1. UVOD

1.1 Čime se sve bavi računarska grafika?

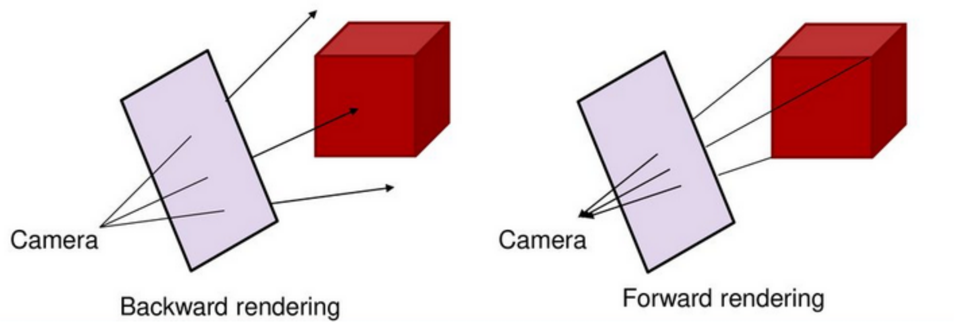
Računarska grafika se bavi pravljenjem modela objekata na sceni (geometrijskog opisa objekata na sceni i opisa kako objekti reflektuju svetlost) i modela osvetljenja scene (matematički opis izvora svetlosne energije, pravaca u kojima se ona emituje, raspodele talasnih dužina svetlosti) i, na osnovu njih, pravljenjem reprezentacija određenog pogleda na scenu (svetlost koja dolazi od nekog imaginarnog oka ili kamere na sceni).

1.2 Koje su osnovne poddiscipline računarske grafike?

- *modelovanje* – bavi se pravljenjem matematičke specifikacije tela i njegovih vizuelnih svojstava na način koji je moguće sačuvati na računaru
- *renderovanje* – to je proces kreiranja realistične dvodimenzione digitalne slike na osnovu (dvodimenzionog ili trodimenzionog) modela i (realističnog ili nerealističnog) modela ponašanja svetlosti
- *animacija* – to je proces kreiranja nizova slika koje, kada se prikažu brzo jedna za drugom, daju utisak glatkog kretanja
- *obrada slika* – bavi se zapisivanjem i obradom slika (kao što su isecanje dela slike, skaliranje slike, kombinovanje više slika, . . .), kao i rekonstrukcijom dvodimenzionih ili trodimenzionih objekata na osnovu njihovih slika.
- *virtuelna realnost* – pokušava da korisnika “ubaci” u 3D virtuelni svet, korišćenjem napredne 3D grafike i naprednih uređaja za prikaz
- *3D skeniranje* – koristi tehnologiju baziranu na pronalaženju opsega za pravljenje merljivih 3D modela
- *računarska fotografija* – korišćenje metoda iz oblasti računarske grafike i obrade slika za omogućavanja novih načina fotografskog pamćenja objekata, scena i okruženja

1.3 Koja je razlika između renderovanja unapred i renderovanja unazad?

Renderovanja unapred predstavlja proces renderovanja koji je najčešće podržan u hardveru i koji koristi OpenGL. U ovoj vrsti renderovanja primitive se transformišu od modela ka uređaju za prikaz.
S druge strane, rejtrejsing algoritam predstavlja primer koncepta *renderovanja unazad* kod koga se kreće od tačke na slici i onda se utvrđuje koje se primitive projektuju na nju.



1.4 Šta se podrazumeva pod tim da je proces modelovanja hijerarhijski?

Modelovanje obuhvata pravljenje modela, primenu materijala na modele, postavljanje modela na scenu, pozicioniranje svetla na sceni, postavljanje kamere. Dijagram stabla obezbeđuje hijerarhijski, vizuelni metod za izražavanje odnosa “sastavljen od”.

1.5 Koje dve paradigme razlikujemo u računrskoj grafici? Koje su njihove osnovne prednosti, a šta su njihovi nedostaci?

Grafika zasnovana na uzorku koristi diskretne uzorke za opisivanje vizuelne informacije. Pikseli se mogu kreirati digitalizacijom slike. U ovom pristupu pikseli su lokacije tačaka sa pridruženim vrednostima uzorka, najšešće intenziteta svetlosti, transparentnosti i drugim kontrolnim informacijama. Kada se slika definiše kao niz piksela, nju je moguće jednostavno izmeniti (izmene kreirane od strane korisnika) i obraditi (algoritamske operacije koje se izvode na slici bez intervencije korisnika). Prednosti ovog pristupa su da kada se slika jednom definiše u terminima boje na (x, y) poziciji mreže, ona se lako može modifikovati izmenom lokacije ili vrednosti boje, informacije o pikselima jedne slike se mogu iskopirati u drugu, zamenom ili kombinovanjem sa prethodnim pikselima. Mane ovog pristupa su što ne postoje dodatne informacije već važi paradigma What You See Is All You Get.

Grafika zasnovana na geometriji (skalabilna vektorska grafika ili objektno-orijentisana grafika) - kreiraju se i čuvaju matematički opisi ili modeli geometrijskih elemenata i pridruženih atributa i onda se oni uzorkuju za vizuelizaciju (vrši se rasterizacija). Korisnik najčešće ne može da direktno radi nad individualnim pikselima u geometrijski zasnovanim programima, već dok korisnik radi sa geometrijskim elementima, program iznova uzorkuje i prikazuje elemente. Renderovanje sve više kombinuje geometrijski zasnovanu grafiku i grafiku zasnovanu na uzorku, da bi se povećao kvalitet finalnog proizvoda.

1.6 Šta označava pojam interaktivne računarske grafike?

Interaktivna računarska grafika podrazumeva dinamički način prikaza slike na računaru uz aktivno učešće čoveka u stvaranju i izmeni slike, gde su rezultati odmah vidljivi. Korisnik kontroliše sadržaj, strukturu i izgled objekata i njihovih slika koje se prikazuju korišćenjem brze vizuelne povratne informacije.

1.7 Koja je razlika između adresivosti i rezolucije uređaja za prikaz?

Adresivost je broj pojedinačnih tačaka po inču koje mogu biti kreirane. Može da se razlikuje horizontalna adresivost i vertikalna adresivost.
Rezolucija je broj razlučivih od strane posmatrača ili uređaja različitih linija po inču (na primer, naizmenično crnih i belih) koje uređaj može da kreira. Rezolucija ne može biti veća od adresivosti.

1.8 Kako se generiše slika kod vektorskih sistema?

Kod njih se linija dobija tako što se digitalne koordinate krajnjih tačaka transformišu u analogni napon za elektronski zrak koji pada na površinu ekrana. Slika se osvežava obično 30 do 60 puta u sekundi (30-60 Hz). Na vektorskim monitorima moguće je iscrtati i do 100000 linija. Pritom su linije glatke, ali se obojene površine teško prikazuju. Pogodni su samo za mrežne modele.

1.9 Kakav izgled imaju kose linije u vektorskim, a kakav u rasterskim sistemima?

Kod vektorskih sistema su linije glatke, a u rasterskim sistemima kose linije su „stepenaste“.

1.10 Čemu služi frejm bafer? Čemu služi video kontroler?

Na *frejm bafer* se može gledati kao na računarsku memoriju organizovanu u vidu dvodimenzionog niza tako da svaka adresibilna lokacija (x, y) odgovara jednom pikselu. U njemu se čuva sadržaj koji se zatim prikazuje na ekranu.

Video kontroler pristupa frejm baferu i prikazuje liniju po liniju na ekranu. On je zadužen da stalno osvežava sadržaj ekrana.

1.11 Navesti bar četiri oblasti u kojima se koristi računarska grafika.

Grafički korisnički interfejsi, Interaktivna izrada crteža u nauci i tehnologiji, CAD i CAM, Simulacije, Industrija zabave, Industrijski dizajn, Virtuelna realnost…

2. RASTERIZACIJA

2.1 Šta je rasterizacija?
Rasterizacija je konvertovanje projektovane primitive u skup piksela.

2.2 Kako bi izgledala vektorska, a kako rasterska slika dva pravougaonika sa stranicama paralelnim koordinatnim osama? Koja od ove dve slike bi zauzimala manje memorije?
Kod vektorske grafike je slika predstavljena kao kolekcija geometrijskih figura, a kod rasterske je ekranu pridružena matrica piksela.
Vektorska slika bi zauzimala manje memorije, zbog rezolucije i bitske dubine kod rasterske slike.

2.3 Navesti primer gde se u praksi koristi vektorska grafika, a gde rasterska.
Vektorska se koristi za fontove, logotipove, štampani materijal, animacije… Rasterska se koristi za fotografije, u veb dizajnu...

2.4 Kako ocenjujemo kvalitet dobijene rasterske slike?
Kvalitet jedne rasterske slike određuje ukupan broj piksela (rezolucija) kao i broj vrednosti za svaki pojedinačni piksel (dubina boje).

2.5 Koji su zahtevi kod crtanja linije?
Niz piksela treba da bude što bliži idealnoj liniji i treba da prolazi kroz krajnje tačke. Slika duži treba da bude nezavisna od redosleda kojim su date njene krajnje tačke. Crtanje treba da bude što je moguće brže. Sve linije treba da budu iste osvetljenosti i sve tačke svake linije treba da budu iste osvetljenosti (ili da tako izgledaju) bez obzira na njihov nagib i dužinu.

2.6 Ako je koeficijent pravca prave (a) m = 1/2, (b) m = 3, da li crtamo po jedan piksel u svakom redu ili koloni?
a) $m = ½ < 1$ → po jedan piksel u svakoj koloni
b) $m = 3 > 1$ → po jedan piksel u svakom redu

2.7 Koliko piksela (računajući i krajnje tačke) treba da bude uključeno ukoliko na rasterskom uređaju treba linijom spojiti tačke: (a) (1,2) i (53,30) (b) (1,2) i (22,50) (c) (1,2) i (31,32)
a) max(53-1, 30-2) + 1 = 53
b) max(22-1, 50-2) + 1 = 49
c) max(31-1, 32-2) + 1 = 31

2.8 Kako radi algoritam grube sile za crtanje duži?
Najjednostavnija strategija za crtanje duži određene tačkama (x_0, y_0) i (x_1, y_1) sastoji se od narednih koraka:

- izračuna se koeficijent pravca: $m = \frac{(y1 - y0)}{(x1 - x0)}$
- x se povećava za 1 počev od x koordinate početne tačke $(x_0\ y_0)$ duži
- za svako x_i računa se vrednost $y_i = mx_i + B = m(x_i - x_0) + y_0$
- osvetljava se piksel sa koordinatama $(x_i, \mathrm{round}(y_i))$, pri čemu je $\mathrm{round}(y_i) = \lfloor y_i + 0.5 \rfloor$

2.9 Koji su nedostaci algoritma grube sile za crtanje duži? Na koji način se oni mogu eliminisati?

Ako je $x_1 < x_0$ ništa se ne crta (rešenje: promeniti poredak tačaka ukoliko je $x_1 < x_0$). Algoritam je neefikasan zbog broja operacija i korišćenja realne aritmetike - svaki korak zahteva oduzimanje, sabiranje, množenje i zaokruživanje jer je y promenljiva realnog tipa.

2.10 Koja je ideja inkrementalnih algoritama?

U svakom koraku, izračunavanja pravimo na osnovu prethodnog koraka.

2.11 Kako funkcioniše midpoint algoritam za crtanje duži?

Bresenham je razvio algoritam za crtanje duži koji koristi samo celobrojnu aritmetiku. Ista tehnika može da se koristi i za krug. Pitteway i Van Aken su razvili midpoint algoritam koji se za duži i celobrojne krugove ponaša isto kao Bresenhamov algoritam, ali može da se uopšti na proizvoljne krive drugog reda.

Pretpostavimo da smo označili piksel P (x_p, y_p) i da treba da izaberemo između piksela jednu poziciju desno (E=east) i piksela jednu poziciju desno i jednu poziciju iznad (NE=northeast).

2.12 Koja je ključna ideja midpoint algoritma za crtanje duži?

Odluku pravimo između tačno 2 piksela.

2.13 Čemu nam u midpoint algoritmu služi tačka M između tačaka N i NE?

Ako je tačka M “ispod” prave koja sadrži duž, onda je pravoj bliža NE, a inače E.

2.14 Šta je promenljiva odlučivanja?

Da bismo odredili položaj tačke M potrebno je jedino odrediti znak izraza $F(M) = F(x_p+1, y_p+ ½)$. S obzirom na to da se odluka bazira na vrednosti ove funkcije, vrednost $d = F(x_p+1, y_p+ ½)$ zovemo promenljiva odlučivanja. Ako je $d < 0$, onda se tačka M nalazi iznad prave, te biramo piksel E; ako je $d > 0$, onda se tačka M nalazi ispod prave i biramo piksel NE; ako je $d = 0$, onda je sve jedno i, po dogovoru, biramo piksel E.

Promenljiva odlučivanja je vrednost implicitne jednačine prave u središnjoj tački.

2.15 Zašto prvu vrednost promenljive odlučivanja množimo sa 2 u realizaciji midpoint algoritma?

Da bismo dobili celobrojnu vrednost.

2.16 Izračunaj prvu vrednost promenljive odlučivanja za duž koja povezuje tačke sa koordinatama (1,2) i (7,4).

$d_{start} = 2*dy – dx = 2*2 – 6 = 4 – 6 = -2$

2.17 Ako se midpoint algoritmom iscrtava duž između tačaka (2,4) i (12,9), čemu je jednaka početna vrednost promenljive odlučivanja?

$d_{start} = 2*dy – dx = 2*10 – 5 = 20 – 5 = 15$

2.18 Kako funkcioniše algoritam grube sile za iscrtavanje kruga? Koji su njegovi nedostaci?

· Iz jednačine kruga: $x^2 + y^2 = R^2$, sledi: $y=\pm\sqrt{R^2-x^2}$

· Za i = 1, 2, 3,…, pri čemu se x_i inkrementira za po 1, treba osvetliti tačku: $(x_i, [\sqrt{R^2-x_i^2}+0.5])$

Tačno (koliko je to moguće), ali neefikasno.

· Sličan neefikasan metod, korišćenjem parametarske jednačine kruga: crtati tačke

$([Rcos\varphi_i+0.5],[Rsin\varphi_i+0.5])$ za $\varphi_i = 1^\circ, 2^\circ, 3^\circ, ..., 90^\circ$.

· Kako vrednost x raste, povećavaju se praznine između tačaka; to je lako rešiti korišćenjem i simetrije kruga po još jednoj osi (pravoj y = x).

2.19 Šta se menja ukoliko krug nije u koordinatnom početku?

Razlikuje se samo početni piksel – umesto (0, R), prvi uključeni piksel treba da bude (a, R+b), a umesto uslova y > x, treba koristiti uslov y-b > x-a, ili efikasnije y > x+(b-a).

2.20 Zašto se u midpoint algoritmu za crtanje kruga razmatra samo osmina kruga?

Ostali slučajevi se obrađuju simetrično; efikasnije.

2.21 Koju vrednost u algoritmu rasterizacije kruga zovemo promenljiva odlučivanja?

$d = F(x_p+1, y_p- ½) = (x_p+1)^2 + (y_p- ½)^2 – R^2$

2.22 Koja je suštinska razlika u vrednosti promenljive odlučivanja kod crtanja kruga u odnosu na crtanje duži?

Prva vrednost za d kod crtanja kruga nije celobrojna, a kod crtanja duži jeste.

2.23 U unapređenoj verziji midpoint algoritma za crtanje kruga, zašto je neophodno da svaki put ažuriramo obe promenljive ΔE i ΔSE?

Ne znamo koja će nam biti potrebna u daljem radu.

2.24 Koje veličine nazivamo razlikama drugog reda?

Vrednosti za koje se uvećavaju promenljive ΔE i ΔSE su razlike drugog reda.

2.25 Koja je vremenska složenost midpoint algoritma za crtanje: (a) duži, (b) kruga?

a) linearna po broju piksela koji će biti iscrtani

b) linearna u funkciji poluprečnika kruga

3. POPUNJAVANJE

3.1 Kada za region kažemo da je 4-povezan, a kada da je 8-povezan?

Kažemo da je region *4-povezan* ako se svaka dva piksela tog regiona mogu povezati nizom piksela korišćenjem samo poteza nagore, nadole, ulevo i udesno. Nasuprot ovome, region je *8-povezan* ako se svaka dva piksela tog regiona mogu povezati nizom piksela korišćenjem poteza nagore, nadole, ulevo, udesno, gore-levo, gore-desno, dole-levo, dole-desno.

3.2 Kako se zadaje region definisan unutrašnjošću, a kako region definisan granicom?

Region definisan unutrašnjošću je najveći povezani region piksela čija je boja ista kao boja piksela P. Region definisan granicom je najveći povezani region piksela čija boja nije jednaka nekoj graničnoj vrednosti boje.

3.3 Na kakve regione se primenjuje flood-fill algoritam a na kakve boundary-fill algoritam? Kako svaki od njih funkcioniše?

FloodFill – algoritam za popunjavanje regiona definisan unutrašnjošću. Kreće se od početnog piksela u unutrašnjosti (seed-a), proverava se da li je boja piksela stara, ako jeste piksel se boji novom bojom i pomeramo se iz ovog piksela u 4 (ili 8) smerova. Granica oblasti se može sastojati od različitih boja. Ovakva implementacija je praktično neupotrebljiva.
BoundaryFill – algoritam za popunjavanje regiona definisanog granicama. Kreće se od početnog piksela u unutrašnjosti, proverava se da li je boja piksela različita od granične vrednosti boje i od nove vrednosti boje, ako jeste piksel se boji novom bojom i pomeramo se iz ovog piksela u 4 (ili 8) smerova. Rekurzivna priroda algoritma može prouzrokovati prerani prekid rada ako se nova boja pojavi u regionu koji je definisan granicom.

3.4 Navesti primer gde boundary-fill algoritam neće ispravno obojiti region.

Neće dati dobar rezultat ako je unutrašnjost regiona podeljena na 2 dela (granica koja ih deli je drugačije boje od unutrašnjosti).

3.5 Šta je scan linija? Koliko scan linija se razmatra u algoritmu za scan popunjavanje poligona?

Scan linija je horizontalna linija na rasterskom sistemu.

3.6 Kod scan popunjavanja poligona koje se presečne tačke scan linije sa stranicama poligona broje, a koje ne?

Kod tačaka sa neparnim stanjem treba započeti bojenje, a kod tačaka sa parnim treba ga zaustaviti.

3.7 Na koji način obrađujemo scan linije koje prolaze kroz neko teme poligona?

Za stranicu poligona računamo tačku preseka sa najmanjom y koordinatom, a ne računamo tačku preseka sa najvećom y koordinatom parnost se menja osim ako je u pitanju lokalni minimum/maksimum.

3.8 Koje se tačke boje kod horizontalnih stranica poligona?

Kod horizontalnih preseka ne bojimo nijednu tačku preseka. Efekat ovog pravila je da se donje horizontalne stranice boje, a gornje ne; takođe ne boje se i desne tačke stranica.

3.9 Šta je sliver?

Sliver je uska poligonalna oblast koja sadrži 0 ili 1 piksel za neke scan linije. Ovo je posledica toga da se boje samo pikseli koji su u unutrašnjosti ili na levim ili donjim stranicama poligona. Antialiasing tehnika se može koristiti za rešavanje ovog problema.

3.10 Kako se računa presek (i+1)-ve scan linije na osnovu na i-te scan linije?

Koristi se inkrementalni pristup da bi se izračunali preseci jedne scan linije na osnovu prethodne scan linije, bez toga da analitički računamo presek scan linije sa svakom stranicom poligona. Ideja je nalik midpoint algoritmu, samo je potrebno uvek birati tačke u unutrašnjosti poligona.

3.11 Na koji način prelazimo na celobrojnu aritmetiku – koje dve celobrojne vrednosti razmatramo i na koji način vršimo obradu?

Na celobrojnu aritmetiku prelazimo određivanjem preseka scan linije sa levom stranicom poligona. Razmatramo slučaj kada je m > 1 i razmatramo posebno celi i razlomljeni deo vrednosti:

$$x_i + \frac{x_{max} - x_{min}}{y_{max} - y_{min}}$$

3.12 Koja je vremenska složenost algoritma LeftEdgeScan?

Linearna, $O(y_{max}-y_{min})$

3.13 Koje se tačke preseka dobijaju algoritmom LeftEdgeScan za duž sa krajnjim tačkama (2, 3) i (5, 8)?

Iz algoritma: (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8)

3.14 Zašto u algoritmu LeftEdgeScan vrednost promenljive inkrement kreće od vrednosti imenilac?

To nam je važno za zaokruživanje na unutrašnji piksel. Nakon što mu dodamo brojijlac, ispitujemo da li je došlo do prekoračenja i zaokružujemo na sledeći piksel.

4. ODSECANJE

4.1 Na kakav kliping region se primenjuje Cohen-Sutherland-ov algoritam?
Na pravougaonik.

4.2 Na koliko regiona se u ovom algoritmu deli ravan? Šta svaki od bitova u 4-bitnom kodu tačke označava?
Na 9 regiona. Prvi bit - prava se nalazi iznad gornje granice regiona; drugi bit – ispod donje granice; treći bit – desno od desne granice; četvrti bit – nalazi se levo od leve granice kliping regiona.

4.3 Koji kôd imaju tačke u unutrašnjosti kliping regiona?
0000

4.4 Kod Cohen-Sutherlandovog algoritma šta važi za duž koja se secka ako je $K_1=0$ i $K_2=0$? Šta važi ako je $K_1\&K_2 <> 0$?
Za prvu važi da se nalazi unutar kliping regiona, a za drugu da se nalazi van i trivijalno se odbacuje.

4.5 Koja je vremenska složenost Cohen-Sutherland algoritma?
Konstantna složenost.

4.6 Na kakve kliping regione se može primeniti Cyrus-Backov algoritam?
Primenljiv je na proizvoljni konveksni kliping poligon.

4.7 Koji uslov važi za presečnu tačku duži P_0P_1 i normalu stranice?
$N \cdot (P(t) - P_E) = 0$

4.8 Da bi se tačka preseka klasifikovala kao potencijalno ulazna/izlazna razmatra se odnos koja dva vektora?
Vektora $D = P_0P_1$ i vektora N – normala stranice poligona usmerena ka spoljašnosti poligona.

4.9 Šta treba da važi da bi tačka bila okarakterisana kao potencijalno ulazna/izlazna?
Potencijalno ulazna: $N \cdot D < 0$; Potencijalno izlazna: $N \cdot D > 0$, $D = P_0P_1$

4.10 Šta znači kada je $t_L < t_E$?
Tada ne postoji presek.

4.11 Koja je vremenska složenost Cyrus-Beckovog algoritma?
Složenost je linearna po broju temena poligona.

4.12 Na kakav se region može primeniti Liang-Barsky varijanta algoritma za kliping?
Na pravougaonik sa horizontalnim/vertikalnim stranicama.

5. GEOMETRIJSKE TRANSFORMACIJE

5.1 Kako se može jednostavno utvrditi da li duž P_0P_1 seče ravan datu jednačinom Ax + By + Cz + D = 0?

Zamenom P_0 i P_0 u jednačinu ravni, treba dobiti rešenja suprotnog znaka.

5.2 Iz kog razloga se prelazi na rad sa homogenim koordinatama? Navesti bar dve različite reprezentacije Dekartove tačke (−3, 2) homogenim koordinatama.

Bilo bi dobro da translacija, skaliranje, smicanje i rotacija imaju istu formu. Koristeći homogene koordinate sve ove transformacije imaju formu množenja matrica. Uz pomoć homogenih koordinata 2D tačka se predstavlja trojkom vrednosti (x, y, W).
(-3, 2, 6) ili (-3/6, 2/6, 1)=(½, 1/3, 1)

5.3 Koja je transformacija inverzna smicanju za faktor *a* po x osi?

Smicanje za faktor -*a*.

5.4 Izvesti matricu smicanja u 3D u smeru koordinatne ravni Oxy.

$$\begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.5 Navesti primer situacije u kojoj je pogodno izvršiti promenu koordinatnog sistema.

Ovaj pristup je pogodniji kada više objekata u svojim pojedinačnim koordinatnim sistemima treba objediniti u jedinstven koordinatni sistem.

5.6 Od čega se sastoji graf scene?

Čvorova objekata (kocke, valjci, lopte, . . .) koji su podrazumevano jedinične veličine i pozicionirani u koordinatnom početku; čvorova atributa (boja, tekstura, . . .); čvorova transformacija.

5.7 Čemu služe čvorovi grupisanja u grafu scene?

Ako imamo više sličnih komponenti na sceni, moguće je ponovo iskoristiti već definisane grupe objekata.

5.8 Čemu je jednaka matrica skaliranja? Da li se skaliranjem čuvaju uglovi između pravih u ravni?

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Ne čuvaju se uglovi, osim kada je Sx = Sy.

5.9 Kako izgleda matrica rotacije oko koordinatnog početka za ugao φ?

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

5.10 Kako se homogenim koordinatama zadaje tačka u 2D?

Tačka se predstavlja trojkom vrednosti (x, y, w).

5.11 Koja je to tačka sa homogenim koordinatama (x, y, 0)?

Beskonačno daleka tačka u pravcu (x, y).

5.12 Čemu u Dekartovom prostoru pripadaju sve homogenizovane koordinate tačaka?

Pripadaju jednoj pravoj tog prostora.

5.13 Kako u homogenim koordinatama predstaviti naredne transformacije: skaliranje, rotaciju, transliranje?

Skaliranje, rotacija: $\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$ Translacija: $\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$

5.14 Šta važi za afine transformacije?

Čuvaju kolinearnost, odnose rastojanja između kolinearnih tačaka i paralelnost, ali ne čuvaju nužno uglove i dužine. To su translacija, rotacija, skaliranje i smicanje.

5.15 Koja je transformacija inverzna skaliranju/translaciji/rotaciji?

Skaliranje za faktore 1/Sx i 1/Sy; translacija za vektor $\begin{bmatrix} -tx \\ -ty \end{bmatrix}$; rotacija za ugao -φ.

5.16 Šta je kompozicija više skaliranja/translacija/rotacija?

Skaliranje; translacija; rotacija.

5.17 Šta je smicanje?

Transformacija pri kojoj objekat treba iskositi u smeru neke koordinatne ose.

5.18 Kog je oblika matrica izometrijskih transformacija?

$$\begin{bmatrix} a & b & tx \\ c & d & ty \\ 0 & 0 & 1 \end{bmatrix}$$

5.19 Koje transformacije su izometrijske?

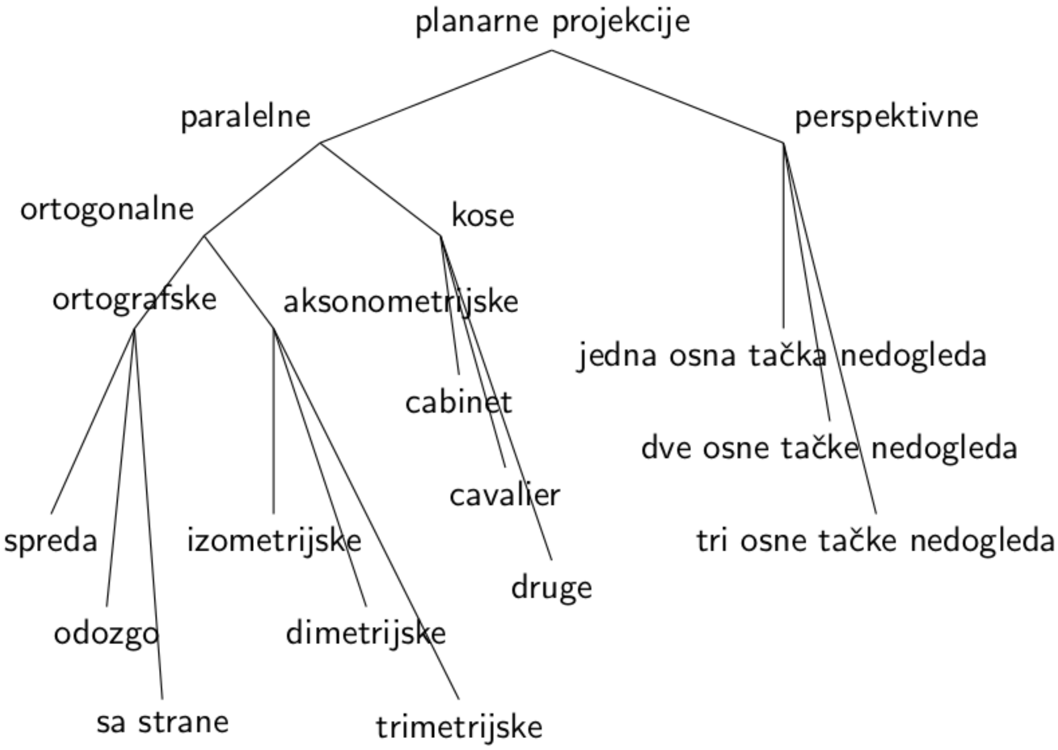
Translacija i rotacija.

5.20 Kako se može predstaviti svaka afina transformacija?

$$\begin{bmatrix} a & b & tx \\ c & d & ty \\ 0 & 0 & 1 \end{bmatrix}$$

6. PROJEKTOVANJE

6.1 Skicirati klasifikaciju 3D planarnih projekcija.



6.2 Koje paralelne prave se seku kod perspektivne projekcije? Kako zovemo tačku u kojoj se one seku?

Paralelne prave koje nisu paralelne ravi projekcije u tački nedogleda.

6.3 Kod koje projekcije je ravan projekcije upravna na koordinatnu osu: (a) ortografska projekcija (b) izometrijska projekcija (c) cavalier projekcija?

Ortografske, cavalier.

6.4 Kod koje projekcije je projekciona ravan upravna na pravac projekcije: (a) aksonometrijska projekcija (b) ortografska projekcija (c) cabinet projekcija?

Aksonometrijske.

6.5 Kod koje projekcije normala ravni projekcije zahvata jednak ugao sa sve tri koordinatne ose? (a) kod aksonometrijske (b) kod izometrijske (c) kod trimetrijske projekcije

Izometrijske.

6.6 Šta je projekcija?

Projekcija je preslikavanje iz koordinatnog sistema dimenzije *n* u koordinatni sistem dimenzije manje od *n*.

6.7 Čime je određena projekcija?

Centrom projekcije i ravni projekcije.

6.8 Kako se mogu klasifikovati 3D planarne projekcije? U odnosu na šta se vrši klasifikacija?

Perspektivne i paralelne. U odnosu na to šta se zadaje - pravac ili centar projekcije.

6.9 Koje su karakteristike perspektivne projekcije?

Centar projekcije je na konačnom rastojanju od ravni projekcije.

6.10 Kako zovemo tačku u kojoj se seku prave paralelne koordinatnoj osi? Koliko najviše takvih tačaka može postojati?

Osna tačka nedogleda. Ima ih najviše 3.

6.11 Prema čemu se dele paralelne projekcije i koje dve vrste paralelnih projekcija razlikujemo?

Dele se prema odnosu pravca projekcije i normale ravni projekcije. Razlikujemo ortogonalne i kose projekcije.

6.12 Kako se dele ortogonalne projekcije u odnosu na ugao koji zahvata ravan projekcije sa koordinatnim osama?

Na ortografske i aksonometrijske.

6.13 Koliko tipova ortografskih projekcija razlikujemo?

3 tipa: pogled spreda, pogled odozgo, pogled sa strane.

6.14 Kako se dele aksonometrijske projekcije i šta važi za svaki od tipova?

Dele se na osnovu vrednosti uglova koje normala ravni projekcije zahvata sa koordinatnim osama: izometrijska (jednak ugao sa svim koordinarnim osama), dimetrijska (isti ugao sa 2 koordinatne ose), trimetrijska (međusobno različiti uglovi)

6.15 Šta važi za vektor normale ravni projekcije kod izometrijskog projektovanja?

$|n_1| = |n_2| = |n_3|$ - svim jednakim dužinama odgovaraju jednake dužine projekcija

6.16 Šta važi za ravan projekcije, a šta za pravac projektovanja kod kosog projetovanja?

Ravan projekcije je upravna na neku od koordinatnih osa, a pravac projekcije nije jednak normali ravni projekcije.

6.17 Nabrojati prednosti i mane kosog projektovanja.

Prednosti: može da predstavi tačan oblik jedne strane objekta; nedostatak perspektivnog skraćenja olakšava poređenje veličina; donekle daje utisak 3D izgleda objekta.
Mane: objekti mogu da izgledaju iskrivljeno ukoliko se pozicija ravni projekcije ne izabere pažljivo; nedostatak skraćenja daje nerealističan izgled.

6.18 Kako se bira ravan projekcije kod kosog projektovanja?

Ravan projekcije treba da bude paralelna strani objekta koja je najmanje pravilna, odnosno strani koja sadrži najveći broj zakrivljenih površina; treba da bude paralelna najdužoj glavnoj strani objekta; treba da bude paralelna strani od interesa.

6.19 Koja su dva najčešća tipa kosih projekcija? Koliko je ugao između pravca projektovanja i normale ravni projektovanja kod svakog od ovih tipova?

Cavalier i cabinet. Ugao kod cavalier-a je 45°, a kod cabinet-a je 63.4°.

6.20 Koliku dužinu imaju projekcije duži normalnih na ravan projekcije kod cavalier projektovanja, a koliku kod cabinet projektovanja?

Kod cavalier - iste dužine kao i same ravni, nema skraćivanja. kod cabinet - imaju 2 puta manju dužinu u odnosu na same duži (realističniji prikaz).

6.21 Po čemu se međusobno mogu razlikovati cabinet projekcije?

Po uglu koji zahvata pravac projekcije sa koordinatnim osama.

7. SINTETIČKI MODEL KAMERE

7.1 Nabrojati 6 parametara sintetičkog modela kamere.
Pozicija kamere (tačka), vektor pogleda i vektor nagore (2 vektora), rastojanje prednje i zadnje ravni odsecanja (2 skalara), vidno polje (2 ugla), žižna daljina, dubina vidnog polja.

7.2 Kakav je odnos tačkaste kamere i modela sintetičke kamere?
Sintetička kamera je nalik tačkastoj kameri - ima malu rupu kroz koju ulaze zraci umesto sočiva, ravan slike je iza rupe, slika scene je obrnuta.

7.3 Šta se zadaje vektorom pogleda, a šta vektorom nagore?
Vektor pogleda - zadaje se smer u kojem je kamera usmerena. Vektor nagore - kako se kamera rotira oko vektora pogleda.

7.4 Šta se zadaje uglovima vidnog polja?
Do kog ugla, gledano u odnosu na vektor pogleda izraženo u stepenima, kamera može da vidi.

7.5 Da li vektor nagore mora biti upravan na vektor pogleda?
Ne.

7.6 Čemu služe prednja i zadnja ravan odsecanja i zašto je bitno da ih definišemo? Kako se one zadaju?
To su paralelne ravni projektovanja i zadaju se njihovim rastojanjem od pozicije kamere. Isecaju zarubljenu četvorostranu piramidu pogleda. Objekti koji se nalaze u okviru ove oblasti biće prikazani na slici, objekti van oblasti neće, a za objekte koji presecaju strane ove oblasti radi se odsecanje.

7.7 Kojim se telom definiše zapremina pogleda kod paralelnog projektovanja? Koji se parametri menjaju u odnosu na perspektivno projektovanje?
Paralelopipedom. Umesto uglova vidnog polja potrebno je zadati visinu i širinu zapremine pogleda.

7.8 Kako izgleda standardna perspektivna zapremina pogleda?
Piramida koja ima granice od -1 do 1 po x i y osi, i od 0 do -1 po z osi.

7.9 Koja je jednačina prednje ravni odsecanje kod standardne perspektivne zapremine pogleda?
$z = -n/f$

7.10 Kako izgleda standardna paralelna zapremina pogleda?
Pravougaoni paralelopiped čiije granice idu od -1 do 1 po x i y osi, i od 0 do -1 po z osi.

7.11 Iz kojih se koraka sastoji prevođenje 3D svetskih koordinata primitiva u 2D koordinate na uređaju za prikaz?

Prvo se geometrijski modeli postavljaju na 3D scenu različitim geometrijskim transformacijama. Nakon toga se ovi modeli posmatraju kroz kameru što rezultuje transformacijom njihovih svetskih koordinata u koordinate u standardnoj perspektivnoj zapremini pogleda, a nakon toga se transformišu u koordinate u standardnoj paralelnoj zapremini pogleda. Konačno, ovi modeli se projektuju u 2D sliku i ova slika se transformiše u prozor za prikaz.

8. REPREZENTACIJA FIGURA

8.1 Zašto se prilikom modelovanja figura pomoću mreža za primitive najčešće biraju trouglovi, a ne poligoni sa većim brojem stranica?

Geometrija je jednostavnija i trougao uvek leži u ravni.

8.2 Kako predstavljamo mrežu u 2D?

Kao kolekciju trouglova u 3D prostoru.

8.3 Kako se definiše granica 2D mreže?

Granica 2D mreže jednaka je sumi granica njenih trouglova.

8.4 Kada kažemo da je 1D mreža zatvorena?

Kada je vrednost granice nula.

8.5 Kada za neku 1D mrežu kažemo da je mreža mnogostrukosti?

Ako je stepen svakog temena 2.

8.6 Koje je složenosti operacija dodavanja novog temena u 1D mrežu, a koje složenosti operacija dodavanja nove ivice?

O(1); O(e)

8.7 Koja se reprezentacija koristi za predstavljanje mreže u 3D?

Pored liste temena i liste trouglova često se čuva i lista susedstva.

8.8 Kada za 2D mrežu kažemo da je mreža mnogostrukosti? Da li je kod njih dozvoljeno dodati novi trougao u mrežu? Da li je dozvoljeno obrisati neki trougao iz mreže?

2D mreža je mreža mnogostrukosti ako se ivice i trouglovi čije je jedno teme *v* mogu poređati u cikličnom poretku: t_1, e_1, t_2, e_2,..., t_n, e_n bez ponavljanja tako da je ivica *e* i stranica trouglova t_i i t_{i+1} (indeksi se računaju po modulu *n*).
Nisu dozvoljena dodavanja novih trouglova, kao ni brisanja.

8.9 Za koju mrežu kažemo da je mreža mnogostrukosti sa granicom?

Dozvoljavamo da se unutar mreže mnogostrukosti javi granično teme i granična ivica.

8.10 Čemu je jednaka vrednost granice kod (a) orijentisanih mreža mnogostrukosti bez granice (b) orijentisanih mreža mnogostrukosti sa granicom?

Za orijentisane mreže mnogostrukosti vrednost granice je 0. Za orijentisane mreže mnogostrukosti sa granicom, granica će se sastojati od tačno onih ivica koje smo identifikovali kao granične.

8.11 Opisati winged edge strukturu podataka. Za šta se ona koristi?

Pogodna je za predstavljanje mreža kod kojih strane nisu trouglovi. Ivice se smatraju građanima prvog reda. Za svaku ivicu čuvaju se informacije o: dva temena koje ivica povezuje; dve strane kojima ivica pripada; prethodnoj i sledećoj ivici prilikom obilaska u pozitivnom smeru leve, odnosno desne strane kojoj ivica pripada. Za svako teme i za svaku stranu čuvaju se informacije o proizvoljnoj ivici koja im pripada.

8.12 Kako glasi Ojlerova formula (a) za površi bez rupa (b) za površi sa rupama?

a) $V - E + F = 2$, V broj temena, E broj ivica, a F broj strana površi
b) $V - E + F = 2 - 2G$

8.13 Opisati operacije sažimanja ivice i zamene ivice.

Ivica se sažima dok joj dužina ne postane 0. Prilikom stapanja dva temena treba izabrati lokaciju za novo teme. Lokacija novog temena zavisi od željenog cilja: ako je cilj minimizovati račun, može se zadržati jedno od starih temena; ako želimo da očuvamo oblik, može se izabrati središte duži određene ovim temenima; ako uprosečavanje pomera puno tačaka a to nije poželjno, nova tačka se može izabrati tako da se minimizuje maksimalno rastojanje tačaka stare mreže od nove tačke mreže.
Zamena ivice transformiše dva dugačka i uska trougla u dva skoro jednakostranična trougla.

9. REPREZENTACIJA KRIVIH U 3D

9.1 Da li je dobro čuvati krive u rasterskom obliku?
Rasterski format nije pogodan: alijasing, nema skalabilnosti, memorijski zahtevno.

9.2 Zašto aproksimacija polinomima stepena 3 predstavlja pogodan izbor?
Polinomi nižeg stepena nisu dovoljno fleksibilni. Polinomi visokog stepena zahtevaju više računa. Najmanji stepen takav da kriva ne pripada nužno ravni.

9.3 Navesti tri različita načina na koje je moguće zadati krivu.
Interpolacija, Hermitova kriva i Bezjeova kriva.

9.4 Definisati Hermitovu krivu.
Za date pozicije P i Q i vektore brzina v i w odrediti funkciju $\gamma : [0, 1] \to R^2$ tako da je:
$\gamma(0) = P$, $\gamma(1) = Q$, $\gamma'(0) = v$, $\gamma'(1) = w$. Rezultujuća kriva naziva se Hermitova kriva za ulaz P, Q, v,w

9.5 Šta je karakteristično za Hermitovu bazu?
U slučaju kada hoćemo da izmenimo samo početnu tačku (ili analogno neki drugi od datih paramatera), ako se koristi Hermitova baza potrebno je izmeniti samo koeficijent uz jedan polinom.

9.6 Šta je geometrijska matrica Hermitove krive i čime je određena?
$$\begin{bmatrix} P1 & P2 & P3 & P4 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}$$

9.7 Definisati Bezijerovu krivu.
Bezjeova kriva se gradi nad 4 tačke P_1, P_2, P_3, P_4. Kriva počinje u tački P_1, završava se u tački P_4, ima inicijalnu brzinu $3(P_2 - P_1)$ i završnu brzinu $3(P_4 - P_3)$.

9.8 Šta je splajn?
Splajn je parametarska kriva vođena kontrolnim tačkama ili kontrolnim vektorima.

9.9 Da li je kod Catmull-Rom splajna dat vektor tangente u svakoj od kontrolnih tačaka?
Ne.

9.10 Navesti razlike između Catmull-Rom splajna i B-splajna.
Razlikuju se po tome što su kubni B-splajnovi: C^2 glatki, tj. i prvi i drugi izvod su neprekidne funkcije; neinterpolišući, tj. oni prolaze blizu kontrolnih tačaka ali ne kroz njih u opštem slučaju.

9.11 Koja su dva načina za crtanje krivih?

1) Povećavanjem vrednosti parametra *t* za konstantnu (malu) vrednost i spajanjem tačke (x(t), y (t), z(t)) pravom linijom sa prethodno određenom tačkom

2) Rekurzivnom podelom krive na segmente dok se ne dođe do pravih linija (tj. približno pravih linija). Za različite tipove krivih razlikuje se proces podele i razlikuju se testovi da li se segment krive može aproksimirati pravom linijom.

10. VIDLJIVOST

10.1 Koja dva cilja razlikujemo pri rešavanju problema određivanja vidljivih površina?
Određivanje vidljivih površi i odbacivanje skrivenih površi.

10.2 Kako se definiše funkcija vidljivosti između tačaka P i Q?
Funkcija vidljivosti V(P, Q) za tačke P i Q na sceni ima vrednost 1 ako ne postoji presek između scene i otvorene duži (P, Q); inače važi V(P, Q) = 0.

10.3 Kako je moguće definisati funkciju vidljivosti u funkciji rastojanja?
Neka je $f(t) = V(P, P + \omega t)$; za $0 \leq t \leq |X - P|$ važi $f(t) = 1$, za $t > |X - P|$ važi $f(t) = 0$.

10.4 Šta je primarna vidljivost?
Primarna vidljivost je vidljivost između tačke sa otvora kamere i tačke na sceni.

10.5 Šta je dubinska složenost zraka?
Dubinska složenost zraka je broj puta koji on preseca scenu.

10.6 Šta je kvantitativna nevidljivost tačke P u odnosu na tačku Q?
Za proizvoljnu tačku P definišemo kvantitativnu nevidljivost kao broj preseka sa drugim primitivama koje se nalaze između početka zraka i tačke P.

10.7 Iz kojih se koraka sastoji rej kasting algoritam?
```
izaberi centar projekcije i prozor na projekcionoj ravni;
za svaku scan liniju na slici uradi sledece:
        za svaki piksel na scan liniji uradi sledece:
        begin
                odredi polupravu iz centra projekcije kroz piksel;
                za svaki objekat na sceni uradi sledece:
                        ako polupravа sece objekat i ako je presecna tacka najbliza do sada onda
                        zapamti presek i ime objekta;
                oboji piksel bojom najblizeg preseka
        end
```

10.8 Čemu odgovaraju unutrašnji čvorovi BSP stabla, a čemu listovi? Koji poluprostor nazivamo pozitivnim, a koji negativnim u odnosu na ravan razdvajanja?
Svaki unutrašnji čvor odgovara jednoj ravni razdvajanja (koja nije deo geometrije scene), a list geometrijskoj primitivi na sceni. Pozitivni poluprostor sadrži sve tačke sa ravni razdvajanja, kao i tačke sa one strane ravni razdvajanja kojoj pripada vektor normale te ravni, a negativni poluprostor sve tačke sa ravni razdvajanja i sa strane suprotne od vektora normale ravni.

10.9 Koja je očekivana složenost algoritma za nalaženje prvog preseka zraka i primitive ako se koristi BSP stablo?
Prosečna složenost je logaritamska u funkciji broja primitiva.

10.10 Šta je bafer dubine (z-bafer)? Koje su njegove moguće primene?
Bafer dubine (ili z-bafer) je dvodimenzioni niz koji odgovara frejm-baferu. Primene:
- dok se scena renderuje, z-bafer sadrži informaciju o utvrđivanju vidljivih površi – nova površ može da pokrije uzorak samo ako je njena dubina u odnosu na kameru manja od vrednosti sadržane u z-baferu
- nakon renderovanja scene, vrednosti u z-bafer opisuju prvi presek sa scenom za svaki zrak iz centra projekcije
- nakon renderovanja scene, korišćenjem z-bafera može se izračunati funkcija vidljivosti u odnosu na centar projekcije O na sledeći način: V(O, Q) = 1 akko je vrednost dubine u tački projekcije Q manja od dubine Q.

10.11 Kako funkcioniše z-bafer algoritam?
Sve vrednosti u frejm baferu su incijalizovane na boju pozadine, a sve vrednosti u z-baferu na z-koordinatu zadnje ravni odsecanja. Maksimalna vrednost z-bafera je z-koordinata prednje ravni odsecanja. Umesto opsega $[-1, 0]$ standardne paralelne zapremine pogleda, u hardveru se koristi vrednost $-z$ i opseg $[0, 1]$, gde vrednost 1 odgovara zadnjoj, a vrednost 0 prednjoj ravni odsecanja. Nekada se vrednosti u z-baferu čuvaju i kao celobrojne vrednosti. Na sve poligone (nije bitan poredak) koji predstavljaju scenu primenjuje se algoritam scan-konverzije. Prilikom scan-konverzije za tačku (x, y), ako ta tačka nije od posmatrača dalja od tačke koja je trenutno u baferu, onda vrednosti te tačke zamenjuju postojeće vrednosti u baferima.

10.12 Koja je ideja algoritama sa listama prioriteta?
Algoritmi sa listama prioriteta implicitno rešavaju problem vidljivosti tako što renderuju elemente scene u redosledu od udaljenijih ka bližim. Zaklonjeni objekti imaju viši prioritet te se prvi renderuju i biće naknadno sakriveni objektima koji se kasnije renderuju.

10.13 Iz kojih koraka se sastoji algoritam sortiranja dubine?
1) svakom poligonu dodeliti ključ sortiranja koji je jednak z-vrednosti temena koje je najdalje od oblasti za prikaz
2) sortirati sve poligone od najudaljenijeg do najbližeg prema vrednosti ključa
3) otkriti slučajeve kada dva poligona imaju dvosmisleno uređenje – ovakve poligone deliti sve dok delovi nemaju eksplicitno uređenje i postaviti ih na pravo mesto u sortiranoj listi
4) renderovati poligone u redosledu prioriteta, od najudaljenijeg do najbližeg

10.14 Kod algoritma sortiranja dubine, koji se testovi koriste prilikom razmatranja dva poligona čije se z koordinate preklapaju?
1) Da li se x koordinate dva poligona ne preklapaju (tj. ne postoje tačke u P i Q sa istom x koord.)?
2) Da li se y koordinate dva poligona ne preklapaju (tj. ne postoje tačke u P i Q sa istom y koord.)?
3) Da li je čitav P sa suprotne strane ravni poligona Q u odnosu na posmatrača?
4) Da li je čitav Q sa iste strane ravni poligona P kao i posmatrač?
5) Da li su projekcije poligona P i Q na projekcionu ravan disjunktne?

10.15 Na koji način je moguće utvrditi da li je neki poligon sa prednje ili zadnje strane objekta?
Pomoću testova (3) i (4).

11. PROSTORNE STRUKTURE PODATAKA

11.1 Šta računaju konzervativne metode za rešavanje upita preseka? Zašto su značajne?

One vraćaju kao odgovor sve primitive koje sadrže preseke. Značajan jer je nekada teško efikasno predstaviti preseke među jednostavnim oblicima.

11.2 Koja je očekivana cena upita preseka na BSP stablima, a koja u najgorem slučaju?

BSP stabla imaju logaritamsko vreme izvršavanja upita preseka pod određenim pretpostavkama.

11.3 Kako se biraju particije kod kd-stabla? Kako se najčešće biraju ravni razdvajanja kod kd-stabala?

Odabir ose na koju je upravna ravan razdvajanja pravi se na osnovu podataka. Neke od često korišćenih mogućnosti su: podela po srednjoj vrednosti u odnosu na opseg; podela po srednjoj vrednosti u odnosu na primitive; podela sa optimalnom cenom.

11.4 Kako izgleda struktura podataka oktri, a kako kuodtri?

Oktri - najčešće se predstavlja korišćenjem 8 pokazivača ka deci. Ne moraju sve ćelije biti iste veličine, već se gušći delovi scene sastoje od većeg broja kocki. Njegov analogon u 2D je kuodtri.

11.5 Koja je razlika između korišćenja BSP stabala i hijerarhija graničnih opsega?

Za razliku od BSP stabala, ovaj tip stabla daje tešnje granice za klastere primitiva i ima konačnu zapreminu. On se često gradi nakon izgradnje BSP stabla, rekurzivnom izgradnjom graničnih opsega, od listova ka korenu.

11.6 Koja je struktura podataka pogodnija za dinamičke podatke: BSP stabla ili mreža?

Mreža.

11.7 Koja je očekivana cena upita preseka na mreži, a koja u najgorem slučaju?

U najgorem slučaju cena upita je O(g·n), a očekivana - algoritam će se izvršavati u vremenu proporcionalnom broju ćelija mreže kojima se prolazi.

11.8 Uporediti mreže sa malom veličinom ćelije sa mrežama kod kojih je veličina ćelije velika.

Ako je g veliko, kvadrati mreže su mali. Ovim se smanjuje broj primitiva koje treba testirati u svakoj nepraznoj ćeliji, ali se povećava broj ćelija koje treba ispitati. Ovaj izbor je dobar za guste scene.
Ako je g malo, kvadrati mreže su veliki. Kod ovakvih mreža zrak se brzo kreće kroz velike prazne prostore, ali se povećava broj primitiva u svakoj nepraznoj ćeliji sa kojom treba vršiti testiranje. Ovo je pogodnije za retke scene.

12. SVETLOST

12.1 Koja svojstva ima monohromatska svetlost?
Jedino svojstvo monohromatske svetlosti je kvantitet svetlosti.

12.2 Kojom skalom je opisan intenzitet svetlosti i zašto?
Pretpostavimo da želimo da predstavimo 256 intenziteta na skali od 0 do 1, tako da ljudskom oku raspodela intenziteta deluje ravnomerno. Međutim, ljudsko oko je osetljivo na odnose intenziteta, a ne na apsolutne intenzitete svetlosti. Stoga, da bismo dobili jednake korake u sjajnosti, intenzitet ne treba da bude opisan linearnom skalom, već logaritamskom.

12.3 Ako je potrebno zadati *n* intenziteta svetlosti počev od I_0 do 1, po kojoj formuli se računa intenzitet I_j u funkciji početne vrednosti intenziteta I_0?
$r = (1/I_0)^{1/n}$, $I_j = r^{j} I_0 = I_0^{(n-j)/n}$

12.4 Šta je gama korekcija?
Često se koriste unapred izračunate vrednosti za određivanje napona tj. vrednosti za jedan piksel i korišćenje ovakvih unapred izračunatih tabela se naziva gama korekcija.

12.5 Šta je polutoniranje i koji su najčešći metodi polutoniranja?
Polutoniranje je tehnika kojom se na bazi minimalnog broja nivoa intenziteta (npr. crna i bela boja) daje privid više različitih nivoa osvetljenosti. Najčešći metodi polutoniranja su uzorkovanje, dithering i prostorni dithering.

12.6 Uzorkovanje daje sliku istih dimenzija kao i polazna? (da) (***ne***)

12.7 Dithering daje sliku istih dimenzija kao i polazna? (***da***) (ne)

12.8 Šta je prostorni dithering?
U prostornom ditheringu se greška koja nastaje za jedan piksel distribuira okolnim pikselima pre nego što se proveri vrednost praga. Postoji uzorak koji u tome daje optimalnu vrednost u smislu minimizacije stvaranja efekta teksture.