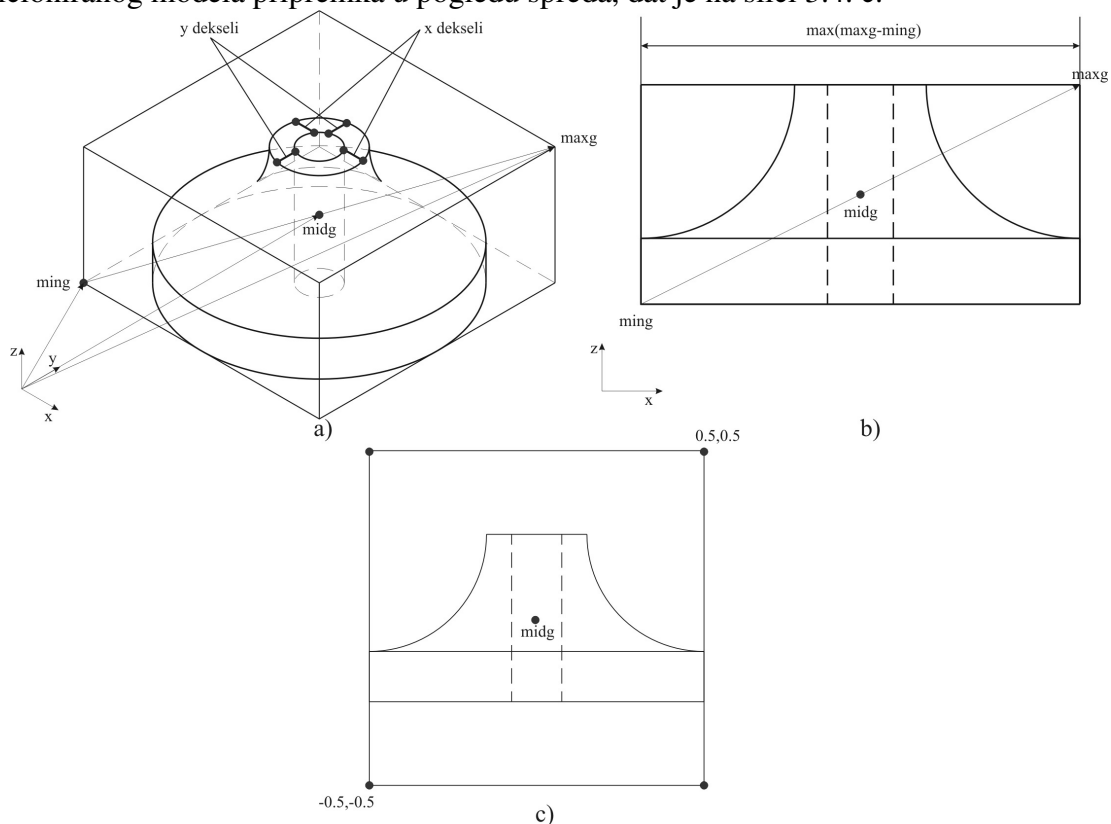
je maksimalna kompleksnost tačke u pogledu (zaklonjenih površina). Na slici 3.3, radi jednostavnosti opisa postupka, prikazan je jedan ravanski slučaj.



Slika 3.3. Prikaz Depth peeling metode na ravanskom slučaju [16]

Prvi korak predstavlja, postavljanje objekta u željeni položaj i definisanje pravca pogleda. Zatim se određuju površine koje su vidljive u datom pogledu (sloj 1 na slici 3.3). Nakon toga se predmet opet senči, pri čemu se prethodno generisane površine zanemaruju i dobija se sloj 2 na slici 3.3. Postupak se završava onda kada više nema površina za senčenje. Sa slike 3.3, može se videti da je kompleksnost objekta u definisanom pravcu pogleda tri.

Za pripremak prikazan na slici 3.4, zraci u y-pravcu (y-dekseli) i u x-pravcu (x-dekseli), seku pripremak u 4 tačke, tako da je kompleksnost pripremk u y i x pravcu 4, dok je kompleksnost u z pravcu 2. Ovo znači da će broj prolaza senčenja modela za generisanje svih površina pripremk biti 4 u x, 4 u y i 2 u z pravcu. Prvi korak pre generisanja tri-deksel tekstura pripremk, predstavlja određivanje granične oblasti pripremk (*ming* i *maxg*). Zatim se određuje središnja tačka pripremk *midg*, kao i maksimalna projekcija na x,y ili z osu vektora $\vec{maxg} - \vec{ming}$ kao što je prikazano na slici 3.4. b. Konačno, model pripremk se skalira i transformiše na takav način da se središnja tačka *midg*, postavi u koordinatni početak, i da granična oblast za generisanje teksture bude u granicama (-0.5,-0.5 ÷ 0.5,0.5). Prikaz pozicioniranog modela pripremk u pogledu spreda, dat je na slici 3.4. c.

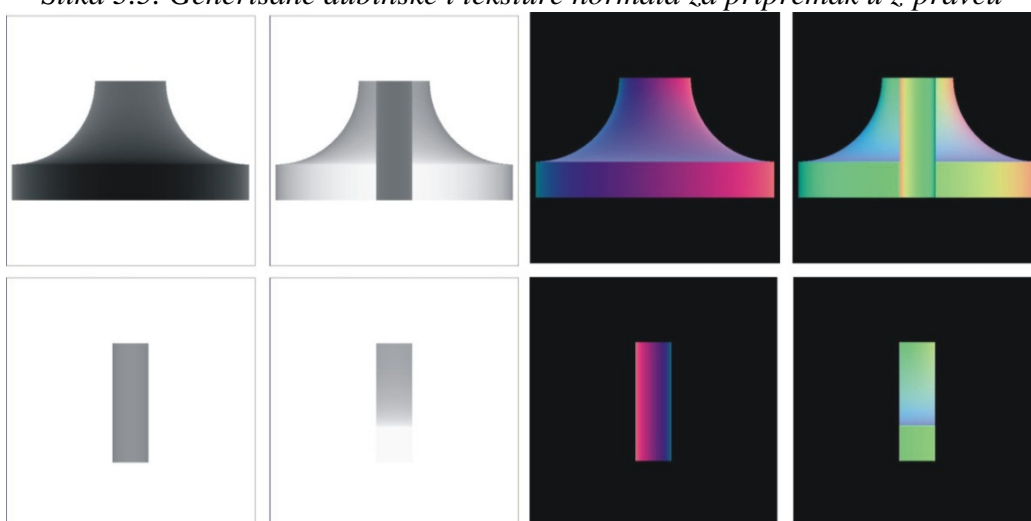


Slika 3.4. a) Metoda određivanja granica pripremk i središnje tačke, b) određivanje maksimalne projekcije pripremk, c) pozicija pripremk nakon transformacije spremnog za generisanje tekstura u y-pravcu

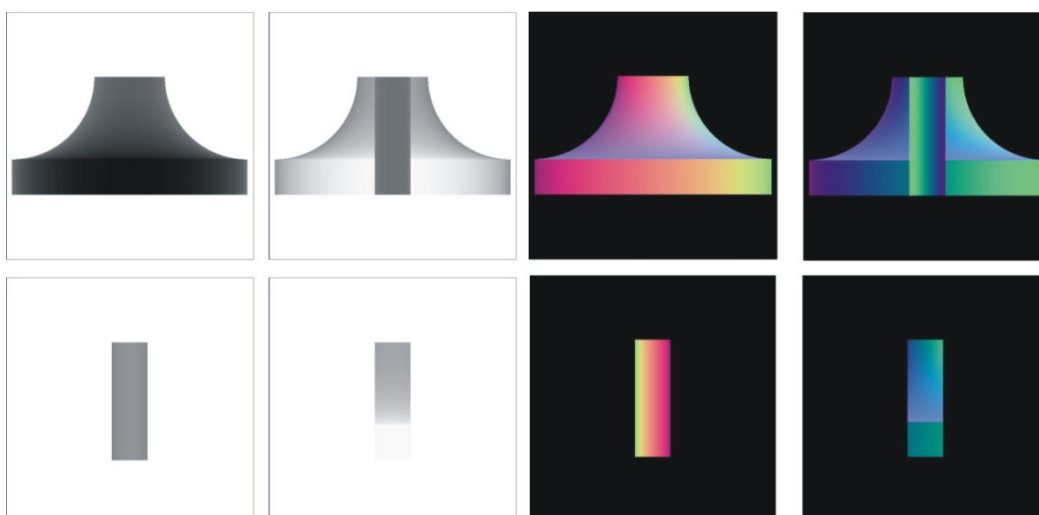
Pri generisanju tekstura modela, za svako senčenje modela, generišu se dve teksture, prva u kojoj svaki piksel predstavlja dubinu tačke u pravcu pogleda u rasponu $0.0 \div 1.0$, i druga u kojoj je normala tačke na modelu kodirana u rasponu $0.0 \div 1.0$ u x, y i z pravcu i predstavljena odgovarajućom bojom [18]. Na ovaj način generisane su teksture priprema u z-pravcu (slika 3.5), y-pravcu (slika 3.6) i x-pravcu (slika 3.7).



Slika 3.5. Generisane dubinske i texture normala za pripremak u z-pravcu



Slika 3.6. Generisane dubinske i texture normala za pripremak u y-pravcu

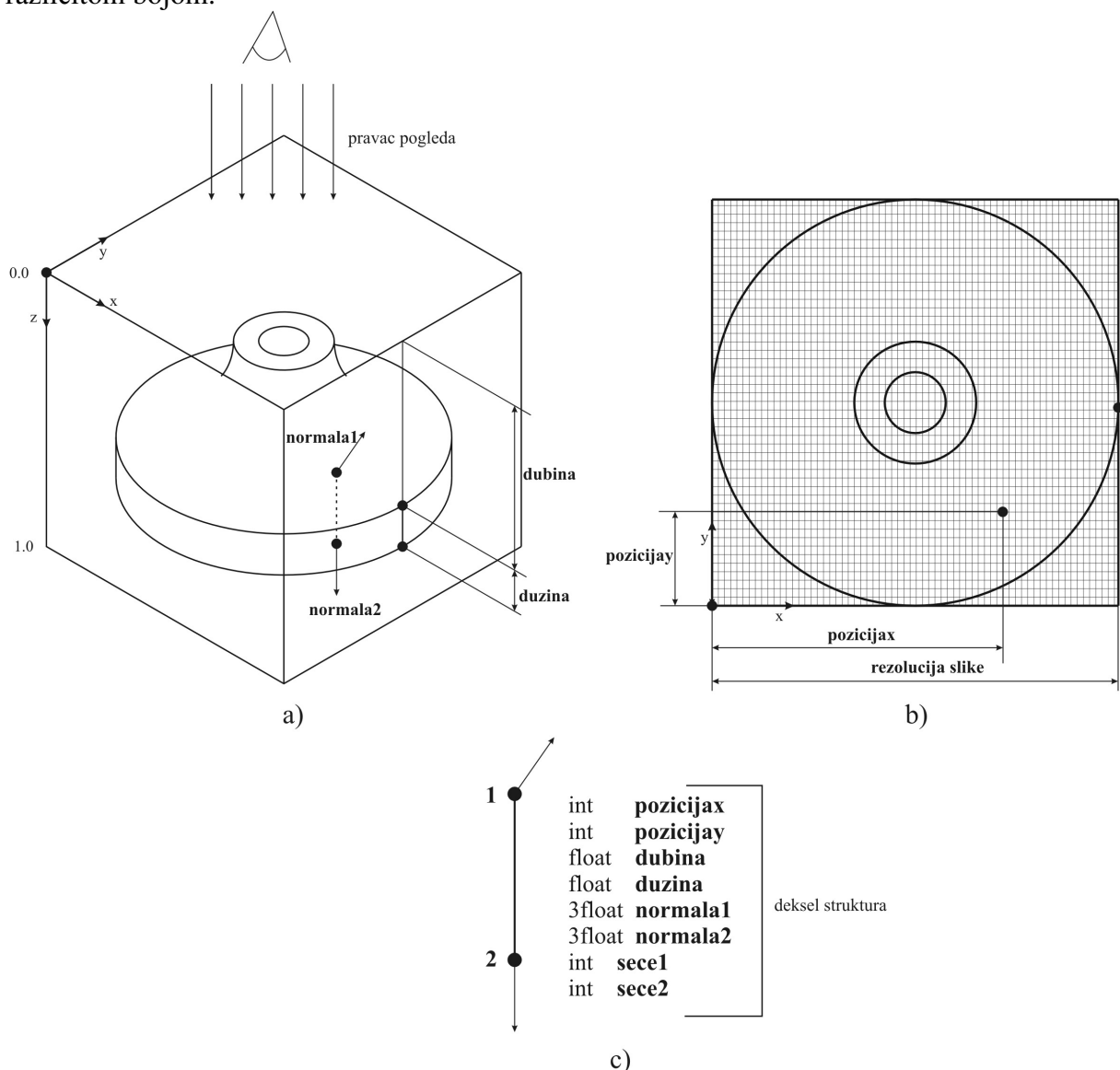


Slika 3.7. Generisane dubinske i texture normala za pripremak u x-pravcu

Nakon generisanja svih tekstura u sva tri ortogonalna pogleda, teksture se prebacuju iz GPU jedinice u glavnu memoriju da bi se izvršila dalja obrada i konvertovanje u tri-deksel strukture modela priprema.

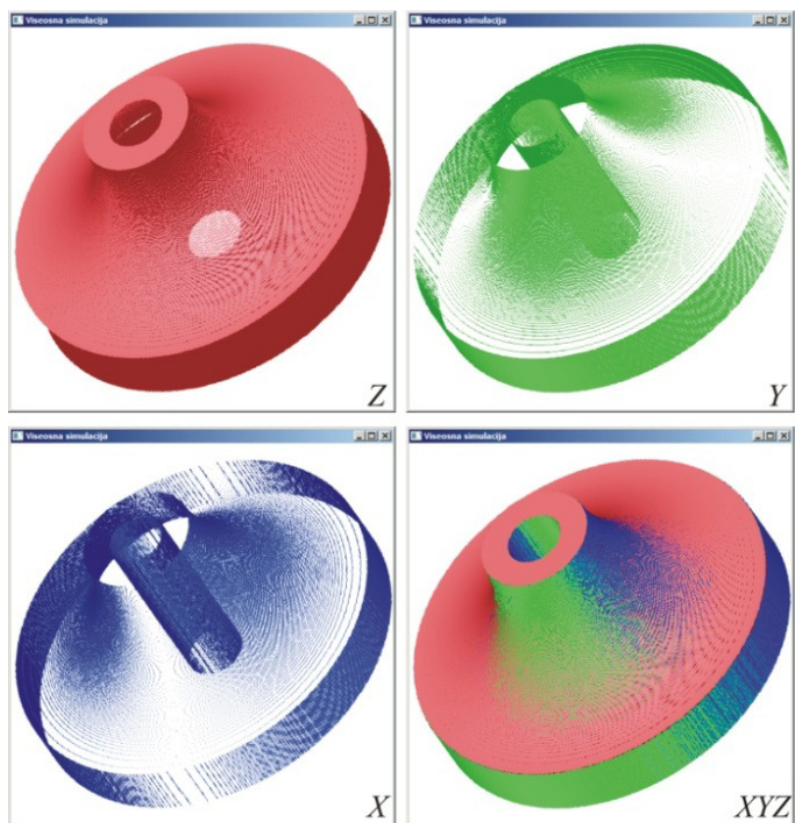
3.2. Generisanje tri-deksel struktura priprema

Da bi se tri-deksel model mogao prebaciti ponovo u memoriju GPU jedinice i da bi se moglo izvršiti oduzimanje alata od priprema, prvi korak predstavlja definisanje strukture za deksela. Struktura deksela definisana je sa osam parametara, koji potpuno opisuju deksel. Na slici 3.8 b., prikazana je tekstura u pogledu odozgo generisana prethodnim metodom. Svaka tekstura ima svoju rezoluciju (*rezolucija slike*). Za deksel prikazan na slici 3,8 a, parametri *dubina* i *duzina*, definišu koordinatu početne i krajnje tačke deksela u pravcu pogleda, dok parametri *pozicijax* i *pozicijay*, definišu poziciju deksela u projektnoj ravni. Takođe, iz tekstura sa normalama, određuju se normale svake tačke deksela *normala1* i *normala2*. Prikaz kompletne strukture za jedan deksel, dat je na slici 3,8 c. Poslednja dva parametra strukture *sece1* i *sece2*, stavljena su u strukturu iz razloga da se pored informacija o geometriji deksela, dodaju i informacije da li je deksel sečen u procesu simulacije, i da se prikaže različitom bojom.



Slika 3.8. Parametri za generisanje strukture deksela

Primer generisanog tri deksel modela priprema i njegov prikaz u programskom sistemu dat je na slici 3.9.

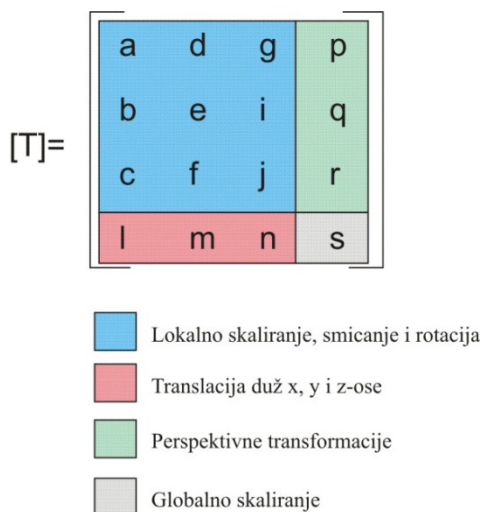


Slika 3.9. Generisani tri-deksel model priprema, Z, Y, X pravac i sva tri pravca

Generisane strukture deksela priprema u sva tri pravca, prebaciju se u memoriju GPU jedinice u obliku tri bafera (poseban bafer za svaki deksel pravac). Takođe za potrebe oduzimanja alata od obradka, formira se i dodatni bafer u GPU jedinici (privremeni bafer), u koji će biti smeštani rezultati oduzimanja.

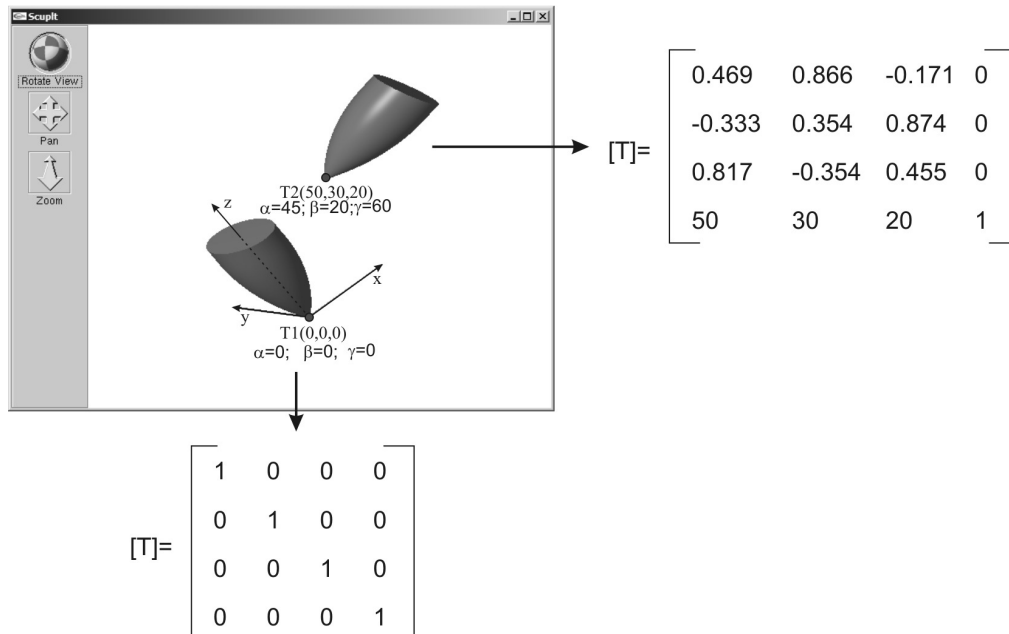
3.3 Generisanje matrica transformacija instanci alata

Pozicija i orijentacija alata u programskom sistemu, definisana je NC upravljačkim programom kod simulacije procesa višeosne obrade i predstavljeni su matricom transformacije [T]. Struktura matrice transformacije prikazana je na slici 3.10.



Slika 3.10. Struktura matrice transformacije

Kao što je prikazano na slici 3.10, matrica $[T]$, 4×4 , sadrži u sebi polja koja predstavljaju lokalno skaliranje, smicanje i rotaciju, translaciju duž x, y i z osa, polja perspektivnih transformacija i globalni faktor skaliranja. Na slici 3.11., prikazana su dva položaja alata sa njihovim matricama transformacije.



Slika 3.11. Položaji alata u programskom sistemu sa odgovarajućim matricama transformacija

Za proračun interpolacionih matrica transformacije alata između dva položaja, iskorišćena je VTK (Visualization ToolKit) biblioteka [19] i klasa *vtkTransformInterpolator*. Parametri potrebni za interpolaciju su matrice transformacija alata u početnom i krajnjem položaju i broj željenih interpolacionih koraka. Na slici 3.12, dat je prikaz komentarisano koda za postupak interpolacije.

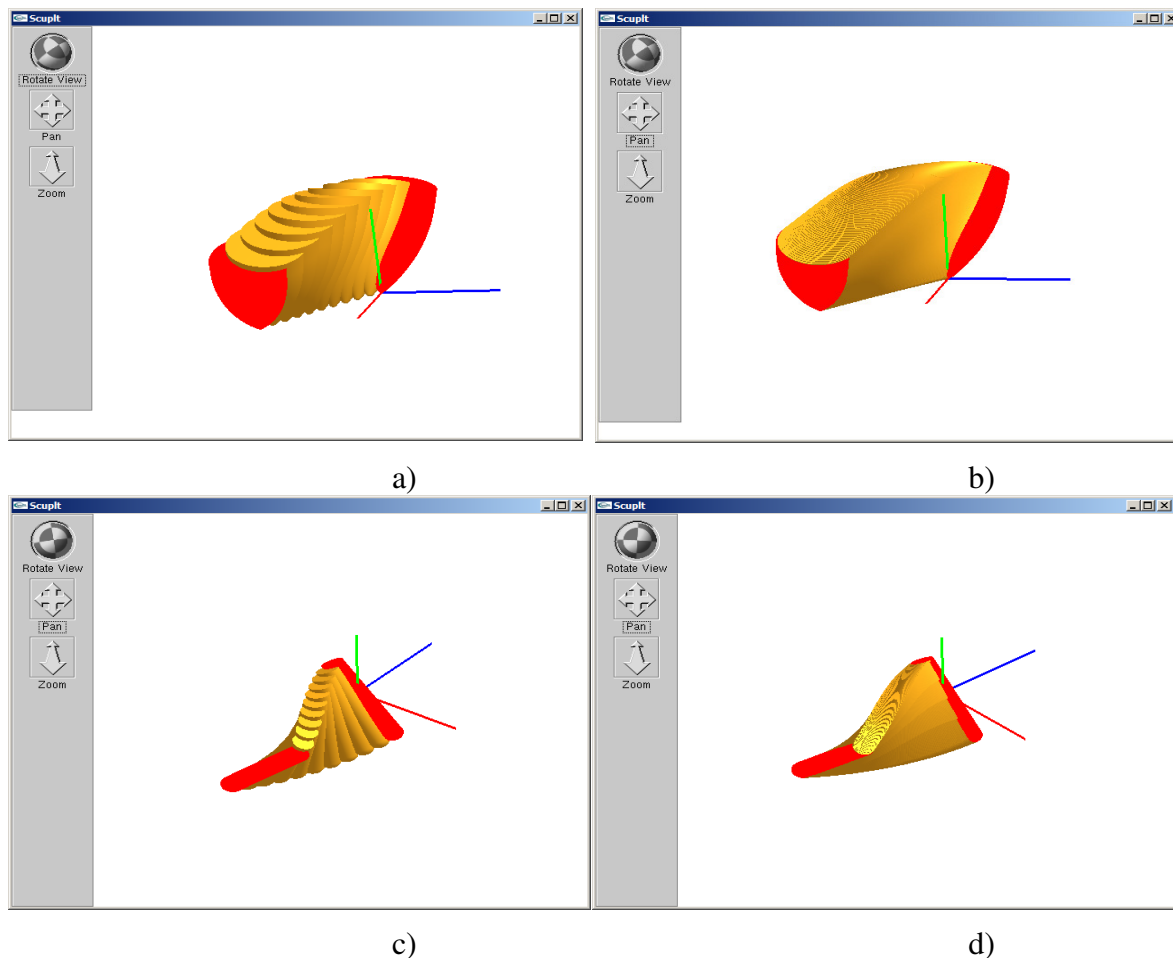
```
vtkSmartPointer<vtkTransform> transp = vtkSmartPointer<vtkTransform>::New();
transp->SetMatrix(vtkmat1); //Početna matrica položaja alata
vtkSmartPointer<vtkTransform> transp1 = vtkSmartPointer<vtkTransform>::New();
transp1->SetMatrix(vtkmat2); //Krajnja matrica položaja alata
vtkSmartPointer<vtkTransformInterpolator> interpolator =
vtkSmartPointer<vtkTransformInterpolator>::New(); //Klasa za interpolaciju
interpolator->AddTransform(0.0,transp); //Početni položaj interpolacije
interpolator->AddTransform(1.0,transp1); //Krajnji položaj interpolacije

za svaki korak i od 0 do bkoraka izračunaj matricu transformacije instance alata vtkmat

vtkSmartPointer<vtkTransform> transfc = vtkSmartPointer<vtkTransform>::New();
interpolator->InterpolateTransform(i*(maxt-mint)/bkoraka,transfc);
vtkSmartPointer<vtkMatrix4x4> vtkmat=vtkSmartPointer<vtkMatrix4x4>::New();
transfc->GetMatrix(vtkmat);
```

Slika 3.12. Procedura proračuna matrica transformacije instanci alata

Na slici 3.13., prikazane su generisane instance alata slobodnog oblika (a i b), i glodala sa zaobljenim čelom (c i d), za različiti broj interpolacionih koraka.



Slika 3.13. Generisane instance alata, alat slobodnog oblika a) 10 interpolacionih koraka, b) 100 inerpolacionih koraka; glodalo sa zaobljenim čelom c) 10 interpolacionih koraka, d) 100 interpolacionih koraka

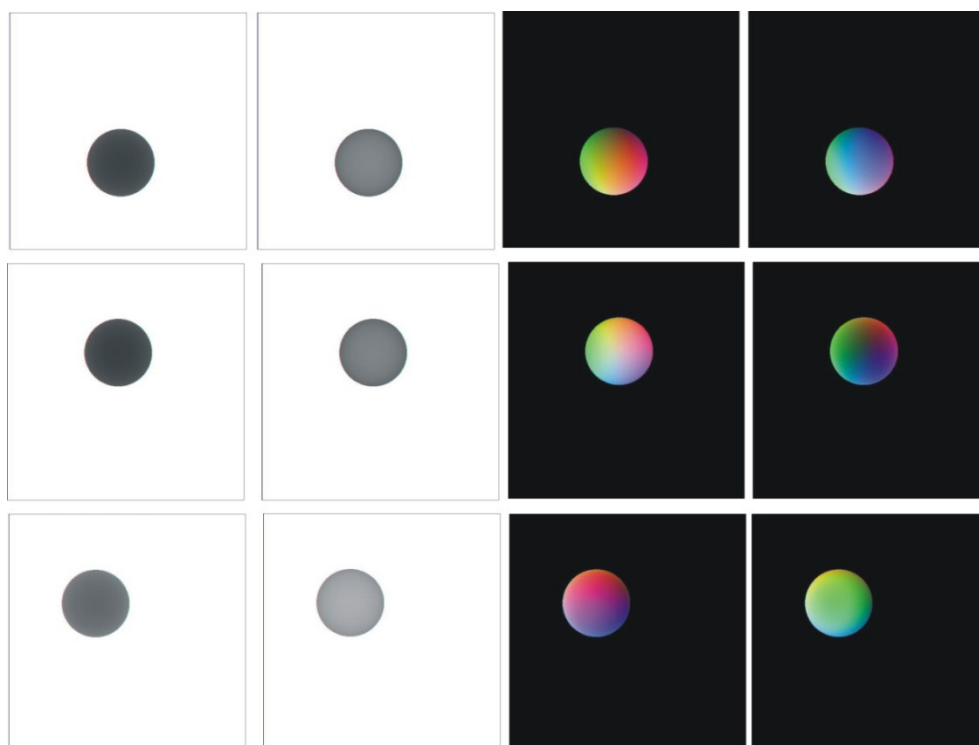
3.4 Transformacija alata i generisanje tri deksel tekstura alata

Pošto je potrebno napraviti tri-deksel strukture za svaku instancu alata, teksture alata se generišu istim postupkom kao i teksture pripremka, s tim što zbog promene položaja alata, alat se transformiše za transformacionu matricu instance alata, određenu u prethodnom poglavlju. Važnu stvar predstavlja činjenica, da se generisane teksture ne moraju prebaciti iz GPU jedinice u glavnu memoriju, već se oduzimanje može izvršiti direktno u GPU jedinici.

Primer generisanih tri deksel tekstura jedne pozicije alata sa zaobljenim čelom u procesu rezanja dat je slici 3.14, dok su na slici 3.15 prikazane tri deksel teksture alata oblika sfere.



Slika 3.14. Generisane tri deksel texture instance alata (glodalo sa zaobljenim čelom) u z, y i x pravcu (odozgo na dole)

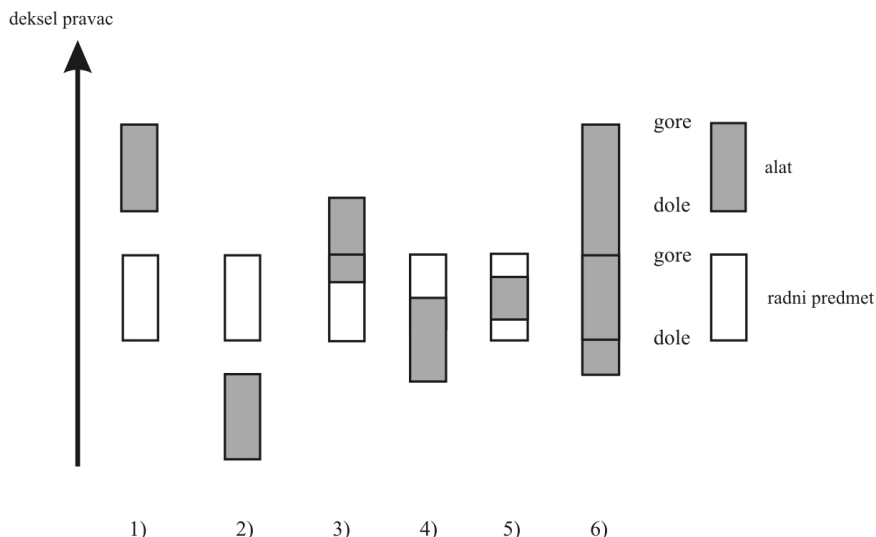


Slika 3.15. Generisane tri deksel texture instance alata (sfera) u z, y i x pravcu (odozgo na dole)

3.5 Oduzimanje alata od obradka

Oduzimanje deksel modela alata od obradka, svodi se na poređenje gornjih i donjih vrednosti dva odgovarajuća deksela alata i obradka (iste x-y koordinate generisanih tekstura). Šest slučajeva koji se mogu javiti pri oduzimanju respektivnih deksela prikazani su na slici

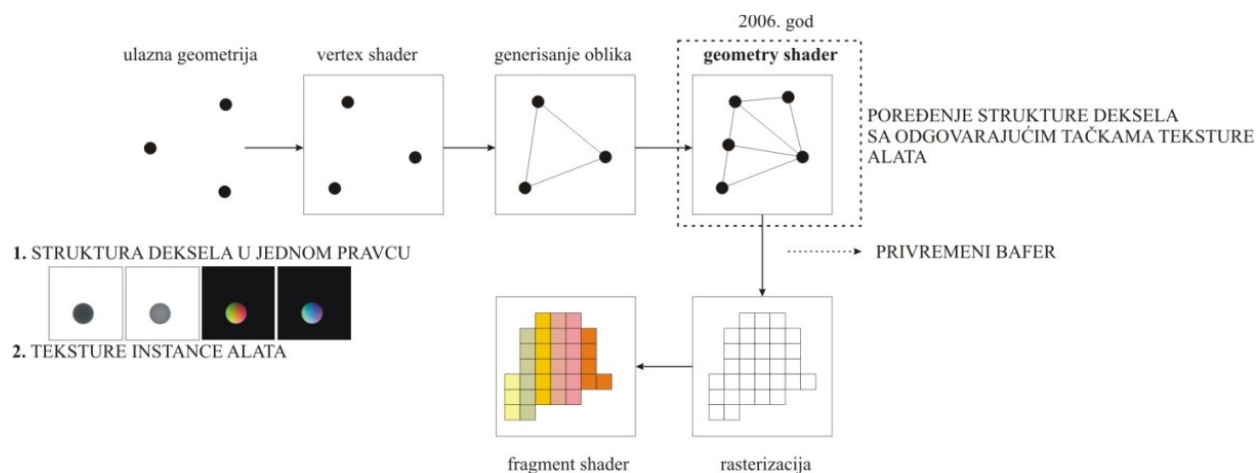
3.16. Deksel alata je iznad (1) ili ispod (2) preseka ne postoji. Alat skida gornji deo materijala sa obradka (3), gornji deksel predmeta se setuje kao donji deksel alata. Alat skida donji deo materijala sa obradka (4), donji deo deksela radnog predmeta se setuje kao gornji deo deksela alata. Ukoliko alat odstranjuje unutrašnji deo predmeta, generišu se dva deksela na radnom predmetu (5). Ukoliko alat skida deksel radnog predmeta kompletno, broj deksela radnog predmeta na datoj poziciji se setuje na nula (6).



Slika 3.16. Šest slučajeva za oduzimanje deksela alatka od obradka

Takođe, ukoliko postoji sečenje, normala površine koju predstavljaju podsečeni ili novi generisani dekseli (*normala1* ili *normala2* u strukturi deksela) setuje se kao inverzna normala deksela alata koji vrši podsecanje.

Struktura GPU jedinica do 2006. god., nije omogućavala da se broj grafičkih primitiva (tačke, linije, trouglovi) dinamički menja u samoj GPU. Bilo je moguće preko *vertex shader*-a, vršiti transformaciju i preko *fragment shader*-a, vršiti napredno senčenje prikaza (slika 3.17). Od 2006. god., sa pojavom OpenGL 3.2 standarda predstavljena je nova vrsta programabilnog shader-a, *geometry shader*. Ova novina omogućila je da se ne samo transformacija i napredno senčenje vrše u samoj GPU jedinici, već i da se generišu novi primitivi, bez potrebe da u tome učestvuje CPU jedinica.



Slika 3.17. Prikaz programabilnog dela GPU jedinice i princip oduzimanja instance alata od deksel strukture

Ukoliko se pogleda peti slučaj oduzimanja deksela alata od obradka, a to je slučaj gde se deksel priprema deli na dva nova deksela, jasno je da je zbog strukture GPU jedinice do 2006. god., simulacija obrade bazirana na deksel pristupu u GPU nije bila moguća.

Oduzimanje alata od obradka treba izvršiti u sva tri pravca. Generisane strukture deksela u sva tri pravca (prikazane u poglavlju 2.2), prebačene su u bafer GPU jedinice. Za svaki deksel pravac, strukture deksela prosleđuju se *geometry shader*-u, zajedno sa odgovarajućim teksturama instance alata (slika 3.17). U *geometry shader*-u se vrši poređenje sa slike 3.16 (za svaki deksel i odgovarajuće parametre *pozicijax* i *pozicijay* iz strukture deksela, poredi se tačka sa istom pozicijom iz teksture alata). Modifikovane strukture deksela se smeštaju u privremeni bafer u GPU jedinici, koji se zatim kopira u odgovarajući bafer sa strukturama deksela za odgovarajuću projekciju. Treba napomenuti da se ovaj korak koristi samo za oduzimanje, dok se prikaz vrši naknadno. Deo programskog koda sa komentarima za ovaj postupak prikazan je na slici 3.18.

```
glEnable(GL_RASTERIZER_DISCARD); //Isključiti prikaz

glBindBuffer(GL_ARRAY_BUFFER, BufferZ); // Aktivirati bafer sa deksel strukturom

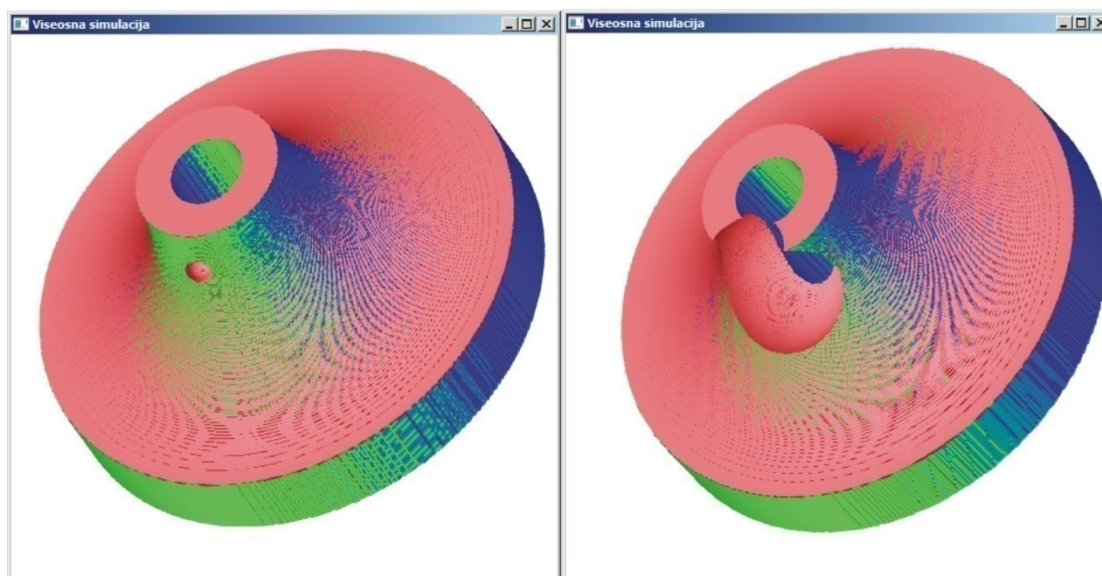
glEnableVertexAttribArray(0); // pozicijaX
glEnableVertexAttribArray(1); // pozicijaY
glEnableVertexAttribArray(2); // dubina
glEnableVertexAttribArray(3); // dužina
glEnableVertexAttribArray(4); // normala1
glEnableVertexAttribArray(5); // normala2
glEnableVertexAttribArray(6); // sece1
glEnableVertexAttribArray(7); // sece2
glVertexAttribPointer( 0,1, GL_INT, GL_FALSE, sizeof(podaci),(const GLbyte*)0 );
glVertexAttribPointer( 1,1, GL_INT, GL_FALSE, sizeof(podaci),(const GLbyte*)4 );
glVertexAttribPointer( 2,1, GL_FLOAT, GL_FALSE, sizeof(podaci),(const GLbyte*)8 );
glVertexAttribPointer( 3,1, GL_FLOAT, GL_FALSE, sizeof(podaci),(const GLbyte*)12 );
glVertexAttribPointer( 4,3, GL_FLOAT, GL_FALSE, sizeof(podaci),(const GLbyte*)16 );
glVertexAttribPointer( 5,3, GL_FLOAT, GL_FALSE, sizeof(podaci),(const GLbyte*)28 );
glVertexAttribPointer( 6,1, GL_INT, GL_FALSE, sizeof(podaci),(const GLbyte*)40 );
glVertexAttribPointer( 7,1, GL_INT, GL_FALSE, sizeof(podaci),(const GLbyte*)44 );

glBeginQuery(GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN, queryT);

glBindBufferBase(GL_TRANSFORM_FEEDBACK_BUFFER, 0, BufferP); //Postavi povratni bafer
glBeginTransformFeedback(GL_POINTS);
glDrawArrays(GL_POINTS, 0, PrimitivesWrittenZ); //Pozovi sve strukture deksela u Z pravcu
glEndTransformFeedback();
std::swap(BufferZ, BufferP); //Prebaci povratni bafer u početni
glDisable(GL_RASTERIZER_DISCARD); //Uključiti prikaz
glEndQuery(GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN);
glGetQueryObjectiv(queryT, GL_QUERY_RESULT, &PrimitivesWrittenZ); //Zapiši novi broj struktura deksela
```

Slika 3.18. Deo programskog koda za slanje deksel struktura u z pravcu GPU jedinici pri oduzimanju alata od obradka

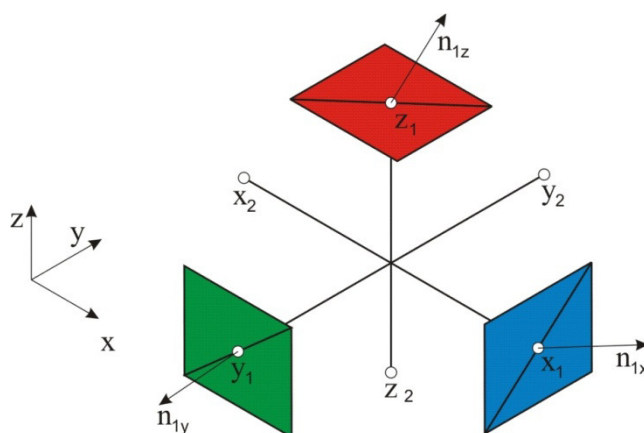
Na slici 3.19, prikazani su tri deksel modeli priprema, od kojeg su oduzeti alat sa zaobljenim čelom (slika 3.13) i sfera (slika 3.14).



Slika 3.19. Tri deksel model priprema od koga je oduzet alat sa zaobljenim čelom (levo), i alat u obliku sfere (desno)

3.6 Prikaz tri deksel modela obradka u procesu simulacije

Metod prikaza tri-deksel modela priprema, u programskom sistemu prikazan je na slici 3.20. Svaka početna i krajnja tačka deksela prikazana je kao dva trougaona poligona kojima je zbog razloga senčenja pridružena normala površine koja je definisana u strukturi deksela na slici 3.8.



Slika 3.20. Metod generisanja trougaonih poligona za prikaz deksel modela

Ovakav način prikaza tri-deksel modela, omogućava interaktivnu promenu pogleda na obradak u procesu simulacije.

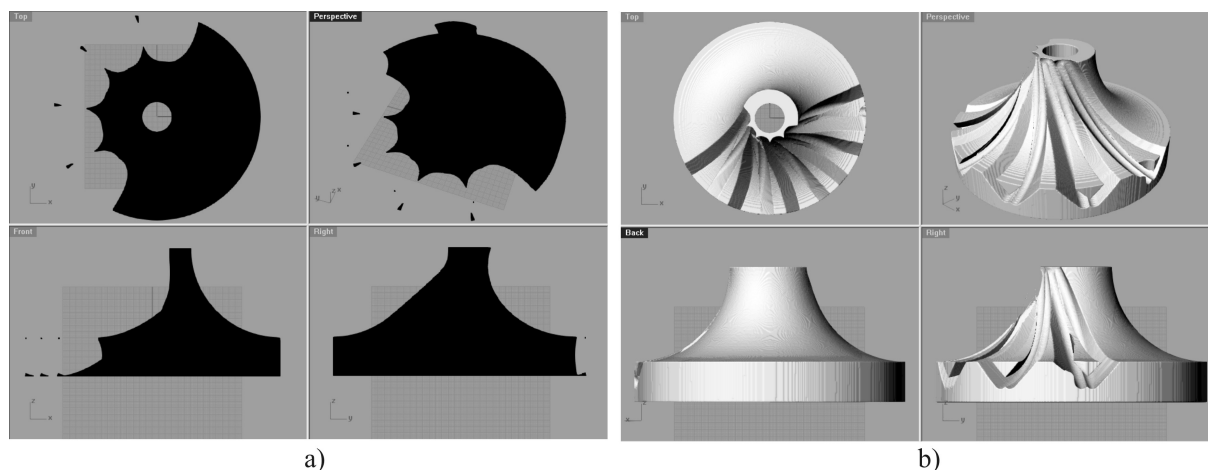
3.7 Eksport tri deksel struktura obradka

Generisane tri deksel strukture obradka, radi naknadnog procesiranja i eventualnog importovanja u neki od komercijalnih sistema, moguće je iz memorije GPU jedinice, prebaciti u glavnu memoriju koja je dostupna CPU jedinici. Sadržaj bafera sa deksel strukturama prebacuje se u glavnu memoriju sledećim kodom (primer za deksele u z pravcu):

```
glBindBuffer(GL_ARRAY_BUFFER, BufferZ);
bufferZ = (podaci*)glMapBuffer( GL_ARRAY_BUFFER, GL_READ_ONLY);
```

Pri čemu *podaci*, predstavlja strukturu podataka, prikazanu na slici 3.8. Kada se deksel strukture prebace iz memorije GPU jedinice moguće ih je naknadno obraditi i primeniti neki od algoritama za generisanje modela na osnovu tri-deksel opisa modela.

Razvijeno programsko rešenje omogućava eksport izradka, na dva načina. Prvi predstavlja eksportovanje modela u obliku oblaka tačaka (point cloud) u .xyz formatu, gde se svaka tačka deksela eksportuje sa svojim koordinatama i normalom. Na ovaj način je moguće generisani fajl importovati u neki od komercijalnih sistema i u njemu izvršiti triangulaciju modela. Primer modela u .xyz formatu koji je importovan u komercijalni sistem prikazan je na slici 3.21 a.



Slika 3.21. a) Importovani tri-deksel model u .xyz formatu, b) importovani tri-deksel model u STL formatu u komercijalni sistem

Drugi način za eksport modela omogućava eksport u STL formatu, koji je pogodan ukoliko se želi na 3D štampaču izraditi izradak koji je rezultat simulacije u razvijenom programskom sistemu. Generisanje STL modela vrši se Marching Cubes algoritmom [23], primenom VTK biblioteke, čiji je ovaj algoritam sastavni deo.

4.0 SOFTVERSKA IMPLEMENTACIJA PREDLOŽENOG REŠENJA

Predloženo programsko rešenje za simulaciju procesa višeosne obrade zasnovano na tri-deksel modelu, implementirano je primenom OpenGL biblioteke, koja predstavlja standard u oblasti interaktivne vizuelizacije. Za interakciju sa korisnikom primenjene su GLUT (OpenGLUtilityToolkit) [20] i GLUI (GLUT-Based User Interface Library) [21] biblioteke. GLUT biblioteka je zadužena za definisanje prozora za prikaz, interakciju sa korisnikom preko menija, tastature i miša. GLUI biblioteka sadrži u sebi kontrole kao što su ček boksovi, spinneri, radio dugmad, koji omogućavaju lakšu interakciju sa korisnikom (npr. kod promene pogleda na radni predmet, promene zuma, itd.). Za programiranje *shader*-a (vertex, geometry i fragment, slika 3.17), primenjen je GLSL (OpenGL Shading Language) [22]. Za određivanje matrica transformacija instanci alata u procesu simulacije, primenjena je VTK biblioteka [19].

Razvijeno programsko rešenje napisano je u C++ programskom jeziku primenom Visual Studio razvojnog okruženja.

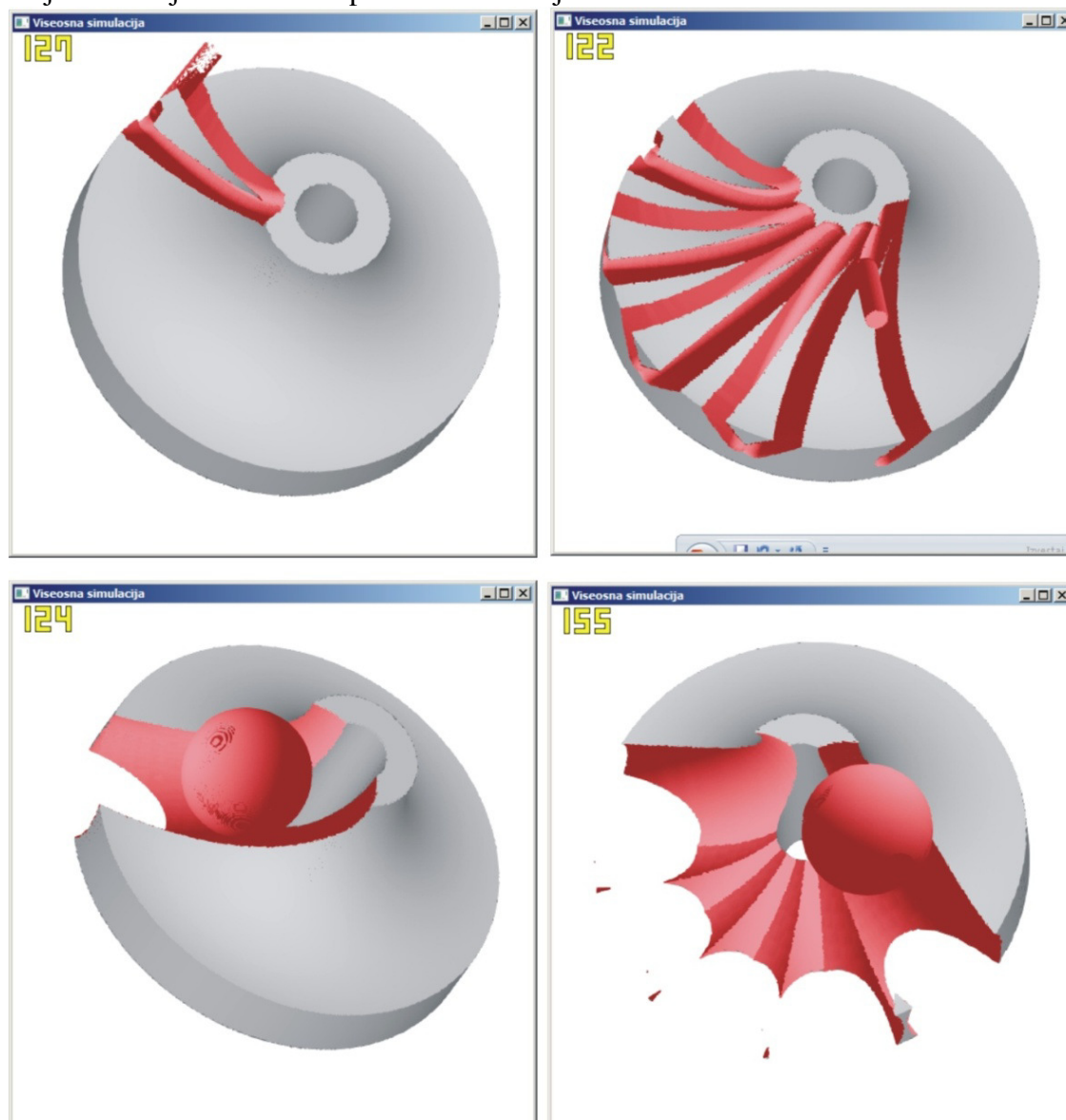
5.0 REZULTATI RAZVIJENOG PROGRAMSKOG REŠENJA

Razvijeno programsko rešenje testirano je na sledećoj konfiguraciji:

- CPU Intel i5 760, 2,8 GHz
- GPU NVidia GTX 660 (960 procesorskih jedinica)
- Memorija 4 GB

Pri rezoluciji od 512x512x512 (za broj deksela obradka reda 4×10^5), oduzimanje alata od obradka i iscrtavanje modela, razvijeni programski sistem je izvršavao prosečno za 8 ms, odnosno 125 fps (iscrtavanja po sekundi). Kod rezolucije 1024x1024x1024 (za broj deksela obradka reda $1,5 \times 10^6$), proračun i iscrtavanje su se izvršavali prosečno za 28 ms, odnosno 30 fps, što je takođe pogodno za interaktivni prikaz.

Prikaz zahvata alata u procesu obrade, sa glodalom sa zaobljenim čelom i sferom, dat je na slici 5.1, za rezoluciju prikaza 512x512x512. U gornjem levom uglu prozora prikazan je i broj iscrtavanja u sekundi u procesu simulacije.



Slika 5.1 Prikaz zahvata u procesu obrade u razvijenom programskom sistemu (glodalo sa zaobljenim čelom- gore, i sfera - dole)

6.0 ZAKLJUČAK

Predloženo tehničko rešenje predstavlja programski sistem za simulaciju višeosne obrade zasnovano na tri-deksel modelu. Jedna od osnovnih karakteristika predloženog rešenja je činjenica da se za proračun oduzimanja alata od obradka u procesu simulacije koristi GPU jedinica. Na ovaj način se vreme proračuna znatno smanjuje od metode gde proračun oduzimanja izvršava CPU jedinica. Zbog smanjenog vremena proračuna, ova metoda je primenljiva kako za simulaciju višeosne obrade tako i za primenu u sistemima gde se zahteva interaktivni prikaz i oduzimanje modela jednog od drugog. Razvijeni programski sistem omogućava simulaciju sa alatom proizvoljnog oblika, koristeći princip instanciranja alata, tako da primenu može naći u programskim sistemima za slobodnoručno modelovanje (gde se pozicija alata zadaje preko heptičkog uređaja). Takođe smanjeno vreme proračuna, stvara mogućnost da se sistem proširi modulom za interaktivni proračun sile u toku procesa uklanjanja materijala. Mogućnost eksportovanja izradka kao oblaka tačaka ili STL modela, omogućava da se generisani izradak importuje u neki od komercijalnih sistema radi naknadne analize ili da se izradi na 3D štampaču.

7.0 LITERATURA

- [1] Voelcker, H. B., Hunt, W. A.: The role of solid modeling in machining process modeling and NC verification, SAE Tech Paper 810195, 1981.
- [2] Spence, A. D., Altintas, Y. A.: Solid modeler based milling process simulation and planning system. Trans ASME 1994;116:61–9.
- [3] Milojević, Z., Navalusić, S., Zeljković, M.: An Exact Approach to 3-Axis Milling NC Simulation and Verification, Manufacturing Engineering, No 3, Vol. V, Prešov, 2006, pp.14-17.
- [4] Anderson, A. A.: Detecting and elimination collisions in NC machining, Computer-Aided Design, Vol. 10, No.4., 1978, pp: 231–237.
- [5] Jerard, R. B., Hussaini, S. Z., Drysdale, R. L., Schaudt, B.: Approximate methods for simulation and verification of numerically controlled machining programs, The Visual Computer 5, Springer-Verlag 1989, pp: 329-348.
- [6] Chappel, I. T.:The use of vectors to simulate material removed by numerically controlled milling, Computer Aided Design, Vol 15 No 3, 1983., pp: 156-158.
- [7] Atherton, P. R., Earl, C., Fred, C.: A graphical simulation system for dinamic five-axis NC verification, Proc. Autofact (Nov.), SME, Dearborn, Mi., 1987., pp:2.1-2.12.
- [8] Van Hook, T.: Real-time shaded NC milling display, Computer Graphics, SIGGRAPH Proceedings, Vol. 20, No 4, 1986, pp: 15-20
- [9] Huang, Y., Oliver, J.H.: NC milling error assessment and tool path correction, Computer Graphics Proceedings (SIGGRAPH '94), 1994, pp: 287-294.
- [10] Wang, W.P., Wang, K.K.: Geometric Modelling for Swept Volume of Moving Solids, IEEE Comput. Graph. Applic. Vol. 6, No 12., 1986., pp: 8-17.
- [11] Benouamer, M. O., Michelucci, D.: Bridging the Gap between CSG and Brep via a Triple Ray Representation, In Proc. Fourth ACM/Siggraph Symposium on Solid Modeling and Applications, 1997., pp:68-79.
- [12] Muller, H., Surmann, T., Stautner, M., Albersmann, F., Weinert, K.: Online Sculpting and Visualization of Multi-Dexel Volumes, Proceeding SM '03 Proceedings of the eighth ACM symposium on Solid modeling and applications, 2003., pp:258 - 261.

- [13] Tukora, B., Szalay, T.: Multi-dexel based material removal simulation and cutting force prediction with the use of general-purpose graphics processing units, *Advances in Engineering Software* 43, 2012., pp: 65–70.
- [14] Peng, X., Zhang, W.: A Virtual Sculpting System Based on Triple Dexel Models with Haptics, *Computer-Aided Design & Applications*, 6(5), 2009, 645-659.
- [15] NVidia: CUDA C Programming Guide - nVIDIA CUDA Toolkit Documentation, <http://docs.nvidia.com/cuda/cuda-c-programming-guide/>, pristupljeno, 13.12.2013.
- [16] Everitt, C.: Interactive order-independent transparency, White paper, nVIDIA 2.6, 2001.
- [17] Bavoil, L., Myers, K.: Order independent transparency with dual depth peeling, NVIDIA OpenGL SDK (2008).
- [18] Wang, C., Leung, Y.S., Chen, Y.: Solid modeling of polyhedral objects by Layered Depth-Normal Images on the GPU, *Computer-Aided Design*, vol.42, no.6, 2010., pp:535-544.
- [19] Schroeder, W., Martin, K., Lorensen, B.: *The visualization toolkit an object-oriented approach to 3d graphics*", 3rd edition, 2002, Pearson education inc.
- [20] <http://www.opengl.org/resources/libraries/glut/>, pristupljeno, 13.12.2013.
- [21] Rademacher, P.: GLUI: A GLUT-Based User Interface Library, version 2.0, 1999.
- [22] http://en.wikipedia.org/wiki/OpenGL_Shading_Language, pristupljeno, 13.12.2013.
- [23] Lorensen, W., Cline, J., Marching, C.: A High Resolution 3D Surface Construction Algorithm, *Journal of Computer Graphics*, Vol. 21, No. 4, 1987, pp: 163-169.

Napomena: Tehničko rešenje predstavlja rezultat istraživanja na projektu "Savremeni prilazi u razvoju specijalnih rešenja uležištenja u mašinstvu i medicinskoj protetici" TR 35025, finansiranog od strane Ministarstva prosvete i nauke Republike Srbije.



УНИВЕРЗИТЕТ
У НОВОМ САДУ



ФАКУЛТЕТ
ТЕХНИЧКИХ НАУКА

Трг Доситеја Обрадовића 6, 21000 Нови Сад, Република Србија
Деканат: 021 6350-413; 021 450-810; Централа: 021 485 2000
Рачуноводство: 021 458-220; Студентска служба: 021 6350-763
Телефакс: 021 458-133; e-mail: ftndean@uns.ac.rs

ИНТЕГРИСАНИ
СИСТЕМ
МЕНАџМЕНТА
СЕРТИФИКОВАН ОД:



Наш број: 01.сл

Ваш број:

Датум: 2013-12-27

ИЗВОД ИЗ ЗАПИСНИКА

Наставно-научног већа Факултета техничких наука у Новом Саду, на 18. редовној седници одржаној дана 26.12.2013. године, донело је следећу одлуку:

-непотребно изостављено-

Тачка 11.1. Верификација нових техничких решења и именовање рецензента

Тачка 11.1.13.: У циљу верификације новог техничког решења усвајају се рецензенти:

- Др Милош Главоњић, редовни професор, Машински факултет, Београд
- Др Ратко Обрадовић, редовни професор, ФТН, Нови Сад

Назив техничког решења:

ПРОГРАМСКО РЕШЕЊЕ ЗА СИМУЛАЦИЈУ ПРОЦЕСА ВИШЕОСНЕ ОБРАДЕ ЗАСНОВАНО НА ТРИ-ДЕКСЕЛ МОДЕЛУ

Аутори техничког решења: др Зоран Милојевић, ванредни професор; др Слободан Навалушић, редовни професор; др Милан Зељковић, редовни професор; др Слободан Табаковић, ванредни професор; др Марија Вићевић, доцент; др Александар Живковић, доцент.

-непотребно изостављено-

2. Записник водила:

Јасмина Димић, дипл. правник

Тачност података оверава:
Секретар

Иван Нешковић, дипл. правник



Проф. др Радослав Дорословачки

Одлуком Наставно научног већа Факултета техничких наука Универзитета у Новом Саду, од 26.12.2013. године, именовани смо за рецензенте Техничког решења под називом „ПРОГРАМСКО РЕШЕЊЕ ЗА СИМУЛАЦИЈУ ПРОЦЕСА ВИШЕОСНЕ ОБРАДЕ ЗАСНОВАНО НА ТРИ-ДЕКСЕЛ МОДЕЛУ“, реализованог у оквиру пројекта „САВРЕМЕНИ ПРИЛАЗИ У РАЗВОЈУ СПЕЦИЈАЛНИХ РЕШЕЊА УЛЕЖИШТЕЊА У МАШИНСТВУ И МЕДИЦИНСКОЈ ПРОТЕТИЦИ“, ев. број ТР-35025, чији су аутори: др Зоран Милојевић, ванредни професор, др Слободан Навалушић, редовни професор, др Милан Зељковић, редовни професор, др Слободан Табаковић, ванредни професор, др Марија Вићевић, доцент и др Александар Живковић, доцент. На основу прегледа Елабората овог техничког решења и увида у публиковане радове о њему подносимо следећи

ИЗВЕШТАЈ

Техничко решење „ПРОГРАМСКО РЕШЕЊЕ ЗА СИМУЛАЦИЈУ ПРОЦЕСА ВИШЕОСНЕ ОБРАДЕ ЗАСНОВАНО НА ТРИ-ДЕКСЕЛ МОДЕЛУ“, аутора: ванредног професора др Зорана Милојевића, редовног професора др Слободана Навалушића, редовног професора др Милана Зељковића, ванредног професора др Слободана Табаковића, доцента др Марије Вићевић и доцента др Александра Живковића је описано у Елаборату који садржи 19 страница формата А4 написаних фонтом 12 са нормалним проредом. Елаборат садржи седам поглавља у које спада и списак коришћене литературе са 23 наслова. Предложено техничко решење садржи 26 илустрација. Наслови поглавља су:

1. Област примене техничког решења
2. Технички проблем
3. Детаљан опис техничког решења
4. Софтверска имплементација предложеног решења
5. Резултати развијеног програмског решења
6. Закључак
7. Литература

Техничко решење припада области софтвера предвиђеног за симулацију процеса вишеосне обраде и реализовано је у оквиру пројекта „Савремени прилази у развоју специјалних решења улежиштења у машинству и медицинској протетици“ ев. број ТР-35025 из програма технолошког развоја Министарства просвете, науке и технолошког развоја Републике Србије. Са становишта применљивости, припада области производног машинства, док са становишта примењених принципа припада области компјутерске графике.

Програмско решење за симулацију вишеосне обраде користи се на Факултету техничких наука као средство за потребе истраживања и унапређење наставе.

У поглављу *Област примене техничког решења* се указује на могућности и подручје примене програмског решења за симулацију вишеосне обраде.

У поглављу *Технички проблем* описана је проблематика из области симулације NC управљачког програма. Указано је на њен значај и дат је приказ метода и поступака који се користе у процесу симулације, као и предности и мане истих. На крају овог поглавља, приказана је концепција предложеног програмског система.

Поглавље *Детаљан опис техничког решења* састоји се из седам подпоглавља у којима је приказан модел предложеног програмског решења за симулацију вишеосне обраде. На почетку је дат упоредни приказ процесорске снаге GPU и CPU

јединица, што представља један од разлога зашто развијени програмски систем користи GPU јединицу за прорачун одузимања алата од обрадка. У прва два подпоглавља, *генерисање три-дексел текстура и структура припремка*, описан је метод и дефинисани су параметри за генерисање "дубинских" текстура и нормала припремка. Дат је приказ параметара структуре која се у програмском систему користи за опис дексела и на конкретном примеру за припремак приказане су генерисане текстуре и дексели у сва три правца. У подпоглављу *генерисање матрица трансформација инстанци алата*, приказан је принцип одређивања инстанци алата између два положаја у процесу симулације. За два алата (глодало са заобљеним челом и сфера), приказане су генерисане текстуре за један положај. Шест случајева за одузимање дексела алата од обрадка приказано је у подпоглављу *одузимање алата од обрадка*. Такође, приказан је принцип одузимања алата од обрадка у GPU јединици, као и део програмског кода за процес одузимања. У подпоглављу *приказ три дексел модела обрадка у процесу симулације*, представљен је начин приказа обрадка у процесу симулације који омогућава интерактивни приказ у реалном времену. Подпоглавље *експорт три дексел структура обрадка*, приказује поступак пребацивања дексел структура из GPU у CPU јединицу, као и две варијанте експорта модела израдка као облака тачака у .xyz формату и у облику STL модела који је погодан за израду на 3Д штампачу.

У поглављу *Софтверска имплементација предложеног решења*, дат је преглед алата и библиотека које су коришћене у развоју програмског решења.

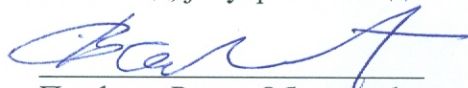
Поглавље *Резултати* садржи приказ симулације вишеосне обраде, као и остварену брзину симулације за различите резолуције модела припремка.

У *закључку* се, на основу представљених чињеница, сумирају могућности развијеног програмског решења, и даје се осврт на могућност примене развијеног решења у другим областима технике.

МИШЉЕЊЕ

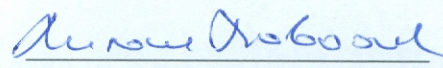
Аутори техничког решења „ПРОГРАМСКО РЕШЕЊЕ ЗА СИМУЛАЦИЈУ ПРОЦЕСА ВИШЕОСНЕ ОБРАДЕ ЗАСНОВАНО НА ТРИ-ДЕКСЕЛ МОДЕЛУ” су детаљно елаборатом описали методологију симулације процеса вишеосне обраде применом три дексел модела. Примењен је метод који за прорачун одузимања алата од обрадка користи GPU јединицу, што у великој мери скраћује време потребно за прорачун и омогућава интерактивни приказ симулације. Предложено техничко решење представља допринос у подручју интерактивне симулације процеса вишеосне обраде, а такође га је могуће применити и у другим областима где се захтева интерактивни приказ одузимања просторних модела једног од другог. Предлажемо Наставно научно већу Факултета техничких наука Универзитета у Новом Саду, да софтвер "ПРОГРАМСКО РЕШЕЊЕ ЗА СИМУЛАЦИЈУ ПРОЦЕСА ВИШЕОСНЕ ОБРАДЕ ЗАСНОВАНО НА ТРИ-ДЕКСЕЛ МОДЕЛУ" (М85) прихвати као ново техничко решење.

Нови Сад, јануар 2014. год.



Проф. др Ратко Обрадовић,
Факултет техничких наука,
Нови Сад

Београд, јануар 2014. год.



Проф. др Милош Главоњић
Машински факултет,
Београд



УНИВЕРЗИТЕТ
У НОВОМ САДУ



ФАКУЛТЕТ
ТЕХНИЧКИХ НАУКА

Трг Доситеја Обрадовића 6, 21000 Нови Сад, Република Србија
Деканат: 021 6350-413; 021 450-810; Централна: 021 485 2000
Рачуноводство: 021 458-220; Студентска служба: 021 6350-763
Телефакс: 021 458-133; e-mail: ftndecan@uns.ac.rs

ИНТЕГРИСАНИ
СИСТЕМ
МЕНАџМЕНТА
СЕРТИФИКОВАН ОД:



Наш број: 01.сл

Ваш број:

Датум: 2014-01-16

ИЗВОД ИЗ ЗАПИСНИКА

Наставно-научног већа Факултета техничких наука у Новом Саду, на 19. седници одржаној дана 15.01.2014. године, донело је следећу одлуку:

-непотребно изостављено-

Тачка 3. Верификација техничких решења

Тачка 3.9: На основу позитивног извештаја рецензената верификује се техничко решења под називом:

ПРОГРАМСКО РЕШЕЊЕ ЗА СИМУЛАЦИЈУ ПРОЦЕСА ВИШЕОСНЕ ОБРАДЕ ЗАСНОВАНО НА ТРИ-ДЕКСЕЛ МОДЕЛУ

Аутори техничког решења: др Зоран Милојевић, ванредни професор; др Слободан Навалушић, редовни професор; др Милан Зељковић, редовни професор; др Слободан Табаковић, ванредни професор; др Марија Вићевић, доцент; др Александар Живковић, доцент.

-непотребно изостављено-

Записник водила:

Јасмина Димић, дипл. правник

Тачност података оверава:
Секретар

Иван Нешковић, дипл. правник



Проф. др Радослав Ђорђевић, Декан