

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
STUDIJŲ PROGRAMA: INFORMATIKA

**Generatyviniais besivaržančiais tinklais sukurtų 3D
modelių tikslumo ir tikroviškumo įvertinimas**

**Evaluating the Accuracy and Realism of 3D Models Generated
by Generative Adversarial Networks**

Baigiamasis magistro darbas

Atliko: Justina Maslovaite (parašas)

Darbo vadovas: prof., dr. Olga Kurasova (parašas)

Recenzentas: asist., dr. Linas Litvinas (parašas)

Vilnius – 2023

Santrauka

Šiame tyrime nagrinėjami 3D-GAN generatyvinių besivaržančių neuroninių tinklų rezultatų įvertinimo metodai ir giliojo mokymosi taikymas trimačių modelių generavime. Literatūros apžvalgoje išsamiai aptariami įvairių 3D reprezentacijos formų pranašumai ir trūkumai bei pateikiami bendri giliojo mokymosi iš įvairių 3D reprezentacijos formų metodai. Tyrimas apima 3D-GAN sukurtų trimačių modelių panašumo ir realumo nustatymą, naudojant įvairias panašumo įvertinimo metrikas. Tyrimo metu siekiama nustatyti kokybės užtikrinimo kriterijus 3D-GAN generatyvinių besivaržančių neuroninių tinklų rezultatų palyginimui su nustatyta tiesa, kartu analizuojant papildomus veiksnius darančius įtaką panašumo įverčiams. Nagrinėjami tokie veiksniai, kaip trimačių vokselių tinklelio dydžio poveikis panašumo metrikų įverčiams bei trimačių vokselių tinklelių tankumo įtaką panašumo įvertinime. Apibendrinant, šiuo tyrimu pabrėžiami kylantys iššūkiai vertinant generatyvinių besivaržančių neuroninių tinklų sukurtų 3D modelių kokybę ir tikroviškumą.

Raktiniai žodžiai: kompiuterinė rega, gilusis mokymasis, 3D modelio generavimas, 3D modelių įvertinimas

Summary

This study explores the use of deep learning techniques with three-dimensional representational forms. The literature review outlines common methods for deep learning from various 3D data formats as well as the benefits and drawbacks of 3D representational forms. The evaluation and comparison of the 3D-GAN generative adversarial neural network's generation outputs are the primary objectives of the study. The research involves measuring the resemblance and realism of 3D-GAN derived 3D models by employing a variety of similarity metrics to examine the similarity between the generated 3D voxel grids and the ground truth. The study also examines the impact of 3D voxel grid density and resolution on the estimate of similarity measures to better understand the elements impacting 3D voxel grid similarity scores. Overall, the study tackles recurring challenges in the evaluation of quality and realism of 3D-GAN produced models.

Keywords: computer vision, deep learning, 3D model generation, 3D model evaluation

Turinys

Santrauka	1
Summary	2
1. Įvadas	6
1.1. Tiriamoji problema.....	6
1.2. Tyrimo tikslas	7
1.3. Tyrimo uždaviniai	7
2. Literatūros apžvalga	8
2.1. Trimačių duomenų reprezentacijos formos	8
2.2. Giliojo mokymosi iš trimačių duomenų metodų apžvalga	9
2.2.1. Gilusis mokymasis iš kelių vaizdų	9
2.2.2. Gilusis mokymasis iš taškų rinkinio	10
2.2.3. Gilusis mokymasis iš vokselių tinklelio.....	11
2.2.4. Gilusis mokymasis iš <i>octree</i> medžio	13
2.3. Neuroninių tinklų architektūros	14
2.3.1. Konvoliuciniai neuroniniai tinklai	14
2.3.2. Generatyviniai besivaržantys tinklai	17
2.4. GAN rezultatų kokybės ir realumo įvertinimas	19
2.5. Apibendrinimas	22
3. Tiriamoji dalis	24
3.1. 3D-GAN modelis	24
3.2. Sørensen-Dice koeficientas	25
3.3. Jaccard indeksas.....	28
3.4. Hausdorff atstumas	29
3.5. Chamfer atstumas	36
3.6. Vokselių tinklelio tankumo įtaka 3D modelių panašumo įverčiams	38
3.7. 3D vokselių tinklelio dydžio įtaka 3D modelių panašumo vertinime	40
3.8. 3D modelių tikroviškumo įvertinimas	42
3.9. Vidutinio 3D modelių panašumo įvertinimas duomenų aibės atžvilgiu	44
4. Rezultatai	46
4.1. Išvados.....	46
Literatūra	48
Priedai	52

Santrumpos

- BFS Pločio pirmoji paieška (angl. *Breadth first search*). 15
- BP Atgalinė sklaida (angl. *Back Propagation*). 14
- CAD Kompiuterio pagalba sukurtas dizainas (angl. *Computer Aided Design*). 11
- CAN Kūrybiniai priešpriešiniai tinklai (angl. *Creative Adversarial Networks*). 6
- CD Chamfer atstumas (angl. *Chamfer distance*). 36, 37
- CDBN Konvoliuciniai giliojo tikėjimo tinklai (angl. *Convolutional Deep Belief Networks*). 11
- CNN Konvoliuciniai giliojo tikėjimo tinklai (angl. *Convolutional Deep Belief Networks*). 9, 10, 12–16
- DL Gilusis mokymasis (angl. *Deep learning*). 6, 8–10, 13, 17, 20, 22, 23, 25, 45, 46
- FC Pilnai sujungtas (angl. *Fully connected*). 11
- FID *Fréchet* pradžios atstumas (angl. *Fréchet inception distance*). 20, 21, 23, 46
- GAN Generatyviniai besivaržantys tinklai (angl. *Generative Adversarial Networks*). 6, 14, 17–23
- HD Hausdorff atstumas (angl. *Hausdorff distance*). 29, 30, 34–36
- IoU Sankirtos virš sąjungos koeficientas (angl. *Intersection over Union*). 28
- IS *Inception* įvertis (angl. *Inception score*). 20, 21, 23, 46
- KDE Branduolio tankio įvertis (angl. *Kernel Density Estimation*). 20, 46
- LIDAR Šviesos aptikimas ir nuotolio nustatymas (angl. *Light detection and ranging*). 12, 13

MVCNN	Daugiavaizdžiai konvoliuciniai neuroniniai tinklai (angl. <i>Multi-view Convolution Neural Networks</i>). 10
RGB-D	Raudonos, žalios ir mėlynos spalvų gylis (angl. <i>Red Green Blue-Depth</i>). 8, 12
RNN	Rekurentiniai neuroniniai tinklai (angl. <i>Recurrent Neural Networks</i>). 11
RPN	Srities pasiūlymo tinklai (angl. <i>Region proposal networks</i>). 12
STN	Erdviniai transformavimo tinklai (angl. <i>Spatial Transformer Networks</i>). 11
TSDF	Sutrumpinto ženklinimo atstumo funkcija (angl. <i>Truncated Signed Distance Function</i>). 12

1. Įvadas

Dirbtinio intelekto ir giliojo mokymosi (angl. *Deep learning*, DL) pritaikymas kasdienybėje įgauna pagreitį ir sulaukia vis daugiau susidomėjimo. Dar 2014 metais Goodfellow ir kt. [GPM⁺14] sukurti generatyviniai besivaržantys tinklai (angl. *Generative Adversarial Networks*, GAN) stebino plačiomis paveikslėlių generavimo galimybėmis, kiek vėliau Elgammal ir kt. [ELE⁺17] praplėsti kūrybiniais priešpriešiniais tinklais (angl. *Creative Adversarial Networks*, CAN), gebančiais išmokti paveikslėlių stilių, tapo vienais sėkmingiausių vaizdų sintezės modelių per pastaruosius kelerius metus. Toks pasisekimas pritaikant neuroninius tinklus įvairioms generavimo užduotims atlikti sulaukė daugiau dėmesio iš kompiuterinės grafikos bendruomenės. Pažanga 3D sensorinės technologijos srityje ir lengvesnis bei pigesnis jutiklių įrangos prieinamumas (pvz. Microsoft Kinect) supaprastino 3D duomenų gavimą ir atvėrė galimybes neuroninių tinklų trimatėje erdvėje realizacijai. Lengvesnis 3D duomenų prieinamumas paskatino mokslininkų bendruomenę gilintis į trimačių objektų klasifikacijos, atpažinimo ar formų paieškos uždavinius bei galimybę generuoti modelius meniniais tikslais besiplečiančioje žaidimų srityje.

Atliktų tyrimų kiekis vaizdų generavimui dvimatėje erdvėje gerokai pranoksta atliktų tyrimų kiekį neuroninių tinklų darbui su trimačiais duomenimis. Kadangi trimatį modelį nusakančių ypatybių išgavimas ir saugojimas atitinkamu formatu yra kertinis žingsnis norint atlikti minėtas užduotis naudojantis neuroniniais tinklais, sudėtingas tokių duomenų apdorojimas, kompiuterinių resursų ribotumas ir įvertinimo kriterijų trūkumas tampa nemenku iššūkiu šių duomenų panaudojimui bei įvertinimui.

1.1. Tiriamoji problema

Kompiuterio grafikai tobulėjant, auga tikslesnės virtualios realybės poreikis. Kompiuterinės grafikos metodai vis dažniau panaudojami įvairiausių programėlių kūrimui, ar kasdienio gyvenimo palengvinimui, o kompiuterio rega ir objektų atpažinimas šiuo metu yra plačiai nagrinėjama tema. Tačiau trimačių modelių kokybės įvertinimas yra neatsiejama grafinio modeliavimo dalis. Esant tokiai gausai įrankių ir trimačių modelių formatų išskyla klausimas: kaip įvertinti ir užtikrinti 3D modelių kokybę, realumą ir tinkamumą tolimesniam šių modelių panaudojimui. Išmatuoti ir pagrįsti trimačių modelių kokybę ar panašumą tiesiogiai apžiūrint 3D modelius nėra taip paprasta. Pirminėse 3D modelio generavimo stadijose, trimačių modelių išvaizda ir forma kiekvienam vertintojui gali sukelti skirtingų asociacijų. Kokybės nustatymo procesas dažnu atveju nėra apibrėžtas ir keičiantis vertintojui, keičiasi vertinimo kriterijai bei galutinis kokybės įvertinimo rezultatas. Norint iš tikrųjų nustatyti 3D modelių kokybę, trimačių modelių vertinimą svarbu traktuoti kaip pastovų procesą ir nustatyti nekintamus metodus bei kokybės kriterijus, kurių pagalba visiems tiriamiems trimačiams modeliams būtų sukurtos vienodos vertinimo sąlygos ir rezultatai nepriklausytų nuo vertinančio žmogaus šališkumo, patirties ar kitų aspektų. Nustatant vertinimo kriterijus svarbu atsižvelgti ir į daugelį trimatį modelį apibūdinančių veiksnių kaip: dydis, pozicija, posūkio kampas, paviršius ir t.t. Tik nuodugniai ištyrus 3D modelį, galima pasakyti, ar tai aukštos kokybės trimatis modelis, ar šį modelį dar reikia tobulinti. Trimačių modelių kokybės užtikrinimo ir

įvertinimo problema išryškėja trimačių modelių generavimo uždavinyje, kadangi neuroninių tinklų išvestį gana sudėtinga nešališkai įvertinti ir palyginti. Generatyviniai besivaržantys tinklai dažnu atveju neturi aiškių gairių ir patarimų kaip kokybiškai įvertinti šių tinklų išvestį. Toks trūkumas apsunkina vertinimą ir reikalauja remtis vizualiu kiekvieno trimačio modelio patikrinimu. Tačiau toks vertinimo metodas reikalauja pastangų, o duomenų, kuriuos reikia peržiūrėti kiekis, gali viršyti fizinės žmogaus galimybes. Tad išlieka klausimas, kaip tinkamai trimačius modelius palyginti ir kaip pagrįsti 3D-GAN tinklo išvesties kokybės gerėjimą bei realumą.

1.2. Tyrimo tikslas

Šio darbo tikslas – ištirti ir nustatyti kriterijus generatyviniais besivaržančiais tinklais sugeneruotų 3D modelių kokybei palyginti.

1.3. Tyrimo uždaviniai

Tyrimo tikslui pasiekti užsibrėžti uždaviniai:

- Apžvelgti esamus giliojo mokymosi iš 3D duomenų metodus bei dažniausiai naudojamą trimačių modelių reprezentacijos formas;
- Išanalizuoti neuroninių tinklų architektūras, skirtas generavimo ir klasifikavimo uždavimams atlikti trimatėje erdvėje;
- Apžvelgti esamus kokybės įvertinimo metodus taikomus 3D modelių generavimo uždavinio rezultato kokybės įvertinimui;
- Atlikti eksperimentinius tyrimus ir ištirti generatyvinių besivaržančių tinklų sukurtų trimačių modelių kokybę bei realumą pagal pasiūlytas metodikas.

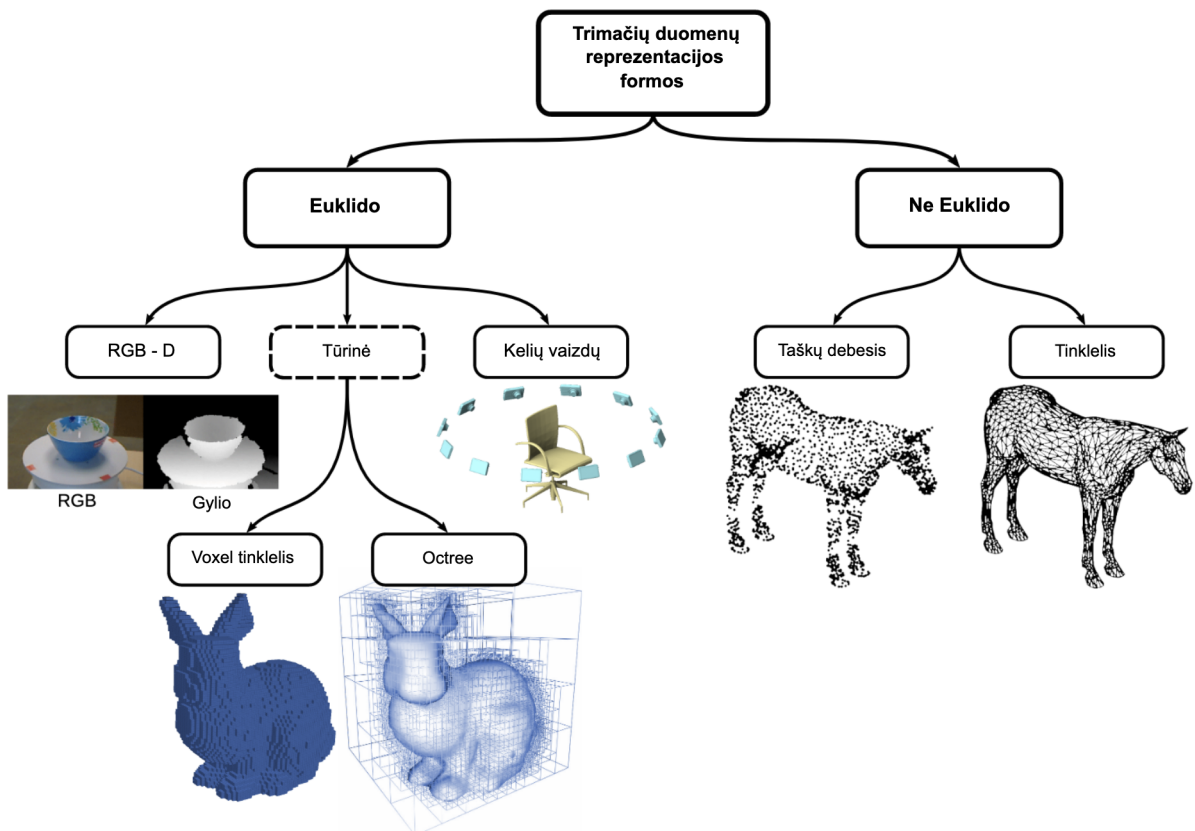
Tyrimo hipotezė: Kaip pasirinkti panašumo įvertinimo metodai atskleidžia 3D modelių tikslumą ir tikroviškumą?

2. Literatūros apžvalga

Šiame skyriuje aptariami metodai, skirti darbui su trimačiais modeliais juos klasifikuojant bei generuojant neuroninių tinklų pagalba. Taip pat apžvelgiamos dažniausiai naudojamos trimačių modelių reprezentacijos formos ir generavimo uždavinio rezultatų įvertinimo būdai įvardinant jų trūkumus ir praplėtimo galimybes.

2.1. Trimačių duomenų reprezentacijos formos

Trimačių modelių išsaugojimas skirtingose formose ir keitimas iš vieno formato į kitą gali ženkliai sumažinti trimatės formos kokybę bei pakeisti formos atvaizdą. Toks pokytis padaro nemenką įtaką DL kokybei. Achmed ir kt. [ICN⁺17] apžvelgia pastaraisiais metais dažniausiai pritaikytas trimačių objektų informacijos saugojimo formas, naudojamas giliojo mokymosi uždaviniams atlikti. Autoriai formas suskirsto į Euklido ir ne Euklido reprezentacijos kategorijas bei analizuoja kiekvienos formos privalumus, trūkumus ir praplėtimo galimybes. Žemiau esančiame paveikslėlyje (1 pav.) pateikiama supaprastinta Achmed ir kt. trimačių duomenų saugojimo formų apžvalga.



1 pav. Adaptuota įvairių 3D duomenų reprezentacijos formų apžvalga pateikiama Achmed ir kt. [ICN⁺17] originaliame darbe.

Suskirstymas į kategorijas apibendrina trimačius modelius pagal jų struktūrinius panašumus ir geometrines savybes. Trimačių duomenų formos, patenkančios į Euklido kategoriją pasižymi tolygumu ir nuoseklumu. Keli šios kategorijos pavyzdžiai: tūriniai vokselių tinkleliai, kelių vaizdų rinkiniai, RGB-D vaizdai. Tuo tarpu ne Euklido kategorijoje duomenys saugomi nenuoseklyje,

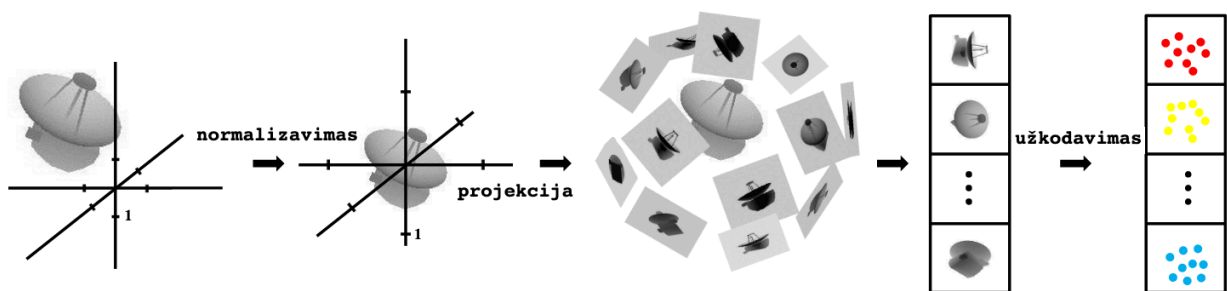
netvarkingoje struktūroje. Tinklelis (angl. *Mesh*) ir taškiniai debesys (angl. *Point cloud*) yra populiariūs šios kategorijos pavyzdžiai. Darbas su pastarosios kategorijos formomis giliojo mokymosi kontekste yra kiek sudėtingesnis.

2.2. Giliojo mokymosi iš trimačių duomenų metodų apžvalga

Pastaruosiu metu, gilusis mokymasis iš 2D vaizdų sparčiai tobulėja kompiuterio regėjimo srityje. Tačiau išplėsti DL modelius pritaikant 3D duomenis iki šiol išlieka nemenku iššūkiu. Sudėtinga 3D duomenų prigimtis ir struktūriniai skirtumai priverčia ieškoti kūrybingų informacijos apdorojimo sprendimų. Šiame poskyryje nagrinėjami giliojo mokymosi metodai: mokymasis iš kelių vaizdų, mokymasis tiesiogiai iš taškų rinkinio, tūrinės informacijos panaudojimas bei tūrinės informacijos suspaudimas. Visi šie metodai naudojami trimačių formų ypatybėms išgauti.

2.2.1. Gilusis mokymasis iš kelių vaizdų

Giliojo mokymosi iš kelių vaizdų metodas analizuoja turimo objekto ypatybes remiantis 2D objekto paveikslėliais iš skirtingų kampų, tai yra viena iš gudrybių panaudotų 3D objektų analizėje. Šią objekto analizę galima interpretuoti kaip skirtingą kameros poziciją objekto atžvilgiu. Tokiu būdu siekiama išskirti trimačio objekto savybes apjungiant išmoktą informaciją iš kiekvieno paveikslėlio į vieną kompaktišką trimačio objekto reprezentaciją. Sukurtas paveikslėlių rinkinys vėliau gali būti panaudojamas giliojo mokymosi kontekste, atskirai apdorojant vaizdus konvoliuciniais neuroniniais tinklais (angl. *Convolutional Neural Networks*, CNN). Šį metodą pritaikė Zhu ir kt. [ZWB⁺16] 2014 metais trimatį modelį projektuodami į dvimatę erdvę.

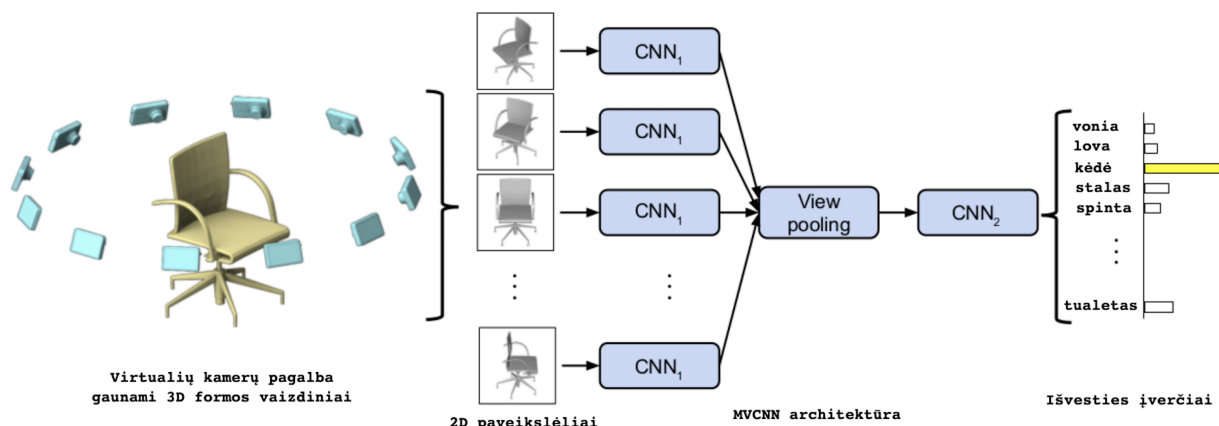


2 pav. Adaptuota 3D formos vaizdavimo srauto diagrama naudojant automatinę kodavimo sistemą pateikta Zhu ir kt. [ZWB⁺16] originaliaame darbe.

Pirma atliekamas kiekvieno trimačio modelio normalizavimas norint pašalinti skirtumus tokius kaip mastelis bei pozicija. Toliau kiekviena forma yra reprezentuojama gylio (angl. *depth*) vaizdų rinkiniu, kuris panaudojamas apmokyti automatinę užkodavimą (angl. *encoder*). Paveikslėlyje (2 pav.) matoma projekcijos srauto diagrama, kurioje spalvoti taškai atvaizduoja iš gylio vaizdų gautas 3D modelio formos savybes.

Su ir kt. [SMK⁺15] darbe taip pat buvo panaudotas 3D modelių vaizdų rinkinys. Pristatyta daugiavaizdžių konvoliucinių neuroninių tinklų (angl. *Multi-View Convolutional Neural Networks*,

MVCNN) architektūra, neuroninių tinklų modelis išmoksta apjungti ir klasifikuoti bet koki trimačio objekto vaizdų skaičių į kompaktišką trimatės formos ypatybių žemėlapij.



3 pav. MVCNN trimačio objekto klasifikavimo ir paieškos proceso žingsniai pateikiami Su ir kt. [SMK⁺15] originaliame darbe.

CNN apmokomas atpažinti formą iš formos vaizdo sugeneruoto atvaizdavimo varikliu (angl. *rendering*) ir pademonstruoja, kad neuroninis tinklas geba atpažinti 3D formą net iš vieno tokio sugeneruoto paveikslėlio. Paveikslėliai generuojami iš 12 skirtingų kampų (3 pav.). Sugeneruotas 3D modelio paveikslėlių rinkinys toliau perduodamas CNN₁ tinklų rinkiniui, kur kiekvienam tinklui tenka vienas objekto vaizdas. Objekto ypatybės apibendrinamos sujungimo sluoksnyje (angl. *pooling layer*) ir CNN₂ pagalba išgaunamas kompaktiškas formos suvokimas. CNN₂ tinklas pirma buvo apmokytas naudojant ImageNet1K duomenų rinkinį ir vėliau suderintas naudojant ModelNet40 [WSK⁺15] duomenų rinkinį. Pagal autorius [ICN⁺17], gauti formos klasifikavimo ir paieškos rezultatai parodė, kad MVCNN pranoko visus kitus tuo metu išbandytus metodus.

Mokymasis iš kelių vaizdų turi savų trūkumų, neretai prireikia daugybės paveikslėlių norint pakankamai apibūdinti turimą 3D formą ir apmokyti modelį. MVCNN naudojo 12 paveikslėlių kiekvienai 3D formai išreikšti, turint 10 000 tūkstančių ir daugiau trimatinių formų apmokymo rinkinyje toks DL modelio apmokymas sukelia skaičiavimo bei duomenų saugojimo resursų išlaidas, o taikomi papildomi duomenų suspaudimo būdai gali lemti DL modelio persimokymą (angl. *overfitting*).

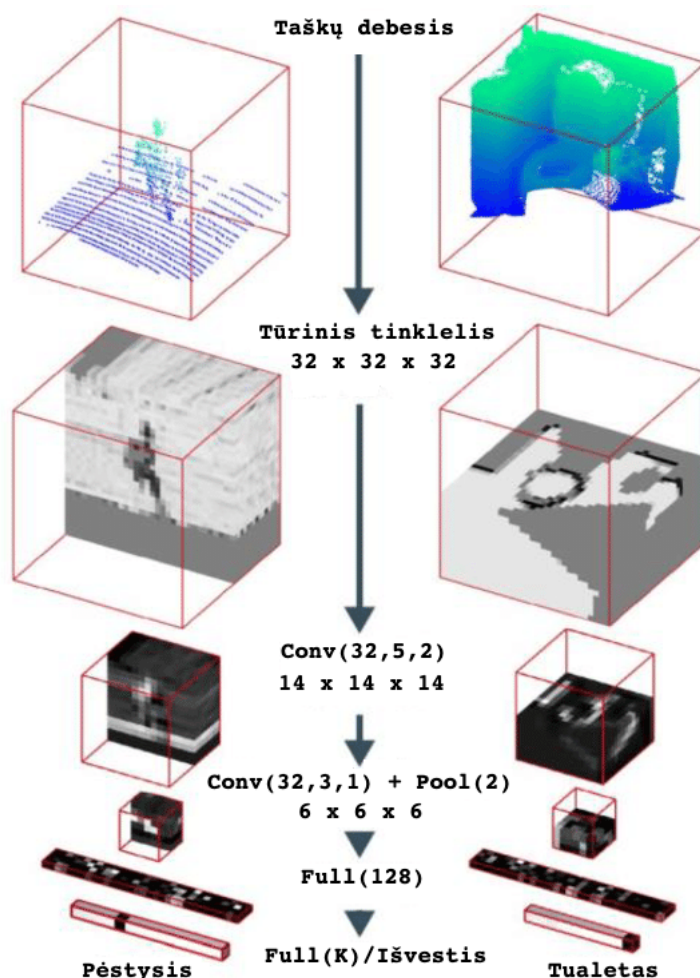
2.2.2. Gilusis mokymasis iš taškų rinkinio

Taškų debesis (angl. *point cloud*) suteikia išraiškingą ir kompaktišką 3D modelio vaizdą, tad iš tokio taškų rinkinio galėtų būti paprasta apmokyti neuroninio tinklo modelį. Tačiau apdoroti taškų rinkinius yra ganėtinai sudėtinga operacija. Svarstant, taškų rinkinio panaudojimą DL modelių mokymui, reikia įvertinti taškų rinkinio netvarkingą ir nenuoseklią struktūrą bei taškų tarpusavio sąryšio nebuvimą. Qi ir kt. [QSM⁺17] pademonstravo, kad neuroninio tinklo architektūra pavadinimu PointNet gali būti apmokoma neapdorotais taškų rinkiniais jų neperteikiant kitomis 3D reprezentacijos formomis. Nesudėtinga PointNet struktūra ir paprasta šio modelio įvestis yra didelis privalumas, kadangi naudojamas (x, y, z) taškas. Esminė tokios architektūros idėja – mak-

simalaus sujungimo sluoksnio (angl. *max-pooling layer*) panaudojimas. Tinklas išmoksta optimizavimo funkciją, kurios pagalba išskirtinių ir informatyvių taškų pasirinkimui kartu užkoduojama asociacija – taško svarbumo įvertis. PointNet esminiai komponentai: erdviniai transformavimo tinklai (angl. *Spatial Transformer Networks*, STN), rekurentiniai neuroniniai tinklai (angl. *Recurrent Neural Networks*, RNN) ir simetrinė funkcija, skirta informacijos apibendrinimui iš kiekvieno debesies taško. STN apdoroja visus duomenis į vieną kanoninę formą išgaunant esmines trimačio objekto ypatybes taip suformuojamas suvokimas apie bendrą 3D objekto skeletą. Vienas pagrindinių PointNet tinklo trūkumų – sunkiai apdorojamos itin smulkios 3D modelių detalės, ko pasekoje dauguma smulkių modelio savybių yra prarandamos ir formos kokybė bei detalumas suprastėja. Tačiau toks RNN tinklo pritaikymas 3D formos ypatybių išgavimui gali būti pavadintas novatorišku autorių sprendimu.

2.2.3. Gilusis mokymasis iš vokselių tinklelio

Naują požiūrį į 3D modelio atvaizdavimą, naudojant visą trimačio objekto struktūrą, pateikė Wu ir kt. [WSK⁺15] 2015 metais kartu su pristatytu modeliu 3D ShapeNets. Modelio architektūra paremta konvoliuciniais giliojo tikėjimo tinklais (angl. *Convolutional Deep Belief Networks*, CDBN), kurie yra efektyvūs apdorojant vaizdus ir atpažįstant juose esančius trimačius objektus. 3D ShapeNets tiesiogiai panaudoja erdvinius modelius, kurių kiekvienas įvedamas vokselių tinkleliu. Vokselis reprezentuoja patį mažiausią galimą diskretų trimatės erdvės elementą, savo panašumu primena pikselį dvimatėje erdvėje. Norint sumažinti parametų reikalingų pilnai sujungtam CDBN tinklui kiekį, pritaikytas konvoliucinis 3D filtras. CDBN sudaro penki sluoksniai, iš kurių trys konvoliuciniai sluoksniai, pilnai sujungtas (angl. *fully connected*, FC) sluoksnis ir išvestis. 3D ShapeNets buvo išbandytas trimačių formų klasifikavimo užduočiai atlikti, taip pat autoriai išleido ModelNet [VGV⁺09] didelio masto CAD duomenų rinkinį iš 662 unikalių klasių.



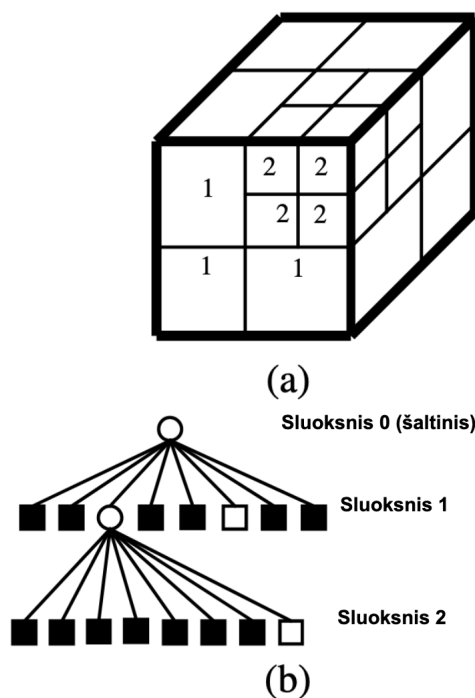
4 pav. Adaptuota VoxNet architektūra pateikta [MS15] autorių originaliame darbe. Kairėje pusėje LIDAR taškų debesis, dešinėje – RGB-D taškų debesis.

Song ir Xiao [SX16] 2016 metais sutrumpinto ženklinimo atstumo funkcijos (angl. *truncated signed distance function*, TSDF) pagalba panaudojo neapdorotą 3D scenos informaciją konvertuodami kiekvieną gylio vaizdą į trimatį vokselių tinklą ir taip įgyvendindami trimačių objektų paiešką 3D scenoje. Song ir Xiao sukurtas 3D ConvNets modelis – tai pilnai sujungtas CNN, dar vadinamas 3D RPN (angl. *Region proposal networks*), kurio paskirtis generuoti skirtingų matų 3D objektų ribojančio kubo kontūrus iš vokselių tinklo, tokiu būdu norėta suvaldyti trimačių objektų dydžių skirtumus. Kiekvienam aptiktam 3D objektui atvaizduojamas jį ribojantis kubo kontūras.

Vokselių tinkliai daugeliu aspektų panašūs į 2D paveikslėlius ir dėl vokselių tinklo nuoseklumo, lengviau duomenis panaudoti konvoliucinėms operacijoms atlikti. Būtent šių operacijų dėka daugėja giliojo mokymosi algoritimų naudojančių trimačių objektų geometriją. Tačiau vokselių panaudojimas taip pat pasižymi didelėmis skaičiavimo sąnaudomis augančiomis didėjant vokselių tinklo dydžiui. Toks sąnaudų augimas paaiškinamas tinklelyje užfiksuojamais tiek užimtais, tiek neužimtais vokseliais. Ko pasekoje autoriai tiek 3D ShapeNets [WSK⁺15] tiek VoxNet [MS15] modeliuose pritaikė $30 \times 30 \times 30$ ir $32 \times 32 \times 32$ binarinių vokselių tinklo dydžius.

2.2.4. Gilusis mokymasis iš *octree* medžio

Tipiškai vokselis turi didelių tuščių sričių, kurių pašalinimas iš esmės sumažintų atminties ir skaičiavimų kiekį. Siekiant išspręsti vokselių tinklelių neefektyvumą ir dideles resursų sąnaudas, DL uždaviniams spręsti pasitelkiama *octree* medžio duomenų struktūra, kurioje kiekvienas vidinis mazgas turi tiksliai aštuonis vaikus. *Octree* dažniausiai naudojama trimatei erdvei padalinti, rekursyviai padalijant ją į aštuonis oktantus (angl. *octants*). Kaip pavyzdį panagrinėjame Wong ir kt. [WSS06] 2006 metais pasiūlytą kompaktiškesnę 3D reprezentaciją pagrįstą LIDAR taškų debesies suskirstymo į 3D plokštumų grupes algoritmą. Algoritmas kompaktiškumą pasiekia padalijimo ir sujungimo metodu modeliuojant hierarchinę į medį panašią struktūrą, kurioje kiekviena ląstelė gali būti rekursyviai išskaidyta į aštuonis oktantus.



5 pav. *Octree* struktūra (a) padalinti segmentai (b) *octree* medžio reprezentacija. Adaptuotas pavyzdys pateikiamas originaliaame Wong ir kt. [WSS06] darbe.

Padalijimo procesas pradedamas turint pilną duomenų rinkinį – LIDAR taškų debesį, dar vadinamą šaltiniu. Jeigu duomenų rinkinio erdvė viršija atstumo ir ploto slenkstį, tuomet tokia erdvė padalinama į 8 lygias poerdves. Skilimas sugeneruoja 8 mazgus (5 pav. (b)), vaizduojančius padalintas erdves. *Octree* pritaikymas išpopuliarėjo trimačių objektų generavimo uždaviniuose. Wang ir kt. [WLG⁺17] 2017 metais pristatė O-CNN modelį paremtą *octree* duomenų struktūra. Pirmiausiai modelis orientuotą 3D formą (pvz., orientuotas trikampis tinklelis (angl. *mesh*) arba normalizuotas taškinis debesis) sukonstruoja į *octree* duomenų struktūrą ir pritaiko CNN konvoliucijos operacijas. O-CNN gali atlikti: formos klasifikavimą, rekonstrukciją, paiešką ir suskirstymą. Neuroninių tinklų modelis pademonstruoja *octree* duomenų išraiškos potencialą suspausti 3D duomenis į patogią medžio struktūrą, tačiau hierarchinė duomenų struktūra padidina jų sudėtingumą ir apsunkina lygiagrečias operacijas, tokias kaip 3D konvoliucijos. Autoriai neeksperimentuoja su pačia CNN

neuroninių tinklų architektūra, tad tinklo architektūros patobulinimas galėtų būti vienas iš būdų didesniajam O-CNN efektyvumui išgauti.

2.3. Neuroninių tinklų architektūros

Trimačių objektų uždavinys neretai pareikalauja pasitelkti kelių rūšių neuroninių tinklų architektūras. Vieni dažniausiai šiuo metu plėtojamų – klasifikuojantys ir generuojantys neuroninių tinklų modeliai. Vykdydami procesą, vadinamą mokymu, neuroniniai tinklai gali išmokti atpažinti įvesties duomenų struktūras ir ryšius, remiantis išmokta informacija, daryti prognozes ar priskirti įvestį atitinkamai klasei. Vienas iš pagrindinių neuroninių tinklų pranašumų yra jų gebėjimas apibendrinti informaciją iš mokymo duomenų, o tai leidžia tiksliai prognozuoti naujus, nematytus duomenis. Šiame skyriuje nagrinėjamos dvi neuroninių tinklų architektūros – generatyviniai besivaržantys tinklai (GAN) ir konvoliuciniai neuroniniai tinklai (CNN) bei jų pritaikymas trimačiame modeliavime.

2.3.1. Konvoliuciniai neuroniniai tinklai

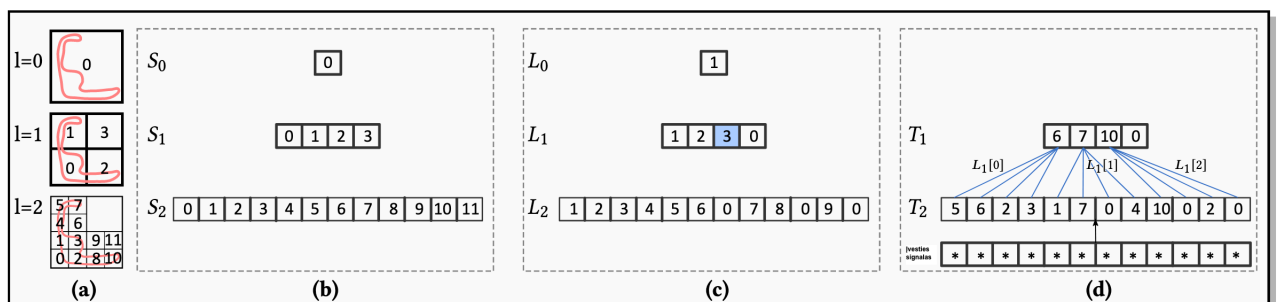
Šiuo metu konvoliucinių neuroninių tinklų (angl. *Convolutional neural networks*, CNN) architektūra yra viena dažniausiai taikomų klasifikavimo užduotims atlikti teksto ir vaizdų sintezės srityse, keletas tokių užduočių pavyzdžių: elektroninių laiškų skirstymas į šlamšto ir ne šlamšto kategorijas, ranka rašytų simbolių atpažinimas, objekto aptikimas ir t.t. CNN įvestis keliauja per daugybę konvoliucinių sluoksnių, kurie iš įvesties išskiria vietines ypatybes. Tada šios savybės yra sujungiamos ir transformuojamos visiškai sujungtais sluoksniais (angl. *fully connected layers*), kurie sukuria klasifikavimo išvestį. Klasifikavimo proceso metu CNN klasifikuojamam objektui priskiria etiketę pagal savybes, kuriomis išmoko tą objektą atpažinti. Norėdami mokytis iš duomenų, CNN naudoja procesą, vadinamą atgaline sklaida (angl. *back propagation*), tokio proceso metu tinklas koreguoja savo neuronų jungčių svorį, kad sumažintų skirtumą tarp prognozuojamų etikečių ir tikrųjų etikečių apmokymo rinkinyje. Per šį procesą CNN išmoksta apibendrinti savo žinias, kad galėtų tiksliai įvertinti naujus, dar nematytus duomenis. Svarbu, kad mokymo duomenų rinkinys būtų suderintas (turėtų pakankamai kiekvienos klasės etikečių) ir pakankamai reprezentuotą problemą. Kadangi klasių etiketės dažnai yra žodinės reikšmės, sėkmingam jų susiejimui su skaitinėmis reikšmėmis panaudojamas etikečių kodavimas – kiekvienai klasei etiketei priskiriamas unikalus sveikasis skaičius pvz. 0 arba 1. CNN yra vieni iš plačiausiai ištirtų neuroninių tinklų tipų. LeCun ir kt. [LBD⁺90] 1990 metais paskelbė LeNet-5 septynių sluoksnių neuroninį tinklą skirtą ranka rašytų skaitmenų atpažinimui. Neuroninio tinklo apmokymui panaudotas atgalinės sklaidos (angl. *back propagation*, BP) algoritmas ir norint aptikti vietines paveikslėlyje esančias ypatybes neapkraunant tinklo papildomais parametrais pritaikyta ryšių tarp mazgų apribojimo schema. LeNet-5 tiesiogiai priima vaizdus ir pademonstruoja neuroninio tinklo gebėjimą veiksmingai atpažinti skaitmenis tiesiai iš neapdorotų paveikslėlio pikselių.

Apmokymo duomenų ir kompiuterinės galios trūkumas yra keletas pagrindinių kliūčių didžiojo kiekio vaizdo įrašų klasifikavimui. Tačiau konvoliuciniai neuroniniai tinklai ženkliai patobulėjo

ir jų pritaikymo galimybės prasiplėtė nuo tuometinio laikotarpio. Krizhevsky ir kt. [KSH17] 2012 metais pristatė klasikinę CNN architektūrą pavadinimu AlexNet dalyvaujant ImageNet didelio masto vizualinio atpažinimo iššūkyje. AlexNet neuroninis tinklas sava architektūra primena LeNet-5 ir geba klasifikuoti didelę apimtį aukštos rezoliucijos paveikslėlių pritaikant 8 sluoksnių konvoliucinį neuroninį tinklą. Paskelbta daugelis kitų darbų siekiant patobulinti konvoliucinių neuroninių tinklų veikimą: ZFNet [ZF14], VGGNet [SZ14], GoogleNet [SLJ⁺15] ir ResNet [HZR⁺16]. Iš architektūrinės CNN evoliucijos galima matyti, kad CNN tapo vis gilesnis, pavyzdžiui ResNet yra maždaug 20 kartų gilesnis nei AlexNet ir 8 kartus gilesnis nei VGGNet. Pagal autorius [GWK⁺18], didinant neuroninio tinklo gylį, tiksliau suderinama tikslo funkcija ir gaunami geresni rezultatai.

O-CNN architektūra

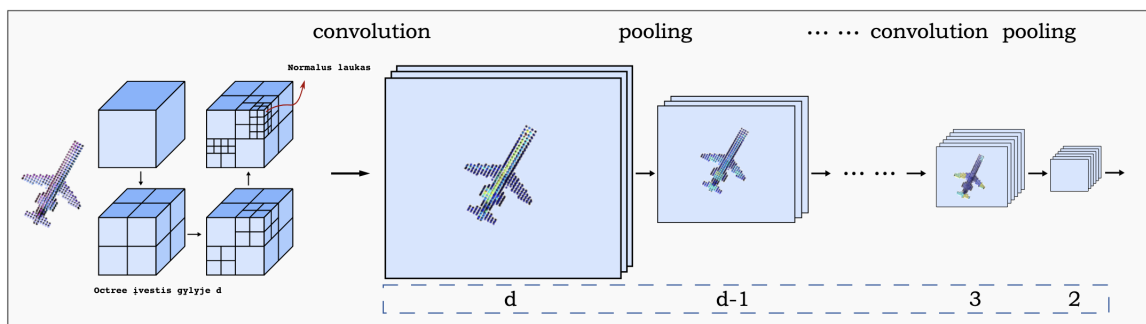
Wu ir kt. [WLG⁺17] O-CNN modelio architektūros idėja paremta *octree* trimačių formų operacijomis. O-CNN įvesčiai naudoja normaliuosius 3D modelio vektorius, sudarytus iš *octree* medžio mazgų, taip išgaunant reikšmingiausias erdvinio modelio ypatybes. Vienas techninių O-CNN iššūkių – hierarchinių medžio struktūros trūkumų pašalinimas skaičiavimų lygiagretumui užtikrinti, tokio iššūkio suvaldymas leistų atlikti skaičiavimus efektyviai.



6 pav. 2D O-CNN įvesties gavimo iliustracija pateikta Wu ir kt. [WLG⁺17] originaliaame darbe.

Paveikslėlyje (6 pav.) autoriai pademonstruoja, kaip algoritmas skirtas *octree* medžiui išgauti, veikia 2D erdvėje: (a) atsižvelgiant į įvesties dvimatę formą (pažymėtą raudonai), sudaromas 2D gylis medis (angl. *quadtree*), visi neužimti regionai yra laikomi tuščiais mazgais, o skaitmenys esantys netuščią regionų viduje yra tų regionų maišymo raktai (angl. *shuffle keys*); (b) nurodomi vektoriai S_l , kurie $l = 0, \dots, 2$ saugo maišymo raktus gilyje l ; (c) nurodomi vektoriai L_l , kurie $l = 0, \dots, 2$ saugo 2D mazgų (angl. *quad nodes*) etiketes gilyje l , tušties mazgams nustatoma etiketė 0, tuo tarpu netuščiams mazgams priskiriama p -toji etiketė žyminti mazgo eilę esamame gilyje, toks etiketės buferis skirtas susieti tėvinius ir vaikų mazgus bei jų tvarką; (d) nurodo CNN konvoliucijos rezultatus gautus iš įvesties signalo, kurie saugomi ypatybių žemėlapiu T_2 , kuomet T_2 išskaidomas konvoliucijos operacijos, pavyzdžiui sujungimo sluoksnio atveju, išskaidyti rezultatai priskiriami pirmam, antram ir trečiam T_1 įrašui. Etikečių vektorius L_1 pirmajame gilyje naudojamas rasti atitiktį tarp mazgų abiejuose gilyuose. Taigi norint sukonstruoti *octree* iš gauto trimačio modelio, pirmiausiai trimačio modelio mastelis padidinamas ir ašimi išlygiuojamas ribojančiu kubu, tuomet rekursyviai visas kubas padalinamas BFS algoritmu. Kiekviename žingsnyje visi netušti mazgai

esantys gilyje l padalinami į aštuonis mazgus, kur sekantis gylis yra $l + 1$. Šis procesas kartojamas iki pasiekiamas iš anksto nustatytas *octree* gylis d . Gauti *octree* mazgai surūšiuojami pagal jų maišymo raktus didėjimo tvarka ir išsaugomi vektorius S_l gilyje l , vektorius vėliau naudojamas 3D octantų kaimynystei sukurti konvoliucijos metu. Taip pat panašiu principu kaip ir minėtame pavyzdyje netuštiems mazgams priskiriamos p etiketės.



7 pav. Adaptuota O-CNN architektūra pateikta Wu ir kt. [WLG⁺17] originaliame darbe.

O-CNN architektūra (7 pav.) gana nesiskiria nuo originalios CNN architektūros ir paremta pakartotiniu konvoliucijos ir sujungimo sluoksnio taikymu *octree* struktūrai nuo apačios į viršų. Aktyvavimui naudojama *ReLU* funkcija ir partijos normalizavimas (angl. *batch normalization*). O-CNN tikslumui palyginti esant skirtingai medžio struktūros dydžiui apmokomi šeši tinklai: O-CNN(3), O-CNN(4), O-CNN(5), O-CNN(6), O-CNN(7), O-CNN(8). Kiekvieno iš tinklų medžio lapų mazgų skaičius atitinkamai yra: 8^3 , 16^3 , 32^3 , 64^3 , 128^3 , 256^3 . O-CNN tinklo testavimo metu pasiekti rezultatai pateikiami autorių [WLG⁺17] darbe:

1 lentelė. O-CNN [WLG⁺17] palyginimas su VoxNet [MS15].

Modelis	Tikslumas (ModelNet40)
VoxNet (32^3) [MS15]	83,3 %
O-CNN(3)	87,1 %
O-CNN(4)	89,3 %
O-CNN(5)	90,4 %
O-CNN(6)	90,6 %
O-CNN(7)	90,1 %
O-CNN(8)	90,2 %

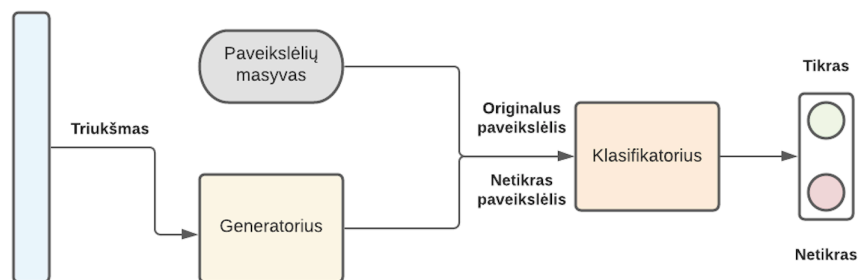
Lentelėje matomi tikslumai (1 lentelė) yra dalis ModelNet40 [WSK⁺15] duomenų rinkinio objektų klasifikavimo uždavinio rezultatų, pateikiamų autorių [WLG⁺17] originaliame darbe. Tikslumo stulpelyje nurodomas ModelNet40 [WSK⁺15] rinkinio objektų atpažinimo tikslumas. Tikslumo nustatymo metu kiekvienas rinkinio objektas buvo pasukamas 12 kartų išgaunant skirtingas objekto pozicijas.

Rezultatai pademonstruoja ženklų *octree* medžio ir CNN neuroninių tinklų pranašumą gebant išsaugoti kompaktišką 3D modelio informaciją. O-CNN palaiko įvairias CNN struktūras ir gali dirbti su įvairiomis 3D formomis. O-CNN modelio architektūra daug žadanti bei gali būti panaudota kaip pagrindas tolimesnei analizei. O-CNN modelis pademonstruoja, kad *octree* medžiu

suspausta 3D forma gali būti taikoma DL modelio apmokyme. Trimačių formų generavimas galėtų būti O-CNN neuroninio tinklo tolimesnis praplėtimas.

2.3.2. Generatyviniai besivaržantys tinklai

Generatyviniai besivaržantys tinklai (GAN) [GPM⁺14] tai viena įdomiausių dešimtmečio idėjų mašininio mokymosi srityje. Šiai nekontroliuojamo (angl. *unsupervised*) mokymosi modelių grupei galime priskirti architektūras, kurių tikslas išmokyti kurti naujoviškus duomenų rinkinio egzempliorius. GAN tinklo veikimas paremtas dviejų modelių konkuravimo principu (8 pav.), kur pirmasis modelis atlieka generatoriaus funkciją (angl. *generator*), generatorius iš triukšmo vektoriaus sugeneruoja netikrą paveikslėlį, o antrasis – klasifikatoriaus (angl. *discriminator*), kuris paveikslėlį įvertina. Generatorius remdamasis grįžtamoju ryšiu iš klasifikuojančio modelio, išmoksta kurti duomenų pavyzdžius kuo panašesnius į originalaus duomenų rinkinio egzempliorius. Tuo tarpu klasifikatoriaus įvertina tikimybę ar gautasis egzempliorius priklauso tikrajam duomenų rinkiniui, ar yra sugeneruotas, netikras pavyzdys. Mokymo procedūros tikslas – padidinti klasifikuojančio modelio tikimybę suklysti, ko pasekoje generatorius mokomas išgauti vis realesnius ir į mokymo duomenis panašesnius pavyzdžius. Svarbu paminėti, kad generatorius sudaro naujus pavyzdžius nekopijuojant jau esamų duomenų rinkinio egzempliorių, tad neretai GAN pasitarnauja meniniais tikslais, kuriant įvairiausių stilių paveikslėlius.



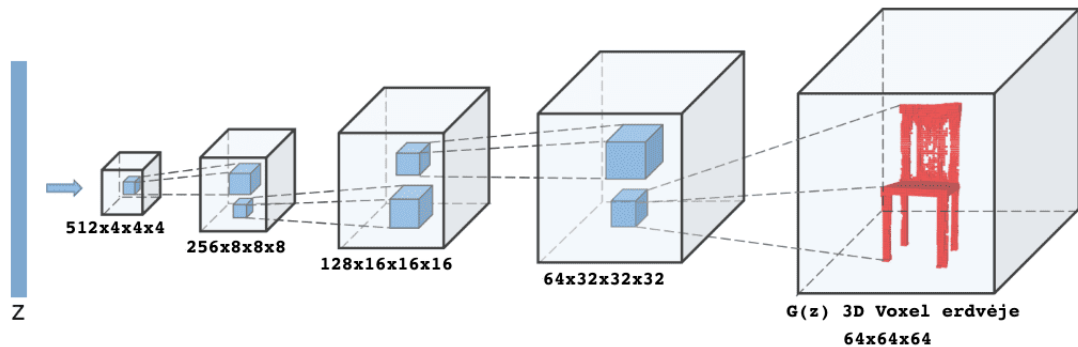
8 pav. GAN architektūros pagrindiniai komponentai.

Viena dažniausių GAN panaudojimo paskirčių – paveikslėlių kūryba, tačiau modelio architektūra ir veikimo būdas neriboja taisyklėmis, nusakančiomis, koks tas turinys ir duomenų pavyzdžiai turėtų būti. Toks GAN lankstumas paskatino tyrėjus eksperimentuoti ir tinklą pritaikyti teksto generavimui bei perkelti GAN kūrybą iš dvimatės į trimatę erdvę. Wu et al. [WZX⁺16] 2016 metais GAN pritaiko 3D vokselių tinklelių generavimui. Autorių sukurtas 3D-GAN tinklas generuoja 3D modelius pasinaudojant naujausia tuometine tūrinių konvoliucinių tinklų (Maturana ir Scherer [MS15]) ir generatyvinių besivaržančių tinklų pažanga.

3D-GAN architektūra

Vienas 3D-GAN [WZX⁺16] architektūros nuo tradicinio GAN [GPM⁺14] skirtumų – 3D įvesties panaudojimas generatoriaus ir klasifikatoriaus mokymui. 3D-GAN generatorius G atvaizduoja atsitiktinį 200 dimensijų vektorių z į $64 \times 64 \times 64$ kubą, reprezentuojantį $G(z)$ objektą 3D vokselių

tinklelyje (9 pav.). Klasifikatorius D grąžina patikimumo koeficientą $D(x)$, įvertindamas ar 3D įvestis x yra reali.



9 pav. Adaptuota 3D-GAN generavimo proceso eiga pateikta originaliame autorių [WZX⁺16] darbe.

Wu et al. paseka tradicinio Ian Goodfellow et al. [GPM⁺14] GAN pavyzdžiu nuostolių funkcijos (angl. *loss function*), dar žinomos kaip *min-max* nuostolių funkcija, pritaikymui trimatėje erdvėje:

$$L_{3D-GAN} = \log D(x) + \log(1 - D(G(z))) \quad (1)$$

Funkcijoje x yra realus objektas $64 \times 64 \times 64$ kube ir z yra atsitiktinai parinktas triukšmo vektorius iš $p(z)$. Kiekvienas z matmuo yra nepriklausomai ir vienodai paskirstytas dydis $[0,1]$ intervale. 3D-GAN generatorių sudaro $4 \times 4 \times 4$ dydžio dviejų žingsnių konvoliuciniai sluoksniai kartu su partijos normalizavimo ir *ReLU* sluoksniais tarp jų bei *sigmoid* sluoksniu pabaigoje. Klasifikatoriaus struktūra nuo generatoriaus skiriasi partijos normalizavimo funkcija, klasifikatoriui pritaikoma *Leaky ReLU* funkcija. Modelio vertinimas susideda iš kelių etapų, pirmiausiai pademonstruojami sugeneruoti trimačių formų pavyzdžiai, tuomet įvertinamos diskriminatoriaus išmoktos formų reprezentacijos. Tipiškas reprezentacijų vertinimo būdas remiasi modelio pasisekimu klasifikuojant ModelNet40 [WSK⁺15] ir ShapeNet [CFG⁺15] trimačių objektų rinkinius. 3D-GAN mokomas septyniomis pagrindinėmis ShapeNet [CFG⁺15] rinkinio klasėmis (kėdės, sofos, stalai, valtys, lėktuvai, šautuvai ir automobiliai) ir testuojamas taip pat naudojantis ModelNet40 [WSK⁺15] duomenų rinkiniu. Kadangi mokymo ir testavimo kategorijos nėra identiškios, 3D-GAN pademonstruoja veiksmingą ypatybių įsisavinimą. Pagal autorius, 3D-GAN klasifikatorius apmokytas nekontroliuojamu būdu, gali būti naudojamas kaip 3D formos ypatybių žemėlapis, turintis 83,0 % tikslumą formų klasifikavimo uždavinyje ir tikslumu gerokai pranoksta 3D ShapeNets [WSK⁺15] modelį. Žemiau pateikiama dalis autorių rezultatų (2 lentelė), rezultatuose matomas 3D-GAN palyginamas su 3D ShapeNets [WSK⁺15] ir VoxNet [MS15] klasifikuojant ModelNet40 [WSK⁺15] duomenų rinkinį:

2 lentelė. Supaprastinti ModelNet40 klasifikavimo uždavinio rezultatai [WZX⁺16].

Apmokymo metodas	Modelis	Tikslumas (ModelNet40)
Klasių etiketės	3D ShapeNets [WSK ⁺ 15]	77,3 %
Klasių etiketės	VoxNet [MS15]	83,0 %
Nekontroliuojamas	3D-GAN [WZX ⁺ 16]	83,3 %

Pagal Radford ir kt. [RMC16], nors GAN yra perspektyvus sprendimas, pažymima, kad tai gana nestabilus tinklas. Šiai problemai spręsti Li ir kt. [LZW⁺19] tinklą modifikuoja į 3D sąlyginį GAN, kuriame vektorius papildomai apdorojamas kartu su klasių etiketėmis. Tačiau tai ne vienintelė darbo su GAN problema. GAN išreikštų rezultatų įvertinimas yra dar vienas iššūkis su kuriuo susiduria tyrėjai, kadangi vien rezultatų įvertinimo mechaniškai nepakanka norint užtikrinti šių kokybę, o tokia problema persikelia ir į trimatę erdvę. Tad išlieka atviras klausimas, kaip užtikrinti sugeneruotų rezultatų kokybę ir įvairovę, o trimatės erdvės atveju ir formų realumą bei tikslumą.

2.4. GAN rezultatų kokybės ir realumo įvertinimas

GAN generatorius yra mokomas iteratyviai per daugelį mokymo epochų. Treniruočių metu yra įprasta naudoti esamą modelio būseną norint sugeneruoti daug sintetinių vaizdų ir išsaugoti esamą generatoriaus, naudojamo vaizdams generuoti, būseną ateičiai. Modeliai gali būti sistemingai išsaugomi visose mokymo epochose, pvz., kas vieną, penkias, dešimt ar daugiau mokymo epochų. Užuo mokomas tiesiogiai, generatoriaus modelis mokomas diskriminatoriaus grįžtamuju ryšiu, tad generatoriaus modeliui nėra taikoma objektyvi funkcija ar kitas objektyvus vertinimo matas. Nesant objektyviam modelio veikimo vertinimo matui, sunku nuspręsti, kada mokymo procesas turėtų sustoti ir kada galutinis modelis turėtų būti išsaugotas vėlesniam naudojimui. Generatoriaus būsenos išsaugojimas mokymo metu leidžia grįžti į ankstesnę generatoriaus būseną generatoriui persimokius, taip pat taip tai leidžia palyginti sugeneruojamus rezultatus mokymo procesui pasibaigus (angl. *post-hoc*). Tenka pastebėti, generatyvinių modelių kokybės ir realumo įvertinimas iškelia nemažai klausimų bei iššūkių ir neapsieina be papildomo vizualaus rezultatų patikrinimo.

Dažniausiai GAN modelio veikimas vertinamas pagal sukurtų sintetinių vaizdų kokybę. Vienas iš paprasčiausių būdų įvertinti GAN – vizualiai tikrinti sugeneruotus pavyzdžius iš skirtingų epochų nusprendžiant, kurie iš GAN sukurtų egzempliorių atrodo realūs ir yra tinkami tolesniam panaudojimui. Tačiau toks būdas turi savų trūkumų: vertinimas tampa subjektyviu ir apima vertinančio žmogaus šališkumą, o srities žinios neretais atvejais yra būtinos norint nuspręsti ar egzempliorius panašus į norimą rezultatą ir neturi esminių klaidų. Taip pat svarbu atsižvelgti ir į generavimo apimtį, kadangi duomenų dydis gali viršyti fizines žmogaus galimybes juos peržiūrėti. Kartu, aiškių gairių ir patarimų kaip kokybiškai peržiūrėti vaizdus trūkumas taip pat apsunkina vertinimą, kiekvienu atveju reikalavimai kinta ir priklauso nuo generuojamų duomenų srities, o visuotinai sutarto būdo, kaip įvertinti tam tikrą GAN generatoriaus egzempliorių nėra. Šiame poskyryje trumpai aptariami dažniausiai naudojami metodai GAN rezultatų kokybės ir realumo įvertinimui dvimatėje ir trimatėje erdvėse.

Branduolio tankio įvertis

Originalaus 2014 m. GAN, kurį pateikė Goodfellow ir kt. [GPM⁺14] sukurtų vaizdų kokybei įvertinti panaudotas branduolio tankio įvertis (angl. *Kernel density estimation*, KDE). KDE yra metodas, įvertinantis duomenų taškų rinkinio tikimybės tankio funkciją. Pagrindinė KDE idėja yra kiekviename duomenų taške taikyti branduolio funkciją (pvz. Gauso funkciją), o tada susumuoti rezultatus, kad būtų įvertintas tikimybės tankis bet kuriame erdvės taške. Įvertis pasižymi ir kintamais parametrais, pakeitus branduolio funkcijos pralaidumą ir amplitudę, pakeičiamas įvertinimo dydis ir forma. Branduolio funkcijos pralaidumas lemia apskaičiuoto tankio funkcijos sklandumą, o mažesnis pralaidumas lemia išsamesnius įvertinimus. Šių parametrų dėka, neuroninis tinklas gali pradėti treniruotis keičiant savo įvertinimus ir pateikti vis tikslesnius rezultatus. Įvertinimo procesui kartojantis, dažnių juostos pločio ir amplitudės įverčiai nuolat atnaujinami, siekiant padidinti įvertintos tikimybės tankio kreivės tikslumą. Vienas iš KDE pranašumų yra tai, kad šis įvertis gali pateikti kiekybinį sugeneruotų duomenų kokybės matą, o ne pasikliauti subjektyviu žmogaus vertinimu. Tačiau tai reikalauja daug skaičiavimo išteklių ir įvertis gali būti jautrus branduolio funkcijos ir pralaidumo pasirinkimui. Tad, norint gauti tikslius ir patikimus rezultatus, svarbu pasirinkti tinkamus šių parametrų nustatymus.

Inception ir *Fréchet Inception* atstumų įverčiai

Inception (IS) ir *Fréchet Inception* (FID) atstumai skirti įvertinti GAN sugeneruotų egzempliorių kokybę pagal iš anksto paruoštą DL klasifikavimo modelį. Pradinį IS įverčio variantą pasiūlė Tim Salimans ir kt. [SGZ⁺16] 2016 metais siekdami automatizuoti daugybės GAN sugeneruotų vaizdų palyginimo procesą. IS matuoja sugeneruotų pavyzdžių kokybę, remiantis keliais kriterijais: kaip gerai sukurti egzemplioriai atspindi įvairias duomenų kategorijas ir kaip sugeneruotą egzempliorių įvertino klasifikavimo modelis. Šį klasifikavimą atlieka tuo metu našiausias Christian Szegedy ir kt. [SLJ⁺14] pristatytas vaizdų klasifikavimo modelis *Inception v3*. Modelis skiria užtikrintumo balą, nusakantį tikimybę egzemplioriaus priklausymo tam tikrai duomenų kategorijai. Tuo tarpu FID pirmą kartą pristatytas Heuselio ir kt. 2017 metais [HRU⁺17], FID yra atstumas tarp dviejų Gauso skirstinių ir yra pagrįstas skirstinių kovariacijos matrica. Mažesnis FID balas rodo geresnį realių ir sugeneruotų vaizdų savybių pasiskirstymą, taigi ir geresnę sugeneruotų pavyzdžių kokybę. FID paprastai laikomas tvirtesniu ir patikimesniu metodu nei IS. FID pranašumas – didesnis atsparumas vaizdo raiškos, stiliaus ir turinio pokyčiams, tad šis įvertis gali būti naudojamas įvairioms vaizdų generavimo užduotims įvertinti, taip pat FID yra mažiau jautrus duomenų rinkinio dydžiui bei klasių skaičiui, šie veiksniai gali turėti įtakos IS įverčiui. Nors tiek IS tiek FID įverčiai sėkmingai atspindi daugelį generatoriaus kokybės aspektų, Chong ir kt. [CF20] pastebi, kad FID pilnai neatspindi klasių įvairovės, imties dydis FID apskaičiavimui turi būti pakankamai didelis, kadangi mažesnių imties dydžių tikrasis FID gali būti pervertintas ir netikslus. Šiuo metu egzistuoja ir daugiau FID įverčio variacijų, tokių kaip: KID (angl. *Kernel Inception Distance*), švarus FID (angl. *Clean FID*) ir t.t. Svarbu paminėti, kad IS ir FID automatizuotas vertinimo metodas dažnai sutapdavo su vizualiu žmogaus vertinimu ir galėtų būti alternatyva vizualiam vertinimui pakeisti.

Alfa precision and recall įvertis

Paprastai dauguma taikomų vertinimo metodų vertinant generatyvinius neuroninius tinklus turi bendrą trūkumą – vertinimo gilumas būna nepakankamas ir atskiri nesėkmių atvejai lieka nepastebėti. Sajjadi ir kt. [SBL⁺18] pabrėžia šią problemą paprastu klasifikavimo pavyzdžiu: abiejų paveikslėlių (10 pav.) FID įvertis panašus, tačiau kairiajame paveikslėlyje skaitmenų imtis, o dešiniajame skaitmenų kokybė yra riboti. Tad galima pastebėti esminę tokio generatoriaus imties problemą, kur generuojami skaitmenys nors ir yra gana geros kokybės ir tikslūs, tačiau šių skaitmenų imtis svyruoja tarp 0 – 3 ir didesnių skaitmenų atvejų (pvz. 7, 8, 9) visiškai nematyti. Tuo tarpu dešinėje pusėje skaitmenų įvairovė didesnė, bet prasta šių skaitmenų kokybė.



10 pav. Sajjadi ir kt. [SBL⁺18] – FID įverčio nepakankamumo problemos pavyzdys pateiktas originaliaame darbe.

Alfa precision and recall [SBL⁺18] yra GAN kokybės nustatymo metrika kuri dažnai naudojama klasifikavimo ir objektų aptikimo uždaviniuose. Ši metrika buvo pasiūlyta kaip alternatyva tradiciniam *precision and recall* vertinant GAN sugeneruotus pavyzdžius. *Alfa precision and recall* atsižvelgia į sugeneruotų pavyzdžių kintamumą ir suteikia išsamesnį GAN generavimo įvertinimą, parodydamas ar GAN sukurti vaizdai vidutiniškai atrodo panašūs į tikrus vaizdus ir juose atsispindi mokymo duomenų įvairovė. Tradicinis šio įverčio variantas yra pagrįstas dvejetainė klasifikacija ir daro prielaidą, kad yra aiški riba tarp realių ir sugeneruotų pavyzdžių. Tačiau ši prielaida gali nepasitvirtinti GAN sugeneruotų egzempliorių atveju, kai riba tarp tikrų ir sugeneruotų pavyzdžių dažnai yra neryški. Tad *Alfa precision and recall* įverčiu siekiama išspręsti šią problemą įvedant derinamą parametą *alfa*, kuris kontroliuoja tikrojo ir sugeneruoto egzempliorių sutapimo dydį. Alfa tikslumu matuojama sukurtų vaizdų, kurie yra tikroviški, dalis, tuo tarpu *recall* dydžiu matuojamas generatoriaus sukurtų rezultatų įvairovės laipsnis. Norint visapusiškai įvertinti GAN efektyvumą, svarbu ištirti tiek generuojamų pavyzdžių kokybę tiek jų įvairovės pasiskirstymą. Šie abu aspektai dažnai vienas kitą veikia, ko pasekoje padidinus modelio generuojamų egzempliorių tikslumą, paprastai sumažėja generuojamų pavyzdžių įvairovė ir atvirkščiai. Taip pat vienas iš pagrindinių *Alfa precision and recall* trūkumų yra tas, kad norint atskirti tikrus ir generuotus pavyzdžius, reikia mokyti klasifikavimo modelį juos tinkamai klasifikuoti. Šio klasifikatoriaus tikslumas gali turėti įtakos įverčio veikimui, kadangi žemo tikslumo modelis gali klaidingai klasifikuoti tikrus egzempliorius kaip sugeneruotus arba atvirkščiai. Bendrai, *Alfa precision and recall* gali būti netinkamas vertinant GAN modelius programose, kuriose sukurti pavyzdžiai turi būti tikroviški, pavyzdžiui, atliekant medicininių vaizdų analizę ar kompiuterinės regos užduotis. Tokiais atvejais dažniau pasirenkami kiti vertinimo metodai, pvz., aukščiau paminėti FID arba IS.

***Chamfer* atstumo įvertis**

Chamfer atstumas Belongie ir kt. [BMP02] metodas, sukurtas dviejų 3D erdvės taškų rinkinių panašumui įvertinti. Dažniausiai naudojamas kompiuterinės regos ir grafikos programose, tokiose kaip taškų debesies formos aptikimas ir derinimas. *Chamfer* atstumas tarp A ir B taškų rinkinių yra apibrėžiamas kaip vidutinis atstumas nuo kiekvieno A rinkinio taško iki artimiausio kaimyno rinkinyje B ir atvirkščiai. Intuityviai ši metrika įvertina, kaip arti taškų rinkinio aibės yra kito rinkinio taškai. Mažesnis atstumas rodo, kad du taškų rinkiniai yra panašūs. GAN sukurtų 3D egzempliorių kontekste *Chamfer* atstumas gali būti naudojamas siekiant įvertinti sugeneruoto modelio ir tikrojo modelio panašumą. Palyginus atstumą tarp sugeneruoto ir realaus modelių, galima gauti kiekybinį sugeneruoto modelio kokybės matą. *Chamfer* atstumo privalumas – vertinimo 3D erdvėje galimybė. Apskaičiuojant *Chamfer* atstumo įvertį, nereikia iš anksto apdoroti taškų debesies rinkinio. Taip pat, ši metrika yra gana atspari triukšmui ir taškų rinkinio duomenų pašaliniais rodikliams. Tačiau ji gali neužfiksuoti visų formų ypatybių tarp dviejų taškų rinkinių ypač kai yra svarbi topologinė rinkinių struktūra. Jei vienoje iš taškų aibių tam tikrose srityse yra didesnis taškų tankis nei kitoje, *Chamfer* atstumo įvertis gali būti labiau paveiktas tankesnio rinkinio ir netinkamai įvertinti rinkinių panašumą. Tad įvertis yra jautrus taškų debesies tankumui ir taškų pasiskirstymui rinkinyje. Bendrai, *Chamfer* atstumas tapo plačiai naudojama kompiuterinės regos ir grafikos metrika, kuri taikoma įvairių užduočių, tokių kaip 3D formų paieškos ar klasifikavimo įvertinimui.

2.5. Apibendrinimas

Apžvelgti giliojo mokymosi iš trimačių duomenų metodai, tokie kaip mokymasis iš kelių vaizdų, tiesioginis taškų rinkinio pritaikymas, tūrinės informacijos perteikimas vokselių tinkleliu bei tūrinės informacijos suspaudimas *octree* medžiu. Kad būtų lengviau trimačių duomenų reprezentacijos formos suskirstomos į Euklido ir ne Euklido kategorijas, pažymint jog ne Euklido kategorijos trimačių modelių reprezentacijos formos yra nenuoseklios ir sudaro sunkumų jas pritaikyti DL uždaviniais spręsti trimatėje erdvėje (pagal [ICN⁺17]). Taikant kiekvieną iš giliojo mokymosi metodų, buvo siekiama išgauti kuo daugiau trimačio modelio ypatybių neprarandant esminių formos detalių. Metodai tarpusavyje skirtingi ir novatoriški, dalis jų manipuliuodavo 3D duomenimis prieš perduodant juos į DL architektūrą, kita dalis metodų keitė DL architektūrą siekiant pateikti 3D duomenis kaip tiesioginę įvestį. Tačiau visus metodus sieja bendras iššūkis – sudėtinga 3D modelių prigimtis ir skaičiavimo resursų trūkumas giliojo mokymosi informacijai apdoroti. Žemiau pateikta (3 lentelė) apžvelgtų DL modelių tikslumo, klasifikuojant ModelNet40 [WSK⁺15] duomenų rinkinį, autorių [ICN⁺17] rezultatų apžvalga. Lentelėje DL modelių tikslumas apskaičiuotas pagal tai kaip DL modeliai klasifikavo skirtingus ModelNet40 [WSK⁺15] duomenų rinkinio objektus.

3 lentelė. Dalis DL modelių tikslumo rezultatų pateiktų originaliame autorių [ICN⁺17] darbe.

Metodas	Įvestis	DL modelis	Tikslumas (ModelNet40)
Brock et al. [BLR ⁺ 16]	3D	CNN	95,54 %
Qi et al. [QSM ⁺ 17]	2D vaizdai	MVCNN	91,40 %
Su et al. [SMK ⁺ 15]	2D vaizdai	MVCNN	90,10 %
Maturana et al. [MS15]	3D vaizdai	3D CNN	83,00 %
Wu et al. [WSK ⁺ 15]	3D	CDBN	77,32 %
Wang et al. [WLG ⁺ 17]	3D	O-CNN	90,20 %

Tiesioginio vokselių tinklelio DL modelio mokymui panaudojimas yra ganėtinai perspektyvus sprendimas norint išgauti formos ypatybes. Tai pademonstruoja neuroninių tinklų architektūros, paremtos vokselių tinklelio pritaikymu trimačių objektų klasifikavimo ir generavimo uždaviniams spręsti: VoxNet [MS15], 3D-GAN [WZX⁺16]. Taip pat galima įvardinti pagrindinius tyrimo iššūkius – nestabili generatyvinių modelių prigimtis ir GAN rezultatų vertinimo kriterijų trūkumas. Automatizuoti IS ir FID vertinimo metodai dažnai sutapdavo su vizualiu žmogaus vertinimu, tačiau dažnu atveju šių įverčių nepakanka DL modelių generavimo įvairovei bei kokybei patikrinti, ko pasekoje dauguma metodų rėmėsi vizualiu pavyzdžių tikrinimu ar priklausė nuo kitų klasifikavimo modelių.

3. Tiriamoji dalis

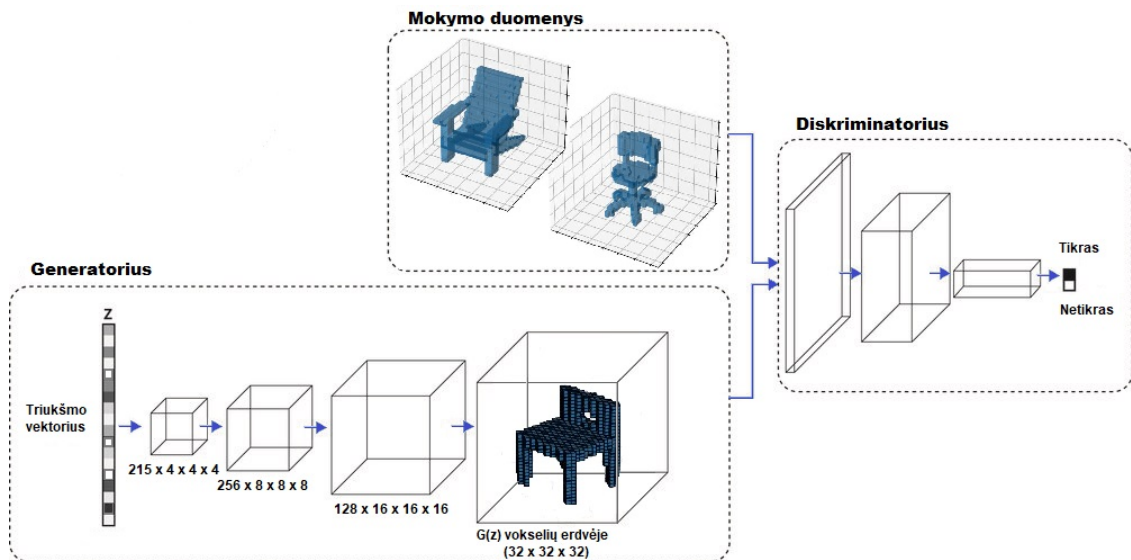
Sugeneruotų 3D modelių tikroviškumo ir tikslumo įvertinimas yra nelengva užduotis, kurios metu reikia patvirtinti arba paneigti šių modelių ir realaus pasaulio objektų, kuriuos jie turi reprezentuoti, panašumą. Tai padaryti gali būti sudėtinga, kadangi trimačių objektų įvairovė yra pakankamai plati, tad kyla klausimas kaip tokius trimačius modelius tarpusavyje vertinti kintant jų stiliui, dydžiui bei pozicijai trimatėje erdvėje. Taip pat 3D objekto forma kiekvienam gali kelti skirtingų asociacijų ir tokių objektų tikroviškumo vertinimas gali būti subjektyvus, o kriterijų pasirinkimas vertinimo metu gali pilnai nepadengti visų realumo aspektų. Tyrimo metu siekiama atrasti 3D objektų panašumui įvertinti tinkančią metriką ir nustatyti bendrus trimačio modelio tikroviškumą bei kokybę atspindinčius kriterijus. Svarbu pabrėžti, kad generavimo proceso metu, 3D modeliai privalo išlaikyti savo naujumą ir bendras objekto klasei priklausančias savybes, o šių trimačių objektų vertinimo procesas turi būti aiškus ir pastovus. Pastovaus proceso ir vienodų kriterijų užtikrinimui pritaikomos panašumo vertinimo metrikos: Sørensen-Dice koeficientas, Jaccard indeksas, Hausdorff atstumas ir Chamfer atstumas. Metrikos taikomos siekiant kiekybiškai palyginti 3D-GAN sugeneruotų trimačių modelių ir ModelNet [VGV⁺09] bei ModelNet40 [WSK⁺15] duomenų rinkinių trimačių modelių ypatybes priskiriant balą, atspindintį šių panašumą. Tiriama papildomi veiksniai veikiantys trimačių modelių kokybę bei panašumo įverčius.

3.1. 3D-GAN modelis

Tyrimo metu trimačių modelių generavimui panaudotas supaprastintas 3D-GAN [WZX⁺16] neuroninio tinklo modelis¹, kurio realizacija įgyvendinta *PyTorch* bibliotekos funkcijomis. Esminis originalaus ir supaprastinto 3D-GAN modelių skirtumas – mažesnis vokselių tinklelio dydis. Šiuo atveju generatorius G atvaizduoja atsitiktine tvarka parinktą vektorių z į $32 \times 32 \times 32$ dydžio tinklelį (11 pav.), vaizduojantį objektą $G(z)$ 3D erdvėje. Originaliame autorių [WZX⁺16] darbe naudojamas $64 \times 64 \times 64$ dydžio vokselių tinklelis. Mažesnis vokselių tinklelio dydis lemia mažesnes skaičiavimų sąnaudas ir iteracijos laiką generatoriaus mokymo metu. Generatorius sudarytas iš tūrinių konvoliucinių filtrų bei partijos normalizavimo ir *ReLU* sluoksnių tarp jų. Galutinio neuronų aktyvavimo metu naudojamas *Sigmoid* sluoksnius.

Diskriminatorius D išveda patikimumo vertę $D(x)$, nusakančią ar 3D objekto įvestis x yra tikroji ar sintetinė. Diskriminatoriaus klasifikavimo rezultatui įvertinti naudojama nuostolių funkcija (1). 3D-GAN sugeneruotas vektorius $G(z)$ konvertuojamas į binarinį trimatį vokselių tinklelį. Šiame darbe kiekvienas z matmuo yra pasiskirstęs $[0, 1]$ intervale, tad konvertavimo metu nustatomas slenkstis, nusakantis ar tikimybinės vektoriaus $G(z)$ reikšmės yra užpildytas ar tuščias vokselis trimačiame vokselių tinklelyje. Slenkstį pasiekusios ir didesnės reikšmės laikomos užpildytais vokseliais ir žymimos 1, slenkščio nepasiekusios ir mažesnės reikšmės laikomos tuščiu vokseliu tinklelyje ir žymimos – 0. Tolimesnėje tyrimo dalyje 3D vokselių tinklelių konvertavimui į binarinę 3D vokselių tinklelių formą pasirinktas 0,6 dydžio slenkstis.

¹Modelio realizacija pasiekama: <https://github.com/xchhuang/simple-pytorch-3dgan>



11 pav. Tyrimo metu naudoto 3D-GAN [WZX⁺16] modelio pagrindiniai architektūros komponentai.

3D-GAN modelio mokymui pritaikyti ModelNet [VGV⁺09] įvairaus stiliaus kėdžių objektai išreikšti binariniais trimačiais masyvais. Mokymo ciklą sudaro 500 iteracijų, kiekvienos iteracijos pabaigoje išsaugoma generatoriaus modelio būseną. Iteracijos pabaigoje neuroninio tinklo modelis testuojamas ir atvaizduojami sugeneruoti kėdžių egzemplioriai, tiriamas šių egzempliorių ir mokymo duomenų panašumas. Vokselių tinklelių dydžio normalizavimui ir dydžio keitimui trimačių modelių vertinimo metu naudojama *skimage* bibliotekos funkcija *transform*, funkcija padidina įvesties vokselių tinklelio mastelį iki nurodyto dydžio. Mastelio keitimas atliekamas apskaičiuojant kiekvieno matmens mastelio koeficientą pagal norimą dydį ir įvesties vokselių tinklelio formą. Tyrimo metu pagrindinis dėmesys skiriamas 3D-GAN rezultatų palyginimui ir įvertinimui.

3.2. Sørensen-Dice koeficientas

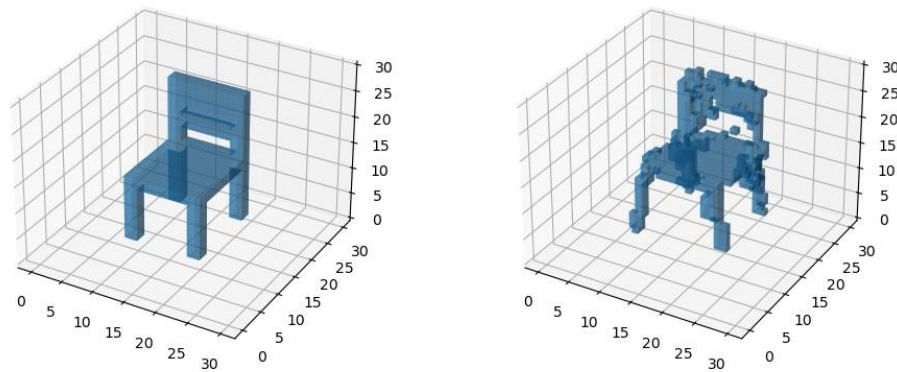
Sørensen-Dice koeficientas [GS16], dar žinomas kaip Dice, yra statistinis matas, naudojamas dviejų duomenų aibių panašumui įvertinti. Dice koeficientas plačiai pritaikomas vaizdų palyginimo, natūralios kalbos apdorojimo ar giliojo mokymosi uždaviniuose. Pavyzdžiui vertinant tekstinių dokumentų panašumą, išmatuojant DL modelių, atliekančių įvairias vaizdų klasifikavimo užduotis našumą, kur lyginamos modelių prognozės su nustatyta tiesa. Dice koeficientas yra naudingas savo paprastumu ir efektyvumu bei yra tinkamas norint kiekybiškai įvertinti dviejų duomenų aibių sutapimą. Tarp dviejų lyginamų aibių X ir Y , Dice apibrėžiamas kaip aibių sankirtos, padaugintos iš dviejų ir aibių elementų skaičiaus sumos santykis, kuris išreiškiamas formule:

$$Dice(X,Y) = \frac{2|X \cap Y|}{|X| + |Y|} \quad (2)$$

Dviejų vaizdų palyginimo atveju, Dice apskaičiuojamas abiejų vaizdų bendrą pikselių skaičių dauginant iš dviejų ir dalijant iš abiejų paveikslėlių pikselių skaičiaus sumos. Dice svyruoja nuo 0 iki 1, kur 1 nusako dviejų aibių identišumą.

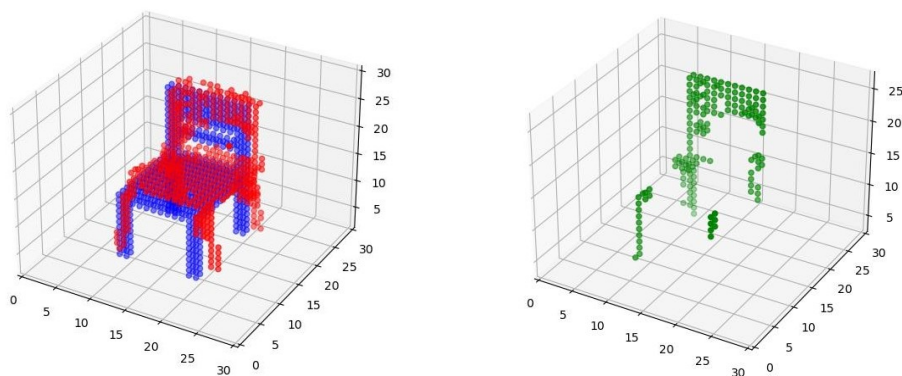
Sørensen-Dice koeficiento pritaikymas 3D modelių palyginime

Sørensen-Dice koeficiento pritaikymo metu vertinami 3D modeliai iš ModelNet [VGV⁺09] apmokymo duomenų aibės ir 3D-GAN sugeneruoti kėdžių egzemplioriai. Kėdžių modeliai pateikiami kaip tūrinės duomenų struktūros – vokselių tinkleliai, išreikšti binariniais trimačiais masyvais. Trimatės kėdžių formos perteiktos vokselių tinkleliu matomos 12 pav., kur kairėje pusėje ModelNet [VGV⁺09] rinkinio kėdės modelis, dešinėje – 3D-GAN sugeneruotas ir į binarinį vokselių tinklelį konvertuotas kėdės egzempliorius. Dviejų modelių vokselių tinkleliai palyginami, apskaičiuojant abiejų tinklelių bendrą vokselių skaičių, padaugintą iš dviejų, ir abiejų tinklelių visų vokselių sumos santykį pagal formulę (2). Šiuo atveju sankirta nurodo bendrą užimtų vokselių skaičių abiejuose vokselių tinkleliuose.



12 pav. Kėdė iš ModelNet [VGV⁺09] duomenų rinkinio ir 3D-GAN sugeneruotas kėdės egzempliorius.

Įvertinant 3D modelių porą Dice koeficientu, galima apskaičiuoti šių trimačių modelių panašumo įvertį ir nustatyti, kurie 3D modeliai yra panašesni sava forma ar vokselių išsidėstymu trimačiame vokselių tinklelyje. Vertinant 12 pav. atvaizduotus kėdžių modelius galima būtų spėti, kad šie modeliai sava forma yra panašūs, tačiau apskaičiuotas Dice koeficiento įvertis siekia 0,27. Dice koeficientas labiau pabrėžia vokselių tinklelių sutapimą, o ne bendrą šių dydį. Tad žemas Dice panašumo įvertis gali būti paaiškinamas 13 pav. vizualiai matoma abiejų kėdžių modelių sankirta, šiuo atveju 3D-GAN sugeneruotas kėdės modelis tinklelyje pozicijuotas aukščiau ModelNet [VGV⁺09] kėdės modelio ir bendrų taškų dalis abiemis tūriniais modeliams yra maža.



13 pav. Kėdžių modelių persidengimas. Raudona spalva žymimas 3D-GAN sugeneruotas kėdės modelis, mėlyna – 3D modelis iš ModelNet [VGV⁺09] apmokymo duomenų rinkinio, žalia – abiejų 3D modelių sankirta.

Nors vizualaus vertinimo metu buvo spėjama, kad 12 pav. atvaizduoti modeliai yra gana panašūs, tačiau šių trimačių modelių Dice panašumo įvertis parodo, kad šių modelių panašumas yra mažas ir siekia tik 27 %.

Kiekybinio Dice pritaikymo metu, 3D modelių palyginimui pasitelkta skirtingų stilių kėdžių modeliai iš ModelNet40 [WSK⁺15] duomenų aibės ir skirtingų stilių 3D-GAN sugeneruoti pavyzdžiai matomi 21 pav. Trimačiai modeliai lyginami tarpusavyje apskaičiuojant šių Dice panašumo koeficientą. Palyginimo metu kiekvienas 3D modelis normalizuojamas į $60 \times 60 \times 60$ vokselių tinklelio dydį. Žemiau (4 lentelė) matomi visų 3D modelių tarpusavio palyginimo rezultatai:

4 lentelė. 3D modelių tarpusavio palyginimo Dice koeficientu įverčiai. Lentelėje matomi 110 - 119 skaitmenys identifikuoja 3D modelio numerį (21 pav.), nuspalvinti laukeliai nurodo panašiausią 3D modelį stulpelyje.

3D modelis	110	111	112	113	114	115	116	117	118	119
110	1,00	0,06	0,13	0,12	0,09	0,07	0,14	0,04	0,09	0,03
111	0,06	1,00	0,14	0,12	0,12	0,04	0,07	0,35	0,04	0,14
112	0,13	0,14	1,00	0,59	0,14	0,02	0,10	0,11	0,05	0,10
113	0,12	0,12	0,59	1,00	0,10	0,02	0,09	0,07	0,04	0,09
114	0,09	0,12	0,14	0,10	1,00	0,04	0,17	0,13	0,06	0,12
115	0,07	0,04	0,02	0,02	0,04	1,00	0,24	0,03	0,04	0,20
116	0,14	0,07	0,10	0,09	0,17	0,24	1,00	0,08	0,04	0,18
117	0,04	0,35	0,12	0,07	0,13	0,03	0,08	1,00	0,06	0,09
118	0,09	0,04	0,05	0,04	0,06	0,04	0,04	0,06	1,00	0,01
119	0,03	0,14	0,10	0,09	0,12	0,20	0,18	0,09	0,01	1,00

Galima pastebėti, kad Dice koeficiento įverčiai bendrai nėra dideli, didžiausias panašumas matomas tarp 112 ir 113 trimačių modelių, kuris siekia 0,59. Matomas 111 ir 117 trimačių modelių panašumas siekiantis 0,35 pagal Dice koeficientą. Vizualiai vertinant 112 ir 113 kėdžių modelius galima būtų sakyti, kad šie modeliai yra labai panašūs tarpusavyje. Žemesnis Dice koeficiento įvertis paaiškinamas skirtingu abiejų trimačių modelių vokselių skaičiumi tinklelyje. Dice įvertis yra

jautrus net ir nežymiam 3D modelio vokselių tankumo skirtumui abiejuose vokselių tinkleliuose ir dažnu atveju bendras panašumo įvertis, esant skirtingam užimtų vokselių skaičiui abiejuose trimačiuose modeliuose, nukenčia. Vizualiai vertinant 111 ir 117 (21 pav.) kėdžių modelius, galima pastebėti panašumų tarp šių modelių, tačiau Dice koeficiento įvertis yra mažas. Dice koeficientu neužfiksuojamas panašumas tarp 115 ir 119 kėdžių modelių, šių modelių panašumas siekia tik 0,11 pagal Dice koeficientą.

3.3. Jaccard indeksas

Jaccard indeksas [NSN⁺13], dar žinomas kaip Jaccard panašumo koeficientas arba IoU (angl. *Intersection over Union*), yra statistinis matas, skirtas kiekybiškai įvertinti duomenų aibių panašumą. Jaccard koeficientas plačiai naudojamas duomenų analizės srityje, nustatant duomenų rinkinių panašumus, asociacijas ir ryšius tarp jų ar pasikartojantį šabloną. Jaccard apibrėžiamas kaip dviejų lyginamų aibių X ir Y sankirtos ir šių aibių sąjungos santykis, išreiškiamas formule:

$$Jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (3)$$

Dviejų vaizdų palyginimo atveju, Jaccard indeksas būtų apskaičiuojamas kaip vaizdų bendrų pikselių skaičiaus ir abiejų vaizdų pikselių sąjungos santykis. Jaccard panašumo koeficientas svyruoja nuo 0 iki 1, kur 1 nurodo aibių identiškumą. Galima pastebėti ryšį tarp Dice ir Jaccard koeficientų. Jaccard indeksas gali būti apskaičiuojamas pagal Dice koeficiento reikšmę ir atvirkščiai, pritaikant šias formules:

$$Jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{2|X \cap Y|}{2(|X| + |Y| - |X \cap Y|)} = \frac{DICE}{2 - DICE} \quad (4)$$

$$Dice = \frac{2JAC}{1 + JAC} \quad (5)$$

Jaccard koeficientas gali būti patikimesnis vertinimo matas esant subalansuotiems, panašaus dydžio duomenų rinkiniams, kadangi Jaccard koeficientas vienodai pabrėžia bendrą sutapimą tarp dviejų duomenų aibių, ko pasekoje vienodai įvertinami bendrieji vokseliai ir vokselių trūkumas abiejuose trimačiuose tinkleliuose. Tuo tarpu Dice nepriklauso nuo lyginamų rinkinių dydžio, tai gali būti naudinga dirbant su nesubalansuotais duomenų rinkiniais, kuomet viena kategorija ar rinkinys gali dominuoti kitoje. Jaccard yra jautresnis nedideliems persidengimo pokyčiams ir gali būti naudojamas kuomet visiškas objektų persidengimo atitikimas yra reikšmingas, pavyzdžiui atliekant vaizdų apdoravimo ar objektų aptikimo užduotis.

Jaccard koeficiento pritaikymas 3D modelių palyginime

Trimačio modelio iš ModelNet [VGV⁺09] apmokymo duomenų rinkinio ir 3D-GAN sugeneruoto kėdės modelio palyginime Jaccard koeficientu (12 pav.), modelių panašumas siekia 0,15. Kaip ir buvo tikėtasi, Jaccard koeficientas jautriau reaguoja į skirtingą trimačių tinklelių vokselių

kiekį nei Dice koeficientas. Kadangi bendrų vokselių sankirta šių 3D modelių palyginime yra maža, Jaccard įvertis paveikiamas labiau.

Kiekybinio Jaccard koeficientu vertinimo metu, modelių palyginimui naudojami skirtingų stilių kėdžių modeliai iš ModelNet40 [WSK⁺15] duomenų aibės ir skirtingų stilių 3D-GAN sugeneruoti kėdžių egzemplioriai, trimačiai modeliai gali būti matomi 21 pav. Trimačiai modeliai lyginami tarpusavyje apskaičiuojant Jaccard panašumo įvertį pagal (3) formulę. Palyginimo metu kiekvienas modelis normalizuojamas į $60 \times 60 \times 60$ vokselių tinklelio dydį. Žemiau (5 lentelė) matomi visų 3D modelių tarpusavio palyginimo rezultatai:

5 lentelė. 3D modelių tarpusavio palyginimo Jaccard koeficientu įverčiai. Lentelėje matomi 110 - 119 skaitmenys identifikuoja 3D modelio numerį (21 pav.), nuspalvinti laukeliai nurodo panašiausią 3D modelį stulpelyje.

3D modelis	110	111	112	113	114	115	116	117	118	119
110	1,00	0,03	0,07	0,06	0,05	0,03	0,07	0,02	0,05	0,02
111	0,03	1,00	0,08	0,07	0,06	0,02	0,03	0,21	0,02	0,08
112	0,07	0,08	1,00	0,42	0,07	0,01	0,05	0,06	0,02	0,06
113	0,06	0,07	0,42	1,00	0,05	0,01	0,05	0,04	0,02	0,05
114	0,05	0,06	0,07	0,05	1,00	0,02	0,09	0,07	0,03	0,07
115	0,03	0,02	0,01	0,01	0,02	1,00	0,14	0,01	0,02	0,11
116	0,07	0,03	0,05	0,05	0,09	0,14	1,00	0,04	0,02	0,10
117	0,02	0,21	0,06	0,04	0,07	0,01	0,04	1,00	0,03	0,05
118	0,05	0,02	0,02	0,02	0,03	0,02	0,02	0,03	1,00	0,00
119	0,02	0,08	0,06	0,05	0,07	0,11	0,10	0,05	0,00	1,00

Galima matyti (5 lentelė), kad įverčiai, 3D modelių panašumą vertinant Jaccard koeficientu, yra žemesni lyginant su Dice gautais įverčiais (4 lentelė). Tačiau bendras modelių panašumo vertinimas beveik visais atvejais atitinka panašumo rezultatus gautus taikant Dice koeficientą. 3D modeliai 112 ir 113 yra laikomi panašiausiais modeliais tarpusavyje, kurių panašumas siekia 0,42 pagal Jaccard koeficientą. Užfiksuojamas nežymus panašumas tarp 111 ir 117 kėdžių modelių, tačiau kiti trimačių modelių panašumo įverčiai yra maži.

3.4. Hausdorff atstumas

Hausdorff atstumo (HD) [HKR93] apskaičiavimas suteikia kiekybinį dviejų taškų rinkinių erdvėje palyginimą. Ši metrika plačiai naudojama įvairiose vaizdo apdorojimo ar kompiuterio regos užduotyse, robotikoje, planuojant roboto kelią, kuomet norima nustatyti atstumą tarp dabartinės roboto padėties ir kelionės tikslo, taip pat HD yra dažnas medicininių vaizdų įvertinimo matas. Hausdorff atstumas apibrėžiamas kaip didžiausias atstumas tarp bet kurio taško vienoje aibėje ir artimiausio tokio taško atitiktoms kitoje aibėje. Kitaip tariant, HD matuoja, kaip toli viena nuo kitos yra dvi aibės tolimiausiuose jų taškuose. Laikoma, kad aibės yra artimos viena kitai, kuomet jų Hausdorff atstumas yra mažiausias, identiškos kuomet atstumas lygus 0. Hausdorff atstumas tarp aibių X ir Y apskaičiuojamas formule:

$$HD(X,Y) = \max(h(X,Y), h(Y,X)) \quad (6)$$

kur X ir Y yra baigtinės aibės erdvėje, o $h(X,Y)$ ir $h(Y,X)$ yra nukreipto Hausdorff atstumo įverčiai, apskaičiuojami pagal formulę:

$$h(X,Y) = \max_{x \in X} \min_{y \in Y} \|x - y\| \quad (7)$$

nukreiptas Hausdorff atstumas $h(X,Y)$ nuo aibės X iki aibės Y žymi tašką $x \in X$, kuris yra toliausiai nuo bet kurio taško $y \in Y$, ir išmatuoja atstumą nuo taško x iki artimiausio kaimyno y . Tokiu būdu aibės X taškai reitinguojami pagal minimalų atstumą iki aibės Y taškų, kur nukreipto Hausdorff atstumo įverčiui naudojamas didžiausias minimalių atstumų įvertis. Atitinkamai apskaičiuojamas nukreipto Hausdorff atstumo įvertis $h(Y,X)$ nuo aibės Y iki aibės X . Galiausiai Hausdorff atstumas $HD(X,Y)$ apibrėžiamas kaip didžiausias nukreipto Hausdorff atstumo įvertis tarp $h(X,Y)$ ir $h(Y,X)$. Šiuo metu esama įvairių algoritmų, skirtų Hausdorff atstumui apskaičiuoti, kai kurie algoritmai gali būti efektyvesni už kitus, tačiau bendru atveju Hausdorff atstumo algoritmo sudėtingumas esant n ir m aibių X ir Y dydžiams, gali būti apskaičiuojamas pagal formulę:

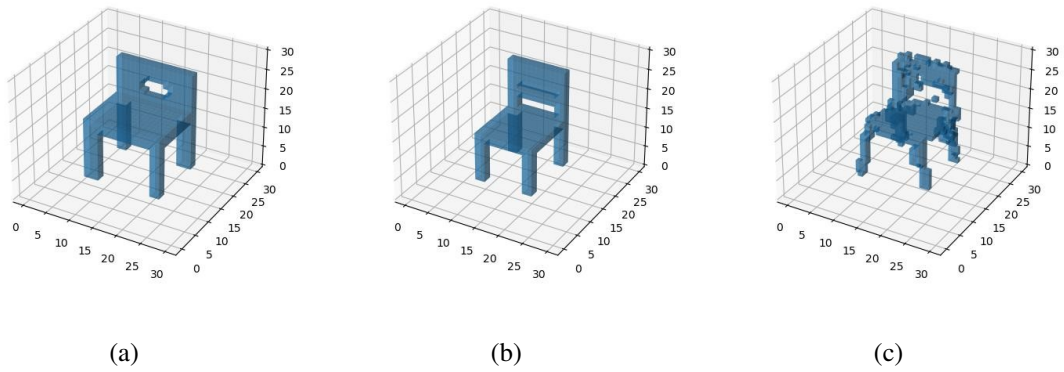
$$O((n + m)\log(n + m)) \quad (8)$$

Hausdorff atstumas įvertina neatitikimą tarp dviejų baigtinių rinkinių, kurie yra fiksuotose padėtyse vienas kito atžvilgiu, tad bendras rinkinių elementų skaičius yra neatsiejamas algoritmo sudėtingumo ir laiko nustatyme.

Hausdorff atstumo apskaičiavimas remiantis 2D trimačių modelių pjūviais

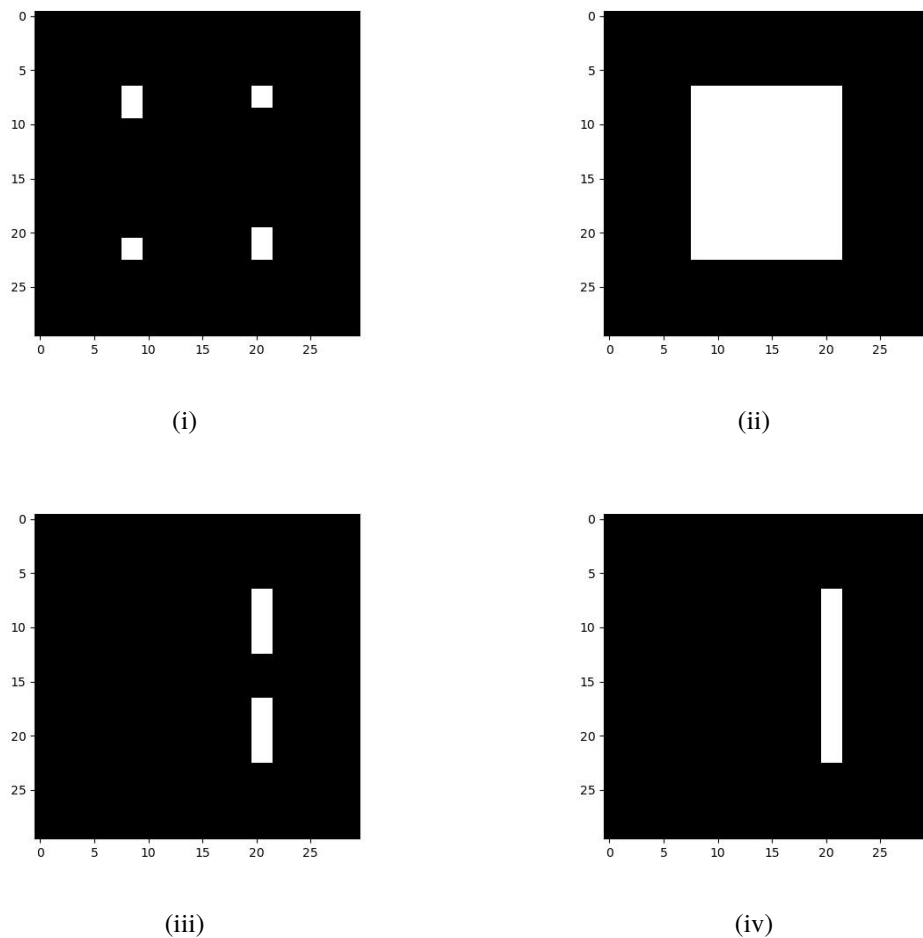
Trimačio vokselių tinklelio skaidymas į 2D pjūvių vaizdus, Hausdorff atstumui tarp jų palyginti, yra vienas iš būdų norint įvertinti trimatę formą. 3D vokselių tinklelis išilgai skaidomas pagal vieną iš XY , YZ ar ZX plokštumų išgaunant 2D pjūvių rinkinį. Kiekvienam atitinkamam 2D pjūviui (pvz., XY pjūviui iš pirmojo modelio ir XY pjūviui iš antrojo modelio), apskaičiuojamas 2D Hausdorff atstumas pagal aukščiau pateiktą formulę (7). Galutiniam įverčiui išrenkamas didžiausias Hausdorff atstumas tarp plokštumų pjūvių pagal (6) formulę.

Tyrimo metu HD atstumui 2D pjūviais įvertinti panaudoti ModelNet [VGV⁺09] duomenų rinkinio kėdžių modeliai ir sugeneruotas 3D-GAN neuroninio tinklo kėdės modelis (modeliai matomi 14 pav.). ModelNet [VGV⁺09] apmokymo duomenų rinkinį sudaro įvairaus stiliaus kėdės, tad pjūviai tarpusavyje lyginant kelis rinkinio modelius gali būti labai skirtingi.



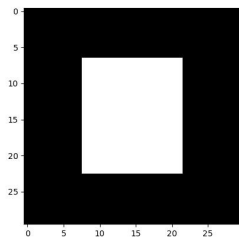
14 pav. (a) ir (b) ModelNet [VGV⁺09] apmokymo rinkinio kėdės modeliai, (c) 3D-GAN sugeneruotas kėdės egzempliorius.

Iteruojamas trimatis vokselių tinklelis sukuria 32 skirtingus binarinius pjūvių vaizdus, kur kiekvienas pjūvio vaizdas išreikštas 32×32 dydžio matrica. Balta spalva žymimi užpildyti vokseliai, tuo tarpu juoda – tuščia vokselių tinklelio erdvė. Paveikslėlyje matomi (žr. 15 pav.) išskirtiniai kėdės (a) 14 pav. modelio pjūviai: (i) kėdės kojelių pjūvis, (ii) kėdės sėdimosios dalies pagrindas, (iii) kėdės atloše esanti tuščia apskrita dalis bei (iv) kėdės atlošo viršus.

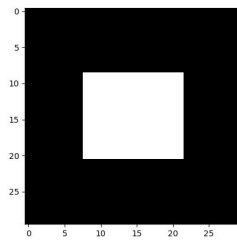


15 pav. Išskirtiniai ModelNet [VGV⁺09] kėdės (a) (14 pav.) 2D pjūviai.

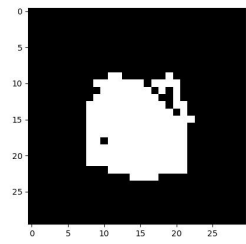
Pjūviai suteikia išsamią informaciją apie patį trimatį modelį, kadangi toks modelio skaidymas pjūviais leidžia pastebėti kiekvieno vokselio nukrypimą nuo bendros modelio struktūros, tad yra patogus būdas trimačio modelio nuoseklumui vizualiai įvertinti. Galima įvertinti, kad lyginant su apmokymo rinkinio modeliais (a) ir (b), 3D-GAN sugeneruotas kėdės modelis (c) atrodo netvaringas (16 pav.). Sugeneruoto modelio pagrindo pjūvio forma ((iii) 16 pav.) nėra nuosekli ir pilnai užpildyta vokseliais. Toks atvejis modelio apmokymo metu yra tikėtinas, tad tinkamai įvertinti modelį esant modelio chaotiškumui yra svarbu, norint pasakyti ar trimatė forma yra kokybiška. Paveikslėlių rinkinyje (16 pav.) kiekvieno modelio atitinkamas pjūvis išdėstytas vertikalia tvarka (pvz., (a) modelio pjūviai išsidėstę stulpeliu žemyn (i), (iv), (vii), (x)).



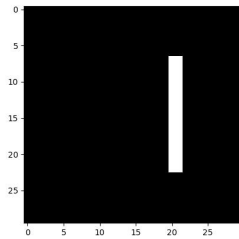
(i)



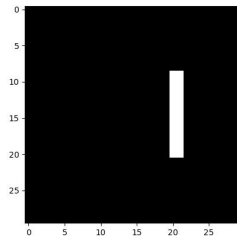
(ii)



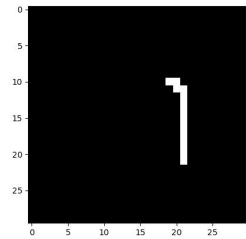
(iii)



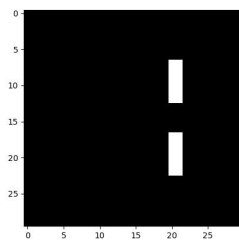
(iv)



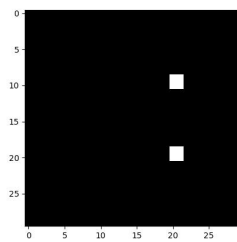
(v)



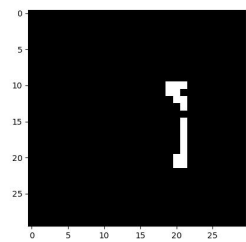
(vi)



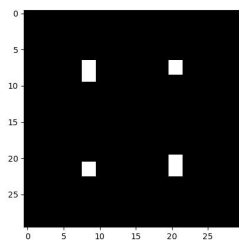
(vii)



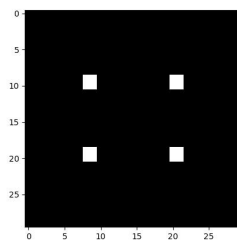
(viii)



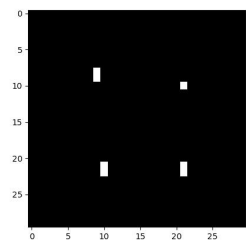
(ix)



(x)



(xi)



(xii)

16 pav. 14 pav. (a), (b) ir (c) trimačių modelių išskirtiniai pjūviai.

Pasiremiant 14 pav. kėdžių išskirtiniais pjūviais apskaičiuojamas Hausdorff atstumas tarp rinkinyje matomų pjūvių tokių kaip: visų trijų kėdžių sėdimosios dalies pagrindas, atlošas ir atloše esanti apskrita dalis, kojelių pjūvis. Atstumui apskaičiuoti (6 lentelė) pritaikyta (6) formulė.

6 lentelė. Išskirtinių (a), (b) ir (c) trimačių modelių pjūvių (16 pav.) palyginimas nukreiptu Hausdorff atstumu (7).

2D Pjūvis	$h(a, b)$	$h(a, c)$	$h(b, c)$
Pagrindas	2,83	2,24	2,65
Atlošo viršus	0,00	1,00	1,00
Atlošo vidurys	2,45	3,32	3,16
Kojelės	0,00	0,00	0,00

7 lentelė. Modelių (a), (b) ir (c) bendras visų 32 pjūvių palyginimas Hausdorff atstumu (6).

HD(a, b)	HD(a, c)	HD(b, c)
8,94	9,11	8,49

Nukreiptas Hausdorff atstumas tarp išskirtinių pjūvių yra nedidelis (lentelė 6), didžiausias atstumas tarp pjūvių siekia 3,32, remiantis šiuo įverčiu būtų galima spėti, kad kėdžių modeliai tarpusavyje panašūs, tačiau atsižvelgiant į bendrą visų trimačių modelių 2D pjūvių Hausdorff atstumą (lentelė 7) matomas ženklus šių trimačių modelių skirtumas. Didžiausio atstumo atveju, atstumas siekia 9,11. Visų modelių pjūvių įvertinimui modeliai išilgai išskirstomi pagal XY , YZ ir ZX plokštumas, kur nukreiptas Hausdorff atstumas apskaičiuojamas kiekvienam atitinkamų pjūvių rinkiniui gražinant maksimalų HD atstumo įvertinimą pagal formulę (6). Svarbu atkreipti dėmesį, kad matuojant atstumą tik tarp išskirtinių pjūvių, toks palyginimas nebūtų tikslingas ir reikalautų selektyviai pjūvius išrinkti bei būtų prarandama labai daug informacijos apie patį modelio skeletą. Bendro pjūvių palyginimo Hausdorff atstumu metu atstumo įverčius paveikia skirtinga 3D modelių padėtis vokselių tinklelyje, trimačių modelių pozicijos ir dydžio vokselių tinklelyje normalizavimas būtų pirminis žingsnis norint sumažinti šių aspektų poveikį Hausdorff atstumo įverčiui.

Kiekybinio Hausdorff atstumo remiantis trimačių modelių 2D pjūviais įvertinimo metu, modelių palyginimui naudojami skirtingų stilių kėdžių modeliai iš ModelNet40 [WSK⁺15] duomenų aibės ir skirtingų stilių 3D-GAN sugeneruoti kėdžių egzemplioriai, trimačiai modeliai gali būti matomi 21 pav. Trimačiai modeliai vertinami apskaičiuojant jų Hausdorff atstumą remiantis 3D modelių 2D pjūviais. Palyginimo metu kiekvienas modelis normalizuojamas į $60 \times 60 \times 60$ vokselių tinklelio dydį. Žemiau (8 lentelė) matomi visų 3D modelių tarpusavio palyginimo rezultatai:

8 lentelė. 3D modelių tarpusavio palyginimo Hausdorff atstumu remiantis 2D pjūviais įverčiai. Lentelėje matomi 110 - 119 skaitmenys identifikuoja 3D modelio numerį (21 pav.), nuspalvinti laukeliai nurodo panašiausią 3D modelį stulpelyje.

3D modelis	110	111	112	113	114	115	116	117	118	119
110	0,00	21,12	25,34	23,43	21,89	20,98	20,98	21,12	21,14	20,98
111	21,12	0,00	23,90	21,93	20,40	19,42	19,42	20,12	19,60	19,42
112	25,34	23,90	0,00	16,28	22,58	22,93	22,67	22,91	21,95	22,43
113	23,43	21,93	16,28	0,00	21,12	20,74	20,71	21,12	18,87	20,45
114	21,89	20,40	22,58	21,12	0,00	19,57	19,57	20,30	19,75	19,47
115	20,98	19,42	22,93	20,74	19,57	0,00	17,66	20,10	16,82	18,00
116	20,98	19,42	22,67	20,71	19,57	17,66	0,00	20,10	18,73	18,55
117	21,12	20,12	22,91	21,12	20,30	20,10	20,10	0,00	20,27	20,10
118	21,14	19,60	21,95	18,87	19,75	16,82	18,73	20,27	0,00	18,19
119	20,98	19,42	22,43	20,45	19,47	18,00	18,55	20,10	18,19	0,00

Hausdorff atstumo, remiantis trimačių formų 2D pjūviais, vertinimo metu galima pastebėti, kad atstumo galutinį įvertį lemia formų pozicija ir dydis vokselių tinklelyje. Trimačiai modeliai 115, 116 ir 119 yra skirtingų stilių, tačiau šių modelių mastelis trimačiame tinklelyje yra panašus, tad vertinimo metu šių 3D modelių panašumas įvertinamas vienodai: 110, 111 ir 117 atvejais. Užfiksuojamas 112 ir 113 trimačių modelių panašumas, tačiau bendras panašumo vertinimas lyginant su Dice ir Jaccard koeficientų rezultatais daugeliu atvejų neatitinka. Gana skirtingų stilių modeliai turi identiškus Hausdorff atstumus, ko pasekoje būtų sudėtinga spręsti kurie iš šių modelių sava forma iš tikrųjų yra panašūs. Toks HD atstumo vertinimo būdas galimai nėra tikslus, trimačių modelių sluoksniavimas pjūviais ir atskirų pjūvių porų vertinimas sąlygoja bendros trimatės formos skeleto informacijos praradimą ir bendrą įverčio netikslumą.

Hausdorff atstumo apskaičiavimas tiesiogiai vertinant 3D vokselių tinklelius

Pasiremiant Taha ir kt. [TH15] pasiūlytu Hausdorff atstumo apskaičiavimu tiesiogiai vertinant 3D vokselių tinklelius tarpusavyje, Hausdorff atstumu išmatuojamas dviejų tūrinių formų, perteiktų vokselių tinkleliu, skirtumas. Tad Hausdorff atstumo tarp dviejų vokselių tinklelių apskaičiavimo metu, lyginamas kiekvienas vieno tinklelio vokselis su kiekvienu kito tinklelio vokseliu. Atstumas tarp dviejų vokselių paprastai apibrėžiamas Euklido atstumu erdvėje. Tuomet Hausdorff atstumas yra didžiausias atstumas tarp vokselio viename tinklelyje ir artimiausio jo kaimyno kitame tinklelyje. Taikant Hausdorff atstumo metriką tiesiogiai vertinant 3D vokselių tinklelius, galima užfiksuoti tūrinius formų skirtumus, atsižvelgiant į erdvinį vokselių pasiskirstymą ir jų atstumus vienas nuo kito. Tuo tarpu, apskaičiuojant Hausdorff atstumą remiantis 2D trimatės formos pjūviais, kiekvienas pjūvis traktuojamas kaip nepriklausoma forma, tad matuojamas atitinkamų dviejų formų pjūvių skirtumas. Šis metodas neatsižvelgia į erdvinį ryšį tarp pjūvių ar į bendrą 3D formos struktūrą.

Hausdorff atstumui apskaičiuoti kiekybiškai palyginami ModelNet40 [WSK⁺15] ir 3D-GAN sugeneruoti kėdžių modeliai. Modeliai lyginami tarpusavyje apskaičiuojant Hausdorff atstumo

įvertį pagal formulę (6) vertinant atstumą tarp dviejų tūrinių vokselių tinklelių. Palyginimo metu kiekvienas modelis normalizuojamas į $60 \times 60 \times 60$ vokselių tinklelio dydį.

9 lentelė. 3D vokselių tinklelių tarpusavio palyginimo Hausdorff atstumu įverčiai. Lentelėje matomi 110 - 119 skaitmenys identifikuoja 3D modelio numerį (21 pav.), nuspalvinti laukeliai nurodo panašiausią modelį stulpelyje.

3D modelis	110	111	112	113	114	115	116	117	118	119
110	0,00	20,90	17,58	18,36	20,02	27,73	25,02	20,02	17,94	24,10
111	20,90	0,00	21,95	22,67	12,65	22,11	22,65	10,44	27,04	19,39
112	17,58	21,95	0,00	3,61	21,26	25,81	27,78	21,95	25,10	21,35
113	18,36	22,67	3,61	0,00	22,36	25,53	27,78	22,67	25,32	21,35
114	20,02	12,65	21,26	22,36	0,00	22,98	19,44	14,35	29,00	21,00
115	27,73	22,11	25,81	25,53	22,98	0,00	19,80	22,11	24,92	27,10
116	25,02	22,65	27,78	27,78	19,44	19,80	0,00	21,56	28,23	14,56
117	20,02	10,44	21,95	22,67	14,35	22,11	21,56	0,00	27,02	19,39
118	17,94	27,04	25,10	25,32	29,00	24,92	28,23	27,02	0,00	21,00
119	24,10	19,39	21,35	21,35	21,00	27,50	14,56	19,39	21,00	0,00

Taikant Hausdorff atstumo apskaičiavimą tiesiogiai vertinant 3D vokselių tinklelius tarpusavyje (9 lentelė), galima pastebėti, kad 3D modelių panašumo įverčiai yra tikslesni, užfiksuojamas ženklus trimačių formų panašumas tarp 112 ir 113 modelių, atstumas tarp šių formų nėra didelis ir siekia 3,61. Taip pat daugeliu atvejų HD atstumo vertinimo rezultatai, tiesiogiai apskaičiuojant atstumus tarp trimačių vokselių tinklelių, sutapo su Dice (4 lentelė) ir Jaccard (5 lentelė) panašumo vertinimo rezultatais. Palyginant panašumo vertinimo rezultatus tarp HD atstumo vertinimo remiantis 3D modelių 2D pjūviais (8 lentelė) ir HD atstumo vertinimo tiesiogiai tarp 3D vokselių tinklelių (9 lentelė), matoma, kad panašumo vertinimas sutapo tik 3 iš 10 vertinimo atvejų.

3.5. Chamfer atstumas

Chamfer atstumas [WPZ⁺21] (CD) yra universali metrika, naudojama kiekybiškai įvertinti dviejų taškų rinkinių ar 3D formų panašumą. Chamfer atstumo matavimas pritaikomas įvairiose srityse, tokiose kaip kompiuterinė rega, vaizdo apdorojimas ir 3D modelių atpažinimas. Chamfer atstumu matuojamas mažiausias vidutinis atstumas tarp lyginamų taškų aibių. Šis atstumas apskaičiuojamas pagal formulę:

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2 \quad (9)$$

kur kiekvienas taškas $x \in S_1$ atitinkamai palyginamas su artimiausiu kaimyniniu tašku $y \in S_2$ ir atvirkščiai. Aibės laikomos identiškomis, kuomet jų Chamfer atstumas yra lygus 0. Galima išvelgti Chamfer atstumo ir Hausdorff atstumo metrių skirtumą. Hausdorff atstumu apskaičiuojamas didžiausias atstumas tarp bet kurio taško vienoje aibėje ir artimiausio taško atitiktoms kitoje

aibėje. Tuo tarpu Chamfer atstumas yra žinomas kaip atstumas nuo taško iki rinkinio, kur matuojamas vidutinis atstumas nuo kiekvieno taško viename rinkinyje iki artimiausio taško kitame rinkinyje. Chamfer atstumas gali būti patikimesnis esant aibių pavienių taškų nuokrypiui, kadangi Chamfer atstumu apskaičiuojamas visų taškų atstumų vidurkis ir smulkūs taškų nukrypimai turi mažesnę įtaką bendrai atstumo vertei.

3D vokselių tinklelio palyginimo ir vertinimo kontekste CD tampa ypač vertingas. Apskaičiuojant Chamfer atstumą tarp dviejų vokselių tinklelių, galima įvertinti, kaip tiksliai dera tinkleliuose esančios trimatės formos ar šių formų struktūros. Tai leidžia įvertinti skirtingų 3D objektų panašumą, atlikti 3D formų atkūrimą, trimačių objektų atpažinimą ar 3D scenos rekonstrukcijos užduotis. Chamfer atstumas leidžia išsamiai analizuoti vokseliais pagrįstų duomenų erdvinį išsidėstymą ir geometrines savybes.

Chamfer atstumo pritaikymas 3D modelių palyginime

Kiekybinio Chamfer atstumo 3D modelių palyginimo metu, modelių palyginimui naudojami skirtingų stilių kėdžių modeliai iš ModelNet40 [WSK⁺15] duomenų aibės ir skirtingų stilių 3D-GAN sugeneruoti kėdžių egzemplioriai (21 pav.). Trimačiai modeliai lyginami tarpusavyje apskaičiuojant Chamfer atstumą pagal (9) formulę. Palyginimo metu kiekvienas modelis normalizuojamas į $60 \times 60 \times 60$ vokselių tinklelio dydį.

10 lentelė. 3D vokselių tinklelių tarpusavio palyginimo Chamfer atstumu įverčiai. Lentelėje matomi 110 - 119 skaitmenys identifikuoja 3D modelio numerį (21 pav.), nuspalvinti laukeliai nurodo panašiausią modelį.

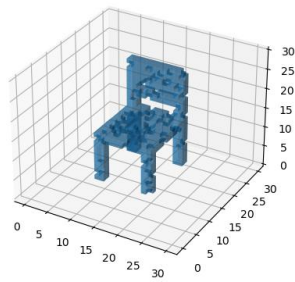
3D modelis	110	111	112	113	114	115	116	117	118	119
110	0,00	5,02	4,27	4,18	4,83	6,88	4,93	5,01	4,23	6,62
111	5,02	0,00	3,87	4,09	3,60	5,43	4,33	1,41	7,93	4,00
112	4,27	3,87	0,00	0,52	4,30	6,80	5,82	4,05	5,81	5,63
113	4,18	4,09	0,52	0,00	4,65	7,40	6,16	4,44	6,10	6,32
114	4,83	3,60	4,30	4,65	0,00	5,86	3,32	3,56	6,37	5,02
115	6,88	5,43	6,80	7,40	5,86	0,00	3,25	5,63	9,01	2,12
116	4,93	4,33	5,82	6,16	3,32	3,25	0,00	4,55	8,31	3,22
117	5,01	1,41	4,05	4,44	3,56	5,63	4,55	0,00	6,90	4,55
118	4,23	7,93	5,81	6,10	6,37	9,01	8,31	6,90	0,00	9,03
119	6,62	4,00	5,63	6,32	5,02	2,12	3,22	4,55	9,03	0,00

Trimačių modelių panašumo Chamfer atstumu vertinimas, matomas 10 lentelėje, daugeliu atvejų atitinka Dice (4 lentelė), Jaccard (5 lentelė) ir Hausdorff atstumo (9 lentelė), vertinant 3D vokselių tinklelius tarpusavyje, rezultatus. Matoma, kad 3D modelių palyginimas Chamfer atstumu sutampa 70% atvejų lyginant su Dice ir Jaccard koeficientų palyginimo rezultatais, taip pat 60% atvejų Hausdorff atstumo vertinimo rezultatais. Chamfer atstumo apskaičiavimo metu sėkmingai užfiksuojamas didžiausias trimačių formų panašumas tarp 3D modelių 112 ir 113, taip pat matomas

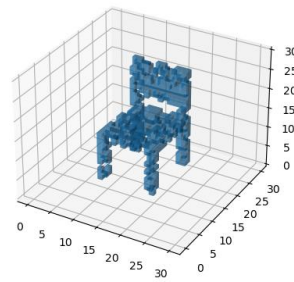
ženklus panašumas tarp 111 ir 117 formų bei 115 ir 119 sugeneruotų formų. Chamfer atstumu užfiksuojami ir tokie panašumai tarp formų kaip kojelių sutapimas 111 ir 114 modelių atveju, taip pat matomas panašumas tarp 110 ir 118 modelių atveju esant skirtingam šių modelių vokselių kiekiui.

3.6. Vokselių tinklelio tankumo įtaka 3D modelių panašumo įverčiams

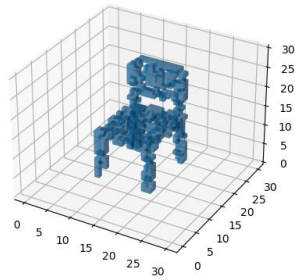
Formų chaotiškumas gana dažnas reiškiny 3D-GAN generatoriaus sukurtuose trimačiuose modeliuose. Kadangi generatoriaus sukurti rezultatai išreiškiami racionaliųjų skaičių vektoriais tinkamam tokių duomenų panaudojimui svarbus tikslingas slenksčio nustatymas, kurio pagalba galima būtų nuspręsti ar vokselis tinklelyje yra užimtas ar tuščias. Slenkstį pasiekusios ir didesnės reikšmės tokiu atveju laikomos užimtais vokseliais, priešingu atveju, laikoma, kad reikšmė nėra pakankama ir yra tuščia vokselių tinklelio erdvė. Šiuo būdu racionaliųjų skaičių išraiška sugeneruotas trimatis modelis konvertuojamas į binarinį ir remiantis šiais duomenimis atliekamas tolimesnis 3D modelio vertinimas. Slenksčio pasirinkimas dažnu atveju sąlygoja užimtų vokselių skaičių vokselių tinklelyje, kuo aukštesnė slenksčio reikšmė tuo mažiau užpildytų vokselių gaunama konvertavimo metu. Vokselių tankumas tinklelyje modelių panašumo vertinimo metu gali daryti reikšmingą įtaką panašumo įverčiams. Norint užfiksuoti kaip stipriai paveikiami įverčiai, kintant trimatės formos vokselių tinklelio tankumui, vieno iš trimačių modelių vokselių skaičius atsitiktiniu būdu sumažinamas 15 %, 30 %, 45 % ir 60 % nuo pradinio 3D modelio vokselių skaičiaus, toks modelis toliau lyginamas su tuo pačiu, nepaveiktu 3D modeliu ir įvertinamas remiantis Dice, Jaccard, Hausdorff atstumo ir Chamfer atstumo metrikomis. Hausdorff atstumas apskaičiuojamas tiesiogiai vertinant atstumą tarp trimačių tinklelių. Žemiau (17 pav.) matoma, kaip vokselių tankumo ir nuoseklumo sumažinimas paveikia trimatę formą:



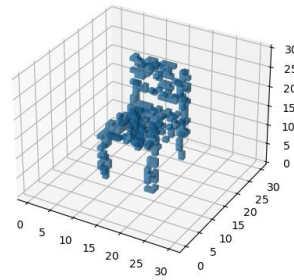
(a)



(b)



(c)



(d)

17 pav. ModelNet [VGV⁺09] apmokymo duomenų 3D modelio vokselių skaičius sumažinamas (a) 15 %, (b) 30 %, (c) 45 % (d) 60 % nuo pradinio vokselių skaičiaus.

Sumažėjus vokselių skaičiui modelyje, modelio forma tampa nenuosekli ir skylėta, tačiau šio tyrimo metu išlaikomas bendras trimatės formos skeletas padėsiantis tokią formą vizualiai identifikuoti.

Panašumo įverčiai, matomi 11 lentelėje, apskaičiuoti pagal Dice (2), Jaccard (3), Hausdorff atstumo (6) ir Chamfer atstumo (9) formules. Rezultatai parodo, kad visi įverčiai yra paveikiami vokselių skaičiaus 3D modelyje pokyčio.

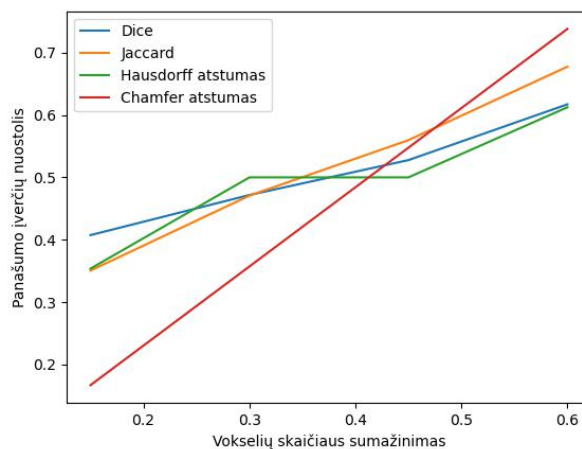
11 lentelė. Vokselių tankumo įtaka panašumo įverčiams

Vokselių skaičius modelyje	Dice	Jaccard	Hausdorff atstumas	Chamfer atstumas
720	1,00	1,00	0,00	0,00
615 (15 % mažiau)	0,92	0,85	1,00	0,07
501 (30 % mažiau)	0,82	0,70	1,41	0,15
417 (45 % mažiau)	0,73	0,58	1,41	0,23
305 (60 % mažiau)	0,60	0,42	1,73	0,31

Sumažėjus vokselių skaičiui, sumažėja 3D modelio detalumas, ko pasekoje prarandama erdvinė trimačio modelio informacija. Paveikto vokselių tinklelio ir pilno vokselių tinklelio panašumo vertinimo rezultatai (18 pav.) rodo, kad Dice, Jaccard, Hausdorff atstumo ir Chamfer atstumo metrikų įverčiai yra paveikiami vokselių skaičiaus trimatčiame modelyje pokyčio. Iš šių keturių metrikų,

Hausdorff atstumo įverčiai yra mažiausiai paveikiami vokselių skaičiaus sumažėjimo 3D modelyje. Tuo tarpu Chamfer atstumo įverčiai paveikiami daugiausiai. Sumažėjus vokselių tankumui, lieka vis mažiau taškų trimatės formos paviršiui atvaizduoti, tai gali žymiai paveikti Chamfer atstumo įverčius. Dėl sumažėjusio taškų tankumo atsiranda didesni atstumai tarp paviršiuje esančių taškų, o tai turi įtakos bendram vidutinio atstumo apskaičiavimui.

Dice ir Jaccard metrikų jautrumą vokselių skaičiui galima paaiškinti, abiejų metrikų vertinamu dviejų vokselių tinklelių persidengimu arba panašumu, atsižvelgiant į jų atitinkamų vokselių rinkinių sankirtą ar sąjungą. Sumažėjus vokselių skaičiui 3D modelyje, sankirtos ir sąjungos rinkinių dydžiai keičiasi, o tai lemia panašumo įverčių pokyčius. Tad skirtingas 3D modelių vokselių skaičius, gali turėti reikšmingą įtaką šių modelių panašumo vertinimo rezultatams.



18 pav. Dice, Jaccard, Hausdorff atstumo ir Chamfer atstumo įverčių pokytis kintant vokselių skaičiui modelyje.

Nors sumažėjęs vokselių skaičius sąlygoja prarandamą modelio erdvinę informaciją, Hausdorff atstumo apskaičiavimas sėkmingai užfiksuoja bendrą trimačių objektų panašumą ir formą.

3.7. 3D vokselių tinklelio dydžio įtaka 3D modelių panašumo vertinime

Vokselių tinklelio dydis turi reikšmingos įtakos metrikų algoritmų vykdymo laikui. Vokselių tinklelio dydis lemia trimačio objekto detalumą bei vokselių kiekį pačiame 3D modelyje. Trimačiai vokselių tinkleliai pasižymi saugomais tiek tuščiais tiek užimtais vokseliais, ko pasekoje nustatant skirtumą ar panašumą tarp dviejų vokselių tinklelių iteruojami ir tikrinami visi tinklelyje esantys vokseliai.

Vokselių tinklelio dydžio įtakos Dice, Jaccard, Hausdorff ir Chamfer atstumo algoritmų vykdymo laikui nustatymo metu vertinami ModelNet40 [WSK⁺15] ir 3D-GAN sugeneruoti egzemplioriai. Vykdymo laikui išmatuoti naudojama *Pytorch timeit* biblioteka. Skaičiuojamas 3D modelių panašumas Dice, Jaccard, Hausdorff ir Chamfer atstumo metrikomis bei vertinamas šių algoritmų atlikimo laikas kintant vokselių tinklelio dydžiui. Tyrimo metu vertinami 30³, 40³ ir 60³ tinklelių dydžiai. Prieš vertinimą, kiekvienas 3D modelis normalizuojamas į vertinamą tinklelio dydį ir skai-

čiuojami šių modelių panašumo koeficientai lyginant su vienu ir tuo pačiu sugeneruotu 3D-GAN egzemplioriumi (22 pav.).

12 lentelė. Algoritmų vykdymo laikas 3D-GAN sugeneruoto modelio (A) 22 pav. ir modelio (B) 23 pav. palyginime.

Rezoliucija	Vokselių skaičius A	Vokselių skaičius B	Dice (s)	Jaccard (s)	Hausdorff atstumas (s)	Chamfer atstumas (s)
30 ³	511	932	0,00016	0,00019	0,00198	0,00682
40 ³	1205	2218	0,00024	0,00025	0,00493	0,03837
60 ³	4088	7456	0,00076	0,00075	0,01956	0,40361

13 lentelė. Algoritmų vykdymo laikas 3D-GAN sugeneruoto modelio (A) 22 pav. ir modelio (C) 24 pav. palyginime.

Rezoliucija	Vokselių skaičius A	Vokselių skaičius C	Dice (s)	Jaccard (s)	Hausdorff atstumas (s)	Chamfer atstumas (s)
30 ³	511	725	0,00017	0,00020	0,00183	0,00562
40 ³	1205	1842	0,00028	0,00029	0,00489	0,03286
60 ³	4088	5939	0,00071	0,00085	0,02288	0,33341

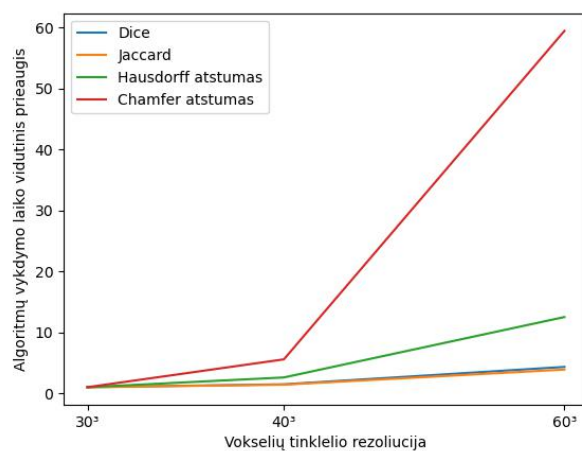
14 lentelė. Algoritmų vykdymo laikas 3D-GAN sugeneruoto modelio (A) 22 pav. ir modelio (D) 25 pav. palyginime.

Rezoliucija	Užimtų vokselių skaičius A	Užimtų vokselių skaičius D	Dice (s)	Jaccard (s)	Hausdorff atstumas (s)	Chamfer atstumas (s)
30 ³	511	1207	0,00017	0,00022	0,00259	0,00872
40 ³	1205	2673	0,00023	0,00032	0,00658	0,04568
60 ³	4088	9528	0,00072	0,00075	0,03785	0,52298

Lentelėse (12, 13 ir 14) minima rezoliucija nurodo vokselių tinklelio tiriamą dydį, taip pat nurodomas tik užimtų vokselių skaičius kiekvieno trimačio modelio vokselių tinklelyje kintant tinklelio dydžiui, apskaičiuoti įverčiai nusako kiekvienos metrikos algoritmo vykdymo laiką išreikštą sekundėmis.

Galima patsebt, kad Chamfer atstumo ir Hausdorff atstumo apskaičiavimas užtrunka ilgiau nei Dice ar Jaccard koeficientų apskaičiavimas. Skaičiuojant atstumus tarp trimačių modelių, algoritmas turi palyginti atstumą tarp kiekvieno vieno vokselių tinklelio vokselio su artimiausiu vokseliu kitame vokselių tinklelyje Hausdorff atveju ir vidutinį atstumą nuo kiekvieno taško viename rinkinyje iki artimiausio taško kitame rinkinyje Chamfer atveju. Chamfer atstumo apskaičiavimo vykdymo laikui įtakos galimai turi papildomos vidutinio atstumo suradimo ir artimiausio kaimyninio taško paieškos operacijos. Kita vertus, Hausdorff atstumo skaičiavimas apima didžiausio atstumo

tarp dviejų taškų rinkinių, paprastai vokselių tinklelio paviršiaus taškų, nustatymą. Nors norint nustatyti maksimalų atstumą, vis tiek reikalinga tikrinti rinkinio taškus ir atlikti atstumo skaičiavimus, pagrindinis dėmesys skiriamas didžiausio atstumo nustatymui, kuris gali būti paprastesnis ir galbūt labiau tiesioginis nei Chamfer atstumo atveju. Didėjant vokselių tinkleliui, didėja vokselių, kuriuos reikia palyginti skaičius. Dėl šios padidėjusios skaičiavimo apkrovos Hausdorff atstumo ir Chamfer atstumo algoritmų vykdymo laikas pailgėja. Kita vertus, tokių metrikų kaip Jaccard ir Dice skaičiavimas apima vokselių tinklelių sankirtos ir sąjungos identifikavimą. Nors šiuos algoritmus taip pat veikia vokselių tinklelio dydis, jų vykdymo laikas paprastai yra mažiau paveiktas, palyginti su atstumo metrikomis, kadangi vokselių tinklelio dydis daugiausia veikia bendrą vokselių skaičių ir jų persidengimą, o ne reikalauja atstumo nuo taško iki taško skaičiavimų.



19 pav. Dice, Jaccard, Hausdorff ir Chamfer atstumo metrikų palyginimas kintant vokselių tinklelio dydžiui. Algoritmų pradinis laikas esantis 30^3 vokselių tinklelio rezoliucijai laikomas vienetu.

Taikant panašumo palyginimo metrikas, pastebima, kad Chamfer atstumo matavimas užtrunka ilgiausią laiką, lyginant su kitomis panašumo nustatymo metrikomis, algoritmo vykdymo laikas esant 60^3 vokselių tinklelio dydžiui išauga ≈ 59 karto nuo pradinio algoritmo vykdymo laiko esant 30^3 vokselių tinklelio dydžiui. Taip pat išauga ir Hausdorff atstumo nustatymo algoritmo vykdymo laikas, kuris padidėja ≈ 12 kartų nuo pradinio algoritmo vykdymo laiko. Dice ir Jaccard panašumo koeficientų apskaičiavimo vykdymo laikas yra trumpiausias. Renkantis atitinkamą vokselių tinklelio dydį, svarbu atsižvelgti į skaičiavimo laiko nuostolį.

3.8. 3D modelių tikroviškumo įvertinimas

Vokselių tinklelio nuoseklumo nustatymas yra esminis žingsnis vertinant vokseliais pagrįstų modelių tikroviškumą ir kokybę. Realaus pasaulio objektai paprastai pasižymi detalio tekstūra, yra pastovaus dydžio ir nuoseklios struktūros, turi šešėlių ar vizualų vaizdo gylį. Tyrimo metu pasirinkta nustatyti vokselių tinklelio jungumą. Šio kriterijaus pagalba siekiama kiekybiškai įvertinti 3D-GAN sukurtų trimačių modelių tikroviškumą. Nors tokio kriterijaus pasirinkimas pilnai nepadengia visų 3D modelių realistiškumo vertinimo aspektų, trimačio vokselių tinklelio jungumo

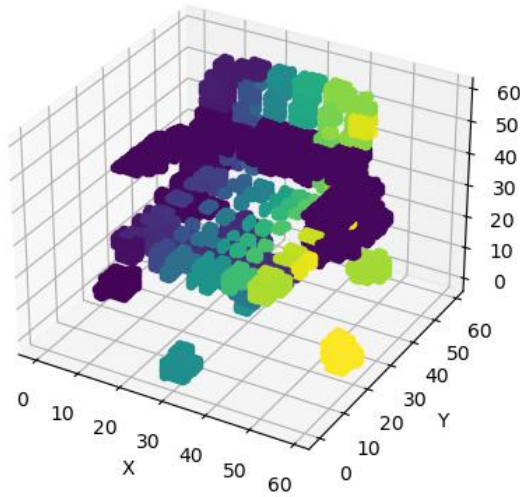
vertinimas gali suteikti vertingų įžvalgų apie modelio ir realaus pasaulio objektų atitikimą bei kokybę.

Jungumo palyginimo metu tiriami ModelNet40 [WSK⁺15] ir 3D-GAN sugeneruoti kėdžių modeliai, skaičiuojamas modelio laisvų komponentų skaičius. Komponentų kiekis (15 lentelė) nurodo kiek atskirų vokselių klasterių sudaro 3D vokselių tinklelio modelį. Klasterių skaičiui apskaičiuoti panaudota *scipy.ndimage* bibliotekos funkcija *label*. Funkcija nustato ryšio schemą, kuri naudojama sujungtiems komponentams identifikuoti. Ryšys reiškia taisyklę, apibrėžiančią, kaip gretimi elementai laikomi sujungtais. Pavyzdžiui 2D kubelių jungumas atsižvelgia į artimiausius kaimynus (viršuje, apačioje, kairėje ir dešinėje).

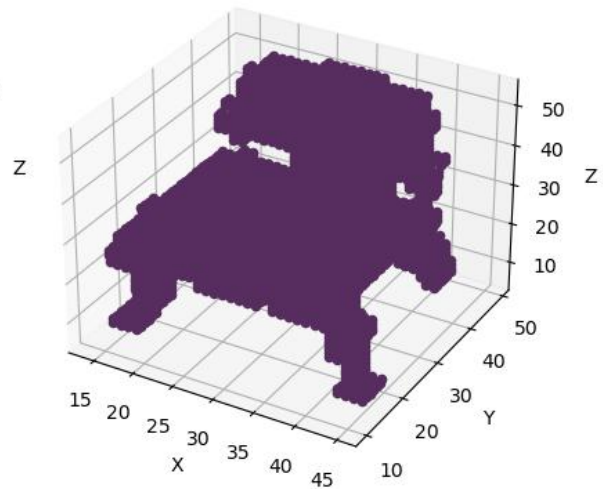
15 lentelė. ModelNet40 [WSK⁺15] ir 3D-GAN sugeneruotų kėdžių modelių nuoseklumo palyginimas 21 pav.

3D modelio numeris	Klasterių kiekis
110	39
111	49
112	3
113	72
114	55
115	5
116	19
117	97
118	64
119	1

Nuoseklumo įvertinimo metu matoma (15 lentelė), kad nuosekliausi modeliai yra 119, 112 ir 115, kur 119 trimačio modelio visi esantys vokseliai liečiasi su greta esančiais kitais vokseliais sudarydami pilnai sujungtą, nuoseklų trimatį objektą. Tuo tarpu 112 trimatį vokselių tinklelį sudaro 3 klasteriai ir 115 – 5 klasteriai. Trimatis modelis 117 įvertintas kaip labiausiai nenuoseklus rinkinio 3D modelis, šį modelį sudaro 97 atskiri komponentai (20 pav.).



(a) Klasterių skaičius: 97



(b) Klasterių skaičius: 1

20 pav. ModelNet40 [WSK⁺15] ir 3D-GAN sugeneruotų kėdžių klasterių analizė.

Remiantis šiais duomenimis, galima būtų teigti, kad 119 yra tikroviškiausiai atrodantis trimatis modelis, kadangi modelis yra nuoseklus ir visi modelio komponentai yra sujungti. Vokselių nuoseklumo įvertinimas yra būtinas norint analizuoti erdvinius ryšius bei nustatyti ar 3D modelis yra pilnai sujungtas. Nuoseklumo įvertinimo metu išgaunama vertinga informacija apie trimačio objekto ribas bei realistiškumą, kadangi 3D modeliai sudaryti iš pavienių, nesiribojančių komponentų realiomis sąlygomis negalėtų išlaikyti savo formos.

3.9. Vidutinio 3D modelių panašumo įvertinimas duomenų aibės atžvilgiu

Vidutinio sugeneruotų 3D modelių ir apmokymo duomenų rinkinio panašumo įvertinimas yra vienas iš žingsnių nustatant generuojančio modelio veikimo kokybę. Išmatuojant sugeneruotų modelių ir duomenų rinkinio panašumo laipsnį, galima nustatyti, kaip gerai modelis išmoko reprezentuoti pagrindinį duomenų pasiskirstymą. Vidutinio panašumo nustatymo metu vertinami ModelNet40 [WSK⁺15] ir 3D-GAN sugeneruoti kėdžių modeliai. 3D modeliai lyginami tarpusavyje apskaičiuojant vidutinį kiekvieno 3D modelio ir likusio duomenų rinkinio (21 pav.) panašumą remiantis Dice, Jaccard, Hausdorff atstumo ir Chamfer atstumo metrikomis. Tyrimo metu nuspręsta nebetęsti vertinimo Hausdorff atstumu remiantis trimačių modelių 2D pjūviais, kadangi toks 3D modelių panašumo vertinimas yra netikslus.

16 lentelė. ModelNet40 [WSK⁺15] ir 3D-GAN sugeneruotų kėdžių modelių vidutinis Hausdorff atstumas 21 pav.

3D modelio numeris	Dice	Jaccard	Hausdorff atstumas	Chamfer atstumas
110	0,09	0,04	21,30	5,11
111	0,22	0,17	17,66	3,85
112	0,25	0,20	18,76	4,09
113	0,24	0,19	19,03	4,41
114	0,21	0,16	18,12	4,07
115	0,18	0,15	21,15	5,06
116	0,22	0,17	20,20	4,33
117	0,21	0,17	17,72	3,90
118	0,15	0,13	23,07	6,61
119	0,21	0,17	18,39	4,43

Remiantis aukščiau pateiktais rezultatais (16 lentelė) pagal Dice ir Jaccard koeficientų balus, 112 ir 113 trimačiai modeliai yra vieni panašiausių tiriama duomenų rinkinio atžvilgiu. Šių modelių panašumas siekia $\approx 0,25$ pagal Dice koeficientą ir $\approx 0,20$ pagal Jaccard indeksą. Tuo tarpu gauti rezultatai vertinant 3D modelių vidutinį panašumą taikant tiesioginio Hausdorff atstumo ir Chamfer atstumo apskaičiavimą rodo, kad panašiausi 3D modeliai yra 111 ir 117, kurių vidutinis atstumas lyginant visą duomenų rinkinį siekia ≈ 18 vertinant Hausdorff atstumu ir ≈ 4 Chamfer atstumu. Taip pat galima pastebėti, kad daugelio kitų trimačių modelių panašumas nėra labai nutolęs ir nemaža dalis trimačių modelių tarpusavyje yra panašūs. Vertinant modelių vidutinį panašumą duomenų rinkinio atžvilgiu svarbu išmatuoti ir sukurtų 3D modelių įvairovės laipsnį bei užtikrinti, kad įvairovės laipsnis būtų išlaikomas modelių generavimo proceso metu. Nors labai panašus sugeneruotų modelių rinkinys gali sudaryti geros generatoriaus kokybės įvaizdį, tai taip pat gali reikšti, kad neuroninių tinklų modelis išmoko atkartoti apmokymo duomenis ir nebegali generuoti naujų ir įvairių pavyzdžių. Tad norint įvertinti, ar DL modelis yra efektyvus ir kokybiškas būtina nustatyti ar toks modelis geba apibendrinti nematytus pavyzdžius išlaikant tam tikrą tokių duomenų įvairovės laipsnį.

4. Rezultatai

Sudėtinga 3D formų prigimtis ir bendrų vertinimo kriterijų trūkumas apsunkina trimačių duomenų taikymą generavimo kontekste, tad dauguma vertinimo metodų remiasi vizualiu rezultatų tikrinimu ir priklauso nuo kitų klasifikavimo modelių. Generatyviniais besivaržančiais tinklais sukurtų 3D modelių tikslumo ir tikroviškumo tyrimo metu atlikta analizė DL modelių darbo su trimatėmis formomis srityje, nagrinėjant dažniausiai naudojamą 3D modelių reprezentacijos formas, metodus skirtus 3D duomenų apdorojimui prieš jų pateikimą neuroninių tinklų mokymui bei apžvelgtos esamos neuroninių tinklų architektūros skirtos darbui su trimatėmis duomenimis. Apžvelgti dažniausiai naudojami 3D-GAN tinklo kokybės įvertinimo metodai kaip: KDE, IS, FID, *Alpha precision and recall* bei Chamfer atstumas.

Tyrimo metu atlikti eksperimentai naudojantis supaprastinta 3D-GAN [WZX⁺16] tinklo architektūra ir ModelNet [VGV⁺09] bei ModelNet40 [WSK⁺15] apmokymo duomenų rinkiniais. 3D-GAN tinklo generuojami trimačiai modeliai perteikti į 32^3 dydžio vokselių tinklą, atliktas trimačių modelių vokselių tinklo dydžio normalizavimas, keičiant tinklo dydį ir nustatant dydžio pokyčio įtaką panašumo vertinimo įverčiams. Tikslumo ir 3D modelių tikroviškumo nustatymo metu taikyti: Dice koeficientas [GS16], Jaccard indeksas [NSN⁺13], Hausdorff atstumas [HKR93] bei Chamfer atstumas [WPZ⁺21]. Iškeltos esminės problemos susijusios su 3D modelių palyginimu ir kokybės nustatymu: platus formų stiliaus pasirinkimas, dydžio ar pozicijos trimačiame vokselių tinklelyje skirtumai, 3D vokselių tinklo tankumo skirtumai bei vokselių tinklo dydžio įtaka panašumo metrikų algoritmų vykdymo laikui. Paminėti aspektai veikia galutinius 3D-GAN sukurtų formų panašumo, realumo ir kokybės įverčius.

Dalis tyrimo rezultatų pristatyti konferencijoje „Lietuvos magistrantų informatikos ir IT tyrimai“ 2023 m., taip pat rezultatai publikuoti konferencijos medžiagoje [JDK⁺23].

4.1. Išvados

1. Tyrimo rezultatai rodo, kad Hausdorff atstumo matavimas, remiantis trimačio modelio 2D pjūviais, nepasiteisino ir nėra rekomenduojamas kaip tinkamas trimačių formų panašumo įvertinimo metodas. Šis metodas lemia bendros trimatės formos erdvinės informacijos praradimą ir demonstruoja nepakankamą tikslumą lyginant su kitais panašumo įvertinimo matais, tokiais kaip Dice koeficientas, Jaccard indeksas, 3D vokselių tinklo tarpusavio Hausdorff atstumu palyginimas bei Chamfer atstumas.
2. Taikant panašumo įvertinimo metrikas, svarbu atsižvelgti į algoritmo vykdymo laiką renkantis reikiamą vokseliais grįsto trimačio tinklo dydį. Pastebėta, kad Chamfer atstumo apskaičiavimas užtrunka ilgiausiai lyginant su Dice koeficiento, Jaccard indekso bei Hausdorff atstumo apskaičiavimu, kur Chamfer algoritmo vykdymo laikas esant 60^3 vokselių tinklo dydžiui išauga ≈ 59 kartus nuo pradinio algoritmo vykdymo laiko esant 30^3 vokselių tinklo dydžiui. Vertinant tokį algoritmo vykdymo laiko nuostolį, vokseliais grįstų, aukštos raiškos trimačių tinklo panašumo vertinimas Chamfer atstumu nėra rekomenduojamas.

3. Atsižvelgiant į mokymo duomenų rinkinio ir 3D-GAN sugeneruoto trimačio modelio vidutinį Hausdorff atstumą, kartu kombinuojant sugeneruoto trimačio modelio vokselių tinklelio nuoseklumo nustatymą, galėtų būti perspektyvus sprendimas vertinant 3D-GAN sugeneruoto trimačio modelio tikroviškumą ir panašumą duomenų rinkinio atžvilgiu.

Remiantis 3D-GAN sukurtų trimačių modelių kokybės bei realumo tyrimo rezultatais, vokseliais grįstų trimačių modelių suspaudimas į *octree* medžio struktūrą ir šių 3D modelių panašumo bei tikroviškumo tyrimas galėtų būti tolimesnė tyrimo dalis. *Octree* medžio struktūros panaudojimas trimačių modelių palyginime galėtų būti perspektyvus sprendimas norint sumažinti skaičiavimų sąnaudas ir algoritmų vykdymo laiką augant vokselių tinklelio dydžiui. Skaičiavimo sąnaudų sumažinimas leistų apsvarstyti Chamfer atstumo pritaikymą vidutinio trimačių modelių panašumo, duomenų rinkinio atžvilgiu, palyginime.

Literatūra

- [BLR⁺16] Andrew Brock, Theodore Lim, James M Ritchie ir Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016.
- [BMP02] S. Belongie, J. Malik ir J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002. DOI: 10.1109/34.993558.
- [CF20] Min Jin Chong ir David Forsyth. Effectively Unbiased FID and Inception Score and Where to Find Them. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020-06.
- [CFG⁺15] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan ir k.t. ShapeNet: An Information-Rich 3D Model Repository. Tech. atask. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.
- [ELE⁺17] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny ir Marian Mazzone. Can: Creative adversarial networks, generating” art” by learning about styles and deviating from style norms. *arXiv preprint arXiv:1706.07068*, 2017.
- [GPM⁺14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville ir Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.
- [GS16] Alonso Gragera ir Vorapong Suppakitpaisarn. Semimetric properties of sørensen-dice and tversky indexes. *WALCOM: Algorithms and Computation: 10th International Workshop, WALCOM 2016, Kathmandu, Nepal, March 29–31, 2016, Proceedings 10*, p.p. 339–350. Springer, 2016.
- [GWK⁺18] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma ir k.t. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.
- [HKR93] Daniel P Huttenlocher, Gregory A. Klanderman ir William J Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863, 1993.
- [HRU⁺17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler ir Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [HZR⁺16] Kaiming He, Xiangyu Zhang, Shaoqing Ren ir Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p.p. 770–778, 2016.

- [ICN⁺17] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos ir Ioannis Kompatsiaris. Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys (CSUR)*, 50(2):1–38, 2017.
- [JDK⁺23] Bernatavičienė Jolita, Gintautas Dzemyda, Olga Kurasova ir Julius Žilinskas. Proceedings of the Conference quot;Lithuanian MSc Research in Informatics and ICTquot; *Vilnius University Open Series*:1–5, 2023-05. URL: <https://www.journals.vu.lt/open-series/article/view/32211>.
- [KSH17] Alex Krizhevsky, Ilya Sutskever ir Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM*, 60(6):84–90, 2017-05. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: <https://doi.org/10.1145/3065386>.
- [LBD⁺90] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard ir Lawrence Jackel. Handwritten Digit Recognition with a Back-Propagation Network. D. Touretzky, redaktorius, *Advances in Neural Information Processing Systems*, tom. 2. Morgan-Kaufmann, 1990. URL: <https://proceedings.neurips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf>.
- [LZW⁺19] Haisheng Li, Yanping Zheng, Xiaoqun Wu ir Qiang Cai. 3D model generation and reconstruction using conditional generative adversarial network. *International Journal of Computational Intelligence Systems*, 12(2):697–705, 2019.
- [MS15] Daniel Maturana ir Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p.p. 922–928. IEEE, 2015.
- [NSN⁺13] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn ir Supachanun Wanapu. Using of Jaccard coefficient for keywords similarity. *Proceedings of the international multiconference of engineers and computer scientists*, tom. 1 numeris 6, p.p. 380–384, 2013.
- [QSM⁺17] Charles R Qi, Hao Su, Kaichun Mo ir Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p.p. 652–660, 2017.
- [RMC16] Alec Radford, Luke Metz ir Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2016. arXiv: 1511.06434 [cs.LG].
- [SBL⁺18] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet ir Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.
- [SGZ⁺16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford ir Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

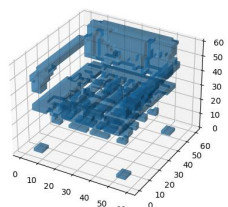
- [SLJ⁺14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke ir Andrew Rabinovich. Going Deeper with Convolutions. *CoRR*, abs/1409.4842, 2014. arXiv: 1409 . 4842. URL: <http://arxiv.org/abs/1409.4842>.
- [SLJ⁺15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke ir Andrew Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p.p. 1–9, 2015.
- [SMK⁺15] Hang Su, Subhransu Maji, Evangelos Kalogerakis ir Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. *Proceedings of the IEEE international conference on computer vision*, p.p. 945–953, 2015.
- [SX16] Shuran Song ir Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p.p. 808–816, 2016.
- [SZ14] Karen Simonyan ir Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [TH15] Abdel Aziz Taha ir Allan Hanbury. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1):1–28, 2015.
- [VGV⁺09] Kashi Venkatesh Vishwanath, Diwaker Gupta, Amin Vahdat ir Ken Yocum. Modelnet: Towards a datacenter emulation environment. *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*, p.p. 81–82. IEEE, 2009.
- [WLG⁺17] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun ir Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)*, 36(4):1–11, 2017.
- [WPZ⁺21] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu ir Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. *arXiv preprint arXiv:2111.12702*, 2021.
- [WSK⁺15] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang ir Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p.p. 1912–1920, 2015.
- [WSS06] Wai-Tak Wong, Frank Y Shih ir Te-Feng Su. Thinning algorithms based on quadtree and octree representations. *Information Sciences*, 176(10):1379–1394, 2006.
- [WZX⁺16] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman ir Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in Neural Information Processing Systems*, p.p. 82–90, 2016.

- [ZF14] Matthew D Zeiler ir Rob Fergus. Visualizing and understanding convolutional networks. *European conference on computer vision*, p.p. 818–833. Springer, 2014.
- [ZWB⁺16] Zhuotun Zhu, Xinggang Wang, Song Bai, Cong Yao ir Xiang Bai. Deep learning representation using autoencoder for 3D shape retrieval. *Neurocomputing*, 204:41–50, 2016.

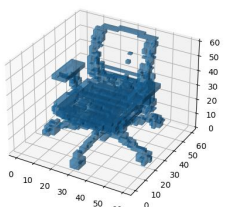
Priedai

Kiekybinis 3D modelių panašumo metrikomis

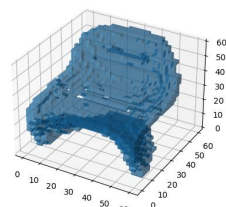
Kiekybinio trimačių modelių panašumo metrikomis palyginimo metu naudoti ModelNet40 [WSK⁺15] ir 3D-GAN sugeneruoti kėdžių modeliai.



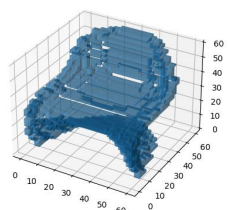
110



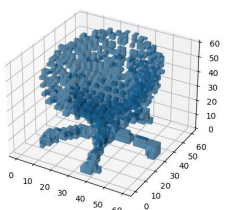
111



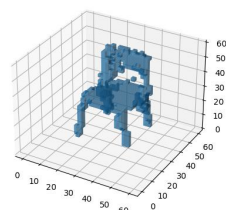
112



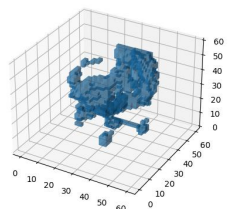
113



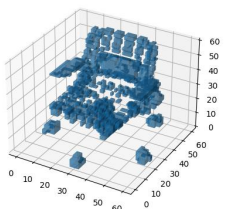
114



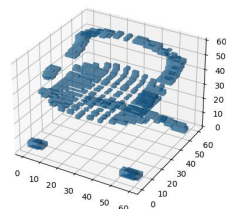
115



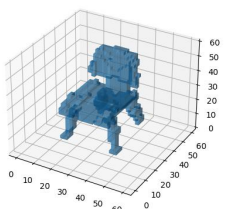
116



117



118

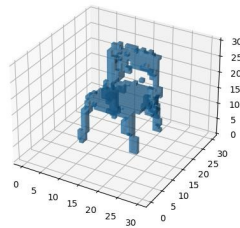


119

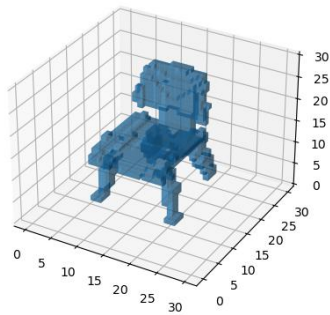
21 pav. ModelNet40 [WSK⁺15] ir 3D-GAN sugeneruoti kėdžių modeliai, kur 110 – 119 žymi 3D modelio numerį.

Vokselių tinklelio dydžio įtaka algoritmų vykdymo laikui

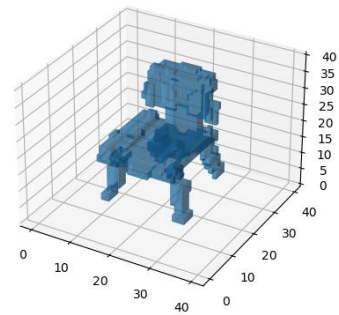
3D modeliai, naudoti vokselių tinklelio dydžio ir algoritmo vykdymo laiko tyrimo metu.



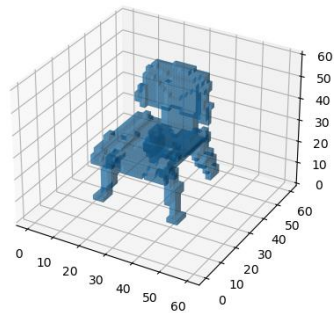
22 pav. 3D-GAN sugeneruotas lyginamas modelis.



30^3 vokselių tinklelio rezoliucija

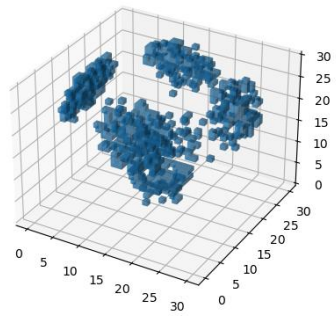


40^3 vokselių tinklelio rezoliucija

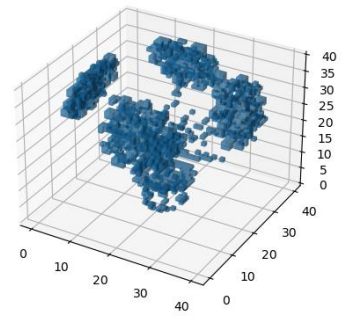


60^3 vokselių tinklelio rezoliucija

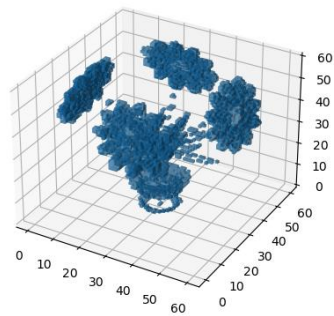
23 pav. 3D-GAN sugeneruotas modelis skirtingomis rezoliucijomis.



30^3 vokselių tinklelio rezoliucija

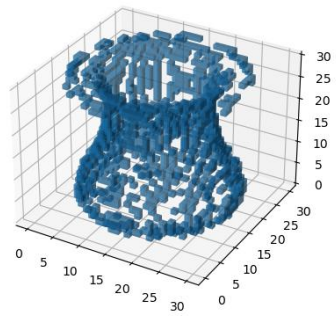


40^3 vokselių tinklelio rezoliucija

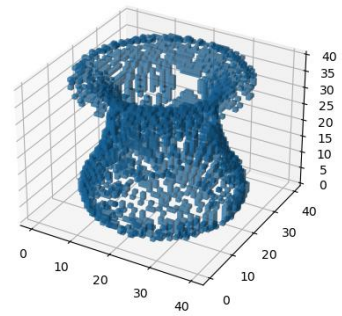


60^3 vokselių tinklelio rezoliucija

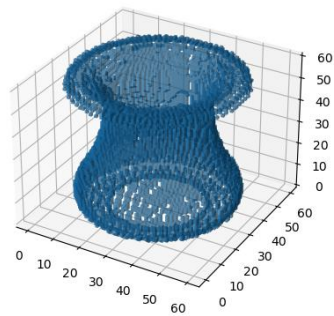
24 pav. Modelis iš ModelNet40 [WSK⁺15] apmokymo duomenų rinkinio skirtingomis rezoliucijomis.



30^3 vokselių tinklelio rezoliucija



40^3 vokselių tinklelio rezoliucija



60^3 vokselių tinklelio rezoliucija

25 pav. Modelis iš ModelNet40 [WSK⁺15] apmokymo duomenų rinkinio skirtingomis rezoliucijomis.