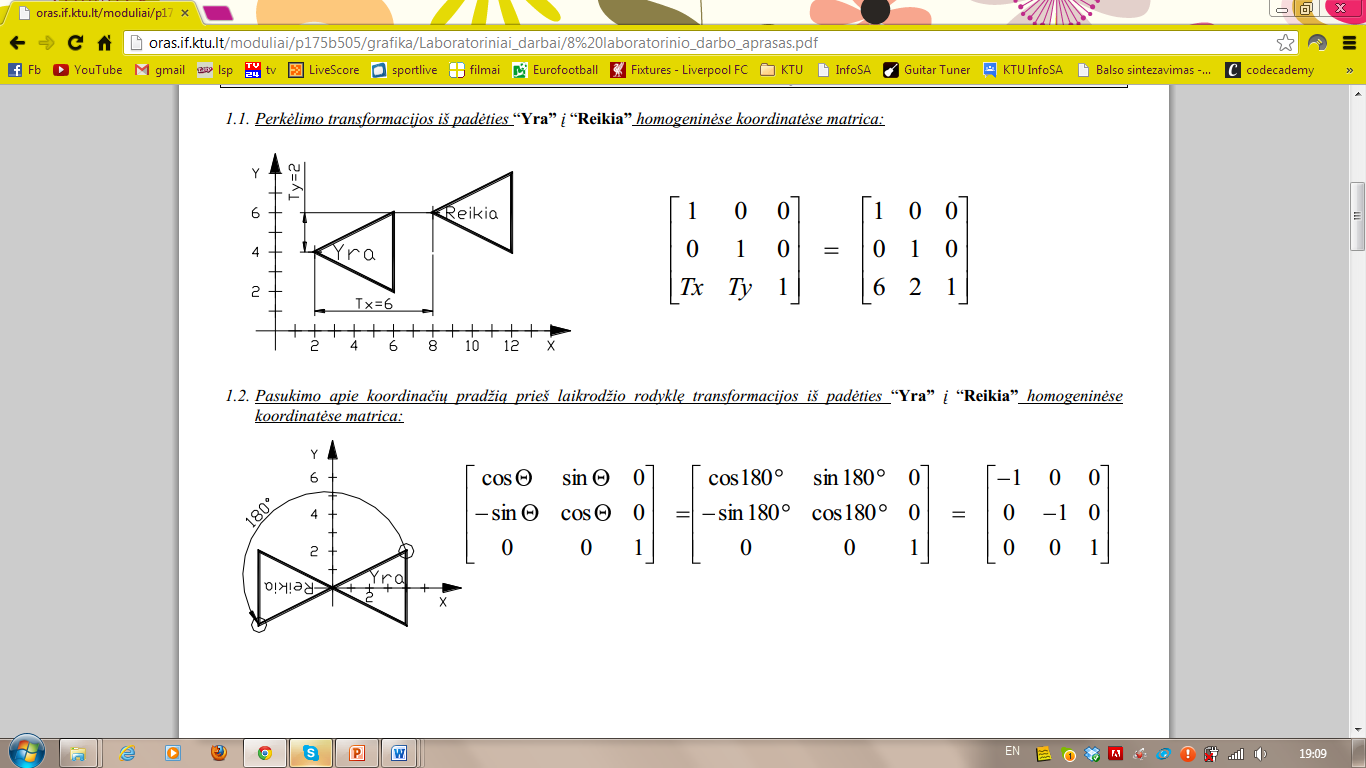
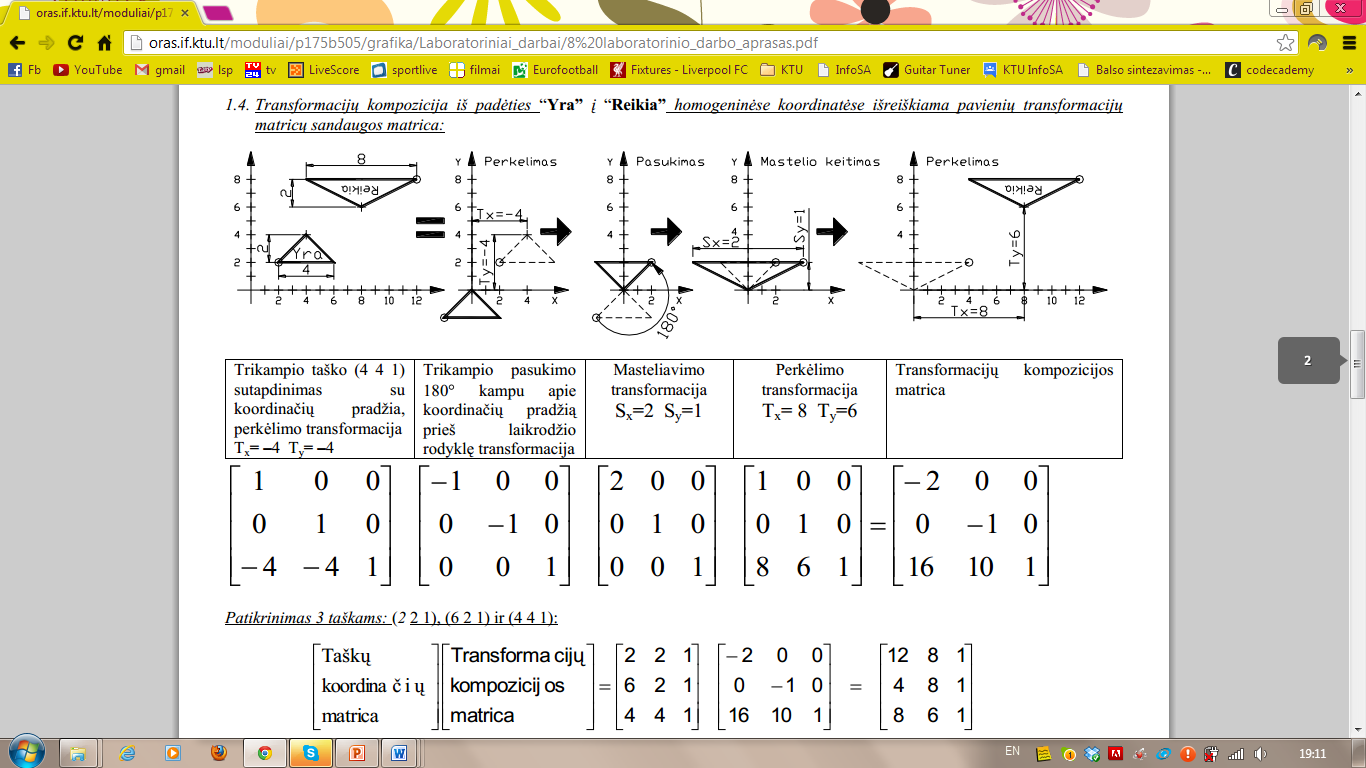
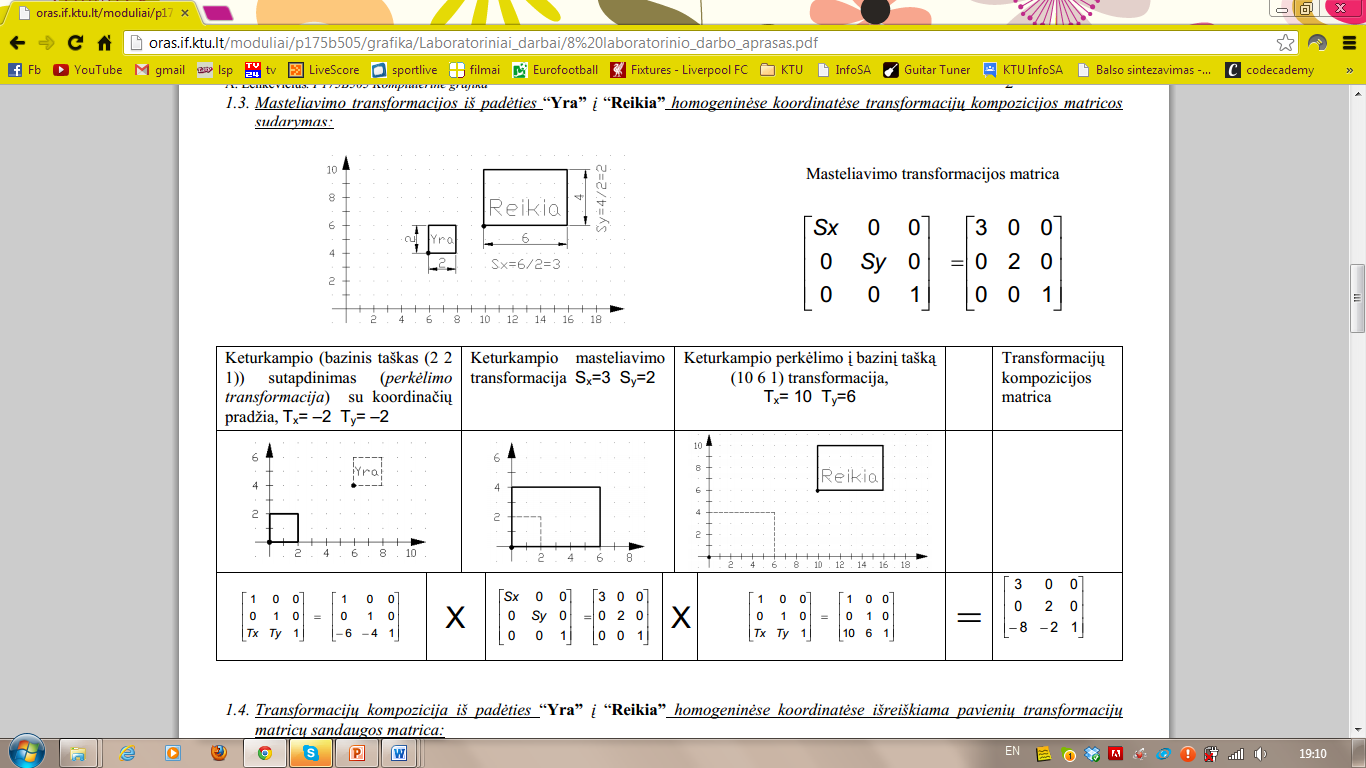
**Teminiai egzamino klausimai:**

1. **Uždavinys** - geometrinės transformacijos plokštumoje.





1. Vaizdo formavimo principai rastriniuose ir vektoriniuose vaizduokliuose.

Dauguma šiuolaikinių grafinių vaizduoklių yra taškiniai (rastriniai). Čia elektronų spindulys skleidžiamas skersai ekrano nuo viršaus į apačią eilutė po eilutės. Skleidžiant eilutę, spindulys gali būti įjungtas arba išjungtas. Įjungtas spindulys sukelia tam tikrų ekrano taškų švytėjimą, o išjungtas nekeičia ekrano spalvos.

Šiuose vaizduokliuose vaizdas saugomas atminties lauke, kuris vadinamas kadro arba regeneravimo buferiu. Šiame buferyje yra viso ekrano taškų švytėjimo ryškumo reikšmės. Jos vartojamos vaizdui ekrane sudaryti. Minimalus taškinės grafikos elementas yra taškas (pixel, pel). Kiekvieno taško spalvą ir ryškumą galima nurodyti neatsižvelgiant į visą vaizdą. Taškiniuose vaizduokliuose vaizdas regeneruojamas nuo 60 iki 80 kartų per sekundę.

Vektoriniais vaizduokliais vaizdas gaunamas braižant atkarpas, jungiančias du taškus. Tam elektronų spindulys nukreipiamas į vieną atkarpos galo tašką. Jį perkeliant į kitą tašką brėžiama atkarpa, kaip pavaizduota.

Regeneravimo greitis vektoriniuose vaizduokliuose priklauso nuo linijų, kurios turi būti brėžiamos, skaičiaus. Visų vaizdo komponentų išvedimas kartojamas nuo 30 iki 60 kartų per sekundę. Per regeneravimo ciklą geros kokybės vaizduokliais išvedama iki 100000 trumpų atkarpų.

Vektoriniai vaizduokliai skirti vaizdams iš linijų sudaryti. Jie netinka tikroviškoms spalvotoms scenoms vaizduoti. Tačiau šiais vaizduokliais gaunamos sklandžios linijos, kadangi čia elektronų spindulys tiesiog brėžia objektų briaunas. Taškiniais vaizduokliais iš taškų formuojamos linijos būna dantytos (jagged).

1. Brezenheimo rastrizavimo algoritmas atkarpai sudaryti (*išvedimas*).

Tarkime, intervale xk + 1 tiesės ir taškų (xk + 1, yk), (xk + 1), (xk + 1, yk + 1) ordinačių skirtumai yra d1 ir d2. Tiesės ordinatės reikšmė intervale xk + 1 gali būti apskaičiuota:

*y = m (xk + 1) + b,* **Tada:**

*d1 = y - yk = m (xk + 1)+b - yk*

*d2 = (yk + 1) – y = yk + 1 - m (xk + 1) – b*

**Ordinačių skirtumas:** *d1 – d2 = 2m (xk + 1) - 2yk + 2b -1*

**Išreiškiame pk:**

*pk = Dx(d1-d2) = 2Dy xk – 2Dx yk + c,* Dx > 0

Taško parinkimo algoritmo žingsnyje k + 1 parametras:

*pk+1 = 2Dy xk+1 – 2Dx yk+1 + c*

Atėmus:

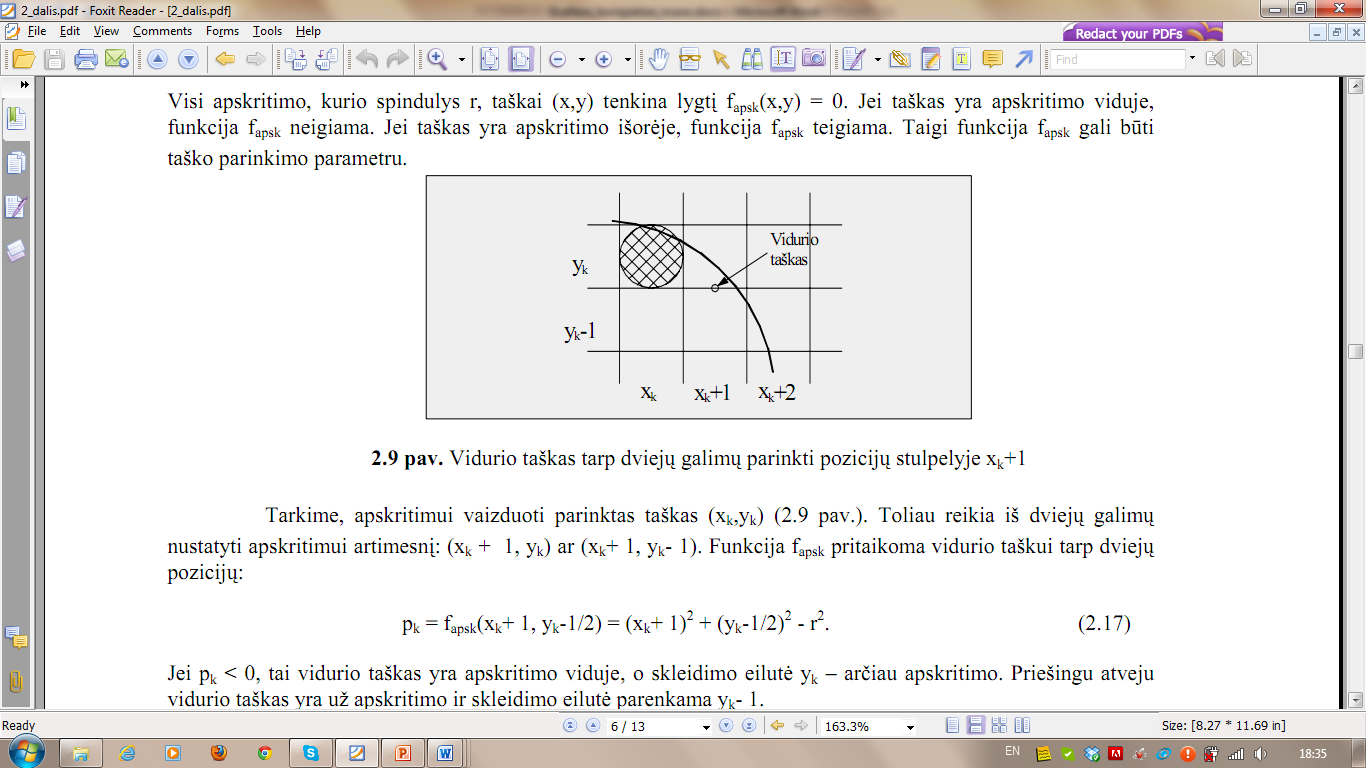
*pk+1 - pk = 2Dy (xk+1 – xk ) - 2Dx( yk+1 - yk)*

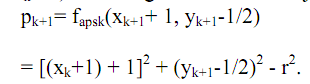
Kadangi *xk+1 – xk = 1,* tai *pk+1 = pk + 2Dy - 2Dx( yk+1 - yk). Čia yk+1* turi reikšmę 0 arba 1, kuri priklausonuo parametro *pk* ženklo. *xk = 0, yk = 0 ir b = 0,* pirmoji reikšmė lygi:

*p0  = 2Dy – 2Dx*

1. Vidurio taško rastrizavimo algoritmas apskritimui sudaryti (*išvedimas*).

Naudosimės vidurio taško algoritmo funkcija: . Visi apskritimo, kurio spindulys r, taškai (x,y) tenkina lygtį fapsk(x,y) = 0. Jei taškas yra apskritimo viduje, funkcija fapsk neigiama. Jei taškas yra apskritimo išorėje, funkcija fapsk teigiama. Taigi funkcija fapsk gali būti taško parinkimo parametru.

Tarkime, apskritimui vaizduoti parinktas taškas (xk,yk). Toliau reikia iš dviejų galimų nustatyti apskritimui artimesnį: (xk + 1, yk) ar (xk+ 1, yk- 1). Funkcija fapsk pritaikoma vidurio taškui tarp dviejų pozicijų:

Jei pk < 0, tai vidurio taškas yra apskritimo viduje, o skleidimo eilutė yk – arčiau apskritimo. Priešingu atveju vidurio taškas yra už apskritimo ir skleidimo eilutė parenkama yk- 1. Rekursinė kito žingsnio (xk+1+ 1 = xk+ 2) išraiška gaunama iš funkcijos fapsk įrašius:

Tada: čia yk+1 parenkamas yk ar yk-1. Tai priklauso nuo pk. pk+1 reikšmė gaunama prie pk pridėjus 2xk+1+ 1, jei pk neigiamas, arba 2xk+1+ 1 - 2yk+1 kitu atveju. Išraiškas 2xk+1 ir 2yk+1 galima pakeisti: Pradinėje pozicijoje (0, r) šių išraiškų reikšmė atitinkamai 0 ir 2r. Kiekviena kita 2xk+1 reikšmė gaunama prie ankstesnės pridėjus 2, o 2yk+1 reikšmė – iš ankstesnės atėmus 2. Pradinė parinkimo parametro išraiška gaunama įrašius į funkciją fapsk vidurio taško koordinates starto pozicijoje (x0,y0) = (0, r):

arba  Kadangi spindulys r yra sveikasis skaičius, p0 reikšmė apvalinama: 

1. Daugiakampių užpildymo metodai.

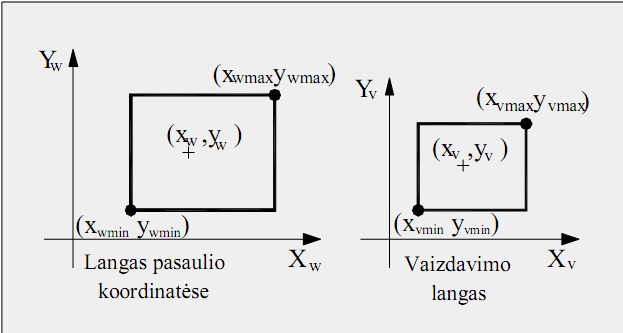
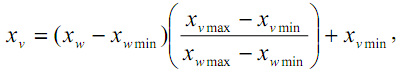
Taškinėse sistemose daugiakampiams užpildyti paplitę du būdai. Vienu būdu nustatomas daugiakampio ir jį kertančios ekrano skleistinės sanklotos intervalas. Kitu būdu užpildyti pradedama nuo daugiakampio vidinio taško ir toliau laipsniškai pildoma tol, kol pasiekiamos daugiakampio ribos.

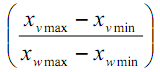
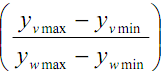
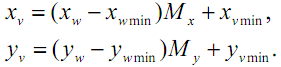
Kaip daugiakampis užpildomas skleistinėmis, pavaizduota 2.11 paveiksle. Nustatomi kiekvienos daugiakampį kertančios skleistinės ir plotą ribojančio daugiakampio briaunų sankirtos taškai. Šie taškai rūšiuojami poromis iš kairės į dešinę. Į atitinkamas kadro buferio pozicijas įrašomos kiekvienos sankirtos poros taškų spalvos.

Kai daugiakampio ribos fiksuotos ir nustatytas vidaus taškas, tikrinami jo gretimieji taškai. Nustatoma, kurie jų nepriklauso daugiakampio ribai, ir tiems priskiriama daugiakampio spalva. Kitu žingsniu nagrinėjami pastarųjų gretimieji. Tai tęsiama tol, kol daugiakampyje telieka daugiakampio vidaus ir jo ribos taškai.

1. Vaizdo transformavimas tarp langų (*išvedimas*).

Lango transformavimo iš pasaulio koordinačių į normalizuotąsias išraiškos yra šios:

ir  Iš čia išreiškiamos vaizdavimo koordinačių reikšmės:  

Čia ir  mastelių koeficientai Mx ir My. Tada ankstesnes lygybės lygybės gali būti išreikštos taip: 

1. Atkarpų atkirtimo algoritmai.

Atkarpų atkirtimo algoritmą galima padalyti į dvi dalis:

1. Patikrinti visas atkarpas ir atskirti tas, kurios kerta lango briaunas.

2. Atkirsti atkarpas nustačius jų susikirtimus su lango briaunomis.

Algoritmo pirmoji dalis gali būti realizuota kiekvieną atkarpą priskiriant vienai tokių kategorijų:

1. Matoma. Abu atkarpos galai yra lange.
2. Nematoma. Jei atkarpos abu galai yra už vienos lango ribos, jos nereikia vaizduoti.
3. Neapibrėžta. Atkarpa nepriklauso nė vienai aukščiau nagrinėtų kategorijų ir reikia nustatyti jos atkirtimą.

Gana senas ir dar populiarus Koheno ir Sačelendo (Cohen-Sutherland) atkarpų atkirtimo algoritmas. Jame yra du etapai:

1. Kiekvienos atkarpos galas susiejamas su keturių skilčių dvejetainiu kodu, kuris vadinamas srities kodu. Sritys susietos su atkirtimo lango ribomis. Kodas identifikuoja atkarpos galo vietą lango ribų atžvilgiu. Tam kodo skiltys susiejamos su atkarpos galo padėtimi.
2. Atkarpos galų kodai lyginami tarpusavyje. Jei atkarpos galų koduose toje pačioje skiltyje yra vienetai, tai atkarpa nematoma. Atkarpa yra lange, jei jos abiejų galų kodai 0000. Atkarpa yra neapibrėžta, jei jos galų koduose kurie yra skirtingose skiltyse yra tik po vieną 1. Iš kodų negalima nustatyti, ar atkarpos kerta langą.
3. Daugiakampių atkirtimo algoritmai.

Daugiakampiai gali būti atkertami naudojantis atkarpų atkirtimo algoritmais. Tokie algoritmai atkerta daugiakampio briaunas ir gaunamos tarpusavyje nesusietos atkarpos, kurios priklauso nuo daugiakampio padėties lango atžvilgiu. Daugiakampį atkirtus taisyklingai, gaunamos daugiakampio dalys, . Daugiakampiams atkirsti reikia algoritmo, kuris daugiakampyje išskirtų vieną ar kelias apribotas, į langą patenkančias jo dalis, kurios pakeitimo skleidimu atvaizduotų būdingus užpildytus plotus. Daugiakampių atkirtimo rezultatu turėtų būti daugiakampio briaunas apibrėžiančių viršūnių seka, pateikiama kaip jų sąrašas.

Daugiakampis atkertamas taisyklingai, jei jo briaunos apdorojamos kiekvienos lango briaunos atžvilgiu. Tai atliekama visą daugiakampio viršūnių seką apdorojant Sačelendo ir Hodžmeno (Sutherland-Hodgeman) algoritmu. Pradedama nuo pradinės daugiakampio viršūnių sekos. Daugiakampis atkertamas, pavyzdžiui, kairiosios lango briaunos atžvilgiu. Gaunama nauja daugiakampio viršūnių seka. Ta naujoji viršūnių seka atkertama dešiniosios lango briaunos atžvilgiu ir vėl gaunama nauja viršūnių seka. Panašiai daroma likusių lango briaunų atžvilgiu. Kiekvienu algoritmo žingsniu sudaroma nauja daugiakampio viršūnių seka, kuri kitame žingsnyje naudojama kaip pradiniai atkirtimo kitos lango briaunos atžvilgiu duomenys.



**4 daugiakampio apibrėžiančių viršūnių apdorojimo atvejai**

1. Pirmoji analizuojamos viršūnių poros viršūnė P1 yra lango išorėje, o antroji P2– viduje.



Naujų viršūnių sąrašas =

sankirtos su briauna taškas 

lango viduje esanti viršūnė P2

2. Abi analizuojamos viršūnės - P1 ir P2 yra lange.

Naujų viršūnių sąrašas:

antroji viršūnė P2

3. Pirmoji viršūnė P1 yra lango viduje, o antroji P2 - išorėje.

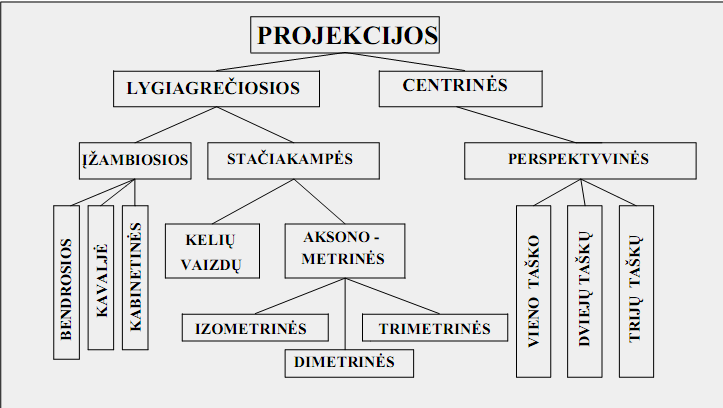
Naujų viršūnių sąrašas:

lango briaunos sankirtos taškas 

4. Abi daugiakampio viršūnės - P1 ir P2 - yra lango išorėje.

Nauju virsuniu nera

1. Plokščiųjų geometrinių projekcijų klasifikacija.



1. Įžambiųjų projekcijų vaizdavimo plokštumose z=0, y=0 arba x=0 skaitmeninio sudarymo transformacijos (*išvedimas*).



Is siu lygciu pertvarke gauname:

xv=x+z(l1cosβ)

yv=y+z(l1sinβ)



1. Perspektyvinių projekcijų vaizdavimo plokštumose z=0, y=0 arba x=0 skaitmeninio sudarymo transformacijos (*išvedimas*).

Iš trikampių CP\*B ir CPA panašumo: ir Iš trikampių COB ir CDA panašumo:





Pertvarke gauname: ; 

1) z=0 yra vaizdavimo plokštuma

2) projektavimo centras ašyje z



Kai projektavimo centras ašyje X ir vaizdavimo plokštuma X=0

Kai projektavimo centras ašyje Y ir vaizdavimo plokštuma Y=0

1. Izometrinių projekcijų vaizdavimo plokštumose z=0, y=0 arba x=0 skaitmeninio sudarymo transformacijos (*išvedimas*).





1. Vaizduojamieji tūriai, jų atkirtimas, pervedimas į kanonį tūrį.
2. lygiagrečiosios projekcijos vaizduojamasis tūris stačiakampis gretasienis
3. perspektyvinės projekcijos vaizduojamasis tūris piramidė

**Normalizuotas/kanoninis tūris**

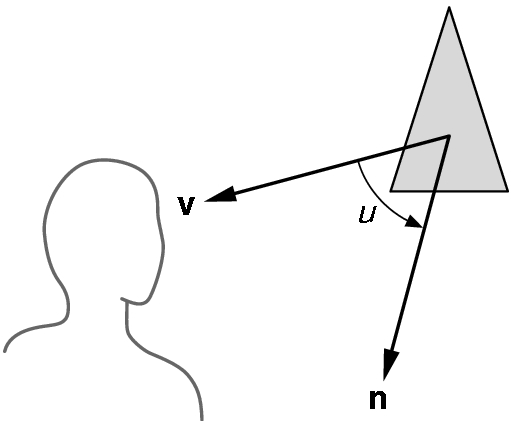
x=0, x=1,

y=0, y=1,

z=0, z=1

**Stačiakampių projekcijų tūrio – stačiakampio gretasienio - pervedimas į kanoninį:**





1. Galinių plokštumų nustatymo metodas.

Taškas (x,y,z) daugiakampio vidinėje pusėje, jei Ax + By + Cz + D < 0.

Čia: A, B ir C – plokštumos lygties koeficientai

Tolesniojo paviršiaus matomumo nustatymas:

v - žiūrėjimo krypties vektorius

n - briaunainio normalės vektorius

u - kampas tarp v ir n

jei n•v ≥ 0 - tai tolesnioji objekto pusė nematoma

Kai žiūrėjimo kryptis lygiagreti su vaizdavimo koordinačių ašimi zV,

⇓

V = (0, 0, VZ)

 ⇓

V N = VZC

Jei daugiakampio C≤0 - daugiakampis nematomas

1. Z-buferio algoritmas.

Populiariausias matomų paviršių nustatymo algoritmas:

* Lengvai realizuojamas
* Suderinamas su nuoseklios vaizdo formavimo architektūros (*grafikos konvejerio*) principais.

Algoritmas:

Kiekvieną daugiakampio paviršiaus poziciją *(x, y, z)* atitinka stačiakampių projekcijų taškas *(x, y)* vaizdo plokštumoje.

Kiekvieno taško *(x, y, z)* atstumas iki projektavimo centro randamas iš z reikšmės.

Šis atstumas lyginamas su šį tašką atitinkančia Z-buferyje

Jeigu atstumas didesnis - tai šis taškas nematomas

Priešingu atveju keičiama reikšmė z-buferyje

**Z buferio realizacija normalizuotosiose koordinatėse**

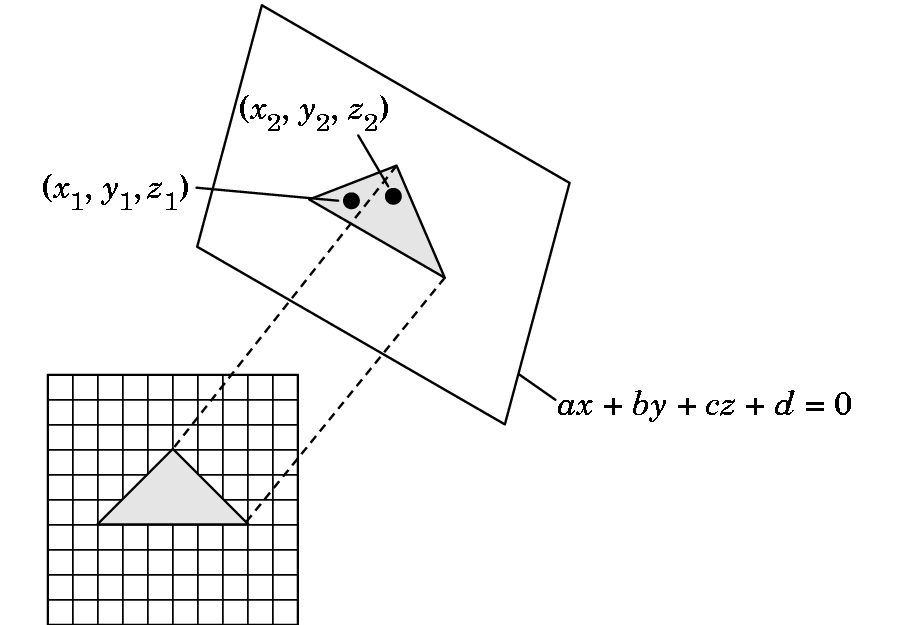
* Z reikšmės gali kisti nuo z=0 iki z=1
* Realizacijai reikalingi 2 buferiai:
  + 1 (*gylio*) – kiekvienos pozicijos gylio reikšmėms saugoti
  + 2 (*vaizdo atnaujinimo*) – visų pozicijų spalvų kodams saugoti

1. Daugiakampio pozicijos (gylio) reikšmės nustatymas Z-buferio algoritme.

Daugiakampio pozicijos (x, y) gylio reikšmės nustatymas iš atitinkamos plokštumos lygties:

(x,y)->z tai 

(x+1,y)->z\* tai  



1. Apšvietimo modelio dedamosios (išvardinti).

**Spalvinimo ir apšvietimo modeliai**

Apšvietimo (spalvinimo) modeliai – šviesos intensyvumui (spalvai) nustatyti

**Pagrindiniai šviesos šaltinių tipai:**

1. Šviesą generuojantys
2. Šviesą atspindintys (sugeriantys)

**Generuojantys:**

1. Foninio apšvietimo (*ambient light*)
2. Taškiniai (*point sources*)
3. Prožektoriai (*spotlights*)
4. Nutolę (*distant light*)

Atspindėjimas

* sklaidusis
* veidrodinis
* difuzinis

1. Plokščiųjų paviršių apšvietimo skaičiavimo modeliai (lygaus spalvinimo, Guru, Fongo).

**1. Lygusis spalvinimas**

Lygaus spalvinimo prielaidos:  
1. Vaizduojamasis objektas yra briaunainis, bet ne objekto kreivųjų paviršių aproksimacija;  
2. Visi šviesos šaltiniai pakankamai toli nuo objekto ir N·L bei šviesos intensyvumo slopimas yra pastovūs;   
3. Stebimas paviršius pakankamai toli nuo stebėtojo ir V·R pastovus.   
**Viršūnės V vienetinė normalė:**



**2. Guru šviesos intensyvumo interpoliavimo metodas**

1. Paviršių normalių skaičiavimas.
2. Nustatoma kiekvienos daugiakampio viršūnės normalė kaip gretimųjų daugiakampių vienetinių normalių vidurkis;
3. Pagal apšvietimo skaičiavimo modelį apskaičiuojamas kiekvienos viršūnės apšvietimo intensyvumas;
4. Daugiakampių spalvinimas tiesiškai interpoliuojant šviesos intensyvumų reikšmes išilgai briaunų ir tarp briaunų.

**Gretimų horizontalių visų skleistinių taškų intensyvumo apskaičiavimo formulė:**

**Taške y:**





**Taške y-1:**



Iš čia:



**3. Fongo spalvinimo metodas interpoliuojant normalių vektorius**

* nustatomas kiekvienos daugiakampio viršūnės gretimųjų daugiakampių vienetinių normalės vektorių vidurkis
* tiesiškai interpoliuojamos viso daugiakampio paviršiaus viršūnių normalių reikšmės;
* pagal apšvietimo skaičiavimo modelį nustatomas kiekvieno skleistinės taško apšvietimo intensyvumas.

Daugiakampių briaunų ir skleistinių sankirtos taškų paviršiaus normalių interpoliavimo išraiškos:



**Fongo metodo privalumai prieš Guru metodą:**

* gaunami tikroviškesni ryškiai apšviestų objektų vaizdai
* sumažinamas Macho juostų efektas
* apdoroja sudėtingesnius daugiakampius

Fongo metodo trūkumai lyginant su Guru metodu

* reikia daugiau skaičiuoti (*kiekvienam pikseliui*)

1. Homogeninių koordinačių naudojimo pranašumai.
2. Lygiagrečiojo/perspektyvinio projektavimo vaizduojamasis tūris ir jo pervedimo į kanoninį (normalizuotąjį) tūrį veiksmų seka.
3. Spalvų modeliai ir jų naudojimo sritys.

Spalvų modelis - priemonė tiksliam konceptualiam ir kokybiškam spalvos aprašymui standartinėmis matematinėmis išraiškomis.

**Spalvų modelių klasifikacija pagal veikimo principą:**

1. Adityviniai (spinduliuojantys) (RGB), *besiremiantys spalvų sumavimu*
2. Subtraktyviniai (sugeriantys) (CMY, CMYK), *besiremiantys spalvų atimtimi*
3. Percepciniai (HSB, HLS, Lab, YCC), *besiremiantys spalvų suvokimu*

**RGB** modelyje pirminės spalvos atidėtos ant vienetinio kubo briaunų. Šio kubo taškas (0,0,0) atitinka juodą (*black*) spalvą, o taškas (1,1,1) – baltą (*white*). Įstrižainė tarp šių taškų vaizduoja visus pilkumo atspalvius. Visas kitas spalvas atitinka taškai kubo viduje. Naudojamas monitoriuose.

**CMY** modelyje pirminės spalvos yra žydra, purpurinė ir geltona (*Cyan, Magenta, Yellow*),. Šiame modelyje, priešingai nei RGB, balta spalva vaizduojama tašku (0,0,0), o juoda – tašku (1,1,1). CMY modelis vartojamas spausdinimo ir braižymo įrenginiuose pigmentų spalvoms parinkti ir įvairiems atspalviams gauti. Trijų pirminių spalvų kombinacijos purškiamos vienos ant kitų ir leidžiama joms susimaišyti prieš išdžiūstant. Tačiau juodai spalvai gauti vartojami juodi dažai, kadangi, sudarant CMY modeliu apibrėžiamą juodą spalvą, gaunama tik tamsiai pilka. Taigi vaizdams spausdinti vartojamas CMYK (*Cyan, Magenta, Yellow, blacK*) modelis

**Percepsiniai spalvų modeliai:**

HSB (HSL) (*liet*. AGS)

HSB (HSL) modelio privalumai:

* Aparatūrinė nepriklausomybė
* Paprastesnis ir intuityviai suprantamesnis spalvos suvokimas

HSB (HSL) modelio trūkumai:

* Spalvų spektro ribotumas
* Abstraktus pobūdis – atspalvis ir sodrumas nurodomi netiesiogiai: per raudoną, žalią ir mėlyną spalvas