



**Kauno technologijos universitetas**

Informatikos fakultetas

# **Skenuotų 3D objektų taškų filtravimo metodų tyrimas**

Baigiamasis magistro projektas

---

**Mantas Mykolaitis**

Projekto autorius

**Doc. dr. Armantas Ostreika**

Vadovas

---

**Kaunas, 2019**



**Kauno technologijos universitetas**  
Informatikos fakultetas

**Skenuotų 3D objektų taškų filtravimo metodų tyrimas**  
Baigiamasis magistro projektas  
Informatika (6211BX007)

---

**Mantas Mykolaitis**

Projekto autorius

**Doc. dr. Armantas Ostreika**

Vadovas

**Doc. dr. Antanas Lenkevičius**

Recenzentas

---

**Kaunas, 2019**



**Kauno technologijos universitetas**

Informatikos fakultetas

Mantas Mykolaitis

## **Skenuotų 3D objektų taškų filtravimo metodų tyrimas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Manto Mykolaičio, baigiamasis projektas tema „Skenuotų 3D objektų taškų filtravimo metodų tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

---

(vardą ir pavardę įrašyti ranka)

---

(parašas)

Mykolaitis Mantas. SKENUOTŲ 3D OBJEKTŲ TAŠKŲ FILTRAVIMO METODŲ TYRIMAS. Magistro baigiamasis projektas, vadovas doc. dr. Armantas Ostreika; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Fiziniai mokslai, informatika.

Reikšminiai žodžiai: *filtravimas, modelis, 3D, skenavimas, taškinis debesis, pašalinis taškas.*

Kaunas, 2019. 96 p.

## **SANTRAUKA**

Apžvelgiama kalibravimo eiga bei pateikiamos išvalgos, atliekant kalibravimo procesą. Pateikiami skirtingi objektų skenavimo būdai bei aprašomas jų veikimo procesas. Išnagrinėjami skirtingi modelio pateikimo, duomenų tipai: taškinis debesis, NURBS, poligoninis tinklelis bei aprašomi dažniausiai naudojami failo formatai praktikoje – .STL ir .OBJ, kurie naudojami skenuotų modelių perkėlimui tarp skirtingų sistemų. Aprašomas eksperimentinės dalies pasiruošimo procesas. Išanalizuojami trys filtravimo metodai, iš kurių du skirti pašalinių taškų atpažinimui ir jų pakeitimui – „Koordinatės pakeitimas stipriuotu tašku“ ir „Statistinis pašalinio taško pakeitimas“ bei modelio taškų sutankinimui ir aštrių briaunų sumažinimui – „Valdomas taškų filtravimas“. Atliekami eksperimentai su atskirais filtravimo modeliais, lyginant atstumus tarp originalaus modelio taškų ir filtruoto modelio taškų. Atlikus eksperimentus su pavieniais filtrais, atliekami eksperimentai, poromis sujungtais filtrais, kai naudojami pašalinių taškų filtravimo bei taškų sutankinimo metodai, naudojant skirtingas filtravimų pateikimo tvarkas.

Mykolaitis Mantas. *FILTERING METHOD ANALYSIS OF SCANNED 3D OBJECTS*: Master's Final Degree Project, supervisor assoc. prof. Armantas Ostreika. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Physical science, informatics.

Key words: filtering, model, 3D, scanning point cloud, outlier.

Kaunas, 2019. 96 p.

## **SUMMARY**

The course of calibration process is examined and insights are written down. Different object scanning approaches and its processes are described. Analysis of different model data types is done: point cloud, NURBS and polygonal mesh. Most commonly used data format types are inspected, that are used in practice to export model between multiple platforms – .STL and .OBJ formats. Analysis of three different filtering methods is done, two of which are used to find the outlier point and modify its value – „Coordinate change, using strong point“ and „Statistic method of outlier point switching“ and a method to increase point density of the model and reduce model spikes – „Controlled point filtering“. Experiments are done, using different filtering methods, comparing distances between original model points and filtered model points. After analyzing the performance of separate filers, a combination of outlier filters and point densification filter in different filtering orders.

# TURINYS

Įvadas.....	7
Projekto aktualumas.....	7
Darbo tikslas: .....	7
Dokumento struktūra: .....	7
1. 3D SKENAVIMO METODAI .....	9
1.1. Pasiruošimas skenavimams - kalibravimas.....	9
1.2. Lazerinės trianguliacijos skenavimas.....	9
1.3. Struktūrizuotos šviesos skenavimas .....	10
1.4. Fotogrametrinis skenavimas .....	11
2. SKENUOTŲ OBJEKTŲ DUOMENŲ PATEIKIMAS.....	12
2.1. Skenuotų objektų struktūriniai modeliai.....	12
2.1.1. Taškinis debesis .....	12
2.1.2. NURBS.....	12
2.1.3. Poligoninis tinklelis .....	13
2.2. Skenuoto modelio formatai.....	14
2.2.1. STL formatas .....	14
2.2.2. OBJ formatas.....	15
3. TAŠKINIO DEBESIES FILTRAVIMO METODAI.....	16
3.1. Koordinatės pakeitimas „stipriuoju tašku“.....	16
3.2. Statistinis pašalinio taško pakeitimas.....	18
3.3. Valdomas taškų filtravimas.....	21
4. FILTRAVIMO SISTEMA IR SKENUOTŲ DUOMENŲ PARUOŠIMAS.....	24
4.1. Sistemos reikalavimai .....	24
4.2. Sistemos planas .....	24
4.3. 3D modelio duomenų paruošimas .....	26
4.4. Eksperimentinių rezultatų palyginimas .....	30
4.5. Pasikartojančių taškų indeksavimas .....	31
5. TAŠKŲ FILTRAVIMO EKSPERIMENTINIS TYRIMAS .....	34
5.1. Koordinatės pakeitimas stipriuoju tašku .....	34
5.2. Statistinis pašalinio taško pakeitimas.....	44
5.3. Valdomas taškų filtravimas.....	48
5.4. Siūlomi metodų apjungimai.....	54
6. IŠVADOS.....	65
Literatūros sąrašas.....	67
Priedai.....	68

## **Įvadas**

### **Projekto aktualumas**

Nepaisant to, jog trimačio skenavimo įrenginiai gyvuoja virš pusės amžiaus, ši technologija yra pakankamai nauja, o jos pritaikymas tik dabar pradeda vystytis. Trimatis skenavimo panaudojimas apima didelį spektrą sričių: filmų bei žaidimų industriją, gamybos ir architektūros projektavimą, mediciną, nekilnojamojo turto verslą, kokybės užtikrinimą, virtualią realybę, 3D spausdinimą ir 3D rekonstrukciją. Norint efektyviai panaudoti trimatį skenavimą, reikalinga atlikti didelį kiekį paviršiaus nuskaitymų, kurie užima nemažai laiko. Taip pat net ir atliekant detalių modelio skenavimą, dažnai sudaroma klaidinga modelio geometrija, su tam tikrais paviršiaus geometrijos trikdžiais. Svarbiausi skenuoto modelio reikalavimai:

- Modelio paviršiaus išvalymas, pakeičiant geometrinius elementus, kurie nukrypsta nuo normalių verčių, panaudojus automatinis ar pusiau automatinius filtras. Į šį procesą įeina topologinių nukrypimų, dubliuotų ar neapibrėžtų taškų viršūnių, mažų komponentų, išnykstančių arba susikertančių paviršių savitumo pašalinimas.
- Pozicionavimas, orientavimas, norint dirbti su apibrėžtu modeliu, reikia perorientuoti ar sukurti specifinę atskaitos sistemą, kad dirbti su gamybiniu modeliu būtų paprasčiau, centruojant, atidedant atskaitos taškus ar panaudojant interaktyvius manipulatorius.

### **Darbo tikslas:**

Skenuoto trimačio modelio taškų filtravimo metodų analizė, naudojant atskirus bei kombinuotus filtravimo metodus.

### **Darbo uždaviniai:**

- Palyginti automatinio turinio kūrimo metodus.
- Išanalizuoti naudojamus skenuoto modelio filtravimo metodus.
- Eksperimentiškai iširti filtravimo metodus ir jų taikymą.
- Iširti filtravimo metodų kombinavimo galimybes, siūlyti modifikacijas.
- Įvertinti algoritmų / metodų efektyvumą, siūlyti patobulinimus.

### **Dokumento struktūra:**

Dokumentą sudaro 5 skyriai. Pirmame skyriuje pateikiami skirtingi skenavimo metodai, bei pateikiama jų analizė, skenavimo metodų privalumai ir trūkumai. Antrame skyriuje aprašomas skenuoto modelio duomenų pateikimas, kur nurodoma, kokios struktūros skenavimo duomenys yra gaunami atlikus skenavimą. Taip pat, išnagrinėjami, dažniausiai naudojami, skenuoto modelio pateikimo formatai. Trečiajame skyriuje atliekama taškinio debesies filtravimo metodų analizė, kur

aprašoma metodų eiga bei pateikiamos algoritmų schemas. Ketvirtajame skyriuje aprašoma filtravimo sistema ir eksperimentų pasiruošimo procesas. Eksperimentų pasiruošimas susidaro iš: duomenų paruošimo filtravimams, rezultatų palyginimo metodikos bei duomenų indeksavimo. Penktajame skyriuje pateikiamas individualių ir kombinuotų filtravimo metodų efektyvumas bei palyginimas. Tarpusavyje lyginami individualūs filtrai bei skirtingos filtravimų kombinacijos bei seka.



## **1. 3D SKENAVIMO METODAI**

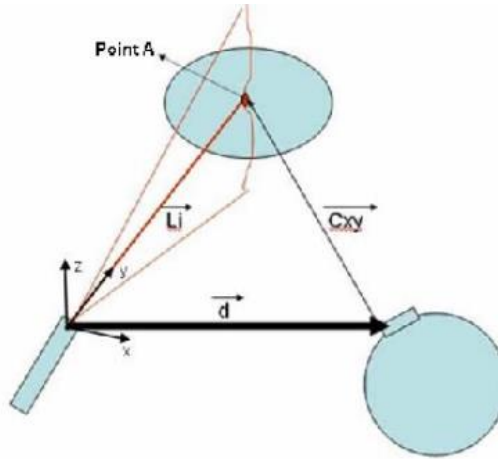
Skenavimo metodai gali būti kontaktiniai ir nekontaktiniai. Kadangi kontaktiniai skenavimo metodai yra labai artimi koordinacių matavimo įrenginiams, jų veikime yra mažiau trikdžių bei vaizdo nukrypimų, tačiau procesas užtrunka ilgiau užfiksuojant bei atidedant taškus. Nekontaktiniai skenavimo aparatai leidžia nuskaityti informaciją per atstumą, tokioje skenavimo metodikoje taip pat dalyvauja daugiau skaitmeninių informacijos apdirbimo procesų lyginant su kontaktiniais metodais.

### **1.1. Pasiruošimas skenavimams - kalibravimas**

Kalibravimas turi būti atliekamas visuose lazeriniuose bei šviesos srauto skenavimuose, kur atstumas tarp objekto nėra pririštas, o yra koreguojamas, priklausomai nuo erdvinės objekto padėties. Norint atidėti skenuojamojo objekto koordinates erdvėje, kamera bei šviesos šaltinis turi būti sukalibruoti. Dažniausiai kalibravimas turi būti atliekamas prieš tam tikrą atskaitos objektą – šachmatų lentos rašto paviršių, kurios plokščia tinklinė struktūra apibrėžia daug lengvai nuskaitytų taškų. Tiesioginis tiesinės transformacijos kalibravimas naudoja atitikmenis tarp kalibravimo lentos taškų ir kameros taškų, įvertindamas kameros parametrus, pasukimo laipsnį ir atstumą. Šis kalibravimo metodas išnaudoja faktą, jog foto kameros modelis sudaro panašumų ryšį, kuris gali būti išsprendžiamas tiesia linijinės transformacijos algoritmu, panašumo ryšius atkuriant tiesine homogenine lygtimi, kuri gali būti išsprendžiama standartiniais metodais. Norint panaudoti šį algoritmą, reikalingos tikslios trimatės erdvės koordinatės. Žinant, jog kalibravimo lentos kvadratų kampai yra tolygiai išsidėstę, koordinatės gaunamos atsižvelgus į kiekvieno kvadrato plotį. Atlikus kalibravimą, kamera ir šviesos šaltinis negali būti judinami. Keičiamas atstumas nuo kalibravimo lentos arba atskirų komponentų, kameros ar šviesos šaltinio, sudaro sistemos nestabilumą, kadangi sistema automatiškai iškalibruojama, o koordinatės praranda skaitinį stabilumą.

### **1.2. Lazerinės trianguliacijos skenavimas**

Lazerinės trianguliacijos metodu, skenuojamas objektas apšviečiamas lazerio juosta ar srautu, kuris užfiksuojamas kamera. Kuo didesnis plotas atidedamas lazeriu, tuo greičiau atliekamas skenavimas. Priklausomai koku atstumu yra atidėtas lazeris, kameros užfiksuojamame lauke jis pasirodys skirtingoje vietoje. Trianguliacijos trikampį apibūdina trys pagrindiniai parametrai: atstumas tarp kameros ir lazerio, kampas tarp lazerio spindulio ir kampas tarp kameros. Trianguliacija atliekama dviem žingsniais: aptikimu ir sprendimu. Pirmiausiai nustatomi objekto pikseliai, veikiami lazerio spindulio, kai gaunami du vaizdai kamera, vienas vaizdas objektą apšvietus lazeriu, antrasis neapšvietus, tarp šių dviejų naudojamas slenkstinis filtras, užfiksuojantis reikiamus pikselius. Užfiksavus pikselius, trianguliacijos lygtys yra sprendžiamos, o virtualus modelis yra paruošiamas eksportavimui arba vizualizacijai. Duomenų sistemos surinkimo elementai atvaizduojami (1.1 pav.) [1], kur ant objekto esantis lazerio taškas suformuoja trikampių vektorius. Lazerinės trianguliacijos esmė yra rasti taško erdvines koordinates apibrėžtame atskaitos taške.



**1.1 pav.** Trianguliacijos atvaizdavimas

čia:  $\vec{Li}$  – vektorius tarp lazerio pradžios taško ir taško A;

$\vec{d}$  – vektorius tarp lazerio pradžios taško ir kameros optinio centro;

$\vec{C_{xy}}$  – vektorius tarp kameros optinio centro ir taško A;

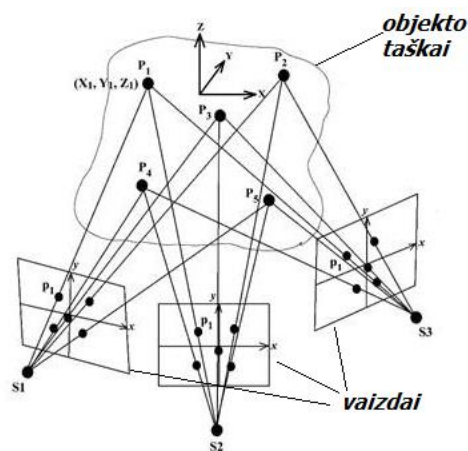
Lazerinės trianguliacijos pranašumai prieš kitus skenavimo metodus yra žema kaina, mažiau jautrus aplinkos šviesai, nuskaitymo greitis ir tikslumas, kuris gali būti gaunamas šimtųjų tikslumu. Metodo trūkumai, sunkiau nuskaityti permatomus arba atspindinčius paviršius, mažesnė ekrano raiška nei struktūrizuotos šviesos skenavime, ribotas skenavimo atstumas iki kelių metrų.

### 1.3. Struktūrizuotos šviesos skenavimas

Apšviečiant objektą skirtingais šviesos raštais, susidedančiais iš skirtingo tankumo juostų bei kryptių, gaunamas atitinkamas kiekis, skirtingai apšviesto objekto pavyzdžių. Kadangi kameros, ekrano raiškos, lauko gylio aptikimas yra apribojamas, naudojami skirtingi šviesos tinklelio raštai [2], nes panaudojus vieną, sutankintą šviesos tinklelio raštą, taškai negali būti užfiksuojami. Skenavimuose šviesos rašto vaizdas gaunamas geometriškai deformuotas, dėl objekto paviršiaus formos pokyčių, taip užfiksuojant trimates koordinates erdvėje. Šviesos raštai turi keletą signalų, padedančių užfiksuoti erdvės koordinatę, tai linijų tankumas, linijų plotis, kiekis, kurie priklauso nuo objekto parametrų bei objekto atstumo. Pasinaudojus rašto atpažinimo metodu, užfiksuojamos bei suskaičiuojamos šviesos rašto linijos. Taip pat naudojamas Grėjaus kodo sekos, nustatant juostų kiekį, apšviečiantį objektą. Optinė skiriamoji geba priklauso nuo rašto tankumo, šviesą spinduliuojančio įrenginio kokybės, taip pat optinė skiriamoji geba apribojama šviesos bangos ilgiu [2]. Struktūrizuotos šviesos skenavimo pranašumai: dažniausiai tikslesni skenavimai, nei lazeriniu metodu skenuojant iš mažo atstumo, didelė vaizdo rezoliucija, didelis tikslumas iki 10 mikronų. Metodo trūkumai, sunku atlikti kalibravimą, ryškioje šviesos aplinkoje, sudėtinga skenuoti lauke.

#### 1.4. Fotogrametrinis skenavimas

Fotogrametrija paremta fotografijų matavimais, kai matuojamas atstumas tarp dviejų paviršiaus taškų esančių plokštumoje. Fotogrametrija naudoja projekcinės geometrijos savybes [3], tokias kaip projekcinė erdvė. Šio metodo principas yra gauti bent dvi nuotraukas iš skirtingų vietų, pasinaudojus trianguliacija, sudarant bendro taško koordinatę erdvėje (1.2 pav.). Gaunama informacija gali būti: geometrinė – nusakanti erdvinę padėtį ir objekto formą, fizinė – nurodanti elektromagnetinės spinduliuotės savybes, spinduliuotės energiją, bangos ilgį, poliarizaciją, laikina – palyginanti keletą vaizdų, padarytų skirtingais laikais [3]. Fotogrametrija gali būti atlikta iš oro ir žemės. Oro fotogrametrija, kai fotografuojama iš viršaus, didelio aukščio, ir daroma daugybę persidengiančių fotografijų, tai skirta dideliems topografiniams modeliams ar topografiniams žemėlapiams kurti. Žemės fotogrametrija, naudojama trimačiams modeliams kurti, kurie gali būti nuo smulkių detalių iki didelių struktūrų ar pastatų. Algoritmais dažniausiai stengiamasi surasti mažiausių kvadratų sumą, sumažinant koordinatinių bei santykinųjų poslinkių nuokrypį, pasinaudojant Levenbergo – Marquardarto algoritmu. Fotogrametrijoje yra atvaizduojamas aukščiausios kokybės vaizdas, palyginus su kitais skenavimo įrenginiais, tai nebrangus būdas. Skenavimo minusai – kadangi procesas atliekamas naudojant optiką, yra sunku nuskaityti blizgius, nedetalizuotus, glotnius paviršius, reikia didelių kompiuterinių resursų apdorojant nuotraukas.



1.2 pav. Koordinatinių atidėjimas iš 3 vaizdų

## 2. SKENUOTŲ OBJEKTŲ DUOMENŲ PATEIKIMAS

### 2.1. Skenuotų objektų struktūriniai modeliai

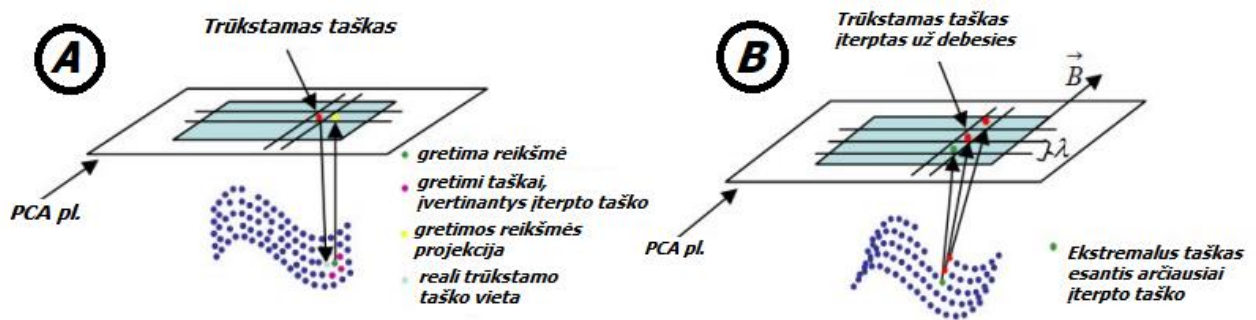
Skenuoto objekto modelis nusako kokia duomenų struktūra yra atvaizduojamas modelis. Atitinkamai nuo naudojamojo objektinio modelio pateikimo, naudojami skirtingi metodai bei filtrai, pakeisti modelio struktūrą, detalizuoti ar supaprastinti nagrinėjamąjį modelį.

#### 2.1.1. Taškinis debesis

Taškinis debesis, tai apibrėžtų taškų aibė koordinačių sistemoje [4], kuri nusako objekto išorinį paviršių trimatėje erdvėje geometriškai. Taškinis debesis gali būti gaunamas nuskanavus objektą, kai taškai erdvėje atidedami, priklausomai nuo skenuojamojo įrenginio atstumo iki skenuojamojo objekto. Erdvėje atidėti taškai gali būti vizualiai peržiūrėti, nustatant ar nėra sugeneruotų atsitiktinių, netikslių koordinačių, kurios galėtų paveikti kuriamą modelį. Taškai erdvėje apibrėžiami trimis koordinatėmis, kurios priklausomai nuo nagrinėjamąsios koordinačių sistemos gali turėti tiek teigiamas, tiek neigiamas vertes. Taškinis debesis gali būti apjungiamas į poligoninį tinklą arba apjungiamas į NURBS paviršių, naudojant paviršiaus rekonstrukcijos procesą. Todėl dėl taškinių debesies lankstumo, jis tampa ypač naudingas atvirkštinei inžinerijai, kur taškinių debesų koordinačių duomenys tiksliai atvaizduojami CAD (*angl. computer aided design*) modelyje. Perkėlus taškinių debesį į skirtingas programines sistemas, galima rankiniu būdu arba automatiniais filtrais jį apdoroti bei naudoti dirbant su CNC (*angl. computer numeric control*) staklėmis, 3D spausdintuvais ar kitais kompiuterio pagalba užprogramuotais, tiksliais apdirbimo įrenginiais.

#### 2.1.2. NURBS

NURBS tai yra kreivės bei paviršiai, išreikštos matematiškai, kai keičiant aprašytus parametrus, keičiasi NURBS kreivės ir paviršiaus forma. Taškai, kurie apjungiami NURBS kreive yra aprašomi matematine išraiška, kur kreivė yra interpoliuojama pagal taškų išsidėstymą ir taškų kiekį. NURBS kreivė bei paviršius turi kontrolės taškus, kurie gaunami atidėjus arba nuskenavus taškus. Kontrolės taškai leidžia manipuluoti modelio geometrija, išlaikant glotnumą tarp skirtingų kreivių, keičiant kontrolinių taškų padėtį. Pagal kontrolės taškus interpoliuojami taškai, esantys tarp šių taškų. Vienas iš būdų, gauti NURBS paviršių iš netvarkingai išdėlioto, nesugrupuoto taškinių debesies yra atliekamas dviem etapais. Pirmas etapas sureguliuoja ir pritaiko NURBS paviršių, antrasis optimizuoja NURBS paviršiaus pritaikymą, panaikindamas taškus esančius už debesies, per reguliavimo etapą. Reguliavimo etape debesies ekstremumų taškai yra išprojektuojami ir užfiksuojami, kur nustatomi kraštiniai taškai, tuomet sudaromi tinkeliai iš ekstremumo taškų, jeigu tinklelio srityje nėra nei vieno taško, reikia įterpti tašką centre ir jeigu įterptas taškas yra debesies viduje, yra remiamasi (2.1 pav. A), o jeigu įterptas už debesies ribų (2.1 pav. B) [7]. Jeigu yra du taškai elemente, reikia panaikinti tašką, kuris yra tolimiausias nuo centro.



2.1 pav. Reguliavimo etapas

Paviršiaus optimizavimo procesas yra naudojamas išlaikyti gautų taškų įtaką didesnę nei įterptų taškų, norint, jog nepakistų debesies topologija, sumažinant aproksimavimo paklaidą pasinaudojus formule (1):

$$E(S) = d_{p,S} < \delta \quad (1)$$

čia:  $d_{p,S}$  – atstumas tarp taškų;  $\delta$  – nuokrypos tolerancija;  $S$  – reguliuoti taškai.

Iš čia gaunami kontroliuojami taškai, tačiau tik tie, kurie priklauso taškiniam debesiai. Tada atliekamas optimizavimo procesas, kuris sugrupuojamas  $k$ -reikšmių metodu (*angl. k – means*) [7].

### 2.1.3. Poligoninis tinklelis

Poligonis tinklelis apjungia modelio taškus paprastomis geometrinėmis formomis, dažniausiai trikampaiais arba stačiakampiais. Poligoninis tinklelis, tai taškų kraštinių bei apjungtų paviršių rinkinys, sudarantis modelio paviršių. Priklausomai nuo taškų kiekio, skiriasi tinklelio tikslumas, todėl poligoninis tinklelis gali būti greitai apdorojamas modelis, esant mažesniai taškų kiekiui. Kadangi tinklelis neišreiškiamas matematiškai, jį aprašyti sudėtinga, todėl apjungti taškai yra nepriklausomi vieni nuo kitų. Pavieniai taškai gali būti lengvai manipuluojami bei keičiami, kadangi pakeitimo metu, pakinta tik ta paviršiaus dalis, kurios taškų padėtis, buvo pakeista. Poligoninio tinklelio taškai tarpusavyje nėra grupuojami, todėl tai yra naudinga keičiant tam tikrų taškų padėtį, tačiau keičiant modelio struktūrą rankiniu būdu yra žymiai sudėtingiau nei NURBS modelyje. Dažniausiai poligoninis tinklelis gaunamas Delaunay trianguliacijos metodu. Delaunay trianguliacijos metu, taškai apjungiami apskritimų kraštinėmis, kur kiekvieno taško koordinatės sutampa su bent vieno nubrėžto apskritimo kraštine. Sudarytuose apskritimuose esantys taškai sujungiami trikampaiais, sudarant paviršiaus tinklelį. Kiekvieno trikampio kampas yra tikrinamas pagal kampo sąlygą, kur atidėti duomenų taškai, esantys trikampiuose, turi sudaryti kampą tarpusavyje, kiek įmanoma didesnį, kad palyginus su priešingų kraštinių kampais, santykis neiškreiptų skaičiavimų. Kriterijus aprašantis ryšį tarp dviejų gretimų poligoninio tinklelio trikampių, nurodo, jog suma tarp vienos pusės vidinių gretimų kampų didesnė už sumą tarp kitų dviejų kampų [8]. Optimali trianguliacija yra apibūdinama išmatavus trikampių kampus, kraštines, aukštį arba paviršiaus elementus. Norint panaikinti tarpus poligoniniame

modelyje, galima pridėti papildomų viršūnių tuščiose modelio ertmėse, su sąlyga, jog jos nedeformuotų modelio kraštinių. Nelygumus tinklelyje taip pat galima panaikinti panaudojus glotninimo (*angl. smooth*) funkciją.

## 2.2. Skenuoto modelio formatai

Dažniausiai skenuoti modeliai yra naudojami mechaniniam apdirbimui, naudojant CAD programines įrangas, kuriose jie yra modifikuojami arba naudojant 3D spausdinimo proceso metu. Todėl nuskenuto modelio naudojamas formatas turi būti lengvai perkeliamas tarp skirtingų programinių įrangų, atliekant filtravimo, koregavimo procesus bei atliekant jo gamybą.

### 2.2.1. STL formatas

STL failo formatas turi užkoduotą trimačio modelio paviršiaus geometrijos informaciją. Jame išsaugojama informacija apie paviršiaus taškus bei plokštumas ir neperteikiama jokia kita vizualinė informacija kaip tekstūros. STL failas yra vienas universaliausių modelio failų formatų, kadangi jį lengvai galima perkelti tarp skirtingų sistemų platformų. STL failas yra palaikomas didžiausių 3D projektavimo/modeliavimo programų, tokių kaip: CATIA, SolidWorks, Inventor, AutoCAD, Rhinoceros, 3Ds MAX, Maya, Blender. STL failas užkoduojamas pasinaudojus išklojimo (*angl. tessellation*) būdu, kai visas paviršius yra apjungiamas vieno tipo geometrinėmis figūromis, dažniausiai trikampaiais. STL formato tikslumas priklauso nuo taškų kiekio, pagal kurias taškai tinkleliu apjungiami į paviršių. STL failas daugiausiai naudojamas pramonėje, kur naudojama 3D spausdinimo technologija, todėl pritaikymas gali būti nuo trimačių vizualinių modelių, iki medicinos srities, mechaninės pramonės ar maisto technologijos. Šis formatas gali būti perteikiamas dvejomis koduotėmis ASCII ir dvejetainė, jų pavyzdžiai pažymėti lentelėje (2.1 lentelė) [9].

#### 2.1 lentelė ASCII bei Dvejetainės koduotės formatai

ASCII koduotės formatas	Dvejetainės koduotės formatas
facet normal $n_x$ $n_y$ $n_z$	foreach triangle
outer loop	REAL32[3] - Normal vector
vertex $v_{1_x}$ $v_{1_y}$ $v_{1_z}$	REAL32[3] - Vertex 1
vertex $v_{2_x}$ $v_{2_y}$ $v_{2_z}$	REAL32[3] - Vertex 2
vertex $v_{3_x}$ $v_{3_y}$ $v_{3_z}$	REAL32[3] - Vertex 3
endloop	UINT16 - Attribute byte count
endfacet	end

ASCII koduotėje skaitomos trys paviršiaus normalės, kiekvienoje koordinačių ašyje, kiekvieno ciklo iteracijoje aprašomi trys trikampio taškai, su atitinkamomis padėtimis skirtingose koordinačių ašyse. Paviršiaus normalės koordinatė gali turėti neigiamą reikšmę, tačiau taškų koordinatės vidiniame cikle visos turi būti teigiamos. Dvejetainėje koduotėje normalės bei taškai aprašomi 32 bitų reikšmėmis, kur pirmiausiai aprašoma vektoriaus normalė kiekvienoje koordinačių ašyje, o sekančioje eigoje įvardijamos kiekvieno trikampio taško koordinatės. Dvejetainės koduotės pranašumas prieš ASCII yra tas, jog tai yra labiau kompaktiškas būdas perteikti informacija bei išlaikyti aukštą kokybę.

### 2.2.2.OBJ formatas

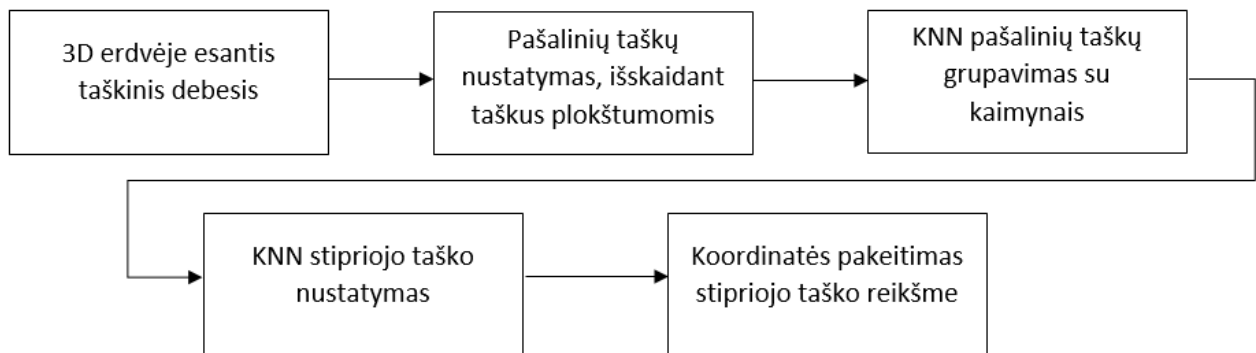
OBJ - objektinis failas turi užkoduotą informaciją apie paviršiaus geometriją, spalvas bei tekstūras. OBJ formatas toks pats universalus kaip .STL, jis yra palaikomas didžiausių 3D projektavimo/modeliavimo programų, tokių kaip: CATIA, SolidWorks, Inventor, AutoCAD, Rhinoceros, 3Ds MAX, Maya, Blender. OBJ formatas palaiko įvairesnių paviršiaus geometrijos užkodavimo būdų: išklajimo su poligoniniais paviršiais, laisvųjų kreivių (*angl. free-form curves*) bei laisvųjų paviršių (*angl. free-form surfaces*) [10]. Todėl paviršius ne tik užfiksuojamas poligonais bet ir Bezier kreivėmis, Spline ar NURBS paviršiais. OBJ failai naudoja ASCII koduotę ir išsaugoja: taškų informaciją, laisvasias kreives bei paviršius ir jų savybes, elementus, paviršių apjungimą, grupavimą, atvaizdavimą (*angl. rendering*) [11]. Todėl OBJ failas yra labiau kompleksinis, kadangi turi daugiau užkoduotų parametrų bei universalesnis, nes duomenys išsaugomi paprastu tekstu, kur jie lengvai gali būti modifikuojami, lyginant su AMF ar 3MF formatais, kurie taip pat naudojami modelių pateikime.

**Išvados:** taškinis debesis yra pagrindinis skenuotų objektų duomenų tipas iš kurio gaunami tiek NURBS paviršiai, tiek poligoninis tinklelis. Todėl taškinių debesies koregavimas patikslina gaunamus rezultatus naudojant NURBS ar poligoninio tinklelio duomenų pateikimo būdus. NURBS modelis yra lengviau bei sklandžiau valdomas erdvėje, jis nesudaro modelio kampuotumo, kadangi koreguojant kreivių reikšmes, gretimi taškai yra aproksimuojami, taip išlaikant glotnią paviršiaus kreivę. Poligoninis modelis yra mažiau priklausomas nuo gretimų poligonų, todėl tokį modelį yra žymiai paprasčiau modifikuoti, nesuardant modelio struktūros. Tiek STL tiek OBJ tipo formatai, yra tekstiniai – jie lengvai nuskaitomi bei modifikuojami. STL formatas laiko informaciją apie taškus bei kraštines, o OBJ formatas gali laikyti ir vizualinę informaciją tokią kaip spalvas bei tekstūras bei grupuotų elementų informaciją.

### 3. TAŠKINIO DEBESIES FILTRAVIMO METODAI

#### 3.1. Koordinatės pakeitimas „stipriuoju tašku“

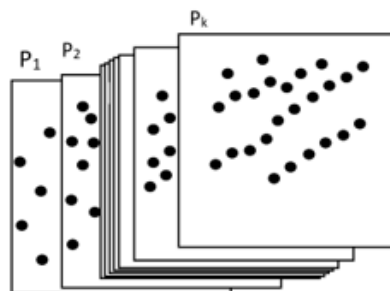
Šis filtravimo metodas sutankina objekto paviršiaus taškus, pakeisdamas pašalines taškų koordinates, nagrinėjamosiose plokštumose, į gretimų, trimatėje erdvėje išsidėsčiusių, stipriųjų taškų koordinates. Stiprieji taškai, nustatomi pagal taškų ryšį su gretimais erdviniais taškais, kuo arčiau kaimyniniai taškai išsidėstę prie nagrinėjamojo taško, tuo jis laikomas stipresniu. Žemiau pavaizduojami metodo griaučiai (3.1 pav.).



3.1 pav. Koordinatės pakeitimo stipriuoju tašku schema

##### 3.1.1. 3D išsiskiriančių taškų suradimas

Taškinis debesis  $M$  turi taškų skaičių  $m$ , taškai gali būti aprašomi vektoriumi  $v = [x, y, z]$  kaip  $x, y, z \in R^{m \times 1}$  arba  $v \in R^{m \times 3}$  [12]. Taškiniame debesyje taškai išdėliojami erdvėje - tūkstantųjų tikslumu, todėl mažoje erdvėje jų padėtis yra tiksliai nusakoma. Todėl norint sutankinti taškus bei sumažinti išsidėstymo tikslumą, paviršiaus taškų koordinatės yra suapvalinamos iki artimiausios reikšmės  $\|v_{x,y,z}\|$ , čia  $v_{x,y,z}$  yra trimatis taškas erdvėje [12]. Taškų reikšmės yra apvalinamos priklausomai nuo taškų kiekio erdvėje, kuo didesnis taškų kiekis, tuo didesnis apvalinamos taško koordinatės tikslumas yra parenkamas. Suapvalintos reikšmės sutankina taškus, taip palengvinant jų sugrupavimą, kuris atliekamas randant taškų kiekį plokštumoje. Kūno erdvė yra suskirstoma plokštumomis, kai plokštumos padėtis, koordinatės atžvilgiu, yra priimama priklausomai nuo skenuoto modelio gabaritų bei jo pozicijos. Plokštumų kiekis yra tiesiogiai proporcingas nagrinėjamos krypties koordinačių reikšmių aibei  $Q = [v_1, \dots, v_k]$  ir yra lygiagrečios viena kitai  $P_1 || P_k$  (3.2 pav).



3.2 pav. Nagrinėjama ašimi išdėstytos plokštumos su taškais.



Atliekant plokštumos analizę, apskaičiuojamas išsidėsčiusių taškų kiekis nagrinėjamoje plokštumoje, jeigu taškų kiekis plokštumoje užima didesnę vertę nei ribinė reikšmė, tai taškai priimami kaip plokštumai priklausantys. Esant mažesniai taškų kiekiui už ribinę reikšmę, taškai nepriklauso plokštumai ir yra priimami kaip pašaliniai (2).

$$P_i = \begin{cases} v_i > k, v_i \in r_i \\ v_i < k, v_i \notin r_i \end{cases} \quad (2.)$$

čia:  $P_i$  – plokštuma,  $v_i$  – nagrinėjamas taškas,  $k$  – kaimyninė reikšmė,  $r_i$  – erdvinė sritis, kurioje yra pašalinis taškas  $i$  su artimiausiais  $k$  kaimyniniais taškais.

### 3.1.2. KNN grupavimas

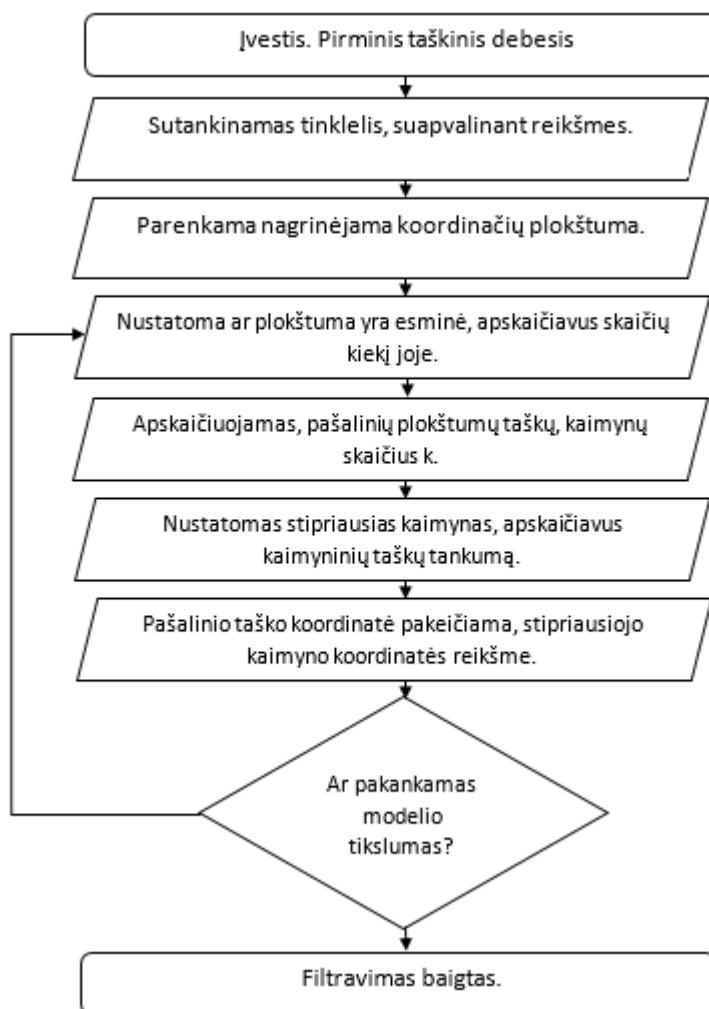
Pašaliniai taškai yra priimami kaip paviršių deformuojantys trikdžiai, kadangi šie taškai smarkiai išsiskiria nuo aplinkinių taškų bei iškreipia trimačio modelio vaizdą, sukuria santykinai didelius iškilumus. Norint ištaisyti pašalinio taško padėtį yra naudojamas, stipriausio taško koordinatės pakeitimo, metodas. Trimačio debesies taškai yra neaprašyti ryšiais, todėl nėra žinoma taškus jungianti sąsaja, galinti numatyti ryšį tarp taškų, todėl nustatant ryšį tarp taškų, reikalinga panaudoti artimiausio kaimyno klasifikavimo algoritmą, kuriuo metu pašalinis taškas apjungiamas su  $k$  artimiausiais taškais, kur  $k$  – nusako artimiausių ieškomų taškų skaičių. Šiuo metodu užtenka apskaičiuoti pašalinių taškų kaimynines reikšmes, o ne visos erdvinės srities taškų kaimynus. Priklausomai nuo bandymų, reikšmė  $k$  gali būti koreguojama, didinant ar mažinant ryšio kiekį su sąveikaujančiais kaimyniniais taškais. Sukuriant ryšį tarp taškų, galima pavaizduoti binarinę matricą  $C \in R^{m_r \times m_r}$  sudarytą iš šių elementų (3):

$$C_{ij} = \begin{cases} 1 & \text{if } i, j \in r_i \\ 0 & \text{else} \end{cases} \quad (3.)$$

čia:  $C_{ij}$  – binarinė matrica.

### 3.1.3. Stipriausiasis taškas

Suradus gretimų taškų kiekį  $k$ , ieškomas stipriausias taškas iš šios kaimyninių taškų erdvės. Stipriausias taškas yra nustatomas pasinaudojus KNN algoritmu, kur ieškoma kaimyninė, stipriausio taško vertė, turinti mažiausią koeficientą, lyginant su kitomis reikšmėmis. Taip pat stipriausią tašką galima nustatyti, pasirinkus  $r$ -ball metodą, ieškant artimiausių reikšmių aplink nagrinėjamą tašką, spindulio –  $r$  diapazone. Stipriausias taškas yra nustatomas pagal taško erdvės  $r_i$  dydį. Taškas  $r_{ik}$  esantis mažiausioje erdvėje, kai gretimi taškai yra išsidėstę tankiai, yra laikomas stipriausiu tašku [13]. Pašalinio taško vienos koordinatės reikšmė yra pakeičiama priimant stipriausio taško reikšmę. Pakeičiama koordinatė yra parenkama, priklausomai nuo pasirinktos nagrinėjamos erdvinės plokštumos, pagal kurią randami pašaliniai taškai. Koordinatės pakeitimo stipriuoju tašku filtravimo algoritmo schema pateikiama (3.3 pav.).



3.3 pav. Koordinatės pakeitimo stipriuotu tašku filtravimo algoritmo schema.

### 3.2. Statistinis pašalinio taško pakeitimas

Statistiniu metodu paremtas pašalinių taškų radimas, naudojant normalųjį skirstinį. Normalusis skirstinys yra tikimybės funkcija, apibūdinanti kintamųjų pasiskirstymą. Kintamųjų pasiskirstymas, normaliojo skirstinio metu yra simetriškas vidutinei reikšmei, nuo kurios atitinkamu koeficientu, kitos duomenų reikšmės tolygiai mažėja. Normalusis skirstinys yra parametrizuotas, nusakantis pagal kurias parametrų reikšmės priklauso jo forma, kuri atvaizduojama grafiškai, bei tikimybės. Pagrindiniai normaliojo skirstinio parametrai yra standartinis nuokrypis bei vidurkis. Vidurkis nusako normaliojo skirstinio centrinę padėtį, grafiniame reikšmių pasiskirstyme. Vidurkio centrinėje padėtyje yra susidaręs didžiausias reikšmių tankumas, kur pagrindinė dalis reikšmių išsidėsčiusios būtent prie vidurkio. Vidurkis yra išreiškiamas formule (4):

$$\bar{x} = \frac{(\sum x_i)}{n} \quad (4.)$$

čia:  $\bar{x}$  – duomenų rinkinio vidurkis,  $x_i$  – tos pačios rūšies kintamųjų reikšmės duomenų rinkinyje,  $n$  – kintamųjų kiekis duomenų rinkinyje.

Standartinis nuokrypis nusako normaliojo skirstinio grafiko plotą, kuo labiau reikšmių tankumas yra pasiskirstęs erdvėje, tuo didesnis standartinis nuokrypis bei normaliojo skirstinio plotas, gaunamas grafike. Jis taip pat nusako, statistinį atstumą tarp stebimųjų taškų vidutinių reikšmių. Standartinis nuokrypis išreiškiamas formule (5):

$$\sigma = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}} \quad (5.)$$

čia:  $\sigma$  – standartinis nuokrypis.

### 3.2.1. Koordinačių normalizavimas

Ieškant normaliojo skirstinio yra naudinga atlikti išankstinį apdorojimą, normalizuoti duomenis. Normalizavimas gali būti: savybinis mastelio pakeitimas (feature scaling) min-max metodu arba standartizavimo  $z$  – įverčiu [14]. Min-max metodu duomenys išdalijami į intervalą [0;1]. Šis normalizavimo būdas sumažina normaliojo skirstinio plotą, kadangi duomenys yra apriboti intervale. Min-max normalizavimo formulė (6):

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (6.)$$

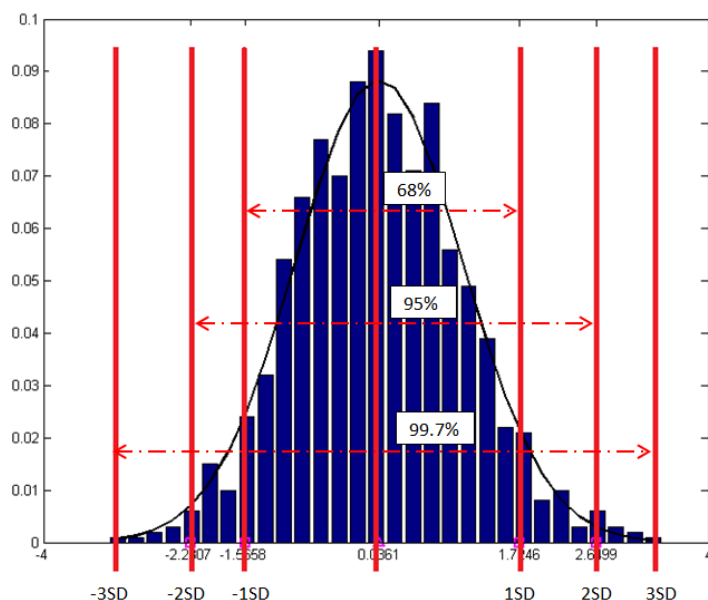
čia:  $X_{norm}$  – normalizuota duomenų reikšmė,  $X_{min}$  – minimali duomenų reikšmė,  $X_{max}$  – maksimali duomenų reikšmė.

Standartizuojant  $z$  – įverčiu, duomenys yra išdalijami tarp medianos, nurodant kokiu standartiniu skirstiniu skiriasi nagrinėjamas dydis.  $Z$  įverčio intervalas yra [-3;3]. Standartizavimas  $z$  – įverčiu leidžia nustatyti duomens padėtį lyginant su mediana, kurios standartinis nuokrypis yra  $\delta = 0$ . Standartizavimas  $z$  – įverčiu yra išreiškiamas formule (7):

$$z = \frac{x - \mu}{\delta} \quad (7.)$$

čia:  $z$  – standartizavimo įvertis,  $\mu$  – vidurkis.

Naudojant normalųjį skirstinį, nustatomos pašalinės reikšmės, kurios yra labiausiai nutolusios nuo vidurkio, jos būna 3 normaliojo skirstinio kvartilyje (3.4 pav.). Kadangi 3 kvartilyje esantys duomenys yra duomenų sklaidos gale ir labiausiai išsiskiria savo dydžiu, duomenys šiame skirsnyje yra priimami kaip pašaliniai.



3.4 pav. Normaliojo skirstinio kvartiliai

Normalusis skirstinys sudaromas pagal artimiausius kiekvieno taško kaimynus, kur tankiausi bei rečiausiai išdėstę pašaliniai taškai, esantys trečiame neigiamame bei teigiamame kvartiliuose, yra priimami kaip duomenų paklaidos, išsiskiriančios iš standartinės duomenų grupės. Artimiausi kaimynai randami pasinaudojus KNN algoritmu, priimant euklidinį arba Minkowskio atstumą. Euklidinis atstumas, išreiškiamas formule (8):

$$d(A, B) = \sqrt{\sum_{i=1}^m (p_i - q_i)^2} \quad (8.)$$

čia:  $d$  – atstumas tarp taškų,  $m$  – duomenų kiekis,  $p_i$  – pirmasis taškas,  $q_i$  – antrasis taškas.

Minkowskio atstumas, išreiškiamas formule (9):

$$d(A, B) = \left( \sum_{i=1}^m |p_i - q_i|^r \right)^{\frac{1}{r}} \quad (9.)$$

### 3.2.2. Pašalinio taško filtravimas

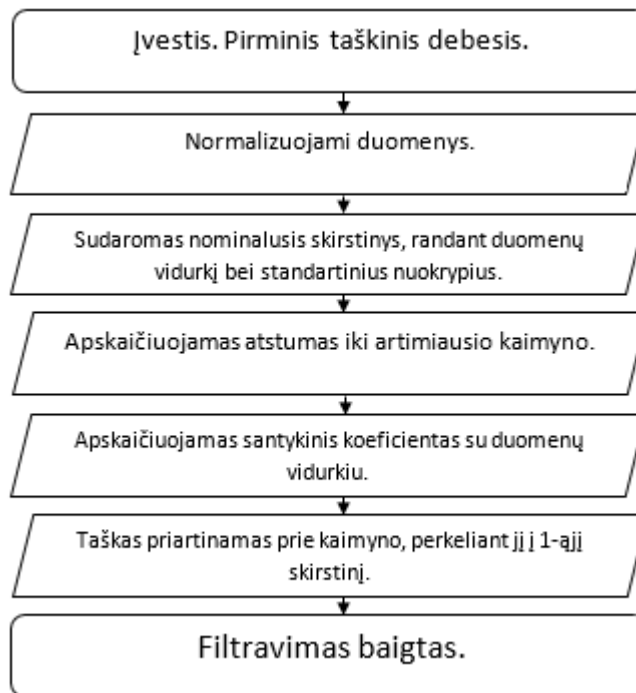
Pašalinis taškas yra filtruojamas padidinant arba sumažinant apkrovimą, kol pašalinio taško reikšmė patenka į pirmąjį kvartilį ir pereina vidurinę skirstinio vertę. Pašalinių taškų reikšmės, kurios paimamos iš identifikuotų 3 kvartilio rėžių, yra naudojamos paskaičiuojant koeficientą, nurodantį santykį tarp nagrinėjamojo pašalinio duomens bei paskaičiuoto duomenų imties vidurkio (10):

$$koef = \frac{x_i}{\bar{x}} \quad (10.)$$

Gautasis koeficientas yra naudojamas artinant, nagrinėjamąjį kintamąjį, prie kaimyno reikšmės. Nagrinėjamajam dydžiui esant didesniu už kaimyninę reikšmę, pridedamas apkrovimas, priartinant jį prie kaimyninės reikšmės, kai dydis yra mažesnis už kaimyninę reikšmę, apkrovimas yra sumažinamas (11).

$$p_i = \begin{cases} p_i > q_i, -\frac{p_i - q_i}{koef} \\ p_i < q_i, -\frac{p_i - q_i}{koef} \end{cases} \quad (11.)$$

Statistinio pašalinių taškų filtravimo algoritmo schema pavaizduota (3.5 pav.).



3.5 pav. Statistinio pašalinių taškų filtravimo algoritmo schema.

### 3.3. Valdomas taškų filtravimas

Valdomas taškų filtravimas tai laikui našus, išsaugantis formas bei jas tikslinantis – filtravimas. Valdomo taškų filtravimo metu priimama, jog išvesties vaizdas yra linijinė pradinio vaizdo transformacija [15]. Valdomas taškų filtravimas remiasi taškų palyginimu su jų lokaliais kaimynais. Todėl prieš atliekant filtravimą, reikia nustatyti kiekvieno taško kaimyninius taškus, kurių reikšmės lyginant su nagrinėjamu tašku, yra mažiausiai nutolusios erdvėje.

#### 3.3.1. Centroidės nustatymas

Kiekvienas kaimyninis taškas randamas pasinaudojus  $k$  – artimiausio kaimyno (KNN) algoritmu arba  $r$ -ball algoritmu, kuris randa kaimynus esančius tam tikrame spindulyje. Iš gautų kaimyninių taškų rinkinio paskaičiuojama reikšmė  $\bar{p}_i$ , ši reikšmė vadinama kaimyninių taškų centroide. Kaimyninių

taškų centroidė nusako vidutinį taškų pasiskirstymą, nagrinėjamoje trimačio erdvinio modelio srityje. Centroidės  $\bar{p}_i$  reikšmė gaunama iš formulės (12):

$$\bar{p}_i = \frac{1}{|N(p_i)|} \sum_{p_{ij} \in N(p_i)} p_{ij} \quad (12.)$$

čia:  $|N(p_i)|$  nusako  $N(p_i)$  kardinalumą, unikalių reikšmių kiekį nagrinėjamoje srityje.

### 3.3.2. Linijinio modelio koeficientų radimas

Sekančiame žingsnyje, suradus  $p_i$  kaimynines reikšmes, priimama prielaida, jog išfiltruoto modelio taškinis debesis turi lokalų linijinį modelį su valdomais taškais. Visi nagrinėjami taškai yra tarpusavyje susieti ryšiu, kur taško reikšmė yra patikslinama, centroidine reikšme  $\bar{p}_i$ . Kaimyninių  $N(p_i)$  taškų reikšmių linijinė transformacija atliekama, norint gauti išvesties tašką  $p'_{ij}$  (13).

$$p'_{ij} = a_i p_{ij} + b_i \quad (13.)$$

čia  $p'_{ij}$  yra išvesties taškas, kuris yra filtruotas,  $a_i$  ir  $b_i$  linijinio modelio koeficientai, kurie gaunami apskaičiuojant sumažinant funkciją  $N(p_i)$  (14).

$$J(a_i, b_i) = \sum_{p_{ij} \in N(p_i)} \left( (a_i p_{ij} + b_i - p_{ij})^2 + \epsilon a_i^2 \right) \quad (14.)$$

čia:  $\epsilon$  – parametras nusakantis padidinanti filtro poveikį.

Linijinio modelio koeficientai  $a_i$  ir  $b_i$  gaunami, atitinkamai (15) (16):

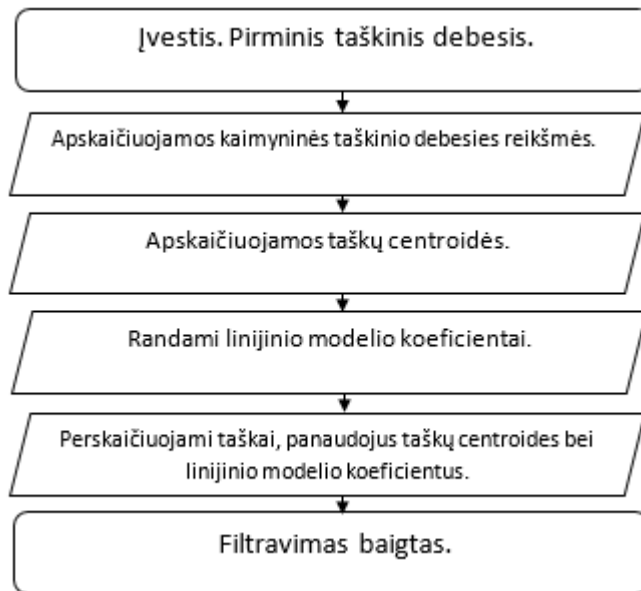
$$a_i = \frac{\left( \frac{1}{|N(p_i)|} \sum p_{ij} \cdot p_{ij} - \bar{p}_i \cdot \bar{p}_i \right)}{\left( \left( \frac{1}{|N(p_i)|} \sum p_{ij} \cdot p_{ij} - \bar{p}_i \cdot \bar{p}_i \right) + \epsilon \right)} \quad (15.)$$

$$b_i = \bar{p}_i - a_i \cdot \bar{p}_i \quad (16.)$$

Gavus linijinio modelio koeficientus  $a_i$  ir  $b_i$ , atliekama kiekvieno taško linijinė transformacija. Atliekant linijinę transformaciją, išvedama filtruoto taško reikšmė  $p'_{ij}$ , galinti turėti keletą reikšmių, kadangi  $p_{ij}$  gali priklausyti skirtingiems kaimynams, todėl  $p'_i$  reikšmė turi būti perskaičiuojama su savo kaimynu  $N(p_i)$ , kur išvedama formulė (17):

$$p'_i = a_i p_i + b_i \quad (17.)$$

Žemiau esančiame paveikslėlyje (3.6 pav.) yra atvaizduojama valdomojo taškų filtravimo schema.



**3.6 pav.** Valdomo taškų filtravimo algoritmo schema

## **4. FILTRAVIMO SISTEMA IR SKENUOTŲ DUOMENŲ PARUOŠIMAS**

### **4.1. Sistemos reikalavimai**

Sistemos funkciniai reikalavimai apima programos bei jos funkcijų atliekamas užduotis, įvardytais reikalavimais tikimasi pasiekti reikiamus užduoties rezultatus. Nefunkciniai reikalavimai aprašo sistemos veikimo kokybę bei našumą.

Sistemos funkciniai reikalavimai:

1. Taškinio debesies glotninimas;
2. Galimybė reguliuoti filtravimą reguliuojančius parametrus;
3. Galimybė pakartotinai naudoti filtrus iteracijomis.

Sistemos nefunkciniai reikalavimai:

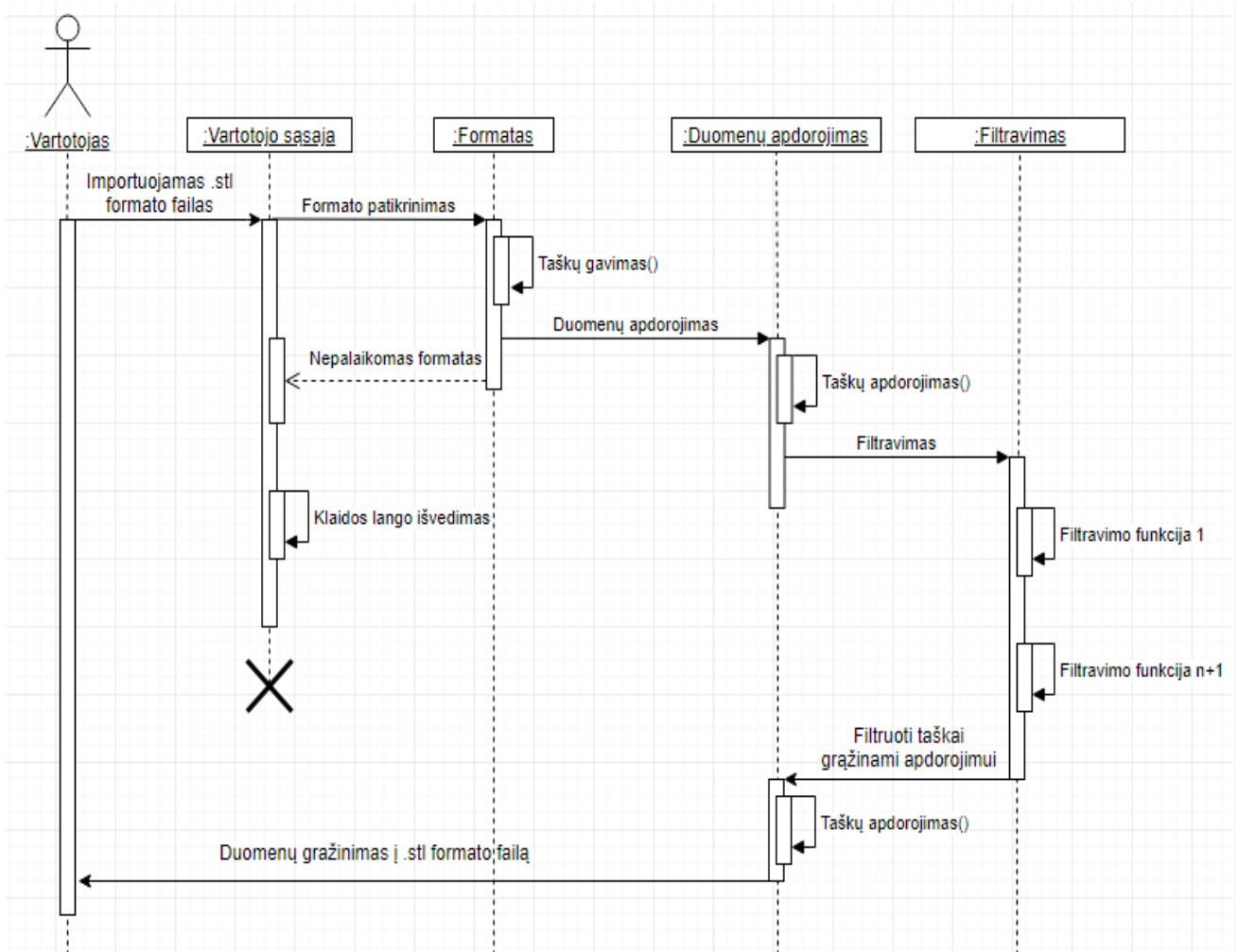
1. Filtravimo efektyvumas naudojant skirtingus parametrus.
2. Filtravimo įtaka apjungiant filtrus.

Siekiamas kokybės kriterijus yra filtravimo kokybės efektyvumas, pašalinant analizuojamas problemas: pakeičiant taškų, sudarančių pašalinius paviršiaus iškylimus, reikšmes.

### **4.2. Sistemos planas**

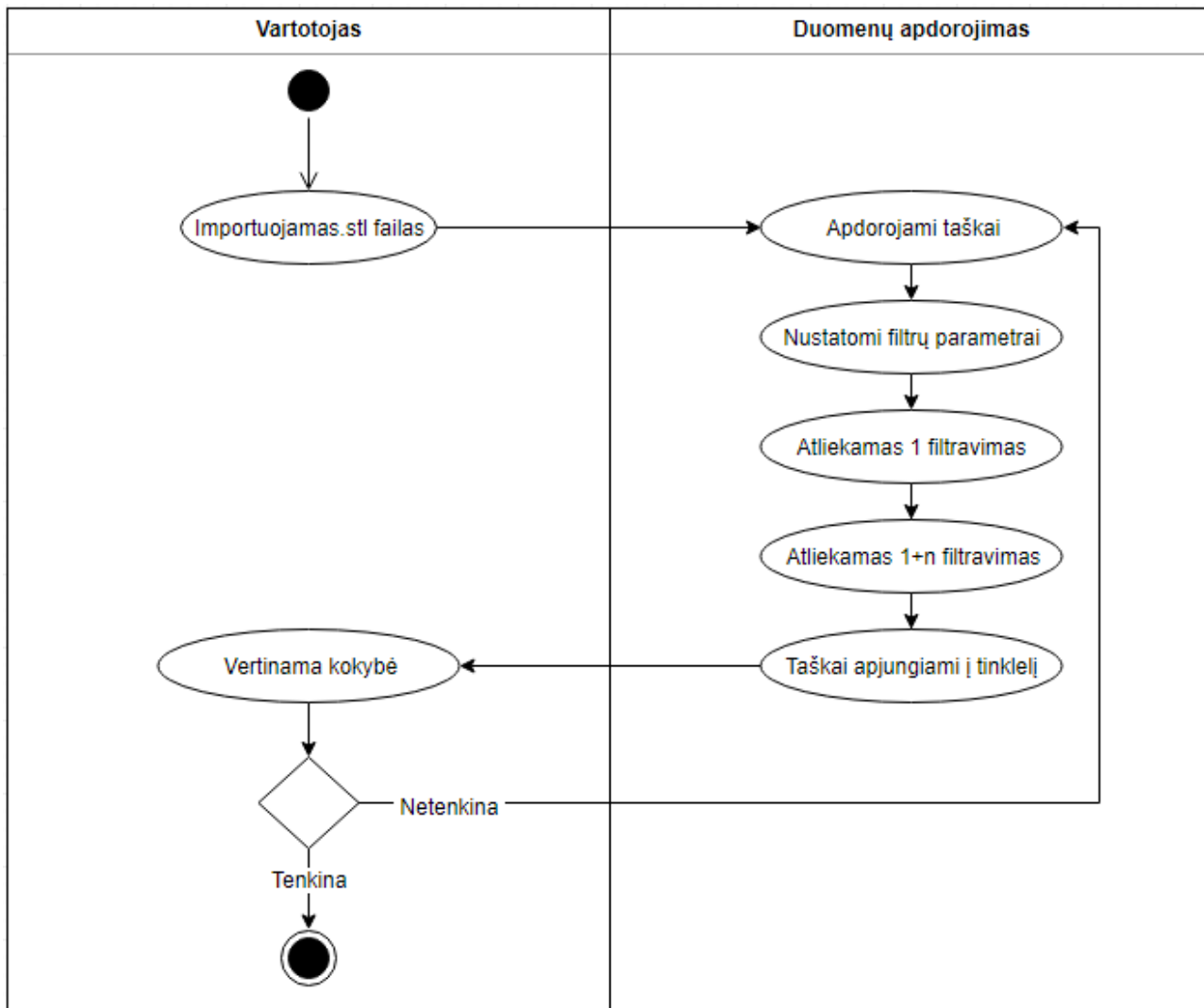
UML panaudos atvejų diagrama aprašo projektuojamos sistemos galimybes bei veiksmus, kuriuos sistema geba atlikti. Sistemoje yra veikėjas, kuris importuoja į sistemą .stl formato failą. Vartotojo sąsaja sukuria užklausą ar importuotas failas yra palaikomo formato, jeigu užklausa neigiama, iškviečiamas klaidos perspėjimas. Jeigu formatas palaikomas, paimamos taškinio debesies koordinatės ir jos apdorojamos bei paruošiamos filtravimui. Filtravimo žingsnyje atliekami taškų filtravimai, priklausomai nuo filtrų kiekio. Atlikus filtravimą filtruotas taškinis debesis gražinamas veikėjui, kuris yra vartotojas. UML sekos diagrama pavaizduota (4.1 pav.).





4.1 pav. UML sekos (sequence) diagrama

UML veiklos diagrama schematiškai atvaizduoja veiksmų eigą. Vartotojas importuoja .stl failą, duomenų funkcija apdoroja .stl formate esančius taškus. Apdorojus taškus, pradedamos filtravimo iteracijos, kur parenkami filtravimo parametrai, po kurių eilės tvarka atliekami taškinio debesies filtravimai. Gavus filtruotus taškus, jie apjungiami į tinkelį ir atvaizduojami vartotojui, jeigu vartotojo kokybė netenkina, duomenys gražinami, kur pakeičiamos filtrų parametru vėrtės. Įvedus naujas parametru reikšmes, atliekama sekanti iteracija, po kurios taškinio debesies duomenys dar kartą gražinami vartotojui. Jeigu duomenų kokybė toliau netenkina vartotojo, modelio duomenys yra siunčiami sekantiai iteracijai, jeigu kokybė tenkina vartotoją, duomenys išeksportuojami. UML veiklos diagrama pavaizduota (4.2 pav.).



4.2 pav. UML veiklos (activity) diagrama

### 4.3. 3D modelio duomenų paruošimas

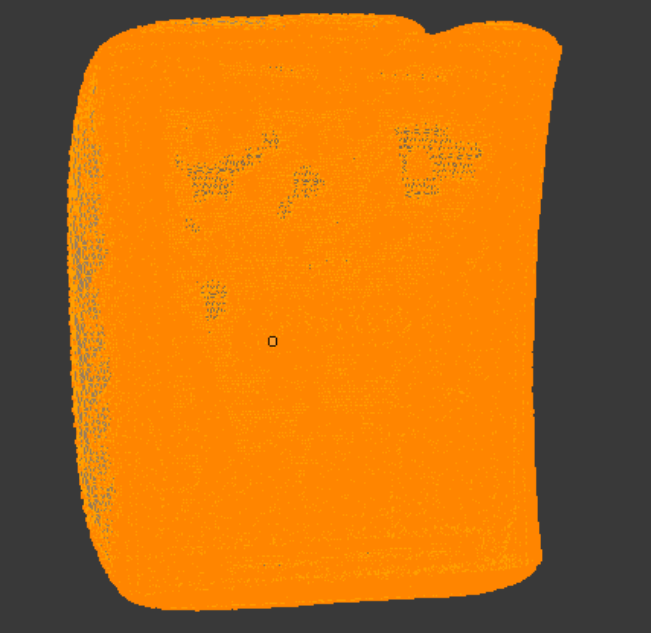
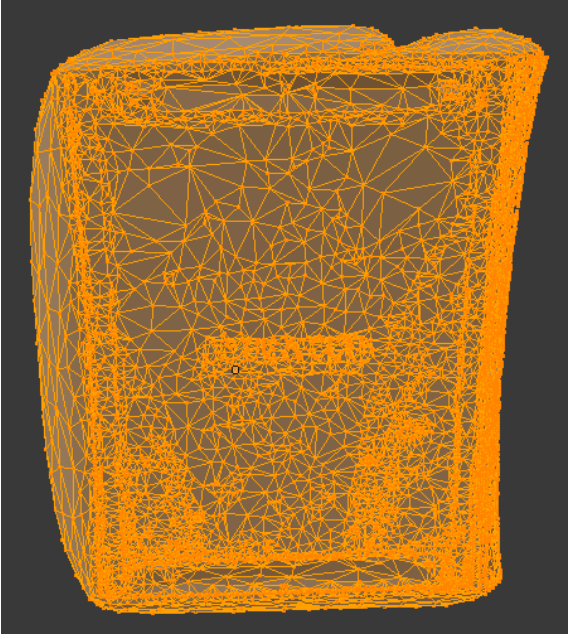
Norint atlikti taškinio debesies filtravimą, turi būti pasirenkamas objektas, kuris yra sudarytas iš erdvėje išdėliotų taškų. Skenuotas objektas, yra gautas pasinaudojus „David – Laser scanner“ programine įranga. Nuskenuotas objektas yra Microsoft kompanijos įrenginys – „Xbox360“ žaidimų konsolė. Atlikus 10 skenavimų, apytiksliai iš 120° diapazono, gaunamas vaizdas, kuris nurodytas lentelėje (4.1 lentelė).

#### 4.1 lentelė Realus bei nuskenuto modelių palyginimas

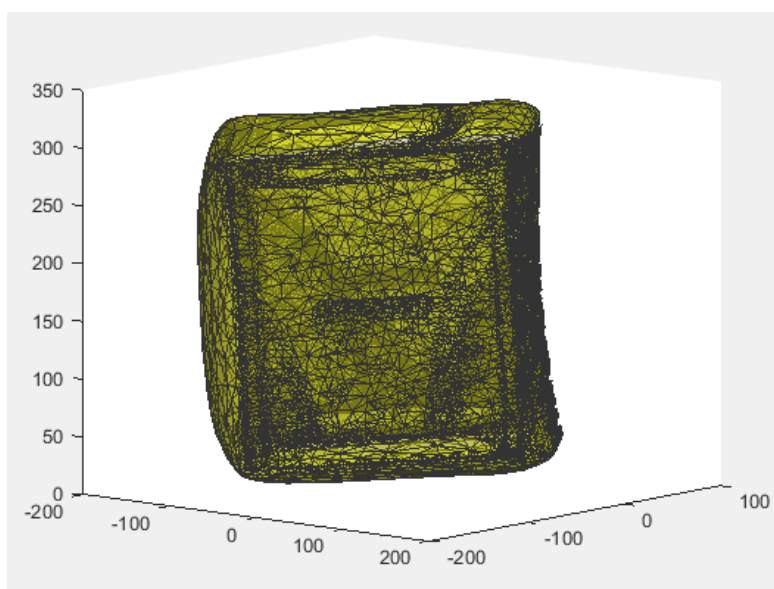
Realus vaizdas	Nuskenutas vaizdas
	
	

„David – Laser scanner“ programine įranga, failas yra išeksportuojamas .obj formatu. Pasinaudojus „Blender“ programine įranga importuojamas .obj failas, kur modelio tinklelio tikslumas yra sumažinamas, taškai iš kurių sudaromas tinklelis yra išretinami. Tinklelis išretinamas, panaikinant dalį taškų, santykiu 1:50 (4.2 lentelė). Modelio taškai gali būti rankiniu būdu modifikuojami, sukuriant pašalinius taškus bei modelio paviršiaus trikdžius, kurie bus naudojami filtravime – įdubos, iškylimai, taškų sutankėjimas. Sukūrus norimus trūkumus, objekto taškinio debesies taškuose, modelis išeksportuojamas .stl failo formatu.

#### 4.2 lentelė Modelio tinklelio sutankinimas

Modelio tinklelio tankumas po importavimo	Modelio tinklelio tankumas po taškų retinimo
	

Modelio filtravimas atliekamas „MATLAB“ programine įranga, kur yra importuojamas .stl formato failas. Pasinaudojus „MATLAB“ puslapyje pateiktu scenarijumi, importuojami dvejetainio .stl formato duomenys – taškai bei kraštinių seka, jungianti erdvėje išdėliotus taškus (1p.1 lentelė). Žemiau pateiktas nuskenuotas „Xbox360“ modelis – „MATLAB“ aplinkoje, su apjungtais taškais.



4.3 pav. „MATLAB“ aplinkoje atvaizduotas modelis

Perkelta struktūra turi du duomenų rinkinius – taškų padėtį (V) bei paviršių apjungimą (F) (4.4 pav.). Šiame faile taškai atvaizduojami trimis stulpeliais, kur atitinkamai aprašytos x, y bei z, koordinatės. Paviršiai taip pat aprašomi trimis stulpeliais, taip nusakant modelio paviršiaus briaunas, kurios jungia

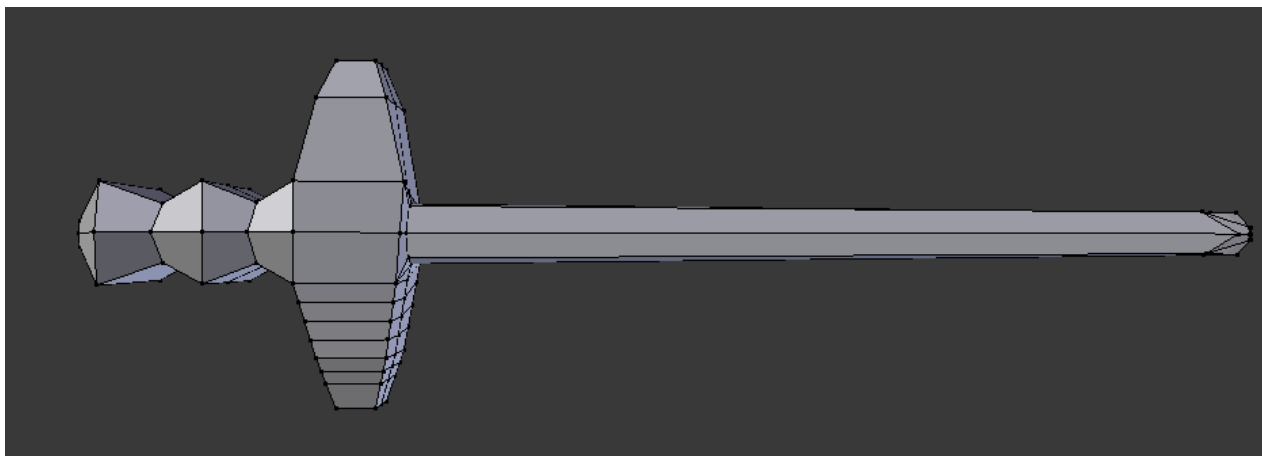
taškus. Kadangi paviršiai apjungti trikampaiais, jų faile yra tris kart mažiau nei taškų, nurodant kurie taškai yra kartu apjungiami.

Import	Name ▲	Size	Bytes	Class
<input checked="" type="checkbox"/>	FV	1x1	4241632	struct

```
faces: [44180x3 double]
vertices: [132540x3 double]
```

**4.4 pav.** Failo informacija „MATLAB“ aplinkoje

Kadangi objektas sudarytas iš 132540 taškų, eksperimentinę dalį atlikinėti yra nenašu su tokiu duomenų rinkiniu, dėl ilgų atskirų filtravimo skaičiavimų. Kiekvieno filtravimo metodo metu, galima keisti filtravimo koeficientus, nuo kurių skiriasi filtravimo efektyvumas bei rezultatai. Todėl atskiri eksperimentai yra atliekami su ranka sukurtu modeliu, kuris šiuo atveju yra mažo poligonų kiekio kardas (4.5 pav.), paspartinantis filtravimo eksperimentinį procesą. Modelis sukurtas pasinaudojus „Blender“ programine įranga, kuriant modelį, jis nebuvo kuriamas ašiai simetriškas, kad būtų galima palyginti atskirus taškų pokyčius su skirtingais taškų išsidėstymais, gaunant taškų reikšmes, kurios nėra vienos kitom kolinearios.



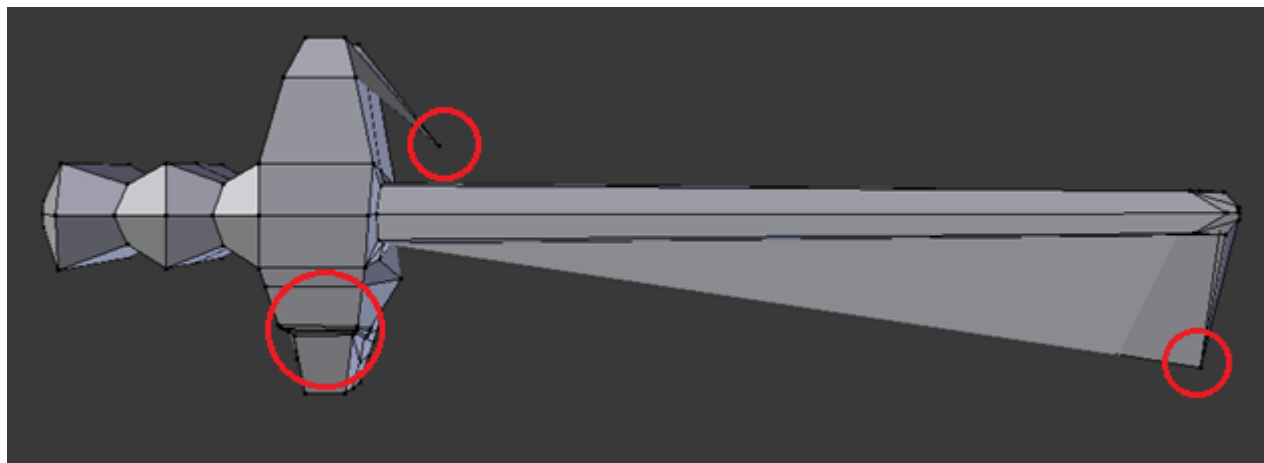
**4.5 pav.** Kardo trimatis modelis su pagal kurį lyginami filtrai

Iš pagrindinio modelio, sukuriamas modifikuotas trimatis modelis su trikdžiais (4.6 pav.), kur nėra pridėta papildomų taškų, todėl tiek pradinio trimačio modelio, tiek su trikdžiais modifikuoto trimačio modelio, taškų indeksai bei taškų kiekis atitinka. Modifikuotame modelyje yra naudojama bendra globalinė koordinatinių sistema, todėl nepakeisti taškai išlieka toje pačioje vietoje, jų koordinatinių reikšmės sutampa. Modifikuotame modelyje, yra sukuriami papildomi trikdžiai (4.3 lentelė **Klaida! Nerasas nuorodos šaltinis.**), kurie gali atsirasti skenuojant modelį:

1. Trikdžis. Modelyje sukuriamas pašalinis taškas (*angl. outlier*), kuris erdvėje yra atitolęs nuo kitų taškų, tai gali nutikti, kai objektas skenuojamas per daug dideliais žingsniais keičiant skenavimo kampą.

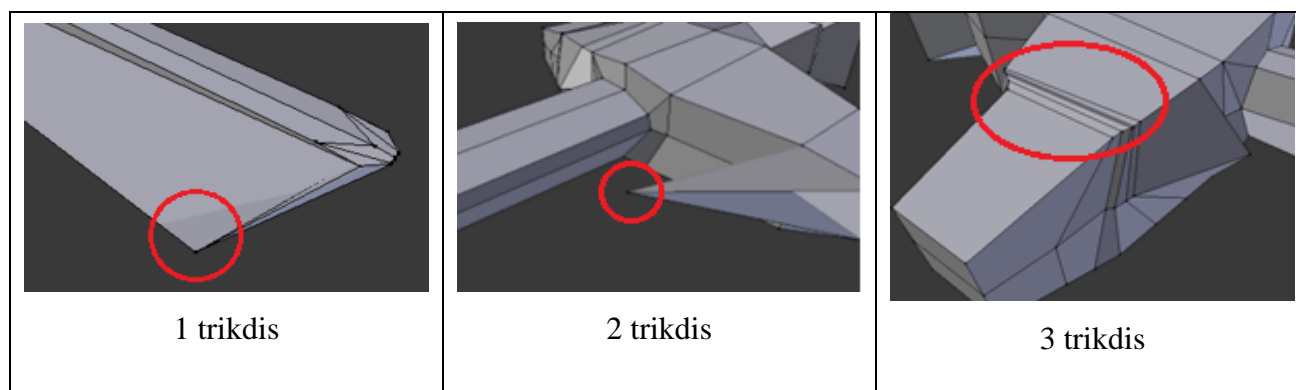
2. Trikdžiai. Modelyje sukuriama pašalinis taškas, kuris yra geometriškai nukrypęs nuo savo erdvinės padėties, tačiau taško padėtis erdvėje nenutolsta nuo modelio, o išlieka arti gretimų modelio taškų, taip gali nutikti, kai modelio sukimo žingsnis yra padidinamas viename skenavime.

3. Trikdžiai. Taškai per daug susitelkę vienoje modelio vietoje, jie gali atsirasti, modeliui sujudėjus, kai pasukamas skenavimo kampas arba atliekant per mažą skenavimo žingsnį, kai modelis pasukamas mažu kampu tarp skirtingų skenavimų.



4.6 pav. Rankiniu būdu sukurti trikdžiai trimačiame modelyje

4.3 lentelė Sukurti modelio trikdžiai



#### 4.4. Eksperimentinių rezultatų palyginimas

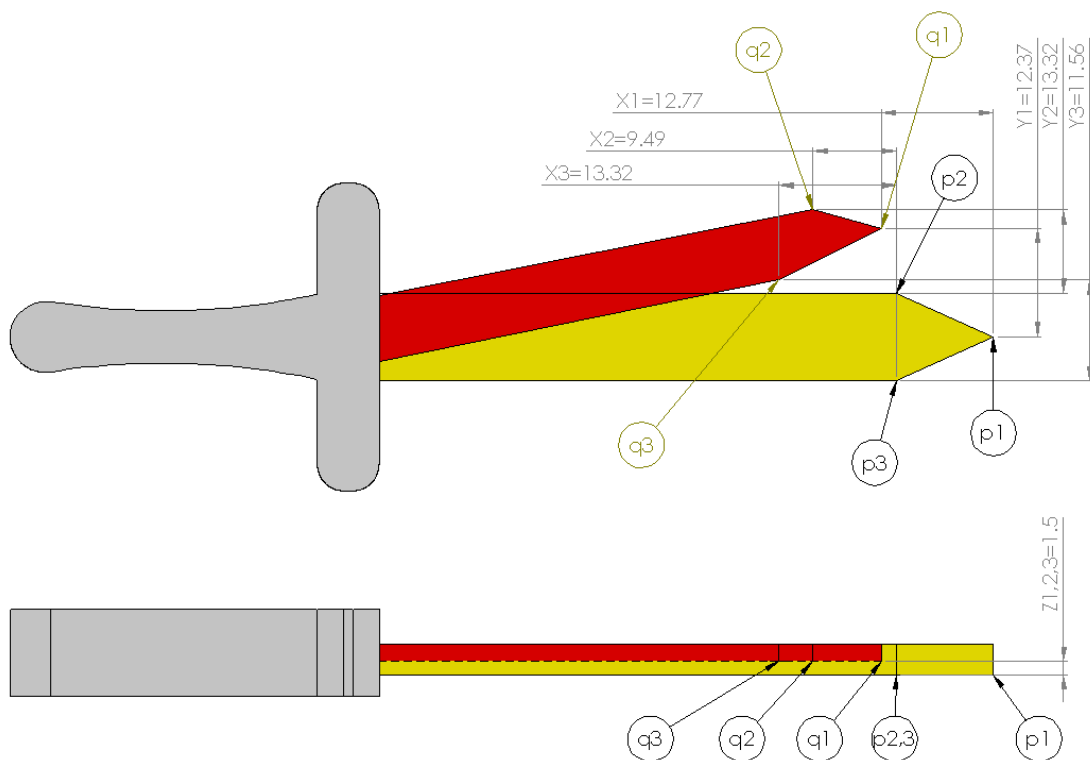
Rezultatų palyginimų metu, yra atliekama dviejų modelių koordinatinių palyginimo analizė. Abu modeliai yra atidedami globalioje koordinatinių sistemoje, kur didžioji dalis taškų yra išsidėstę toje pačioje pozicijoje. Kadangi modelis su trikdžiais kurtas iš pirminio modelio, su kuriuo lyginami rezultatai, visi abiejų modelių taškų indeksai atitinka ir atspindi tuos pačius taškus. Taškai lyginami paskaičiavus skirtumo koeficientą, naudojant euklidinį, trimatės erdvės atstumo nustatymą (18):

$$(p1_i, p2_i) = \sqrt{\sum_{i=1}^m (p1x_i - p2x_i)^2 + (p1y_i - p2y_i)^2 + (p1z_i - p2z_i)^2} \quad (18.)$$

Vizualiai palyginimas atvaizduojamas su dviem modeliais (4.7 pav.), pilka spalva pažymėta bendra dalis, kurios koordinatės sutampa, skirtingomis spalvomis (raudona ir geltona) išskiriami modeliai. Palyginimo proceso metu, paimamos to paties indekso koordinatės  $x$ ,  $y$  bei  $z$  ašyje ir paskaičiuojamas atstumas tarp to paties indekso taškų. Paskaičiuotus atstumus gausis dviejų tipų reikšmės:

1. Atstumai tarp sukurtų pašalinių taškų, nusakantys filtravimo efektyvumą ties konkrečia modelio deformacija.
2. Bendras skirtumo koeficientas, koku bendru atstumu filtruotas modelis skiriasi nuo realaus modelio.

Rezultatai vertinami tuo geriau, kuo atstumas tarp taškų yra mažesnis, idealiu atveju, atstumas tarp erdvėje išsidėsčiusių taškų yra lygus 0. Pirmojo trikdžio skirtumas  $s_1 = 2.6500$ , antrojo trikdžio skirtumas  $s_2 = 2.7432$ , bendra skirtumo reikšmė  $skirtumas = 11.0650$ .



4.7 pav. Erdvinio modelio taškų palyginimas

#### 4.5. Pasikartojančių taškų indeksavimas

Nuskaičius .stl failą su užkoduotu taškinio debesies rinkiniu, ta pati koordinatė reikšmė yra naudojama kelis kartus, apjungiant skirtingas modelio kraštines. Todėl duomenų masyve taško reikšmė pasikartoja daugiau nei vieną kartą ir yra identifikuojama kaip unikali reikšmė. Perskaičiuoti tą pačią reikšmę kelis kartus yra nenašu, kadangi kiekviena koordinatė yra pasikartojanti mažiausiai tris kartus, nes elementariausias trimačio modelio paviršius apjungiamas trikampio forma. Taškų rinkinys yra naudojamas beveik visose funkcijose, todėl skaičiavimai ženkliai padidėja, skaičiuojant

pasikartojančius taškus. Taip pat pasikartojantys duomenys gali iškreipti pašalinių taškų nustatymą, kadangi pašalinių taškų reikšmės gali būti identifikuojamos, kaip tų pačių koordinacių unikalios reikšmės, arba skaičiuojant baigtinę taškų sritį, ne visi pasikartojantys taškai bus išfiltruojami ir modelis gausis nepakitęs arba dar labiau deformuotas, kadangi bus išfiltruota tik dalis duomenų. Norint išvengti taškų pasikartojimo, sukuriamas naujas masyvas, kuriame užfiksuojama unikali taško reikšmė, o esant taško pasikartojimui, duomenų masyve išsaugomas pasikartojančios reikšmės indeksas, taip išvengiant taško reikšmės pasikartojimo masyve. Šios funkcijos pseudo kodas aprašytas žemiau (4.4 lentelė):

**4.4 lentelė** Pasikartojančių taškų indeksavimo pseudo kodas

```
Pasikartojantys(p)
unique_p ← [0]
size_u
  for i = 1 to length(p)
    a = length(unique_p)
    for j = 1 to a
      if p != unique_p
        unique_p(i,1:3) = p(i)
        unique_p(j,4) = i
      else p == unique_p
    for k = 5 to size_u
      if unique_p(j,k)==0
        unique_p(j,k)=i
  return unique_p
```

Pirmiausiai paimamas visų koordinacių taškų ilgis – „i“, apskaičiuojama reikšmė – „size\_u“, nurodanti iki kiek pasikartojančių taškų gali sukaupti masyvas, ši reikšmė priklauso nuo duomenų kiekio. „Size\_u“ reikšmė nėra pastovi, kadangi kuo daugiau kraštinių jungiasi su nagrinėjamuoju tašku, tuo daugiau pasikartojančių reikšmių turi taškų masyvas. Taip pat kuo detalesnis modelis, tuo daugiau taškų yra pasikartojančių, todėl „size\_u“ atitinkamai didėja. Taip pat sukuriamas – „unique\_p“ masyvas, kuriame išsaugojamos visos unikalios taškų koordinacių reikšmės ir pagrindinio masyvo reikšmių indeksai. Pirmas „for“ ciklas eina per taškų rinkinio masyvą, antrasis ciklas lygina ar reikšmė nėra pasikartojanti su sukurtu – „unique\_p“ masyvo reikšmėmis. Jeigu pereinamas visas ciklas ir koordinatės reikšmė užfiksuojama kaip unikali, jos x, y, z koordinatės yra išsaugojamos 1-3 stulpeliuose, o indeksas yra išsaugojamas 4 stulpelyje. Jeigu ji pasikartojanti, tuomet randama eilutė su jau išsaugotomis reikšmėmis 1-3 stulpeliuose ir pradedamas sekantis ciklas „for“, kuris ieško nuo



5 stulpelio reikšmės ar stulpelis yra tuščias, jeigu jame yra išsaugota indekso reikšmė, einama prie kito stulpelio, jeigu stulpelyje nėra reikšmės, tuomet išsaugoma indekso reikšmė ir ciklas nutraukiamas. Taip pridant pasikartojančios reikšmės indeksą, iš pradinio masyvo. Funkcijos kodas, indeksuojantis pasikartojančius masyvo taškus, pavaizduotas (2p.1 lentelė).

## 5. TAŠKŲ FILTRAVIMO EKSPERIMENTINIS TYRIMAS

### 5.1. Koordinatės pakeitimas stipriuotu tašku

Koordinatės pakeitimas stipriuotu tašku yra pašalinių taškų koordinačių pakeitimas tam tikra ašimi, šis filtravimas gali būti atliekamas viena ašimi arba keliomis, pasirenkama tvarka. Metodas tikrina ar taškas, esantis erdvėje, yra stiprioje plokštumoje, kurioje išsidėsčiusių taškų kiekis yra didesnis už ribinę vertę ar silpnoje plokštumoje, kur taškų kiekis yra mažesnis už ribinę vertę. Kai taškų kiekis plokštumoje mažesnis už ribinę vertę, šios plokštumos taškai perkeliama į stipriąją plokštumą. Rezultatuose gaunamos tik stiprios plokštumos, kur taškų kiekis jose yra didesnis arba lygus, ribinei, pereinamajai slenksčio vertei.

#### 5.1.1. Programinis kodas

Naudojant koordinatės pakeitimą stipriuotu tašku, reikia jog pagrindiniai modelio paviršiai, užimantys didžiausią erdvę, būtų lygiagretūs nagrinėjamajai ašiai, tokiu būdu modelio paviršius gaunamas be paviršių deformacijų, kurios gali sudaryti spygliuotą paviršių. Kai pagrindiniai paviršiai lygiagretūs, nagrinėjamajai ašiai, jų paviršius gaunamas glotnus. Todėl, norint pakeisti modelio padėtį, reikia jo koordinates transformuoti. Transformacijos funkcija su atitinkamais modelio kampais pavaizduota (3p.1 lentelė). Pirmiausiai sudaroma homogeninių koordinačių matrica – C. Kiekviena koordinatė yra atskirai transformuojama apie vieną iš trijų ašių. Transformacija atliekama globalioje koordinačių sistemoje. T1 matrica pasuka koordinates apie Y ašį, T2 matrica pasuka koordinates apie X ašį, T3 matrica pasuka koordinates apie Z ašį. Koordinačių pasukimo kampas valdomas atitinkamai skirtingas kintamaisiais, kiekvienoje ašyje. Apskaičiavus transformacijas, visos homogeninės koordinačių matricos koordinatės padauginamos iš transformacijos matricos, pakeičiant koordinačių padėtį erdvėje.

Toliau kviečiama funkcija, surandanti unikalių plokštumų reikšmes, nagrinėjamoje ašyje (5.1 lentelė). Filtravimo metu, atitinkamai pagal taškų kiekį erdvėje, parenkamas koordinačių tikslumas, suapvalinant koordinačių reikšmes, tai atliekama pasinaudojus „round“ funkcija „MATLAB“ aplinkoje. Po suapvalinimo taškai yra sutankinami, kadangi sumažėja jų išsisklaidymas erdvėje, kartu sumažinant unikalių plokštumų kiekį. Kintamasis „axis“ nurodo, kuria koordinačių ašimi bus atliekami filtravimai: 1 – x, 2 – y, 3 – z. „List“ masyvas sudaro unikalių plokštumų kiekį erdvėje. Pirmasis „for“ ciklas eina per visus masyvo taškus, antrasis eina per unikalių plokštumų sąrašą. Antrojo ciklo metu, tikrinama ar reikšmė nesikartoja su sąraše esančia reikšme, jeigu reikšmė nepasikartojo sąraše, tai yra unikali reikšmė, kuri pridedama prie sąrašo. Funkcijos kodas pateiktas (3p.2 lentelė).

### 5.1 lentelė Funkcijos pseudo kodas, surandantis unikalias plokštumas, nagrinėjama ašimi.

```
Oulier_listing(p)
  for i = 1 to length(p)
    b = length(list(:,1))
    a ← 0
    for j = 1 to b
      if p(i, axis) ≠ list(j, 1)
        a = a + 1
      if a == b
        list(b + 1, 1) = p(i, axis)
  return (list)
```

Priklausomai nuo taškų kiekio parenkama kintamojo „threshold“ reikšmė, kuri įvertina ar nagrinėjama plokštuma yra pagrindinė ar šalutinė. Funkcijoje, ieškančioje taškų, pašalinėse plokštumose (5.2 lentelė), pirmasis ciklas „for“ parenka sąrašo išsaugotas unikalias plokštumų reikšmes, o antrasis „for“ ciklas eina per visus nagrinėjamos ašies taškus ir tikrina ar sąrašė esantis taškas sutampa su koordinacių ašies masyvo reikšme. Jeigu sutampa, į „listing“ masyvą įtraukiama sutapusios ašies reikšmė ir indeksas bei užfiksuojamas reikšmės kiekis „a“. Antroji „if“ sąlyga tikrina ar nagrinėjamos plokštumos reikšmių kiekis neviršija pereinamosios „threshold“ reikšmės, jeigu kiekis „a“ yra didesnis už „threshold“ reikšmę, tai tuomet nagrinėjamoje plokštumoje yra pakankamas taškų kiekis ir ši plokštuma priimama kaip pagrindinė. Suveikus antrajam sąlyginiam patikrinimui – „if“, ištrinamos indekso bei sąrašo indekso reikšmės ir antrasis „for“ ciklas nutraukiamas, o „a“ reikšmė pakeičiama į nulinę vertę. Masyvo sąrašė esančios reikšmės, laikomos pašalinėmis, jeigu jų kiekis plokštumoje yra mažesnis už ribinę „threshold“ vertę, tuomet nustatoma, jog reikšmės priklauso pašalinei plokštumai, kurios turi būti panaikintos, o joje esančios taškų reikšmes, perkeltos į stipriąją plokštumą. Funkcijos kodas pateiktas (3p.3 lentelė).

### 5.2 lentelė Funkcijos pseudo kodas, surandantis reikšmes, pašalinėse plokštumose.

```
Threshold(p, list, threshold)
  sz = 0
  for i = 1 to length(list)
    temp=0;
    for j = 1 to length(p)
      if list(i, 1) == p(j, axis)
        sz = sz + 1
        listing(sz, :) = (list(i, 1), j)
        a = a + 1
        if a > threshold
          listing = listing(end - threshold + 1, :);
          sz = sz - (threshold + 1)
          break
    a ← 0
  return (listing)
```

Funkcija skaičiuoja kaimyninių taškų – „kk“, indeksus bei užfiksuoja atstumus iki jų (5.3 lentelė). Sukuriamas atstumo masyvas – „d“, kurio eilučių ilgis atitinka taškų skaičių, o stulpelių ilgis atitinka kaimynų reikšmę – kk. Visoms atstumo masyvo – „d“ reikšmėms, priimama teigiamos begalybės reikšmė. Pirmieji du „for“ ciklai paskaičiuoja euklidinį atstumą iki nagrinėjamojo taško, abu „for“ ciklai eina per visas koordinatų reikšmes. Trečiasis „for“ ciklas eina per norimą kaimyninių reikšmių, stulpelių ilgį. Trečiojo „for“ ciklo metu atliekamas tikrinimas ar gautas euklidinis atstumas yra mažesnis už to paties indekso, atstumo masyvo „d“ reikšmę, jeigu atstumas – mažesnis, reikšmė yra pakeičiama, o indeksas išsaugojamas ir ciklas nutraukiamas, jeigu ne, tikrinama sekanti kaimyninė reikšmė. Jeigu atstumas didesnis už visus nagrinėjamo indekso kaimynus, tai ši reikšmė nėra kaimyninė, ciklas baigiamas ir imama sekanti koordinatė, kurios euklidinis atstumas toliau tikrinamas su išsaugotomis reikšmėmis. Funkcijos kodas pateiktas (3p.4 lentelė).

**5.3 lentelė** Funkcijos pseudo kodas, surandantis kaimynines reikšmes.

```

Euklidinis_atstumas(p, kk)
for i = 1 to length(p)
    for j = 1 to length(p)
        
$$dt(i) = \sqrt{(p(i, 1) - p(j, 1))^2 + (p(i, 2) - p(j, 2))^2 + (p(i, 3) - p(j, 3))^2}$$

        for l = 1 to kk
            if dt < d(i, l)
                index(i, l+1) = index(i, l)
                d(i, l+1) = d(i, l)
                index(i, l) = j
                d(i, l) = dt
            break
    return (p, d, index)

```

Funkcija ieško stipriausiojo taško (5.4 lentelė), pagal kurią pakeičiama pašalinio taško koordinatė. Priklausomai nuo „k“ reikšmės, parenkamas koks kaimynų kiekis bus tikrinamas. Reikšmė – „k“ turi būti mažesnė už „kk“ reikšmę, kadangi skaičiuojant euklidinį atstumą tarp pašalinių plokštumų buvo iš anksto ieškomi stipriojo taško kaimynai. Pirmasis „for“ ciklas eina per visą sudarytą pašalinių taškų sąrašą, kuris buvo patikrintas su pereinamąja „threshold“ reikšme. Antrasis „for“ ciklas vykdomas su artimiausiomis kaimyninėmis reikšmėmis „k“, jo metu paskaičiuojamas bendras stipriojo taško kaimyninių reikšmių vidurkis – „kk“. Po šio ciklo pirmoji reikšmė priimama kaip stipriausiojo taško reikšmė. Trečiuoju „for“ ciklu lyginamos, pirmųjų kaimynų – „k“, vidurkio atstumo reikšmės. Kiekvienos trečiojo „for“ ciklo iteracijos metu tikrinama ar vidurkis yra didesnis už prieš tai buvusio kaimyno reikšmę, taip nustatant tašką, kurio pozicija tarp gretimų taškų yra artimiausia – stipriausiąjį

tašką. Pirmajam „for“ ciklui pasibaigus, atitinkamos ašies koordinatė pakeičiama surastojų, stipriausiojo kaimyninio taško atitinkama koordinatė. Tuomet taškas yra perkeliamas į stipriausiąją plokštumą, kurios taškų kiekis yra didesnis arba lygus nustatytam ribiniam slenksčiui. Funkcijos kodas pateiktas (3p.5 lentelė).

**5.4 lentelė** Funkcijos pseudo kodas, pakeičiantis koordinatę, stipriausio kaimyno ašies reikšme.

```

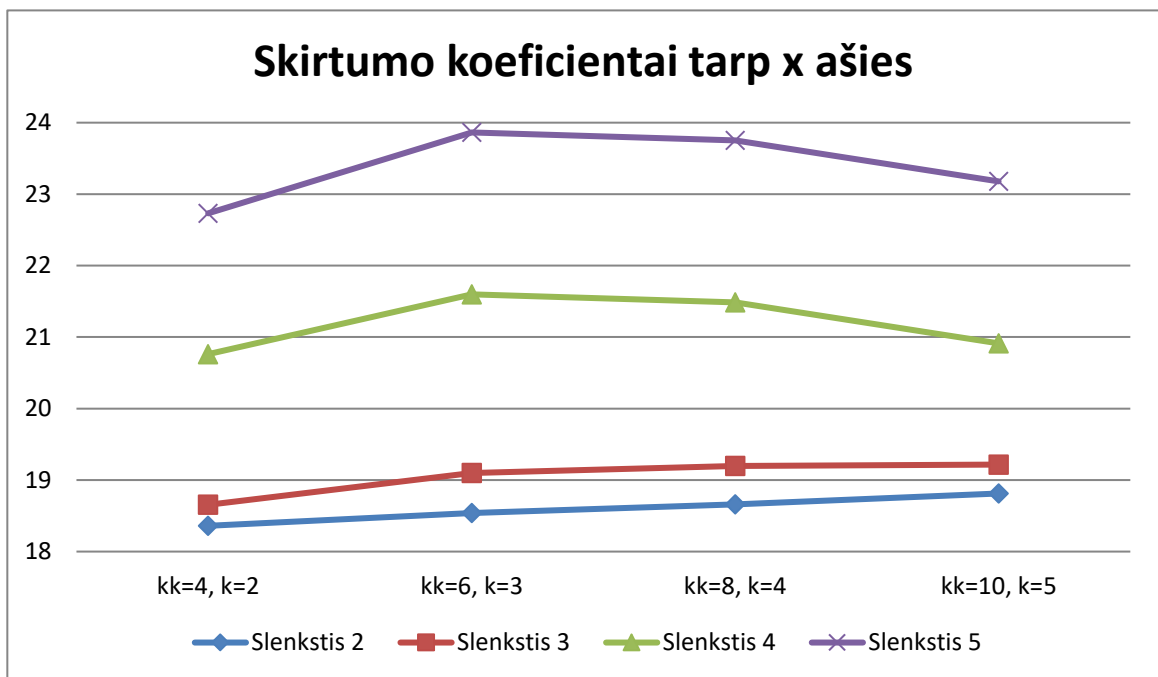
Outlier_removal(p)
 $k = \frac{kk}{2}$ 
for i = 1 to length(p)
    for j = 1 to k
         $spmean(i, j) = \frac{\text{sum}(d(\text{index}(\text{listing}(i, 2), b):))}{kk}$ 
         $sp(i, 1) = spmean(a, 1)$ 
    for l = 2 to k
        if  $sp(i, 1) > spmean(i, l)$ 
             $sp(i, 1) = spmean(i, l)$ 
             $sindex(i, 1) = l$ 
for i = 1 to length(p)
 $p(\text{listing}(i, 2), \text{axis}) = p(\text{index}(\text{listing}(i, 2), sindex(i, 1)), \text{axis})$ 
return p

```

### 5.1.2. Pavienių ašių filtravimas

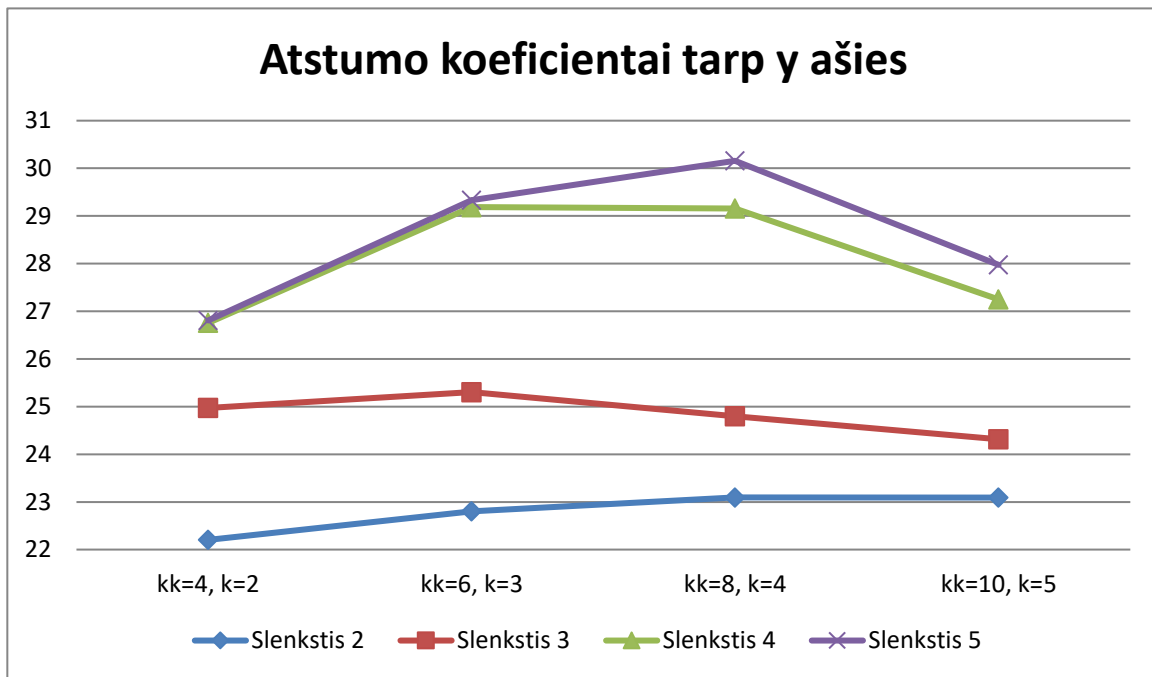
Pavienių ašių filtravimo metu, nagrinėjamos tik vienos ašies filtravimas, erdvinėje srityje. Bandymų metu nagrinėjamos visos trys ašys, kur yra keičiami koeficientai: ribinė slenksčio reikšmė, kaimyninis koeficientas – k ir stipriųjų kaimynų nustatymo koeficientas – kk.

Pakeičiant pašalinių taškų koordinates x ašimi, kai didinamas slenksčio ribinis koeficientas bei kaimyninių taškų kiekis, visų bandymų metu, skirtumas didėja, filtravimo duomenys x ašimi, atvaizduoti (3p.6 lentelė). Priėmus slenksčio ribą 4 ir 5, o kaimyninių reikšmių vertes 4 ir 5, skirtumo reikšmės šiek tiek sumažėja, tačiau visų koeficientų reikšmės mažiausios, kai priimamas mažiausias kaimyninių taškų kiekis k=2, atliekant filtravimą x ašimi. Visų bandymų metu, trikdžio  $s1 = 2.6831$  reikšmė išlieka pastovi, su mažiausia kaimynine reikšme, minimalus antrasis trikdys gaunamas  $s2 = 2.1788$ . Pats mažiausias skirtumas nustatytas, kai slenksčio riba=2, k=2, kk=4, tuomet  $skirtumas = 18.3606$ . Vizualiai filtruotas modelis su šiais parametrais atvaizduotas (3p.7 lentelė). Grafiškai skirtingų slenksčių reikšmių su skirtingomis kaimyninėmis reikšmėmis, skirtumai, atvaizduoti (5.1 pav.). Grafiko pateiktuose duomenyse matoma, jog didinant slenksčio ribinį koeficientą, atitinkamai didėja skirtumo koeficientas, su mažesniu ribiniu koeficientu 2 ir 3, didinant kaimynines reikšmes, skirtumas didėja. Tuo tarpu, su didesniu ribiniu koeficientu 3 ir 4, didinant kaimynines reikšmes, skirtumo vertės linkusios mažėti.



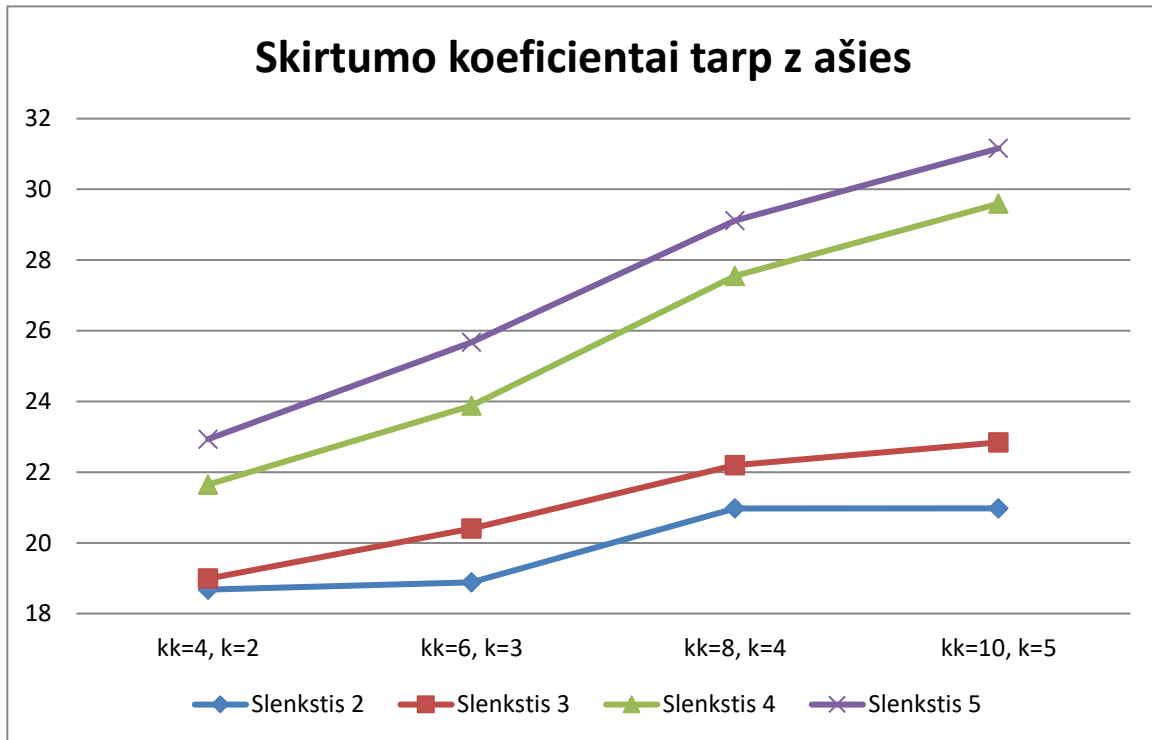
5.1 pav. Skirtumo koeficientai, nagrinėjant x ašį.

Pakeičiant pašalinių taškų koordinatės y ašimi, didinant ribinį koeficientą bei kaimyninių taškų kiekį, visų bandymų metu, skirtumas yra linkęs didėti, tačiau penktosios kaimyninės reikšmės metu, skirtumo koeficientas, visų bandymų metu, sumažėja. Filtravimo duomenys y ašimi, atvaizduoti (3p.8 lentelė). Skirtingų koeficientų reikšmės mažiausios, kai priimamas mažiausias kaimyninių taškų kiekis  $k=2$ , išskyrus, kai slenksčio riba=3, tuomet skirtumas mažiausias, kai  $k=5$ . Nagrinėjamas trikdys  $s_1 = 0.4138$ , sumažėja kai slenksčio riba 4 ir 5. Trikdys  $s_2 = 3.3865$  išlieka pastovus su visomis skirtingomis koeficientų reikšmėmis. Pats mažiausias skirtumo koeficientas *skirtumas* = 18.3606, nustatytas, kai slenksčio riba=4,  $k=2$ ,  $kk=4$ . Vizualiai filtruotas modelis su šiais parametrais atvaizduotas (3p.9 lentelė). Grafiškai skirtingų slenksčių reikšmių, su skirtingomis kaimyninėmis reikšmėmis, skirtumai atvaizduoti (5.2 pav.). Grafiko pateiktuose duomenyse matoma, jog didinant slenksčio ribinį koeficientą, atitinkamai didėja skirtumo koeficientas, kai ribinis koeficientas lygus – 3, 4 ir 5, didinant kaimynines reikšmes, skirtumo vertės mažėja. Kai slenksčio riba 2, didinant kaimynines reikšmes, skirtumo koeficientas nusistovi.



5.2 pav. Skirtumo koeficientai, nagrinėjant y ašį.

Pakeičiant pašalinių taškų koordinates z ašimi, kai didinamas ribinis koeficientas bei kaimyninių taškų kiekis, visų bandymų metu skirtumo koeficientas didėja. Filtravimo duomenys z ašimi, atvaizduoti (3p.10 lentelė). Visų skirtumų koeficientų reikšmės mažiausios, kai priimamas mažiausias kaimyninių taškų kiekis  $k=2$ , nagrinėjant z ašyje. Trikdžiai  $s_1$  ir  $s_2$  nekinta, visų nagrinėtų reikšmių metu, jie išlieka:  $s_1 = 2.5740$ ,  $s_2 = 2.7266$ . Pats mažiausias skirtumo koeficientas  $skirtumas = 18.6789$ , nustatytas, kai slenksčio riba=4,  $k=2$ ,  $kk=4$ . Vizualiai filtruotas modelis su šiais parametrais atvaizduotas (3p.11 lentelė). Skirtingų slenksčių reikšmių su skirtingomis kaimyninėmis reikšmėmis, skirtumai grafiškai atvaizduoti (5.3 pav.). Grafiko pateiktuose duomenyse matoma, jog didinant slenksčio ribinį koeficientą atitinkamai didėja skirtumo koeficientas, kai ribinis koeficientas lygus – 3, 4 ir 5. Didinant kaimynines reikšmes, skirtumo vertės didėja, o kai slenksčio riba yra lygi 2, didinant kaimynines reikšmes, tuomet skirtumo koeficientas nusistovi.



5.3 pav. Skirtumo koeficientai, nagrinėjant z ašį.

**Išvados:** filtruojant modelį viena ašimi, mažiausias skirtumo koeficientas bei  $s_2$  trikdžio gautas x ašyje, atitinkamai:  $skirtumas = 18.3606$ ,  $s_2 = 2.1788$ . Mažiausias  $s_1$  trikdžius gautas filtruojant y ašimi:  $s_1 = 0.4138$ , kai buvo priimtas didesnis kaimynų kiekis – 3 ir 4. Todėl filtravimas priklauso nuo taškų kiekio bei išsidėstymo, filtruojant modelį skirtingomis ašimis. Filtruojant viena ašimi, nėra nustatomas optimalus rezultatas, kadangi taškų išsidėstymas pakinta tik vienos ašies atžvilgiu. Toks metodas tinka, kai filtruojamas objektas skenuojamas daug kartų, tuomet susidaro taškų sutankėjimai, kur objektas geriau išsiskiria nuo pašalinių taškų, tuomet pašaliniai taškai išsidėstę retesniu tankumu ir yra lengviau išfiltruojami.

### 5.1.3. Filtravimas trimis ašimis

Atliekant filtravimus, kai nagrinėjamos trys ašys, pradedant filtruoti x ašimi, didinant kaimynų skaičių bei ribinį slenksčio koeficientą, skirtumo koeficiento reikšmės nuosekliai didėja. Filtravimo duomenys, kai pradedama filtruoti x ašimi, atvaizduoti (3p.12 lentelė). Kai ribinė plokštumos reikšmė priimama 4 bei 5, o kaimynų skaičius lygus  $k=2$ , filtruojant x, z, y ašimis, tuomet pašalinio trikdžio koeficientas  $s_1$  sumažėja nuo  $s_1 = 2.7132$  iki  $s_1 = 0.6751$ . Vizualiai filtruotas modelis su šiais parametrais atvaizduotas (3p.13 lentelė).  $S_2$  reikšmė išlieka stabiliai apie  $s_2 \approx 3$ , su visomis skirtingomis parametru reikšmėmis. Mažiausios skirtumo koeficiento reikšmės gaunamos: filtruojant x, y, z ašių eiliškumo tvarka, tuomet  $skirtumas = 25.6036$ , kai slenksčio riba 2 ir kaimynų skaičius lygus  $k=2$ . Filtruojant x, z, y ašių eiliškumu,  $skirtumas = 24.9825$ , kai slenksčio riba 2 ir kaimynų skaičius lygus  $k=3$ .



Filtruojant trimis ašimis, kai pradėdama filtruoti y ašimi, didinant kaimynų skaičių bei ribinį slenksčio koeficientą, skirtumo koeficiento reikšmės nuosekliai didėja. Filtravimo duomenys, kai pradėdama filtruoti y ašimi, atvaizduoti (3p.14 lentelė). Kai ribinė plokštumos reikšmė priimama 4 bei 5, o kaimynų skaičius lygus  $k=2$ , tuomet pašalinio trikdžio koeficientas  $s_1$  sumažėja nuo  $s_1 = 2.6560$  iki  $s_1 = 0.1012$ . Vizualiai filtruotas modelis su šiais parametrais atvaizduotas (3p.15 lentelė).  $S_2$  reikšmė išlieka stabiliai apie  $s_2 \approx 3$ , su visomis skirtingų parametrų reikšmėmis. Abiejų filtravimo ašių eiliškumu,  $skirtumas = 24.418$  yra mažiausias kai slenksčio riba 2 ir kaimynų skaičius lygus  $k=2$ .

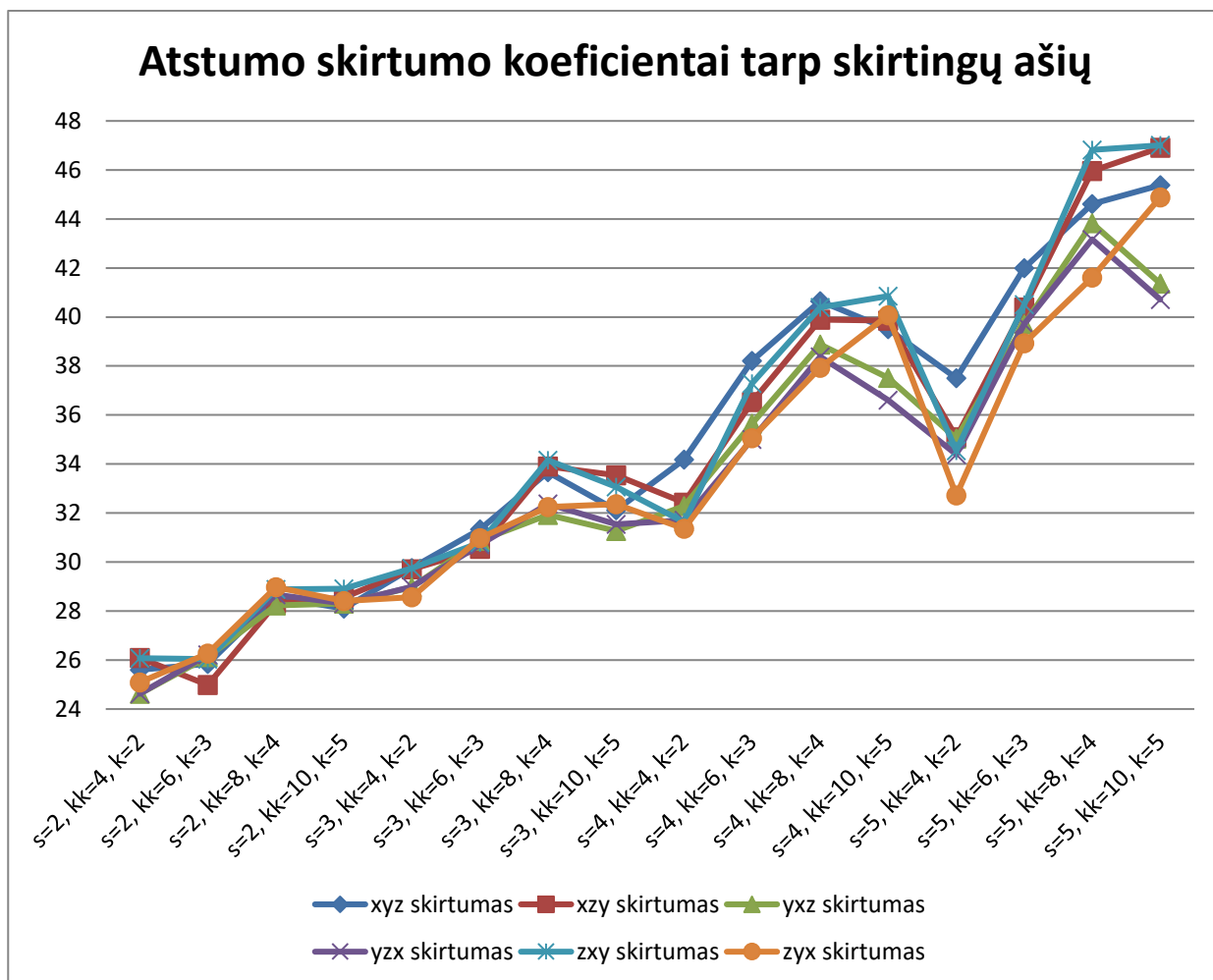
Filtruojant trimis ašimis, kai pirmiausiai pradėdama filtruoti z ašimi, didinant kaimynų skaičių bei ribinį slenksčio koeficientą, skirtumo koeficiento reikšmės nuosekliai kyla. Filtravimo duomenys, kai pradėdama filtruoti z ašimi, atvaizduoti (3p.16 lentelė). Kai ribinė plokštumos reikšmė priimama 4 bei 5, o kaimynų skaičius lygus  $k=2$ , tuomet pašalinio trikdžio koeficientas  $s_1$  sumažėja nuo  $s_1 = 2.7117$  iki  $s_1 = 0.6751$ , kai filtruojama z, x, y ašimis, o filtruojant z, y, x ašių eiliškumu,  $s_1 = 0.1012$ . Vizualiai filtruotas modelis su šiais parametrais atvaizduotas (3p.17 lentelė).  $S_2$  reikšmė išlieka stabiliai apie  $s_2 \approx 3$ , renkantis skirtingus filtravimo parametrus. Mažiausios skirtumo koeficiento reikšmės gaunamos: filtruojant z, x, y ašių eiliškumu, tuomet  $skirtumas = 26.0387$ , kai slenksčio riba 2 ir kaimynų skaičius lygus  $k=3$ . Filtruojant z, y, x ašių eiliškumu, kai slenksčio riba 2 ir kaimynų skaičius lygus  $k=2$ , tuomet  $skirtumas = 25.0915$ .

Lyginant atstumo skirtumo koeficientą tarp ašių (5.5 lentelė), kai naudojami tie patys filtravimo parametrai, buvo nustatyta, jog mažiausią skirtumo koeficientą, turėjo ašys: z, y, x – 6 kartus, y, z, x – 4 kartus, y, x, z – 4 kartus, x, z, y – 2 kartus, x, y, z – 1 kartą, o ašys z, x, y neturėjo mažiausio skirtumo koeficiento nei karto. Nagrinėjant filtrų parametrus, su skirtingais slenksčiais, filtravimo skirtumo reikšmės buvo mažiausios, kai  $kk=4$  ir  $k=2$ . Didinant slenksčio parametro reikšmę nuo 4, skirtumo koeficientas tarp skirtingų ašių padidėja, todėl didinant slenksčio ribinę reikšmę, ašių eiliškumo tvarka, turi didesnę įtaką, gaunamam skirtumo koeficientui.

**5.5 lentelė** Skirtingų ašių rinkinių, skirtumo koeficientai

	xyz skirtumas	xzy skirtumas	yxz skirtumas	yzx skirtumas	zxy skirtumas	zyx skirtumas
s=2, kk=4, k=2	25.6036	26.0936	24.6218	24.6218	26.0733	25.0915
s=2, kk=6, k=3	25.8378	24.9825	26.1215	26.2213	26.0387	26.2675
s=2, kk=8, k=4	28.5685	28.3074	28.2359	28.6610	28.8837	28.9782
s=2, kk=10, k=5	28.1078	28.5487	28.3025	28.3025	28.9145	28.4078
s=3, kk=4, k=2	29.7625	29.6998	28.9744	28.9814	29.7473	28.5595
s=3, kk=6, k=3	31.3387	30.5450	30.8697	30.7011	30.7981	30.9745
s=3, kk=8, k=4	33.6747	33.9009	31.9198	32.3702	34.1543	32.2426
s=3, kk=10, k=5	32.1153	33.5432	31.2667	31.5461	33.0622	32.3629
s=4, kk=4, k=2	34.1836	32.4302	32.2886	31.7009	31.6348	31.3551
s=4, kk=6, k=3	38.2092	36.5297	35.6454	35.0030	37.2933	35.0587
s=4, kk=8, k=4	40.6466	39.8922	38.8752	38.3722	40.3925	37.9310
s=4, kk=10, k=5	39.5115	39.8585	37.5199	36.6128	40.8505	40.0658
s=5, kk=4, k=2	37.5007	35.0944	35.0472	34.3906	34.5373	32.7233
s=5, kk=6, k=3	42.0001	40.3916	39.6950	39.7088	40.5049	38.9309
s=5, kk=8, k=4	44.6085	45.9613	43.8292	43.1691	46.8167	41.6069
s=5, kk=10, k=5	45.3805	46.9092	41.3626	40.7109	47.0077	44.8825

Grafiškai interpretuotame, atstumo skirtumo koeficiento, atvaizdavime (5.4 pav.), filtruojant skirtingomis ašimis, pastebima, jog z, y, x filtravimo tvarka yra viena iš aukštesniųjų kreivių, lyginant su kitomis ašimis – pirmuosiuose trijose bandymuose, o toliau tampa viena iš žemesniųjų, net 6 skirtingų reikšmių metu, skirtumo koeficientas, šia nagrinėjama tvarka, yra mažiausias. Sekančios dvi žemiausios kreivės yra y, z, x bei y, x, z, lyginant pagal bendra skirtumo atstumo koeficientą. Kai parenkama slenksčio reikšmė 3, o kaimynų skaičius  $kk=6$  ir  $k=3$ , tuomet skirtumų reikšmės pereina 30 skaičiaus ribą. Kai slenkstis  $=5$ ,  $kk=4$  ir  $k=2$ , tuomet visų ašių skirtumas sumažėja, kreivė nusileidžia žemyn, todėl naudojant didesnes reikšmes, galima gauti skirtumo koeficiento sumažėjimus.



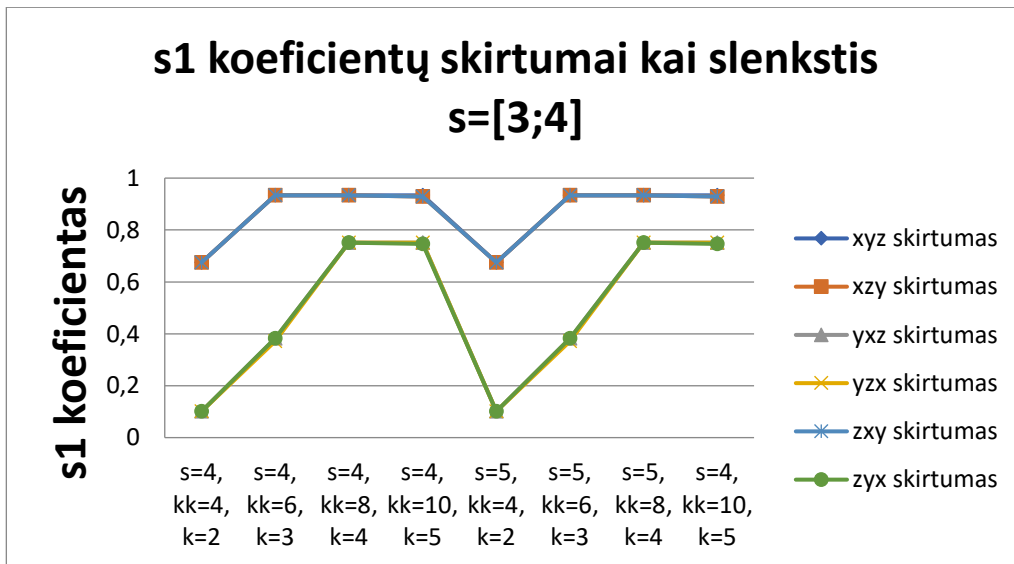
5.4 pav. Skirtumo koeficientai su skirtingais ašių rinkiniais.

Nagrinėjant trikdžių koeficientą  $s_1$  (5.6 lentelė), su pirmomis 2 slenksčio reikšmėmis gauti koeficientai yra panašūs, tarp jų svyruoja nuo  $s_1=2.6560$  iki  $s_1=2.7132$ , kur skirtumas tarp šių koeficientų yra 0.0572, taip pat didžioji dalis reikšmių yra vienodos. Kai nagrinėjami bandymai nuo 4 slenksčio parametrinių reikšmių, tuomet ženkliai sumažėja koeficientas  $s_1$ . Mažiausias gaunamas  $s_1$  trikdžio koeficientas yra  $s_1 = 0.1012$ . Visais nagrinėtais atvejais, nuo slenksčio reikšmės  $s = 3$ , koeficientas  $s_1$  buvo mažiau už 1,  $s_1 < 1$ . Lyginant su pradiniu sugeneruotu nuokrypiu,  $pradinis\_s_1 = 2.6500$ , geriausiu nagrinėjamu atveju, skirtumas sumažinamas:  $2.6500 - 0.1012 = 2.5488$ .

5.6 lentelė Skirtingų ašių, s1 koeficientų lentelė

Parametrai	s1 xyz	s1 xzy	s1 yxz	s1 yzx	s1 zxy	s1 zyx
s=2, kk=4, k=2	2.6560	2.6560	2.6560	2.6560	2.6560	2.6560
s=2, kk=6, k=3	2.7132	2.7132	2.7132	2.7132	2.7132	2.7132
s=2, kk=8, k=4	2.7132	2.7132	2.7117	2.7132	2.7132	2.7132
s=2, kk=10, k=5	2.7117	2.7117	2.7117	2.7117	2.7117	2.7117
s=3, kk=4, k=2	2.6560	2.6560	2.6560	2.6560	2.6560	2.6560
s=3, kk=6, k=3	2.7132	2.7132	2.6560	2.6560	2.7132	2.6560
s=3, kk=8, k=4	2.7132	2.7132	2.6560	2.6560	2.7132	2.6560
s=3, kk=10, k=5	2.7132	2.7117	2.6560	2.6560	2.7117	2.7117
s=4, kk=4, k=2	0.6751	0.6751	0.1012	0.1012	0.6751	0.1012
s=4, kk=6, k=3	0.9336	0.9336	0.3823	0.3714	0.9336	0.3823
s=4, kk=8, k=4	0.9336	0.9336	0.7512	0.7512	0.9336	0.7512
s=4, kk=10, k=5	0.9336	0.9292	0.7512	0.7512	0.9292	0.7458
s=5, kk=4, k=2	0.6751	0.6751	0.1012	0.1012	0.6751	0.1012
s=5, kk=6, k=3	0.9336	0.9336	0.3823	0.3714	0.9336	0.3823
s=5, kk=8, k=4	0.9336	0.9336	0.7512	0.7512	0.9336	0.7512
s=5, kk=10, k=5	0.9336	0.9292	0.7512	0.7512	0.9292	0.7458

Grafiškai atvaizduoti s1 koeficientų skirtumai, nuo slenksčio 3 ribos (5.5 pav.). Efektyviausi išfiltruotos ašys: y, x, z; y, z, x ir z, y, x, kurių mažiausios reikšmės  $s_1 = 0.1012$  pasiektos su dviem skirtingais parametrais. Ašių x, y, z; x, z, y ir z, x, y s1 koeficiento reikšmės visais bandytais atvejais, nepasiekia 1 reikšmės, tačiau lyginant mažiausias reikšmes, ašių x, y, z; x, z, y ir z, x, y mažiausios reikšmės skiriasi 6 kartais nuo kitų trijų ašių.



5.5 pav. S1 koeficientai su skirtingais ašių rinkiniais.

**Išvados:** Nagrinėjant pavienių ašių filtravimą, x ašimi rastas mažiausias s2 trikdžio koeficientas  $s_2 = 2.1778$ , kuris buvo mažiausias, net lyginant su pradine koeficiento reikšme  $s_2 = 2.7432$ . Mažiausia kombinuoto filtravimo reikšmė gauta  $s_2 = 2.9557$ , kai buvo filtruojama z, x, y ašių eiliškumu. Filtruojant taškus y, x, z ašimi bei y, z, x mažiausia skirtumo koeficiento reikšmė buvo: *skirtumas* = 24.6218. Mažiausia s1 koeficiento reikšmė gauta filtruojant abiem atvejais kai pirmoji ašis parenkama

y bei z, y, x, tuomet  $s_1 = 0.1012$ , lyginant su pradine reikšme  $s_1 = 2.6500$ . Koeficientas  $s_1$  buvo gautas mažiausias, kai nagrinėjama viena ašis y, tuomet  $s_1 = 0.4138$ . Pagal pirminį filtravimą, filtruojant viena ašimi, galima nustatyti, kuria ašimi reikėtų pradėti filtruoti, kombinuojant filtruojamas ašis. Filtravimo rezultatai tiesiogiai priklauso nuo filtruojamų ašių eiliškumo, filtruojamas modelis išfiltruojamas efektyviau, kai modelio pozicija yra lygiagreti nagrinėjamai ašiai. Didinant filtravimo parametrus didėja bendras skirtumo koeficientas, todėl filtruojant efektyviau naudoti mažą slenksčio ribą, pagal kurią nustatoma pašalinė plokštuma. Filtravimas efektyviausiai panaudotas prie  $s_1$  trikdžio, kaip priimamos didesnės ribinės slenksčio reikšmės, kadangi trikdžis  $s_3$ , kur tankiai išsidėstę taškai, trukdė aptikti pašalinę plokštumą, kurioje buvo trikdžis  $s_1$ .

## 5.2. Statistinis pašalinio taško pakeitimas

### 5.2.1. Programinis kodas

Funkcija normalizuoja duomenų reikšmes į skaičių sritį  $[0;1]$ , sumažinant duomenų sklaidą bei panaikinant neigiamas vertes (5.7 lentelė). Pirmas „for“ ciklas eina per visus taškus, antrasis ciklas „for“ vykdomas per ašių koordinates taip normalizuojant kiekvienos koordinatės ašies reikšmę. Funkcijos kodas pateiktas (4p.1 lentelė).

5.7 lentelė Normalizavimo pseudo kodas.

```

Normalizavimas(p)
min ← min(p)
max ← max(p)
  for i = 1 to length(p)
    for j = 1 to 3
       $p'(i, j) = \frac{p(i, j) - \min}{\max - \min}$ 
  return p'

```

Sekanti funkcija priartina nagrinėjamą pašalinę reikšmę, esančią skirstinio teigiamoje  $\sigma_3$  dalyje, prie kaimyninio taško, pagal statistinį vidurkį (5.8 lentelė). Pirmiausiai pasinaudojus (5.3 lentelė) funkcija, randami artimiausi kiekvieno taško kaimynai, kur taip pat išsaugomos kaimynų atstumo reikšmės. Tuomet taško apkrovimo funkcijoje (5.8 lentelė), paskaičiuojamas kaimyninių reikšmių bendras vidurkis – „vid“, kuris naudojamas palyginant nagrinėjamojo pašalinio taško atstumą iki artimiausio kaimyninio taško. Atstumo reikšmės yra išrikiuojamos pagal dydį, nuo mažiausios iki didžiausios, kintamasis „I“ nurodo išrikiuotų taškų indeksų pozicijas, o kintamasis „D“ nurodo taškų atstumų reikšmes. Turint išrikiuotą sąrašą, atliekamas normaliojo skirstinio kvartilio išskyrimas, kur nustatomos teigiamos  $\sigma_3$  reikšmės, toliausiai nutolusios nuo vidurkio. Pirmasis ciklas „for“ eina per nagrinėjamojo kvartilio ilgį, tuomet ciklas „while“ tikrina ar „vidvisu“, bendras kaimyninių taškų

atstumo koeficientas, yra mažesnis už nagrinėjamojo taško atstumą iki kaimyno. Jeigu taško vidurkio koeficientas didesnis, naudojamas antras ciklas „for“, kuris tikrina kiekvienos ašies koordinatės dydį. Kai koordinatės dydis mažesnis už kaimyninio taško, tos pačios ašies, koordinatę, skirtumo koeficientas pridamas prie koordinatės, jeigu atstumas didesnis už kaimyninę reikšmę, koordinatės dydis sumažinamas koeficientu, priartinant ją prie kaimyninės reikšmės. Taip naudojant atstumo santykį, priklausomai nuo skirtumo tarp koordinatės ašių, reikšmė yra artinama prie kaimyninės koordinatės reikšmės. Kai „while“ ciklas baigiamas, paskaičiuojamas dabartinis atstumas iki artimiausio kaimyno ir vėl vykdomas loginis patikrinimas, lyginantis vidurkio reikšmę tarp dabartinio atstumo. Funkcijos kodas pateiktas (4p.2 lentelė).

5.8 lentelė Taško apkrovimo pseudo kodas.

```

Stat( $p'$ ,  $d$ )
 $vid \leftarrow \text{mean}(d)$ 
 $[D, I] \leftarrow \text{sort}(d)$ 
 $[D, I] \leftarrow (\text{length} * 0.997 : \text{length}(I))$ 
  for  $i = 1$   $\text{length}(I)$ 
    while  $vid < D$ 
       $koef = \frac{D}{vid}$ 
      for  $j = 1$  to 3
        if  $p'(I(i), j) < p'(\text{index}(I(i), 1), j)$ 
           $p'(I(i), j) = p'(I(i), j) + \frac{p'(\text{index}(I(i), 1), j) - p'(I(i), j)}{koef}$ 
        elseif  $p'(I(i), j) > p'(\text{index}(I(i), 1), j)$ 
           $p'(I(i), j) = p'(I(i), j) + \frac{p'(\text{index}(I(i), 1), j) - p'(I(i), j)}{koef}$ 
      Euklidinis_atstumas( $p'(I(i))$ )
      return ( $D(I(i))$ )
    return  $p'$ 

```

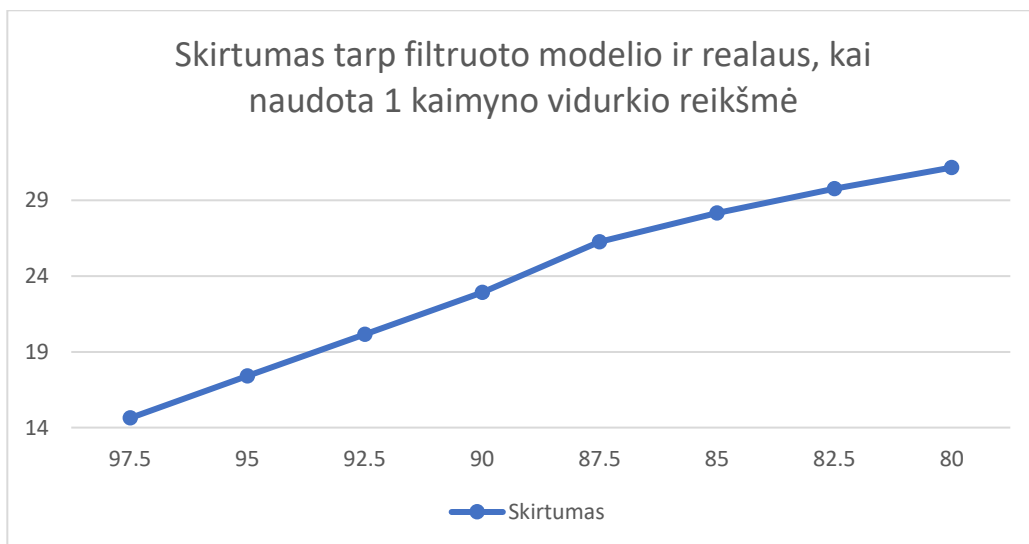
Atnaujinus standartinio skirstinio pašalinių taškų reikšmes, visoms koordinatės reikšmėms taikomas atvirkštinis normalizavimas. Pirmasis „for“ ciklas eina per taškų kiekį, antrasis „for“ ciklas per ašių koordinates, taip gražinant koordinates į tikruosius atstumus. Funkcijos kodas pateiktas (4p.3 lentelė).

### 5.9 lentelė Funkcijos pseudo kodas, grąžinantis iš normalizavimo.

```
Grazinimas_normalizavimo( $p', p$ )  
min ← min( $p$ )  
max ← max( $p$ )  
  for  $i = 1$  to length( $p$ )  
    for  $j = 1$  to 3  
       $p(i, j) = p(i, j) \cdot (max - min) + min$   
  return  $p$ 
```

#### 5.2.2. Statistinio filtravimo bandymai

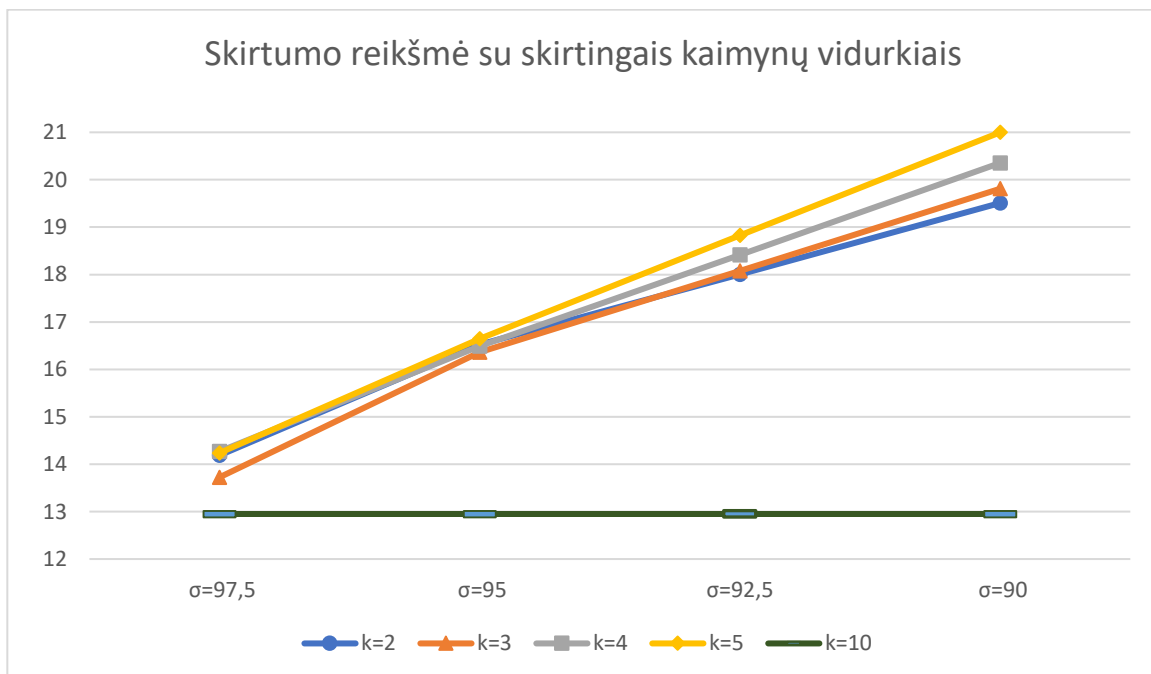
Filtruojant taškus, kai lyginimo koeficientas buvo ieškomas tik pagal pirmąją kaimyno reikšmę, kur vidurkio atstumas lygus  $d = 6.1263 \cdot 10^{-4}$ , optimaliausios trikdžių reikšmės, buvo randamos naudojant antrojo standartinio nuokrypio reikšmę  $\sigma_2 = 97.5\%$ , tuomet trikdžių koeficientų reikšmės lygios:  $s_1 = 0.3286$ ,  $s_2 = 3.1458$ ,  $skirtumas = 14.6279$  (4p.4 lentelė). Eksperimentiniuose tyrimuose, standartinio nuokrypio reikšmė parenkama ne  $\sigma_3 = 99.7\%$ , o mažesnė  $\sigma_2 = 97.5\%$ , dėl nagrinėjamų duomenų skaičiaus. Didinant parenkamų pašalinių taškų skaičių, bendras skirtumas nuosekliai didėja, taip mažėjant skenuojamojo modelio tikslumui, kadangi yra didesnė dalis modelio taškų yra pakeičiama. Grafiškai skirtumo pokytis atvaizduojamas (5.6 pav.), didinant  $\sigma$  reikšmes, skirtumas tolydžiai kyla. Mažiausias skirtumas gaunamas, kai  $\sigma = 97.5$ . Filtruotas modelis su statistinio filtravimo reikšmėmis  $\sigma_2 = 97.5\%$  ir  $\sigma_2 = 87.5\%$ , pavaizduotas (4p.5 lentelė).



5.6 pav. Skirtumo koeficientas, naudojant 1 kaimyno vidurkio reikšmę

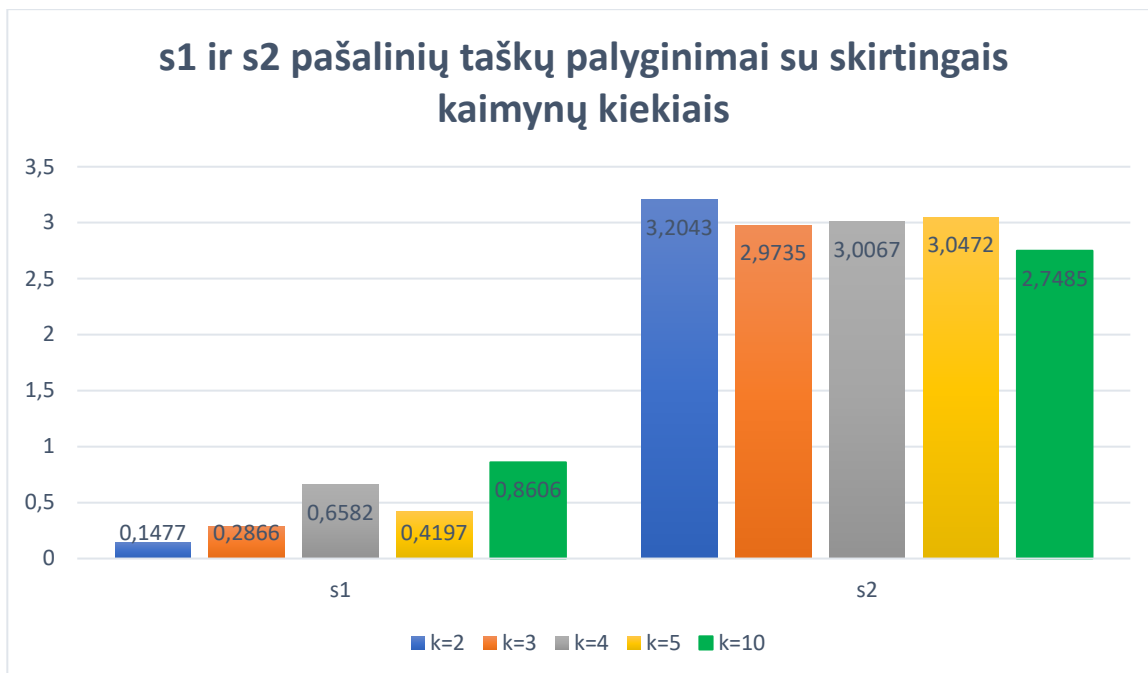
Mažiausias skirtumo koeficientas, statistiniu metodu filtruojamų taškų, kai atstumas iki kaimyno lyginamas su bendromis vidurkio reikšmėmis (4p.6 lentelė), gaunamas, kai  $\sigma = 97.5\%$ . Analizuojant skirtingas kaimynines reikšmes tarpusavyje, kai didinamas kaimynų kiekis, skirtumo koeficientas gali kilti bei leistis, nėra konkretaus pastovumo. Kai parenkamas parametras  $\sigma = 97.5\%$ , lyginant  $k=2$  su

$k=3$ , skirtumas sumažėja nuo  $skirtumas = 14.1878$  iki  $skirtumas = 13.7241$ , priėmus parametą  $k=4$ , skirtumo koeficientas pakyla iki  $skirtumas = 14.2706$ . Todėl skirtumo reikšmės sunku tiesiogiai koreliuoti su lyginamų kaimynų skaičiumi, tačiau yra matoma skirtumo augimo tendencija, didinant  $\sigma$  reikšmę. Skirtumo koeficiento augimas didėja, kadangi didesnė taškų imtis yra paveikiama filtro, kai didinama  $\sigma$  reikšmė. Mažiausias  $s1$  trikdžio koeficientas gaunamas  $s1 = 0.1477$ , kai priimami du kaimynai, tuomet bendra vidurkio atstumo reikšmė, už kurią turi būti mažesnis nagrinėjamas taškas iki kaimyno, yra  $d = 6.8048 \cdot 10^{-4}$ . Mažiausias bendras skirtumas gaunamas, kai kaimyninė reikšmė  $k=10$ , o vidurkio atstumas  $d = 24 \cdot 10^{-4}$ , tuomet  $skirtumas = 12.9497$ . Nagrinėjant skirtumo koeficientą grafiškai (5.7 pav.). Kai parenkama reikšmė  $k=10$ , skirtumas nepakinta prie skirtingų parametų. Kai parenkama reikšmė  $k=3$ , skirtumo reikšmės kyla, bet yra skirtumo koeficientai yra žemiausi, lyginant su  $k=4$  bei  $k=5$  parametų filtravimais, dviejų pirmųjų  $\sigma$  reikšmių atveju. Priėmus  $k=5$ , gauti didžiausi skirtumo koeficientai.



5.7 pav. Skirtumo koeficientas, naudojant bendrą kaimyno vidurkio reikšmę

Lyginant  $s1$  ir  $s2$  koeficiento reikšmes su skirtingais kaimynų vidurkais, grafiškai atvaizduojamos vertės (5.8 pav.).  $S1$  koeficiento reikšmė mažiausia prie kaimyninių reikšmių  $k = 2$ , tuomet  $s1 = 0.1477$ , lyginant su pradine trikdžio  $s1$  reikšme  $s1 = 2.6500$ .  $S2$  koeficiento mažiausia reikšmė yra gaunama prie kaimyninių reikšmių  $k = 10$ , tuomet  $s2 = 2.7485$ , kai pradinė trikdžio  $s2$  reikšmė  $s2 = 2.7432$ . Todėl lyginant  $s1$  reikšmę, visais eksperimento atvejais, pašalinis taškas buvo priartintas prie pradinės reikšmės. Lyginant  $s2$  reikšmę, pašalinis taškas  $s2$ , buvo priartintas prie artimiausių kaimynų, kurie atstumu buvo arčiau nei pradinė reikšmė, todėl  $s2$  koeficientas padidėjo. Filtruotas modelis su parametru  $\sigma = 97.5\%$ , bei skirtingais kaimynais atvaizduotas (4p.7 lentelė).



5.8 pav. Trikdžių s1 ir s2 palyginimas, naudojant bendrą kaimyno vidurkio reikšmę

**Išvados:** Pašaliniai taškai efektyviau išfiltruojami, kai pasirenkama bendra kaimynų vidurkio reikšmė, lyginant su pirmu, arčiausiu kaimynų, vidurkiu. Kadangi su pirmuoju arčiausiu kaimynu mažiausios gautos reikšmės:  $s1 = 0.3286$ ,  $s2 = 3.1458$  bei  $skirtumas = 14.6279$ , o su bendra kaimynų vidurkio reikšme, reikšmės lygios:  $s1 = 0.1477$ ,  $s2 = 2.7485$  bei  $skirtumas = 12.9497$ . Kadangi statistinis pašalinių taškų filtravimas skirtas filtruoti pašalinius taškus, našiausiai pašaliniai taškai išfiltruojami, kai naudojama didžiausia  $\sigma$  reikšmė. Efektyviausiai išfiltruota pirmojo trikdžio reikšmė, kai lyginamasis kaimynų skaičius  $k=2$ , tuomet  $s1 = 0.1477$ . Kai lyginamasis kaimynų skaičius  $k=10$ , tuomet  $s2 = 2.7485$  ir  $skirtumas = 12.9497$  yra mažiausi. Tačiau lyginant su pradinėmis trikdžių reikšmėmis  $s2 = 2.7432$  ir  $skirtumas = 11.0650$ , reikšmės išlieka didesnės, o  $s1$  reikšmė  $s1 = 0.1477$ , ženkliai mažesnė už  $s1 = 2.6500$ . Filtravimo metodas veikia optimaliausiai, kai pašaliniai taškai yra nutolę erdvėje, nuo tiriamojo objekto paviršiaus.

### 5.3. Valdomas taškų filtravimas

#### 5.3.1. Programinis kodas

Pirmiausiai funkcija nustato artimiausius kaimyninius taškus (5.3 lentelė). Pasinaudojus centroidės funkcija (5.10 lentelė), apskaičiuojamos centroidinės, kiekvieno taško reikšmės. Pirmasis „for“ ciklas eina per visas taško reikšmes, antrasis „for“ ciklas skaičiuoja to taško x, y, z koordinatų reikšmes, priklausomai nuo parinkto kaimynų kiekio – k. Apskaičiavus visų koordinatų reikšmes, jos dalinamos iš kaimynų kiekio, nustatant geometrinę taškų reikšmę – centroidę. Funkcijos kodas pateiktas (5p.1 lentelė).



**5.10 lentelė** Pseudo kodas apskaičiuojantis centroidines reikšmes.

```
cent(p, k)  
  for i = 1 to length(p)  
    for j = 1 to k  
       $c(i, 1) = c(i, 1) + p(\text{index}(i, j), 1)$   
       $c(i, 2) = c(i, 2) + p(\text{index}(i, j), 2)$   
       $c(i, 3) = c(i, 3) + p(\text{index}(i, j), 3)$   
  
   $c = \frac{c}{k}$   
return c
```

Sekančios funkcijos metu (5.11 lentelė), apskaičiuojamas linijinio modelio koeficientas  $a_i$ , kur pirmasis „for“ ciklas vykdomas per visus objekto taškus, o antrasis „for“ ciklas vykdomas per kaimynines taško reikšmes. Pirmiausiai randamas kvadratinis skirtumas tarp kaimyninių koordinačių bei centroidžių, kur kiekvieno ciklo metu, sumuojamos gautos reikšmės. Apskaičiavus bendras sumines reikšmes, apskaičiuojamas vidurkis, padalinant suminių reikšmių koeficientus iš kaimynų kiekio. Galiausiai gautos suminės reikšmės naudojamos apskaičiuoti linijinio modelio  $a_i$  koeficientus, naudojant patikslinančią reikšmę  $\varepsilon$ . Funkcijos kodas pateiktas (5p.2 lentelė).

**5.11 lentelė** Pseudo kodo funkcija, randanti linijinio modelio  $a_i$  koeficientus.

```
pi_sum(p, k)  
  for i = 1 to length(p)  
    for j = 1 to k  
       $p_{ss}(1,1) = p_{ss}(1,1) + (p(\text{index}(i, j), 1))^2 - (c(i, 1))^2$   
       $p_{ss}(1,2) = p_{ss}(1,2) + (p(\text{index}(i, j), 2))^2 - (c(i, 2))^2$   
       $p_{ss}(1,3) = p_{ss}(1,3) + (p(\text{index}(i, j), 3))^2 - (c(i, 3))^2$   
  
   $p_{ss} = \frac{p_{ss}}{k}$   
  
   $a_i(i, 1:3) = \frac{p_{ss}}{p_{ss} + \varepsilon}$   
  
   $p_{ss} \leftarrow [0]$   
return  $a_i$ 
```

Galutinė filtravimo metodo funkcija skaičiuoja linijinį modelį  $b_i$  ir pakeistas reikšmes. Pirmuoju „for“ ciklu, kuris eina per visas taškų reikšmes, apskaičiuojami linijinio modelio koeficientai  $b_i$ , o po to perskaičiuojamos galutinės taškų reikšmės –  $p_i$  (5.12 lentelė). Linijinio modelio  $b_i$  reikšmė gaunama, padauginus atitinkamo indekso taško  $a_i$  reikšmę iš centroidės bei gautą reikšmę atėmus iš centroidės.

Galutinė  $p_i$  reikšmė gaunama, padauginus linijinio modelio  $a_i$  reikšmę iš centroidės reikšmės bei pridėjus linijinio modelio  $b_i$  reikšmę. Funkcijos kodas pateiktas (5p.3 lentelė).

**5.12 lentelė** Pseudo kodas randant  $b_i$  koeficientus bei grąžinanti pakeistas reikšmes.

```
Valdomas_t( $p, k$ )  
  for I = 1 to length( $p$ )  
    for J = 1 to 3  
       $b_i(I, J) = c(I, J) - (a_i(I, J) \cdot c(I, J))$   
       $p_i(I, J) = a_i(I, J) \cdot c(I, J) + b_i(I, J)$   
  return  $p$ 
```

### 5.3.2. Valdomo taško rezultatai

Valdomo taško filtravimo bandymų metu, naudojamos skirtingos kaimyninių parametrų reikšmės, pakeičiant taškus reikšmes į centroidžių reikšmes, kurios gaunamos panaudojant linijinio modelio  $a_i$  ir  $b_i$  reikšmes. Kiekvieno bandymo metu, pasirenkama kaimyninė reikšmė, su kuria yra atliekamas filtravimas. Su gautais filtravimo rezultatais atliekami pakartotiniai valdomo taško filtravimai, kurie nurodo iteracijos skaičių. Rezultatų lentelėje pateikiami 5 pirmųjų bandymų rezultatai bei dešimtosios iteracijos rezultatai. Eksperimento metu lyginami trikdžių koeficientai, naudojant skirtingus kaimyninių parametrų koeficientus bei atliekant skirtingas iteracijas. Bandymų metu buvo naudojamos skirtingos  $\varepsilon$  reikšmės, jos gali būti parenkamos iš skaičių imties [0.01;0.99], pagal skaičiuojamąjį tikslumą, reikšmė kisdavo tūkstantųjų tikslumu, todėl bandymuose buvo priimta reikšmė:  $\varepsilon = 0.5$ .

Parinkus kaimyninę reikšmę  $k=2$ , pirmosios iteracijos metu, trikdis  $s_1$  bei skirtumo koeficientas yra mažiausias, lyginant su kitomis iteracijomis. Trikdis  $s_2$  mažiausias 4 bei 10 iteracijų metu (5p.4 lentelė). Mažiausi trikdžių koeficientai su šiais parametrais yra gaunami:  $s_1 = 0.3037$ ,  $skirtumas = 47.2205$ ,  $s_2 = 2.8398$ . Didinant iteracijų skaičių, skaičiuojamasis skirtumo koeficientas nuo 47.2205 padidėja iki 61.9766, čia gaunamas didžiausias skirtumo šuolis, tarp visų iteracijų:  $|47.2205 - 61.9766| = 14.7561$ . Sekančiose iteracijose skirtumo koeficientas išlaiko mažesnę skirtumo augimą.  $s_1$  koeficientas nuo trečios iteracijos išlieka panašus, kur  $s_1 \approx 1$ . Filtruojamųjų modelių vaizdų palyginimas, skirtingose iteracijose, kai taškai pakeičiami pasirinkus 2 kaimyninius taškus, atvaizduojamas (5p.5 lentelė). Filtruoto modelio projekcijos, kai atstumai tarp analizuojamųjų trikdžių, yra mažiausi atvaizduotos (5p.6 lentelė).

Parinkus kaimyninę reikšmę  $k=3$ , pirmosios iteracijos metu, trikdžiai  $s_1$  ir  $s_2$  bei skirtumo koeficientas yra mažiausi, lyginant su kitomis iteracijomis (5p.7 lentelė). Mažiausi trikdžių koeficientai su šiais parametrais yra gaunami:  $s_1 = 0.5172$ ,  $s_2 = 2.9225$ ,  $skirtumas = 50.2660$ . Didinant iteracijų skaičių, skaičiuojamasis skirtumo koeficientas padidėja nuo 50.2660 iki 68.6013, kur didžiausias

skirtumo šuolis, tarp visų iteracijų:  $|50.2660 - 68.6013| = 18.3353$ . Sekančiose iteracijose skirtumo koeficientas išlaiko mažesnį skirtumo augimą. S1 koeficientas nuo trečios iteracijos išlieka panašus, kur  $s1 \approx 1$ . Filtruojamųjų modelių vaizdų palyginimas, skirtingose iteracijose, kai taškai pakeičiami pasirinkus 3 kaimyninius taškus, atvaizduojamas (5p.8 lentelė). Filtruoto modelio projekcijos, kai atstumai tarp analizuojamųjų trikdžių, yra mažiausi atvaizduotos (5p.9 lentelė).

Parinkus kaimyninę reikšmę  $k=4$ , pirmosios iteracijos metu, trikdžiai  $s1$  ir  $s2$  bei skirtumo koeficientas yra mažiausi, lyginant su kitomis iteracijomis (5p.10 lentelė). Mažiausi trikdžių koeficientai su šiais parametrais yra gaunami:  $s1 = 0.5401$ ,  $s2 = 0.6276$ ,  $skirtumas = 49.2058$ . Didinant iteracijų skaičių, skaičiuojamasis skirtumo koeficientas padidėja nuo 49.2058 iki 66.9965, kur didžiausias skirtumo šuolis, tarp visų iteracijų:  $|49.2058 - 66.9965| = 17.7907$ . Sekančiose iteracijose skirtumo koeficientas išlaiko mažesnį skirtumo augimą. S1 koeficientas nuo trečios iteracijos išlieka panašus, kur  $s1 \approx 1$ . Filtruojamųjų modelių vaizdų palyginimas, skirtingose iteracijose, kai taškai pakeičiami pasirinkus 4 kaimyninius taškus, atvaizduojamas (5p.11 lentelė). Filtruoto modelio projekcijos, kai atstumai tarp analizuojamųjų trikdžių, yra mažiausi atvaizduotos (5p.12 lentelė).

Parinkus kaimyninę reikšmę  $k=5$ , pirmosios iteracijos metu, trikdžiai  $s1$ ,  $s2$  bei skirtumo koeficientas yra mažiausi, lyginant su kitomis iteracijomis (5p.13 lentelė). Mažiausi trikdžių koeficientai su šiais parametrais yra gaunami:  $s1 = 0.6942$ ,  $s2 = 2.8704$ ,  $skirtumas = 54.2879$ . Didinant iteracijų skaičių, skaičiuojamasis skirtumo koeficientas padidėja nuo 49.2058 iki 66.9965, kur didžiausias skirtumo šuolis, tarp visų iteracijų:  $|54.2879 - 74.2312| = 19.9433$ . Sekančiose iteracijose skirtumo koeficientas išlaiko mažesnį skirtumo augimą. S1 koeficientas nuo antros iteracijos išlieka panašus, kur  $s1 \approx 1$ . Filtruojamųjų modelių vaizdų palyginimas, skirtingose iteracijose, kai taškai pakeičiami pasirinkus 5 kaimyninius taškus, atvaizduojamas (5p.14 lentelė). Filtruoto modelio projekcijos, kai atstumai tarp analizuojamųjų trikdžių, yra mažiausi atvaizduotos (5p.15 lentelė).

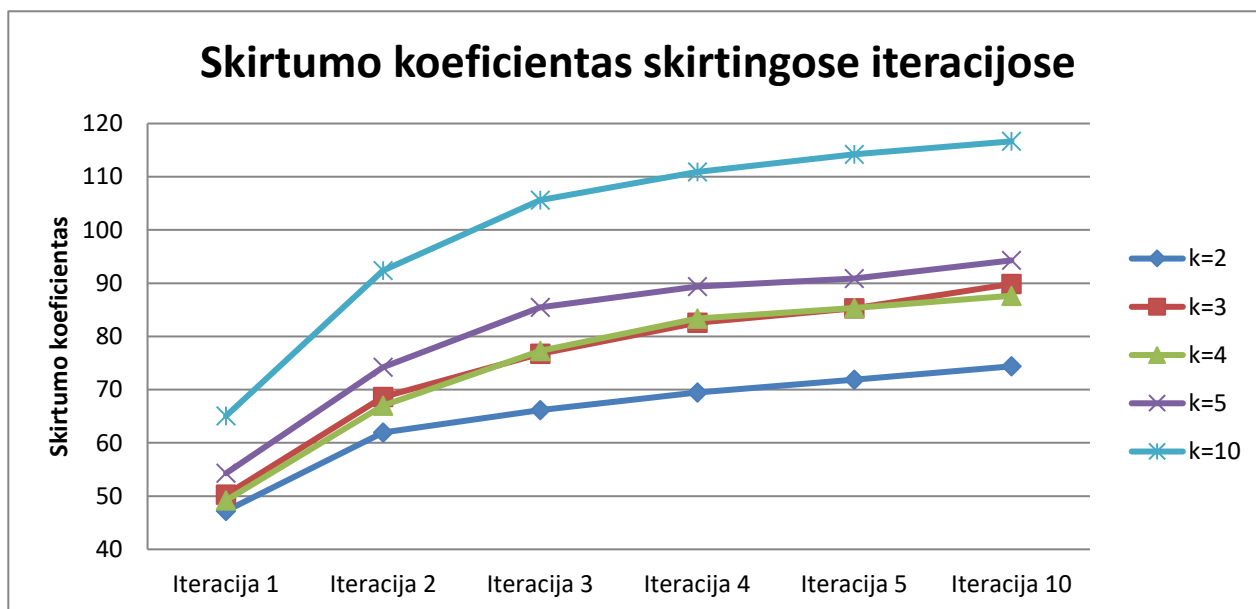
Parinkus kaimyninę reikšmę  $k=10$ , pirmosios iteracijos metu,  $s1$ ,  $s2$  trikdžiai bei skirtumo koeficientas yra mažiausi, lyginant su kitomis iteracijomis (5p.16 lentelė). Mažiausi trikdžių koeficientai su šiais parametrais yra gaunami:  $s1 = 0.7974$ ,  $s2 = 2.8896$ ,  $skirtumas = 65.0613$ . Didinant iteracijų skaičių, skaičiuojamasis skirtumo koeficientas padidėja nuo 65.0613 iki 92.3628, kur didžiausias skirtumo šuolis, tarp visų iteracijų:  $|65.0613 - 92.3628| = 27.3015$ . Sekančiose iteracijose skirtumo koeficientas išlaiko mažesnį skirtumo augimą. S1 koeficientas nuo antros iteracijos išlieka panašus, kur  $s1 \approx 1$ . Filtruojamųjų modelių vaizdų palyginimas, skirtingose iteracijose, kai taškai pakeičiami pasirinkus 5 kaimyninius taškus, atvaizduojamas (5p.17 lentelė). Filtruoto modelio projekcijos, kai atstumai tarp analizuojamųjų trikdžių, yra mažiausi atvaizduotos (5p.18 lentelė).

Nagrinėjant skirtumo koeficientus (5.13 lentelė), pirmos iteracijos metu, tarp skirtingų kaimyninių reikšmių, skirtumo koeficiento reikšmė kinta nuo 47.2205 iki 54.2879, kur skirtumas tarp mažiausios

ir didžiausios, pirmosios iteracijos, reikšmių yra lygus:  $sk_{25} = |47.2205 - 54.2879| = 7.0674$ . Čia skaičiuojami pirmieji 4 parametrai, kaimyninio parametro  $k=10$  skirtumo reikšmė neskaičiuojama, kadangi  $k=10$  skirtumo reikšmės yra ženkliai didesnės ir sudarytų bendrą duomenų nuokrypį. Pirmos iteracijos metu, tarp mažiausių skirtumo koeficientų  $k=2$  bei  $k=4$  gaunama reikšmė yra  $sk_{24} = |47.2205 - 49.2058| = 1.9853$ . Didinant iteracijų skaičių, grafike (5.9 pav.) pastebima, jog skirtingų kaimyninių reikšmių skirtumo koeficientai didėja, tarpusavyje išlaikant eilės tvarką. Skirtumas tarp mažiausios bei didžiausios reikšmės proporcingai auga, didinant iteracijų skaičių. Lyginant  $k=3$  bei  $k=4$  kaimynų skirtumus, matomas kreivių supanašėjimas. Padidinus iteracijų skaičių,  $k=3$  kreivė prieš tai turėjusi antrą mažiausią skirtumo reikšmę, skirtumo koeficiento dydžiu pralenkia  $k=4$  reikšmę. Todėl negalima teigti, jog mažesnis kaimynų kiekis, reiškia mažesnę bendrą skirtumo koeficientą.

**5.13 lentelė** Skirtumo koeficientai, skirtingų iteracijų metu.

	Skirtumo koeficientas skirtingose iteracijose tarp skirtingų kaimyninių reikšmių				
	k=2	k=3	k=4	k=5	k=10
Iteracija 1	47.2205	50.2660	49.2058	54.2879	65.0613
Iteracija 2	61.9766	68.6013	66.9965	74.2312	92.3628
Iteracija 3	66.1795	76.7256	77.3050	85.4715	105.5933
Iteracija 4	69.4517	82.5550	83.3626	89.3686	110.9078
Iteracija 5	71.8830	85.2846	85.3410	90.8947	114.1974
Iteracija 10	74.3855	89.8630	87.6477	94.2879	116.6532



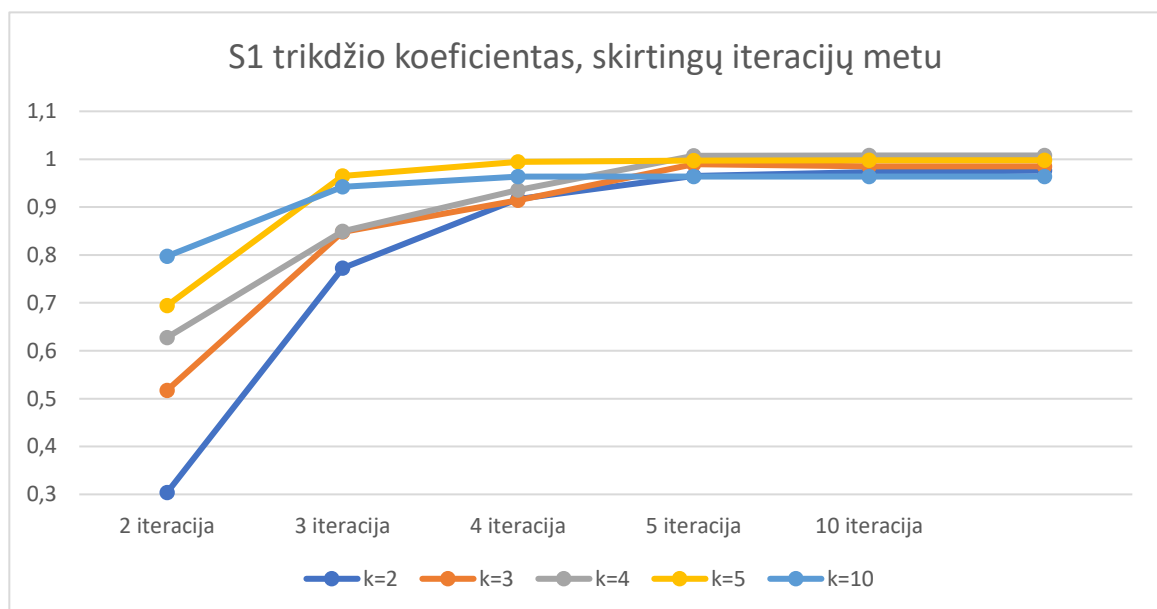
**5.9 pav.** Skirtumo palyginimas, skirtingų iteracijų metu

Lyginant trikdžio  $s_1$  pokytį, skirtingų iteracijų metu, naudojant skirtingas kaimynines reikšmes (5.14 lentelė), mažiausios  $s_1$  reikšmės gaunamos pirmosios iteracijos metu. Kur skirtumas tarp mažiausios bei didžiausios reikšmės  $sk_{110} = |0.3037 - 0.7974| = 0.4937$ . Didinant iteracijų skaičių,  $s_1$  reikšmių koeficientai kyla iki  $s_1 \approx 1$ , tuomet nepaisant skirtumo tarp kaimynų, trikdžio  $s_1$  reikšmė

supanašėja ir nuo trečios iteracijos skiriasi  $sk_{35} = |0.9142 - 0.9947| = 0.0805$ . Toliau didinant iteracijų skaičių, tarpusavio skirtumai dar labiau sumažėja. Grafiko pradžioje (5.10 pav.), pirmųjų dviejų iteracijų metu, mažiausia  $s_1$  reikšmė gaunama su kaimyniniu parametru  $k=2$ , nuo 4 iki 10 iteracijos, filtravimas su  $k=10$  kaimyniniu parametru turi mažiausią  $s_1$  koeficientą.

**5.14 lentelė** Trikdžių  $s_1$  koeficientai, skirtingų iteracijų metu.

Iteracija	k=2	k=3	k=4	k=5	k=10
1	0.3037	0.5172	0.6276	0.6942	0.7974
2	0.7727	0.8478	0.8493	0.9650	0.9425
3	0.9169	0.9142	0.9359	0.9947	0.9637
4	0.9652	0.9896	1.0071	0.9968	0.9638
5	0.9733	0.9850	1.0078	0.9978	0.9638
10	0.9761	0.9851	1.0076	0.9979	0.9638



**5.10 pav.** Trikdžio  $s_1$  koeficientai, skirtingų iteracijų metu.

**Išvados:** Mažiausias *skirtumas* bei  $s_1$  koeficientas, gaunami kai  $k=2$ , pirmosios iteracijos metu, tuomet  $skirtumas = 47.2205$ ,  $s_1 = 0.3037$ . Koeficientas  $s_2$  mažiausias, kai  $k=5$ , pirmosios iteracijos metu, tuomet  $s_2 = 2.8704$ . Koeficientas nesumažina trikdžio  $s_2$  reikšmės, todėl kai taškas yra atidėtas prie kitų modelio taškų, filtravimas netiksliai pakeičia trikdžio reikšmę. Lyginant su pradinėmis trikdžių reikšmėmis:  $skirtumas = 11.0650$ ,  $s_1 = 2.6500$ ,  $s_2 = 2.7432$ , valdomas taško filtravimas sumažina  $s_1$  koeficientą, tačiau  $s_2$ , optimaliausiu atveju padidėja nuo 2.6500 iki 2.8704. Skirtumo koeficientas, didinant iteracijas, smarkiai išauga, nes metodas skirtas sutankinti visus taškus, pagal taškų centroidines reikšmes, todėl pakeičiamos visų taškų padėty, atitinkamai padidinat skirtumo koeficientą. Atsižvelgus į rezultatus, metodas naudingas glotninant modelio paviršių, kai modelyje yra susidariusių smailių briaunų. Efektyviausias skirtumo koeficientas gaunamas pirmoje iteracijoje, didinant iteracijas, smarkiai padidėja skirtumas, kadangi kartojamas taškų artinimas prie kaimyninių reikšmių. Naudojant didesnes iteracijas, trikdžių koeficientai supanašėja, todėl naudoti

didesnį iteracijų skaičių yra neefektyvu, norint išlaikyti modelio kontūro savybes. Jeigu modelis turi smulkių detalizuotų paviršiaus dalių, filtravimą naudinga atlikti dalimis, išskiriant tokias smulkias modelio dalis arba, atlikti filtravimą tik su taškais, turinčiais didesnę atstumą tarpusavyje.

#### 5.4. Siūlomi metodų apjungimai

##### 5.4.1. Koordinatės pakeitimas stipriuotu tašku – Valdomas taškų filtravimas

Apjungiant filtravimo metodus (6p.1 lentelė), kai pirmiausiai atliekamas koordinatės pakeitimas stipriuotu tašku, o po jo atliekamas valdomas taškų filtravimas, mažiausias trikdžių  $s_1 = 0.7512$  bei  $s_2 = 2.9759$  rezultatas gaunamas, kai ašys filtruojamos  $x, y, z$ ;  $x, z, y$  bei  $z, x, y$  eilės tvarka, su slenksčio riba 4 bei kaimynine reikšme  $k=2$ , o valdomas taškų filtravimas atliekamas su kaimynine reikšme  $k=2$ . Mažiausios skirtumų reikšmės gautos  $skirtumas = 57.0489$ , kai ašys filtruojamos  $y, x, z$  bei  $y, z, x$  eilės tvarka, su slenksčio riba 4 bei kaimynine reikšme  $k=2$ , o valdomas taškų filtravimas atliekamas su kaimynine reikšme  $k=4$ . Naudojant tą pačią kaimyninę valdomų taškų reikšmę  $k$ , trikdžiai  $s_1$  bei  $s_2$  yra mažiausi su trejomis skirtingomis ašių kombinacijomis, tačiau skirtumo koeficientai tiesiogiai su tuo nekoreliuoja, kadangi mažiausi trikdžiai  $s_1$  bei  $s_2$  yra gaunami su kitokia ašių filtravimo tvarka, nei mažiausi skirtumo koeficientai. Vienintelio eksperimento metu, mažiausi trikdžių ir skirtumo koeficientai yra gaunami mažiausi su vienodais parametrais, kai pasirenkama  $z, y, x$  ašių tvarka, o valdomo taškų filtravimų kaimyninės reikšmės  $k=10$ .

##### 5.4.2. Valdomas taškų filtravimas – Koordinatės pakeitimas stipriuotu tašku

Apjungiant filtravimo metodus (6p.2 lentelė/6p.1 lentelė), kai pirmiausiai atliekamas valdomas taškų filtravimas, o po jo atliekamas koordinatės pakeitimas stipriuotu tašku, mažiausias trikdžio koeficientas  $s_1 = 0.5806$  gaunamas, kai ašys filtruojamos  $y, x, z$ ;  $y, z, x$  bei  $z, y, x$  eilės tvarka, su slenksčio riba 4 bei kaimynine reikšme  $k=2$ , o valdomas taškų filtravimas atliekamas su kaimynine reikšme  $k=4$ . Mažiausias trikdžio koeficientas  $s_2 = 2.7842$ , gaunamas, kai valdomas taškų skaičius  $k=2$ , o koordinatės pakeitimo stipriuotu tašku parametrai: slenksčio riba 4 ir kaimynų skaičius 2, ašių eilės tvarka nesvarbi. Mažiausias  $skirtumas = 62.5445$  gaunamas, kai ašys filtruojamos  $x, y, z$  eilės tvarka, su slenksčio riba 4 bei kaimynine reikšme  $k=2$ , o valdomas taškų filtravimas atliekamas su kaimynine reikšme  $k=4$ . Naudojant tą pačią kaimyninę valdomų taškų reikšmę  $k$ , trikdžiai  $s_1$  bei  $s_2$  yra mažiausi bent su trejomis skirtingomis ašių kombinacijomis.  $S_2$  koeficientas yra lygus, visų nagrinėjamų ašių atžvilgiu, todėl nuo ašių eiliškumo nepriklauso. Mažiausio skirtumo reikšmė su mažiausiomis trikdžių  $s_1$  bei  $s_2$  reikšmėmis koreliuoja, kai valdomų taškų kaimynų kiekis parenkamas 5 ir 10, tuomet  $y, x, z$  ašys turi mažiausius  $s_1, s_2$  trikdžių bei skirtumo koeficientus.

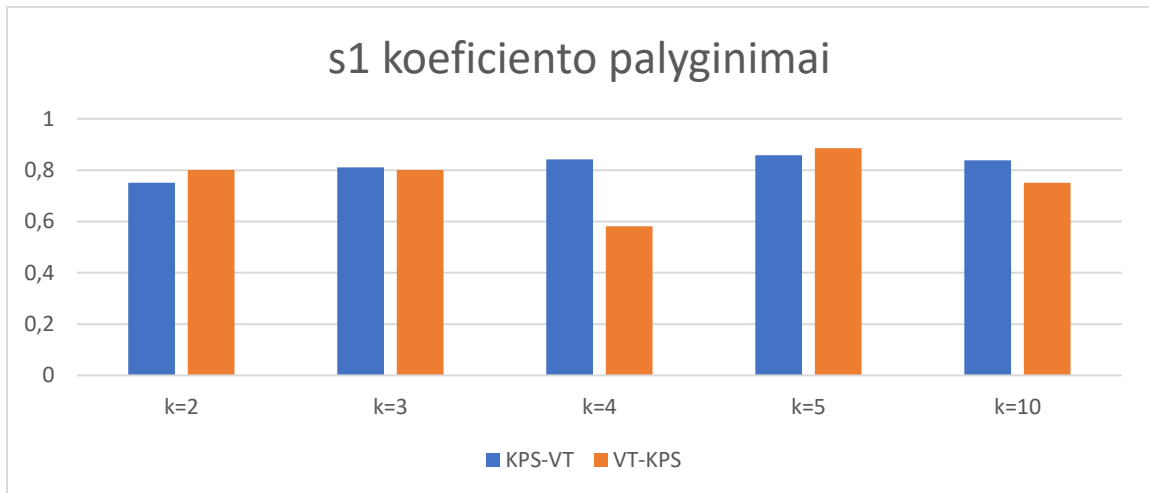
### 5.4.3. Rezultatų palyginimas tarp valdomo taškų filtravimo ir koordinatės pakeitimo stipriuojų tašku

Pradedant filtruoti koordinatės pakeitimo metodu, mažiausios koeficientų reikšmės gaunamos:  $s_1 = 0.7512$ ,  $s_2 = 2.9759$ ,  $skirtumas = 57.0489$ . O pradedant filtruoti nuo valdomo taškų metodo:  $s_1 = 0.5806$ ,  $s_2 = 2.7842$ ,  $skirtumas = 62.5445$ . Duomenys, naudojant skirtingą metodų filtravimo tvarką, pateikti (5.15 lentelė). Grafikuose, atspindinčiuose skirtumus tarp skirtingo metodų eiliškumo, naudojami optimaliausi rezultatai. Todėl grafikuose yra parinktos reikšmės, su kuriomis gaunamas mažiausias  $s_1$ ,  $s_2$  trikdžių bei skirtumo rezultatų rinkinys, kur pirmenybė teikiama  $s_1$  bei  $s_2$  trikdžių sumažinimui ir tada skirtumui. Kadangi, kai yra naudojamas valdomas taškų filtravimas, visos taškų reikšmės yra pakeičiamos, bendras skirtumo koeficientas didėja, todėl skirtumo koeficientas yra priimamas kaip mažiausiai svarbus koeficientas. Lyginant  $s_1$  koeficientą (5.11 pav.), pradedant filtruoti nuo valdomo taškų metodo,  $s_1$  reikšmė su šiuo eiliškumu yra mažesnė prie  $k=3$ ,  $k=4$  bei  $k=10$ . Skirtumas tarp  $s_1$  koeficiento, lyginant tarp skirtingo filtrų eiliškumo, išlieka panašus su visomis kaimyninėmis reikšmėmis, tačiau kai parenkama kaimyninė reikšmė  $k=4$ , tuomet skirtumas tarp trikdžio –  $s_1$  labiausiai išsiskiria:  $|0.8414 - 0.5806| = 0.2608$ . Lyginant  $s_2$  reikšmę (5.12 pav.), su visomis  $k$  – kaimyninėmis reikšmėmis, pradėti filtruoti, valdomu taškų metodu, yra efektyviau, tuomet gaunamas mažesnis  $s_2$  koeficientas. Lyginant skirtumų koeficientą (5.13 pav.), mažesni skirtumai su visomis kaimyninėmis reikšmėmis, gaunami, pradedant filtruoti koordinatės pakeitimo stipriuojų tašku metodu. *Skirtumas* gaunamas mažesnis pradedant filtruoti koordinatės pakeitimu stipriuojų tašku, kadangi pirmiausiai visos pašalinės plokštumos koordinatės perkeliamos į pagrindines plokštumas, o tuomet visi taškai yra sutankinami valdomu taškų pakeitimo filtru. Kai pirmiausiai atliekamas valdomas taškų filtravimas, jis sucentruoja taškus pagal kaimynus, todėl taškinis modelis yra tankesnis, dėl šios priežasties mažesnis taškų kiekis yra filtruojamas, kadangi didesnė taškų dalis yra identifikuojama, jog yra priklausanti pagrindinei plokštumai. Vizualiai optimaliausi filtravimai, su skirtingomis kaimyninėmis reikšmėmis, pavaizduoti (5.16 lentelė).

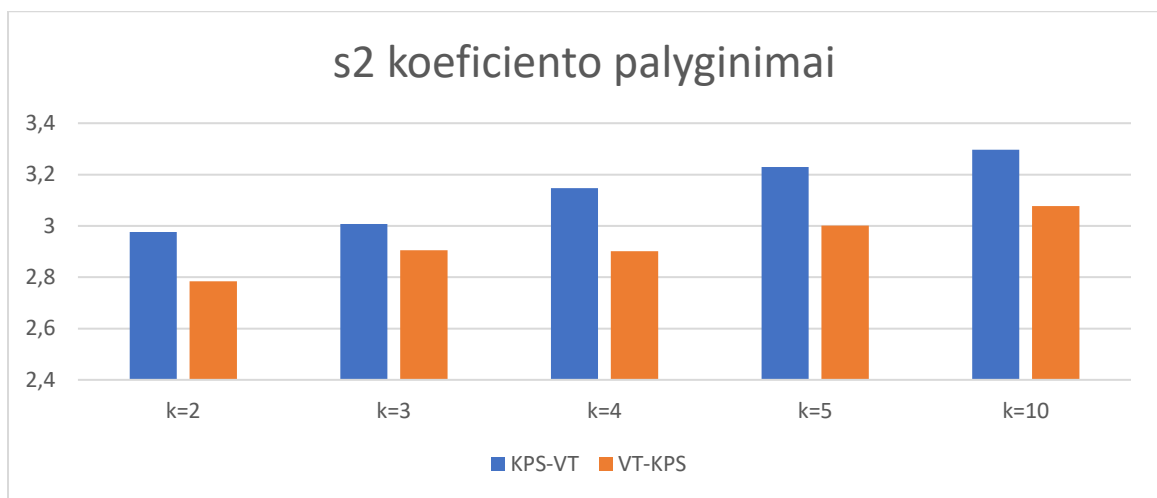
5.15 lentelė Skirtingo eiliškumo, metodų palyginimai

VT – parametrai		KPS – parametrai				Koordinatės pakeitimas stipriuoju tašku – Valdomas taškų filtravimas			Valdomas taškų filtravimas – Koordinatės pakeitimas stipriuoju tašku		
k	iteracija	ašys	s.riba	k	kk	s1	s2	skirtumas	s1	s2	skirtumas
2	1	xyz	4	2	4	0.7512	2.9759	58.3447	0.8414	2.7842	65.2893
		xzy				0.7512	2.9759	59.0395	0.8414	2.7842	64.1912
		yxz				0.8414	2.9657	57.6910	0.8003	2.7842	64.7609
		yzx				0.8414	2.9657	57.6910	0.8855	2.7842	66.4034
		zxy				0.7512	2.9759	58.7866	0.8414	2.7842	64.5108
		zyx				0.8414	2.9657	58.1329	0.8855	2.7842	66.0058
3	1	xyz	4	2	4	0.8111	3.0079	61.0642	0.8414	2.9050	68.3888
		xzy				0.8111	3.0079	61.7147	0.8414	2.9050	68.4143
		yxz				0.8720	3.0011	60.3567	0.8003	2.9050	69.1004
		yzx				0.8720	3.0011	60.3567	0.8003	2.9050	69.6965
		zxy				0.8111	3.0079	61.2332	0.8414	2.9050	68.2810
		zyx				0.8720	3.0011	60.5257	0.8003	2.9050	69.0810
4	1	xyz	4	2	4	0.8414	3.1465	57.6334	0.9336	2.9011	62.5445
		xzy				0.8414	3.1465	58.4978	0.9336	2.9011	64.8416
		yxz				0.8851	3.1416	57.0489	0.5806	2.9011	67.0458
		yzx				0.8851	3.1416	57.0489	0.5806	2.9011	66.0062
		zxy				0.8414	3.1465	58.1640	0.9336	2.9011	65.7344
		zyx				0.8851	3.1416	57.5794	0.5806	2.9011	67.6103
5	1	xyz	4	2	4	0.8578	3.2294	63.5738	0.9736	3.0005	69.8871
		xzy				0.8578	3.2294	64.0354	0.9736	3.0005	69.8476
		yxz				0.8657	3.2256	63.0228	0.8855	3.0005	69.1755
		yzx				0.8657	3.2256	63.0228	0.8855	3.0005	69.3518
		zxy				0.8578	3.2294	63.3630	0.9736	3.0005	69.2405
		zyx				0.8657	3.2256	62.8127	0.8855	3.0005	69.5975
10	1	xyz	4	2	4	0.8776	3.2982	74.9769	0.7512	3.0767	73.8366
		xzy				0.8776	3.2982	75.2925	0.7512	3.0767	73.7229
		yxz				0.8376	3.2963	74.6053	0.7512	3.0767	73.2383
		yzx				0.8376	3.2963	74.6053	0.7512	3.0767	73.4824
		zxy				0.8776	3.2982	74.3226	0.7512	3.0767	73.6278
		zyx				0.8376	3.2963	73.9506	0.7512	3.0767	74.1266

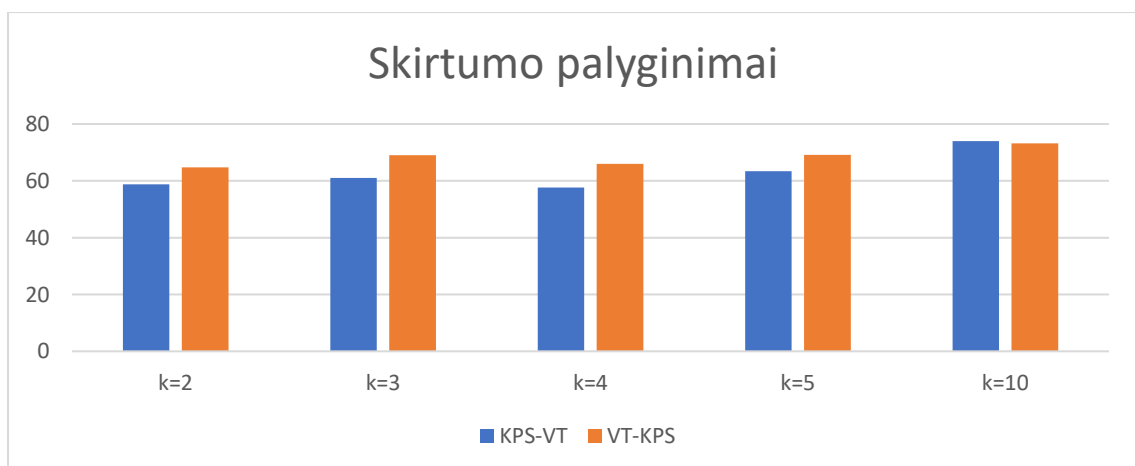




**5.11 pav.** S1 koeficientų grafinis palyginimas.

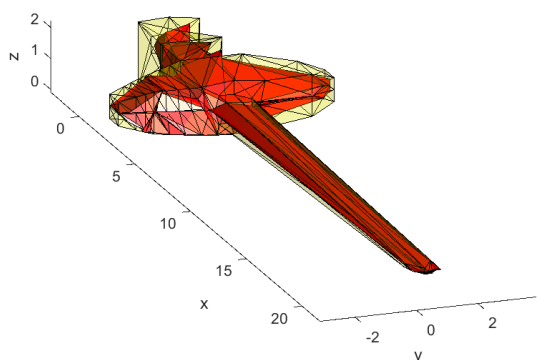
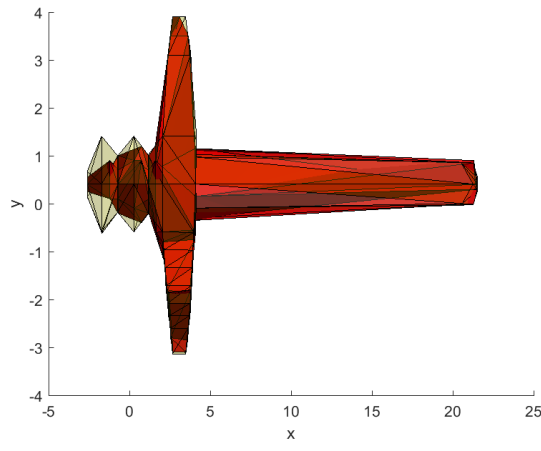
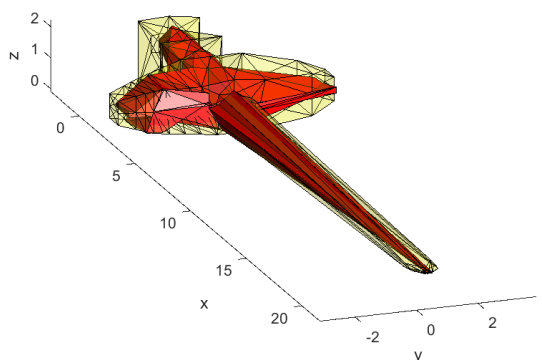
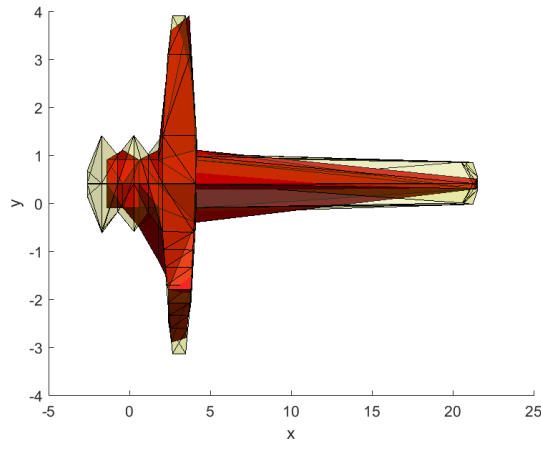
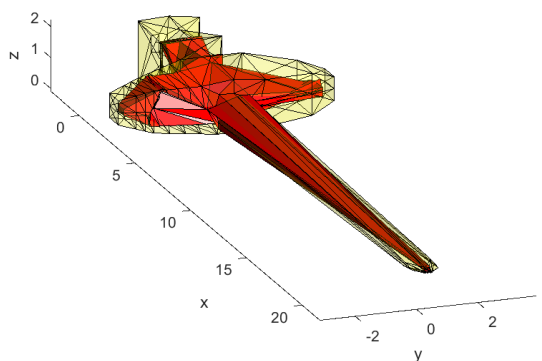
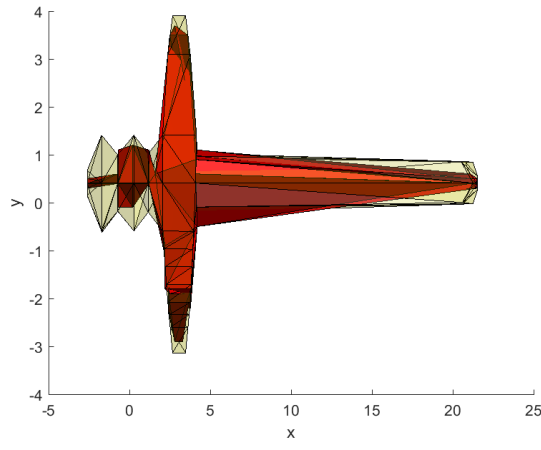


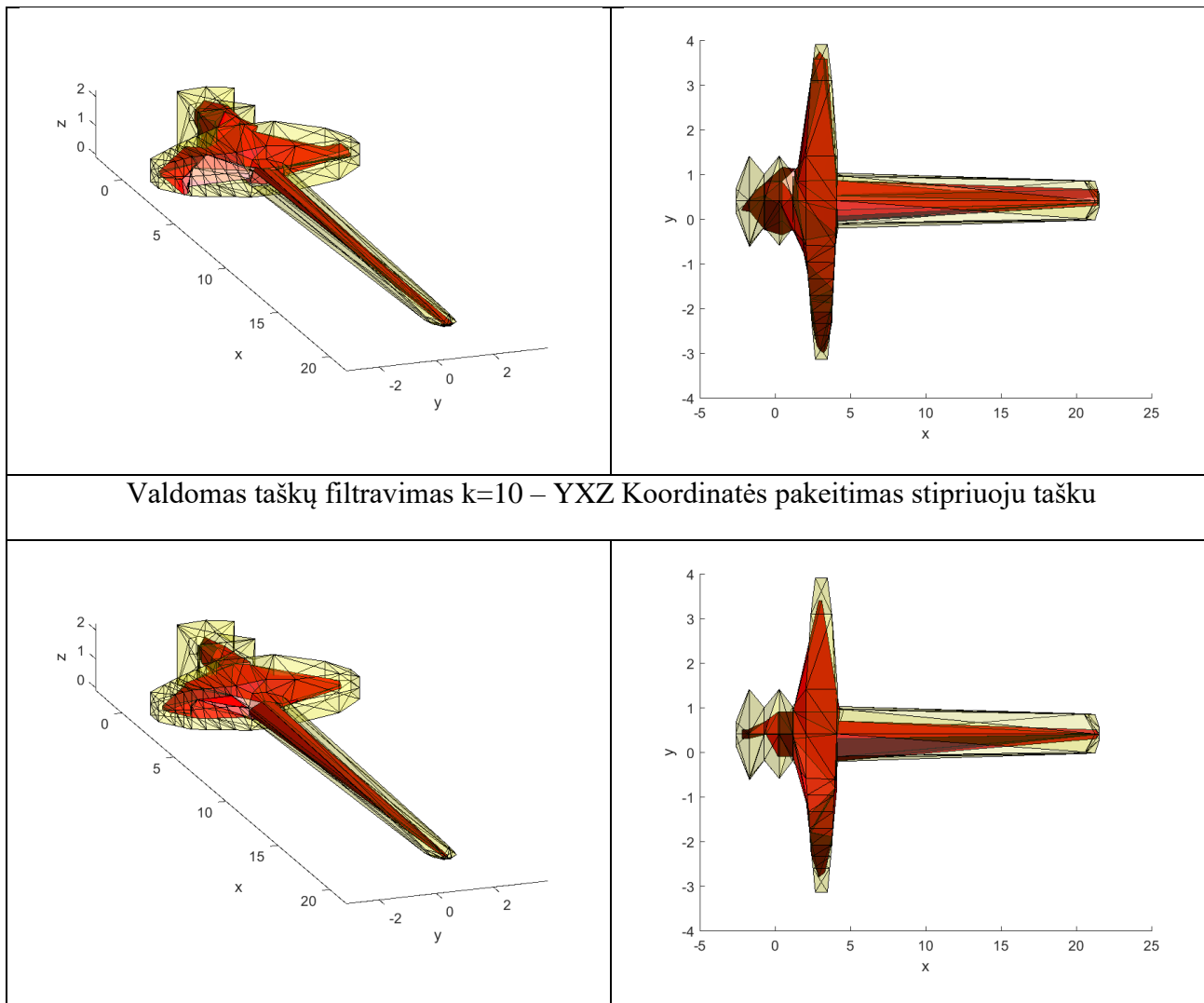
**5.12 pav.** S2 koeficientų grafinis palyginimas.



**5.13 pav.** Skirtumo koeficientų grafinis palyginimas.

5.16 lentelė Filtruoto modelio atvaizdavimas su skirtingomis k reikšmėmis.

XYZ Koordinatės pakeitimas stipriuoju tašku – Valdomas taškų filtravimas k=2	
	
Valdomas taškų filtravimas k=3 – ZYX Koordinatės pakeitimas stipriuoju tašku	
	
Valdomas taškų filtravimas k=4 – YZX Koordinatės pakeitimas stipriuoju tašku	
	
ZXY Koordinatės pakeitimas stipriuoju tašku – Valdomas taškų filtravimas k=5	



#### 5.4.4. Statistinis pašalinio taško pakeitimas – Valdomas taškų filtravimas

Apjungiant filtravimo metodus (6p.3 lentelė), kai pirmiausiai atliekamas statistinis pašalinio taško pakeitimas, o po jo atliekamas valdomas taškų filtravimas, mažiausias trikdžio koeficientas gaunamas  $s1 = 0.2989$ , su valdomo taško kaimynine reikšme  $k=2$ , su visomis lyginamų kaimynų reikšmėmis. Mažiausias *skirtumas* = 47.1835 gaunamas, kai valdomo taško koeficientas  $k=2$ , o lyginamų kaimynų reikšmė lygi 10. Koeficientas  $s2 = 2.8620$ , gaunamas mažiausias, kai valdomo taško koeficientas  $k=5$ , o lyginamų kaimynų vertė yra 10. Pirmųjų 3 bandymų metu,  $s1$  koeficientas išlieka vienodas su visomis lyginamų kaimynų vertėmis, o  $s2$  koeficientas turi bent 3 pasikartojančias vertes, naudojant skirtingą kaimynų kiekį, visų atliktų bandymų metu. Skirtumo koeficientas su pirmaisiais trimis valdomo taško parametrais, tarp skirtingų lyginamų kaimynų, apytiksliai skiriasi 1, palyginus minimalią skirtumo reikšmę su maksimalia reikšme. Todėl pradėdant filtruoti statistinio taškų pakeitimo metodu, su mažesnėmis kaimyninėmis reikšmėmis, galima laisviau manipuluoti skirtingomis parametru reikšmėmis, kadangi pradžioje tai nedaro žymios įtakos skirtumo koeficiento rezultatui.

#### 5.4.5. Valdomas taškų filtravimas – Statistinis pašalinio taško pakeitimas

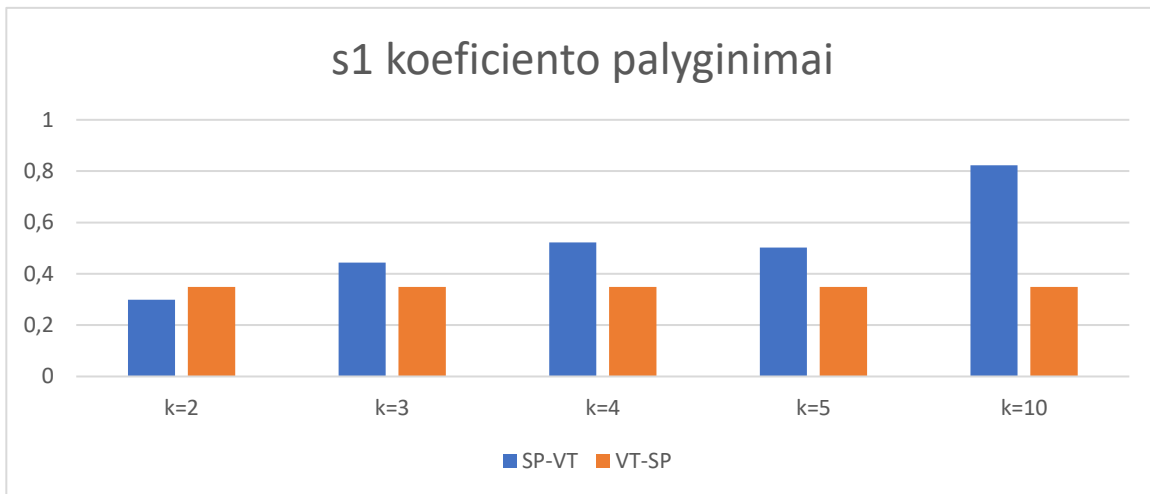
Apjungiant filtravimo metodus (6p.4 lentelė), kai pirmiausiai atliekamas valdomas taškų filtravimas, o po jo atliekamas statistinis pašalinio taško pakeitimas, mažiausias trikdžio koeficientas  $s_1 = 0.3486$  gaunamas, kai lyginamų kaimynų skaičius 2, su visomis valdomų taškų reikšmėmis. Mažiausias trikdžio koeficientas  $s_2 = 2.8377$  gaunamas, kai lyginamų kaimynų skaičius 5, su visomis valdomų taškų reikšmėmis. Mažiausias skirtumo koeficientas  $skirtumas = 48.3754$  yra gaunamas, kai lyginamų kaimynų skaičius 2, o valdomų taškų  $k$  reikšmė lygi 10. Trikdžių koeficientų  $s_1$  bei  $s_2$  reikšmės išlieka vienodos, su vienodomis lyginamų kaimynų reikšmėmis, kai yra keičiama valdomų taškų kaimyninė reikšmė. Skirtumo koeficientas taip pat išlieka panašus, tarp skirtingų valdomų kaimyninių taškų reikšmių, todėl valdomo taško kaimyninės reikšmės parametras neturi didelės įtakos bei gali būti laisvai keičiamas.

#### 5.4.6. Rezultatų palyginimas tarp valdomo taškų filtravimo ir statistinio pašalinio taško pakeitimo

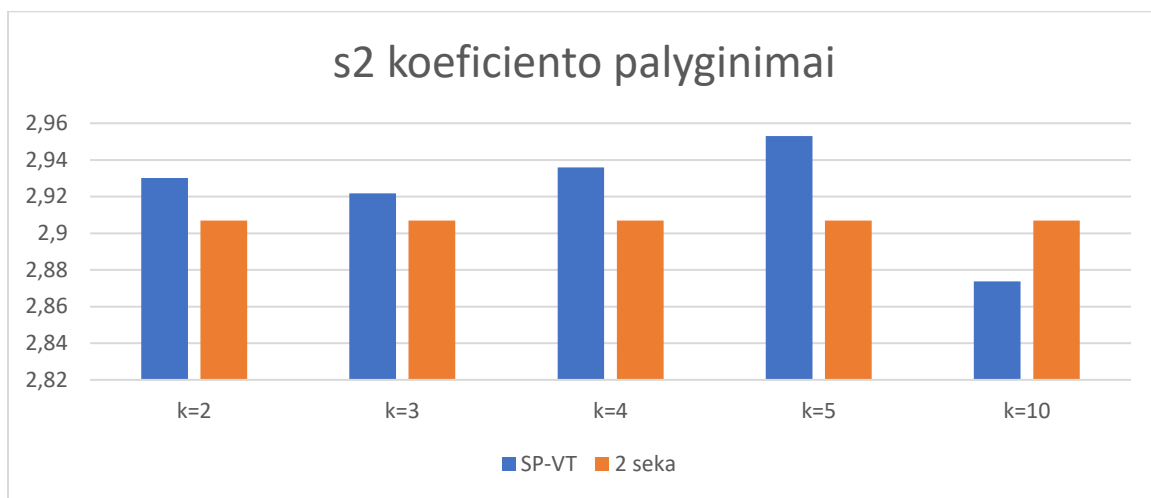
Kai pradama filtruoti statistinio pašalinio taško pakeitimo metodu, mažiausios koeficientų reikšmės gaunamos:  $s_1 = 0.2989$ ,  $s_2 = 2.8620$ ,  $skirtumas = 47.1835$ . O pradant filtruoti valdomo taškų metodu:  $s_1 = 0.3486$ ,  $s_2 = 2.8377$ ,  $skirtumas = 48.3754$ . Duomenys, naudojant skirtingą metodų filtravimo tvarką, pateikti (5.17 lentelė). Grafikuose, atspindinčiuose skirtumus tarp skirtingo metodų eiliškumo, naudojami optimaliausi rezultatai. Todėl grafikuose yra parinktos reikšmės, su kuriomis gaunamas mažiausias  $s_1$ ,  $s_2$  trikdžių bei skirtumo rezultatų rinkinys, kur pirmenybė teikiama  $s_1$  bei  $s_2$  trikdžių sumažinimui ir tada skirtumui. Kadangi, kai yra naudojamas valdomas taškų filtravimas, visos taškų reikšmės yra pakeičiamos, bendras skirtumo koeficientas didėja, todėl skirtumo koeficientas yra priimamas kaip mažiausiai svarbus koeficientas. Lyginant  $s_1$  koeficientą (5.14 pav.), pradant filtruoti valdomo taškų metodu, su pirmąja kaimynine reikšme  $k=2$ ,  $s_1$  koeficientas yra gaunamas didesnis, tačiau su visomis kitomis reikšmėmis,  $s_1$  yra mažesnis. Lyginant  $s_2$  reikšmę (5.15 pav.), su pirmomis 4 kaimyninėmis reikšmėmis,  $s_2$  koeficientas gaunamas mažesnis. Todėl abu trikdžių koeficientai išfiltruojami geriau, kai pradama filtruoti – valdomų taškų filtravimo metodu. Lyginant skirtumų koeficientą (5.16 pav.), duomenys, su pirmosiomis trimis kaimyninėmis reikšmėmis, yra panašūs, lyginant optimaliausius rezultatus. Pradedant filtruoti valdomų taškų metodu, skirtumo koeficientas išlieka tolygus. Pradedant filtruoti statistiniu metodu, skirtumo reikšmės tolydžiai kyla. Vizualiai optimaliausi filtravimai, skirtingų kaimyninių reikšmių metu, pavaizduoti (5.18 lentelė).

5.17 lentelė Skirtingo eiliškumo, metodų palyginimai

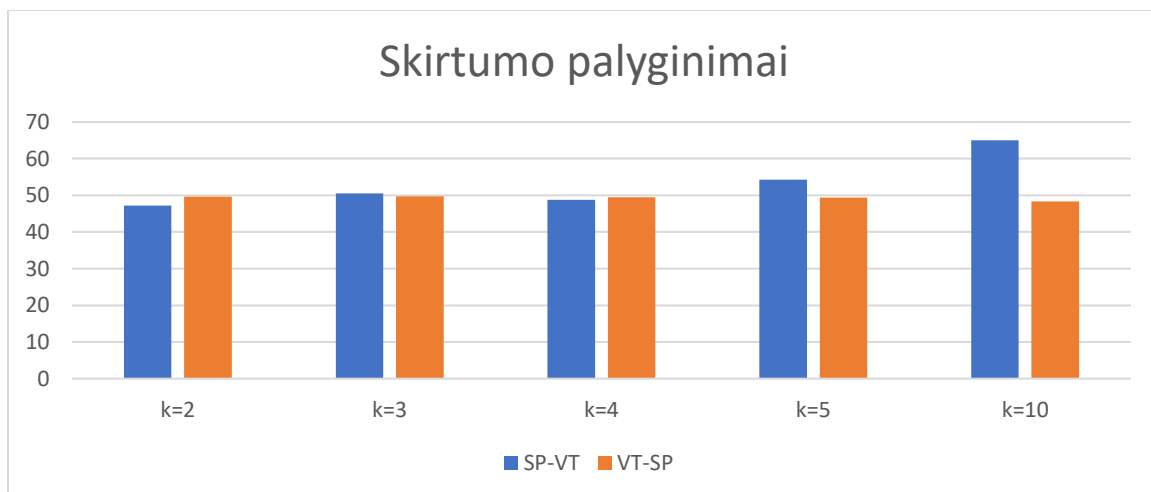
VT – parametrai		SP – parametrai	Statistinis pašalinio taško pakeitimas – Valdomas taškų filtravimas			Valdomas taškų filtravimas – Statistinis pašalinio taško pakeitimas		
k	iteracijos	k	s1	s2	skirtumas	s1	s2	skirtumas
2	1	2	0.2989	3.0527	47.8179	0.3486	2.9070	49.6417
		4	0.2989	3.0527	47.2106	0.5096	2.9197	51.6779
		6	0.2989	3.0527	47.3716	0.6307	2.9675	51.3698
		8	0.2989	3.0527	47.3560	0.6896	2.8377	55.5607
		10	0.2989	2.9302	47.1835	0.7883	2.9084	66.3496
4	1	2	0.4441	3.0541	51.3467	0.3486	2.9070	49.7685
		4	0.4441	2.9217	50.8046	0.5096	2.9197	51.1113
		6	0.4441	2.9217	51.2343	0.6307	2.9675	50.8382
		8	0.4441	2.9217	50.9948	0.6896	2.8377	55.3845
		10	0.4441	2.9217	50.5317	0.7883	2.9084	66.4540
6	1	2	0.5220	3.1770	49.7701	0.3486	2.9070	49.4763
		4	0.5220	2.9359	48.7545	0.5096	2.9197	51.1776
		6	0.5220	2.9359	48.8577	0.6307	2.9675	50.7419
		8	0.5220	2.9359	49.7352	0.6896	2.8377	55.3210
		10	0.5220	2.9359	48.7539	0.7883	2.9084	66.1748
8	1	2	0.5025	3.0712	54.7865	0.3486	2.9070	49.3722
		4	0.5025	2.9531	54.2758	0.5096	2.9197	51.0902
		6	0.6151	2.9531	54.3340	0.6307	2.9675	51.0320
		8	0.5025	2.9531	54.4295	0.6896	2.8377	55.2089
		10	0.6151	2.8620	54.1918	0.7883	2.9084	66.0730
10	1	2	0.6984	3.3574	65.7162	0.3486	2.9070	48.3754
		4	0.6991	3.1951	65.3225	0.5096	2.9197	50.1456
		6	0.7438	3.1951	65.3824	0.6307	2.9675	49.5493
		8	0.6991	3.1951	65.4537	0.6896	2.8377	54.2771
		10	0.8227	2.8737	65.0269	0.7883	2.9084	65.3113



**5.14 pav.** S1 koeficientų grafinis palyginimas.

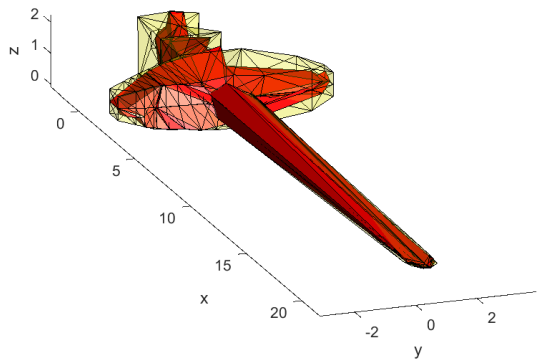
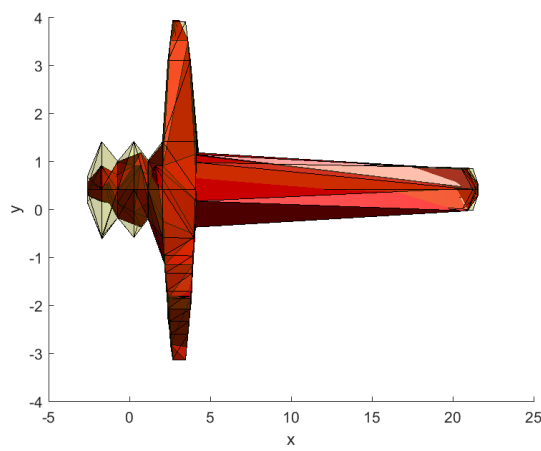
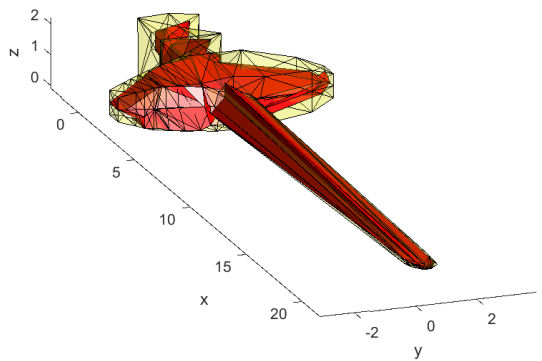
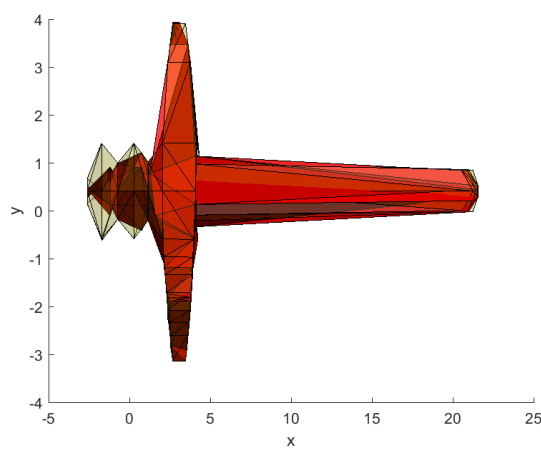
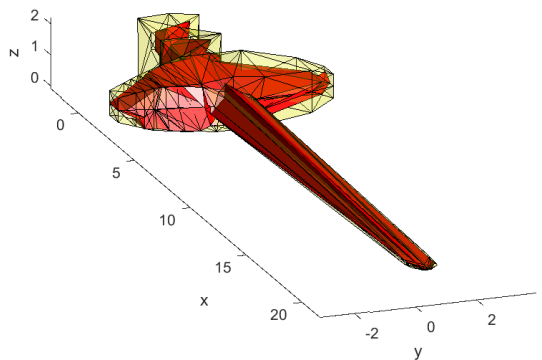
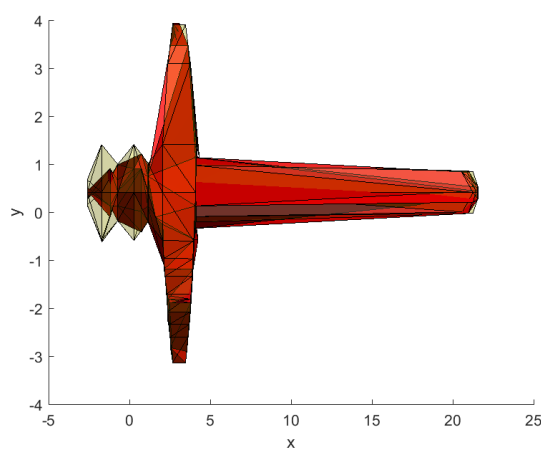


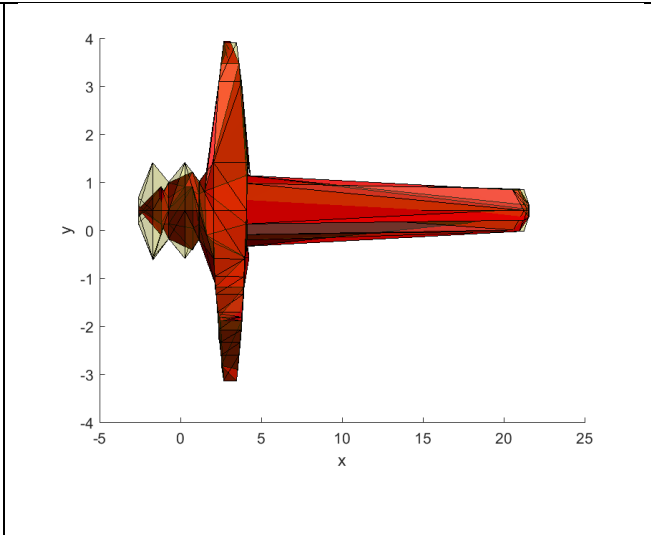
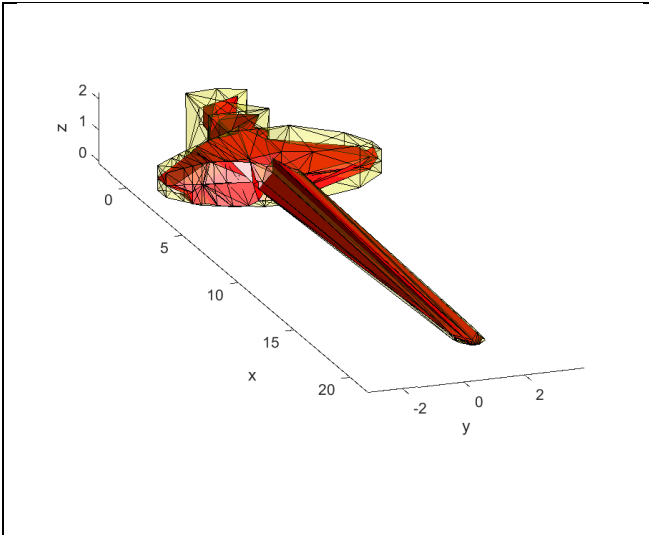
**5.15 pav.** S2 koeficientų grafinis palyginimas.



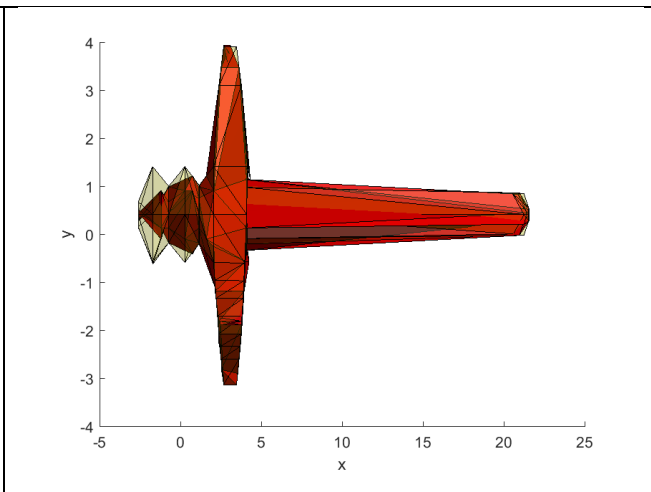
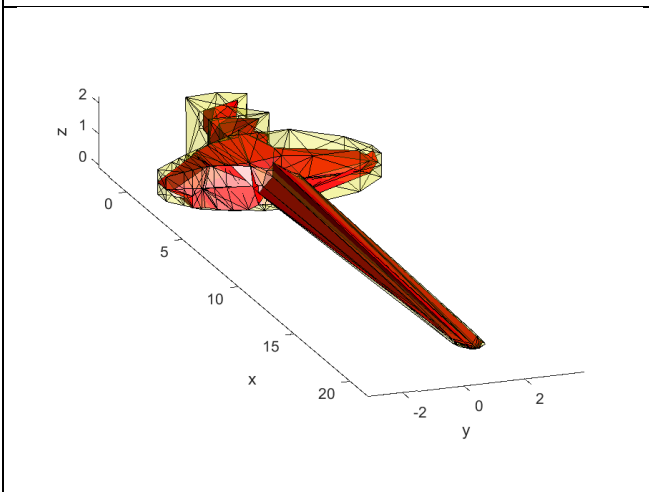
**5.16 pav.** Skirtumo koeficientų grafinis palyginimas.

5.18 lentelė Filtruoto modelio atvaizdavimas su skirtingomis k reikšmėmis.

Statistinis pašalinio taško pakeitimas $k=10$ – Valdomas taškų filtravimas $k=2$	
	
Valdomas taškų filtravimas $k=3$ – Statistinis pašalinio taško pakeitimas $k=10$	
	
Valdomas taškų filtravimas $k=6$ – Statistinis pašalinio taško pakeitimas $k=2$	
	
Valdomas taškų filtravimas $k=8$ – Statistinis pašalinio taško pakeitimas $k=2$	



Valdomas taškų filtravimas  $k=10$  – Statistinis pašalinio taško pakeitimas  $k=2$





## 6. IŠVADOS

1. Atlikus skenavimo įrenginių analizę, akivaizdaus kokybės skirtumo ar pritaikymo skirtumo, tarp struktūrizuotos šviesos ir lazerinio skenavimo įrenginio, nėra. Todėl projekto skenavimo įranga pasirenkama bekontaktė – struktūrizuotos šviesos, pateikianti duomenis taškinio debesio forma. Fotogrametrinis modelio skenavimas sudaro didesnę modelio tikslumą, tačiau procesas reikalauja daugiau kompiuterinių resursų.
2. Filtravimas atliekamas su taškinio debesies duomenimis, kadangi taškinį debesį galima apjungti į NURBS paviršius bei poligoninį tinklą, todėl taškinio debesų duomenų tipas yra universaliausias pritaikymui.
3. Modelio formatas parinktas .STL, kadangi eksperimentiškai tiriama tik modelio erdvinė geometrija, sudaryta iš taškinio debesies, todėl tekstūrų ar kitų elementų koduotė yra nereikalinga, taškų filtravimo pritaikyme. Šis formatas yra vienas universaliausias, nes yra palaikomas didžiosios dalies projektavimo/ modeliavimo programų, taip pat yra vienas praktiškiausių gamyboje, kadangi tiesiogiai naudojamas paruošiant 3D adityvinei bei CNC gamybai.
4. Koordinatės pakeitimo stipriuoju tašku, filtravimo metodas, yra skirtas aptikti pašalinius taškus, skirtingose ašyse, bei perkelti juos į stipriąją plokštumą, kur taškų pasiskirstymas nagrinėjamoje erdvėje yra tankiausias. Išanalizavus rastą, jog metodas efektyviausias su modeliu, kuriame yra tolygus taškų pasiskirstymas bei su modelių struktūromis, kurias įmanoma pozicionuoti lygiagrečiai ašimis, taip išsaugant bendrą modelio struktūrą filtruojant.
5. Išanalizavus, statistinį pašalinio taško pakeitimo, filtravimo metodą, nustatyta, jog metodu galima filtruoti ne tik pašalinius taškus, esančius  $\sigma_3$  kvartilio dalyje, tačiau galima mažinti kvartilį, didinant filtruojamųjų taškų imtį. Tačiau atitinkamai, turi būti didinamas modelio taškų kiekis, norint išlaikyti modelio struktūros sandarą.
6. Valdomas taškų, filtravimo metodas, skirtas sutankinti visus taškus erdvėje, priartinant juos prie artimiausių taškų centrinės reikšmės, taip suglotninant modelio paviršių. Padaryta išvada, jog norint išsaugoti smulkias modelio detales ar savybes, atliekant filtravimą – valdomų taškų metodu, naudinga įvesti ribinę reikšmę taškams, kurie bus paveikiami filtro arba nustatyti mažesnę kaimynų kiekį, filtruojamai modelio daliai.
7. Nagrinėjant atskirus metodus, su nagrinėjamu trimačiu modeliu, naudingiausias buvo, koordinatės pakeitimas stipriausiuoju tašku. Šiuo filtravimo optimaliausiu atveju trikdžių koeficientai gauti:  $s_1 = 0.1012$ ,  $s_2 = 2.9753$  bei  $skirtumas = 31.3551$ .  $S_1$  reikšmė sumažėjo labiausiai, lyginant su pradine reikšme  $s_1 = 2.6500$ . Daroma išvada, jog pakeitus koordinatę, arčiausio kaimyninio taško reikšmę, gaunama tiksliausia reikšmė, jeigu

skenuojant taškas erdvėje buvo nutolęs nuo jo tikrosios vietos. Tai atvaizduoja filtruotos  $s_1$  reikšmės pokytis, jis ženkliai sumažėjo. Jeigu taškas nukrypo prie kitų modelio taškų, jis nebus filtruotas optimaliausiai, tai matoma iš filtruotos reikšmės  $s_2$  vertės, ji padidėjo.

8. Nagrinėjant apjungtus metodus, naudingiausias metodų junginys buvo: statistinis pašalinio taško pakeitimas – valdomas taškų filtravimas. Tuomet optimaliausiu atveju  $s_1 = 0.2989$ ,  $s_2 = 2.9302$  bei  $skirtumas = 47.1835$ .  $S_1$  koeficiento reikšmė sumažėjo labiausiai lyginant su pradine  $s_1 = 2.6500$ . Daroma išvada, jog apjungus metodus, pašalinės koordinatės filtruojamos optimaliausiai, kadangi filtruoti taškai išlaiko tankų pasiskirstymą erdvėje. Tuomet sutankinant reikšmes, modelio kraštinės tarpusavyje suglotninamos, o filtruotos taškų pozicijos išlaiko modelio struktūrą.
9. Lyginant tiesiogiai dvi skirtingų filtravimų grupes, skirtingu eiliškumu, padaryta išvada, jog mažesni rezultatai gaunami su statistiniu pašalinio taško pakeitimu. Statistinio pašalinio taško pakeitimą su valdomu taškų filtravimu, optimaliausių rezultatų reikšmės gautos režiuose:  $s_1 = [0.2989; 0.8227]$ ,  $s_2 = [2.8737; 2.9531]$  bei  $skirtumas = [48.3754; 65.0269]$ . Koordinatės pakeitimo stipriuotu tašku su valdomu taškų filtravimu, optimaliausių rezultatų reikšmės gautos režiuose:  $s_1 = [0.5806; 0.8578]$ ,  $s_2 = [2.7842; 3.2963]$  bei  $skirtumas = [57.6334; 73.9506]$ .
10. Turint tankų, gerai detalizuotą taškinį debesį, yra siūloma apjungti metodus, pirmiausiai išfiltruojant pašalinius taškus, o tuomet sutankinant taškus. Atliekant filtravimą tokia tvarka, pašaliniai taškai yra tiksliau išfiltruojami, kadangi reikšmės yra lengviau aptinkamos, o valdomų taškų filtravimo metodu, suglotninamas modelio paviršius, sumažinant aštrius modelio kampus.

## Literatūros sąrašas

1. „Quality basics, how does 3D scanning work?” I. Wright 2016m.  
<https://www.engineering.com/AdvancedManufacturing/ArticleID/12390/Quality-Basics-How-Does-3D-Laser-Scanning-Work.aspx> [peržiūrėta 2018-01-10]
2. „Structured light scanning“ K. McDonald 2015m.  
<http://www.instructables.com/id/Structured-Light-3D-Scanning/> [peržiūrėta 2018-01-10]
3. „A brief introduction to photogrammetry and remote sensing“  
<https://www.gislounge.com/a-brief-introduction-to-photogrammetry-and-remote-sensing/>  
[peržiūrėta 2018-01-10]
4. „From point clouds to triangular meshes“ M. Ioannides, Cyprus, 2013m.
5. „Point based reconstruction of thin objects“ B. Ummenhofer, T. Brox, Germany, 2013m.
6. „Evaluation of algorithms and tools for 3D modeling of laser scanning“ J. A. Setkowicz  
Norwegia, 2014m.
7. „Simple Method for Constructing NURBS Surfaces from Unorganized Points“, N. Leal, E.  
Leal, J. W. Branch, Columbia.
8. „Polygonal Surface Remeshing with Delaunay Refinement“, T. K. Dey, G. Li, T. Ray, USA,  
2010m.
9. „STL File Format (3D Printing) – Simply Explained.“ Dibya Chakravorty  
2018 <https://all3dp.com/what-is-stl-file-format-extension-3d-printing> [peržiūrėta 2018-12-  
27]
10. „What is an OBJ File“ [https://all3dp.com/1/obj-file-format-3d-printing-cad/#what-is-an-  
obj-file](https://all3dp.com/1/obj-file-format-3d-printing-cad/#what-is-an-obj-file) [peržiūrėta 2018-12-27]
11. „Object Files (.obj)“ <http://paulbourke.net/dataformats/obj/> [peržiūrėta 2018-12-27]
12. „Filtering method for 3D laser scanning point cloud“ D. Liu, L. Wang, Y. Hao, J. Zhang,  
2015m.
13. „Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction“ K. Wolff,  
C. Kim, H. Zimmer, C. Schroers, M. Botsch, O. Sorkine-Hornung, A. Sorkine-Hornung,  
2016m. <https://www.sciencedirect.com/science/article/pii/S2288430017302117>  
[peržiūrėta 2018-12-10]
14. „Fundamentals of Machine Learning for Predictive Data Analytics“ J.D.Kelleher, B. M.  
Namee, A. D'Arcy 2015m.
15. „Guided 3D point cloud filtering“ X. Han, J. S. Jin, M. Wang, W. Jiang, China, 2017m.  
<https://www.researchgate.net/publication/320599421> [peržiūrėta 2018-06-03]

## Priedai

### 1. Priedas. STL formato įkėlimas.

#### 1p.1 lentelė STL formato įkėlimo į „MATLAB“ aplinką funkcija

```
function [F,V,N] = stlbinary(M)

    F = [];
    V = [];
    N = [];

    numFaces = typecast(M(81:84),'uint32');
    if numFaces == 0
        warning('MATLAB:stlread:nodata','No data in STL file.');
```

```
        return
    end

    T = M(85:end);
    F = NaN(numFaces,3);
    V = NaN(3*numFaces,3);
    N = NaN(numFaces,3);

    numRead = 0;
    while numRead < numFaces
        i1 = 50 * numRead + 1;
        i2 = i1 + 50 - 1;
        facet = T(i1:i2)';

        n = typecast(facet(1:12),'single');
        v1 = typecast(facet(13:24),'single');
        v2 = typecast(facet(25:36),'single');
        v3 = typecast(facet(37:48),'single');

        n = double(n);
        v = double([v1; v2; v3]);
        fInd = numRead + 1;
        vInd1 = 3 * (fInd - 1) + 1;
        vInd2 = vInd1 + 3 - 1;

        V(vInd1:vInd2,:) = v;
        F(fInd,:) = vInd1:vInd2;
        N(fInd,:) = n;

        numRead = numRead + 1;
    end
end
```

### 2. Priedas. Pasikartojančių taškų indeksavimas.

#### 2p.1 lentelė Pasikartojančių taškų indeksavimo funkcija.

```
i=length(Vert(:,1));
stop=round(i/(10*3));
p_Vert=zeros(1,stop);
defaultinis=p_Vert;
for a= 1:i
    k=length(p_Vert(:,1));
    c=1;
    for b=1:k
        if (Vert(a,:)~= p_Vert(b,1:3)) & (c == k);
            p_Vert(b,1:3)=Vert(a,:);
            p_Vert(b,4)=a;
```

```

        p_Vert(b+1,:) = default;
    elseif Vert(a,:) == p_Vert(b,1:3);
        for d=5:stop
            if p_Vert(b,d) == 0
                p_Vert(b,d) = a;
                c=0;
                break
            end
        end
    end

    end
c=c+1;
if c==0
    break
end
end
end

```

### 3. Priedas. Koordinatės pakeitimas stipriuoju tašku.

#### 3p.1 lentelė Koordinatinių transformacijų apie ašį.

```

C=[X; Y; Z; ones(1,numel(X))];
alpha = 2*pi-0.748;
T=[cos(alpha) -sin(alpha) 0 0;
   sin(alpha)  cos(alpha) 0 0;
   0            0         1 0;
   0            0         0 1];
C=T*C;
beta = 2*pi+0.05;
T1=[cos(beta) 0 sin(beta) 0;
    0          1  0       0;
   -sin(beta) 0 cos(beta) 0;
    0          0  0       1];
gamma = 2*pi-0.215;
T2=[1  0  0  0;
    0 cos(gamma) -sin(gamma) 0;
    0 sin(gamma)  cos(gamma) 0;
    0  0  0  1];
C=T*T1*T2*C;

```

#### 3p.2 lentelė Unikalių plokštumų identifikavimas, nagrinėjama ašimi.

```

i=length(p_Vert(:,1));
p_Vert=round(p_Vert,1);
asis=1;
list = min(p_Vert(:,asis));
for a = 1:i
    b=length(list(:,1));
    aa=0;
    for c = 1:b
        if p_Vert(a,asis) ~= list(c,1)
            aa=aa+1;
        end
    end
    if aa == b
        list(b+1,1) = p_Vert(a,asis);
    end
end
end

```

### 3p.3 lentelė Pašalinių taškų radimas plokštumose.

```
i=length(p_Vert(:,1));
ii=length(list(:,1));
slenkstis=round(i/100,0);
in=0;
for a = 1:ii
    o=0;
    for b = 1:i
        if list(d,1)== p_Vert(b,asis)
            in=in+1;
            listindeksas(in,:)=[list(a,1),b];
            o=o+1;
        end
        if o>slenkstis
            listindeksas = listindeksas(1:end-(slenkstis+1),:);
            in=in-(slenkstis+1);
            break
        end
    end
end
o=0;
end
```

### 3p.4 lentelė Kaimyninių reikšmių radimas, naudojant euklidinį atstumą.

```
kk=8;
k=round(kk/2,0);
d=inf(ii,kk);
indeksas=zeros(ii,kk);
for a=1:ii
    for b=1:ii
        dt=(p_Vert(a,1)-p_Vert(b,1))^2+(p_Vert(a,2)-
p_Vert(b,2))^2+(p_Vert(a,3)-p_Vert(b,3))^2;
        dt=sqrt(dt);
    end
    for c=1:kk
        if dt<d(a,c)
            testd=d(a,:);
            testi=indeksas(a,:);
            for cc=c:kk-1
                d(a,cc+1)=testd(1,cc);
                indeksas(a,cc+1)=testi(1,cc);
            end
            d(a,c)=dt;
            indeksas(a,c)=b;
            break
        end
    end
end
end
end
```

### 3p.5 lentelė Stipriausių taškų radimas ir koordinatės pakeitimas ašimi.

```
k=round(kk/2,0);
for a = 1:i
    for b = 1:k
        stip_task_vid(a,b)=(sum(d(indeksas(listindeksas(a,2),b),:)))/kk;
    end
    stip_taskas(a,1)=stip_task_vid(a,1);
    for c = 2:k
        if stip_taskas(a,1)>stip_task_vid(a,c);
            stip_taskas(a,1)=stip_task_vid(a,c);
            s_indeksas(a,1)=c;
        end
    end
end
```

```

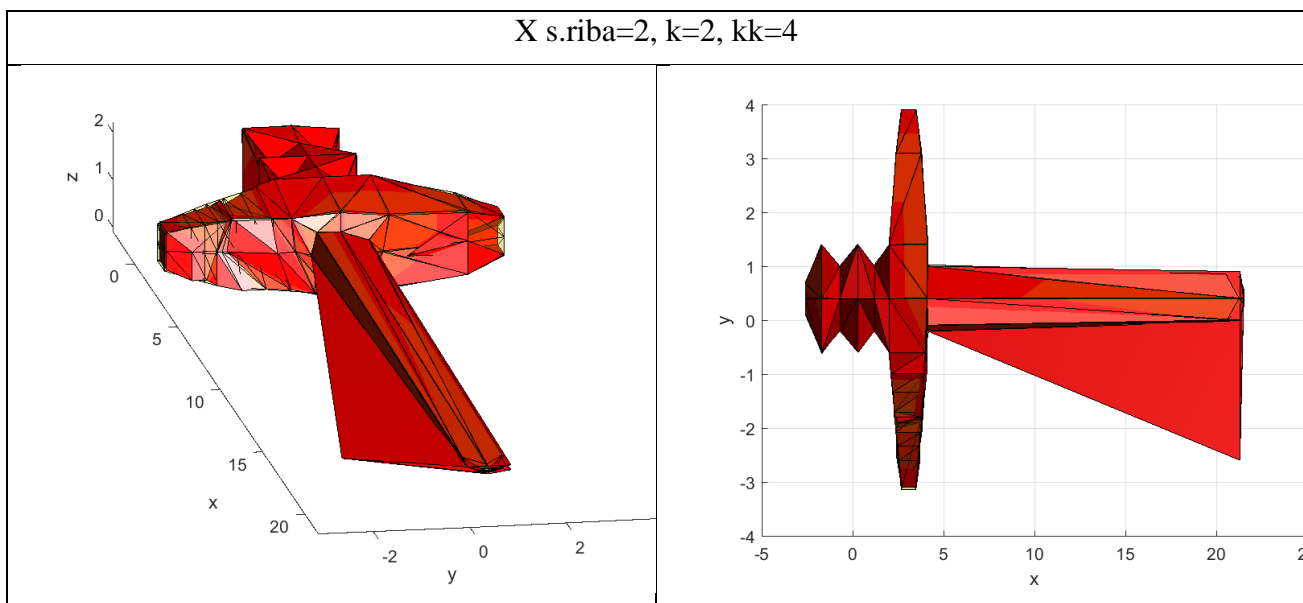
end
end
end
for a=1:i
p_Vert(listindeksas(a,2),asis)=p_Vert(indeksas(listindeksas(a,2),s_indeksas(a,1)),asis);
end

```

**3p.6 lentelė** Filtravimo rezultatai, pasirinkus x ašį.

ašis	slenksčio riba	k	kk	s1	s2	skirtumas
x	2	2	4	2.6831	2.1788	18.3606
		3	6	2.7398	2.2053	18.5391
		4	8	2.7398	2.2053	18.6610
		5	10	2.7398	2.2053	18.8124
x	3	2	4	2.6831	2.1788	18.6546
		3	6	2.7398	2.2053	19.1012
		4	8	2.7398	2.2053	19.1985
		5	10	2.7398	2.2053	19.2171
x	4	2	4	2.6831	2.2053	20.7619
		3	6	2.7398	2.2053	21.5983
		4	8	2.7398	2.2053	21.4852
		5	10	2.7398	2.2053	20.9138
x	5	2	4	2.6831	2.2053	22.7318
		3	6	2.7398	2.2053	23.8630
		4	8	2.7398	2.2053	23.7498
		5	10	2.7398	2.2053	23.1784

**3p.7 lentelė** Optimaliausio x ašimi filtravimo atvaizdavimas.

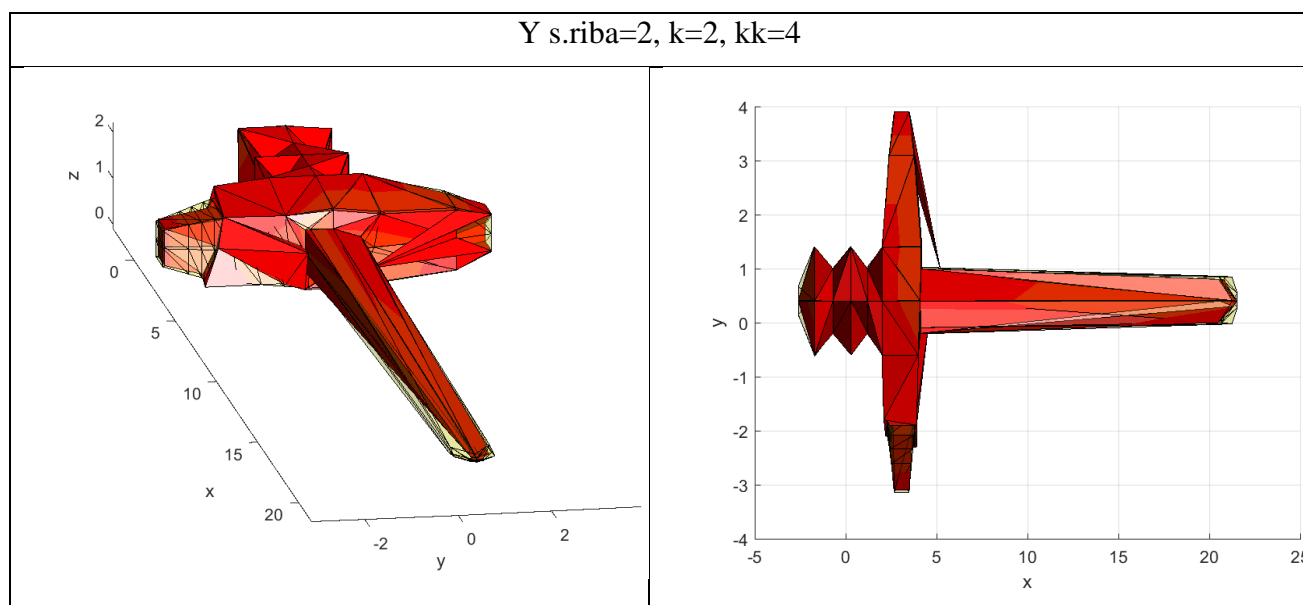


**3p.8 lentelė** Filtravimo rezultatai, pasirinkus y ašį.

ašis	slenksčio riba	k	kk	s1	s2	skirtumas
y	2	2	4	2.6019	3.3865	22.2035
		3	6	2.6019	3.3865	22.8017

		4	8	2.6019	3.3865	23.0945
		5	10	2.6019	3.3865	23.0930
y	3	2	4	2.6019	3.3865	24.9738
		3	6	2.6019	3.3865	25.3041
		4	8	2.6019	3.3865	24.7988
		5	10	2.6019	3.3865	24.3146
y	4	2	4	0.4138	3.3865	26.7559
		3	6	0.5290	3.3865	29.1865
		4	8	0.5290	3.3865	29.1535
		5	10	0.5290	3.3865	27.2466
y	5	2	4	0.4138	3.3865	26.8081
		3	6	0.5290	3.3865	29.3315
		4	8	0.5290	3.3865	30.1576
		5	10	0.5290	3.3865	27.9710

3p.9 lentelė Optimaliausio y ašimi filtravimo atvaizdavimas.



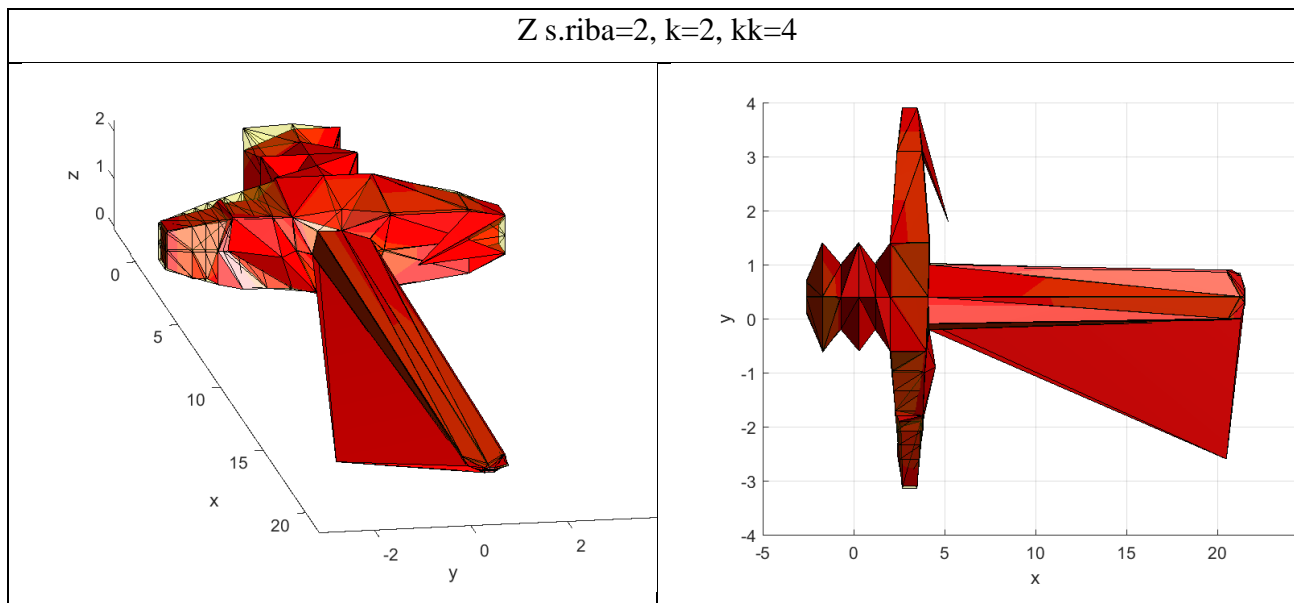
3p.10 lentelė Filtravimo rezultatai, pasirinkus z ašį.

ašis	slenksčio riba	k	kk	s1	s2	skirtumas
z	2	2	4	2.5740	2.7266	18.6789
		3	6	2.5740	2.7266	18.8816
		4	8	2.5740	2.7266	20.9742
		5	10	2.5724	2.7266	20.9726
z	3	2	4	2.5740	2.7266	18.9893
		3	6	2.5740	2.7266	20.4085
		4	8	2.5740	2.7266	22.2011
		5	10	2.5740	2.7266	22.8430
z	4	2	4	2.5740	2.7266	21.6500
		3	6	2.5740	2.7266	23.8808
		4	8	2.5740	2.7266	27.5539
		5	10	2.5740	2.7266	29.5946
z	5	2	4	2.5740	2.7266	22.9349
		3	6	2.5740	2.7266	25.6765



		4	8	2.5740	2.7266	29.1178
		5	10	2.5740	2.7266	31.1585

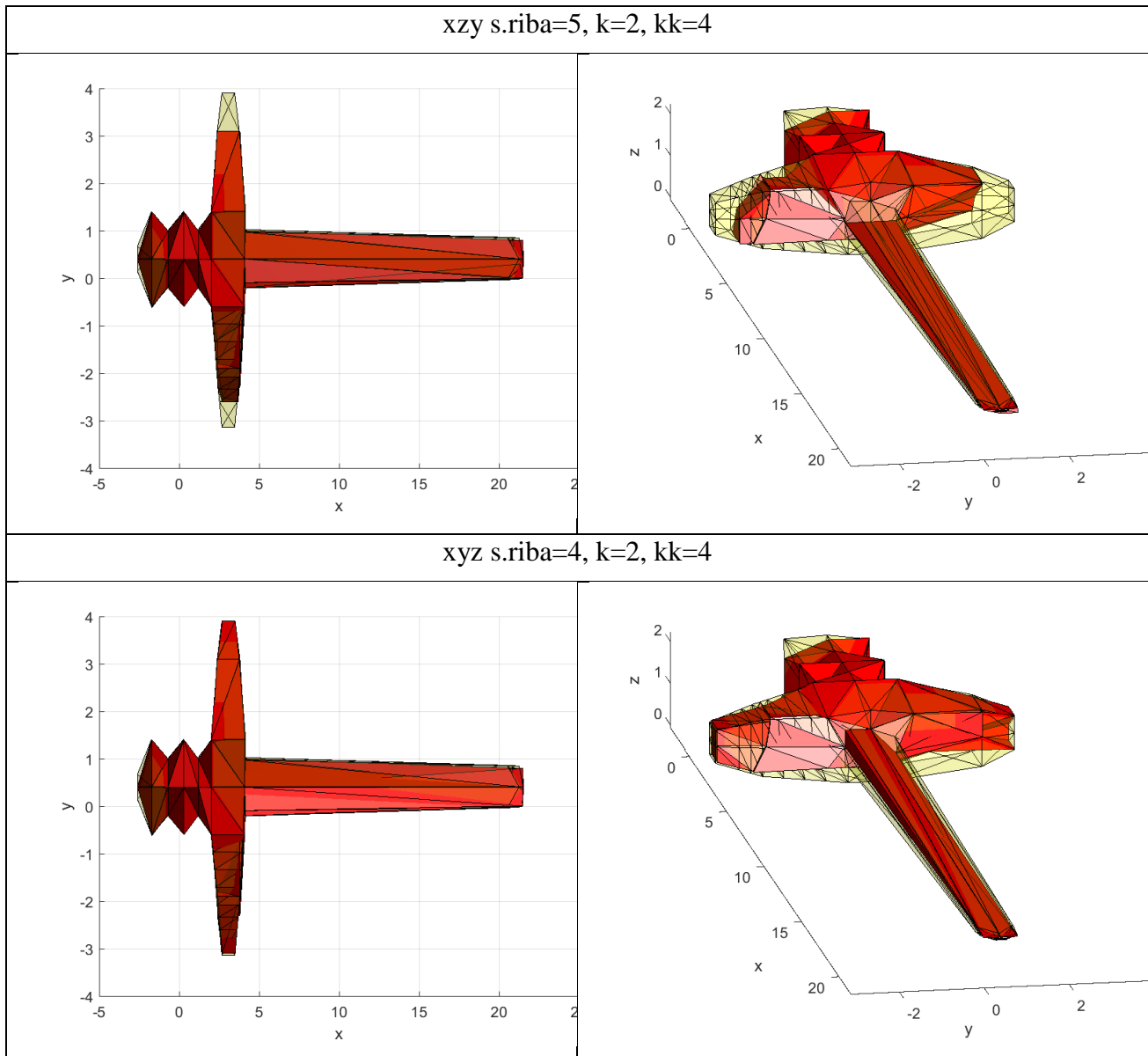
3p.11 lentelė Optimaliausio z ašimi filtravimo atvaizdavimas.



3p.12 lentelė Filtravimo rezultatai, filtruojant trimis ašimis, pradedant x ašimi.

ašys	s.riba	k	kk	s1	s2	skirtumas	ašys	s.riba	k	kk	s1	s2	skirtumas
xyz	2	2	4	2.6560	2.9557	25.6036	xzy	2	2	4	2.6560	2.9557	26.0936
		3	6	2.7132	2.9753	25.8378			3	6	2.7132	2.9753	24.9825
		4	8	2.7132	2.9753	28.5685			4	8	2.7132	2.9753	28.3074
		5	10	2.7117	2.9753	28.1078			5	10	2.7117	2.9753	28.5487
	3	2	4	2.6560	2.9557	29.7625		2	4	2.6560	2.9557	29.6998	
		3	6	2.7132	2.9753	31.3387		3	6	2.7132	2.9753	30.5450	
		4	8	2.7132	2.9967	33.6747		4	8	2.7132	2.9753	33.9009	
		5	10	2.7132	2.9967	32.1153		5	10	2.7117	3.0731	33.5432	
	4	2	4	0.6751	2.9753	34.1836		2	4	0.6751	2.9753	32.4302	
		3	6	0.9336	2.9753	38.2092		3	6	0.9336	2.9753	36.5297	
		4	8	0.9336	2.9967	40.6466		4	8	0.9336	2.9753	39.8922	
		5	10	0.9336	2.9967	39.5115		5	10	0.9292	3.0731	39.8585	
	5	2	4	0.6751	2.9753	37.5007		2	4	0.6751	2.9753	35.0944	
		3	6	0.9336	2.9753	42.0001		3	6	0.9336	2.9753	40.3916	
		4	8	0.9336	2.9791	44.6085		4	8	0.9336	2.9753	45.9613	
		5	10	0.9336	3.0105	45.3805		5	10	0.9292	2.9753	46.9092	

**3p.13 lentelė** Optimaliausių filtravimų atvaizdavimas, pradedant x ašimi.

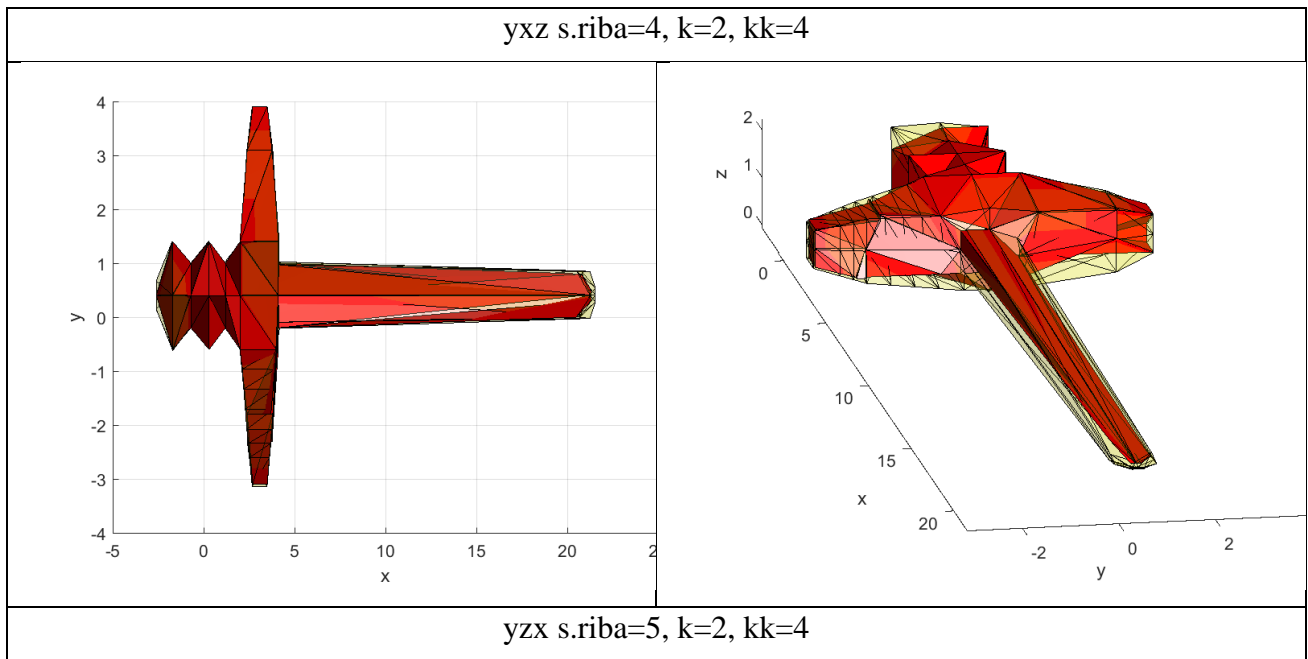


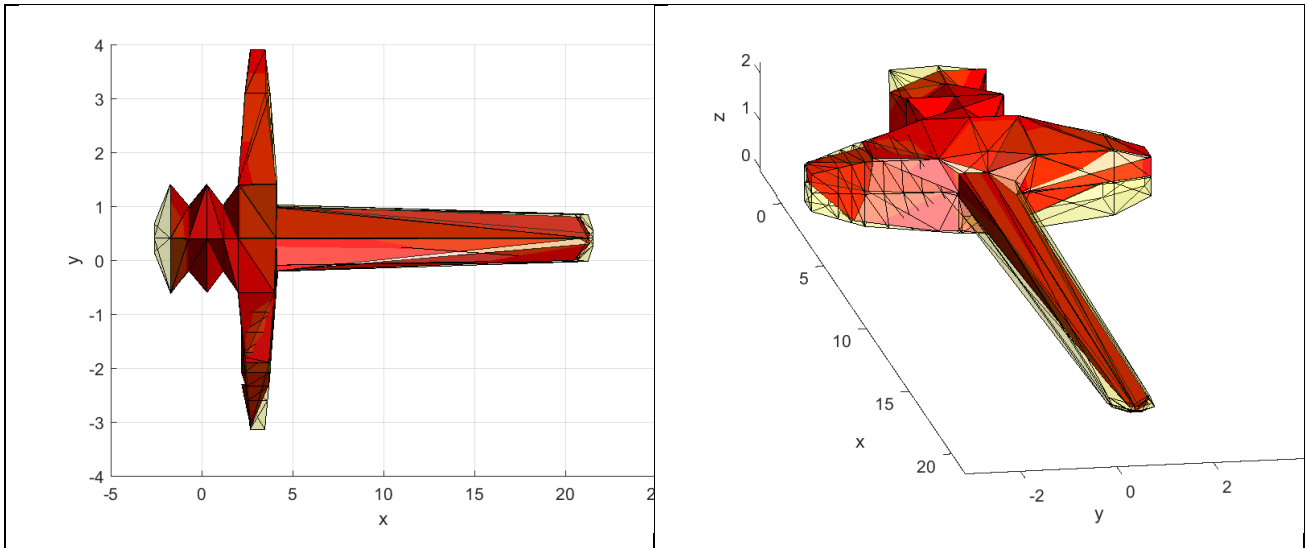
**3p.14 lentelė** Filtravimo rezultatai, filtruojant trimis ašimis, pradedant y ašimi.

ašys	s.riba	k	kk	s1	s2	skirtumas	ašys	s.riba	k	kk	s1	s2	skirtumas
yxz	2	2	4	2.656 0	2.975 3	24.6218	yzx	2	2	4	2.656 0	2.975 3	24.6218
		3	6	2.713 2	2.975 3	26.1215			3	6	2.713 2	2.975 3	26.2213
		4	8	2.711 7	2.975 3	28.2359			4	8	2.713 2	2.975 3	28.6610
		5	1 0	2.711 7	2.975 3	28.3025			5	1 0	2.711 7	2.975 3	28.3025
	3	2	4	2.656 0	2.975 3	28.9744		3	2	4	2.656 0	2.975 3	28.9814
		3	6	2.656 0	2.975 3	30.8697			3	6	2.656 0	2.975 3	30.7011
		4	8	2.656 0	2.975 3	31.9198			4	8	2.656 0	2.955 7	32.3702

	4	5	1 0	2.656 0	2.975 3	31.2667		4	5	1 0	2.656 0	2.975 3	31.5461
		2	4	0.101 2	2.975 3	32.2886			2	4	0.101 2	2.975 3	31.7009
		3	6	0.382 3	2.996 7	35.6454			3	6	0.371 4	2.975 3	35.0030
		4	8	0.751 2	2.996 7	38.8752			4	8	0.751 2	2.975 3	38.3722
		5	1 0	0.751 2	2.996 7	37.5199			5	1 0	0.751 2	2.975 3	36.6128
	5	2	4	0.101 2	2.975 3	35.0472		5	2	4	0.101 2	2.975 3	34.3906
		3	6	0.382 3	2.979 1	39.6950			3	6	0.371 4	2.975 3	39.7088
		4	8	0.751 2	3.010 5	43.8292			4	8	0.751 2	2.975 3	43.1691
		5	1 0	0.751 2	3.010 5	41.3626			5	1 0	0.751 2	2.975 3	40.7109

3p.15 lentelė Optimaliausių filtravimų atvaizdavimas, pradedant y ašimi.

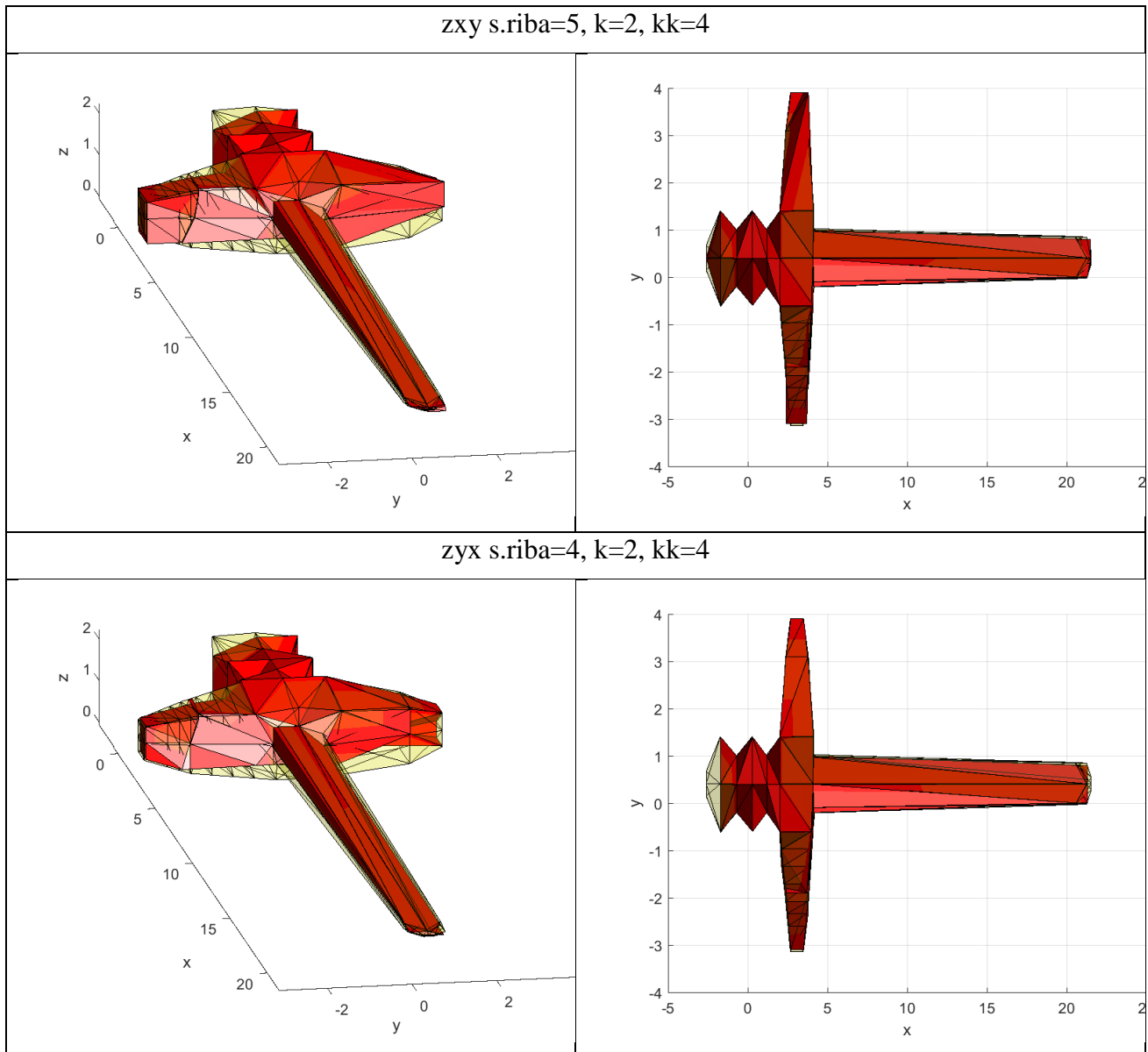




3p.16 lentelė Filtravimo rezultatai, filtruojant trimis ašimis, pradedant z ašimi.

ašys	s.riba	k	kk	s1	s2	skirtumas	ašys	s.riba	k	kk	s1	s2	skirtumas
zxy	2	2	4	2.6560	2.9557	26.0733	zyx	2	2	4	2.6560	2.9753	25.0915
		3	6	2.7132	2.9557	26.0387			3	6	2.7132	2.9753	26.2675
		4	8	2.7132	2.9753	28.8837			4	8	2.7132	2.9753	28.9782
		5	10	2.7117	2.9753	28.9145			5	10	2.7117	2.9753	28.4078
	3	2	4	2.6560	2.9557	29.7473		3	2	4	2.6560	2.9753	28.5595
		3	6	2.7132	2.9557	30.7981			3	6	2.6560	2.9753	30.9745
		4	8	2.7132	2.9753	34.1543			4	8	2.6560	2.9557	32.2426
		5	10	2.7117	3.0731	33.0622			5	10	2.7117	2.9753	32.3629
	4	2	4	0.6751	2.9753	31.6348		4	2	4	0.1012	2.9753	31.3551
		3	6	0.9336	2.9753	37.2933			3	6	0.3823	3.0731	35.0587
		4	8	0.9336	2.9753	40.3925			4	8	0.7512	2.9753	37.9310
		5	10	0.9292	3.5639	40.8505			5	10	0.7458	2.9753	40.0658
	5	2	4	0.6751	2.9753	34.5373		5	2	4	0.1012	2.9753	32.7233
		3	6	0.9336	2.9753	40.5049			3	6	0.3823	2.9753	38.9309
		4	8	0.9336	2.9753	46.8167			4	8	0.7512	2.9753	41.6069
		5	10	0.9292	2.9753	47.0077			5	10	0.7458	2.9753	44.8825

### 3p.17 lentelė Optimaliausių filtravimų atvaizdavimas, pradedant z ašimi.



## 4. Priedas. Statistinis pašalinio taško pakeitimas

### 4p.1 lentelė Duomenų normalizavimo kodas.

```
minimali=min(min(p_Vert));  
maksimali=max(max(p_Vert));  
ipas=length(p_Vert);  
P=[];  
for i = 1:ipas  
    for ii = 1:3  
        P(i,ii)=(pasikartojimai_Vert(i,ii)-minimali)/(maksimali-minimali);  
        p_Vert(i,ii)=P(i,ii);  
    end  
end  
end
```

#### 4p.2 lentelė Statistinio filtravimo funkcija.

```

vidvisu = mean(mean(d(:,1:k)));
[D,I] = sort(d(:,1));
I=I((length(I)*0.997):length(I));
Dkopija=D((length(D)*0.997):length(D));
koef=0;
I_ilgis=length(I);
for ii = 1:I_ilgis
    while vidvisu < Dkopija(ii)
        koef = Dkopija(ii)/vidvienas;
        for iii = 1:3
            if p_Vert(I(ii),iii) < p_Vert(indeksas(I(ii),1),iii)
                skirtumas=(p_Vert(indeksas(I(ii),1),iii)-p_Vert(I(ii),iii))/koef;
                p_Vert(I(ii),iii) = p_Vert(I(ii),iii)+skirtumas;
            elseif p_Vert(I(ii),iii) > p_Vert(indeksas(I(ii),1),iii)
                skirtumas=(p_Vert(I(ii),iii)-
p_Vert(indeksas(I(ii),1),iii))/koef;
                p_Vert(I(ii),iii) = p_Vert(I(ii),iii)-skirtumas;
            end
        end
        Dkopija(ii)=(p_Vert(I(ii),1)-
p_Vert(indeksas(I(ii),1),1))^2+(p_Vert(I(ii),2)-
p_Vert(indeksas(I(ii),1),2))^2+(p_Vert(I(ii),3)-p_Vert(indeksas(I(ii),1),3))^2;
        Dkopija(ii)=sqrt(Dkopija(ii));
    end
end
end

```

#### 4p.3 lentelė Normalizuotų duomenų gražinimo kodas.

```

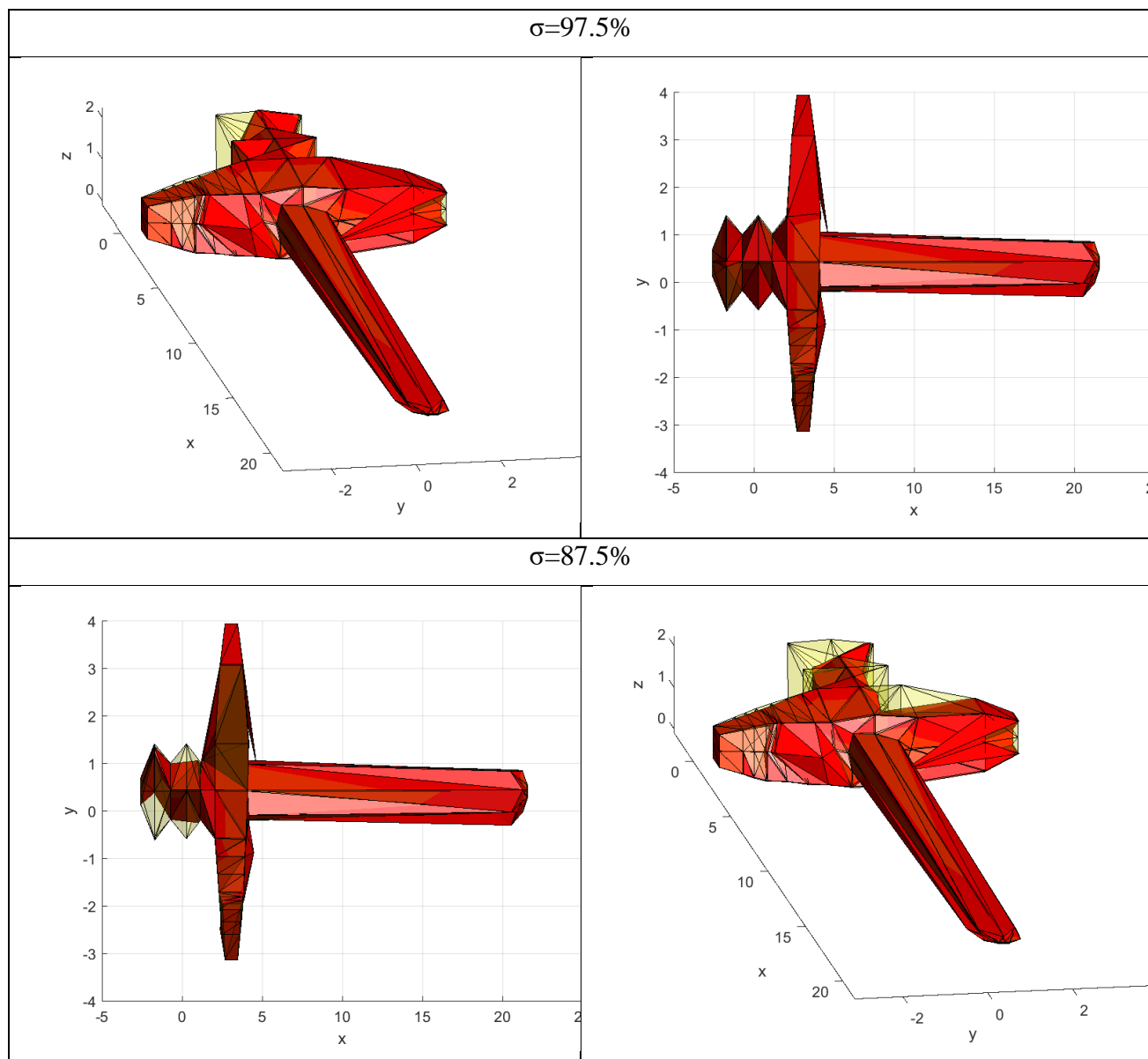
for i = 1:ipas
    for ii = 1:3
        p_Vert(i,ii)=p_Vert(i,ii)*(maksimali-minimali)+minimali;
    end
end
end

```

#### 4p.4 lentelė Statistinio filtravimo duomenys, kai naudota 1 kaimyninė reikšmė.

Lyginami kaimynai	d, 10 <sup>-4</sup>	k	σ, %	s1	s2	skirtumas
1	6.1263	3	97.5	0.3286	3.1458	14.6279
			95	0.3286	3.1458	17.3969
			92.5	0.3286	3.1458	20.1426
			90	0.3286	3.1458	22.9255
			87.5	0.3286	3.1458	26.2621
			85	0.3286	3.1458	28.1547
			82.5	0.3286	3.1458	29.7539
			80	0.3286	3.1458	31.1605

4p.5 lentelė Modelio atvaizdavimas po statistinio filtravimo, naudojant 1 kaimyninę reikšmę.

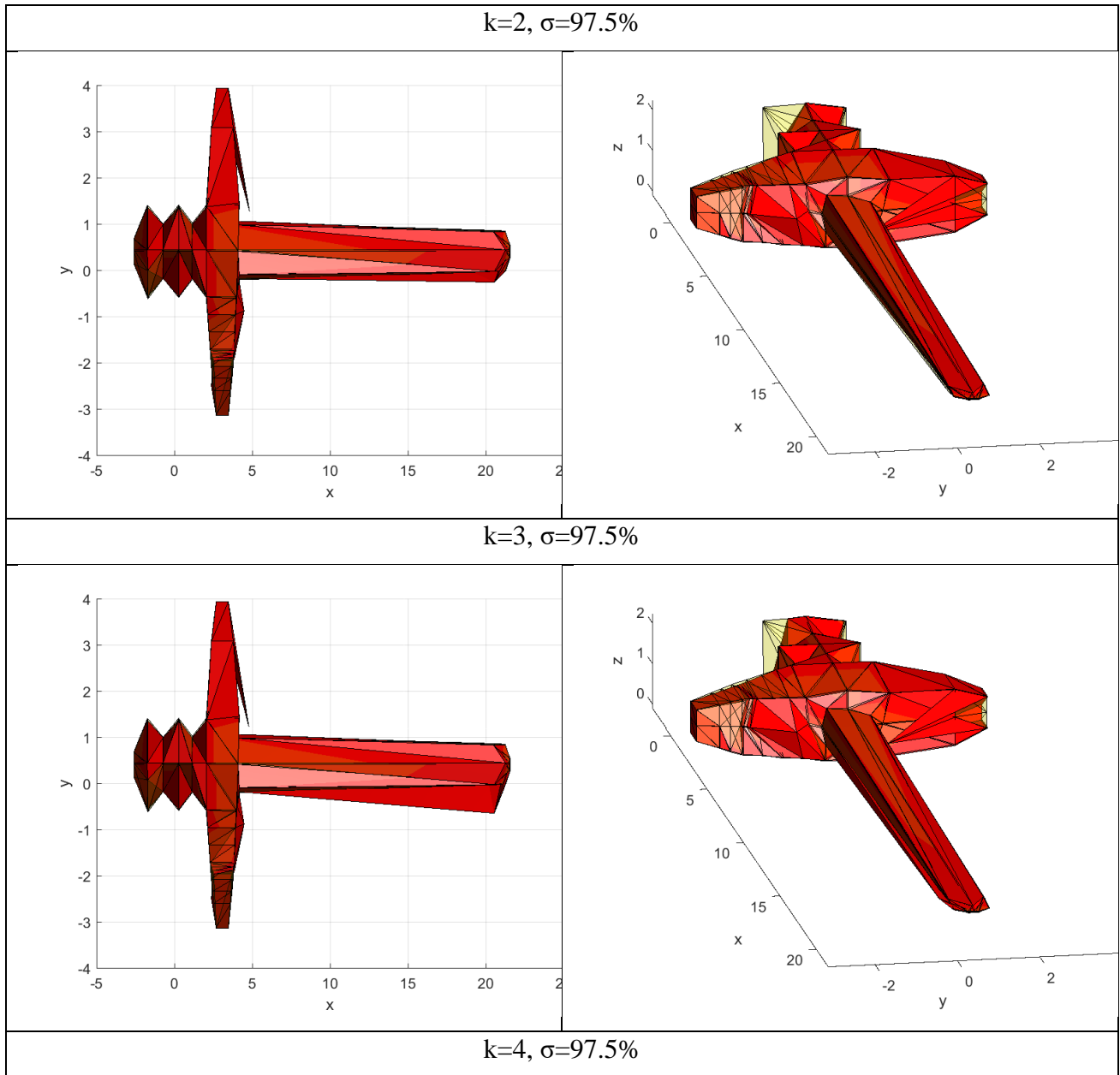


4p.6 lentelė Statistinio filtravimo duomenys, naudojant bendrą kaimyninę reikšmę.

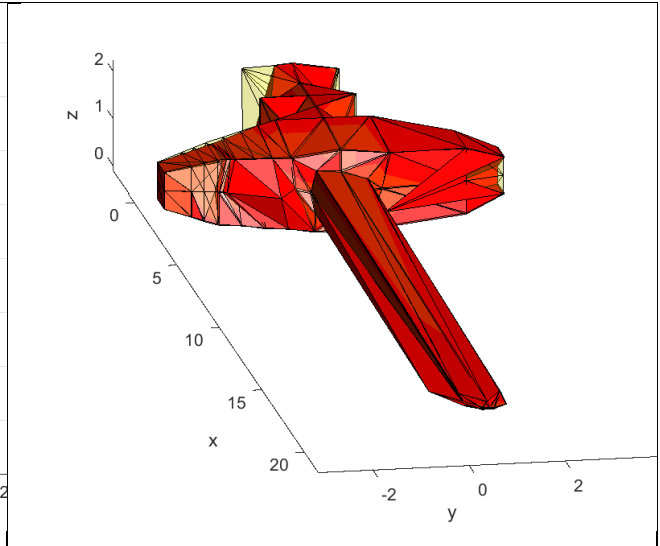
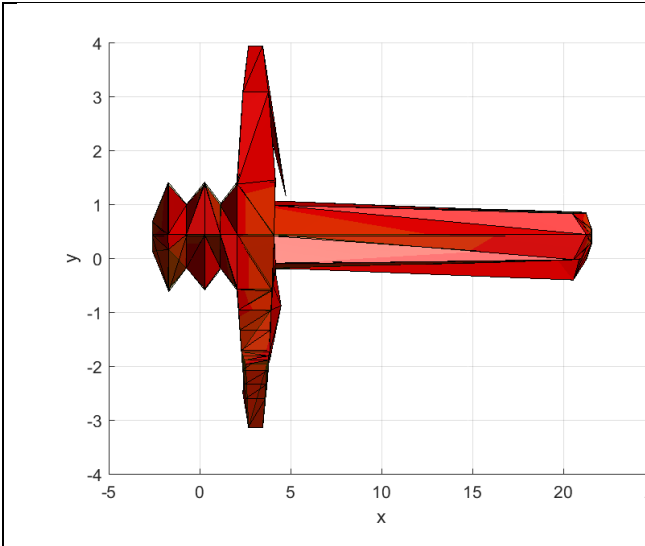
Lyginami kaimynai	d, $10^{-4}$	k	$\sigma$ , %	s1	s2	skirtumas
2	6.8048	2	97.5	0.1477	3.2043	14.1878
			95	0.1477	3.2043	16.5238
			92.5	0.1477	3.2043	18.0026
			90	0.1477	3.2043	19.5093
3	7.8116	3	97.5	0.2866	2.9735	13.7241
			95	0.2866	2.9735	16.3636
			92.5	0.2866	2.9735	18.0791
			90	0.2866	2.9735	19.8107
4	8.7233	4	97.5	0.6582	3.0067	14.2706
			95	0.6582	3.0067	16.4888
			92.5	0.6582	3.0067	18.4187
			90	0.6582	3.0067	20.3539
5	9.7880	5	97.5	0.4197	3.0472	14.2380

			95	0.4197	3.0472	16.6468
			92.5	0.4197	3.0472	18.8272
			90	0.4197	3.0472	21.0002
10	24	10	97.5	0.8606	2.7485	12.9497
			95	0.8606	2.7485	12.9497
			92.5	0.8606	2.7485	12.9497
			90	0.8606	2.7485	12.9497

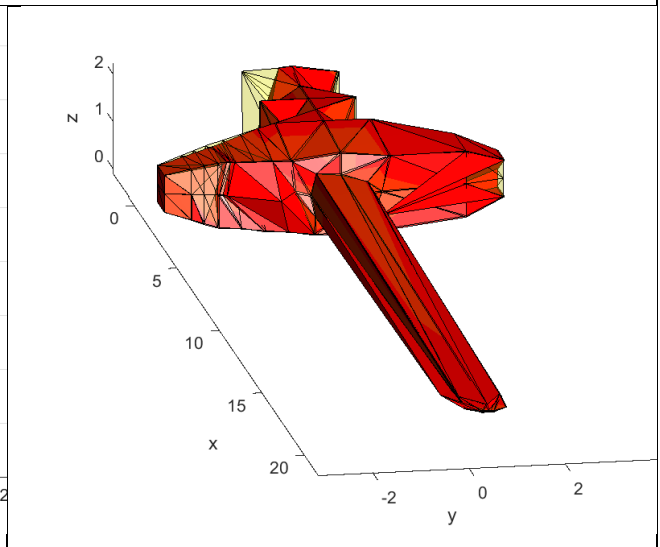
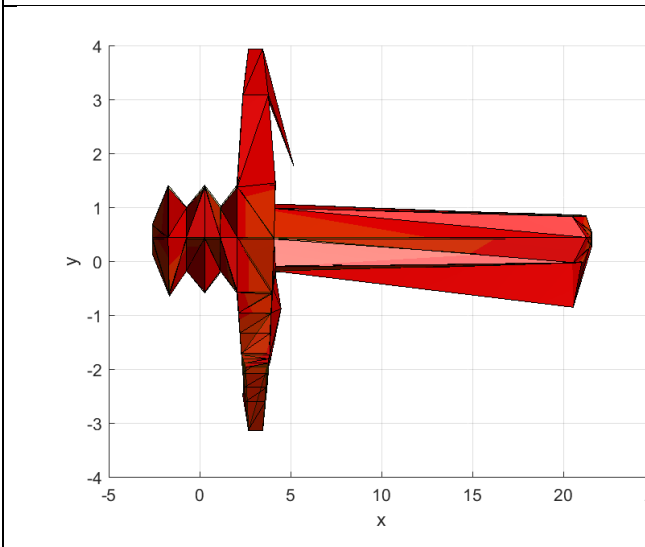
4p.7 lentelė Modelio atvaizdavimas po statistinio filtravimo, naudojant bendrą kaimyninę reikšmę.



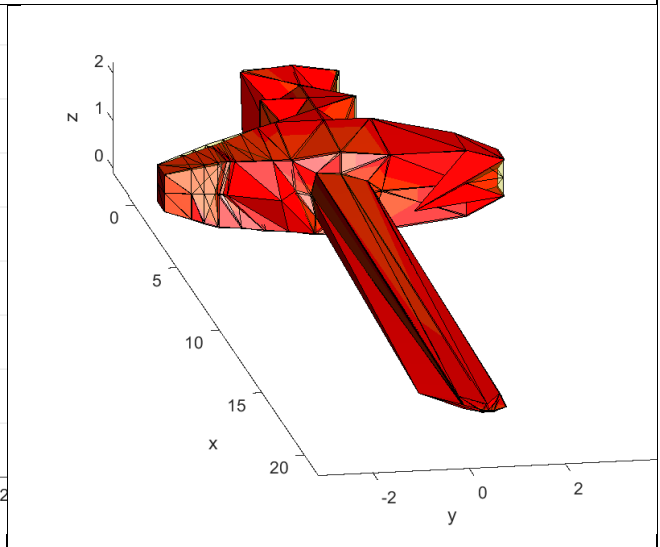
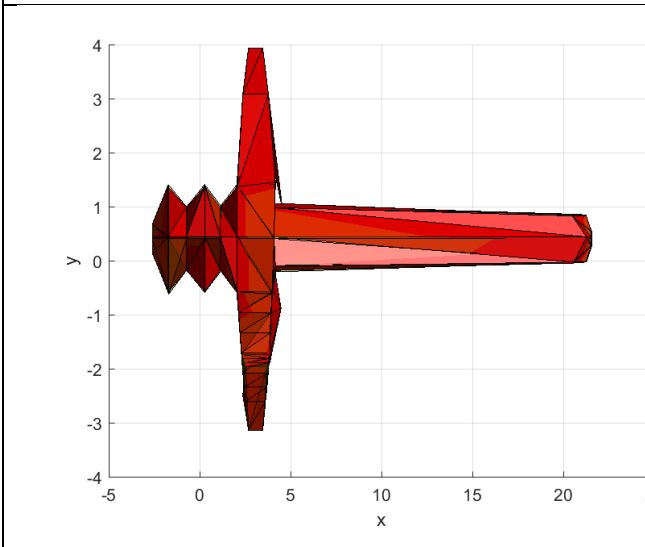




$k=5, \sigma=97.5\%$



$k=10, \sigma=97.5\%$



## 5. Priedas. Valdomas taškų filtravimas

### 5p.1 lentelė Funkcija paskaičiuojanti centroidines reikšmes

```
for x = 1:i
pi_sum(x,:)=zeros(1,3);
    for y = 1:k
        pi_sum(x,1)=pi_sum(x,1)+ p_Vert(indeksas(x,y),1);
        pi_sum(x,2)=pi_sum(x,2)+ p_Vert(indeksas(x,y),2);
        pi_sum(x,3)=pi_sum(x,3)+ p_Vert(indeksas(x,y),3);
    end
end
pi_sum=pi_sum/k;
```

### 5p.2 lentelė Funkcija paskaičiuojanti linijinio modelio $a_i$ koeficientą

```
pij_ss=zeros(1,3);
ai=zeros(1,3);
for xx = 1:i
    for yy = 1:k
        pij_ss(1,1)= pij_ss(1,1)+(p_Vert(indeksas(xx,yy),1))^2-(pi_sum(xx,1))^2;
        pij_ss(1,2)= pij_ss(1,2)+(p_Vert(indeksas(xx,yy),2))^2-(pi_sum(xx,2))^2;
        pij_ss(1,3)= pij_ss(1,3)+(p_Vert(indeksas(xx,yy),3))^2-(pi_sum(xx,3))^2;
    end
    skliaustai= pij_ss/k;
    ai(xx,:)=skliaustai/(skliaustai+epsi);
    pij_ss=zeros(1,3);
end
```

### 5p.3 lentelė Funkcija randanti linijinio modelio $b_i$ koeficientą bei paskaičiuoja pakeistas reikšmes

```
for xxx = 1:i
    bi(xxx,1)=pi_sum(xxx,1)-(ai(xxx,1)*pi_sum(xxx,1));
    bi(xxx,2)=pi_sum(xxx,2)-(ai(xxx,2)*pi_sum(xxx,2));
    bi(xxx,3)=pi_sum(xxx,3)-(ai(xxx,3)*pi_sum(xxx,3));
    pi(xxx,1)=ai(xxx,1)*pi_sum(xxx,1)+bi(xxx,1);
    pi(xxx,2)=ai(xxx,2)*pi_sum(xxx,2)+bi(xxx,2);
    pi(xxx,3)=ai(xxx,3)*pi_sum(xxx,3)+bi(xxx,3);
end
```

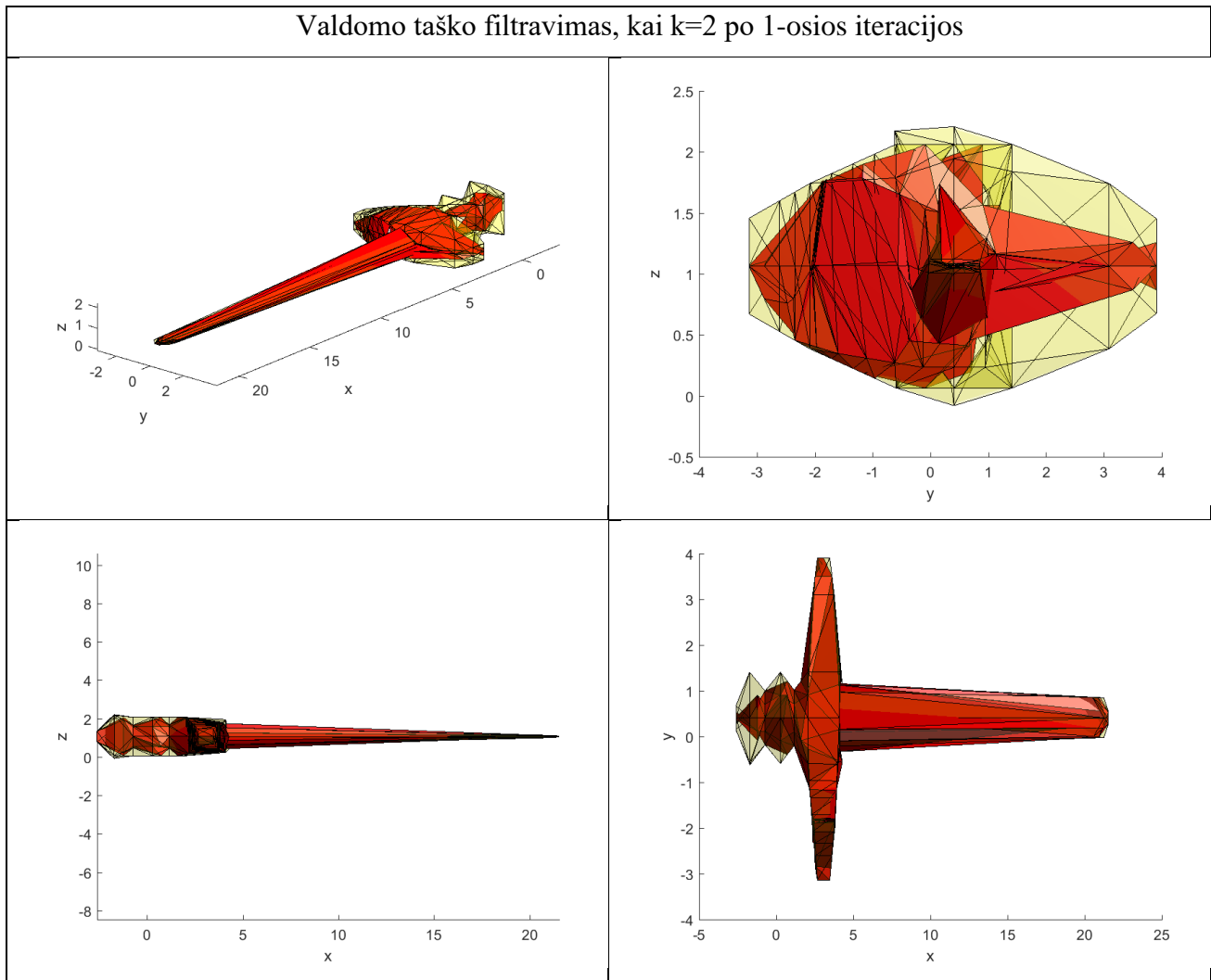
### 5p.4 lentelė Valdomo taško duomenys, kai $k=2$ .

k	iteracija	s1	s2	skirtumas
2	1	0.3037	2.9136	47.2205
	2	0.7727	2.9077	61.9766
	3	0.9169	2.8937	66.1795
	4	0.9652	2.8398	69.4517
	5	0.9733	2.8937	71.8830
	10	0.9761	2.8398	74.3855

5p.5 lentelė Valdomo taško metodu filtruotas modelis, kai  $k=2$ , skirtingų iteracijų metu.

		Kaimynų skaičius $k=2$	
Iteracija 1		Iteracija 2	
			Iteracija 4
Iteracija 5		Iteracija 6	

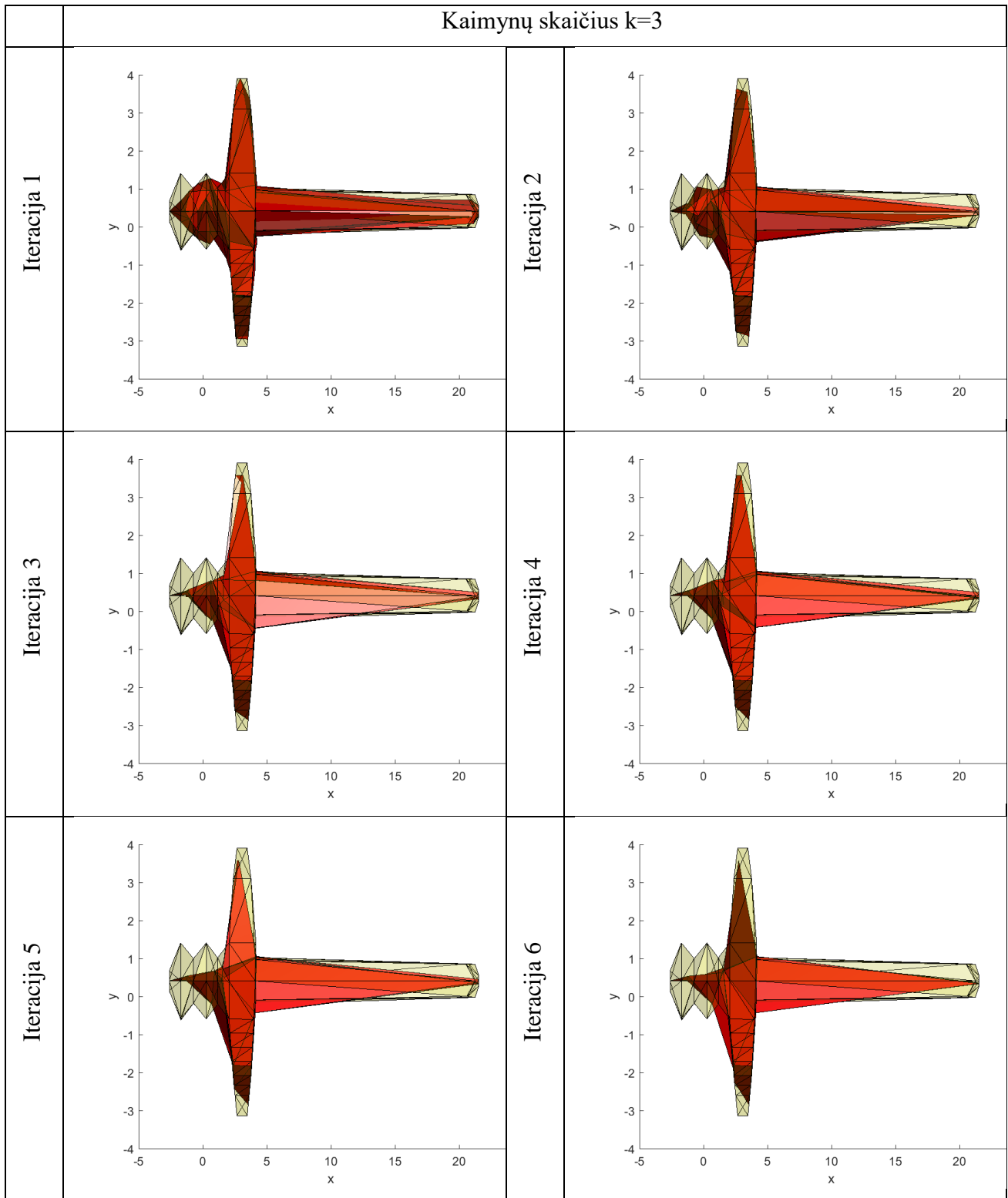
5p.6 lentelė Filtruoto modelio skirtingų projekcijų vaizdai, kai  $k=2$  po 1-osios iteracijos.



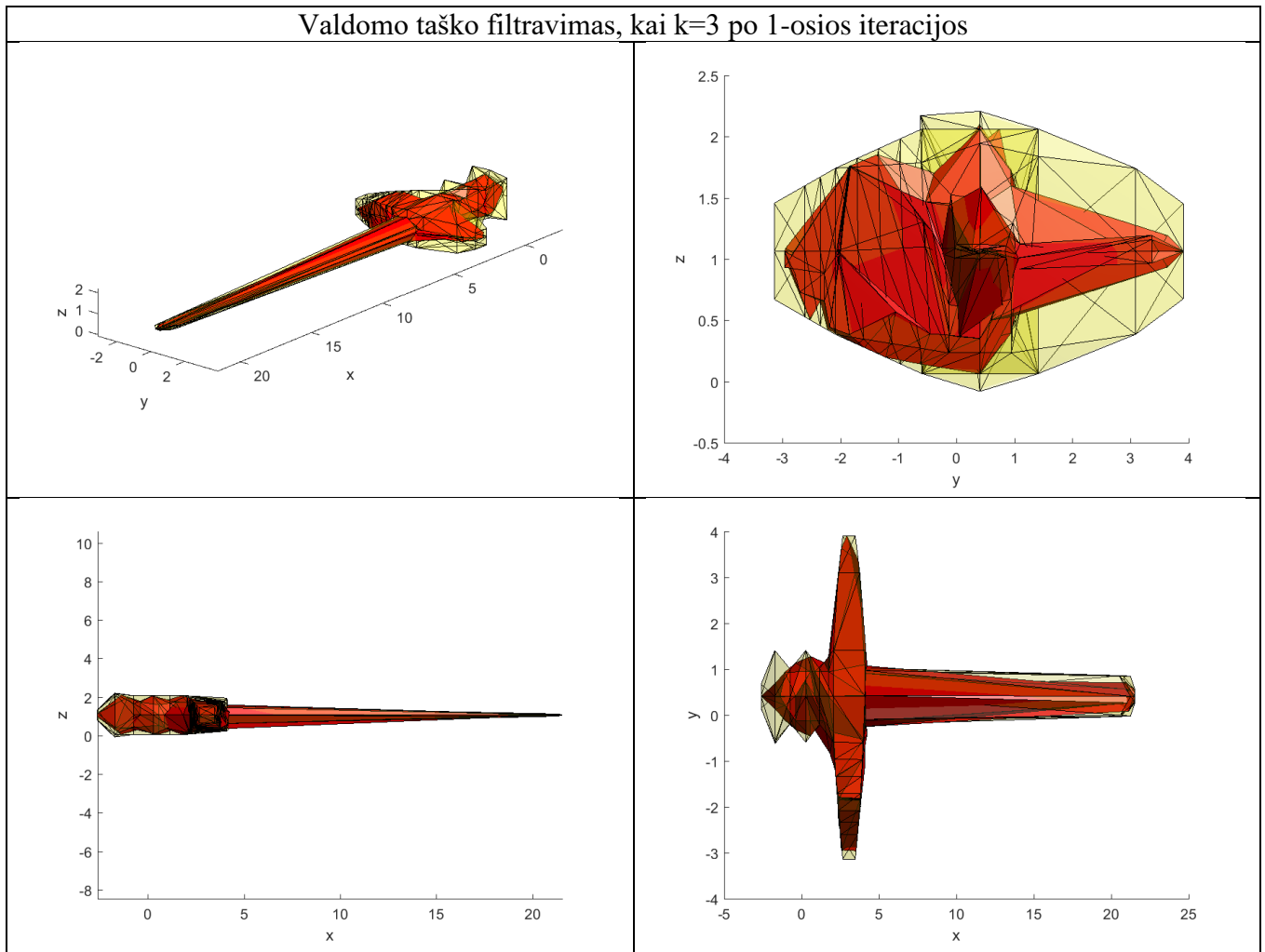
5p.7 lentelė Valdomo taško duomenys, kai  $k=3$ .

k	iteracija	s1	s2	skirtumas
3	1	0.5172	2.9225	50.2660
	2	0.8478	2.9467	68.6013
	3	0.9142	2.9325	76.7256
	4	0.9896	2.9413	82.5550
	5	0.9850	2.9343	85.2846
	10	0.9851	2.9366	89.8630

5p.8 lentelė Valdomo taško metodu filtruotas modelis, kai  $k=3$ , skirtingų iteracijų metu.



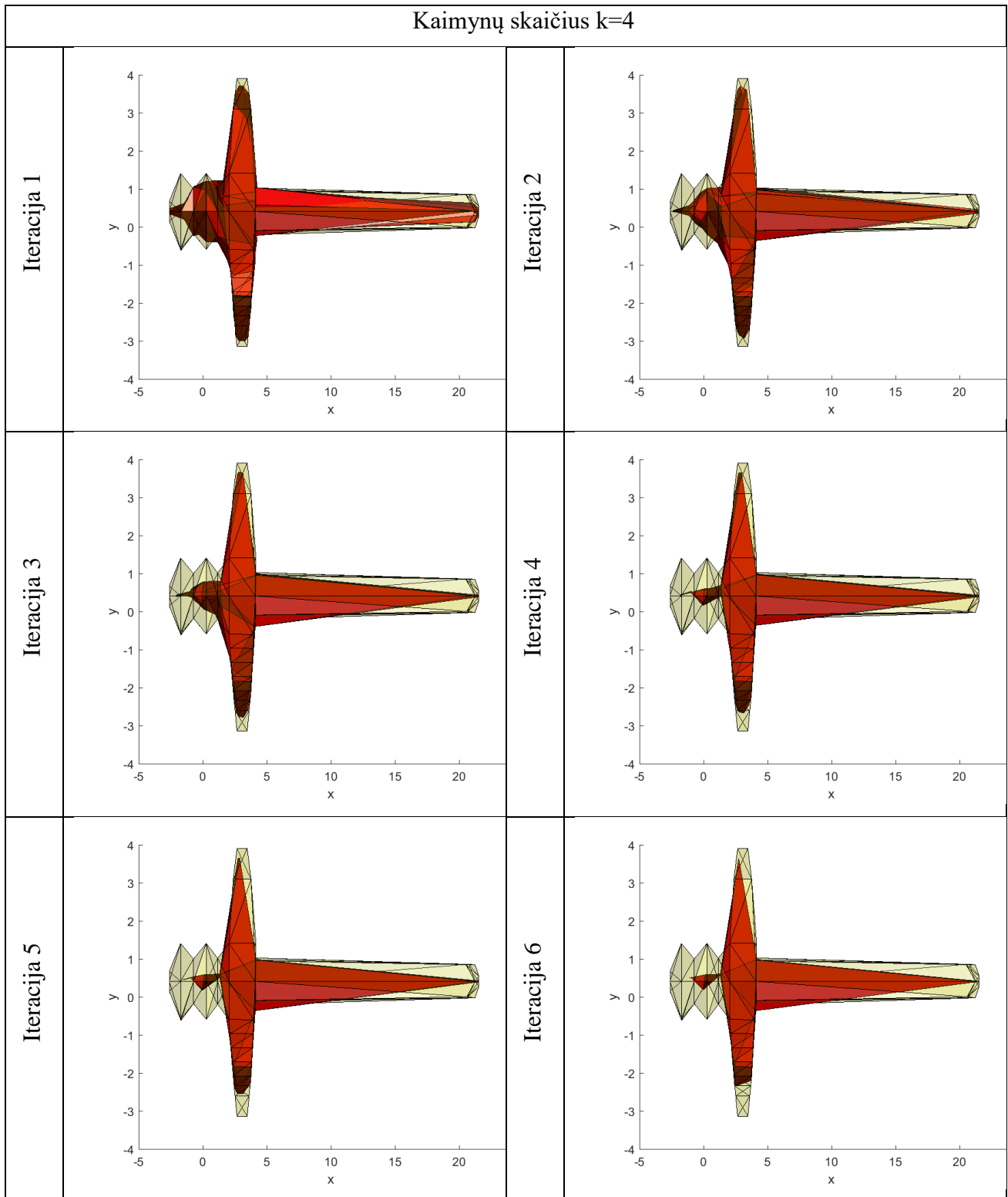
**5p.9 lentelė** Filtruoto modelio skirtingų projekcijų vaizdai, kai  $k=3$  po 1-osios iteracijos.



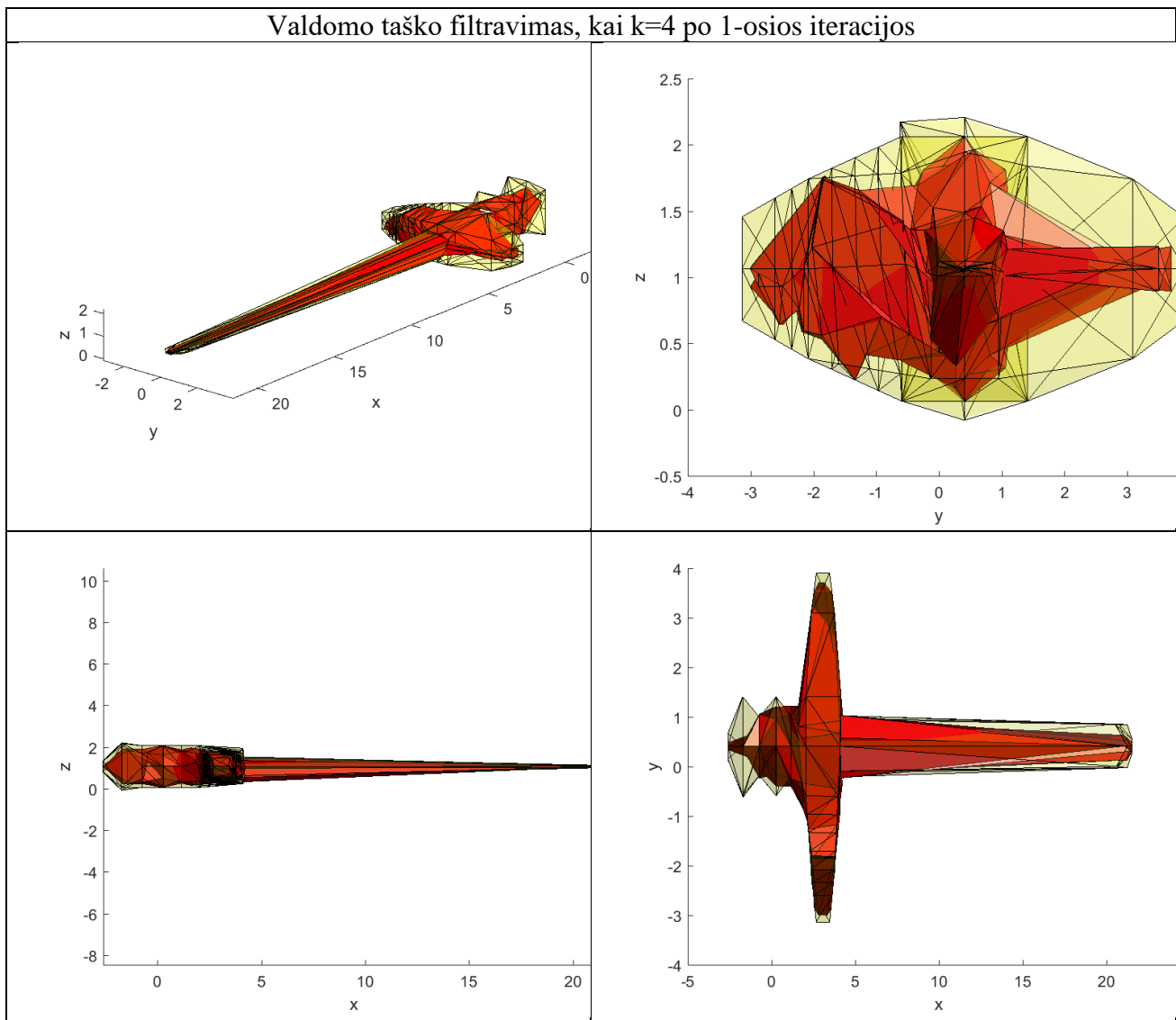
**5p.10 lentelė** Valdomo taško duomenys, kai  $k=4$ .

k	iteracija	s1	s2	skirtumas
4	1	0.6276	2.9401	49.2058
	2	0.8493	2.9982	66.9965
	3	0.9359	3.0058	77.3050
	4	1.0071	3.0058	83.3626
	5	1.0078	3.0078	85.3410
	10	1.0076	3.0074	87.6477

5p.11 lentelė Valdomo taško metodu filtruotas modelis, kai  $k=4$ , skirtingų iteracijų metu.



5p.12 lentelė Filtruoto modelio skirtingų projekcijų vaizdai, kai  $k=4$  po 1-osios iteracijos.

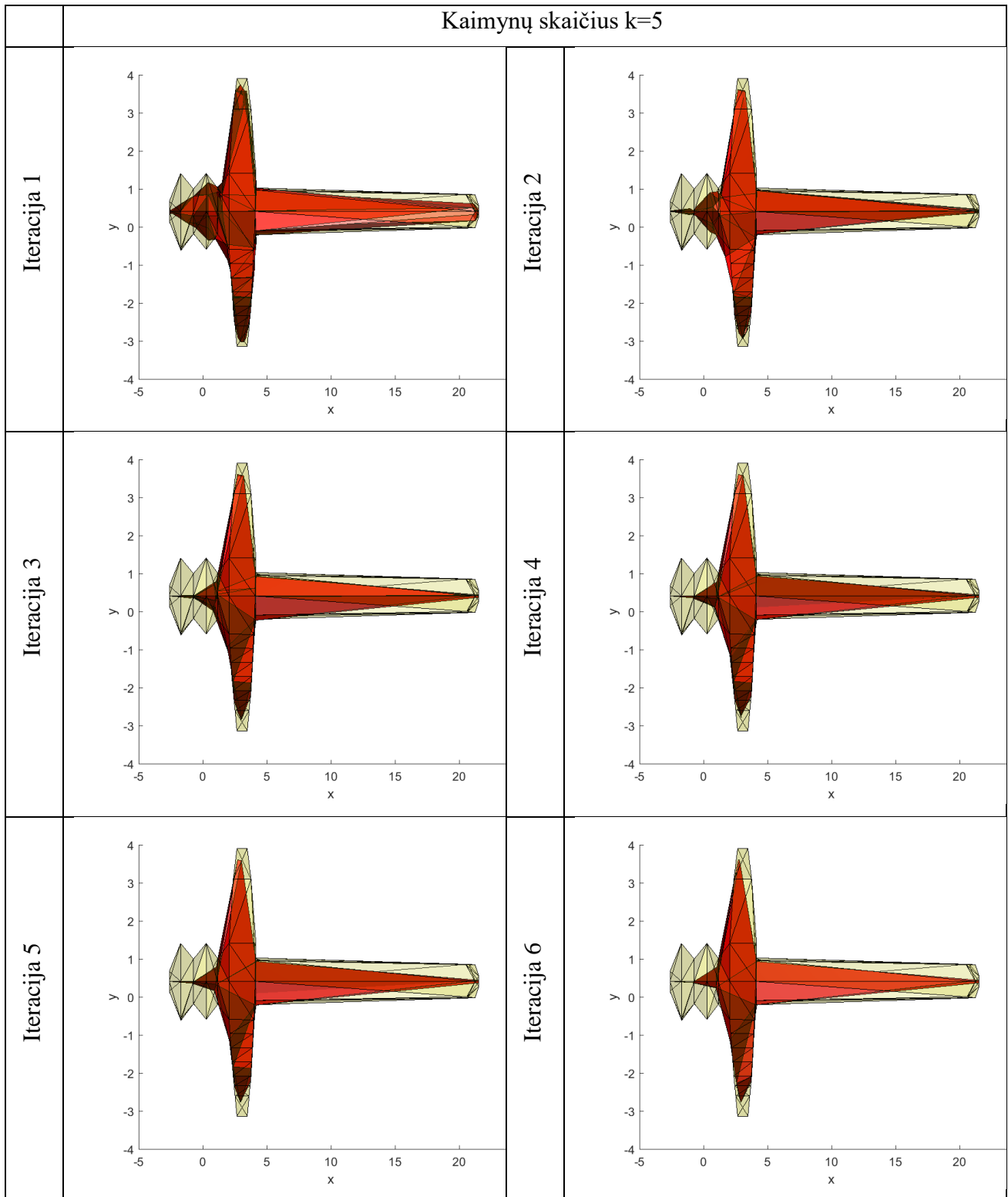


5p.13 lentelė Valdomo taško duomenys, kai  $k=5$ .

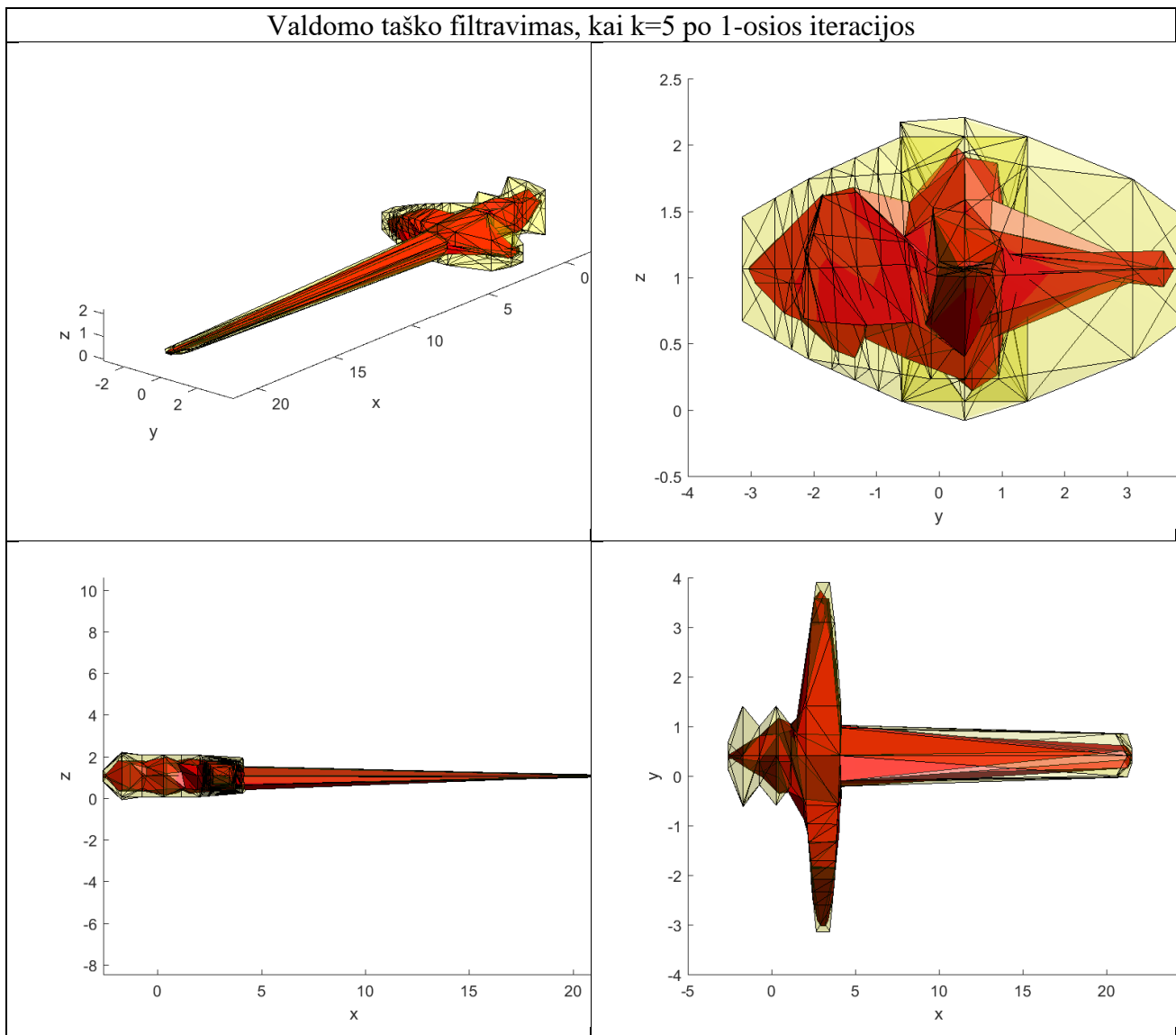
k	iteracija	s1	s2	skirtumas
5	1	0.6942	2.8704	54.2879
	2	0.9650	3.0562	74.2312
	3	0.9947	3.0525	85.4715
	4	0.9968	3.0532	89.3686
	5	0.9978	3.0531	90.8947
	10	0.9979	3.0532	94.2879



5p.14 lentelė Valdomo taško metodu filtruotas modelis, kai  $k=5$ , skirtingų iteracijų metu.



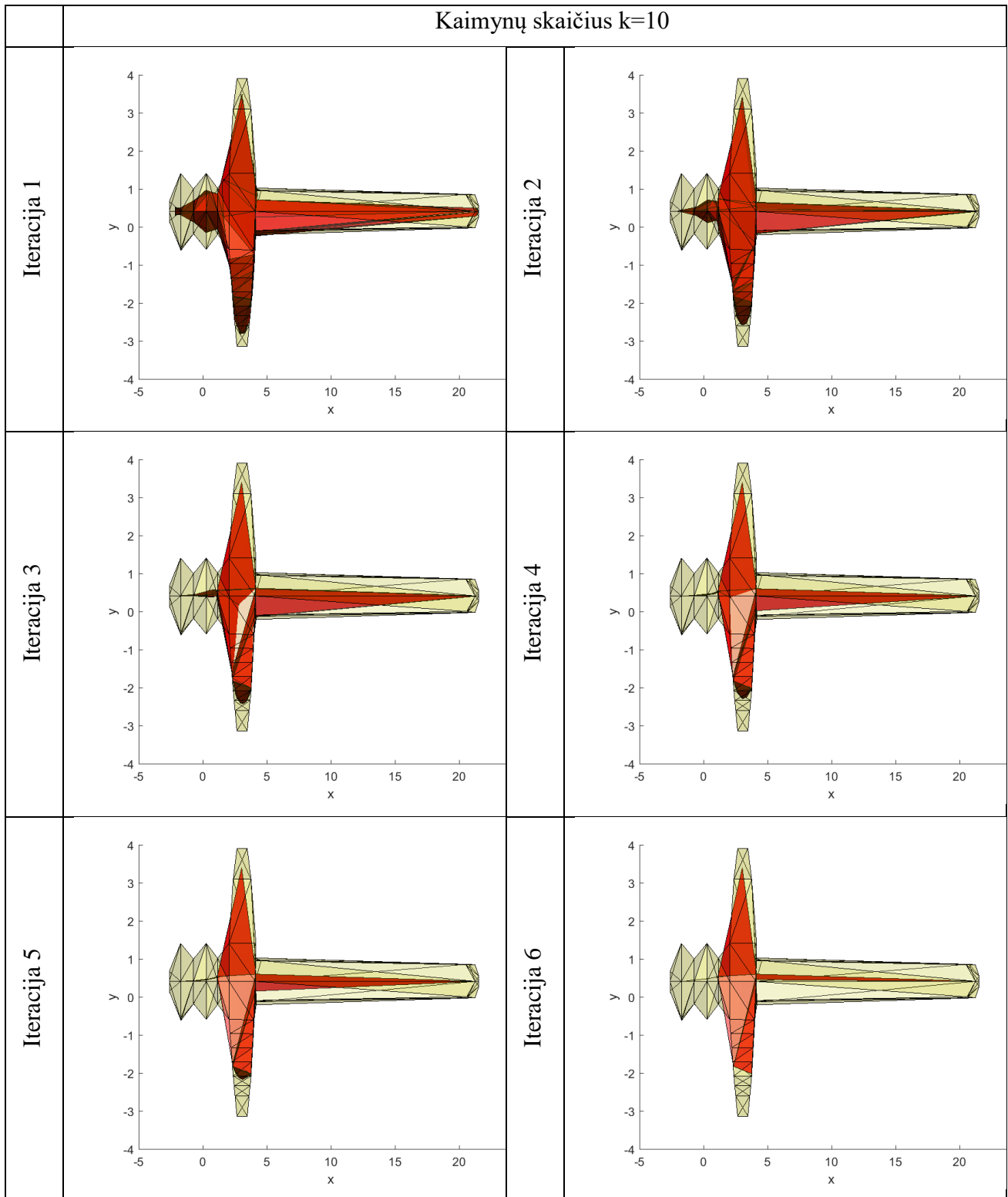
**5p.15 lentelė** Filtruoto modelio skirtingų projekcijų vaizdai, kai  $k=5$  po 1-osios iteracijos.



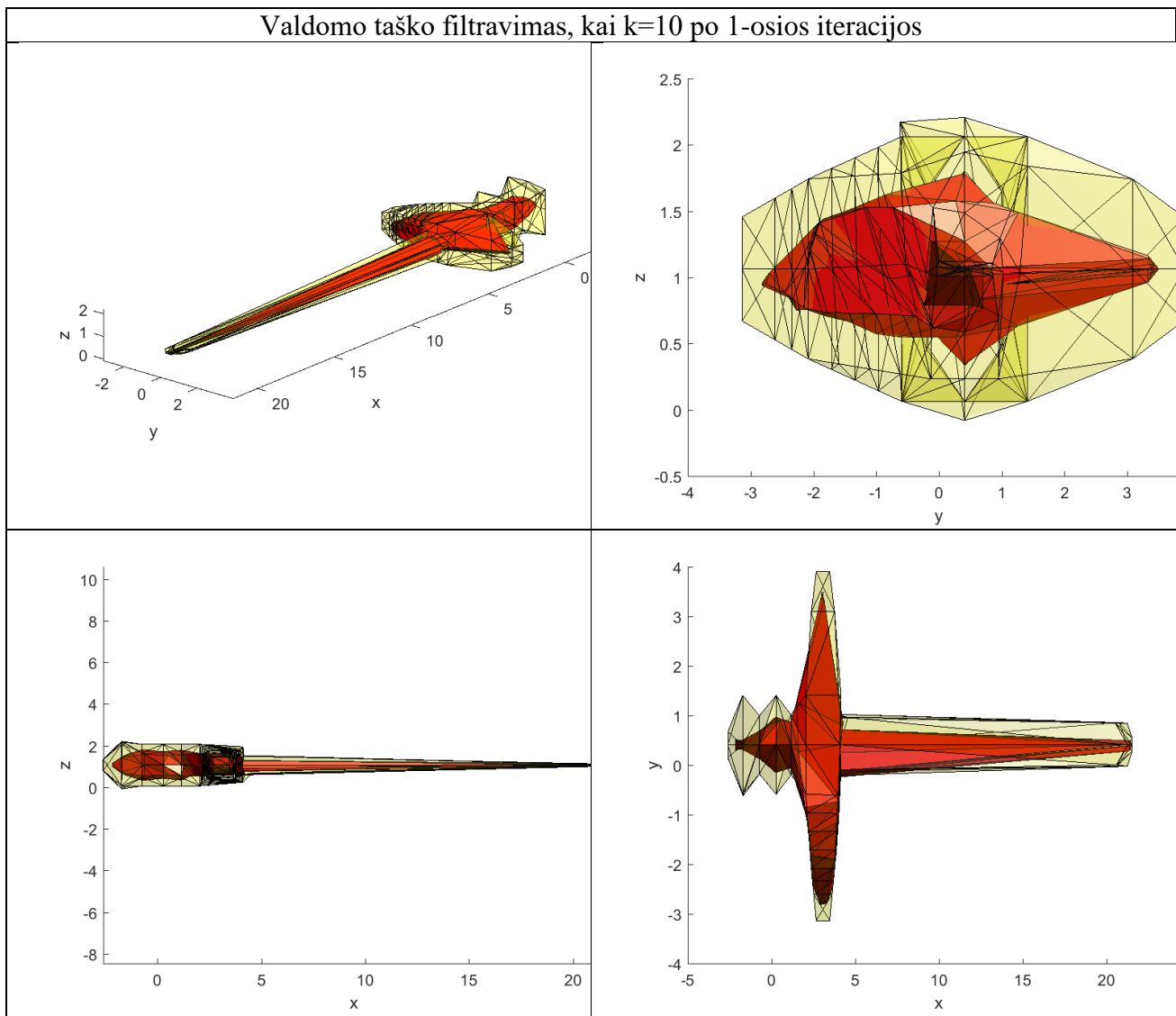
**5p.16 lentelė** Valdomo taško duomenys, kai  $k=10$ .

k	iteracija	s1	s2	skirtumas
10	1	0.7974	2.8896	65.0613
	2	0.9425	3.3823	92.3628
	3	0.9637	3.3781	105.5933
	4	0.9638	3.3786	110.9078
	5	0.9638	3.3785	114.1974
	10	0.9638	3.3785	116.6532

5p.17 lentelė Valdomo taško metodu filtruotas modelis, kai  $k=10$ , skirtingų iteracijų metu.



**5p.18 lentelė** Filtruoto modelio skirtingų projekcijų vaizdai, kai k=10 po 1-osios iteracijos.



## 6. Priedas. Apjungtų metodų filtravimas

**6p.1 lentelė** Koordinatės pakeitimas stipriuoju tašku – Valdomas taškų filtravimas

ašys	s.riba	(Koordinatės pakeitimas stipriuoju) k	kk	(Valdomas taškų) k	iteracija	s1	s2	skirtumas
xyz	4	2	4	2	1	0.7512	2.9759	58.3447
xzy						0.7512	2.9759	59.0395
yxz						0.8414	2.9657	57.6910
yzx						0.8414	2.9657	57.6910
zxy						0.7512	2.9759	58.7866
zyx						0.8414	2.9657	58.1329
xyz	4	2	4	3	1	0.8111	3.0079	61.0642
xzy						0.8111	3.0079	61.7147
yxz						0.8720	3.0011	60.3567

yzx						0.8720	3.0011	60.3567
zxy						0.8111	3.0079	61.2332
zyx						0.8720	3.0011	60.5257
xyz	4	2	4	4	1	0.8414	3.1465	57.6334
xzy						0.8414	3.1465	58.4978
yxz						0.8851	3.1416	57.0489
yzx						0.8851	3.1416	57.0489
zxy						0.8414	3.1465	58.1640
zyx						0.8851	3.1416	57.5794
xyz	4	2	4	5	1	0.8578	3.2294	63.5738
xzy						0.8578	3.2294	64.0354
yxz						0.8657	3.2256	63.0228
yzx						0.8657	3.2256	63.0228
zxy						0.8578	3.2294	63.3630
zyx						0.8657	3.2256	62.8127
xyz	4	2	4	10	1	0.8776	3.2982	74.9769
xzy						0.8776	3.2982	75.2925
yxz						0.8376	3.2963	74.6053
yzx						0.8376	3.2963	74.6053
zxy						0.8776	3.2982	74.3226
zyx						0.8376	3.2963	73.9506

**6p.2 lentelė** Valdomas taškų filtravimas – Koordinatės pakeitimas stipriuoju tašku

(Valdomas taškų) k	iteracija	ašys	s.riba	(Koordinatės pakeitimas stipriuoju) k	kk	s1	s2	skirtumas
2	1	xyz	4	2	4	0.8414	2.7842	65.2893
		xzy				0.8414	2.7842	64.1912
		yxz				0.8003	2.7842	64.7609
		yzx				0.8855	2.7842	66.4034
		zxy				0.8414	2.7842	64.5108
		zyx				0.8855	2.7842	66.0058
3	1	xyz	4	2	4	0.8414	2.9050	68.3888
		xzy				0.8414	2.9050	68.4143
		yxz				0.8003	2.9050	69.1004
		yzx				0.8003	2.9050	69.6965

		zxy				0.8414	2.9050	68.2810
		zyx				0.8003	2.9050	69.0810
4	1	xyz	4	2	4	0.9336	2.9011	62.5445
		xzy				0.9336	2.9011	64.8416
		yxz				0.5806	2.9011	67.0458
		yzx				0.5806	2.9011	66.0062
		zxy				0.9336	2.9011	65.7344
		zyx				0.5806	2.9011	67.6103
5	1	xyz	4	2	4	0.9736	3.0005	69.8871
		xzy				0.9736	3.0005	69.8476
		yxz				0.8855	3.0005	69.1755
		yzx				0.8855	3.0005	69.3518
		zxy				0.9736	3.0005	69.2405
		zyx				0.8855	3.0005	69.5975
10	1	xyz	4	2	4	0.7512	3.0767	73.8366
		xzy				0.7512	3.0767	73.7229
		yxz				0.7512	3.0767	73.2383
		yzx				0.7512	3.0767	73.4824
		zxy				0.7512	3.0767	73.6278
		zyx				0.7512	3.0767	74.1266

6p.3 lentelė Statistinis taško pakeitimas – Valdomas taškų filtravimas

Lyginami kaimynai	d, 10 <sup>-4</sup>	(Statistinis) k	σ, %	(Valdomo taško) k	iteracija	s1	s2	skirtumas
2	6.8048	2	97.5	2	1	0.2989	3.0527	47.8179
3	7.8116	3				0.2989	3.0527	47.2106
4	8.7233	4				0.2989	3.0527	47.3716
5	9.7880	5				0.2989	3.0527	47.3560
10	24	10				0.2989	2.9302	47.1835
2	6.8048	2	97.5	3	1	0.4441	3.0541	51.3467
3	7.8116	3				0.4441	2.9217	50.8046
4	8.7233	4				0.4441	2.9217	51.2343
5	9.7880	5				0.4441	2.9217	50.9948
10	24	10				0.4441	2.9217	50.5317
2	6.8048	2	97.5	4	1	0.5220	3.1770	49.7701
3	7.8116	3				0.5220	2.9359	48.7545

4	8.7233	4				0.5220	2.9359	48.8577
5	9.7880	5				0.5220	2.9359	49.7352
10	24	10				0.5220	2.9359	48.7539
2	6.8048	2	97.5	5	1	0.5025	3.0712	54.7865
3	7.8116	3				0.5025	2.9531	54.2758
4	8.7233	4				0.6151	2.9531	54.3340
5	9.7880	5				0.5025	2.9531	54.4295
10	24	10				0.6151	2.8620	54.1918
2	6.8048	2				97.5	10	1
3	7.8116	3	0.6991	3.1951	65.3225			
4	8.7233	4	0.7438	3.1951	65.3824			
5	9.7880	5	0.6991	3.1951	65.4537			
10	24	10	0.8227	2.8737	65.0269			

**6p.4 lentelė** Valdomas taškų filtravimas – Statistinis taško pakeitimas

(Valdomų taškų) k	iteracija	Lyginami kaimynai	d, 10 <sup>-4</sup>	(Statistinio taško) k	σ, %	s1	s2	skirtumas
2	1	2	6.8048	2	97.5	0.3486	2.9070	49.6417
		3	7.8116	3		0.5096	2.9197	51.6779
		4	8.7233	4		0.6307	2.9675	51.3698
		5	9.7880	5		0.6896	2.8377	55.5607
		10	24	10		0.7883	2.9084	66.3496
3	1	2	6.8048	2	97.5	0.3486	2.9070	49.7685
		3	7.8116	3		0.5096	2.9197	51.1113
		4	8.7233	4		0.6307	2.9675	50.8382
		5	9.7880	5		0.6896	2.8377	55.3845
		10	24	10		0.7883	2.9084	66.4540
4	1	2	6.8048	2	97.5	0.3486	2.9070	49.4763
		3	7.8116	3		0.5096	2.9197	51.1776
		4	8.7233	4		0.6307	2.9675	50.7419
		5	9.7880	5		0.6896	2.8377	55.3210
		10	24	10		0.7883	2.9084	66.1748
5	1	2	6.8048	2	97.5	0.3486	2.9070	49.3722
		3	7.8116	3		0.5096	2.9197	51.0902
		4	8.7233	4		0.6307	2.9675	51.0320
		5	9.7880	5		0.6896	2.8377	55.2089

		10	24	10		0.7883	2.9084	66.0730
10	1	2	6.8048	2	97.5	0.3486	2.9070	48.3754
		3	7.8116	3		0.5096	2.9197	50.1456
		4	8.7233	4		0.6307	2.9675	49.5493
		5	9.7880	5		0.6896	2.8377	54.2771
		10	24	10		0.7883	2.9084	65.3113