

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS KATEDRA

Baigiamasis bakalauro darbas

**Matomos geometrijos išankstinio skaičiavimo  
optimizacijos**

Atliko: 4 kurso 9 grupės studentas  
Vytautas Šaltenis .....  
(parašas)

Darbo vadovas:  
dr. Rimvydas Krasauskas.....  
(parašas)

Vilnius  
2007

## Turinys

Įvadas .....	2
1. Literatūros apžvalga .....	3
2. Tikslus matomumo nustatymas iš zonos .....	6
2.1. Plückerio erdvė.....	6
2.2. Nirensteino metodas .....	7
2.3. Matomumo užklausa.....	7
2.4. Tiesių erdvė tarp dviejų poligonų.....	8
2.4.1. CSG Plückerio erdvėje .....	9
2.5. Nirensteino siūlytos optimizavimo strategijos.....	9
2.5.1. Menamų tiesių ignoravimas .....	9
2.5.2. Trivialus pripažinimas .....	9
2.5.3. Matomumą užstojančių objektų prioritetai .....	11
2.5.4. Užklausa grupėms .....	11
2.5.5. Pakartotinis gaubiančiosios figūros matomumo išnaudojimas ...	11
2.5.6. Virtualūs vaizdo blokai.....	12
3. Optimizacijų realizacija .....	13
3.1. Hierarchijos įvedimas .....	13
3.2. Konstravimas .....	13
3.3. Paieška: Dvimatis atvejis .....	14
3.4. Paieška: Trimatis atvejis .....	14
3.4.1. Gaubiančioji sfera .....	17
3.4.2. Gaubiantysis kuboidas su ašimis statmenomis sienomis .....	17
3.4.3. Orientuotas gaubiantysis kuboidas .....	18
3.4.4. Orientuotas gaubiantysis cilindras.....	18
4. Siūlomo metodo palyginimas su ankstesniais darbais .....	19
4.1. Menamų tiesių ignoravimas .....	19
4.2. Trivialus pripažinimas .....	19
4.3. Matomumą užstojančių objektų prioritetai .....	20
4.4. Užklausa grupėms .....	20
4.5. Pakartotinis gaubiančiosios figūros matomumo išnaudojimas .....	20
4.6. Virtualūs vaizdo blokai.....	21
Išvados, rezultatai ir tolimesnis darbas .....	22
Summary .....	24

## Įvadas

Kompiuterinėje grafikoje nuolat aktuali problema yra norimos pavaizduoti geometrijos kiekis. Spartėjant kompiuteriams, vis daugiau jos galima pavaizduoti, bet tuo pačiu didėja ir vartotojų lūkesčiai, tad kol kas negalima kalbėti apie kiek norima didelio geometrijos kiekio pavaizdavimą.

Mažinant apdorojamos geometrijos kiekį, vienas iš pagrindinių metodų yra rasti patikimus būdus visiškai ignoruoti geometriją, kuri yra neaktuali konkrečiai paimtai scenai ar kadru.

Vienas iš ilgą laiką taikomų ir pasiteisinusių būdų yra iš anksto apskaičiuoti, kiek ir kokios geometrijos gali būti matoma stebėtoju nepaliekant tam tikro sektoriaus ribų. Stipriausia šio metodo pusė yra ta, kad turint tokią informaciją, yra dirbama *tik* su geometrija, kuri gali būti pamatyta "čia ir dabar" arba artimiausiu metu iš kažkurios gretimos pozicijos. Pagrindiniai metodo trūkumai yra du:

1. Toks metodas gali būti taikomas tik statinėms arba beveik statinėms<sup>1</sup> scenoms;
2. Bandant realizuoti, susiduriama su labai dideliais komputaciniais sunkumais, kai kurie autoriai netgi kėlė klausimą, ar egzistuoja praktiškas bendrojo atvejo sprendimas [Pla92].

Šio darbo tikslas yra išplėsti kai kuriuos algoritmus, naudojamus išankstiniam matomos geometrijos apskaičiavimui. Koncentruojamasi ties egzistuojančių algoritmų paspartinimu, siūlant optimalesnius būdus gauti tam pačiam rezultatui.

Pirmiausia šiame darbe pristatomas egzistuojančių sprendimų karkasas, aptariamoms problemoms, kurias išsprendė ankstesni autoriai. Sudaromos gairės galimoms optimizacijoms ir patobulinimams.

Toliau pateikiamas šiame darbe realizuotas sprendimas, aprašomi ir pagrindžiami pasirinkti algoritmai, sprendimai, technologijos. Pabrėžiami skirtumai lyginant su ankstesnių autorių darbais.

---

<sup>1</sup>"Beveik statinės" yra tokios scenos, kuriose laikui bėgant matomumo informacija keičiasi labai mažai arba labai dėsningai

## 1. Literatūros apžvalga

Matomumas kompiuterinėje grafikoje gali būti nustatomas daugeliu būdų, skirtingais tikslais. Pavyzdžiui, iki grafikos spartintuvų atsiradimo, kurie leido *de facto* naudoti Z-buferį [FDF+90], buvo aktualu nustatyti tikslią objektų tarpusavio padėtį erdvėje. Priešingu atveju neįmanoma pavaizduoti bent kiek sudėtingesnių scenų: reikia žinoti, kuris objektas kurį užstoja, kad toliau esantis objektas neužstotų artimesniojo.

Šį poreikį praktiškai eliminavo grafikos spartintuvai, kurie sugeba vaizduoti geometriją su Z-buferiu ar be jo vienodu greičiu. Atsirado poreikis globalesniam požiūriui į matomumo nustatymą. Matomumo nustatymo algoritmai gali būti apibūdinami ir tarpusavyje lyginami pagal keletą kriterijų:

- Matomumas iš taško (From-Point visibility) priešpastatomas matomumui iš zonos (From-Region visibility).

Pirmieji algoritmai skaičiuoja matomą geometriją iš konkrečiai paimto taško. Natūralu, kad dažniausiai tai yra taškas, kuriame yra stebėtojas konkrečiu momentu [LG95]. Tuo tarpu matomumas iš zonos skaičiuojamas kažkokiai apribotai erdvei (zonai), kuri suskaičiuojama vieną kartą ir galioja bet kuriai stebėtojo pozicijai toje zonoje. Tai ir yra pagrindinė pastarųjų algoritmų stiprybė — kadangi matomumo informacija galioja daugeliui kadrų iš eilės, matomumo skaičiavimas yra amortizuojamas. Kitas privalumas yra galimybė numatyti artimiausius veiksmus. Pasinaudojant gretimos zonos matomumo informacija, galima iš anksto apdoroti reikiamus duomenis (siųsti matomumo informaciją per tinklą, krauti iš disko į atmintį ar pan). Pagrindiniai matomumo iš zonos trūkumai yra ypač sudėtingi išankstiniai skaičiavimai bei sunkus jų pritaikymas dinamiškoms scenoms.

- Sektoriai su portalais priešpastatomi bendresnėms scenoms.

Kai kurie algoritmai yra iš esmės pritaikyti architektūrinėms ir panašioms scenoms. Vaizduojant architektūrą iš vidaus, beveik visada yra galimybė išnaudoti natūralų scenos išskaidymą į sektorius (patalpos) bei portalus (langai, durys). Remiantis šiuo metodu, išnaudojamas faktas, kad didžioji vaizdo dalis yra užstojama artimiausių sienų, matomumas galimas tik pro sąlyginai nedidelius portalus. Tačiau ne visa norima vaizduoti geometrija gali būti lengvai suskirstyta tokiais sektoriais. Pavyzdžiui, norima apskaičiuoti matomumą iš zonos urbanistinėms, gamtinėms ir pan. scenoms, kuriose taip pat didžiąją

dalį geometrijos konkrečiame kadre užstoja artimesni objektai, tačiau scenoje nėra tokio natūralaus išskirstymo, koks yra architektūroje.

- Konservatyvumas priešpastatomas aproksimacijai.

Dauguma matomumą skaičiuojančių algoritmų yra konservatyvūs. Jie pervertina matomų objektų aibę vardan garantijos, kad gautas vaizdo kadras bus tikslus. Tuo tarpu apytiksliai metodai negarantuoja, kad gautas kadras bus tikslus [GSF99]. Jų tikslas yra gauti "pakankamai tikslų" vaizdą, kur "pakankamumas" gali būti apibrėžiamas labai skirtingai, bet dažniausiai remiasi vienokiu ar kitokiu objekto indėlio į galutinį kadrą įvertinimu [SLC+04] [NB04].

- Aproksimacijos tikslumas.

Kadangi dauguma algoritmų yra konservatyvūs, labai įdomus kriterijus yra metodo pervertinimo dydis. Smarkiai pervertinantis algoritmas gali sukurti tokią didelę potencialiai matomų objektų aibę, kurios nepavyks apdoroti realiu laiku vaizduojant kadrus.

Šiame darbe apsiribojama konservatyviais metodais skaičiuoti matomumą iš zonos bendroms scenoms. Išsamiau apie kitus metodus galima paskaityti kitų autorių sudarytose apžvalgose [CCS+03] [Dur99].

Pradžią matomumo nustatymui iš zonos davė Airey ir kt. [Air91] [ARB90], bei Tellerio darbai [TS91]. Šie autoriai daugiausiai koncentravosi ties architektūrinėmis scenomis. Viena iš to priežasčių yra ta, kad šis apsiribojimas leidžia susikurti eilę problemos supaprastinimų. Jie padėjo pagrindus daugeliui vėlesnių darbų, apibrėžė tokias dabar svarbias sąvokas kaip "potencialiai matoma aibė", "konservatyvus matomumo nustatymas" bei "tankiai užstotos scenos".

Matomumo iš zonos skaičiavimai bendresnėms scenoms, kaip buvo minėta, yra sudėtingesni, todėl praktiškų rezultatų ilgą laiką nebuvo. Problema yra ta, kad efektyvų rezultatą duoda tik tie matomumą užstojantys objektai, kurie yra didesni už norimą zonos dydį [NFL+99], o tai retai pasitaiko realiose situacijose.

Link norimo rezultato buvo žengta pritaikius keletą strategijų: blokų suliejimas (occluder fusion), išplėstas projekcijas (extended projections), virtualius blokus (virtual occluders).

Blokų suliejimą 2000-aisiais metais pasiūlė Schaufleris [SDD+00]. Metodo idėja yra pasirinkus matomumo bloką, pažymėti visą jo užstojamą erdvę kaip nematomą. Schaufleris tai atlieka iš anksto diskrečiai padalinęs erdvę. Wonka pasiūlė metodo modifikaciją, kur erdvė nėra diskretizuojama [WWS00].

Išplėstųjų projekcijų metodas labai panašus į blokų suliejimą. To paties rezultato siekiama kiek kitokiu būdu: išnaudojamas jungtinis keleto vaizdo blokų indėlis į užstojamą vaizdą. Tik vietoje užstotos erdvės sužymėjimo, čia blokai yra projektuojami ir dirbama su jų projekcijomis [DDT+00].

Virtualių blokų idėją iškėlė Koltunas *et al* [KCC00]. Tai dar vienas būdas sujungti keleto vaizdo blokų indėlių į scenos uždengtumą. Pagrindinė metodo idėja yra apskaičiuoti keletui greta esančių objektų vieną bendrą vaizdo bloką, kuris ir toliau bus naudojamas vaizdo uždengtumui skaičiuoti. Šiame darbe dar bus minimi virtualūs vaizdo blokai.

Nirensteinas pasiūlė būdą tiksliai analitiškai apskaičiuoti matomumą iš zonos. Smulkiau šis metodas aprašomas žemiau, nes juo daugiausia remiamasi šiame darbe.

Po Nirensteino publikacijos, 2005-aisiais, Laine'as pasiūlė kryptinės matomumo aibės (directed visible set) sąvoką [Lai05]. Jo metodas pagerina skaičiavimus, geriau išnaudodamas erdvės vientisumą: vieną kartą aptikus, kad objektas blokuoja vaizdą, ši informacija propaguojama toliau per erdvę.

## 2. Tikslus matomumo nustatymas iš zonos

Šiame skyriuje pateikiamas metodas, pirmą kartą publikuotas Nirensteino ir kitų [NBG02] ir vėliau išplėtotas Nirensteino disertacijoje [Nir03]. Metodas sudaro pagrindą šiam darbui, juo remiamasi kuriant etaloninį šio darbo metodą, su kuriuo lyginamos modifikuotos versijos.

Iki pat 2002-aisiais metais publikuoto Nirensteino darbo visus bandymus sukurti Plückerio projekcinės erdvės reprezentacijomis besiremiantį matomumo nustatymą galima suskirstyti į dvi pagrindines grupes: algoritmai, sprendžiantys problemą analitiškai ir algoritmai, vienaip ar kitaip diskretizuojantys sprendimų aibę ir sprendžiantys matomumo uždavinį "kvantuotai".

Bandymai spręsti analitiškai iš esmės baigdavosi nesėkmėmis: buvo susiduriama su problemomis, kurios kėlė sunkumų taikant algoritmą bendru atveju; jos buvo paliekamos ateities darbams.

Diskretizuojantys algoritmai veikė neblogai, bet dėl diskretizavimo jų negalima vadinti bendro atvejo sprendimu, tik aproksimacija.

2002-aisiais metais Nirensteinas ir kiti [NBG02] pagaliau išsprendė visas analitinio metodo problemas ir pasiūlė algoritmą, išsprendžiantį bendrąjį atvejį.

### 2.1. Plückerio erdvė

Šio darbo kontekste visi ar dauguma algoritmų vienoje ar kitoje stadijoje naudojami Plückerio projekcine erdve. Šiame poskyryje pateikiama teorija, susijusi su skaičiavimais Plückerio erdvėje.

1992 metais buvo pasiūlyta [Tel92], o 1999 metais išplėtotą [HT99] metodiką, paremta poligono kraštines sudarančių atkarpų išreiškimu Plückerio koordinatėmis.

Plückerio koordinatės yra patogus būdas išreikšti kryptinėms trimatės erdvės tiesėms. Kiekviena pora nesutampančių taškų  $p = (p_x, p_y, p_z)$  ir  $q = (q_x, q_y, q_z)$  apibrėžia tiesę  $l$  erdvėje  $\mathbb{R}^3$ . Ši tiesė atitinka šešetą koeficientų, vadinamų tiesės Plückerio koordinatėmis  $\Pi_l$ :

$$\Pi_l = (\pi_{l0}, \pi_{l1}, \pi_{l2}, \pi_{l3}, \pi_{l4}, \pi_{l5}). \quad (1)$$

Kiekviena Plückerio koordinatė yra žemiau pateiktos matricos 2x2 minoro determinantas:

$$\begin{pmatrix} p_x & p_y & p_z & 1 \\ q_x & q_y & q_z & 1 \end{pmatrix}.$$

Jei  $a$  ir  $b$  yra tiesės ir  $\Pi_a$ ,  $\Pi_b$  yra jas atitinkančios išraiškos Plückerio koordinatėmis, tuomet tų tiesių tarpusavio padėtį *side* ( $a$ ,  $b$ ) galima apibrėžti tokia išraiška:

$$\text{side}(a, b) = \Pi_a \odot \Pi_b = (\pi_{a0}\pi_{b4} + \pi_{a1}\pi_{b5} + \pi_{a2}\pi_{b3} + \pi_{a3}\pi_{b2} + \pi_{a4}\pi_{b0} + \pi_{a5}\pi_{b1}), \quad (2)$$

kuri lygi nuliui, jeigu tiesės kertasi arba yra lygiagrečios, o jeigu nelygi nuliui, tai ženklas nurodo, ar tiesės prasilenkia pagal laikrodžio rodyklę, ar prieš.

Remiantis tokiais skaičiavimais galima trivaliai patikrinti, ar duota tiesė kerta poligoną. Jeigu pavyksta rasti tiesę, kuri kerta du poligonus, bet nekerta nė vieno poligono, kuris gali užstoti vaizdą, reiškia tarp jų įmanomas matomumas.

## 2.2. Nirensteino metodas

Nirensteinas matomumo uždavinį suformulavo šitaip: turint du poligonus, tarp kurių reikia patikrinti matomumą (t. y. apskaičiuoti, ar egzistuoja tiesė tarp bent vieno taško poligone A ir bent vieno taško poligone B, kuri nekerta jokios kitos geometrijos — vaizdo matomumo bloką), visų įmanomų tiesių aibė yra išreiškiama analitiškai kaip briaunainis  $\mathbb{R}^5$  erdvėje. Panašiai išreiškiami ir vaizdą blokuojantys objektai. Tuomet atliekamos CSG<sup>2</sup> atimties operacijos, atimant iš matomumą reprezentuojančio objekto tuos, kurie reprezentuoja blokus. Jeigu atėmus visus blokus, iš matomumą reprezentuojančio objekto kažkas lieka, reiškia matomumą reprezentuojančių tiesių aibė nelieka tuščia ir matomumas yra galimas. Jeigu ne — negalimas.

Toliau šis metodas aprašomas smulkiau.

## 2.3. Matomumo užklausa

Matomumo užklausa (visibility query) vadiname algoritmų visumą, kuri iš pradinųjų duomenų — dviejų poligonų, kuriems skaičiuojamas matomumas, bei vaizdą blokuoti galinčių objektų scenoje — nulemia, ar egzistuoja bent viena tokia taškų pora (vienas poligone A, kitas — poligone B), tarp kurių nubrėžta tiesė nekirstų nė vieno vaizdą užstojančio objekto scenoje.

Turint mechanizmą, galintį atlikti tokią užklausa, galima skaičiuoti scenos matomumą iš zonos. Norima zona apibrėžiama kaip stačiakampis gretasienis (ar, esant

<sup>2</sup>Constructive Solid Geometry: operacijos su objektais erdvėje, panašios į aibių operacijas: konjunkcija, disjunkcija, atimtis ir pan.



poreikiui, koks nors kitas briaunainis). Tuomet scenos objektų matomumas yra nustatomas iš kiekvienos šios zonos sienos. Susumavus rezultatus, gaunama aibė objektų, kurie gali būti matomi iš šios zonos. Neretai tokia informacija vadinama PVS (Potentially Visible Set — potencialiai matoma aibė), bet šiame darbe šio termino vengiama, nes jis istoriškai labiau siejamas su konservatyvesniais metodais, kurie smarkiau pervertina *potencialiai* matomų objektų aibę.

## 2.4. Tiesių erdvė tarp dviejų poligonų

Kaip buvo minėta, matomumo uždavinys yra ne kas kita kaip paieška tarp visų galimų matomumo tiesių, ieškant bent vienos tokios, kuri nekirstų jokios kitos geometrijos. Pagrindinė problema, su kuria susiduriama sprendžiant tokį uždavinį yra reprezentacija. Matomumo tiesių tarp dviejų paviršių gali būti be galo daug, tad vienintelis galimas variantas yra analitinė reprezentacija. Jeigu tos tiesės reprezentuojamos tiesiogiai ant Plückerio hiperpaviršiaus, susiduriama su manipuliacijų sudėtingumu.

Vietoje to, poligonų kraštines reprezentuojantys  $\mathbb{P}^5$  taškai yra suprojektuojami į  $\mathbb{R}^5$ .  $\mathbb{P}^5$  taškas, anksčiau atitikęs  $\mathbb{R}^6$  plokštumą, dabar atitinka  $\mathbb{R}^5$  plokštumą ir iš šių plokštumų sukonstruojamas briaunainis  $\mathbb{R}^5$  erdvėje, kuris pilnai reprezentuoja aibę matomumo tiesių tarp dviejų dominančių poligonų.

Turint aibę plokštumų, tokia briaunainio reprezentacija vadinama puserdvine reprezentacija (H-representation, halfspace-representation). Tačiau parankiau būtų turėti viršūnių aibę. Tokia reprezentacija vadinama reprezentacija viršūnėmis (V-representation, vertex-representation). Žinoma, kad turint *H-representaciją*, ją galima paversti *V-representacija* [AF92] [FP96]. Tai padaroma tam, kad būtų galima atlikti CSG (Constructive Solid Geometry) operacijas.

Čia reikia atkreipti dėmesį į tai, kad matomumo tieses reprezentuojantis briaunainis bendru atveju nėra uždaras kūnas. Tačiau tai nereiškia, kad jis neefektyviai apima (arba neapima išvis) aibės matomumo tiesių. Dažniausiai gaunamas prizminis objektas, kuris neturi galus ribojančių plokštumų. Šių plokštumų stoka nesudaro principinės problemos. Jos gali būti apskaičiuotos su tam tikru rezervu, užtikrinant, kad jos yra pakankamai toli nuo briaunainio centro, kad nedarytų įtakos matomumo tiesių aibei. Konkretūs skaičiavimai šiame darbe nėra labai aktualūs; esant susidomėjimui, rekomenduojama skaityti 5.2.5 skirsnį Nirensteino disertacijoje [Nir03].

### 2.4.1. CSG Plückerio erdvėje

Tam, kad galima būtų pašalinti iš matomumo tiesių poaibį, kurį užstoja matomumo blokas, reikia rasti ir jo reprezentaciją. Tai padaroma panašiai kaip ir su visais matomumo tieses reprezentuojančiu briaunainiu. Gaunamas neuždaras briaunainis, bet uždarumas šiuo atveju nėra būtinas. Pakanka turėti aibę hiperplokštumų  $\mathcal{O}$ , kuriomis vėliau galima "karpyti" matomumo briaunainį.

Turint briaunainius  $\mathbb{R}^5$  erdvėje, iš matomumo briaunainio yra atimami briaunainiai, reprezentuojantys matomumą užstojančius objektus. Konkretūs skaičiavimai aprašyti Nirensteino disertacijos 5.2 poskyryje, šiame darbe juos kartoti nėra poreikio. Čia pateikiama tik schema, vizualiai iliustruojanti procesą, žr. 1 paveikslą 10 puslapyje.

Taip pat verta paminėti *briaunainių komplekso* (polytope complex) sąvoką [BP96]. Tai yra vieną kompleksiską figūrą reprezentuojančių paprastesnių, iškilųjų figūrų aibė  $\mathcal{C}$ . Skaidant briaunainį, jis neišvengiamai sudėtingėja. Didėja kompleksas, kurio visuma sudaro briaunainį.

## 2.5. Nirensteino siūlytos optimizavimo strategijos

Kaip matome iš aukščiau aprašytų metodų, matomumo skaičiavimai yra ypač brangūs. Nesiimant jokių optimizavimo priemonių, negalima tikėtis praktiškų rezultatų. Šiame poskyryje aprašomos optimizavimo strategijos, kurios buvo siūlomos anksčiau.

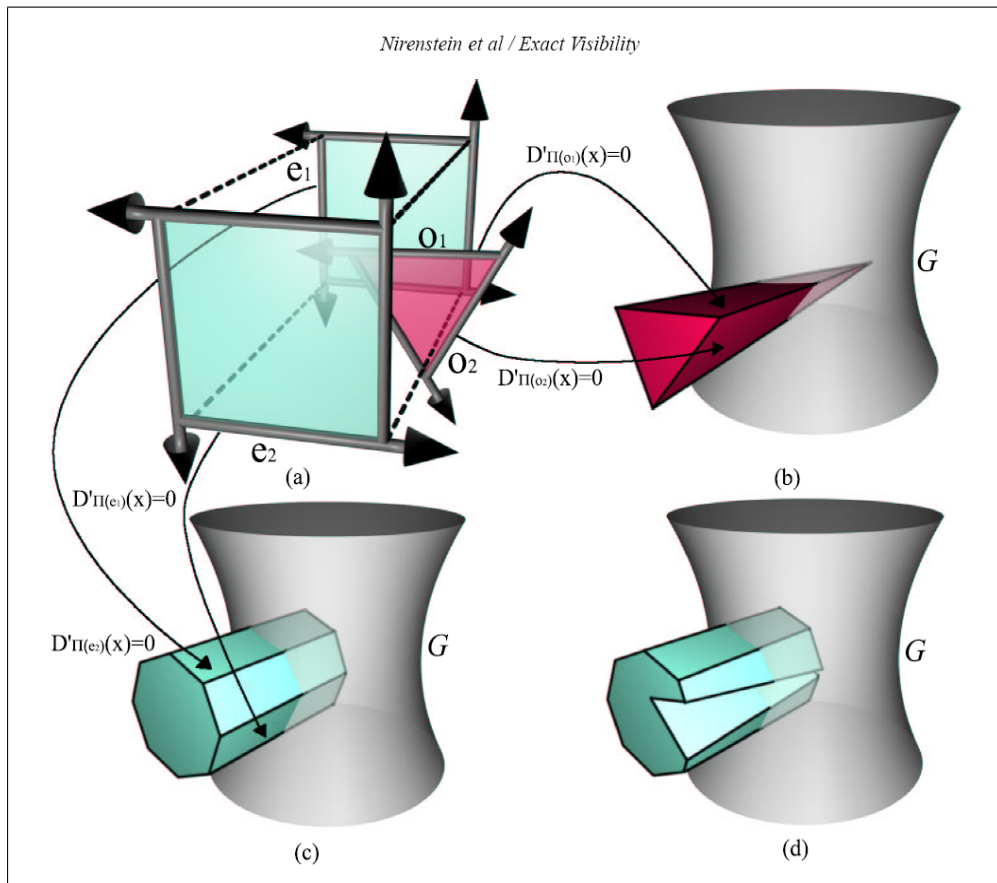
### 2.5.1. Menamų tiesių ignoravimas

Visų pirma, turint matomumą užstojančią objektą reprezentuojančią plokštumų aibę  $\mathcal{O}$  ir skaidant matomumo briaunainių kompleksą  $\mathcal{C}$ , iš rezultato pašalinami briaunainiai, kurie nesikerta su Plückerio hiperpaviršiumi. Tai leidžia turėti norimą kompleksą tik Plückerio hiperpaviršiaus zonoje, ignoruojant tai, ko nereikia.

Briaunainiai, nesiliečiantys ar nekertantys Plückerio hiperpaviršiaus yra neaktualūs skaičiavimams. Taip yra todėl, kad tik  $\mathbb{P}^5$  taškai iš Plückerio hiperpaviršiaus turi *realius* atitikmenis  $\mathbb{R}^3$  erdvėje. Kiti taškai atitinka tieses su menamu krypties koeficientu ir matomumo skaičiavimo kontekste neturi prasmės.

### 2.5.2. Trivialus pripažinimas

Matomumo patikrinimo operacija yra brangi, o kai kuriais atskirais atvejais matomumą tarp objektų galima patikrinti trivialiai. Vienas iš būdų yra pabandyti



1 pav.: *CSG operacija*. (a) Tipiška matomumo nustatymo problema: norima nustatyti matomumą tarp dviejų keturkampių, kai vaizdą (ar jo dalį) užstoja kita geometrija (trikampis). (b) Užstojančios geometrijos reprezentavimas  $\mathbb{R}^5$ . Figūra  $G$  čia vaizduoja Plückerio hiperpaviršių. (c) Toks pat reprezentavimas matomumo briaunainiui  $\mathbb{R}^5$ . (d) Dalinai matomumą užstojantis briaunainis atimamas iš pradinio tūrio. Likusi figūra pilnai reprezentuoja matomumo tieses, kurios prasilenkia su užstojančiąja geometrija.

apskaičiuoti keletą (atsitiktinių arba dėsningų) matomumo tiesių, ir sukirsti jas su kita geometrija. Jeigu tokia tiesė nekerta jokios kitos geometrijos, tik dominančius poligonus, matomumas yra trivialiai pripažįstamas — matomumo užklausiai neįdomu *kiek* objekto matoma, vienintelis dominantis dalykas yra *ar matoma bent kiek*.

### 2.5.3. Matomumą užstojančių objektų prioritetai

Aukščiau minėto trivialaus testo rezultatas ne visada būna pripažinimas. Kartais, išbandžius visas tieses iki tam tikro norimo skaičiaus, aptinkama, kad nė viena iš jų nepatvirtina matomumo. Tokiais atvejais matomumą užstojančios objektai gauna didesnę prioritetą priklausomai nuo to, kelias tieses jie užstojo. Tai grindžiama tuo, kad objektas, užstojęs daugiausiai trivialaus testo tiesių, turi didesnę tikimybę įdėti didesnę indėlį į nematomumo garantavimą, atliekant tikslaus matomumo užklausa.

### 2.5.4. Užklausa grupėms

Praktiškai visos scenos, kurios labiausiai domina, savyje turi tam tikrą objektų aibę. Tai suteikia galimybę prieš tikrinant paties objekto matomumą, patikrinti jo gaubiančiojo briaunainio matomumą. Jeigu gaubiantysis briaunainis nėra matomas, tai akivaizdu, kad objektas, kurį jis gaubia taip pat negali būti matomas. Tai yra paranku, nes patikrinti šešias sienas turinčio stačiakampio gretasienio matomumą yra nepalyginamai pigiau, nei kelis šimtus ar net tūkstančius poligonų turinčio objekto, kurį jis gaubia.

Matomumo patikrinimas tarp dominančios zonos ir gaubiančio briaunainio iš principo reiškia visų sienų kombinacijų matomumo patikrinimą. Tokių kombinacijų yra 36, tačiau ignoruojant priešingas normales turinčius poligonus, daugumai tokių porų skaičiavimų neprireiks. Jeigu matomumo užklausa vienai iš tokių porų gražins "matoma", tai gaubiantysis briaunainis laikomas matomu. Šią testą pakanka atlikti iki pirmojo teigiamo rezultato, tačiau žemiau paaiškinama, kodėl apsimoka pabaigti šią operaciją iki galo.

### 2.5.5. Pakartotinis gaubiančiosios figūros matomumo išnaudojimas

Atlikus aukščiau minėtą gaubiančiosios figūros matomumo testavimą iki galo, gaunama reprezentacija *visų* neužstotų matomumo tiesių, kurios prasideda dominančioje zonoje ir kerta gaubiančiąją figūrą. Šią informaciją galima įsivaizduoti kaip *anti-pusšėšėlį* (anti-penumbra) [Tel92a]. Turint šią reprezentaciją, galima ją

panaudoti skaičiuojant gaubiamojo objekto matomumą. Visi gaubiamojo poligonai, kurie yra matomi, *privalo* kirsti šį "anti-pusšėšėlį".

Taip gaunamas greitas, konservatyvus, bet sąlyginai tikslus testas dideliems objektą sudarančių poligonų kiekiams atmesti. Tik ta dalis objekto poligonų, kuri praeina šį testą, yra toliau individualiai testuojami tiksliau matomumo testu.

### 2.5.6. Virtualūs vaizdo blokai

Skaičiuojant gaubiančiųjų briaunainių matomumą, atsiranda galimybė sukonstruoti *virtualius vaizdo blokus* (virtual occluders). Virtualūs vaizdo blokai yra objektai, kurie nėra originalios scenos geometrijos dalis, bet vis dėlto reprezentuoja vaizdą užstojančius objektus [KCC00]. Jeigu gaubiančiojo briaunainio matomumo testas parodė, kad jis nematomas, reiškia, šį kuboidą galima panaudoti kaip virtualų vaizdo bloką. Maža to, visas gaubiamasis objektas gali būti išmestas iš potencialių vaizdo blokų sąrašo, nes gaubiančiojo briaunainio ne tik pakanka, bet dar ir *daugiau, nei pakanka*.

Skaičiuojant matomumą, scena yra apdorojama apytiksliai atstumo didėjimo tvarka [KS00]. Tai leidžia efektyviai išnaudoti virtualius vaizdo blokus: artimiausi objektai yra trivialiai pripažįstami, tolimesni objektai sunaudoja daugiausiai skaičiavimų, o tolimiausi, naudojant virtualius matomumo blokus, yra ypač pigiai atmetami kaip nematomi.

### 3. Optimizacijų realizacija

Aukščiau aprašytos optimizavimo strategijos, kurių ėmėsi minėtieji autoriai, duoda akivaizdžių rezultatų. Toliau pristatomas siūlymas, kaip visą procesą būtų galima dar paspartinti.

Trumpai pristatant, pagrindinė optimizacijos idėja yra pigiai sumažinti potencialių vaizdo blokų aibę dar  $\mathbb{R}^3$  erdvėje, prieš atliekant sudėtingus aukščiau minėtus skaičiavimus. Čia pateikiami metodai, kaip to pasiekti vienkartinio hierarchinio scenos suskaidymu per  $\mathcal{O}(n \log n)$  laiką, ir vėliau kiekvieną paiešką sumažinti iki  $\mathcal{O}(\log n)$  laiko vietoje  $\mathcal{O}(n)$ . Tai leidžia labai ženkliai sumažinti duomenų kiekį, kuris yra toliau apdorojamas aukštesnio sudėtingumo algoritmais. Toks sumažinimas galimas atliekant greitą volumetrinę paiešką erdvėje.

#### 3.1. Hierarchijos įvedimas

Pagrindinis metodo paspartinimas gaunamas ženkliai sumažinant pradinį duomenų kiekį ankstyvoje skaičiavimų stadijoje. Toks paspartėjimas galimas tik turint hierarchiją, kuri leidžia operatyviai ieškoti objektų erdvėje.

Alternatyvų hierarchiniam organizavimui yra nemažai; štai keletas populiariausių metodų:

- BSP medis (Binary Space Partitioning: dvejetainis erdvės dalinimas) [Thi87]
- KD-medžiai
- Octree (aštuonetainis medis)
- AABB medis (Axis Aligned Bounding Box: gaubiantysis kuboidas su ašimis statmenomis sienomis)

Visais atvejais hierarchijos konstravimas yra linearitminio sudėtingumo. Nors tai ir nėra labai pigu, bet toks konstravimas turi būti atliktas tik vieną kartą visai scenai, tad iš esmės įtakos visiems skaičiavimams praktiškai neturi.

Dėl realizacijos paprastumo ir argumentų, kodėl kitokia hierarchija turėtų duoti ženkliai skirtingus rezultatus, stokos, šio darbo tikslams buvo pasirinktas Octree.

#### 3.2. Konstravimas

2 paveikslas 15 puslapyje iliustruoja medžio konstravimą. Dvimačiu atveju procesas vyksta taip, kaip parodyta paveiksle. Trimačiu atveju atsiranda papildoma plokštuma, ir kiekvienas medžio lygis turi po aštuonis vaikus.

Konstruojant medį, sudėtingiausia dalis yra apsispręsti ką daryti su objektais, kurie dalinai patenka į medžio skirsnį (paryškinti objektai paveiksle). Galimi keletas variantų:

1. Objekto skaidymas
2. Objekto priskyrimas kažkuriam vienam skirsniumi
3. Objekto priskyrimas visiems skirsniams

Pirmasis variantas yra aktualesnis tada, kai hierarchija naudojama vaizdavimui, o ne išankstiniams skaičiavimams. Tačiau netgi vaizdavimui šiais laikais naudojamas vaizdo plokščių teikiamas  $Z$  buferis, tad papildomas scenos kompleksiskumo kėlimas yra nepraktiškas.

Antrasis variantas gali būti neretai naudojamas. Jis taupiai naudoja atmintį, kiekvieno objekto lieka po vieną kopiją. Tačiau esama situacijų, kai objekto priskyrimas vienam skirsniumi gali duoti neteisingus rezultatus. Toks atvejis matomas 3 paveiksle, 16 puslapyje: apatinis objektas gali būti surastas, o gali būti nesurastas, priklausomai nuo to, kuriam skirsniumi jis bus priskirtas. Tačiau akivaizdu, kad jis privalo būti surastas.

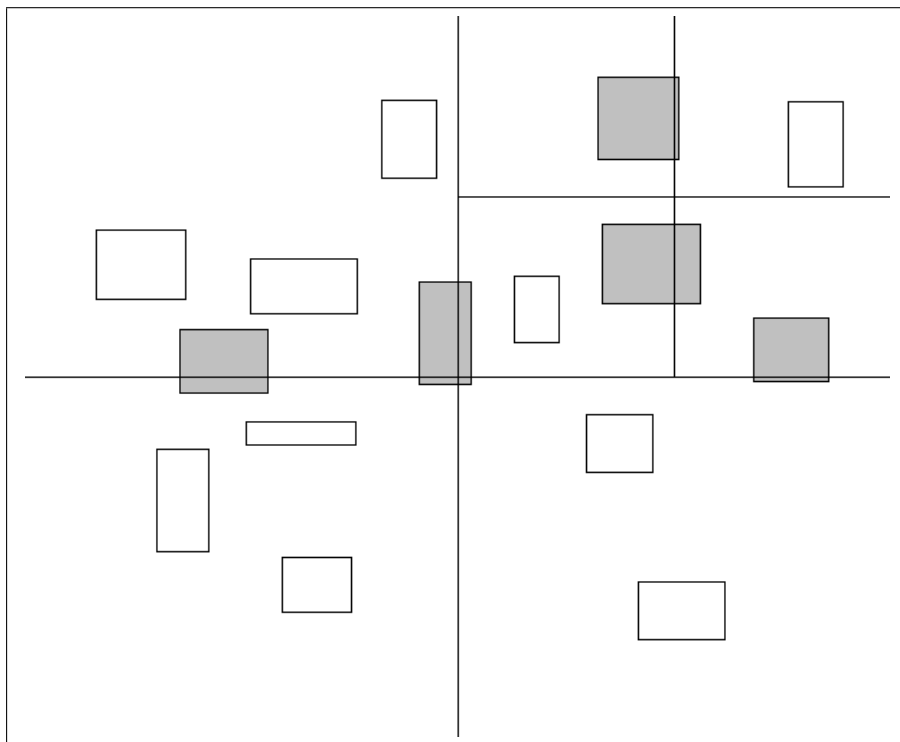
Todėl šiame darbe pasirinktas trečiasis variantas. Truputį padidėja atminties poreikiai, tačiau išvengiama paieškos netikslumų; paieška mūsų atveju privalo būti konservatyvi.

### 3.3. Paieška: Dvimatis atvejis

Pirmiausia buvo realizuotas dvimatis prototipas. Scena, vaizduojanti beveik plokščią mišką, buvo suskaidyta quad-tree (ketvirtainio medžio) metodu, ignoruojant vertikalią dimensiją. Toks būdas pasirinktas pirmiausia vardan paprastumo, nesudėtingam hipotezių patikrinimui, tačiau nereta atveju jis gali būti pakankamas ir kaip pagrindinis metodas. Taip yra todėl, kad matomumo skaičiavimo interesai dažnai būna "dvimatiški": urbanistinės scenos, tankiai užstotos gamtinės scenos (pvz., miškai).

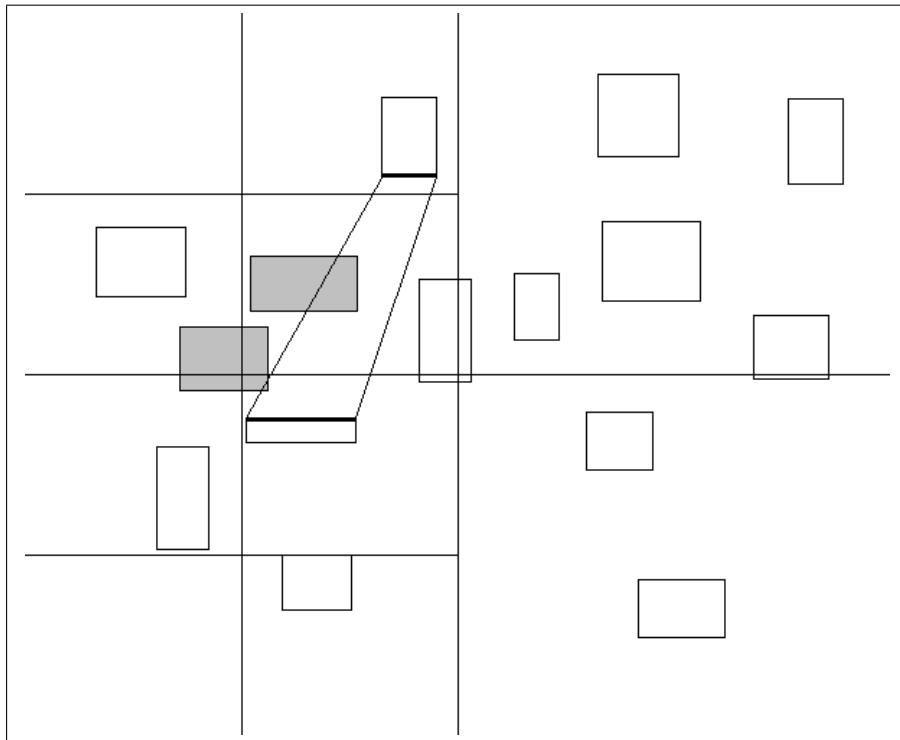
### 3.4. Paieška: Trimatis atvejis

Trimačiame aštuonetainiame medyje paieška atliekama tokiu pačiu principu, kaip ir dvimačiame ketvirtainiame. Smulkesnio aprašymo tai nereikalauja, o vizuali medžiaga per sunkiai suvokiama, kad būtų verta bandyti ją perteikti. Kiekvienas žingsnis vykdomas analogiškai.



2 pav.: *Medžio konstravimas*. Ši schema iliustruoja medžio konstravimo dvimatį variantą. Mažieji stačiakampiai reprezentuoja scenos objektus. Scena pirmiausiai suskaidoma į lygias dalis, medžio šaknies vaikus. Patamsinti objektai reprezentuoja nepilnai patenkančius į sektorių objektus. Viršutiniame dešiniame ketvirtyje parodytas kitas rekursijos lygmuo, tolimesnis skaidymas.





3 pav.: *Paieška ketvirtainiame medyje*. Ši schema iliustruoja paiešką medyje. Mažieji stačiakampiai reprezentuoja scenos objektus. Paryškintos linijos — dominančius poligonus. Paryškintų linijų kraštus siejančios linijos — OBB. Ieškant medyje pirmiausiai atmetami abu kairieji aukščiausio hierarchijos laipsnio ketvirčiai, nes jie nesikerta su dominančia scenos sritimi. Procesas rekursiškai kartojamas, atmetant nedominančius ketvirčius ir apsistoiant ties trim dominančiais, kuriems priklausantys objektai jau yra pilnai enumeruojami.

Visgi, esama ir skirtumų. Pagrindinis sunkumas, su kuriuo susiduriama  $\mathbb{R}^3$  erdvėje, yra dominančio poerdvio reprezentavimas. Dvimačiu atveju reprezentuoti kompaktišką dominančią sritį tarp testuojamų atkarpų yra trivialu: tereikia sujungti atkarpų kraštus papildomomis ribojančiomis tiesėmis. Tuo tarpu trimatėje erdvėje atitinkamas reprezentavimas sukelia tam tikrų sunkumų ar bent jau dvejonų.

Šiame darbe apvarstomi keturi būdai reprezentuoti dominančią sritį:

- Gaubiančioji sfera (Bounding Sphere, BS)
- Gaubiantysis kuboidas su ašims statmenomis sienomis (Axis Aligned Bounding Box, AABB)
- Orientuotas gaubiantysis kuboidas (Oriented Bounding Box, OBB)
- Orientuotas gaubiantysis cilindras (Oriented Bounding Cylinder, OBC)

Žemiau plačiau aprašomas kiekvienas iš jų.

### 3.4.1. Gaubiančioji sfera

Bounding Sphere (gaubiančioji sfera) — tai pats paprasčiausias (bet tuo pačiu ir mažiausiai efektyvus) gaubiančiųjų figūrų variantas. Turint aibę taškų, suskaičiuojama sfera, kuri gaubtų juos visus. Gaubiančiosios sferos privalumas tas, kad itin pigiai atliekami paieškos testai, tereikia apskaičiuoti dominančio taško atstumą iki sferos centro ir palyginti jį su radiusu.

Naudotis vien tik gaubiančiąja sfera neefektyvu, nes kai kurių kombinacijų atveju sfera gali gautis tokia didelė, kad jos tūris bus didesnis už visos scenos.

### 3.4.2. Gaubiantysis kuboidas su ašims statmenomis sienomis

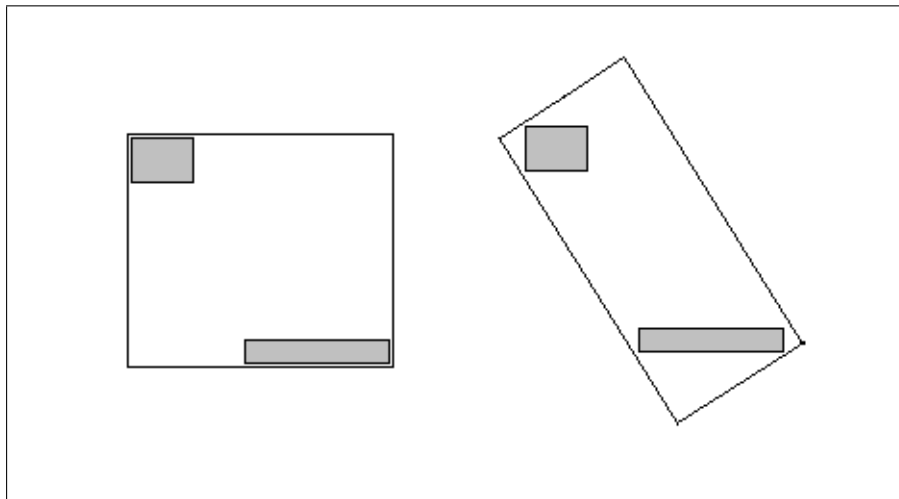
Axis Aligned Bounding Box (gaubiantysis kuboidas su ašims statmenomis sienomis) — tai antra pagal paprastumą gaubiančioji figūra. Tai yra stačiakampis gretasienis, sudarytas iš plokštumų, lygiagrečių bazinėms plokštumoms XY, XZ bei ZY, bei patalpintų ties ekstremaliais dominančio objekto taškais.

AABB privalumas toks, kad jis kartu ir itin lengvai sukonstruojamas, ir labai pigiai naudojamas paieškoje. Tačiau trūkumas (kaip ir gaubiančiosios sferos atveju) yra mažas erdvinis efektyvumas. Skaičiuojant AABB priešinguose scenos kampuose patalpintiems objektams, kuboidas gali gautis beveik tokio pat tūrio, kaip ir visa scena.

### 3.4.3. Orientuotas gaubiantysis kuboidas

Oriented Bounding Box (orientuotas gaubiantysis kuboidas) — tai stačiakampis gretasienis, sudarytas taip, kad efektyviai apribotų erdvę tarp objektų, kuriuos jis gaubia.

4 paveikslas 18 puslapyje iliustruoja AABB ir OBB skirtumą.



4 pav.: *AABB ir OBB*. Kairėje parodyta *AABB*, apskaičiuota dviems objektams plokštumoje, dešinėje — *OBB* tiems patiems dviems objektams. Akivaizdu, kad didėjant atstumui tarp objektų, *OBB* plotas auga tiesiškai, o *AABB* — kvadratiškai.

### 3.4.4. Orientuotas gaubiantysis cilindras

Oriented Bounding Cylinder (orientuotas gaubiantysis cilindras) — šio darbo procese gimusi idėja, kaip pakelti erdvinį bei komputacinį gaubiančiųjų paviršių efektyvumą. *OBC* tūris yra dar mažesnis (nors ir nežymiai) už *OBB*, tuo tarpu skaičiavimai pigesni (tiek konstravimas, tiek ir naudojimas ieškant).

Konstruojant gaubiantįjį cilindrą, pirmiausiai randami dviejų dominančių poligonų centrai. Tarp jų nubrėžiama cilindro ašis. Cilindro spindulys — tolimiausiai nuo centro nutolusios viršūnės atstumas. Iš galų cilindras apribojamas tų pačių poligonų plokštumomis<sup>3</sup>.

Atliekant paiešką scenoje, pirmiausia atmetami hierarchijos mazgai, kurie atsiduria už cilindrą ribojančių plokštumų. Toliau paieška tęsiama tarp likusių hierarchijos mazgų ieškant tokių, kurių atstumas iki cilindro ašies mažesnis už spindulį.

<sup>3</sup>Kadangi šios plokštumos visai nebūtinai turi būti lygiagrečios (priešingas atvejis netgi dažnesnis), gaunamas ne visai cilindras, o greičiau nupjautinis cilindras.

## 4. Siūlomo metodo palyginimas su ankstesniais darbais

Šiame skyriuje analizuojamos optimizavimo strategijos, kurių ėmėsi ankstesnieji autoriai. Jie lyginami su šiame darbe pasiūlytais, kad būtų įsitikinta pastarųjų naudingumu.

Šio skyriaus poskyriai sutampa su 2.5 poskyrio skirsniais, aptariant ir paaiškinant, kaip šio darbo rezultatai sąveikauja su kiekvienu iš minėtų metodų. Pačių metodų aprašymai čia nebekartojami, tik trumpai primenama jų esmė.

### 4.1. Menamų tiesių ignoravimas

Ši optimizacija orientuota į vengimą turėti duomenų perteklių skaičiuojant matomumo užklausa. Ja stengiamasi kuo anksčiau atsikratyti matomumo "pseudotiesių", kurios  $\mathbb{R}^3$  erdvėje turi menamą krypties vektorių.

Akivaizdu, kad ši optimizacija yra matomumo užklauskos dalis, viena iš vidinių skaičiavimų stadijų, norint pasakyti, ar yra įmanomas matomumas. Iš principo tarp pačių metodų nėra daug bendro, tačiau menamų tiesių ignoravimas tiesiogiai priklauso nuo vaizdą blokuojančių objektų, kuriuos teks apdoroti, skaičiaus. O pastarųjų kiekis, naudojant šio darbo rezultatus, yra stipriai sumažinamas ankstyvoje skaičiavimų stadijoje.

Išvada: menamų tiesių ignoravimui šio darbo metodai neigiamos įtakos neturi, turi teigiamą.

### 4.2. Trivialus pripažinimas

Prieš atliekant brangią matomumo užklausa, pabandoma pasitikrinti, ar nepavyks trivialiais metodais užtikrinti teigiamą atsakymą. Tai atliekama naudojant spindulių trasavimą. Trasuojant spindulius vėl yra aktualus vaizdą blokuojančių objektų skaičius. Mažinant blokų skaičių, greitėja trasavimas, tad trivialiam pripažinimui mūsų siūlomas metodas tik į naudą.

Maža to, atliekant hierarchinę vaizdo blokų paiešką, galimas ypatingas rezultatas: tuščia aibė. Jeigu erdvėje tarp dviejų dominančių poligonų nerandama nė vieno vaizdą blokuojančio objekto, tai *jau* yra trivialus pripažinimas; spindulių trasavimo stadija gali būti visai neatliekama, akivaizdu, kad matomumas tapo dominančių poligonų yra ne tik *galimas*, bet ir *būtinis*.

Išvada: trivialiam matomumo pripažinimui hierarchinė paieška turi tik pozityvią įtaką, kartais visiškai eliminuojant poreikį papildomiems veiksams.

### 4.3. Matomumą užstojančių objektų prioritetai

Jeigu aukščiau minėtas spindulių trasavimas yra vykdomas ir parodo neigiamą rezultatą, vaizdą blokuojantiems objektams suteikiamas prioritetas pagal tai, kiek trasuotų spindulių užblokavo konkretus objektas — kuo daugiau, tuo didesnė tikimybė, kad objektas turi *lemiamą* įtaką matomumui.

Hierarchinė paieška niekaip neturi įtakos šiam metodui. Jeigu paieška pati savaime parodo matomumą tarp poligonų, tą patį padarytų ir spindulių trasavimas, tad objektų prioritetai nebūtų skirstomi. Jeigu trasavimas visgi vykdomas ir parodo galimą nematomumą, prioritetai skirstomi visiškai identiškai — nepriklausomai nuo to, ar buvo hierarchinė paieška prieš tai.

Išvada: hierarchinė paieška neturi nei teigiamos, nei neigiamos įtakos matomumą blokuojančių objektų prioritetams.

### 4.4. Užklausa grupėms

Ši optimizacija yra geometriškai paprastesnės gaubiančiosios figūros matomumo testas, prieš testuojant sudėtingesnę gaubiamąją figūrą. Jos tikslas — padaryti prielaidą, kad objektas nematomas ir ją patikrinti, atliekant tai daug pigiau, nei iš tikrųjų tikrinant objektą. Iš principo tai labai efektyvus būdas sumažinti kreipinių į matomumo užklausą skaičių.

Kadangi hierarchinės paieškos nauda yra duomenų *kiekio*, siunčiamo matomumo užklausiai mažinimas, o ne kreipinių *skaičiaus* mažinimas, gaubiančiųjų figūrų panaudojimas veikia kooperatyviai su hierarchine paieška, skirtingomis priemonėmis spartinant bendrą procesą.

Išvada: hierarchinė paieška netrukdo užklausoms grupėmis; turi (bendru mastu) nežymios teigiamos įtakos.

### 4.5. Pakartotinis gaubiančiosios figūros matomumo išnaudojimas

Ši optimizacija išnaudoja aukščiau minėtame žingsnyje gautą objekto gaubiančiosios figūros dalių matomumo/nematomumo informaciją. Ji yra panaudojama atliekant paties gaubiamojo objekto matomumo analizę. Kadangi ši analizė yra matomumo užklaunos stadija, atliekama *tarp* gaubiančiojo objekto matomumo testavimo ir pavienių objektą sudarančių poligonų matomumo testavimo, hierarchinė paieška neturi jokios įtakos šiam metodui.

Išvada: hierarchinė paieška niekaip nedaro įtakos pakartotiniam gaubiančiosios figūros matomumo informacijos panaudojimui.

#### 4.6. Virtualūs vaizdo blokai

Virtualūs vaizdo blokai yra tos gaubiančiosios figūros, kurioms buvo konstatuotas nematomumas. Jeigu objektas nėra matomas, tai natūralu, kad negalima matyti ir nieko už jo. Privalumas toks, kad iš potencialių vaizdą blokuojančių objektų sąrašo pašalinamas gerokai sudėtingesnis gaubiamasis objektas — sumažinamas duomenų kiekis, kuris bus toliau naudojamas procese.

Yra priežasčių manyti, kad ypač tankiai užstotose scenose virtualūs vaizdo blokai yra svarbesnis skaičiavimus greitinantis faktorius, nei hierarchinė paieška. Tačiau problema ta, kad virtualūs vaizdo blokai atsiranda ganėtinai vėlai, jau apdorojus daug geometrijos. Taip yra dėl natūralių priežasčių:

1. Artimiausi testuojami objektai yra trivialiai konstatuojami matomais.
2. Tolstant nuo zonos, triviųjų pripažinimų mažėja, daugėja matomumo užklausų kiekis.
3. Tolstant dar toliau, po kurio laiko atsiranda pirmasis nematomas objektas, pirmasis virtualus vaizdo blokas. Matomumo užklausų kiekis vėl pradeda mažėti.

Hierarchinė paieška ypač naudinga pirmosiose dviejose stadijose. Trečiojoje jos naudingumas galbūt nublinksta prieš virtualių vaizdo blokų naudingumą, bet ji ir netrukdo, nežymiai spartina.

Išvada: naudojant virtualius vaizdo blokus, hierarchinė paieška pagrindinį savo naudingą indėlį padaro skaičiavimų pradžioje, vėliau indėlis mažėja, bet paieška netrukdo. Maža to, kuo rečiau užstota scena, tuo hierarchinė paieška naudingesnė, nes mažėja tikimybė atsirasti virtualiam vaizdo blokui.

## Išvados, rezultatai ir tolimesnis darbas

Pastarųjų metų darbai parodė, kad matomumo skaičiavimo bendras atvejis nėra neišsprendžiamas, kaip buvo manoma anksčiau. Tačiau darbui erdvės dar yra: vartotojų poreikiai auga greičiau, nei techninės galimybės, o turimi metodai nėra tokie efektyvūs, kad būtų galima teigti apie visiškai išsemtą sritį.

Šiame darbe siūlomas būdas paspartinti bendrą skaičiavimų atvejį. Paspartinimas efektyvus ir juntamas. Paskutiniame šio darbo skyriuje išanalizuojami siūlomos hierarchinės paieškos santykiai su kitų autorių siūlytais optimizavimo metodais, su atskiromis bazinio metodo stadijomis. Čia pateikiamas reziumė:

- Menamų tiesių ignoravimui šio darbo metodai neigiamos įtakos neturi, turi teigiamą.
- Trivialiam matomumo pripažinimui hierarchinė paieška turi tik pozityvią įtaką, kartais visiškai eliminuojant poreikį papildomiems veiksams.
- Hierarchinė paieška neturi nei teigiamos, nei neigiamos įtakos matomumą blokuojančių objektų prioritetams.
- Hierarchinė paieška netrukdo užklausoms grupėmis; turi (bendru mastu) nežymios teigiamos įtakos.
- Hierarchinė paieška niekaip nedaro įtakos pakartotiniam gaubiančiosios figūros matomumo informacijos panaudojimui.
- Naudojant virtualius vaizdo blokus, hierarchinė paieška pagrindinį savo naudingą indėlį padaro skaičiavimų pradžioje, vėliau indėlis mažėja, bet paieška netrukdo. Maža to, kuo rečiau užstota scena, tuo hierarchinė paieška naudingesnė, nes mažėja tikimybė atsirasti virtualiam vaizdo blokui.

Kaip matome, siūlomas metodas blogiausiu atveju neturi jokios įtakos kai kuriems kitiems metodams (t. y. iššvaistomas hierarchijos konstravimui sunaudotas laikas), tačiau daugumai turi teigiamos įtakos, kintančios nuo nežymios iki lemiamos.

Paskutinis punktas leidžia daryti prielaidą, kad naudojantis hierarchine vaizdą blokuojančių objektų paieška, praplatėja galimų apdoroti scenų spektras. Tačiau užtikrintai to teigti negalima dėl ribotos turimos informacijos. Problema ta, kad itin retai užstotose scenose išankstinis matomumo skaičiavimas apskritai neturi prasmės: bendru atveju niekas nieko pilnai neužstoja, tik truputį persidengia smulkios detalės. Nubrėžti ribą yra labai sunku, tam reikia papildomų tyrimų. Nenubrėžus

ribos, neaišku, ar išbandžius hierarchinę paiešką mažiau užstotose scenose būtų suskaičiuota prasminga matomumo informacija, ar būtų konstatuota, kad scenoje per mažai vaizdą užstojančių objektų. Tolimesnis darbas šia linkme duotų labai naudingų rezultatų.



## Summary

When dealing with computer graphics, we always face a problem of having more geometry than we are able to process with hardware at hand. One of the best ways to deal with this problem is pre-processing of the scene to know what objects will make contribution to the resulting image beforehand. This method has proven its effectiveness.

Exact from-region visibility computations were a subject of exhaustive research for a long time. But the first complete results were shown only recently. Despite providing a sure improvement in the state of the arts, the available methodologies and algorithms do not yet fulfill the existing demand.

In this study we concentrate on the results of recent research on exact visibility computations, analysing the work of previous authors and seeking for optimisation opportunities.

We present a *volumetric hierarchical search* algorithm that improves the process of visibility pre-processing. An exhaustive analysis is presented, showing the effectiveness of our method. Our method also enables us to use exact visibility pre-processing for less densely occluded environments than it was possible before.

## Literatūros sąrašas

- [AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry*. 8, 1992, pp. 295–313.
- [Air91] J. Airey. *Increasing Update Rates in the Building Walkthrough System with Automatic Model-Space Subdivision and Potentially Visible Set Calculations*. Ph.D. Thesis, University of North Carolina, Chappel Hill, 1991.
- [ARB90] J. Airey, J. Rohlf and F. P. Brooks, Jr. Towards image realism with interactive update rates in complex virtual building environments. *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*. 24(2), 1990, pp. 41–50.
- [BP96] C. L. Bajaj and V. Pascucci. Splitting a Complex of Convex Polytopes In Any Dimension. *Computer Sciences Department*, Purdue University, May 1996.
- [CCS+03] D. Cohen-Or, Y. Chrysanthou, C. Silva and F. Durand. A Survey of Visibility for Walkthrough Applications. *IEEE Transactions on Visualization and Computer Graphics*. 9(3), 2003, pp. 412-431.
- [DDT+00] F. Durand, G. Drettakis, J. Thollot and C. Puech. Conservative visibility preprocessing using extended projections. *Proceedings of SIGGRAPH 2000*. 1, 2000, pp. 239–248.
- [Dur99] F. Durand. *3D Visibility, analysis and applications*. Ph.D. Thesis, U. Joseph Fourier, 1999. <http://graphics.lcs.mit.edu/~fredo>.
- [FDF+90] J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes. *Computer Graphics, Principles and Practice*, Second Edition. Addison-Wesley, Massachusetts, 1990.
- [FP96] K. Fukuda and A. Prodon. Double description method revisited. *Lecture Notes in Computer Science*. 1120, 1996, pp. 91–111.
- [GSF99] C. Gotsman, O. Sudarsky and J. Fayman. Optimized occlusion culling. *Computer & Graphics*. 23(5), 1999, pp. 645–654.
- [HT99] M. Hohmeyer and S. J. Teller. Determining the Lines Through Four Lines. *Journal of Graphics Tools*. 4(3), 1999, pp. 11-22.

- [KCC00] V. Koltun, Y. Chrysanthou and D. Cohen-Or. Virtual occluders: An efficient intermediate pvs representation. *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*. 1, 2000, pp. 59–70.
- [KS00] J. T. Klosowski and C. T. Silva. The prioritized-layered projection algorithm for visible set estimation. *IEEE Transactions on Visualization and Computer Graphics*. 6(2), 2000, pp. 108–123.
- [Lai05] S. Laine. A General Algorithm for Output-Sensitive Visibility Preprocessing. *Proceedings of the 2005 symposium on Interactive 3D graphics and games*. 1, 2005, pp. 31-40.
- [LG95] D. Luebke and C. Georges. *Portals and Mirrors: Simple, Fast Evaluation of Potentially Visible Sets*. Department of Computer Science, University of North Carolina at Chapel Hill, April 1995, <http://www.cs.virginia.edu/~luebke/publications/portals.html>
- [NB04] S. Nirenstein and E. Blake. Hardware Accelerated Visibility Preprocessing using Adaptive Sampling. *Rendering Techniques*. 1, 2004, pp. 207–216.
- [NBG02] S. Nirenstein, E. Blake and J. Gain. Exact from-region visibility culling. *Rendering Techniques 2002: Proceedings of the 13th Eurographics Workshop on Rendering*. 28, 2002, pp. 191-202.
- [NFL+99] B. Nadler, G. Fibich, S. Lev-Yehudi and D. Cohen-Or. A qualitative and quantitative visibility analysis in urban scenes. *Computer & Graphics*. 23(5), 1999, pp. 655–666.
- [Nir03] S. Nirenstein. *Fast And Accurate Visibility Preprocessing*, Ph.D. Thesis, Department of Computer Science, University of Capetown, 2003.
- [Pla92] H. Plantinga. An Algorithm for Finding the Weakly Visible Faces from a Polygon in 3D. *Fourth Canadian Conference on Computational Geometry*. 4, 1992, pp. 45–51.
- [SDD+00] G. Schaufler, J. Dorsey, X. Decoret and F. X. Sillion. Conservative volumetric visibility with occluder fusion. *Proceedings of SIGGRAPH 2000*. 1, 2000, pp. 229–238.
- [SLC+04] E. Sayer, A. Lerner, D. Cohen-Or, Y. Chrysanthou and O. Deussen. Aggressive visibility for rendering extremely complex plant ecosystems. <http://www.cs.tau.ac.il/~alan/Publications/Files/ComplexScenes.pdf>. 4.26 MiB, 2004.

- [Tel92] S. J. Teller. *Visibility Computations in Densely Occluded Polyhedral Environments*, Ph.D. Thesis, Computer Science Department, University of California at Berkeley, 1992.
- [Tel92a] S. J. Teller. Computing the antipenumbra of an area light source. *Computer Graphics (Proceedings of SIGGRAPH 92)*. 26, 1992, pp. 139–148.
- [TH93] S. J. Teller and M. Hohmeyer. Stabbing Oriented Convex Polygons in Randomized  $\mathcal{O}(n^2)$  Time. *Proceedings of the American Mathematical Society*. 178, 1993, pp. 311–318.
- [Thi87] W. Thibault. *Application of Binary Space Partitioning Trees to Geometric Modeling and Ray-Tracing*. Ph.D. Thesis, Georgia Institute of Technology, 1987.
- [TN87] W. Thibault and B. Naylor. Set Operations on Polyhedra Using Binary Space Partitioning Trees. *Computer Graphics*, 21(4), 1987, pp. 153–162.
- [TS91] S. J. Teller and C. H. Sequin. Visibility preprocessing for interactive walkthroughs. *Computer Graphics (Proceedings of SIGGRAPH 91)*. 25(4), 1991, pp. 61–69.
- [WWS00] P. Wonka, M. Wimmer and D. Schmalstieg. Visibility preprocessing with occluder fusion for urban walkthroughs. *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*. 1, 2000, pp. 71–82.