ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი

თორნიკე სულაძე

# სითბური ასახვა: სამგანზომილებიანი ობიექტის სითბური სეგმენტაცია.

(სითბურ ცენტრებზე დაყრდნობით)

კომპიუტერული მეცნიერებები

სამაგისტრო ნაშრომი შესრულებულია კომპიუტერული მეცნიერებების მაგისტრის ხარისხის მოსაპოვებლად

ალექსანდრე გამყრელიძე-

ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტის

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტის

კომპიუტერული მეცნიერებების მიმართულების

სრული პროფესორი

გოდერძი ფრუიძე-

ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტის

ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტის

მოწვეული ლექტორი,

კიბერნეტიკის ინსტიტუტის მათემატიკური კიბერნეტიკის განყოფილების

მეცნიერ თანამშრომელი

თბილისი, 2013

# სარჩევი

# ანოტაცია

ნაშრომის მიზანია შეიქმნას პროგრამული უზრუნველყოფა, რომელიც მოახდენს სამგანზომილებიანი ობიექტის სეგმენტაციას. **სამგანზომილებიანი ობიექტის სეგმენტაცია**(შემდგომში **სოს**) გულისხმობს ერთი მთლიანი ობიექტის დაყოფას უფრო მცირე ზომის მნიშვნელოვან კომპონენტებად. **სოს** არის ფუნდამენტური ამოცანა, რომლის იმპლემენტაციაც მოითხოვს დაბალი დონის პროგრამირებისა და კომპიუტერული გეომეტრიის საფუძვლების ცოდნას. ამ ამოცანის შედეგები და აპლიკაციები გამოიყენება მრავალ სხვადასხვა დარგში, როგორებიცაა მაგალითად: კომპიუტერული ხედვა, კომპიტერულად მათვადი დიზაინი, ბიოინფორმატიკა, სამგანზომილებიანი სამედიცინო დასურათება. დღესდღეისობით არსებობს **სოს**-ის რამდენიმე მიდგომა, თუმცა მათი უმრავლესობა მხოლოდ თეორიული ნაშრომია და არ არის ფართოდ დანერგილი პრაქტიკაში. ამ ნაშრომში გავაანალიზებთ **ობიექტის სითბური სეგმენტაციის** მეთოდს და მიღებულ ცოდნას მივცემთ პრაქტიკულ გამოყენებას. უფრო კონკრეტულად კი მოვახდენთ **HC-Kmeans** ალგორითმის იმპლემენტაციას. სითბური სეგმენტაცია იძლევა იმის გარანტიას, რომ შედეგი იქნება:

- მდგრადი ტოპოლოგიური ხარვეზების მიმართ.
- მდგრადი ზედაპირის სხვადასხვა გარდაქმნების მიმართ.
- ადამიანურ აღქმასთან მიახლოვებული.

# Abstract

The main goal of this work is to create 3D object segmentation software. **3D object segmentation(3DOS)** is the process which divides one whole 3D object to smaller meaningful components. **3DOS** is fundamental task, the implementation of which requires knowledge of computational geometry and low-level programming. The results and applications of **3DOS** are used in areas as diverse as computer vision, computer-aided design, bioinformatics, 3D medical imaging. Nowadays there are several **3DOS** methods, but most of them are just theoretical works and they don't have wide distribution in practice. In this paper we analyze **3D object heat segmentation** and apply acquired knowledge to practice. More specifically we implement **HC-Kmeans** algorithm. Heat segmentation as described in HM satisfies:

- robustness to the topological defects of the surface.
- robustness to the different transformations of the surfaces.
- close conformation to human perception.

# 1. შესავალი

## 1. 1 საფუძვლები

ბოლო წლების განმავლობაში სწრაფად გაიზარდა სამგანზომილებიანი მოდელების გამოყენების სიხშირე სხვადასხვა პროფესიულ დარგებში, რასაც მოჰყვა სამგანზომილებიანი მოდელების ათვისების ტექნოლოგიების განვითარება. შესაბამისად წარმოიშვა მუდმივად ზრდადი მოთხოვნა მოდელების ავტომატური დამუშავებისა და გაანალიზების მიმართ. პირველი მნიშვნელოვანი ნაბიჯი მოდელის დამუშავებისა, არის მისი სეგმენტაცია უფრო მცირე ზომის კომპონენტებად, რაც საკმაოდ რთული ამოცანაა. სეგმენტაციის მიდგომას წინ უდგას შემდეგი პრობლემები

- ზედაპირის იზომეტრიული გარდაქმნებისადმი მგრძნობელობა.
- ზედაპირის ტოპოლოგიური დარღვევებისადმი მგრძნობელობა.
- დასამუშავებელ მოდელში თანდაყოლილი ხარვეზებისადმი მგრძნობელობა.
- სეგმენტაციის პროცესის ადამიანურ აღქმასთან შეუთავსებლობა.

ჩვენ განვსაზღვრავთ სეგმენტაციის სქემას, რომელიც გაითვალისწინებს და გადაჭრის ზემოთხსენებულ პრობლემებს. ამ სქემას ეწოდება **ადამიანის ვიზუალურ აღქმასთან შესაბამისობაში მყოფი სეგმენტაცია(perceptually consistent mesh segmentation (PCMS))(განსაზღვრება 1.1.1)**

**განსაზღვრება 1.1.1 ადამიანის ვიზუალურ აღქმასთან შესაბამისობაში მყოფი სეგმენტაცია(perceptually consistent mesh segmentation (PCMS)):** როდესაც ადამიანი ეცნობა ნებისმიერ უცხო ობიექტსა თუ გაურკვეველ ფორმას, ის ცდილობს მის დაკავშირებას უკვე ნაცნობ სამგანზომილებიან სხეულებთან, ქვეცნობიერად ქმნის ფორმების გარკვეულ შაბლონებს და ამის მიხედვით ცდილობს, რომ დაადგინოს ობიექტის წარმომავლობა. ადმიანის გონება იგივე პრინციპით ახდენს ობიექტების დაყოფას მნიშვნელოვან კომპონენტებად და სწორედ ეს იგულისხმება ჩვენს მიერ მოყვანილი სქემის **ადამიანის ვიზუალურ აღქმასთან შესაბამისობაში.**

ბოლო წლების განმავლობაში **PCMS**-ში ინტეგრირებული მახასიათებლები უფრო მნიშვნელოვანი გახდა მოდელების დამუშავებისა და ანალიზის დროს, რადგან ის უფრო

მეტ ინფორმაციას გვაძლევს მოდელის თვისებების შესახებ. **PCMS**-ი აადვილებს სამგანზომილებიანი მოდელის ინტერპრეტაციას რადგან ის წარმოაჩენს ზედაპირის თავდაპირველად დაფარულ გეომეტრიულ სტრუქტურას. ეს შეიძლება გამოისახოს წმინდა გეომეტრიული ტერმინებში, სემანტიკური ინფორმაციის სახით ან ორივეთი ერთდროულად. შედეგად მიღებულ გეომეტრიულ სტრუქტურას გააჩნია საკმარისი ინფორმაცია იმისათვის, რომ დაყოს ობიექტები ფუნქციონალურ კომპონენტებად. ეს პროცესი მიმდინარეობს ადამინის აღქმასთან მიახლოებული გზით. სეგმენტაციის სხვა მიდგომებთან შედარებით **PCMS**-ი არის უფრო ადეკვატური მოდელის თვისებების გაანალიზებისას, რაც გამოყენებას პოულობს სხვადასხვა სამოდელო აპლიკაციებში, მაგალითად:

ამის მაგალითებია:

- ორი ან მეტი სამგანზომილებიანი ობიექტის ან მათი ნაწილების შესაბამისობის დადგენა.
- **მოდელიდან ჩონჩხის ამოღება (განსაზღვრება 1.1.2).**
- ტექსტურის ასახვა მოდელზე.
- მოდელის გამარტივება.

**განსაზღვრება 1.1.2 მოდელიდან ჩონჩხის ამოღება.** სამგანზომილებიანი ობიექტის ჩონჩხი იგივე დანიშნულებისა და თვისებების მქონე სტრუქტურაა როგორც ადამიანის ჩონჩხი. მოდელის ჩონჩხი:

- შეიცავს ძვლებს და სახსრებს
- გამოყენება კომპიუტერულ ანიმაციაში მოდელის ანიმირებისათვის

როგორც წესი მოდელის ჩონჩხის აგება ხდება ანიმატორების მიერ, ხოლო მოდელიდან ჩონჩხის ამოღება გულისხმობს ჩონჩხის გენერირების ავტომატურ პროცესს.

## 1.2 სეგმენტაციის სხვადასხვა მიდგომები

### 1.2.1 ხელოვნური სხეულების იერარქიული სეგმენტაცია

ეს მიდგომა გამოაჩენს მოდელის ხელოვნურ სტრუქტურას, რაც შედეგად გვაძლევს სეგმენტების იერარქიას. მეთოდი დაფუძნებულია ორ მთავარ კონცეპტზე:

- ვიკვლევთ აზომვების ოჯახს, რომელსაც ეწოდება **გაბნევის მანძილი**(პარაგრაფი 1.2.2).
- წარმოვადგენთ **შუალედური სტრუქტურების**(პარაგრაფი 1.2.3) განზოგადებას.

**გაბნევის მანძილი** განსაზღვრავს სეგმენტაციისათვის საჭირო მახასიათებლების მქონე საზომს. ის მდგრადია ზედაპირის ხარვეზების და ტოპოლოგიური სახესხვაობების მიმართ, იგი აგრეთვე დამოუკიდებელია ზედაპირის დეფორმაციების მიმართ. დეტალიზაციის თითოეულ დონეზე მოდელი დაიყოფა **შუალედურ სტრუქტურასა** და სეგმენტებს შორის ბიექციის გამოთვლით.

### 1.2.2 გაბნევის მანძილი

**გაბნევის მანძილი** პირველად გამოჩნდა სტოქასტურ დინამიურ სისტემებში. მეთოდი გამოიყენეს განზომილებათა შემცირების და მონაცემთა პარამეტრიზაციის პრობლემების გადაწყვეტაში. ამ კონტექსტში გვაქვს **მარკოვის შემთხვევითი ველი (განსაზღვრება 1.2.2.1),** რომელიც განსაზღვრულია $X$ მონაცემთა სიმრავლეზე. გაბნევის მანძილი $x$ და $y$ წერტილებს შორის დროის ბიჯით $t$ მოიცემა შემდეგნაირად:

$$\mathcal{D}_t^2(x,y) = \int_X \big(p_t(x,y) - p_t(y,z)\big)^2 d\mu(z)$$

**ფორმულა 1.2.2.1**

სადაც $p_t(.,.)$ არის ალბათობა იმისა, რომ გავივლით ორ წერტილს შორის $\leq t$ ბიჯით და $\mu(.)$ არის წერტილების განაწილება $X$-ში.

**განსაზღვრება 1.2.2.1 მარკოვის შემთხვევითი ველი(Markov random field (MRF)).** იგივე მარკოვის ქსელი (Markov network) ან არაორიენტირებული **გრაფისებრი მოდელი (განსაზღვრება 1.2.2.2)** არის შემთხვევითი ცვლადების სიმრავლე, რომლებსაც ახასიათებთ არაორიენტირებული გრაფით აღწერილი **მარკოვის თვისება(Markov property) (განსაზღვრება 1.2.2.3).** ფიზიკისა და ალბათობის სფეროში მარკოვის ქსელი გამოყენება მონაცემების გარკვეული დამოკიდებულებების წარმოსაჩენად.

**განსაზღვრება 1.2.2.2 გრაფისებრი მოდელი.** წარმოადგენს ალბათურ მოდელს, რომელშიც გრაფი განსაზღვრავს შემთხვევით ცვლადებს შორის არსებული დამოკიდებულებების ამსახველ სტრუქტურას. ძირითადად გამოიყენება ალბათობის თეორიაში, სტატისტიკაში და მანქანურ სწავლებაში.

**განსაზღვრება 1.2.2.3 მარკოვის თვისება.** ალბათობის თეორიაში და სტატისტიკაში **მარკოვის თვისება** ეწოდება სტოქასტური პროცესის ისეთ თვისებას, რომელიც არ არის დამოკიდებული მეხსიერებაზე. სტოქასტურ პროცესს აქვს **მარკოვის თვისება,** თუ მისი მომავალი მდგომარეობა დამოკიდებულია მხოლოდ მიმდინარე მდგომარეობაზე და არა მისი წინა მდგომარეობების მიმდევრობაზე.

**გაბნევის მანძილი** მიუთითებს $X$-ის წერტილებს შორის კავშირის სიხშირეს. ეს აზომვა იქნება მცირე თუ გვაქვს ბევრი შემთხვევითი გავლა $\leq t$ დროში და აზომვა იქნება დიდი საპირისპირო შემთხვევაში.

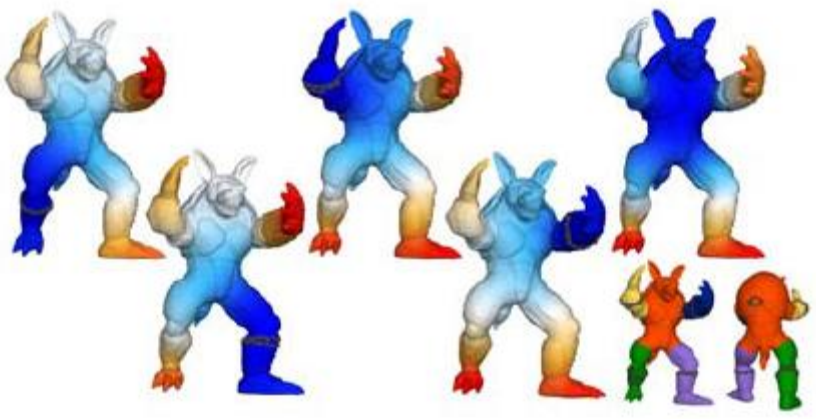
ჩვენ ვზღუდავთ მონაცემთა სიმრავლეს კომპაქტურ ორგანზომილებიან მრავალსახეობა $M$-მდე. ცნობილია, რომ სტოქასტურ პროცესებში $M$-ზე შემთხვევითი გავლების ზრდის ლიმიტი ექვივალენტურია **ბრაუნის გზებისა (განსაზღვრება 1.2.2.4),** ამიტომ ალბათური ფუნქცია $p_t(.,.)$ გადაიქცევა სითბური ბირთვის $\mathcal{K}_t(.,.)$ ფუნქციად $M$-ზე. ამ გაგებით $M$-ის წერტილებს შორის გაბნევის მანძილი ზომავს **ბრაუნის გზებს** t დროში:

$$\mathcal{D}_t^2(x,y) = \int_{\mathcal{M}} \left(\mathcal{K}_t(x,y) - \mathcal{K}_t(y,z)\right)^2 dz$$

**ფორმულა 1.2.2.2**

**განსაზღვრება 1.2.2.4 ბრაუნის გზები.** ეს არის სითხეში ან აირში მოქცეული ნაწილაკების შემთხვევითი მოძრაობა, რომელიც გამოწვეულია სწრაფად მოძრავ ატომებსა და მოლეკულებთან შეჯახებით.

გასაშუალების სქემა, რომელიც წარმოადგინა გაბნევის მანძილმა, გვაძლევს სასურველ მახასიათებლებს სეგმენტაციის პრობლემის გადასაწყვეტად. მიღებული აზომვები მდგრადია ზედაპირის ხარვეზებისა და ტოპოლოგიური დეფექტების მიმართ. გეოდეზური მანძილებისგან განსხვავებით, გაბნევის მანძილი მკვეთრად იზრდება ობიექტის შევიწროვებულ უბნებში და ამის შემდეგ გამოყოფს $M$-ის ამოზნექილ უბნებს (სურათი 1.2.2.1)



**სურათი 1.2.2.1**

სითბური ბირთვი $\mathcal{K}_t(.,.)$ განისაზღვრება **ლაპლასის ოპერატორის საკუთრივი მნიშვნელობებით(eigenvalue) და საკუთრივი ფუნქციებით(eigenfunction)**

$$\mathcal{K}_t\ (x,y) = \sum_{k=0}^{\infty} e^{-\lambda_k t}\, \phi^k(x)\phi^k(y)$$

**ფორმულა 1.2.2.3**

**ლაპლასის ოპერატორი**

მათემატიკაში **ლაპლასის ოპერატორი,** იგივე **ლაპლასიანი,** არის დიფერენციალური ოპერატორი, რომელიც მოქმედებს ევკლიდურ სივრცეში ან მის ღია არეებზე განმარტებულ ფუნქციებზე და განისაზღვრება როგორც გრადიენტის დივერგენცია. **ლაპლასიანი** აღინიშნება $\nabla \cdot \nabla$, $\nabla^2$, $\Delta$ სიმბოლოებით. ჩვენ ვიყენებთ **დისკრეტულ ლაპლასის ოპერატორს,** რომელიც **ლაპლასის ოპერატორის** ანალოგიურია და განიმარტება დისკრეტულ სტრუქტურებზე, როგორიცაა გრაფები და დისკრეტული ბადეები. ზოგადად, **ლაპლასის ოპერატორი** თავს იჩენს დიფერენციალურ გამოსახულებებში, რომლებიც აღწერენ სხვადასხვა ფიზიკურ ფენომენებს, მაგალითად: ელექტრონულ და გრავიტაციულ პოტენციალებს, სითბურ და სითხის ნაკადის გაბნევას, ტალღების გავრცელებასა და კვანტურ მექანიკას.

**განსაზღვრება 1.2.5 საკუთრივი მნიშვნელობა, საკუთრივი ვექტორი, საკუთრივი ფუნქცია.** ვთვათ, მოცემული გვაქვს წრფივი ოპერატორი წრფივ სივრცეში, ან შესაბამისი კვადრატული მატრიცა A, მაშინ მისი **საკუთრივი ვექტორი** განიმარტება, როგორც ისეთი არანულოვანი ვექტორი $v$, რომლისთვისაც $Av = \lambda v$ რაიმე $\lambda$ სკალარისათვის. ამ შემთხვევაში $\lambda$-ს ეწოდება $A$ მატრიცის $v$ ვექტორის შესაბამისი **საკუთრივი რიცხვი** ან **საკუთრივი მნიშვნელობა.** ხოლო თუ ოპერატორი მოქმედებს ფუნქციათა სივრცეზე, მაშინ საკუთრივი ვექტორების ნაცვლად გვექნება საკუთრივი ფუნქციები.

ვთქვათ, $\mathcal{D}$ არის რაიმე დიფერენციალური ოპერატორი, რომელიც განმარტებულია უსასრულოდ დიფერენცირებად, ნამდვილ მნიშვნელობიან ფუნქციათა $C^\infty$ სივრცეზე.

**საკუთრივი ფუნქცია** ეწოდება ისეთ არანულოვან $f$ ფუნქციას, რომელიც აკმაყოფილებს შემდეგ პირობას:

$$\mathcal{D}f = \lambda f.$$

ზედაპირებზე(ან მაღალი განზომილების მრავალსახეობებზე) ხდება ლაპლასის ოპერატორის განზოგადება **ლაპლას-ბელტრამის ოპერატორით(განსაზღვრება 1.2.2.6)**. მისი საკუთრივი მნიშვნელობები ქმნიან ზრდად მიმდევრობას $0 = \lambda_0 < \lambda_1 \leq \cdots \uparrow \infty$ და მისი საკუთრივი ფუნქციები $\phi^k$ იქნება ზრდადი რხევების მქონე ფუნქციები $M$-ზე. თითოეული $(\lambda_k, \phi^k)$ წყვილი შეესაბამება **ჰელმჰოლცის გამოსახულების(განსაზღვრება 1.2.2.7)** ამონახსნს:

$$\Delta\phi^k = -\lambda_k \phi^k$$

**ფორმულა 1.2.2.4**

**განსაზღვრება 1.2.2.6 ლაპლას-ბელტრამის ოპერატორი.** დიფერენციალურ გეომეტრიაში **ლაპლასის ოპერატორის** განზოგადება გამოიყენება ევკლიდეს სივრცეში არსებულ ზედაპირებზე განსაზღვრულ ფუნქციებზე ოპერირებისას, უფრო ზოგადად კი **რიმანის (Riemannian)** და **ფსევდო-რიმანის (pseudo-Riemannian)** მრავალსახეობებზე. ამ განზოგადებას ეწოდება **ლაპლას-ბელტრამის ოპერატორი.** ლაპლასის ოპერატორის მსგავსად, **ლაპლას-ბელტრამის ოპერატორიც** განისაზღვრება როგორც გრადიენტის დივერგენცია და არის წრფივი ოპერატორი, რომელიც ფუნქციას ასახავს ფუნქციაში.

**განსაზღვრება 1.2.2.7 ჰელმჰოლცის განტოლება(Helmholtz equation)** არის კერძოწარმოებულიანი დიფერენციალური განტოლება:

$$\nabla^2 A + k^2 A = 0$$

სადაც $\nabla^2$ არის **ლაპლასიანი,** $k$ არის ტალღის სიხშირე და $A$ არის ამპლიტუდა.

ამიტომ შეგვიძლია შევცვალოთ ფორმულა 1.2.2.3, ფორმულით 1.2.2.2 და გადავწეროთ გაბნევის მანძილი **გაბნევის ასახვის** ტერმინებში:

$$\Phi_t(x) = \left(\cdots e^{-\lambda_k t}\phi^k(x)\cdots\right)^T$$

**ფორმულა 1.2.2.5**

ამის შემდეგ გაბნევის მანძილი გადაიქცევა ორ უსასრულო განზომილებიან ვექტორებს შორის ევკლიდურ მანძილად:

$$\mathcal{D}_t^2(x,y) = \| \Phi_t(x) - \Phi_t(y) \|^2 = \sum_{k=1}^{\infty} e^{-2\lambda_k t}\left(\phi^k(x) - \phi^k(y)\right)^2$$

**ფორმულა 1.2.2.6**

რადგან საკუთრივი მნიშვნელობები $\{\lambda_k\}$ სწრაფად იზრდება, რაღაც მომენტიდან დაწყებული ჯამის შესაკრებები სწრაფად უახლოვდება ნულს და ამიტომ, ჩვენ შეგვიძლია უგულებელვყოთ მწკრივის კუდი და უსასრულო ჯამს მივუახლოვდეთ სასრული ჯამით. მაშინ $\Phi_t(.)$-ში ვიგულისხმოთ სასრულ განზომილებიანი ვექტორი, რომელიც განიმარტება უტოლობით $e^{-\lambda_k t} > \delta$ ვექტორის კომპონენტების მამრავლების მიმართ, სადაც $\delta = 0.1$.

სურათი 1.2.2.2 სეგმენტაციის ალგორითმის მიღებული შედეგები

### 1.2.3 შუალედური სტრუქტურები

ახლა ჩვენ განვსაზღვრავთ უბნისათვის მნიშვნელოვან კომპონენტებს $\mathcal{R} \subset \mathcal{M}$ . ამ პრობლემის გადასაწყვეტად გამოვიყენებთ გაბნევის მანძილს, რომ მივაღწიოთ ბიექციას $R$ სეგმენტებსა და შუალედურ სტრუქტურებს შორის. ამ ბიექციის დასამყარებლად ჯერ გამოვითვლით შუალედურ სტრუქრურებს $R$ სეგმენტების გამოყენებით, ხოლო შემდგომ კი პირიქით - გამოვითვლით სეგმენტებს შუალედური სტრუქტურებიდან.

აღვნიშნოთ $t_{\mathcal{R}}$ როგორც გაბნევის მანძილის დროითი ზრდა, რომელიც საკმარისია $R$-ის გლობალური ჩაზნექილობის თვისებების გამოსავლენად.

#### სეგმენტები → შუალედური სტრუქტურები

გვაქვს სეგმენტი $\mathcal{R}_i \subset \mathcal{R}$ რომელსაც ვახასიათებთ მისი შუალედური სტრუქტურით. ჩვენ განვსაზღვრავთ შუალედურ სტრუქტურას, როგორც სეგმენტის ჩაზნექილი უბნებიდან ყველაზე დაშორებულ წერტილებს. უნდა აღინიშნოს, რომ შუალედური

სტრუქტურა განსხვავდება შუალედური ღერძისგან, რადგან ეს უკანასკნელი შეუალედურია სეგმენტების მიმართ.

$\mathcal{R}_i$-ის შუალედური სტრუქტურის გამოსათვლელად შემოვიღებთ გასაშუალებული გაბნევის მაძილის(Average Diffusion Distance ADD) ფუნქციას:

$$ADD(x) = \frac{1}{A_{\mathcal{R}_i}} \int_{\mathcal{R}_i} \mathcal{D}^2_{t_{\mathcal{R}}}(x, y)\, dy$$

**ფორმულა 1.2.3.1**

$ADD$ წარმოადგენს საზღვრების დიდ მნიშვნელობებს, ექსტრემუმის წერტილებს და მცირე მნიშვნელობებს სეგმენტის ცენტრში მდებარე წერტილებისათვის. რადგან $ADD$ მემკვიდრეობით იღებს გაბნევის მანძილის თვისებებს, ამიტომ მცირე $ADD$ მნიშვნელობის მქონე წერტილები შეესაბამება ამოზნექილობებიდან დაშორებულ წერტილებს(სურათი 1.2.3.1 ა).

**სურათი 1.2.3.1**

**ა) $ADD$ ბ) ინიციალიზაცია გ) სეგმენტაცია**

განვსაზღვრავთ $\mathcal{S}_i$ შუალედურ სტრუქტურას $\mathcal{R}_i$ სეგმენტისათვის :

$$\mathcal{S}_i = \left\{x \in \mathcal{R}_i : ADD(x) = \min_{y \in \mathcal{R}_i} ADD(y)\right\}$$

**ფორმულა 1.2.3.2**

**შუალედური სტრუქტურები $->$ სეგმენტები**

ვიღებთ შუალედურ სტრუქტურებს $\{\mathcal{S}_i\}$ $\mathcal{R}$-დან. $\{\mathcal{R}_i\}$ სეგმენტების აღსადგენად ვიყენებთ კონკურენტულ ფრონტებს. თითოეული ფრონტი $\mathcal{F}_i$ გამოითვლება $\mathcal{S}_i$-ზე დამოკიდებული გაბნევის მანძილის ზრდადი იზო მნიშვნელობით *d:*

$$\mathcal{F}_i(d) = \left\{x \in \mathcal{R}_i : \min_{y \in \mathcal{S}_i} \mathcal{D}^2_{t_{\mathcal{R}}}(x, y) = d\right\}$$

**ფორმულა 1.2.3.3**

შეჯიბრი იწყება შუალედური სტრუქტურიდან $\mathcal{F}_i(0)$ და მთავრდება, როდესაც ყველა ფრონტი ჩამოიშლება.

პროცესის დროს თითოეულ $\mathcal{R}_i$ სეგმენტს მოვნიშნავთ, როგორც უბანს, რომელიც დაფარულია $\mathcal{F}_i$ ფრონტით. რადგან ამოზნექილი უბნები $t_{\mathcal{R}}$-ზე დაშორებულია შეალედური სტრუქტურებიდან, ამიტომ სეგმენტების საზღვრები მოექცევა ამოზნექილ უბენბზე. სურათი 1.2.2.1 გვიჩვენებს ფრონტებს სხვადასხვა შუალედური სტრუქტურებისათვის. ვხედავთ, რომ იზო მნიშვნელობა $d$ მკვეთრად იზრდება მოდელის მნიშვნელოვან ამოზნექილ უბნებთან.

## 1.2.4 იერარქიული სეგმენტაციის ალგორითმი

ალგორითმი მუშაობს ტრიანგულირებულ ბადეებზე $M = (V, E, F)$, თუმცა ჩვენ შეგვიძლია მისი გაფართოება ზედაპირის ასახვის სხვადასხვა წარმოდგენებისათვის.

ტრიანგულირებულ ბადეში $M = (V, E, F)$ ვახდენთ $f$ ფუნქციის დისკრეტიზაციას $\bar{f}$, სადაც $f_i$ არის $f$-ის მნიშვნელობა $i \in V$ წვეროზე. სამკუთხედების შიგნით არსებული წერტილებისათვის ვახდენთ $f$-ის მიახლოებას წრფივი კოორდინატებით და ბარიცენტრული კოორდინატებით.

ფრონტების კონკურირებისათვის ჩვენ ვქმნით გროვას $H$, რომელიც შეიცავს წყვილებს $(f, i)$, სადაც $f$ არის კანდიდატი წახნაგი, რომელიც უნდა დაიფაროს ფრონტით $\mathcal{F}_i$. გროვაში წყვილები დალაგებულია მაქსიმალური მანძილის მიხედვით $f$-ის და შუალედურ სტრუქტურა $\mathcal{S}_i$-ს შორის.

$$\max_{v \in f} \left( \min_{u \in \mathcal{S}_i} \mathcal{D}^2_{t_{\mathcal{R}}}(v, u) \right)$$

**ფორმულა 1.2.4**

ვახდენთ $H$-ის ინიციალიზაციას იმ წახნაგებით, რომლებიც დაკავშირებულია მიმდინარე შუალედური სტრუქტურის $\{\mathcal{S}_i\}$ წერტილებთან. თითოეულ ბიჯზე ჩვენ ვშლით ისეთ წყვილს $(f, i)$, რომელსაც აქვს ყველაზე მცირე მანძილი. თუ $f$ არ არის არცერთ სეგმენტში, მაშინ ვამატებთ $\mathcal{R}_i$-ში. შემდეგ ვანახლებთ $H$-ს $f$-ის მეზობელი წახნაგებით $R$-დან. შეჯიბრი მთავრება როდესაც გროვა ცარიელია.

იმისათვის, რომ განვაახლოთ შუალედური სტრუქტურა, თითოეული $\mathcal{R}_i$ სეგმენტისათვის ჩვენ გამოვითვლით $ADD$-ს მინიმალური მნიშვნელობის მქონე წვეროებს(ფორმულა 1.2.4).

$$ADD(x) = \| \Phi_t(x) \|^2 - 2 \left\langle \Phi_t(x), \frac{\int_{\mathcal{R}} \Phi_t(y)\, dy}{A_{\mathcal{R}}} \right\rangle + \frac{\int_{\mathcal{R}} \| \Phi_t(y) \|^2\, dy}{A_{\mathcal{R}}}$$
$$\simeq \| \Phi_t(x) \|^2 - 2 \left\langle \Phi_t(x), \frac{\sum_{v \in \mathcal{R}} \Phi_t(v) A_v}{A_{\mathcal{R}}} \right\rangle + Q_{\mathcal{R}}$$

**ფორმულა 1.2.4.1**

ბადის დიდი რეზოლუციის გამო შესაძლებელია მოხდეს შუალედური სტრუქტურების დანაწილება. ამ შემთხვევას თავიდან ვიცილებთ $\varepsilon$ სიზუსტის შემოღებით $ADD$-ის მინიმალურ მნიშვნელობაზე. ეს სიზუსტე უნდა იყოს ადაპტირებული $\mathcal{R}_i$-ის ფორმაზე და უნდა იყოს ახლოს 0-თან. ჩვენს შემთხვევაში $\varepsilon$ = 0.01 * $avgADD$, სადაც $avgADD$ არის $ADD$-ს მნიშვნელობების გასაშუალება(ფორმულა 1.2.4.1).

$$\overline{ADD} = \frac{1}{A_{\mathcal{R}}} \int_{\mathcal{R}} ADD(x)\, dx \simeq 2\left(\mathcal{Q}_{\mathcal{R}} - \| \frac{\sum_{v \in \mathcal{R}} \Phi_t(v) A_v}{A_{\mathcal{R}}} \|^2\right)$$

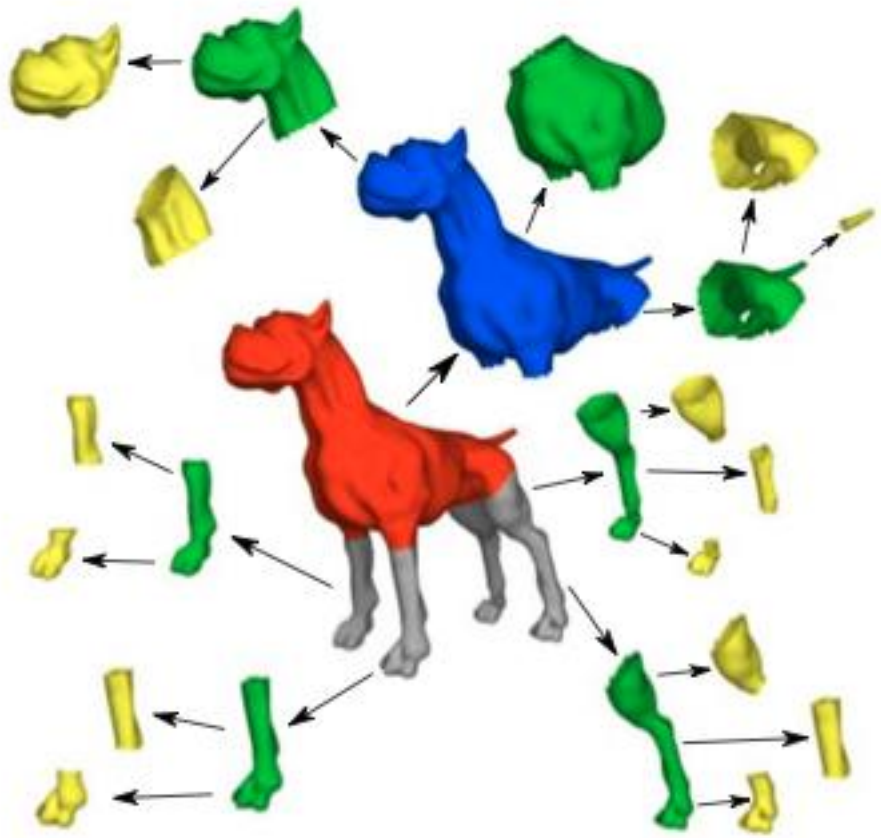
**ფორმულა 1.2.4.2 $avgADD$**

ალგორითმი მუშაობს რეკურსიულად. საწყისი მნიშვნელობა არის $R = M$-ს. შედეგად მიღებული სეგმენტები ქმნიან იერარქიულ ხეს. ეს იერარქია გამოავლენს სეგმენტების იდეალურთან მიახლოებულ განლაგებას და დამოკიდებულებების ხეს. საბოლოო სეგმენტაცია შეგვიძლია მოვჭრათ ხიდან(სურათი 1.2.4)

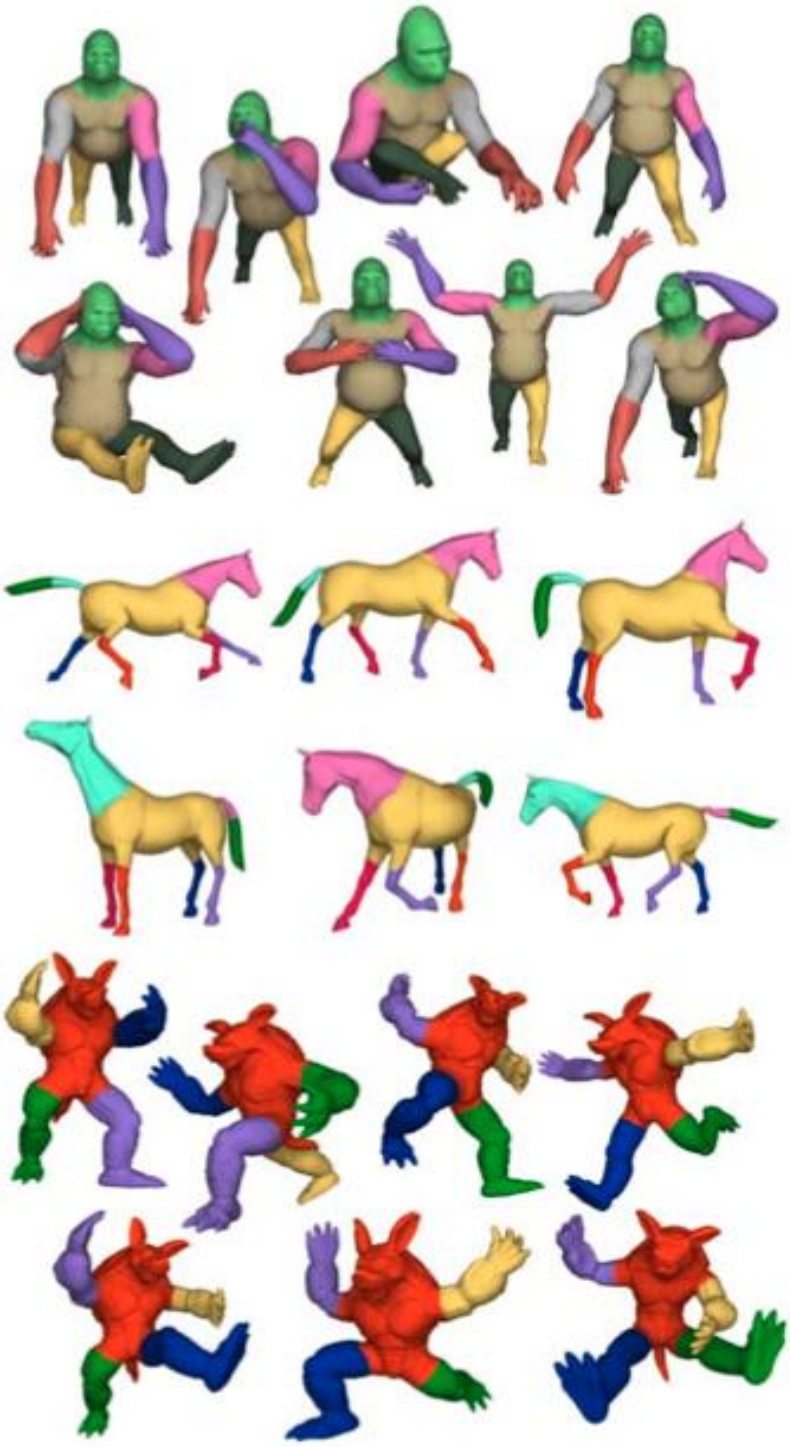
სურათი 1.2.4 იერარქიის სხვადასხვა ჭრილები

## 1.2.5 შედეგები

სურათი 1.2.5-ზე ნაჩვენებია სეგმენტების იერარქიის მაგალითი. მწვანე ფერით აღნიშნულია ფუნქციონალური უბნები. ხოლო ყვითელი ფერით აღნიშნულია მყარი კომპონენტები. სეგმენტები გამოავლენენ მოდელის სტრუქტურას, რაც საშუალებას იძლევა მივიღოთ დეტალიზაციის დონის კომბინაციები.

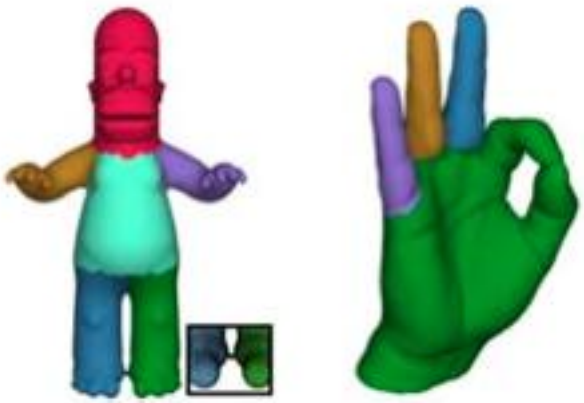


სურათი 1.2.5 ძაღლის მოდელის სეგმენტების იერარქიული ხე



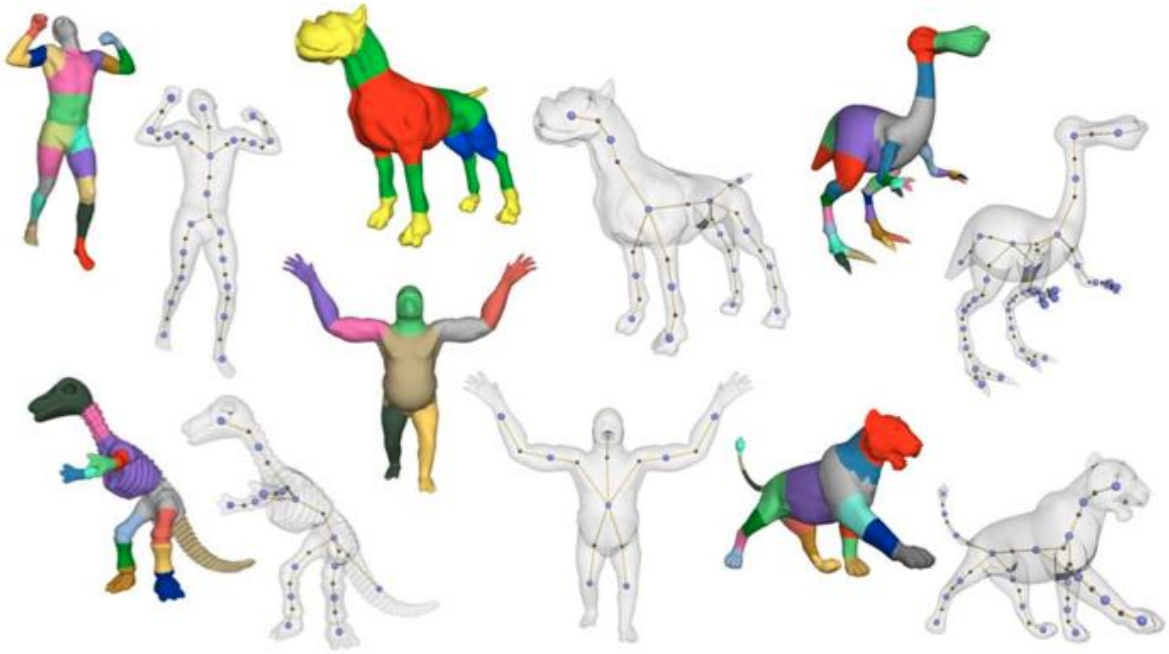
სურათი 1.2.5.1. იერარქიული მეთოდის შედეგების სიმკვრივე სხვადასხვა დეფორმირებული მოდელების მაგალითზე.

სურათი 1.2.5.2 მეთოდი მდგრადია  ზედაპირის ხარვეზიანობის მიმართ



ა) ბ)

სურათი 1.2.5.3 ა) მეთოდი მდგრადია მცირე ტოპოლოგიური ხარვეზების მიმართ. ბ) თუმცა, როგორც ვხედავთ არსებობს არასწორი შედეგის ალბათობაც, როდესაც ტოპოლოგიის დარღვევა უფრო დიდ მაშტაბებს მოიცავს.



სურათი 1.2.5.4 სეგმენტებიდან მოპოვებული ჩონჩხი. ლურჯი სფეროები მიუთითებენ სეგმენტების სიმძიმის ცენტრებს და ნაცრისფერი სფეროები მიუთითებენ პერსონაჟის სახსრების ცენტრალურ წერტილებს.

# 2. სითბურ ცენტრებზე დაფუძნებული სეგმენტაცია

სითბურ ცენტრებზე დაფუძნებული სეგმენტაცია შეადგენს **სამგანზომილებიანი ობიექტის სითბური სეგმენტაციის(განსაზღვრება 2.0.1)** ალგორითმის ნაწილს.

**განსაზღვრება 2.0.1 სამგანზომილებიანი ობიექტის სითბური სეგმენტაცია** იყენებს სამგანზომილებიანი ობიექტის ზედაპირის თბოგამტარობის თვისებებს და შესაბამისი ანალიზის გამოყენებით ახდენს ობიექტის მნიშვნელოვანი და ფუნქციონალური კომპონენტების გამოყოფას. პროცესი შედგება ორი ნაწილისგან:

1. მოდელის წინასწარი ანალიზი, სითბური ცენტრების რაოდენობის პოვნა და მათი იდენტიფიცირება.
2. მიღებული ინფორმაციის გამოყენებით მოდელის უშუალო სეგმენტაცია **HC-Kmeans** ალგორითმით(პარაგრაფი 2.2).

ამ ნაშრომში განვიხილავთ უშუალოდ სეგმენტაციის ალგორითმს და მის იმპლემენტაციას.

## 2.1 სითბური ცენტრების პოვნა და იდენტიფიცირება

ჩვენთვის სასურველი სითბური ცენტრების პოვნა და იდენტიფიცირება წინა სექციაში აღწერილი მეთოდის მსგავსია. ის აკმაყოფილებს შემდეგ პირობებს

- არ არის დამოკიდებული იზომეტრიული დეფორმაციების მიმართ.
- მდგრადია ტოპოლოგიური დარღვევების მიმართ.
- მდგრადია მოდელში თანდაყოლილი ხარვეზების მიმართ.
- გვთავაზობს ადამიანის აღქმასთან მიაზლოებულ სეგმენტაციას

და დაფუძნებულია ლაპლასის ოპერატორის თვისებებზე.

რადგან ამ ნაშრომის თემას წარმოადგენს უშუალოდ სითბური ცენტრებით მართვადი სეგმენტაცია ამიტომ გადავიდეთ მის განხილვაზე.

## 2.2 სითბური ცენტრებით მართვადი სეგმენტაცია

ბადის სეგმენტაიციას შეგვიძლია მივუდგეთ, როგორც მსგავსი წერტილების გეომეტრიულად გააზრებულ ჭდეებში დაჯგუფების პროცესს. გასული ათწლეულის განმავლობაში კლასტერიზაციის ბევრი ტექნიკა იქნა შემუშავებული. ჩვენ გამოვიყენებთ ფართოდ გავრცელებულ კლასტერიზაციის ალგორითმს : Kmeans. ამ ალგორითმს გააჩნია

ორი ცნობილი შეზღუდვა. ის დამოკიდებულია კლასტერების რაოდენობაზე და კლასტერის საწყისი მნიშვნელობის ინიციალიზაციაზე. ამ პრობლემების გადასაჭრელად შემოვიღებთ **სითბური ცენტრებით მართვად Kmeans(HC-Kmeans)-ს,** რომელიც გვაძლევს კლასტერიზაციის სტაბილურ და ეფექტურ პროცედურას.

სითბური ცენტრებით მართვადი სეგმენტაციის ფსევდოკოდი:

1. ალგორითმს გადაეცემა $M$-ის თითოეული წვეროს სითბური კოორდინატი, სითბური ცენტრების რაოდენობა, წვეროების საერთო რაოდენობა $N$ სითბური ცენტრების სიმრავლე $G$
2. ვთქვათ $Cor_{g_i}$ არის $g_i$ სითბური ცენტრის კოორდინატი $G$-ში.
3. --> მოვახდინოთ სითბური ცენტრების ინიციალიზაცია.
4. $Cen(i) = Cor_{g_i}$
5. --> შევასრულოთ Kmeans და გადავცეთ კლასტერების რაოდენობა $C$, სითბური კოორდინატები $Cor$ სითბური ცენტრები $Cen$. გამომავალი $Label$ მიუთითებს თითოეული წვეროს კლასტერის წევრობას.
6. $Label = Kmeans(C, Cor, Cen)$
7. ავსახოთ $Label$ თითოეულ წვეროზე $M$-ში.
8. return $Label$

HC-Kmeans დააბრუნებს უნიკალურ ნიშანს თითოეული წვეროსათვის ისე, რომ შესაძლებელი იქნება მატი დაჯგუფება სხვადასხვა დანაყოფებში, რითაც გამოვლინდება მოდელის სეგმენტაცია.

```
// Cor(i) for the further implementation of Heat Driven Segmentation
QVector<float> Model::calcCor(int i)
{
	QVector<float> tempVect;

	tempVect.resize(i);

	for ( int j = 0; j < i; j++)
	{
		tempVect[j] = exp(-m_eigenVal(i).real()) * m_eigenVect(j,i).real();
	}

	return tempVect;
}
```

**სურათი 2.2.1 მეთოდი რომელიც ითვლის სითბურ კოორდინატებს თითოეული სითბური ცენტრისათვის**

## 2.3 სითბური ცენტრებით მართვადი სეგმენტაციის იმპლემენტაცია

ნაშრომის მთავარი მიზანია დაგროვილი თეორიული ცოდნის პრაქტიკული იმპლემენტაცია და რეალურ პირობებში გამოყენება. რადგან მეთოდები ექცევა დაბალი დონის პროგრამირების აპლიკაციებში ამიტომ მისი ფუნქციონალის იმპლემენტაციისთვის ვიყენებთ შემდეგ პროგრამულ ენებს და ხელსაწყოებს: C, C++, Qt, OpenGL, VisualStudio 2010.

- C და C++ შესაბამისად გამოიყენება აპლიკაციის გამოთვლითი ნაწილის იმპლემენტაციისათვის.
- Qt-ს ბიბლიოთეკები გამოყენება მომხმარებლის ინტერფეისის იმპლემენტაციისათვის.
- OpenGL-ის ბიბლიოთეკები გამოყენება სამგანზომილებიანი ობიექტების და მათზე ჩატარებული ოპერაციების ვიზუალიზაციისათვის.

იმპლემენტაციის ზემოთ ხსენებული მახასიათბლები მულტიპლატფორმულია და ადვილად შეიძლება აპლიკაციის პორტირება სხვა ოპერაციულ სისტემებზე.

## 2.4 პროგრამის მუშაობის აღწერა

პროგრამის მუშაობა იწყება მთავარი ფანჯრის გამოტანით, რომელიც შეიცავს OpenGL-ის ცარიელ სცენას და აპლიკაციის სამომხმარებლო ინტერფეისის კომპონენტებს (სურათი 2.4.1.2). მომხმარებელს შეუძლია ჩატვირთოს მოდელი პროგრამაში, შეცვალოს მოდელისა და განათების ფერები, შეცვალოს მოდელის ასახვის მეთოდები, გაეცნოს ჩატვირთული მოდელის შესახებ სტატისტიკას და რაც მთავარია ჩაატაროს მოდელის სეგმენტაცია.

### 2.4.1 სამგანზომილებიანი ობიექტის იმპორტი და ექსპორტი

აპლიკაციის სტრუქტურაში გათვალისწინებულია მოქნილობა და უკვე არსებულ ტექნოლოგიებთან მაქსიმალურად თავსებადობა. სამგანზომილებიანი მოდელებისთვის არსებობს პოპულარული ფორმატები, როგორებიცაა მაგალითად: .OBJ, .3DM, .3DS, .MAX, .FBX. აქედან ყველაზე მარტივად გამოყენებადი არის .OBJ, თუმცა .FBX-ს ფორმატის მხარდაჭერა აქვს ყველა წამყვან გრაფიკულ და კინემატოგრაფიულ პაკეტებს, აგრეთვე ამ ფორმატს იყენებენ კომპიუტერული თამაშების ძრავები(რაც ჩვენთვის უფრო პრიორიტეტულია) როგორებიცაა UDK/UE3(Unreal Development Kit/Unreal Engine3) და CryEngine.

პროგრამა მუშაობს .OBJ ფაილებთან (ფინალურ ვერსიაში იქნება .FBX ფორმატის სრული მხარდაჭერა). OBJ ფაილის აპლიკაციაში შეტანის იმპლემენტაცია ნაჩვენებია სურათი 2.4.1.1-ზე. OBJ ფაილი არის ტექსტური ფაილის ნაირსახეობა შესაბამისად მოდელის შესახებ ინფორმაციის წაკითხვა ხდება Qt-ს ტექსტური და ფაილური ხელსაწყოების გამოყენებით.

OBJ ფაილის სტრუქტურა:

```
#
# object FILE_NAME
#

v  0.0095 -0.1224 0.0086
v  0.0141 -0.1221 0.0135
v  0.0150 -0.1197 0.0281
v  0.0153 -0.1098 0.0245
...
vn -0.7091 0.5428 -0.4501
vn -0.7122 0.4811 -0.5112
vn -0.7122 0.4811 -0.5112
vn -0.6761 0.5145 -0.5274
...
f 680//3400 491//3401 490//3402
f 490//3403 533//3404 680//3405
f 617//3406 680//3407 533//3408
f 533//3409 532//3410 617//3411
...
# NUMBER_OF faces
```

სადაც FILE_NAME არის ობიექტის სახელი, *v* აღნიშნავს წვეროს და მოსდევს მისი კოორდინატები, *vn* აღნიშნავს წვეროს ნორმალს, *f* აღნიშნავს წახნაგს და მოსდევს იმ წვეროების ინდექსები, რომლებსაც აკავშირებს ერთმანეთთან, NUMBER_OF იქნება ფაილის საერთო წახნაგების რაოდენობა.

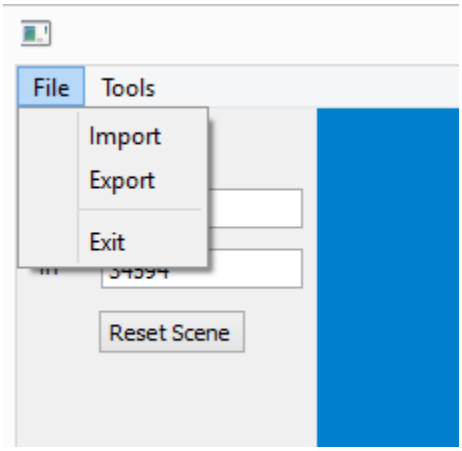
### 2.4.2 სამგანზომილებიანი ობიექტის დამუშავება

აპლიკაციას შეტანილ მოდელი უნდა იყოს წინასწარ დამუშავებული ანუ უნდა გააჩნდეს სრული ინფორამცია სითბური ცენტრების შესახებ. ამ ინფორმაციის გამოყენებით მოხდება სეგმენტაციის ალგორითმის შესრულება და საბოლოო შედეგის მიღება.

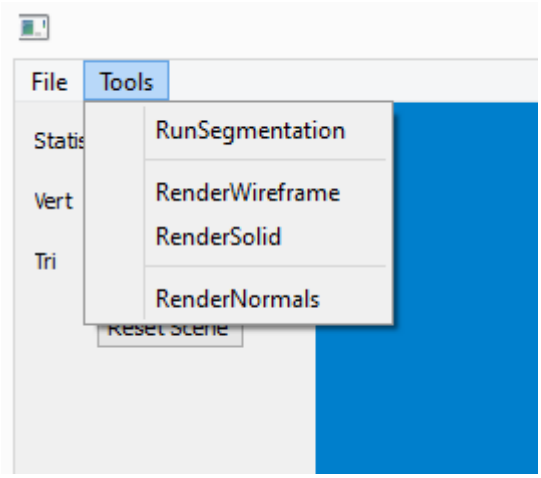
```
void Model::createModel(const QString &filePath)
{
	QFile file(filePath);
	if (!file.open(QIODevice::ReadOnly))
		return;

	Point3d boundsMin( 1e9, 1e9, 1e9);
	Point3d boundsMax(-1e9,-1e9,-1e9);

	QTextStream in(&file);
	while (!in.atEnd())
	{
		QString input = in.readLine();
		if (input.isEmpty() || input[0] == '#')
			continue;

		QTextStream ts(&input);
		QString id;
		ts >> id;
		if (id == "v")
		{
			Point3d p;
			for (int i = 0; i < 3; ++i)
			{
				ts >> p[i];
				boundsMin[i] = qMin(boundsMin[i], p[i]);
				boundsMax[i] = qMax(boundsMax[i], p[i]);
			}
			m_points << p;
		}
		else if (id == "f" || id == "fo")
		{
			QVarLengthArray<int, 4> p;

			while (!ts.atEnd())
			{
				QString vertex;
				ts >> vertex;
				const int vertexIndex = vertex.split('/').value(0).toInt();
				if (vertexIndex)
					p.append(vertexIndex > 0 ? vertexIndex - 1 :
m_points.size() + vertexIndex);
			}

			for (int i = 0; i < p.size(); ++i)
			{
				const int edgeA = p[i];
				const int edgeB = p[(i + 1) % p.size()];

				if (edgeA < edgeB)
					m_edgeIndices << edgeA << edgeB;
			}

			for (int i = 0; i < 3; ++i)
				m_pointIndices << p[i];

			if (p.size() == 4)
				for (int i = 0; i < 3; ++i)
					m_pointIndices << p[(i + 2) % 4];
```

```
}
	}

	const Point3d bounds = boundsMax - boundsMin;
	const qreal scale = 1 / qMax(bounds.x, qMax(bounds.y, bounds.z));
	for (int i = 0; i < m_points.size(); ++i)
		m_points[i] = (m_points[i] - (boundsMin + bounds * 0.5)) * scale;

	m_normals.resize(m_points.size());
	for (int i = 0; i < m_pointIndices.size(); i += 3)
	{
		const Point3d a = m_points.at(m_pointIndices.at(i));
		const Point3d b = m_points.at(m_pointIndices.at(i+1));
		const Point3d c = m_points.at(m_pointIndices.at(i+2));

		const Point3d normal = cross(b - a, c - a).normalize();

		for (int j = 0; j < 3; ++j)
			m_normals[m_pointIndices.at(i + j)] += normal;
	}

	for (int i = 0; i < m_normals.size(); ++i)
		m_normals[i] = m_normals[i].normalize();

	m_enableSegment = false;
}
```

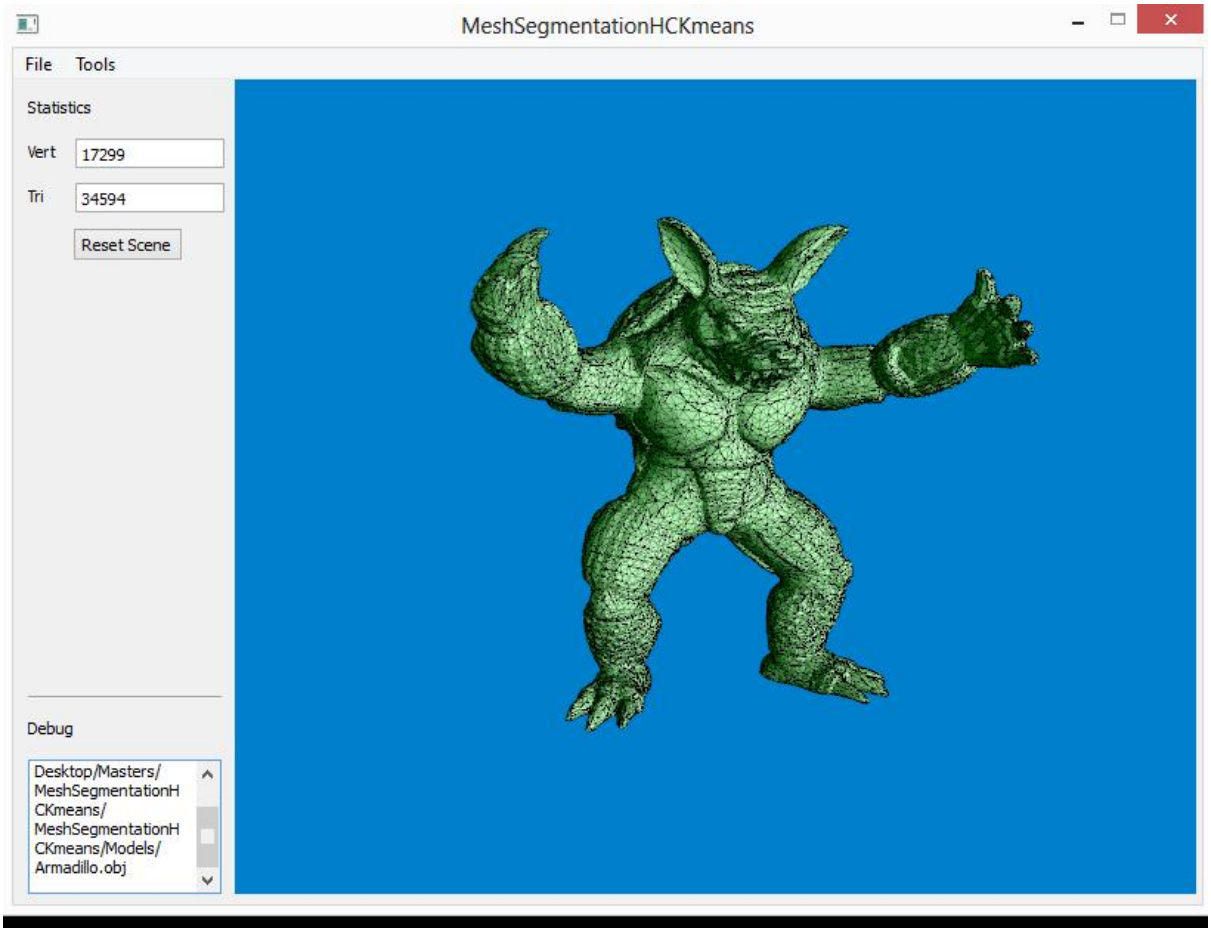
სურათი 2.4.1.1 მეთოდი რომელიც ტვირთავს .OBJ ფაილს პროგრამაში



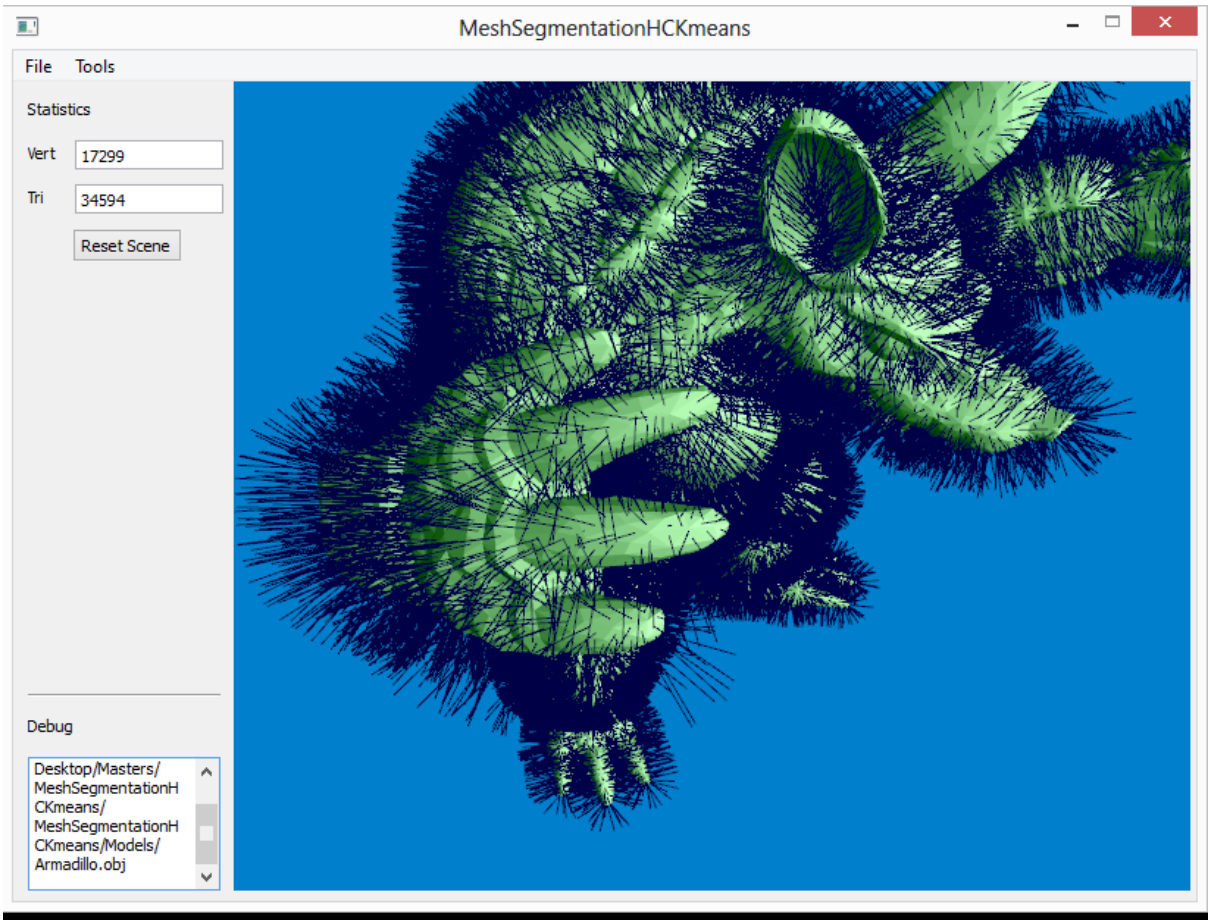

სურათი 2.4.1.2 ფაილის მენიუ



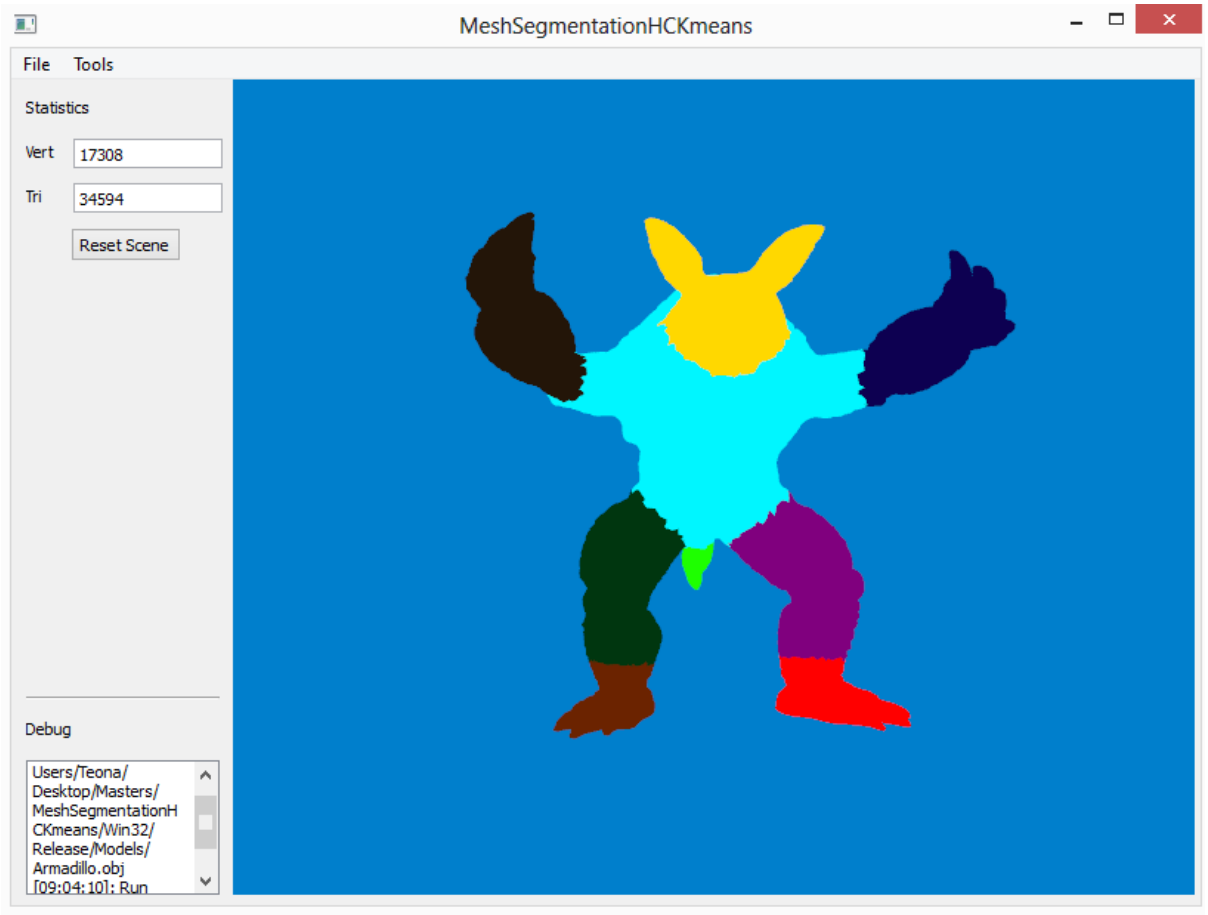

სურათი 2.4.1.3 ხელსაწყოების მენიუ



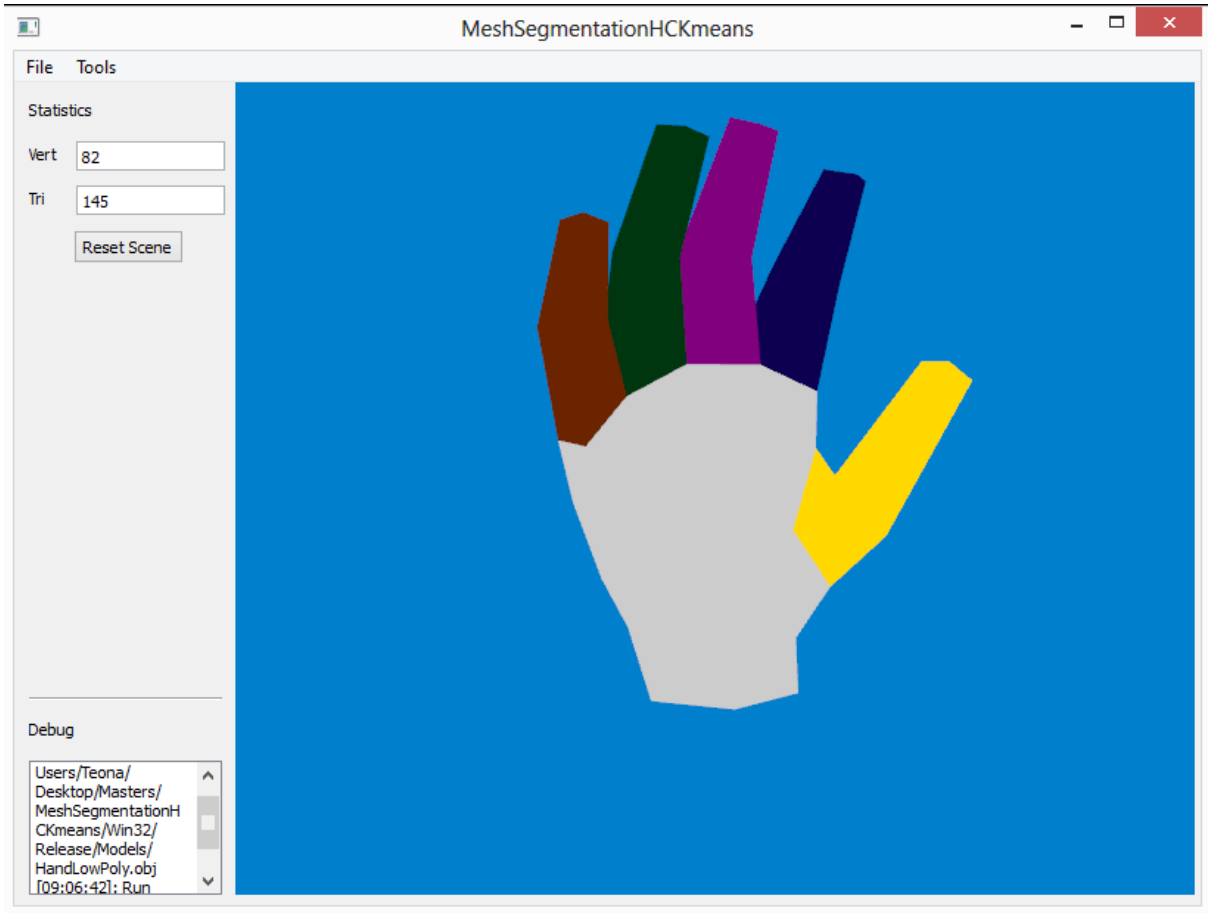

სურათი 2.4.1.4 მოდელის wireframe ვიზუალიზაცია




სურათი 2.4.1.5 მოდელის წვეროების ნორმალების ვიზუალიზაცია



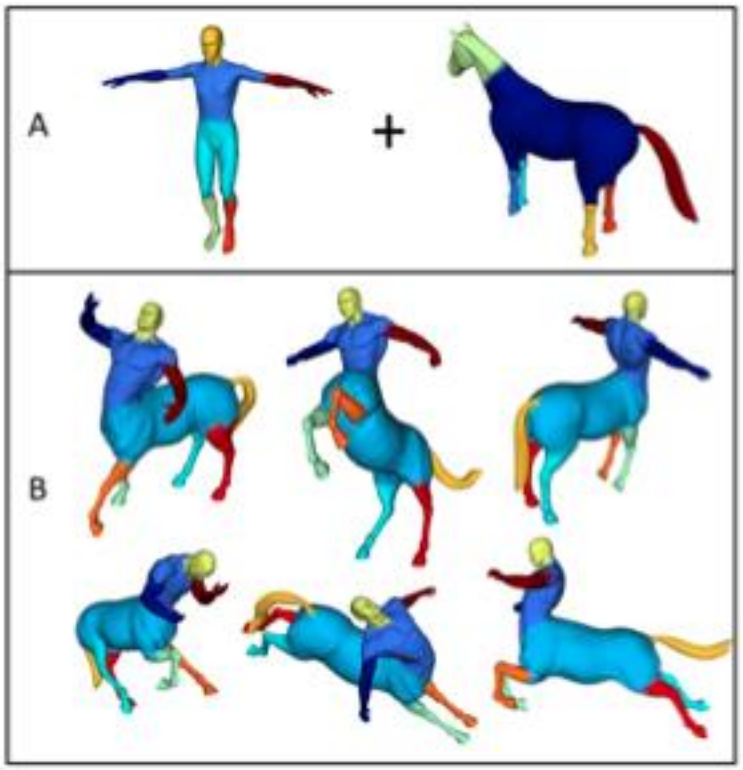

სურათი 2.4.1.6 სეგმენტაციის შედეგები Armadillo-ს მოდელზე



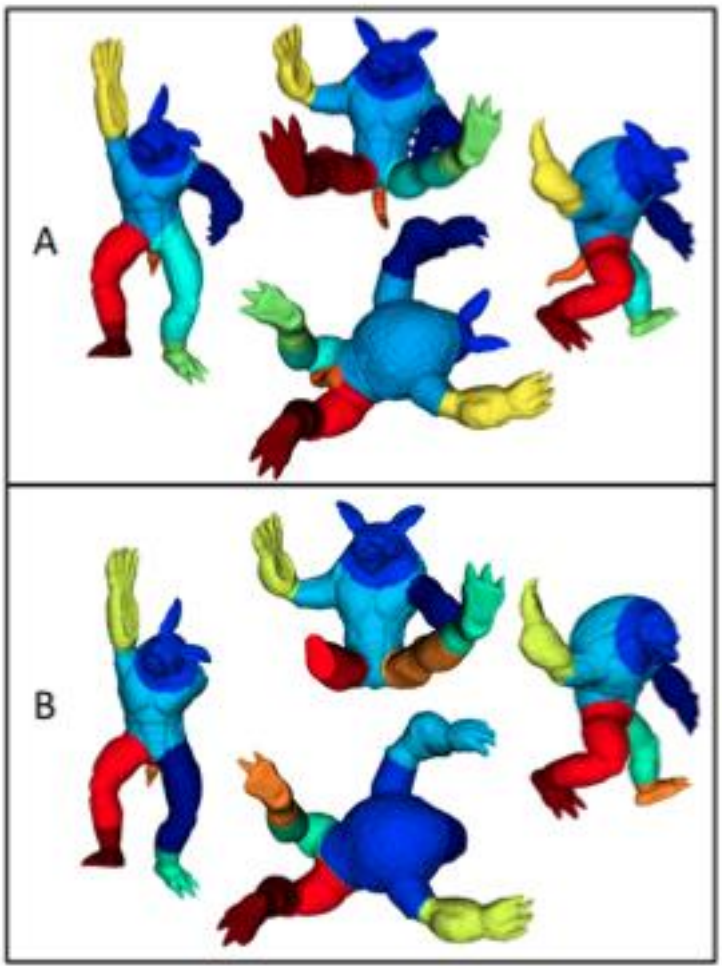

სურათი 2.4.1.7 სეგმენტაციის შედეგები ხელის მოდელზე

# 3. სითბური სეგმენტაციის შედეგები სხვადასხვა სახის სამგანზომილებიან ობიექტებზე

შემდეგ სურათებზე ნაჩვენებია სეგმენტაციის შედეგები წარმოსახვით მოდელებზე:




სურათი 3.0 ა) ადამიანის და ცხენის მოდელის სეგმენტაცია ბ) კენტავრის მოდელის სეგმენტაცია სხვადასხვა პოზებში




სურათი 3.1 ა) Armadillo-ს მოდელის სეგმენტაცია სხვადასხვა პოზებში, ბ) იგივე მოდელის დეფორმირებული სახით სეგმენტაცია(მოდელს აკლია ფეხი და ხელი)

# დასკვნა

ნაშრომში განვიხილეთ სითბური სეგმენტაციის მეთოდი, რომელიც დაფუძნებულია სითბურ ბირთვებზე. დაგროვილი ცოდნის საფუძველზე შევქმენით პროგრამული უზრუნველყოფა, რომელშიც იმპლემენტირებულია HC-Kmeans ალგორითმი და მისი საშუალებით ხდება ობიექტების სეგმენტაცია.

ჩვენს მიერ დასმული ამოცანა მოითხოვდა რამდენიმე მნიშვნელოვანი საკითხის გათვალისწინებას:

- დასამუშავებელი ინფორმაციის დიდი რაოდენობა, რომელიც ობიექტის წვეროების რაოდენობის პირდაპირპროპორციულია.
- დაბალი დონის პროგრამირება და რამოდენიმე განსხვავებული ბიბლიოთეკის დანერგვა ერთ პროექტში.

ოპერატიული მეხსიერების დაზოგვისათვის გამოყენებული იქნა მეხსიერების დინამიური გამოყოფის მეთოდები, ისე, რომ დროის კონკრეტულ მომენტში არ იხარჯება ზედმეტი რესურსები. გამოთვლების დასაჩქარებლად გამოყენებულია გამოთვლების გაპარალელების მეთოდები(გაპარალელების მეთოდები არ მუშაობს ნებისმიერ გამოთვლებზე). წარმატებით დაინერგა სამი სხვადასხვა ბიბლიოთეკების ნაკრები(C++, Qt, OpenGL). მივიღეთ სითბური ასახვის ისეთი სტრუქტურა, რომელიც ეთანადება მოდელების ადამინურ აღქმას, არის მდგრადი და ინტელექტუალური. შედეგი მდგრადია ზედაპირის სხვადასხვაგვარი ხარვეზების მიმართ. მიუხედავად ამისა არ უნდა მოვახდინოთ სითბური ასახვის ბრმად გამოყენება სეგმენტაციისათვის. ამ ექსპერიმენტში არ იყო განხილული ერთი კრიტიკული პარამეტრი : სითბოს გაბნევის დრო. სითბოს გაბნევის დრო უმნიშვნელოვანესი პარამეტრია იმის დასადგენად, თუ, რამდენად გლობალურად ან ლოკალურად ხდება გეომეტრიული სტრუქტურის გამოვლენა. რაც უფრო მეტია სითბოს გაბნევის დრო, მით უფრო გლობალურად ხდება გეომეტრიული სტრუქტურის გამოვლენა. სასურველია შემდგომში განვავითაროთ გამოთვლითი სქემა იმისათვის, რომ დავადგინოთ სტრუქტურის აღსაწერისათვის საჭირო ოპტიმალური დრო. თუმცა ეხპერიმენტებზე დაფუძნებით შეიძლება ითქვას, რომ **PCMS**-ი უპირატესობას ანიჭებს მცირე დროს რათა მოხდეს მაღალი გარჩევადობის დეტალების შენარჩუნება მცირე მაშტაბით.

# გამოყენებული ლიტერატურა


1. [HM] Yi Fang, Mengtian Sun, Minhyong Kim, Karthik Ramani –“ Heat-Mapping: A Robust Approach Toward Perceptually Consistent Mesh Segmentation” – Purdue University, West Lafayette, IN 47907, University of College London, Gower Street, London WCIE, UK. Computer Vision and Pattern Recognition (CVPR), 2011 IEE Conference, Pages 2145-2152.
2. [HS] Fernando de Goes, Siome Goldenstein, Luiz Velho – “A HierarChical Segmentation of Articulated Bodies” - Instituto de Computação - Universidade Estadual de Campinas - Campinas, SP, Brazil - Eurographics Symposium on Geometry Processing 2008, Volume 27, Number 5
3. [MPS] MORTARA M., PATANÈ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Plumber: a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies. In *Proc. of the ninth ACM symp. on solid modeling and applications* (2004), pp. 339–344.
4. OpenGL libraries - http://www.opengl.org/
5. Qt libraries - http://qt-project.org/