Detekcija udubljenja i rupa na 3D objektima

Stanić, Siniša

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:551869

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-26**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij Automobilsko računarstvo i komunikacije

DETEKCIJA UDUBLJENJA I RUPA NA 3D OBJEKTIMA

Diplomski rad

Siniša Stanić

Osijek, 2019.

Sadržaj

1. UVOD 1

1.1. Zadatak rada 1

2. DETEKCIJA SEDLASTIH POVRŠINA 2

3. PROGRAM ZA DETEKCIJU SEDLASTIH POVRŠINA NA DUBINSKIM SLIKAMA 13

4. EKSPERIMENTALNA EVALUACIJA 16

5. ZAKLJUČAK 30

LITERATURA 31

SAŽETAK 32

ABSTRACT 33

ŽIVOTOPIS 34

PRILOZI 35

1. UVOD

Puno predmeta u praksi sadrže rupe i udubljenja. Ladice, posude, police, kutije, vrata i prozori samo su neki od primjera. Sa stajališta robotskog vida, izrazito je korisno detektirati takve geometrijske strukture, jer često predstavljaju spremnik (posude, ladice, kutije) za koje je bitno odrediti granice. S druge strane mogu služiti kao orijentiri za lokalizaciju u prostorijama (prozori i vrata). Također, navedene geometrijske strukture se mogu koristiti kao značajke za prepoznavanje i klasifikaciju [1]. Rubovi rupa i udubljenja na površini različitih objekata često predstavljaju sedlaste površine, odnosno površine koje se sastoje od sedlastih točaka. Sedlasta točka je točka na površini čija derivacija svakog reda iznosi nula, ali nije lokalni ekstrem funkcije.

U ovom diplomskom radu opisana je metoda detekcije sedlastih površina na objektima prikazanim 3D mrežom trokuta (engl. *triangular mesh*), koja se temelji na poliedarskoj aproksimaciji. Izrađen je računalni program koji detektira sedlaste površine na scenama snimljenim 3D kamerom. Računalni program je ispitan na snimkama različitih predmeta, koji imaju rupe odnosno udubljenja.

1.1. Zadatak rada

U ovom diplomskom radu potrebno je napraviti program koji detektira rupe i udubljenja na scenama snimljenim RGB-D kamerom u programskom jeziku C++. Scena snimljena RGB-D kamerom je oblak 3D točaka, gdje svaka točka ima odgovarajuću boju.

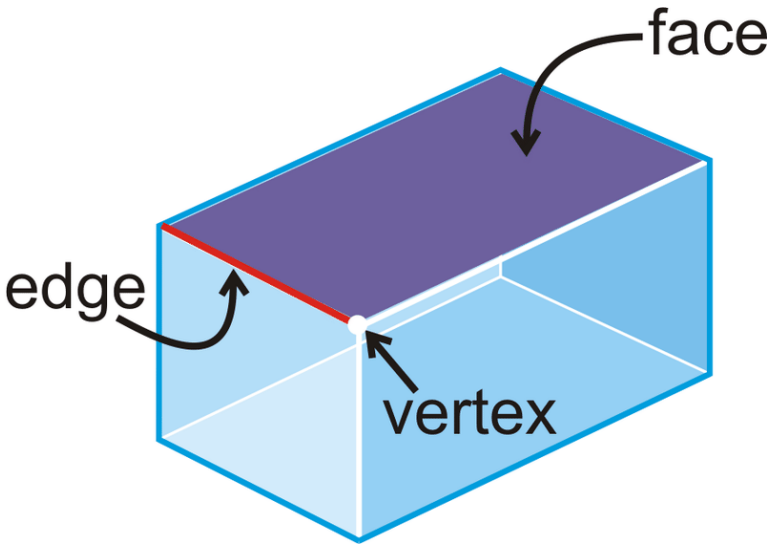
2. DETEKCIJA SEDLASTIH POVRŠINA

U ovom poglavlju objasnit će se detekcija sedlastih površina, no prije početka potrebno je objasniti često korištene pojmove bitne za razumijevanje ovog rada. Prije svega, važno je razlikovati poligon od poliedra. Poligon, ili mnogokut, predstavlja dio ravnine omeđen s konačno mnogo različitih dužina tako da u točki u kojoj završava jedna dužina, počinje druga, osim zadnje dužine, koja je spojena s početnom točkom početne dužine. S druge strane, poliedar je geometrijsko tijelo, odnosno dio prostora, omeđen poligonima tj. mnogokutima [2].

Za ovaj rad, bitni elementi poligona su:

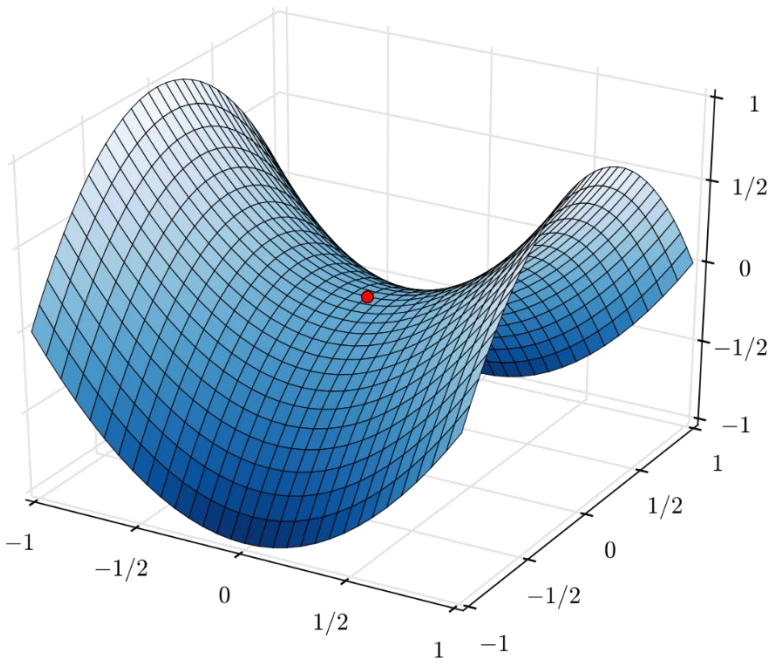
- vrh (engl. *vertex*, množina: *vertices*) predstavlja točku u kojoj se sijeku tri ravnine.
- brid (engl. *edge*) nekog poligona predstavlja dužinu koja spaja dva vrha te ujedno omeđuje poligon. Kod poliedra, brid predstavlja dio pravca gdje se dvije strane sastaju
- strana (engl. *face*) je dio ravnine omeđen s *n* bridova koji ujedno čine poligon (mnogokut)

Na slici 2.1. nalazi se prikaz poliedra, konkretno kvadra, kako bi se predočilo što gore navedeni elementi predstavljaju.



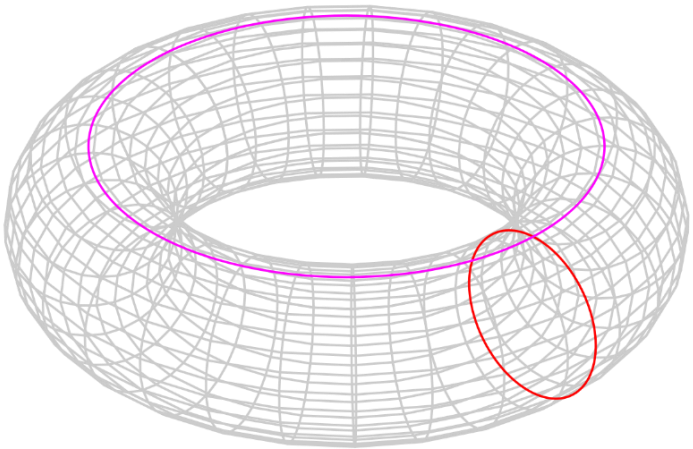
Slika 2.1. Poliedar (kvadar) s istaknutim elementima poliedra

Sedlasta točka je točka na površini čija derivacija svakog reda iznosi nula, ali nije lokalni ekstrem funkcije [2]. Sedlasta površina je površina koja sadrži jednu ili više sedlastih točaka i prikazana je na slici 2.2.



Slika 2.2. Sedlasta točka na sedlastoj površini

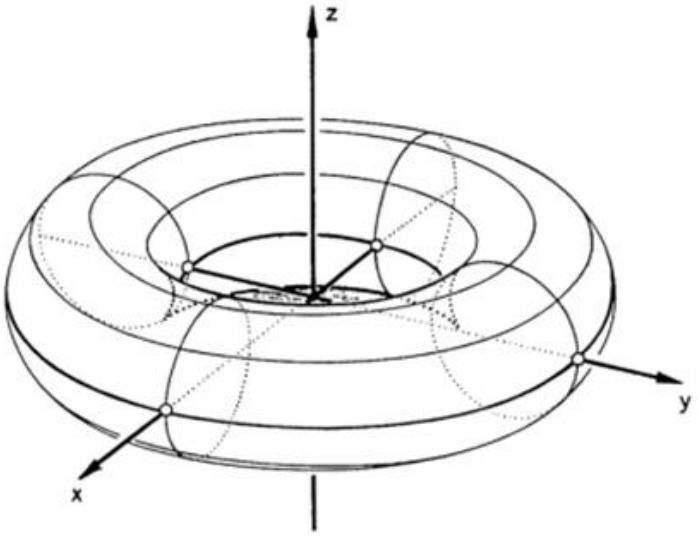
U geometriji, toroid je rotacijska ploha dobivena rotacijom krivulje u trodimenzionalnom sustavu oko osi koja je koplanarna s krivuljom. Dvije ili više točaka su koplanarne ako postoji ravnina unutar koje su sadržane. Torus je podskup toroida, dobiven rotacijom kružnice oko osi. Na slici 2.2 nalazi se prikaz torusa dobivenog rotacijom crvene kružnice [3]. Točke označene ljubičastom bojom su sedlaste, pa su time i dijelovi površine torusa sedlaste površine.



Slika 2.3. Torus [3]

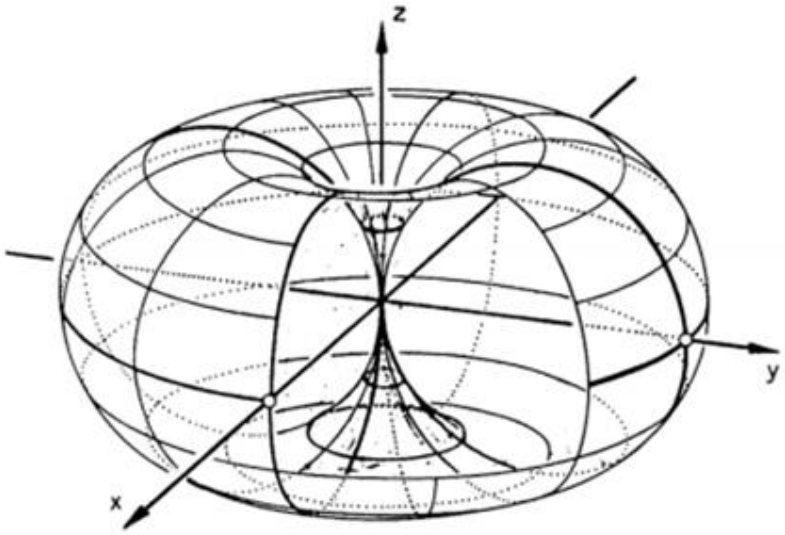
Ovisno o položaju osi u odnosu na kružnicu, razlikujemo tri torusa:

1. Os rotacije i kružnica nemaju niti jednu zajedničku točku, odnosno ne sijeku se. Torus dobiven takvom rotacijom na engleskom se naziva *torus of revolution* [3] i prikazan je na slici 2.4.



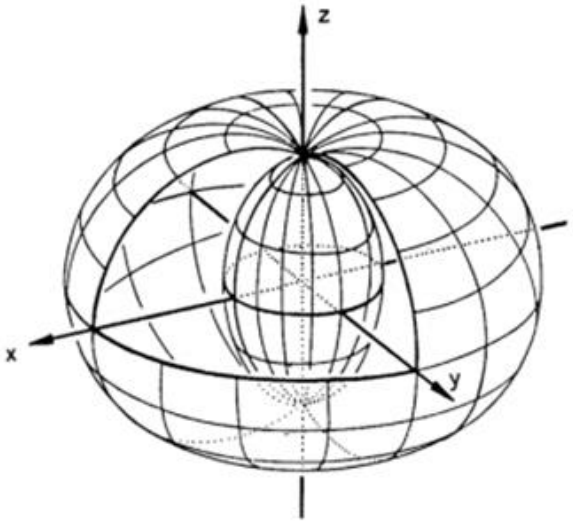
Slika 2.4. Torus of revolution [3]

2. Os rotacije je tangenta kružnice, odnosno os rotacije siječe kružnicu samo u jednoj točki. Torus dobiven takvom rotacijom na engleskom se naziva *horn torus* [3] i prikazan je na slici 2.5.



Slika 2.5. Horn torus [3]

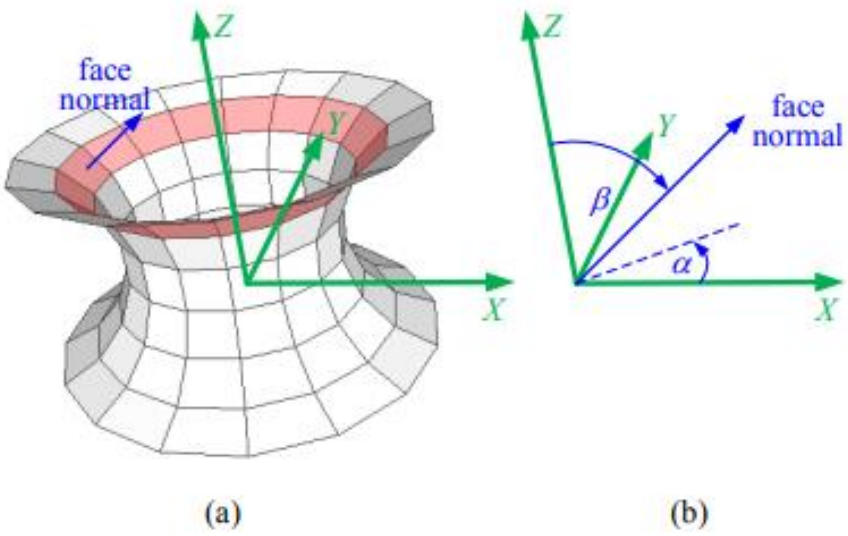
3. Os rotacije prolazi kroz kružnicu, odnosno siječe kružnicu u dvije točke Torus dobiven takvom rotacijom na engleskom se naziva *spindle torus* [3] i prikazan je na slici 2.6.



Slika 2.6. Spindle torus [3]

Sedlastom površinom u ovom radu će se modelirati objekti s rupama/udubljenjima ili savijeni objekti. Neki od primjera su: krofna, zdjela, šalica, banana...

Na slici 2.7. a) nalazi se model sedlaste površine koja je dio torusa. Algoritam za detekciju sedlastih površina predstavljen u ovom radu se temelji na poliedarskoj aproksimaciji površine. Sedlasta površina, a samim time i torus, se modelira kao poliedar koji se sastoji od više strana (engl. *face*) što je prikazano na slici 2.7 a).



Slika 2.7. Sedlasta površina i pripadajući joj koordinatni sustav [4]

Sedlasta površina je definirana u odnosu na koordinatni sustav prikazan zeleno (slika 2.7. b), gdje kut α predstavlja orijentaciju ortogonalne projekcije normale strane na ravninu XY, dok je β kut između normale strane i z-osi [4]. Strane se grupiraju u konuse, gdje strane unutar jednog konusa zatvaraju isti kut β. Jedan takav konus prikazan je na slici 2.7. a) crvenom bojom.

Svaki od *i* konusa definiran je svojim kutem β, gdje i ima vrijednosti $1, \dots, n$. Kutevi β u ovom radu su definirani izrazom (2-1).

$$\beta_i = \frac{(i-4)\,\pi}{8}, \qquad i = 1, \dots, 7 \tag{2-1}$$

Analogno tome, kutevi α su definirani izrazom (2-2).

$$\alpha_j = \frac{(j-1)\pi}{8}, \qquad j = 1, \dots, 16 \tag{2-2}$$

Neka je F_{ij} *j*-ta strana *i*-tog konusa modela sedlaste površine. Normale strane konusa su definirane izrazom (2-3).

$$n_{ij} = [\cos\alpha_j \sin\beta_i \quad \sin\alpha_j \cos\beta_i \quad \cos\beta_i]^T \tag{2-3}$$

Ravnina koja sadržava stranu F_{ij} definirana je jednadžbom (2-4).

$$n_{ij}^T p - d_{ij} = 0 \tag{2-4}$$

gdje je p vektor koordinata točke na promatranoj ravnini u odnosnu na koordinatni sustav sedlaste površine, a d_{ij} je udaljenost ravnine od koordinatnog sustava sedlaste površine. Svaka ravnina definira poluprostor. U ovom radu će biti korištena sljedeća konvencija:

Ako vrijedi jednadžba (2-5)

$$n_{ij}^T p - d_{ij} \leq 0 \tag{2-5}$$

onda točka *p* pripada poluprostoru koji je definiran ravninom (nij, dij).

Svaki *i*-ti konus modela sedlaste površine može se smatrati kao skup točaka koji je dobiven kao unija poluprostora definiranog ravninama (nij, dij), gdje je $j = 1, \dots, n_\alpha$.

Stoga, točka p pripada i-tom konusu ako je barem jedna od jednadžbi (2-5) zadovoljena, gdje je $j = 1, \ldots, n_\alpha$, odnosno:

$$\min_{1 \le j \le n_\alpha} \left(n_{ij}^T p - d_{ij} \right) \le 0 \qquad (2\text{-}6)$$

Cijela sedlasta površina predstavlja presjek setova točaka definiranih sa (2-6) za $i = 1, \ldots, n_\beta$.

Stoga, točka p pripada sedlastoj površini ako su zadovoljene sve jednadžbe sadržane u (2-6) gdje je $i = 1, \ldots, n_\beta$. Odnosno:

$$\max_{1 \le i \le n_\beta} \left(\min_{1 \le j \le n_\alpha} \left(n_{ij}^T p - d_{ij} \right) \right) \le 0 \qquad (2\text{-}7)$$

Točka p leži na sedlastoj površini ako vrijedi jednadžba (2-8).

$$\max_{1 \le i \le n_\beta} \left(\min_{1 \le j \le n_\alpha} \left(n_{ij}^T p - d_{ij} \right) \right) = 0 \qquad (2\text{-}8)$$

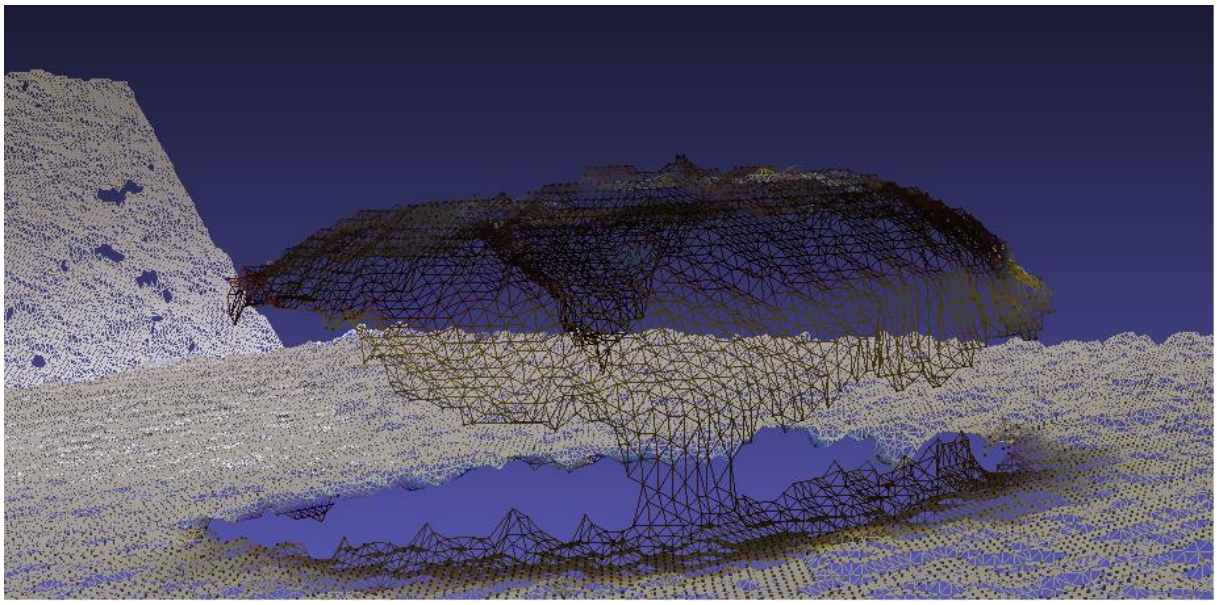
Skup točaka koji pripada sedlastoj površini može se definirati izrazom (2-9).

$$V_T = \left\{ p \in \mathbb{R}^3 \,\middle|\, \max_{1 \le i \le n_\beta} \left(\min_{1 \le j \le n_\alpha} \left(n_{ij}^T p - d_{ij} \right) \right) \le 0 \right\} \qquad (2\text{-}9)$$

Razmatramo slučaj kada se sedlasta površina V_T koristi za modeliranje rupe u objektu. Neka O bude objekt koji nema rupu, a O' isti taj objekt, ali s rupom modeliranom pomoću sedlaste površine. Tada vrijedi:

$$O' = O \cap V_T \qquad (2\text{-}10)$$

Algoritam detektira sedlaste površine tako da detektira konuse na predanoj mu mreži M i koordinatnom sustavu, te detektirane konuse grupira u sedlaste površine. Predana mreža M vizualno predstavlja scenu, a opisana je nizom vrhova, koji svojim povezivanjem čine bridove koji omeđuju strane. Primjer dijela mreže scene može se vidjeti na slici 2.8.



Slika 2.8. Prikaz dijela mreže

Mreža se segmentira na ravne plohe pomoću metode opisane u [4]. Kao rezultat, mreža M je prikazana pomoću skupa ravnih ploha, s odgovarajućim vrhovima i bridovima. Slijedi detekcija konusa tako što se uzimaju u obzir sve tangencijalne ravnine svih bridova koji su potencijalne strane konusa.

Beskonačno mnogo tangencijalnih ravnina leži na bridu koji se nalazi između dvije strane. Neka je onda $T(E)$ skup svih tangencijalnih ravnina brida E. Za specifičan β_i, konusi se detektiraju pretraživanjem povezanih skupova $\mathfrak{E}_C$ koji sadrže bridove iz mreže sa svojstvom da za svaki brid $E \in \mathfrak{E}_C$ postoji tangencijalna ravnina $(n_{ij}, d_{ij}) \in T(E)$ za koju vrijedi jednadžba (2-11)

$$\angle(n_{E,i}, z) = \beta_i \qquad (2\text{-}11)$$

gdje $\angle$ označava kut između dva vektora, a Z je jedinični vektor koji predstavlja z-os koordinatnog sustava sedlaste površine. Skup bridova ε_C je *povezan*, ako bilo koja dva brida $E, E' \in \mathfrak{E}_C$ postoje u nizu $E_1, E_2, \ldots E_n$ tako da $E_1 = E, E_n = E'$ te da dva uzastopna brida E_i i E_{i+1} imaju zajedničku krajnju točku. Nadalje, pretpostavlja se da svaki brid $E \in \mathfrak{E}_C$ definira jednu stranu konusa.

Stoga, $E \in \mathcal{E}_C$ ne bi trebao biti sadržan unutar poluprostora definiranog ravninom ($n_{E',i}$, $d_{E',i}$) drugog brida $E' \in \mathcal{E}_C$, odnosno svi parovi bridova $E, E' \in \mathcal{E}_C$ trebaju zadovoljiti uvjet (2-12)

$$n_{E',i}^T p_{E,k} - d_{E',i} > 0, \qquad k = 1,2 \tag{2-12}$$

gdje su $p_{E,1}$ i $p_{E,2}$ krajnje točke brida E.

Kako bi algoritam bio robustan, umjesto jednadžbe (2-12), koristi se uvjet (2-13)

$$n_{E',i}^T p_{E,k} - d_{E',i} > -\varepsilon, \qquad k = 1,2 \tag{2-13}$$

gdje je ε eksperimentalno određena tolerancija.

Iz skupa bridova $\mathcal{E}_c$ sa svojstvima opisanim ranije, kreira se konus. Za svaki $j = 1, \ldots, n_\alpha$ računa se d_{ij} izrazom (2-14).

$$d_{ij} = \min_{\substack{E \in \mathcal{E} \\ k=1,2}} \left(n_{ij}^T p_{E,k} \right) \tag{2-14}$$

Dobiveni parametri d_{ij} s normalama n_{ij} definiraju kompletan konus. Svaki d_{ij} je definiran s normalnom n_{ij} i jednom krajnjom točkom brida $E \in \mathcal{E}_c$, za koji je skalarni produkt $n_{ij}^T p_{E,k}$ minimalan. Krajnja točka koja definira d_{ij} će se u ovom radu odnositi na *referentnu točku* ravnine $\left(n_{ij}, d_{ij}\right)$ a označavati sa p_{ij}.

Detektirani konusi se tada povezuju u sedlaste površine. Sedlasta površina se kreira iz niza konusa, gdje svaki od konusa odgovara jednom kutu $\beta_i, i = 1, \ldots, n_\beta$. Dva konusa pripadaju istoj sedlastoj površini ako sve *referentne točke* tih konusa leže na sedlastoj površini koja se sastoji od ta dva konusa. To znači da sve *referentne točke* dva konusa gdje je kut prvog konusa β_i a drugog konusa β_{i+1}, moraju zadovoljavati jednadžbu (2-15).

$$\max_{i,i+1} \left(\min_{1 \le j \le n_\alpha} \left(n_{ij}^T p - d_{ij} \right) \right) = 0 \tag{2-15}$$

Jednadžba (2-15) je poseban slučaj jednadžbe (2-8). Budući da se d_{ij} računa pomoću jednadžbe (2-14) sve *referentne točke* i-tog konusa moraju zadovoljiti jednadžbu (2-16).

$$\min_{1 \le j \le n_\alpha} \left(n_{ij}^T p - d_{ij} \right) = 0 \tag{2-16}$$

Osim toga, ako je p_{i+1} *referentna točka* ravnine $\left(n_{i+1,j}, d_{i+1,j}\right)$ i ako vrijedi jednadžba (2-17)

$$n_{ij}^T p_{i+1,j} - d_{ij} \leq 0 \qquad (2\text{-}17)$$

tada vrijedi

$$\min_{1 \leq j \leq n_\alpha} \left(n_{ij}^T p - d_{ij}\right) \leq 0 \qquad (2\text{-}18)$$

Stoga, ako jednadžba (2-18) vrijedi za sve *referentne točke* dva uzastopna konusa, onda jednadžba (2-16) vrijedi za sve *referentne točke* ta dva konusa.

Kako algoritam ne bi bio osjetljiv na mjerni šum, umjesto jednadžbe (2-17), koristi se jednadžba (2-19)

$$n_{ij}^T p_{i+1,j} - d_{ij} \leq \varepsilon \qquad (2\text{-}19)$$

Konačno, sedlasta površina je kreirana od niza konusa, gdje svaka dva uzastopna konusa zadovoljavaju ranije opisane kriterije.

Algoritam 1 koristi se za detekciju sedlastih površina, a preuzet je iz [4] koji ujedno predstavlja implementaciju metoda opisanih u prethodnom poglavlju.

Algoritam 1 [4]

Input: $\mathcal{V}, \mathcal{E}, \alpha_j, j = 1, \ldots, n_\alpha, \beta_i, i = 1, \ldots, n_\beta, \varepsilon, n_{mincone}$
Output: $\mathcal{D}$

Compute $\mathcal{T}(E)$ for all edges $E \in \mathcal{E}$.
Create empty graph $\mathcal{G}$.
$\mathcal{D} \leftarrow \emptyset$
For $i = 1$ **to** n_β
  $\mathcal{C}_i \leftarrow \emptyset$
  Determine the set $\mathcal{E}_i$ of edges $E \in \mathcal{E}$ for which a tangential plane $(n_{E,i}, d_{E,i}) \in \mathcal{T}(E)$ exists.
  $\mathcal{E}_{assigned} \leftarrow \emptyset$
  For every edge $E \in \mathcal{E}_i$
    If $E \notin \mathcal{E}_{assigned}$ **then**
      $\mathcal{E}_{connected} \leftarrow$ the connected set of edges $E' \in \mathcal{E}_i \setminus \mathcal{E}_{assigned}$ containing E
      $\mathcal{E}_{assigned} \leftarrow \mathcal{E}_{assigned} \cup \mathcal{E}_{connected}$
      If $|\mathcal{E}_{connected}| \geq n_{mincone}$ **then**
        Determine $\mathcal{E}_c \subseteq \mathcal{E}_{connected}$ containing edges which satisfy (10) by greedy search.
        If $|\mathcal{E}_c| \geq n_{mincone}$ **then**
          Compute d_{ij} for every $j = 1, \ldots, n_\alpha$ by applying (11) to all edges $E \in \mathcal{E}_c$.
          Create vector d whose components are values d_{ij}.
          Create matrix P, where the jth column of P is the coordinate vector p_{ij} of the reference point of the plane (n_{ij}, d_{ij}).
          Add $C = (d, P)$ to $\mathcal{C}_i$.
          Insert C as a node into $\mathcal{G}$.
          If $i > 1$ **then**
            For every $C' \in \mathcal{C}_{i-1}$
              If all reference points of C and C' satisfy (14), **then** Connect nodes C and C' by a directed edge, where C is the source node and C' is the target node.
            end for
          end if
        end if
      end if
    end if
  end for
end for
Detect all sequences of connected nodes in graph $\mathcal{G}$ starting with a node, which is not a target node of any directed edge in $\mathcal{G}$, and terminating in a node, which is not a source node of any directed edge in $\mathcal{G}$.
For every node sequence $C_i = (d_i, P_i), i = i_{min}, \ldots, i_{max}$, detected in line 30, where $C_i \in \mathcal{C}_i$.
  Create a torus from the node sequence. A torus is represented by the index i_{min} and vector $d = \left[d_{i_{min}}^T, \ldots, d_{i_{max}}^T \right]^T$.
  Insert pair (i_{min}, d) into $\mathcal{D}$.
end for
return

Ulaz u algoritam je skup vrhova V, skup bridova $\mathcal{E}$ dobiven pomoću segmentacije mreže u planarne patcheve na način opisan u [4], dok korisnik određuje kuteve $\alpha_j, j = 1, \ldots, n_\alpha$, i $\beta_i, i = 1, \ldots, n_\beta$, toleranciju ε, te minimalan broj bridova za kreiranje konusa $n_{mincone}$.

Algoritam vraća skup sedlastih površina $\mathcal{D}$ koji se sastoji od niza parova (i_{min}, d) gdje je i_{min} indeks kuta β_i prvog konusa, a d vektor sastavljen od parametara d_{ij} svih strana sedlaste površine. Vektor d predstavlja povezivanje vektora d_i u lanac čime definira strane pojedinih konusa.

Sedlasta površina je definirana jednadžbom (2-20)

$$\max_{i_{min} \le i \le 1_{max}} \left(\min_{1 \le j \le n_\alpha} \left(n_{ij}^T p - d_{ij} \right) \right) = 0 \qquad (2\text{-}20)$$

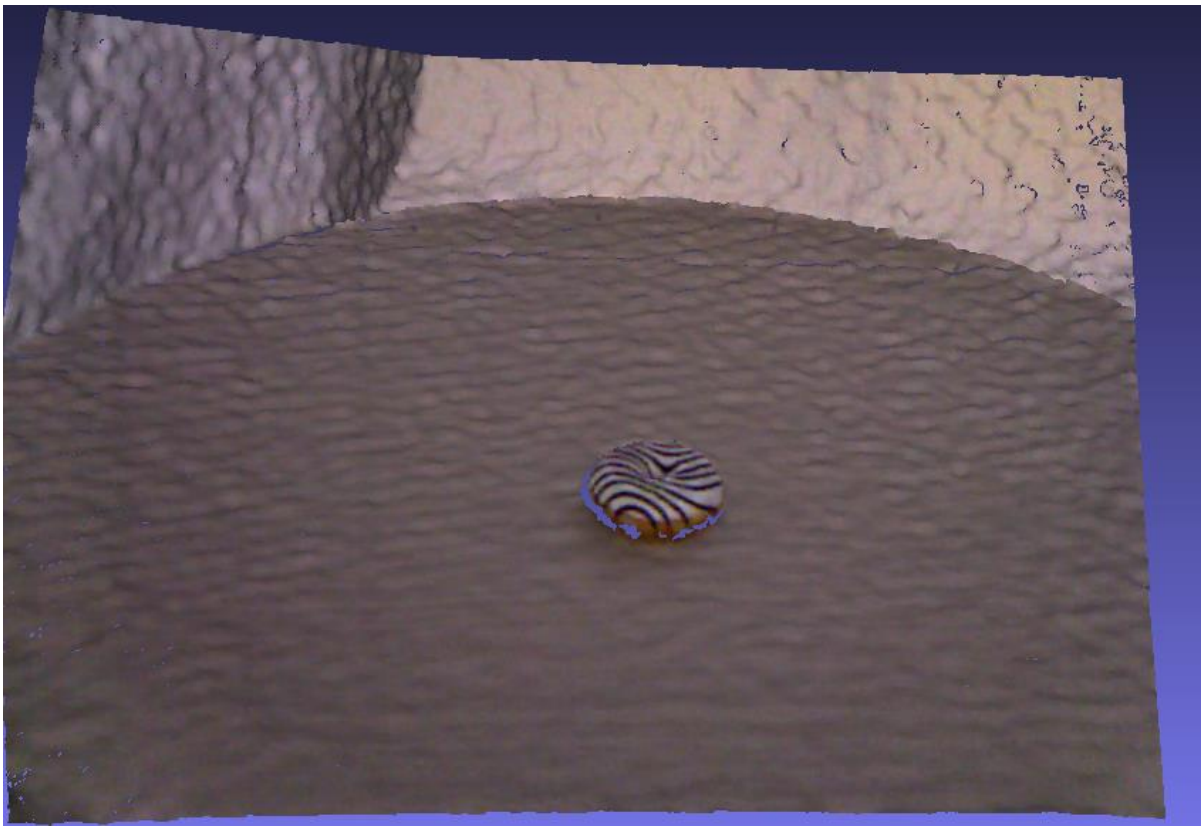
Implementacija „Greedy search"-a opisanog u liniji 13 Algoritma 1 opisana je u nastavku. Prvo, kompatibilnost svakog para bridova (E', E'') je određena pomoću jednadžbe (2-13). Svakom bridu $E' \in \mathcal{E}_{connected}$ je dodijeljena težina koja predstavlja sumu duljina svih bridova koji nisu kompatibilni s E' bez duljine E'. Nakon toga, kreirana je lista bridova, gdje su bridovi poredani uzlazno, ovisno o njihovoj težini. Na kraju, bridovi se dodaju u $\mathcal{E}_c$, jedan po jedan, počevši od onog s najmanjom težinom, dok se bridovi koji nisu kompatibilni sa svim bridovima koji se već nalaze u $\mathcal{E}_c$, ne dodaju.Tako se u $\mathcal{E}_c$ nalaze samo bridovi koji su međusobno kompatibilni prema jednadžbi (2-13). „Greedy search" teži maksimiziranju sume duljine svih bridova u $\mathcal{E}_c$.

3. PROGRAM ZA DETEKCIJU SEDLASTIH POVRŠINA NA DUBINSKIM SLIKAMA

Program za detekciju sedlastih površina realiziran je u okviru programske biblioteke *Robot Vision Library (RVL)* razvijene na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija (FERIT) u Osijeku.

Izrađena je aplikacija *RVLTorusDetectionDemo* koja detektira sedlaste površine na RGB-D slikama. Program koristi samo informaciju o dubini, tj. dubinsku sliku, dok se informacija o boji ne koristi.

Na slici 3.1. prikazana je scena koja se učitava na početku izvođenja programa.



Slika 3.1. Prikaz scene

Mreža učitane scene se prvo segmentira na ravne plohe. Rezultat te segmentacije je spremljen u objekt klase *SurfelGraph*. Točke u kojima se sastaju tri susjedne ravne plohe se u nastavku ovog rada nazivaju vrhovi. Koordinate vrhova su spremljene u polje *vertexArray* klase *SurfelGraph*. Konture koje predstavljaju granice dvije susjedne ravne plohe i povezuju dva vrha, nazivaju se bridovi. Bridovi su spremljeni u polje *edgeArray* klase *SurfelGraph* tako da su im zapisani indeksi krajnjih vrhova.

Ravne plohe se grupiraju u konveksne i konkavne površine pomoću algoritma opisanog u [4]. Te ćemo površine nazvati *segmenti*. Zatim se susjedni segmenti grupiraju u objekte. Od grupe koja se sastoji od najviše dva susjedna segmenta formira se objekt ako najmanja kugla koja sadrži oba segmenta ima polumjer manji ili jednak nekoj zadanoj vrijednosti. Nakon podjele scene u objekte, iteracijom kroz objekte pozivat će se funkcija *ToroidalClusters* koja će tražiti sedlaste površine na svakom od objekata.

Slijedi formiranje konveksne ljuske skupa točaka koji se nalazi u trenutnom objektu. Svrha formiranja konveksnih ljuski je određivanje podskupa točaka scene koje pripadaju samo jednom objektu. Kako algoritam i dalje mora primiti cijelu scenu, odnosno sve vrhove scene, a treba tražiti sedlaste površine samo u određenom području, točnije objektu, kreira se polje čiji broj elemenata odgovara broju vrhova, a popunjava se s *true* ili *false*, ovisno o tome nalazi li se točka scene s istim indeksom unutar ranije kreirane konveksne ljuske. To polje je nazvano *VertexCH*.

Nadalje, Algoritam 1 vrijedi za slučaj kada sedlasta površina leži na ravnini XY, a z-os je normala sedlaste površine oko koje se detektiraju konusi, pa je potrebno definirati koordinatni sustav sedlaste površine. To je ostvareno tako da se za z-os koordinatnog sustava sedlaste površine izabere normala one strane konveksne ljuske objekta, koja ima najveću površinu. Odabrana normala se predaje funkciji kao parametar *Axis_T*. Za ostale osi koordinatnog sustava mogu se uzeti bilo koja dva vektora okomita na z-os. Iz tako odabranih osi koordinatnog sustava formira se rotacijska matrica za prijelaz iz koordinatnog sustava scene u koordinatni sustav sedlaste površine, te se transformiraju svi vrhovi i normale koordinatnog sustava scene u koordinatni sustav sedlaste površine.

Funkcija koja se koristi za detekciju sedlastih površina nazvana je *ToroidalClusters*. Funkcija će na predanoj joj mreži scene pronaći sve objekte koji zadovoljavaju uvjete opisane u Algoritmu 1 te ih spremiti u predan joj povezani popis *TorusList*. Funkcija *ToroidalClusters* traži sedlaste površine samo u bridovima čije krajnje točke imaju vrijednost *true* u polju *VertexCH*, koje joj se predaje kao jedan od argumenata. Time se pretraživanje ograničava samo na jedan objekt. U *TorusList* će se nalaziti sve sedlaste površine detektirane na nekom objektu, a popis će se resetirati prilikom prelaska na novu scenu. *TorustList* je povezani popis koji se sastoji od strukture *Torus* i pokazivača na sljedeći element liste. Struktura *Torus* sadrži sve podatke o jednoj sedlastoj površini, tako da u sebi sadrži niz *ringArray*. Niz *ringArray* predstavlja niz konusa jedne sedlaste površine, a njegov tip podatka je struktura *TorusRing*. *TorusRing* odgovara jednom konusu čije stranice zatvaraju kut β sa z-osi sedlaste površine. Osim kuta β, struktura sadržava i niz

iVertexArray u kojem se nalazi popis indeksa vrhova konusa i niz *iEdgeArray* i kojem se nalazi popis indeksa bridova konusa. Vrijednosti sadržane u polju *iVertexArray* predstavljaju indekse vrhova koji se nalaze u ranije opisanom nizu *vertexArray* klase *SurfelGraph*. Pojednostavljeno, neka scena ima 10 vrhova koji se nalaze u polju *vertexArray*. Neka vrhovi 1, 2, 6 i 7 predstavljaju vrhove konusa. U polje *iVertexArray* će se spremiti samo njihovi indeksi, u ovom primjeru 1, 2, 6 i 7, umjesto da se ponovno spremaju njihove koordinate. Analogno vrijedi i za bridove. Unutar polja *iEdgeArray* sadržani su indeksi bridova, a svaki brid je opisan indeksima njegovih krajnjih vrhova. Vrijednosti sadržane u polju *iEdgeArray* predstavljaju indekse bridova koji se nalaze u ranije opisanom polju *edgeArray* klase *SurfelGraph*.

Popunjeni *TorusList* rezultat je izvođenja funkcije *ToroidalClusters*. Osim predavanja mreže, polja *VertexCH*, normale *Axis_T* i povezanog popisa *TorusList* kojeg će funkcija popuniti, funkcija treba primiti još parametre *alphaArray*, *betaArray* i *tolerance*. Parametri *alphaArray*, *betaArray* i *tolerance* su korisnički definirani i učitavaju se iz popratne datoteke. *AlphaArray* predstavlja kutove α iz jednadžbe (2-2), *betaArray* predstavlja kutove β iz jednadžbe (2-1) a *tolerance* predstavlja ε iz jednadžbe (2-19).

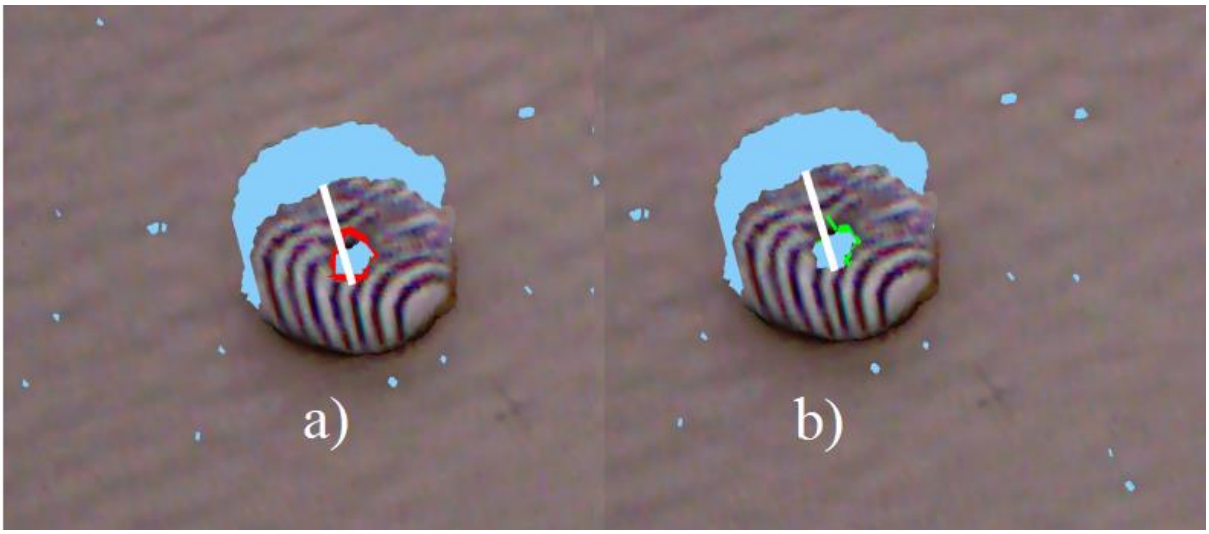
Na kraju, napravljena je funkcija za vizualizaciju sedlaste površine *TorusVisualisation*, koja je realizirana uz pomoć *Visualization Toolkita (VTK)* [6]. Funkcija na sceni iscrtava sedlastu površinu izabranu od strane korisnika. Sedlasta površina je prikazana tako da su bridovi koji definiraju pripadne konuse označeni odgovarajućom bojom. Boja pojedinog brida odgovara kutu β pripadnog konusa. Na kraju se prikazuje normala u odnosu na koju je koordinatni sustav bio transformiran. Na sceni će uvijek biti detektirano više sedlastih površina, stoga se korisniku daje opcija direktnog i cikličkog prikaza. Ako korisnik želi vidjeti određenu sedlastu površinu, pritiskom na tipku „s“ dobiva mogućnost unošenja indeksa sedlaste površine koju želi prikazati. S druge strane, svakim pritiskom na tipku „d“ prikazat će se sljedeća sedlasta površina u nizu, sve dok ne dođe do kraja, kada će ponovno prikazati prva sedlasta površina.

4. EKSPERIMENTALNA EVALUACIJA

Programu je predano 20 različitih scena, a scene su sadržavale krofnu, šalicu, bananu, zdjelu ili rolu wc papira.

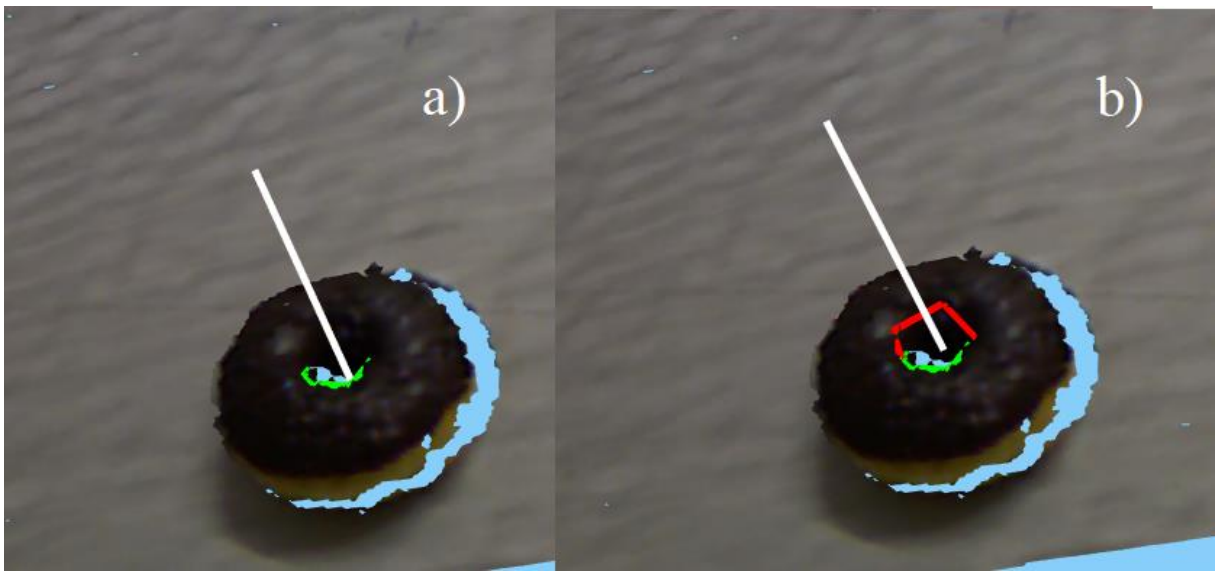
U idealnom slučaju, program bi na sceni pronašao samo jednu sedlastu površinu, ili dvije kada je u pitanju šalica (zbog ručke), što nažalost nije slučaj niti u jednoj od scena. Osim što na jednom objektu program pronalazi više sedlastih površina, one se detektiraju i na ravnini u pozadini (tzv. engl. *false positive*), te oni neće biti prikazani na slikama već će se samo navesti koliko ih je. Na slikama će biti prikazane samo sedlaste površine koje su pronađene na ciljnom objektu.

Na prvoj sceni nalazila se krofna. Pronađeno je sveukupno sedam sedlastih površina, dok se na slici 4.1. vide dvije koje se nalaze na krofni. Uspješno je detektirana rupa na krofni, no program nije uspješno povezao ova dva konusa u jednu sedlastu površinu.



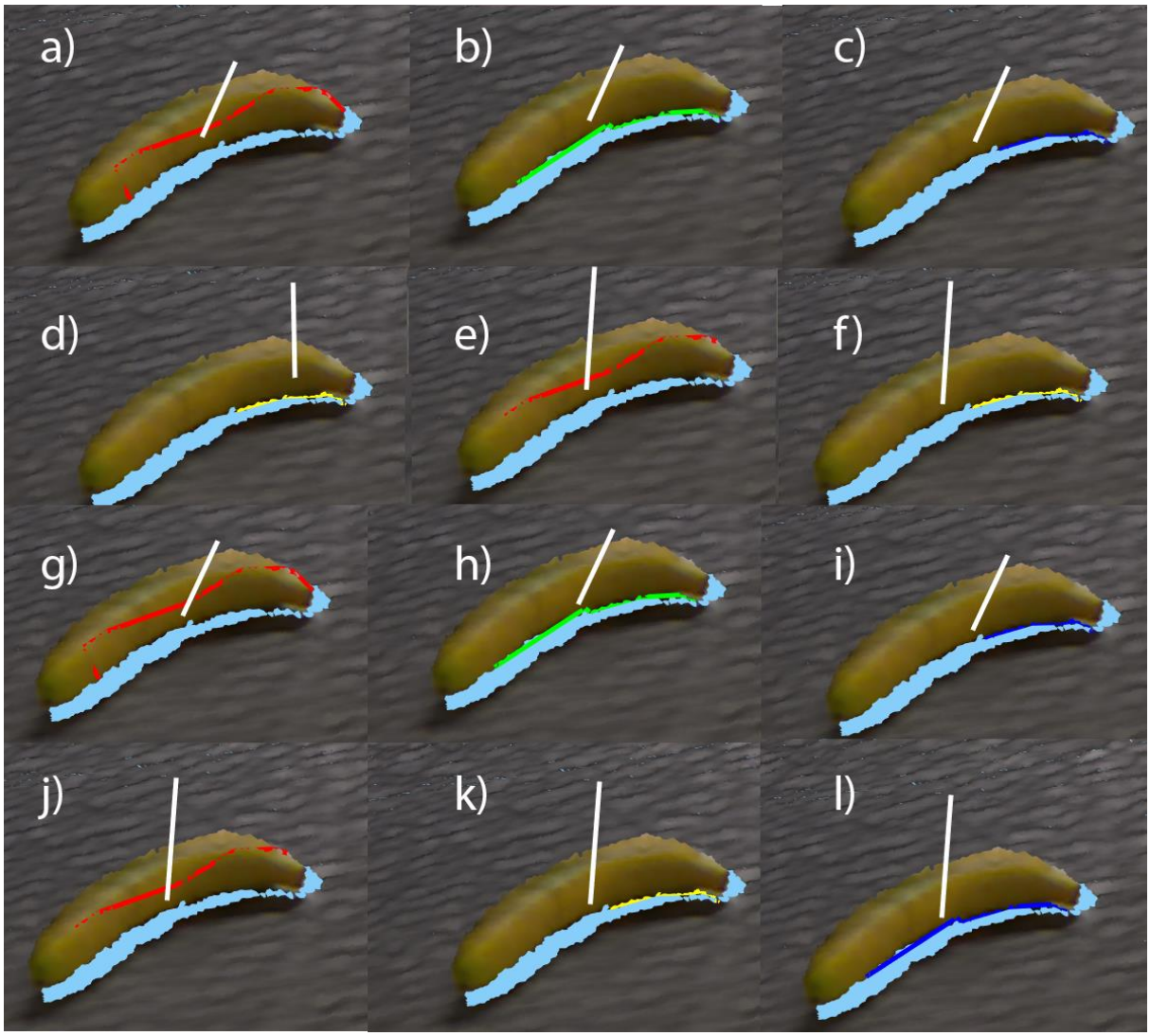
Slika 4.1. Krofna 1

Na sljedećoj sceni se također nalazila krofna. Pronađene su sveukupno četiri sedlaste površine, a na slici 4.2. se nalaze dvije detektirane na krofni. Za razliku od prošle krofne, ova krofna ima znatno manje *false positive* sedlastih površina. Na slici 4.2. b) su pronađena dva konusa, ali niti jedan nije kompletan. Relativno je teško za primijetiti, ali slike a) i b) nemaju istu z-os koordinatnog sustava sedlaste površine, jer ovdje su segmenti koji čine krofnu vjerojatno grupirani na dva načina, što je rezultiralo s dvije konveksne ljuske, gdje svaka ima svoju dominantnu plohu. Od normala te dvije dominantne plohe dviju konveksnih ljuski su nastala dva koordinatna sustava sedlastih površina.



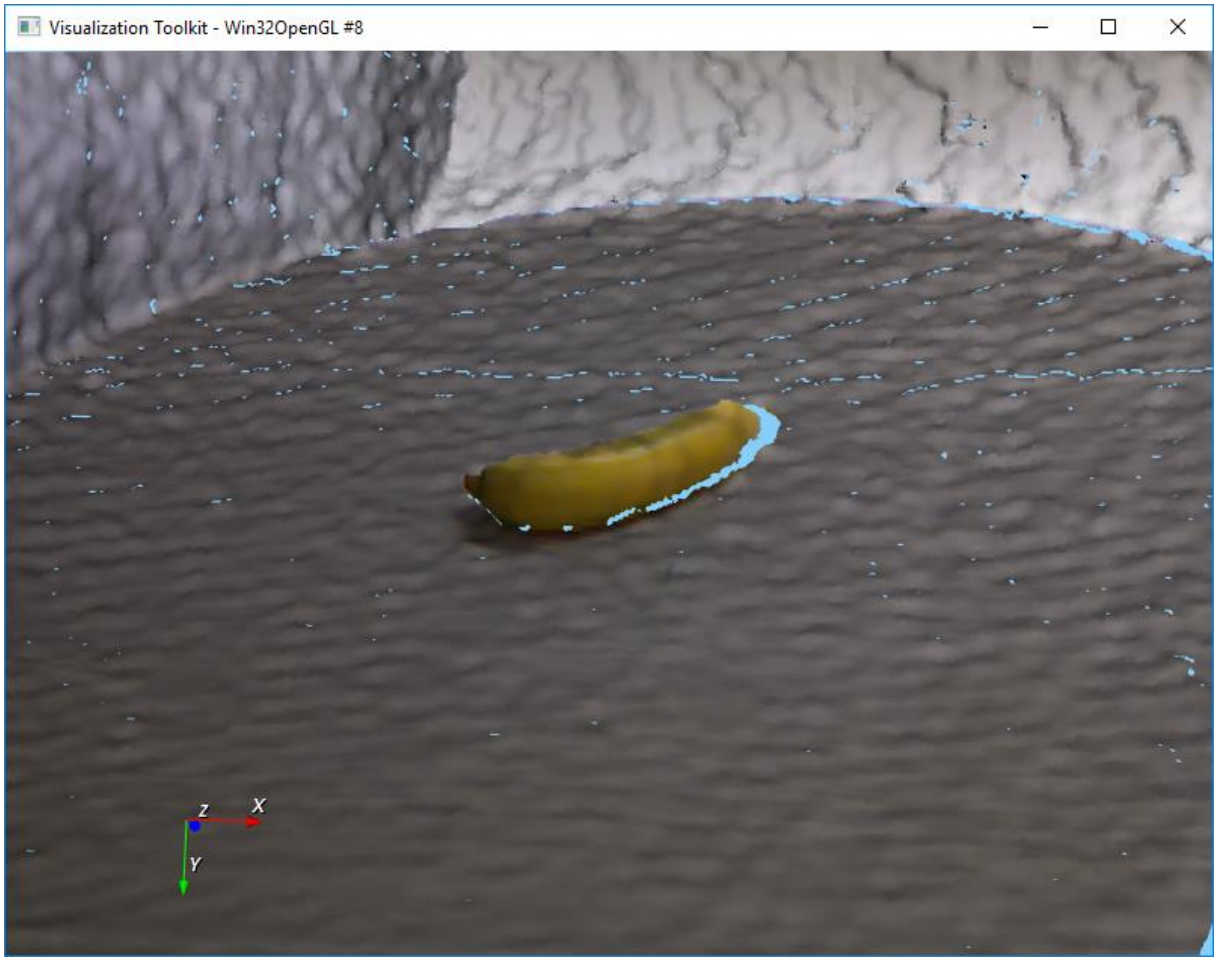
Slika 4.2. Krofna 2

Na sljedećoj sceni nalazila se banana. Pronađeno je sveukupno 16 sedlastih površina, od kojih je 12 bilo na banani i prikazane su na slici 4.3. Na ovoj sceni se dogodila slična stvar kao i na prethodnoj sceni, te sada imamo četiri različite z osi koordinatnog sustava sedlaste površine, odnosno banana je bila podijeljena na četiri različita segmenta, gdje su segmenti koji čine bananu vjerojatno grupirani na četiri načina što je rezultiralo s četiri konveksne ljuske, gdje svaka ima svoju dominantnu plohu. Od normala te četiri dominantne plohe četiriju konveksnih ljuski su nastala četiri koordinatna sustava sedlastih površina. Na slikama a), e), g) i j) svaki puta je označen isti brid, odnosno četiri puta je pronađen isti konus, koji i dalje niti u jednom slučaju nije uspješno povezan u sedlastu površinu.



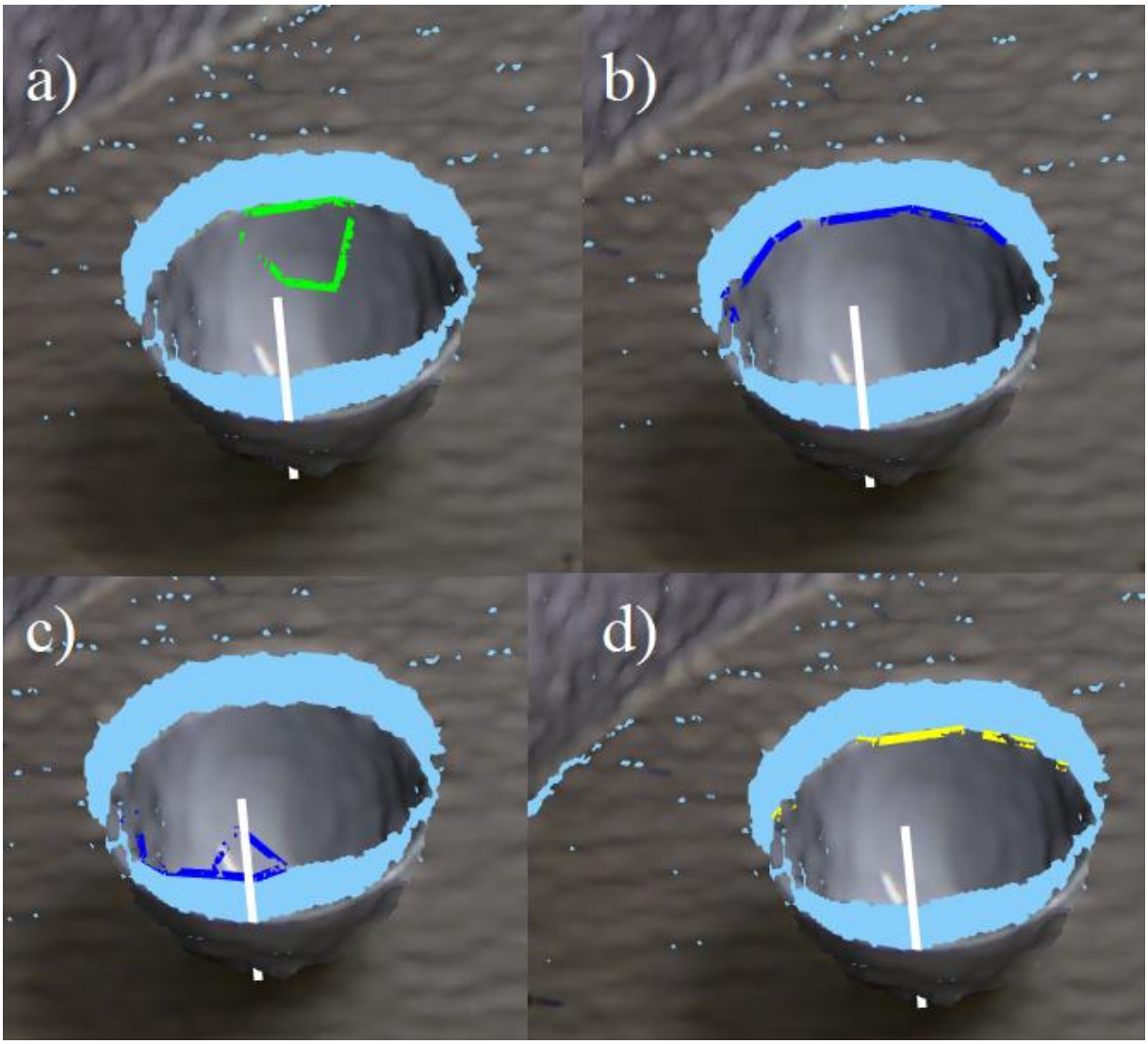
Slika 4.3. Banana 1

Uslijedila je još jedna scena banane, prikazana na slici 4.4, na kojoj nije pronađena niti jedna sedlasta površina, već samo sedam *false positive* sedlastih površina. Iz ovog kuta gledanja, ne vidi se sedlasta površina banane, već banana izgleda kao konveksni objekt.



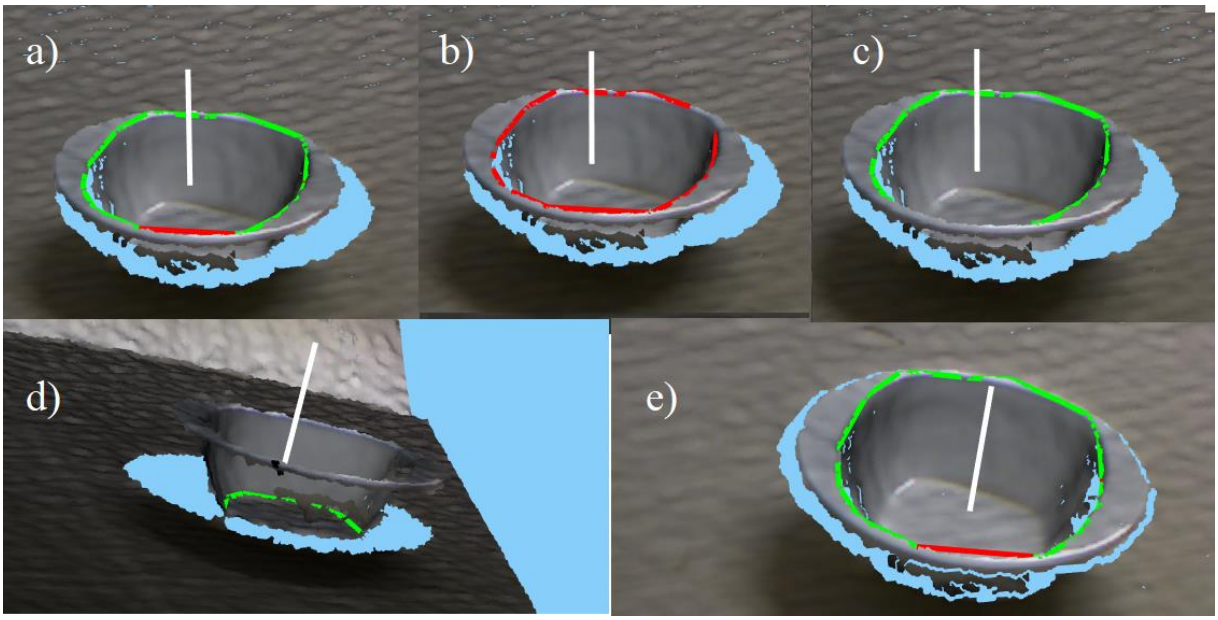
Slika 4.4. Banana 2

Na sljedećoj sceni nalazila se zdjela. Na sceni je pronađeno sveukupno osam sedlastih površina, od kojih su četiri pronađene na zdjeli i prikazane su na slici 4.5. Ovdje problem predstavlja što se objekt sastoji od dva segmenta, jednog konveksnog i drugog konkavnog. Ta dva segmenta generiraju dva konusa za rub zdjele. Pošto program nema mogućnosti povezivanja dva konusa, rub zdjele nije povezan u jedan konus. Drugi problem vidljiv na ovoj slici je da konveksni i konkavni segment, koji predstavljaju zdjelu na slici, nisu uspješno grupirani u objekt. Program je uspješno pronašao rub zdjele, vidljivo na slici b) i d), koji opet nažalost nisu uspješno povezani u sedlastu površinu. Također, dio objekta nedostaje, pa je z-os koordinatnog sustava oko koje je tražena sedlasta površina približno paralelna s podlogom, umjesto da je okomita na nju.



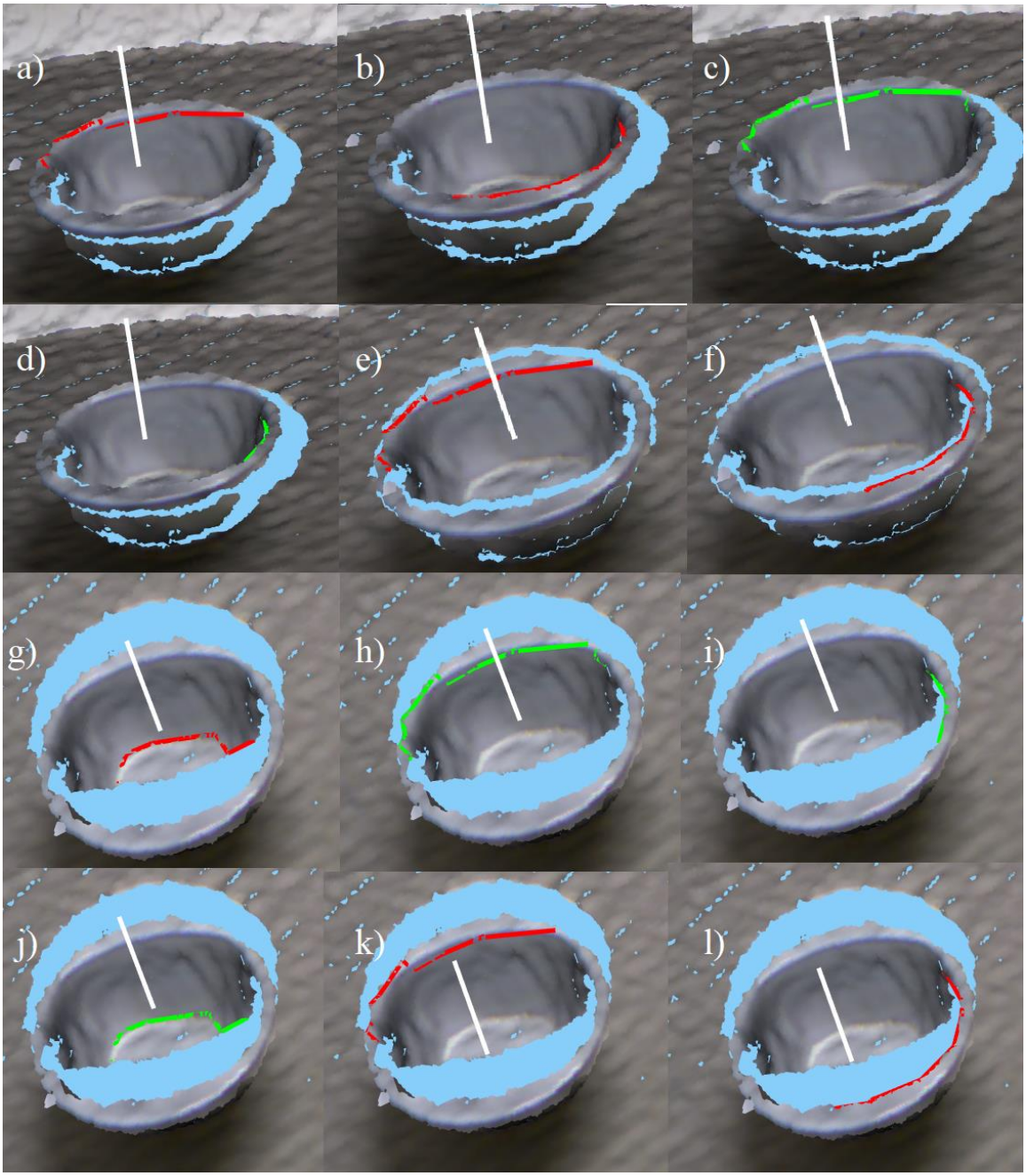
Slika 4.5. Zdjela 1

Na sljedećoj sceni je također zdjela. Na ovoj sceni je program uspješno detektirao sedlastu površinu, koja predstavlja rub udubljenja zdjele. Pronađeno je sveukupno pet sedlastih površina, od kojih se svih pet nalazilo na zdjeli, a obje z osi koordinatnog sustava sedlaste površine su okomite na podlogu, jer se uspješno skeniralo dno. Nažalost, program opet nije povezao konuse u jednu sedlastu površinu, već ih je detektirao kao zasebne sedlaste površine. Nadalje, zbog razloga objašnjenih na slici 4.2, dolazi do višestruke detekcije iste sedlaste površine. Na kraju, zdjela je ipak uspješno obrubljena (vidljivo na slikama a), b), c) i e), kao i njeno dno (slika d). Međutim, dno ne bi trebalo biti detektirano kao sedlasta površina, jer se ne radi o sedlastoj već o konkavnoj površini.



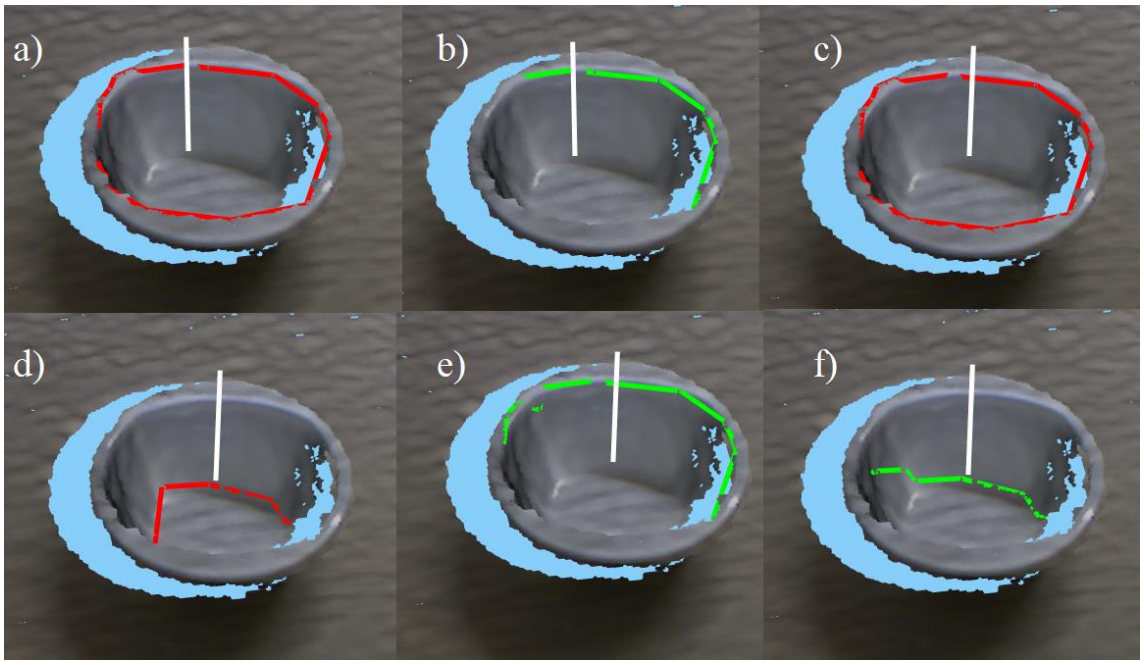
Slika 4.6. Zdjela 2

Na sljedećoj sceni, na kojoj se također nalazila zdjela, pronađeno je sveukupno 17 sedlastih površina, od kojih je 12 pronađeno na zdjeli i vidljive su na slici 4.7. Na ovoj sceni se također pojavio problem nepovoljnog kuta snimanja, kao na slici 4.5., jer se objekt sastoji od dva segmenta, jednog konveksnog i drugog konkavnog. Ta dva segmenta generiraju dva konusa za rub zdjele. Program je uspješno pronašao rub zdjele, ali opet nije povezao konuse u jednu sedlastu površinu, već ih je detektirao kao zasebne sedlaste površine. Višestruka detekcija nastaje zbog problema kao i na slici 4.2. Za razliku od slike 4.5., na ovoj sceni je z-os koordinatnog sustava sedlaste površine uspješno postavljena kao okomita na podlogu.



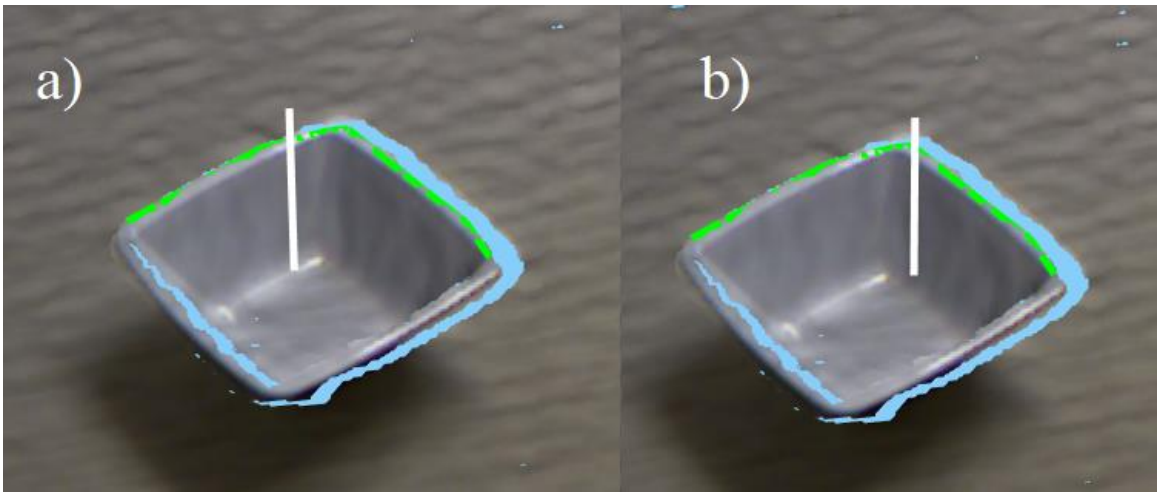
Slika 4.7. Zdjela 3

Na sljedećoj je sceni pronađeno sveukupno šest sedlastih površina, od kojih se svih šest nalazilo na zdjeli, a vidljive su na slici 4.8. Ovdje su također pronađeni djelomični konusi zbog prekida na dijelovima zdjele. Uspješno je pronađen i rub zdjele (slika a, c), uz dio dna (slika d), ali je i pronađen neobičan konus na slici f). Također, kao na slici 4.6, dno ne bi trebalo biti detektirano kao sedlasta površina, jer se ne radi o sedlastoj već o konkavnoj površini.



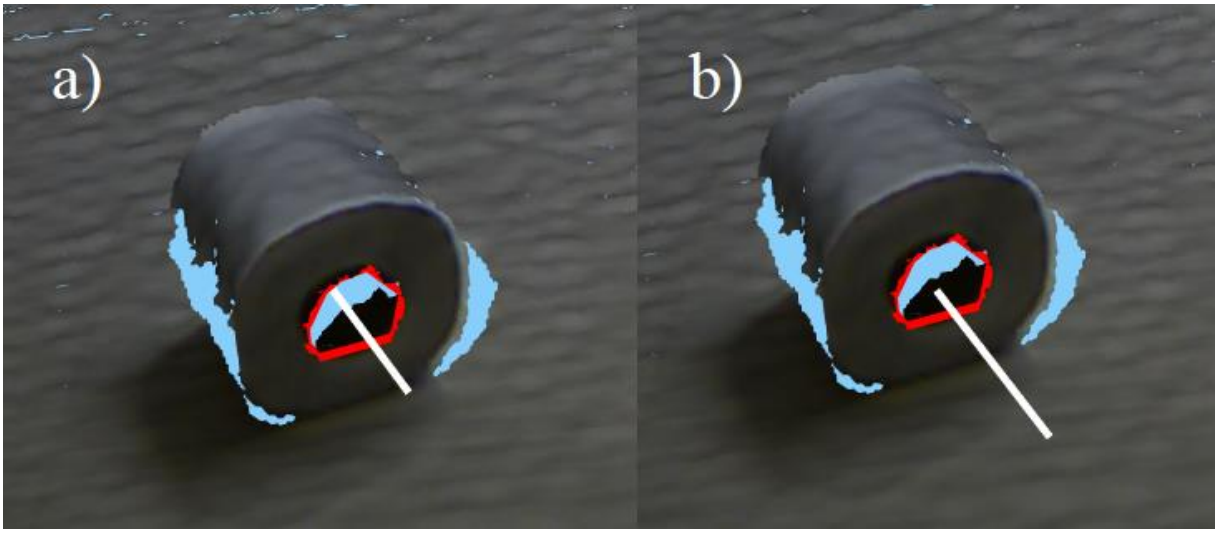
Slika 4.8. Zdjela 4

Sljedeća scena sadržavala je četvrtastu zdjelu. Na sceni je pronađeno šest sedlastih površina, od kojih su dvije pronađene na zdjeli (slika 4.9.). Može se pretpostaviti da konus nije uspješno zaokružen zbog djelomično isprekidane lijeve strane zdjele, ili neodgovarajućih prijelaza na vrhovima.



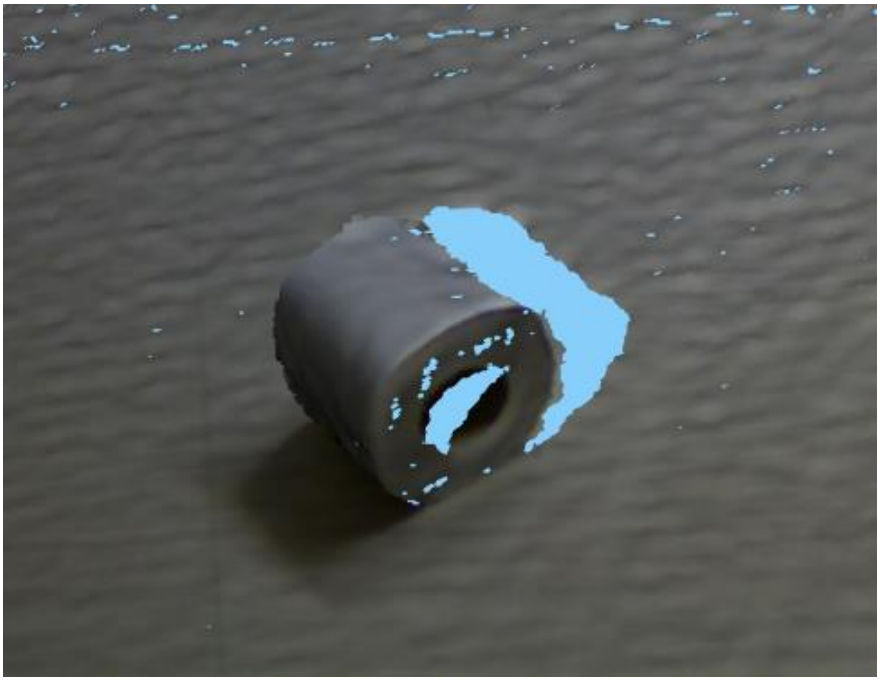
Slika 4.9. Zdjela 5

Na sljedećoj sceni nalazila se rola wc papira, a pronađene su četiri sedlaste površine, od kojih se dvije nalaze na roli, a vidljive su na slici 4.10. Kao i na slici 4.2., pronađen je isti konus, ali s dvije različite z-osi koordinatnog sustava sedlaste površine.



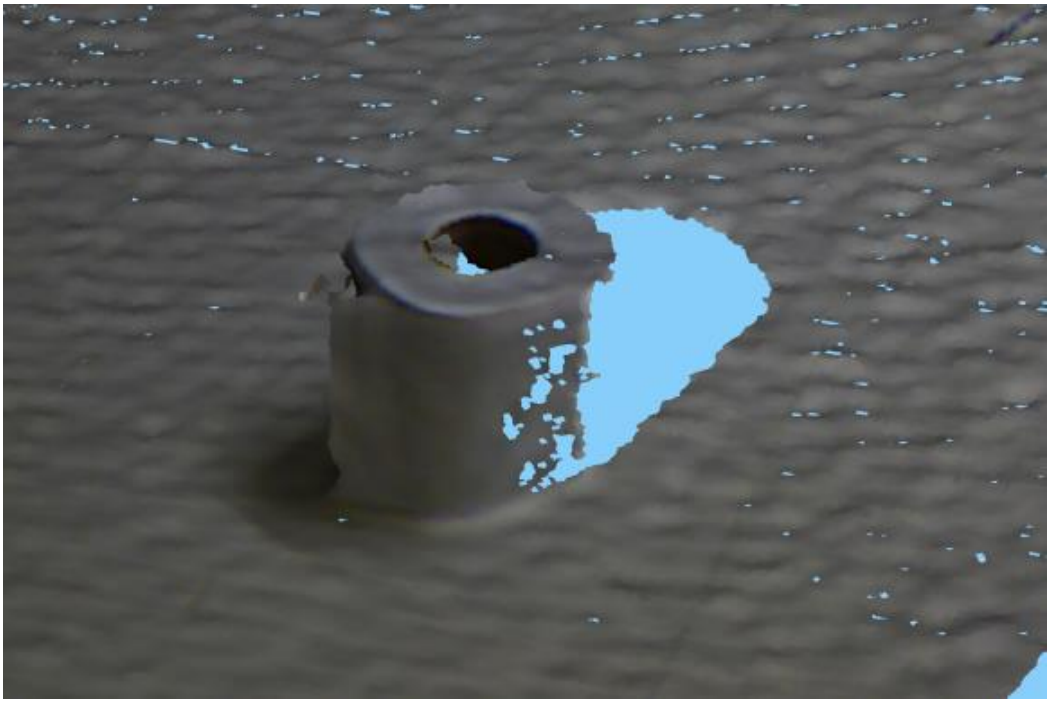
Slika 4.10. Rola wc papira 1

Na sljedećoj sceni se također nalazila rola wc papira, ali ovoga puta je pronađena jedna *false positive* sedlasta površina u pozadini. Scena je vidljiva na slici 4.11. na kojoj se vidi da unutrašnja površina rupe u roli nije prikazana niti jednom ravnom plohom, pa konus nije mogao biti detektiran.



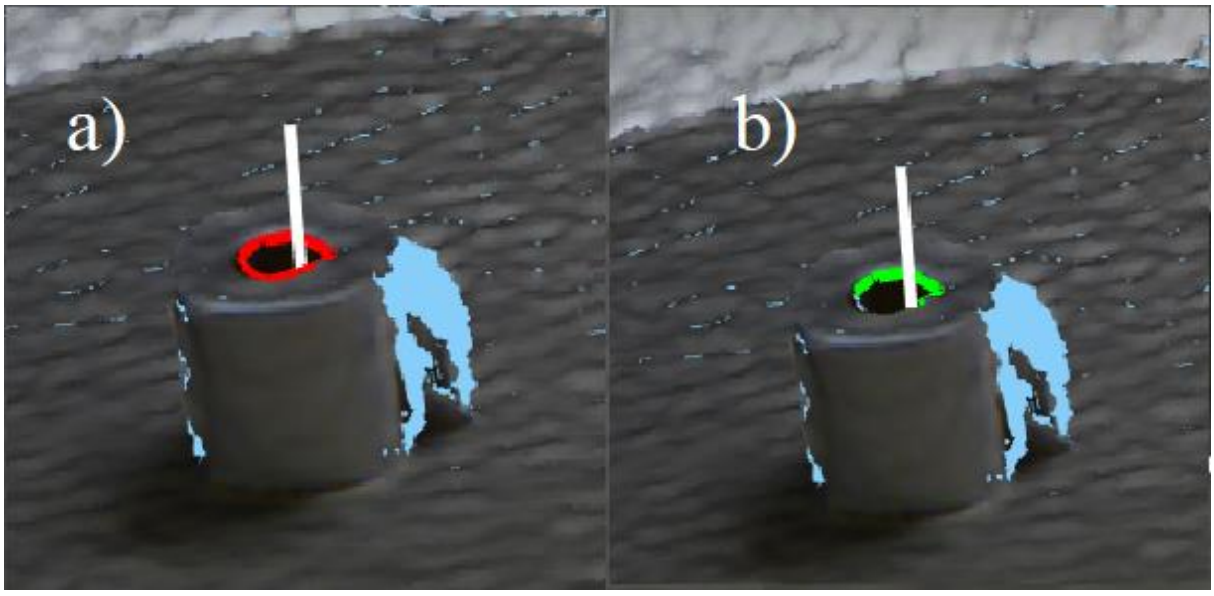
Slika 4.11. Rola wc papira 2

Na sljedećoj sceni, kao i na prethodnoj, pronađen je jedna *false positive* sedlasta površina u pozadini. Scena je vidljiva na slici 4.12. Ovog puta rola je bila uspravna, ali sedlasta površina nije detektirana.



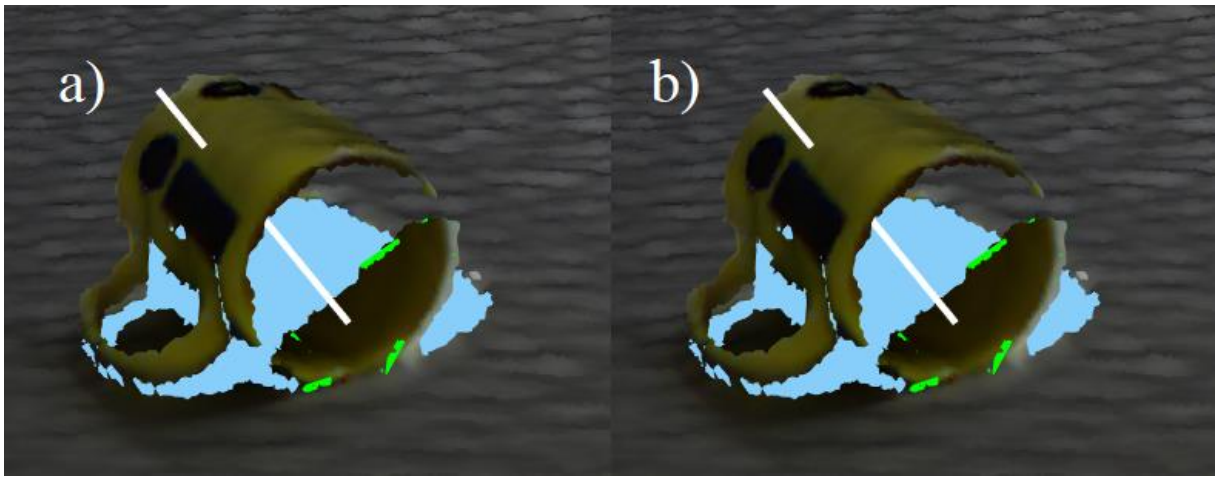
Slika 4.12. Rola wc papira 3

Na sljedećoj sceni se nalazila također uspravna rola wc papira i na njoj su sveukupno pronađene četiri sedlaste površine, od kojih su dvije pronađene na roli i vidljive su na slici 4.13. Kao u slučaju slike 4.2. rola je prikazana s više objekata dobivenih različitim grupiranjima segmenata.



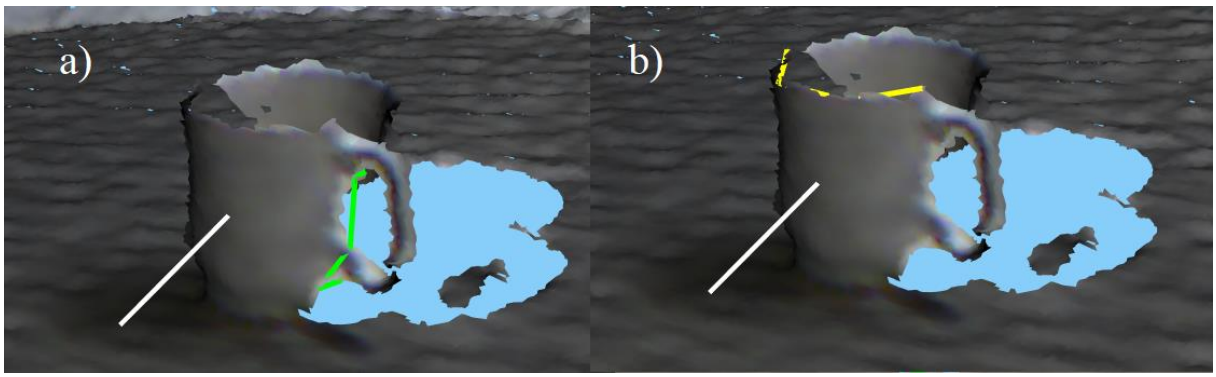
Slika 4.13. Rola wc papira 4

Na sljedećoj sceni nalazila se polegnuta šalica, na kojoj je pronađeno sveukupno 12 sedlastih površina, od kojih dvije na samoj šalici i vidljive su na slici 4.14. Niti jedna od te dvije nije obuhvatila rupu. Šalica je isprekidana, a može se vidjeti da nedostaje dio ruba, sa svake strane, što predstavlja problem. Nadalje, program nije dobro odredio dominantne ravnine oko čije normale bi se tražile rupe, tj. sedlaste površine. Na šalicama, zbog njihove strukture, jako je teško pronaći dominantnu ravninu za koju će normala izlaziti iz njihova otvora.



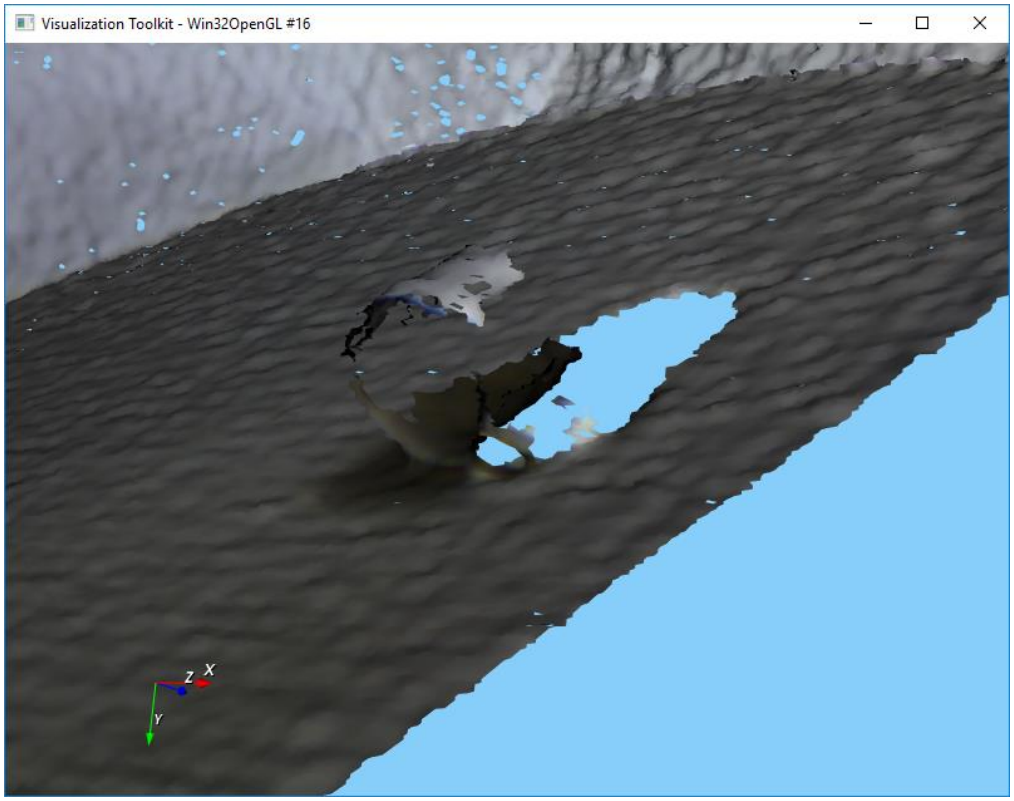
Slika 4.14. Šalica 1

Na sljedećoj sceni nalazila se uspravna šalica, na kojoj su pronađene sveukupno četiri sedlaste površine, od kojih dvije na samoj šalici i vidljive su na slici 4.15. Ovdje je obuhvaćen dio otvora rupe šalice kao i dio rupe hvataljke.



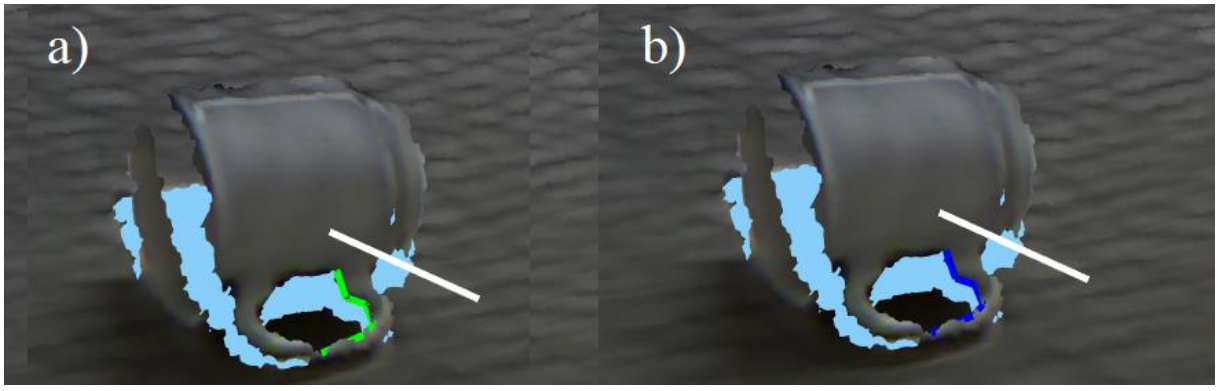
Slika 4.15. Šalica 2

Na sljedećoj sceni se nalazila polegnuta šalica, pronađene su samo dvije *false positive* sedlaste površine. Scena je vidljiva na slici 4.16. i može se primijetiti da je čaša nekompletna, što uzrokuje nepronalazak niti jedne rupe tj. sedlaste površine na šalici



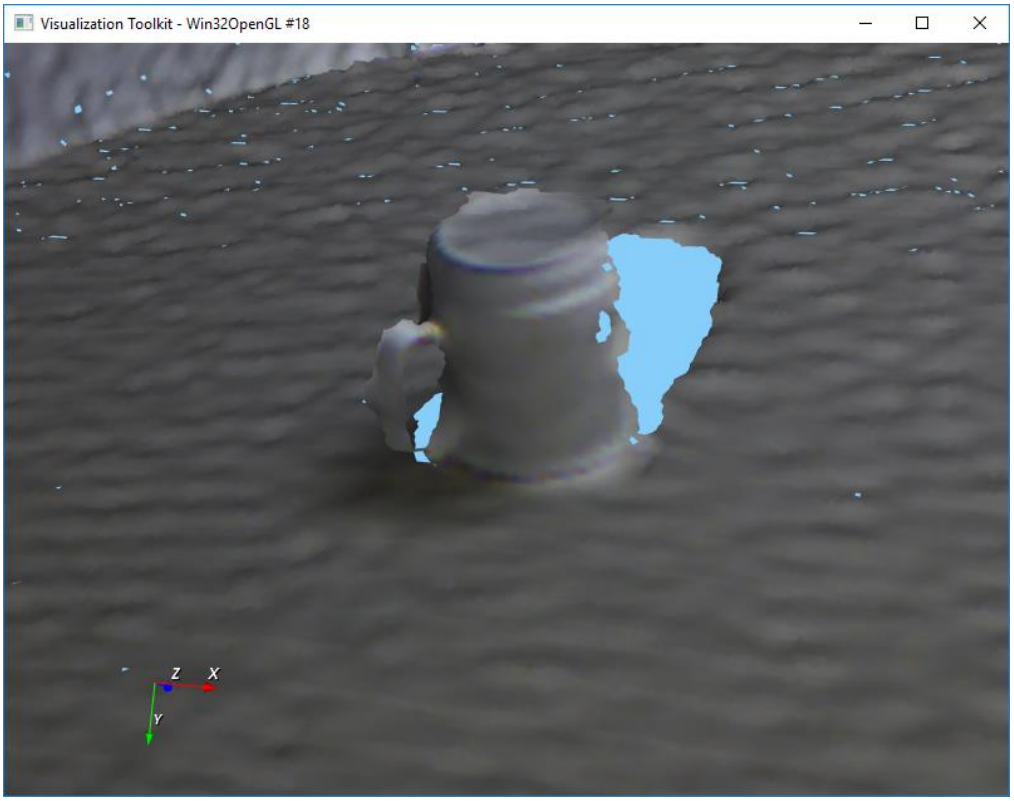
Slika 4.16. Šalica 3

Na sljedećoj sceni se također nalazila polegnuta šalica. Pronađeno je sveukupno 16 sedlastih površina, od kojih se samo dvije nalaze na šalici, a mogu se vidjeti na slici 4.17. Obje sedlaste površine predstavljaju rupe na dršci. Otvor šalice nije detektiran zbog teškoće određivanja dominante plohe, zbog kuta iz kojeg je scena snimljena. Normala bi trebala izlaziti iz otvora šalice oko koje bi se sedlasta površina detektirala.



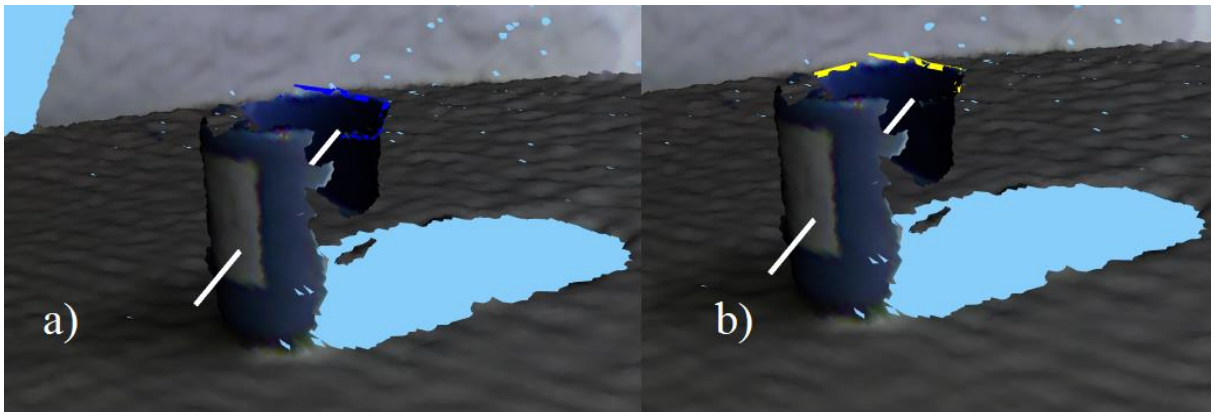
Slika 4.17. Šalica 4

Na sljedećoj sceni se nalazila preokrenuta šalica, pronađene su četiri *false positive* sedlaste površine. Scena je vidljiva na slici 4.18. Na sceni je vidljiva samo rupa na hvataljki, snimljena iz kuta iz kojeg ju je teško detektirati.



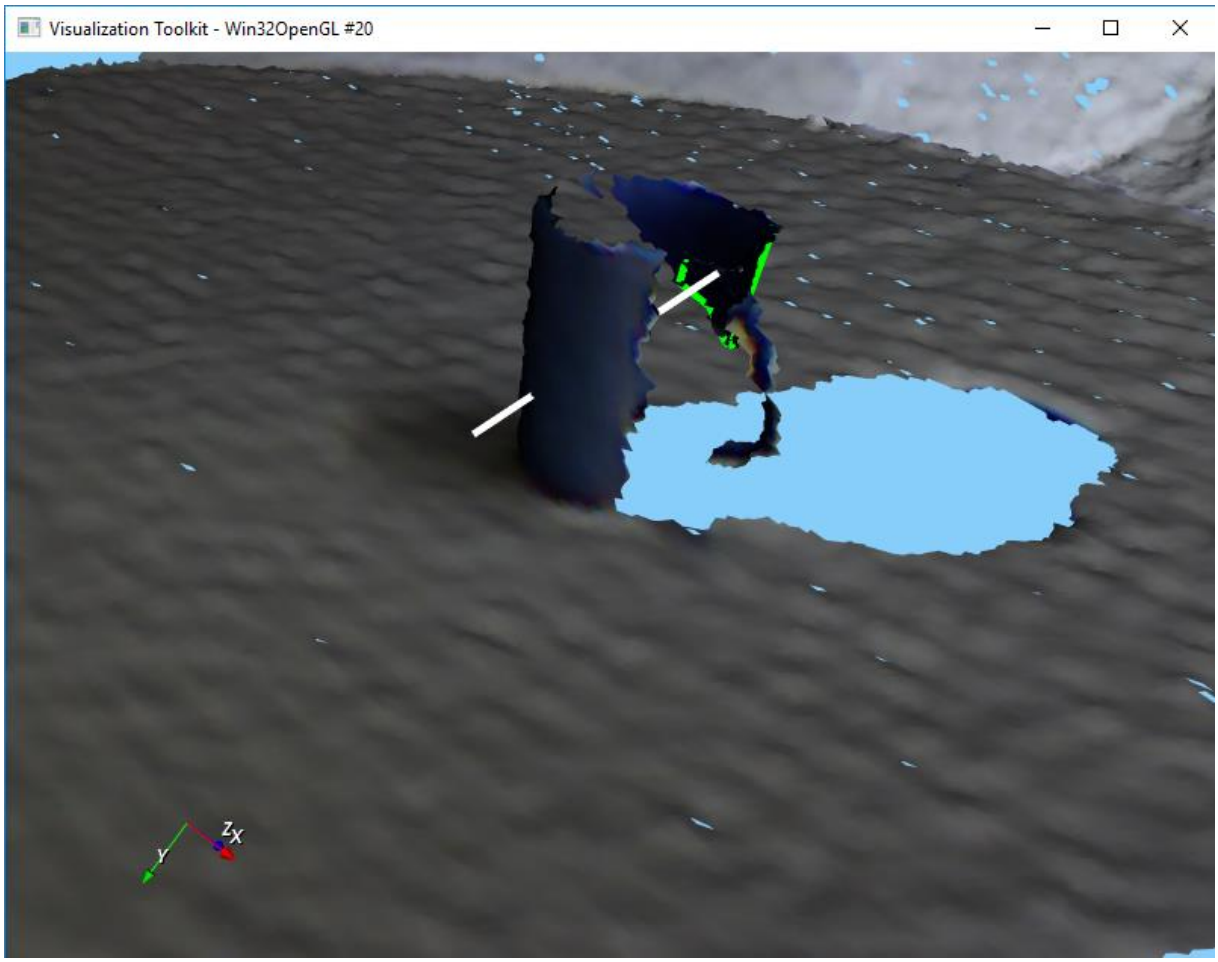
Slika 4.18. Šalica 5

Na sljedećoj sceni nalazila se uspravna šalica, na kojoj su pronađene sveukupno četiri sedlaste površine, od kojih dvije na samoj šalici i vidljive su na slici 4.19. Iako nedostaje velik dio šalice, dio ruba je uspješno detektiran. Kao i do sada, dva segmenta nisu povezana u objekt, stoga nije dobro određena z-os koordinatnog sustava sedlaste površine.



Slika 4.19. Šalica 6

Na posljednjoj sceni nalazila se ista uspravna šalica kao na slici 4.19. samo pomaknuta u stranu. Na njoj su sada pronađene dvije sedlaste površine, od kojih jedna na samoj šalici i vidljiva je na slici 4.20. Problem i dalje predstavlja što dva segmenta nisu povezana u objekt, pa nije dobro određena z-os koordinatnog sustava sedlaste površine.



Slika 4.20. Šalica 7

5. ZAKLJUČAK

Program za detekciju rupa i udubljenja je relativno uspješno realiziran u okviru programske biblioteke Robot Vision Library (RVL). Iako je algoritam detaljno matematički opisan, on je predviđen za detekciju na modelima, koji su idealni, za razliku od realnih scena snimljenih RGB-D kamerom, koje sadržavaju šum te djelomično vidljive objekte.

Tijekom testiranja uočeni su problemi. Jedan od njih je grupiranje segmenata koji sadrže objekt na dva (ili više) načina, što rezultirala s dvije (ili više) konveksne ljuske, gdje svaka ima svoju dominantnu plohu. Od normala tih dominantnih ploha konveksnih ljuski nastaju dva (ili više) koordinatna sustava sedlastih površina koji detektiraju istu rupu (udubljenje), te ih označe kao dvije različite sedlaste površine.

Nadalje, zbog isprekidanosti objekta, objekt se može sastojati od dva segmenta, jednog konveksnog i drugog konkavnog. Ta dva segmenta generiraju dva konusa za rub objekta. Pošto program nema mogućnosti povezivanja dva konusa generirana za neku konveksnu ljusku za isti beta, rub objekta se ne poveže u jedan konus.

Također, nepronalazak dominantne ravnine stvara problem, jer se tada ne uzme dobra normala koja će se koristiti kao z-os koordinatnog sustava, oko koje se traže sedlaste površine, tj. rupe i udubljenja. To se može dogoditi zbog nepotpunog objekta ili kad dva segmenta ne budu povezana u objekt.

Osim detekcije *false positive* sedlastih površina u pozadini, na nekoliko mjesta se dno zdjele detektiralo kao sedlasta površina, što nije točno, jer je dno zdjele konveksno.

U budućem radu moguće je proširiti program da provjerava kombinacije konusa različitih konveksnih ljuski te ih povezuje u jedan zajednički konus ako zadovoljavaju iste uvijete koji se provjeravaju do sada, ali samo unutar iste konveksne ljuske. Uz samo proširivanje programa, možda je potrebno prirediti i kvalitetniji testni skup, na kojem će se nalaziti potpuniji objekti jer su scene na kojima je program testiran, iako ih je mnogo, sve snimljene istom kamerom.

LITERATURA

[1] Cupec, R; VolumeNet: Flexible Model for Shape Classes, (Osijek, 2018.)

[2] Coxeter, H.S.M.; Regular Polytopes, (3rd Edition, Dover, 1973.)

[3] Gerd, F; Mathematical Models, (Springer Fachmedien Wiesbaden GmbH 1986, 2017.

[4] Cupec, R.; Detection of local reference frames and geometric primitives based on planar patches (Osijek, 2018.)

[5] Cupec, R; 3D Mesh Segmentation into Planar Segments Based on Region Growing and Plane Intersection, (Osijek 2016)

[6] VTK , https://vtk.org/Wiki/VTK , 30.06.2019.

SAŽETAK

Cilj ovog diplomskog rada je izrada programa za detekciju rupa i udubljenja na scenama snimljenim RGB-D kamerom. Rubovi rupa i udubljenja matematički predstavljaju sedlaste površine. Objašnjena je matematička pozadina geometrijskih struktura koje će se detektirati. Razvijen je program za detekciju sedlastih površina realiziran u okviru programske biblioteke *Robot Vision Library (RVL).* Izrađena je funkcija koja detektirane površine predočava uz pomoć *Visualization Toolkita (VTK).* Program je testiran na raznim scenama na kojima su se nalazili objekti s rupama i udubljenjima poput šalica, zdjela, banana, krofni i rola wc papira u raznim položajima.

Ključne riječi: detekcija rupa, detekcija udubljenja, Torus, Sedlaste površine, Robot Vision Library

ABSTRACT

This work addresses the problem of recognizing holes and dents in scenes captured by RGB-D camera. Edges of holes and dents are mathematically considered as a part of saddle surfaces. Saddle surfaces from the mathematical point of view and corresponding algorithm for detection of saddle surfaces are explained. Saddle surface detection program in this work is developed as a part of *Robot Vision Library (RVL)* developed by the Faculty of Electrical Engineering, Computer Science and Information Technology (FERIT) in Osijek. *RVL* will be complemented with a function which detects saddle surfaces implementing explained algorithm and a function for visualization of those detected surfaces. The program was tested on multiple scenes containing objects like donut, mug, bowl, banana and toilet paper in different positions.

Key words: hole detection, dent detection, Torus, saddle surface, Robot Vision Library

ŽIVOTOPIS

Siniša Stanić rođen je 1995. godine u Našicama. Pohađao je Osnovnu školu „Tenja“ u Tenji. Prvu gimnaziju Osijek upisuje 2010. godine. Nakon toga upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija (FERIT) u Osijeku gdje stječe naziv prvostupnika računarstva 2017. godine. Svoje obrazovanje nastavlja na diplomskom studiju Automobilsko računarstvo i komunikacije.

_____

Siniša Stanić

PRILOZI

Programski kod je priložen na CD-u uz ovaj rad.