

Računarska grafika, računarska vizija i robotika

U modernoj eri, računari su postali alat koji omogućava modelovanje, razumevanje i interakciju sa fizičkim svetom. Koristeći napredne algoritamske tehnike, u mogućnosti smo da simuliramo složene sisteme, da interpretiramo ogromne količine vizuelnih podataka i dizajniramo algoritme koji rešavaju složene mehaničke izazove. U ovom poglavlju biće reči o oblastima računarske grafike, računarske vizije, geometrijskih algoritama i robotike, koje predstavljaju temelj inovacija kojima se svakodnevno pomeraju granice onoga što mašine mogu da vizuelizuju, razumeju i ostvare.

Fokus računarske grafike je na kreiranju vizuelnog sadržaja koji omogućava razvoj realističnih simulacija, animacija i kreiranje virtuelnih okruženja.

Računarska vizija, s druge strane, omogućava računarima da tumače i analiziraju vizuelne podatke.

Geometrijski algoritmi predstavljaju matematičku osnovu za rešavanje prostornih problema, od detekcije kolizije u video igrama, do planiranja pokreta u robotici.

Konačno, robotika predstavlja sinergiju između ovih oblasti, omogućavajući na taj način inteligentnim mašinama da se kreću i interaguju sa složenim okruženjima.

Zajedno, ove discipline ne samo da oblikuju način na koji doživljavamo i komuniciramo sa digitalnim i fizičkim prostorima, već, takođe, pokreću napredak u različitim industrijama, počev od industrije zabave i zdravstvene zaštite do autonomnih si-

stema i veštačke inteligencije. Cilj ovog poglavlja je da predstavi principe, izazove i brojne primene koje se nalaze u njihovom preseku.

1.1 Računarska grafika

Računarska grafika (engl. computer graphics) je oblast računarstva koja se bavi generisanjem, prikazivanjem i obradom slika uz pomoć računara.

Impresivni vizuelni efekti u filmu "Avatar" i računarska animacija u crtanom filmu "Priča o igračkama" samo su neki od primera koji ilustruju mogućnosti i domet računarske grafike.

Računarska grafika je nastala sredinom dvadesetog veka sa pojavom interaktivnih igara i alata za vizuelizaciju, dizajniranih od strane američke vojske za potrebe avijacije, radarske i raketne tehnike. Od svojih početaka pa do danas ona je iz tehnologije koja je bila dostupna samo tehnološki najrazvijenijim organizacijama prešla u svakodnevicu običnog čoveka.

U nastavku teksta ćemo razmotriti osnovne koncepte računarske grafike: modelovanje svih elemenata scene, različite algoritme za iscrtavanje scene, kao i upotrebu boja, tekstura i osvetljenja u cilju dobijanja realističnih prikaza.

1.1.1 Primene računarske grafike

Računarska grafika danas predstavlja primarnu tehnologiju u raznim oblastima: digitalnoj fotografiji, vizuelizaciji informacija, filmskoj industriji, video igrama i mnogim drugim specijalizovanim primenama.

Jedna od najznačajnijih primena računarske grafike je za razvoj grafičkih korisničkih interfejsa (engl. graphical user interface, GUI). Pored računara, grafički korisnički interfejsi su danas prisutni i na pametnim telefonima, kontrolnim tablama automobila, GPS uređajima, kao i na kontrolama različitih kućnih elektronskih uređaja. Računarska grafika ima primene u dizajniranju delova i proizvoda na računaru (engl. Computer Aided Design), kao i za vođenje procesa njihove proizvodnje na osnovu dobijenih virtuelnih dizajna (engl. Computer Aided Manufacturing).

Korišćenjem računarske grafike moguće je izvoditi različite vrste simulacija, poput simulacija vožnje, letenja, kao i sprovoditi treninge u situacijama koje bi mogle biti kritične po bezbednost korisnika ili bi ih bilo suviše skupo realizovati: na primer za treniranje osoblja za potrebe avijacije, vojske i svemirskog programa.

Računarska grafika se koristi i za vizualizaciju informacija kako bi se omogućilo interpretiranje podataka velikog obima, pronalaženje obrazaca u podacima i nekih trendova. Ovo može biti od koristi naučnicima, inženjerima, ljudima koji se bave finansijama, kao i medicinskom osoblju. Vizuelizacija informacija omogućava prikaz mapa vremenske prognoze, prikaz radarskih slika ili pravljenje smislenih slika skeniranih pacijenata, koje pomažu lekarima da iz ovakvih podataka izdvoje korisne informacije.

Svakako jedan od dominantnijih uticaja računarske grafike je u domenu filmske industrije, televizije i video igara. U kontekstu filmske industrije, koristi se za izradu specijalnih efekata i vizuelnih elemenata koje bi bilo isuviše teško ili skupo napraviti na klasičan način (na primer scene eksplozije u akcionim filmovima). Savremene video igre koriste 3D računarsku grafiku za prikaz realističnog okruženja i likova sa kojima igrači mogu da interaguju.

Sa razvojem računarske grafike razvile su se i nove oblasti poput virtuelne (engl. virtual reality) i proširene realnosti (engl. augmented reality). *Virtuelna realnost* omogućava simuliranje računarski generisanog okruženja, poput virtuelnih putovanja i virtuelnih obilazaka muzeja. Korisnik biva "ubačen" u računarski generisano okruženje sa kojim može da interaguje interaktivnim uređajima poput naočara, kaciga, rukavica i specijalne odeće. Virtuelna realnost igra važnu ulogu u treniranju i sprovođenju simulacija. Za razliku od virtuelne realnosti, gde je kompletan sadržaj koji se korisniku prikazuje računarski generisan, u *proširenoj realnosti* se putem pametnih telefona ili posebnih naočara korisniku prikazuje slika koja kombinuje realističan pogled na okolinu sa sadržajem koji je računarski generisan: informacije u vidu teksta ili grafike se u realnom vremenu integrišu sa objektima iz realnog sveta. Na ovaj način moguće je, recimo, turističke obilaske obogatiti prikazom različitih značajnih informacija.

U novije vreme tehnike veštačke inteligencije, pre svega mašinsko učenje i duboko učenje, unapredile su oblast računarske grafike omogućavajući fotorealističnu grafiku u realnom vremenu, sa nižim troškovima obrade. Ovaj napredak je prisutan u različitim oblastima, od računarskih igara pa sve do naučnih istraživanja. Na primer, korišćenjem veštačke inteligencije moguće je automatski kreirati složena okruženja i objekte, generisati teksture, a takođe i poboljšati animaciju i gestikulaciju karaktera. S druge strane, i računarska grafika ima primenu u razvoju veštačke inteligencije: korišćenjem grafike moguće je kreirati veštačka okruženja za treniranje sistema, posebno u oblasti računarskih igara, autonomnoj vožnji i robotici. Takođe, uz pomoć računarske grafike moguće je generisati sintetičke skupove podataka za obuku modela veštačke inteligencije, posebno u domenima gde je teško ili skupo dobiti realne podatke.

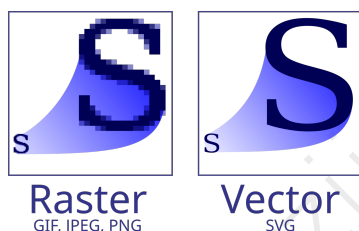
1.1.2 Rasterska i vektorska grafika

Postoje dve osnovne vrste zapisa grafičkog sadržaja: rasterski i vektorski (slika ??).

U *rasterskoj grafici* slika se predstavlja kao pravougaona mreža sićušnih elemenata koje nazivamo *pikseli* (engl. pixel, picture element).

Svaki piksel ima svoju boju (osvetljenost, transparentnost) i kada se svi pikseli posmatraju zajedno, oni formiraju kompletnu sliku. Iako se rasterska slika sastoji od piksela, pri pravoj rezoluciji pojedinačni pikseli nisu vidljivi golim okom. Međutim, zumiranjem slike do visokog stepena moguće je uočiti mrežu pojedinačnih

kvadratića. Svaka rasterska slika sadrži fiksiran broj piksela: broj piksela definiše kvalitet slike i on odgovara njenoj rezoluciji. Piksela može sadržati više ili manje informacija: npr. kod crno-belih slika koristi se samo jedan bit po pikselu, dok slika u boji čuva tri bajta informacije po pikselu. Digitalni fotoaparati, kao osnovni izvor fotografija danas, generišu rasterske slike. I procesom skeniranja kreiraju se rasterske slike. Rasterska grafika takođe predstavlja preovlađujući format na webu. Najpoznatiji rasterski formati su JPG, PNG, TIFF i GIF. Pošto rasterske slike mogu biti velike veličine, posebno pri velikim rezolucijama, one koriste kompresiju kako bi njihovo skladištenje i prenos bili efikasniji. Poenta kompresije je da se smanji veličina odgovarajuće datoteke uz očuvanje što je više moguće vizuelnog kvaliteta.



Slika 1.1: Razlika između rasterske i vektorske grafike (preuzeto sa https://en.wikipedia.org/wiki/Vector_graphics).

U vektorskoj grafici slika se čuva kao kolekcija geometrijskih primitiva (poput tačaka, duži i krivih) i njihovih svojstava, a njihova reprezentacija se temelji na matematičkim jednačinama.

Za svaku od primitiva zadaje se gde će se nalaziti na slici i koje je boje. Najpoznatiji vektorski formati su SVG, EPS, PDF i AI. Za kreiranje (i izmenu) vektorske slike se uobičajeno koristi namenski softver kao što je Adobe Illustrator, CorelDRAW i Inkscape.

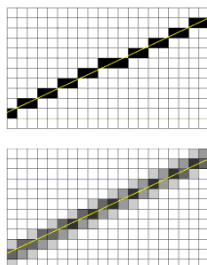
Vektorsku sliku moguće je skalirati (povećati, smanjiti) bez gubitka na kvalitetu slike. Stoga se vektorska grafika pre svega koristi onda kada je potrebno omogućiti skaliranje bez gubitka na kvalitetu slike, poput dijagrama, fontova, logotipova kompanija i slika koje u sebi sadrže tekst. Rastersku sliku moguće je skalirati, međutim, smanjenjem slike u odnosu na izvornu dimenziju gube se informacije o slici. Na primer, ako sliku smanjimo na 50% originalne dimenzije biće prikazane uprosječene vrednosti susednih piksela, čime se gube fini detalji i slika postaje mutna. S druge strane, povećavanjem slike u odnosu na izvornu dimenziju slika postaje nazubljena.

Važna oblast rada sa slikama je *obrada slike* (engl. image processing) koja podrazumeva proces stvaranja nove slike na osnovu postojeće. Nova slika može biti jednostavnija od polazne ili na neki način boljeg kvaliteta: na primer, podešavanjem osvetljenosti ili boja slike, isecanjem dela slike ili eliminisanjem digitalnog šuma.

Napomenimo i to da pošto su današnji ekrani rasterski, slika se, bez obzira na njen format, prilikom prikaza na ekranu mora prevesti u rasterski oblik.

Proces transformisanja vektorske slike u odgovarajući rasterski oblik, tj. u reprezentaciju pikselima naziva se *rasterizacija* (engl. rasterization).

Ona se tipično izvodi tako što se poligoni razbijaju na trouglove, a onda se rešava jednostavniji problem rasterizacije trouglova. Primetimo da će prilikom rasterizacije nužno dolaziti do toga da kose linije izgledaju donekle “nazubljeno” i ovaj efekat naziva se *aliasing* (engl. aliasing). Ovaj efekat se ne može u potpunosti eliminisati, ali postoje brojne tehnike kojima se on može ublažiti i one se nazivaju *antialiasing tehnike* (slika ??).

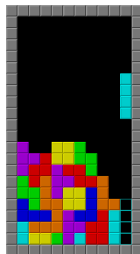


Slika 1.2: Primer kose duži kod koje nije ili jeste primenjena antialiasing tehnika (preuzeto sa <https://www.schorsch.com/en/kbase/glossary/aliasing.html>).

1.1.3 2D i 3D grafika

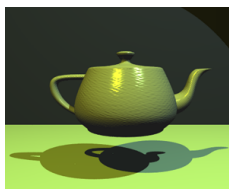
Na osnovu toga sa kakvim modelima radi, razlikujemo dva tipa računarske grafike: dvodimenzionalnu (2D) grafiku i trodimenzionalnu (3D) grafiku. 2D grafika podrazumeva generisanje digitalnih slika pomoću računara, prevashodno na osnovu 2D modela. 2D grafika ima primenu u tipografiji, u grafičkom dizajnu (npr. za kreiranje logotipova i postera), u dizajnu grafičkih korisničkih interfejsa. Većina ranijih video igara se oslanjala na 2D grafiku (slika ??), a i danas se koristi u igrama u kojima je prioritet jednostavnost, na primer u klasičnim igrama poput Super Marija, igrama karata, igrama uloga (engl. role playing games, RPG), retro igrama i igrama edukativnog karaktera.

U 3D grafici se objekti koji su prirodi trodimenzionalni predstavljaju trodimenzionalnim modelima (slika ??). Za razliku od 2D grafike, kod koje je slika zadata svojom širinom i visinom, 3D grafika dodaje nivo dubine kojim se na slici postiže realizam. 3D grafika se koristi u novijim video igrama, u primenama koje uključuju virtuelnu realnost, u 3D animacijama, kao i za vizuelizaciju podataka u medicini,



Slika 1.3: 2D igra tetris.

meteorologiji, astronomiji i dr. Bez sumnje ona je preuzela primat u računarskoj grafici.



Slika 1.4: Čajnik iz Jute, kreiran od strane Martina Njuela 1975. godine i koji predstavlja jedan od referentnih test modela u 3D modelovanju.

1.1.4 Interaktivna grafika

Interaktivna računarska grafika podrazumeva prikaz slike na računaru uz aktivno učešće čoveka u stvaranju i izmeni slike, pri čemu su rezultati odmah vidljivi, kao što je slučaj kod računarskih igara. Nasuprot tome, pod *neinteraktivnom računarskom grafikom* podrazumeva se svako računarsko generisanje ili prikaz unapred pripremljene slike ili video materijala, bez interakcije čoveka; tipičan primer neinteraktivne grafike je screensaver na ekranu računara, kao i računarski generisani filmovi. Danas je većina primena računarske grafike interaktivna.

Prvi interaktivni računarski sistem razvio je Ivan Saderland sedamdesetih godina prošlog veka u MIT-ovim laboratorijama i zvao se Sketchpad. Sketchpad je omogućavao jednostavno iscrtavanje i izmenu slika koje imaju ponovljene elemente. Za svoje doprinose razvoju računarske grafike, počev sa Sketchpad-om, Ivan Saderland je 1988. godine dobio Tjuringovu nagradu. Grafički korisnički interfejsi koji se danas mogu naći na velikom broju uređaja koje koristimo u svakodnevnom životu nastali su na osnovama sistema Sketchpad.

1.1.5 Modelovanje i iscrtavanje

Scena koju iscrtavamo u interaktivnoj računarskoj grafici sastoji se od tri osnovna tipa elemenata: *objekata*, *izvora svetla* i *kamere*.

Svaki od objekata ima svoju geometriju (oblik i veličinu), poziciju na sceni, kao i pridružena svojstva materijala od kojeg je sačinjen. Na sceni, takođe, može postojati jedan ili više izvora svetla. U stvarnom svetu postoje različiti tipovi izvora svetla i svaki od njih ima drugačija svojstva: neki osvetljavaju kompletnu scenu, a neki samo deo scene. Neke od izvora svetla postavljamo direktno na scenu, dok za neke pretpostavljamo da su na beskonačno dalekom rastojanju od objekata. Konačno, prikaz scene zavisi i od parametara kamere: gde se nalazi na sceni, u kom smeru kamera gleda, koji deo scene sagledava i dr.

Računarska grafika se u užem smislu bavi modelovanjem objekata na sceni, modelovanjem osvetljenja scene, modelovanjem virtuelne kamere i algoritmima kako, na osnovu dobijenih modela, iscrtati scenu iz ugla kamere.

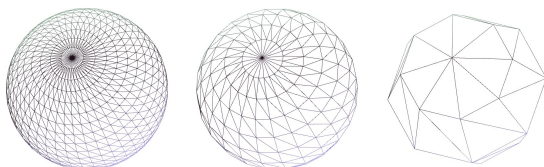
Dakle, dva osnovna zadatka računarske grafike su:

- * *modelovanje* (engl. modelling) i
- * *iscrtavanje*, odnosno *renderovanje* (engl. rendering).

Modelovanje objekata podrazumeva kreiranje njihove matematičke reprezentacije.

Izabrana reprezentacija treba da bude pogodna za obradu putem računara, da je memorijski efikasna i da se upiti koji se često javljaju izvršavaju efikasno.

Jasno je da ne postoji jedan, najbolji model nekog objekta: koliko je model dobar zavisi od karakteristika samog objekta. Dodatno, jedan objekat moguće je modelovati sa različitim *nivoom detaljnosti* (engl. level of details): što je veći nivo detaljnosti to je ciljani objekat vernije modelovan, ali će odgovarajuća reprezentacija biti prostorno manje efikasna i njegova obrada zahtevnija (slika ??).



Slika 1.5: Različiti nivoi detaljnosti pri modelovanju sfere.

Pored geometrije, za svaki objekat potrebno je zadati i njegovu boju, materijal od kog je sačinjen, pridružene teksture i dr. Naime, objekti na različit način reaguju na svetlost: neki su sjajniji, poput metalne kašike, dok su neki matirani, poput glinene saksije. Stoga je za objekat potrebno zadati parametre materijala kojima se određuje da li objekat i u kojoj meri reflektuje svetlost ili je rasipa jednako u svim smerovima.

Pored objekata koji se nalaze na sceni, potrebno je modelovati i izvore svetlosti koji postoje na sceni i utiču na osvetljenost objekata. Svaki od izvora svetla karakteriše njegova pozicija na sceni, boja i intenzitet svetla koje emituje, njegov tip (da li se ponaša poput sijalice, sunca ili reflektora), i dr. Naime, svetlost ćemo različito modelovati u zavisnosti od toga da li želimo da modelujemo sunčevu svetlost, sobnu sijalicu ili reflektorsko svetlo.

Najzad, u procesu modelovanja, potrebno je zadati i parametre kamere koja se nalazi na sceni.

Iscrtavanje, odnosno renderovanje se bavi generisanjem realistične dvodimenzionalne digitalne slike na osnovu modela objekata na sceni i modela ponašanja svetlosti na sceni.

Renderovanje, dakle, podrazumeva transformisanje trodimenzionog prostora u dvodimenzionu sliku.

Prema vremenu u kojem se vrši, razlikujemo dve vrste renderovanja: *renderovanje u realnom vremenu* (koje se naziva i onlajn renderovanje) i *oflajn renderovanje*. Renderovanje u realnom vremenu je tehnika koja omogućava kreiranje slika i animacija brzinom koja je dovoljna da omogući interakciju bez приметnog kašnjenja, kao što je slučaj u računarskim igrama i simulacijama kada je potrebno brzo reagovati na akcije korisnika. Oflajn renderovanje se koristi za kreiranje realističnih slika visokog kvaliteta kada vreme nije kritičan resurs; tipičan primer upotrebe je za specijalne efekte u filmovima, kada je dozvoljeno da se svaka slika renderuje i po nekoliko sati ili dana.

Modelovanje objekata

Objekat najčešće zadajemo u koordinatnom sistemu u kojem ga je najjednostavnije opisati. Na primer, ako treba modelovati objekat sfernog oblika čiji se centar nalazi u tački A sa koordinatama $(2,3,5)$ i koji ima poluprečnik 10, možemo krenuti od jednostavnije postavke problema, u kojoj je zadatak modelovati sferu poluprečnika 1 sa centrom u koordinatnom početku. Nakon toga ćemo različitim transformacijama modelovanja (engl. modelling transformation) objektu postaviti dimenziju (u ovom slučaju postaviti da poluprečnik sfere bude 10), kao i pozicionirati objekat na potrebno mesto na sceni (u ovom slučaju postaviti da centar sfere bude u tački A).

Naravno, nemaju svi objekti iz sveta koji nas okružuje jednostavnu geometriju, poput sfere ili kvadra. Naime, objekti mogu imati proizvoljno složenu geometriju. Razmotrimo problem modelovanja karoserije automobila ili mačka u čizmama u

filmu o Šreku. Ovakvi složeni modeli se sastavljaju od jednostavnijih korišćenjem različitih *geometrijskih transformacija*. Geometrijske transformacije igraju značajnu ulogu u domenu grafike: njima se mogu postaviti dimenzije i pozicija objekta; njima se realizuje preslikavanje iz jednog u drugi koordinatni sistem; njihovim dejstvom na objekte tokom vremena se omogućava kreiranje animacija i dr. Najznačajnije među ovim transformacijama su svakako translacija, skaliranje i rotacija.

Važnu klasu geometrijskih transformacija predstavljaju *linearne transformacije*.

Linearne transformacije su transformacije koje slikaju koordinatni početak sam u sebe i čuvaju kolinearnost tačaka (prave se slikaju u prave). Linearne transformacije su pogodne za rad jer se mogu izraziti u vidu množenja matricom. Linearne transformacije su transformacije koje se mogu predstaviti u obliku $T(\mathbf{x}) = \mathbf{M} \cdot \mathbf{x}$, gde je $\mathbf{M}$ odgovarajuća matrica transformacije, a $\mathbf{x}$ je vektor koordinata tačke koju transformišemo. U ovom slučaju, $\mathbf{M}$ je kvadratna matrica dimenzije $n \times n$, a $\mathbf{x}$ je vektor dimenzije $n \times 1$.

Na primer, rotacija oko koordinatnog početka je linearna transformacija jer njome koordinatni početak ostaje invarijantan, a prave se slikaju u prave. Stoga kompoziciji linearnih transformacija odgovara ponovo linearna transformacija, čija je matrica jednaka proizvodu matrica pojedinačnih transformacija. Skaliranje i rotacija spadaju u linearne transformacije, ali ne i translacija. Naime, pri translaciji koordinatni početak nije invarijantan. Kako bi se omogućio uniforman rad sa pomenutim transformacijama (tj. da i translaciju možemo da izrazimo u vidu množenja matricom), uvodi se koncept tzv. *homogenih koordinata* (engl. *homogenous coordinates*), u kojima se sve navedene transformacije mogu formulisati u vidu množenja matricom. Dodatna pogodnost homogenih koordinata jeste to što omogućavaju predstavljanje i beskonačno dalekih tačaka u nekom smeru.

Tačka u ravni je u homogenim koordinatama određena ne sa 2, već sa 3 vrednosti (x_h, y_h, w_h) . Ove koordinate određuju tačku u ravni sa koordinatama $(x_h/w_h, y_h/w_h)$, pri čemu vrednost w_h može biti proizvoljna.

Primer 1.1

Homogene koordinate:

(4,3,7)
(8,6,14)
(4/7,3/7,1)

Tačka u ravni:

(4/7,3/7)
(4/7,3/7)
(4/7,3/7)

Primer 1.2. Rotacija (linearna transformacija koja čuva koordinatni početak)

Rotacija oko koordinatnog početka je primer linearne transformacije. Posmatrajmo rotaciju za 90° . Matrica rotacije u homogenim koordinatama je:

$$R = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Ispitajmo efekat na koordinatni početak, čija je homogenizovana reprezentacija:

$$O_h = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Množenjem dobijamo:

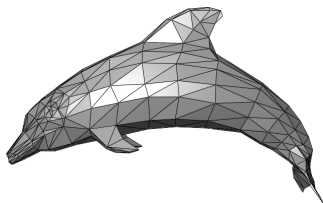
$$R \cdot O_h = \begin{pmatrix} 0 \cdot 0 + (-1) \cdot 0 + 0 \cdot 1 \\ 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 \\ 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Kao što se vidi, koordinatni početak ostaje nepromenjen, tj. $R(O_h) = O_h$.

Postoji veliki broj različitih načina na koje je moguće modelovati dati objekat i odabir najpogodnije reprezentacije zavisi od karakteristika objekta koji se modeluje. Najčešće korišćena reprezentacija objekata jeste *mreža poligona* (engl. polygon mesh). Mreža poligona se sastoji od konačnog, ali često jako velikog broja poligona međusobno povezanih ivicama. Najčešće korišćeni poligoni su trouglovi jer su tri tačke u prostoru uvek komplanarne, tj. leže u ravni, dok to ne mora biti slučaj za veći broj tačaka. Mrežama poligona je jako lako modelovati polidarske površi i tela sa ravnim stranama. Razmotrimo kao primer problem modelovanja objekta oblika kvadra: svaku od strana kvadra je dovoljno podeliti dijagonalom na dva trougla: dobijeni trouglovi obrazuju jednu mrežu trouglova koja potpuno verno opisuje dati objekat.

Primetimo ipak da svet koji nas okružuje nije sačinjen od pljosnatih objekata, poput poliedara. U slučaju kada je objekat gladak, model zasnovan na mreži poligona daje samo dovoljno dobru aproksimaciju polaznog objekta (slika ??).

Kako bismo ovakve objekte prikazali realistično poželjno ih je modelovati glatkim površima. Dodatno, za neke primene, poput fontova, potrebno je omogućiti da prikaz izgleda jednako dobro u svakoj rezoluciji. Takođe, kretanje kamere po sceni ne sme biti "skokovito" i potrebno ga je opisati glatkom krivom. Za ovakve primene koriste se deo-po-deo glatke krive i površi.



Slika 1.6: Delfin modelovan u vidu mreže trouglova.

Modelovanje svetla

Na sceni se mogu naći različiti tipovi izvora svetla. U realnom svetu svaki izvor svetla je trodimenzioni objekat koji ima svoju geometriju (oblik i veličinu). U računarskoj grafici se, međutim, razmatraju pojednostavljeni modeli izvora svetla. Ovde ćemo opisati neke osnovne modele: tačkasti i direkcioni izvor svetla i ambijentalno svetlo.

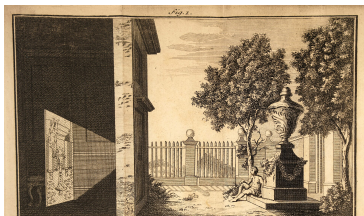
Tačkasti izvor svetla (engl. point light source) je izvor svetla beskonačno male dimenzije, pozicioniran na sceni. On iz svoje pozicije na sceni ravnomerno rasipa svetlost u svim smerovima. Tačkasti model svetla predstavlja dobru aproksimaciju u situaciji kada je izvor svetla značajno manji od veličine objekata na sceni, na primer ako se u prostoriji nalazi mala sijalica.

Direkcioni izvor svetla (engl. directional light source) simulira svetlost koja se emituje iz izvora svetla koji je beskonačno daleko od objekata sa scene, poput sunčeve svetlosti. Svetlost iz direkcionog izvora pada uvek pod istim uglom, te se za njega po pravilu umesto pozicije zadaje smer u kome zraci svetlosti padaju na objekte sa scene.

Dodatno, često se u model osvetljenja dodaje i tzv. *ambijentalno svetlo* (engl. ambient light) koje se koristi za simulaciju prirodnog svetla na sceni. Ono ne dolazi od nekog konkretnog izvora svetla, već podjednako osvetljava sve objekte na sceni. Ambijentalno svetlo obezbeđuje da je svaka površ na sceni donekle osvetljena, čime se sprečava dobijanje nerealističnih tamnih regiona na slici u slučaju objekata koji su daleko od izvora svetla.

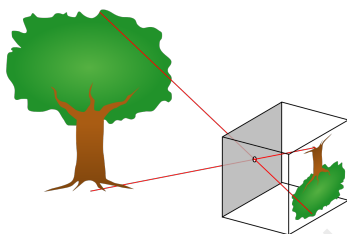
Modelovanje kamere

U računarskoj grafici se bavimo problemom računarske simulacije procesa kojim se generiše slika posredstvom fotoaparata, tj. kamere. Kako bismo mogli da trodimenzionalne objekte predstavimo u ravni, potrebno je izvršiti njihovo *projektovanje* na neku ravan.



Slika 1.7: Ilustracija procesa dobijanja slike korišćenjem kamere opskure.

spravu, preteču fotoaparata, koja se u prošlosti koristila za pravljenje fotografija, ali su je koristili i slikari za pravljenje realističnih slika. Ona funkcioniše po principu zamračene sobe ili kutije koja ima mali otvor ili sočivo na jednoj strani, kroz koji zraci svetlosti ulaze i zatim pogađaju zadnji zid aparature. Na zadnji zid aparature moguće je postaviti fotoosetljivi film, koji kada pogodi zrak svetlosti dešava se hemijska reakcija koja ostavlja trag na filmu (slika ??). Na taj način se na zadnjem zidu aparature formira realistična slika scene, doduše u ovom slučaju izvrnuta naopako.



Slika 1.8: Princip rada kamere opskure.

Opisani postupak generisanja slike odgovara tzv. *perspektivnom projektovanju* sadržaja scene na ravan zadnjeg zida aparature.

Perspektivno projektovanje

Slika se u perspektivnoj projekciji dobija tako što se iz centra projekcije puste zraci do tačke objekta i tačka u kojoj projekтивni zrak preseče ravan projekcije biće perspektivna projekcija te tačke (slika ?? levo).

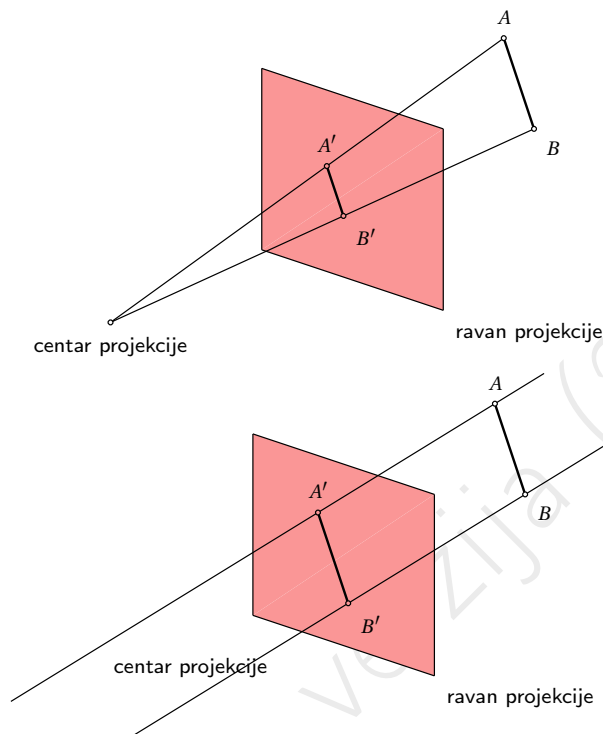
Perspektivna projekcija oponaša način na koji čovekovo oko doživljava dubinu i rastojanja u svetu koji nas okružuje. Naime, objekti koji su nam dalji nam izgledaju manji nego oni koji su nam bliži. Ovo odgovara konceptu perspektivnog skraćivanja.

Otvor kod kamere opskure i sočivo fotoaparata imaju svoju površinu, međutim u kontekstu grafike se on aproksimira tačkom koja odgovara centru projekcije, dok zadnji zid aparature određuje ravan projekcije. Kod kamere opskure je slika izvrnuta u odnosu na scenu jer se projekcija vrši na zid koji se nalazi sa druge strane scene u odnosu na otvor kamere.

Paralelno projektovanje

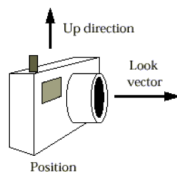
Kod paralelne projekcije centar projekcije je beskonačno daleko od scene, tj. svi projekтивni zraci su međusobno paralelni (slika ?? desno).

Kamera je deo scene u računarskoj grafici i potrebno je specifikovati njene parametre. Najpre je potrebno zadati njen položaj u trodimenzionom prostoru: on je određen pozicijom kamere, smerom u kom kamera "gleda", tzv. vektorom pogleda,



Slika 1.9: Ilustracija perspektivne i paralelne projekcija.

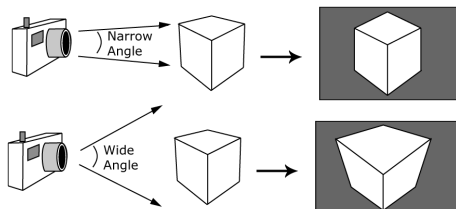
i vektorom usmerenim naviše u odnosu na kameru, kojim se zadaje orijentacija kamere (slika ??).



Slika 1.10: Pozicija, usmerenje i orijentacija kamere (preuzeto sa slajdova University Brown).

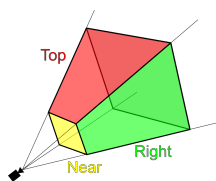
Pored položaja, potrebno je precizirati koji deo prostora kamera sagledava. Potrebno je definisati vidno polje koje želimo da obuhvatimo na slici. Pokazuje se da čovek u svojim svakodnevnim aktivnostima sagledava dosta skromno vidno polje, čiji ugao iznosi samo par desetina stepeni. Za manje vrednosti uglova vidnog polja dobijena slika ima manji stepen perspektivnog iskrivljenja (slika ??). Dodatno, obično se ne razmatraju objekti koji su preblizu kameri jer bi zaklanjali druge objekte

na sceni, i pritom imali visok stepen iskrivljenja. Ne razmatraju se ni objekti koji su predaleko od kamere jer nisu vizuelno značajni, a mogu da troše značajno vreme pri iscrtavanju scene. Ova odsecanja se postižu definisanjem prednje i zadnje ravni odsecanja, koje su paralelne ravni projektovanja.



Slika 1.11: Različiti stepeni perspektivnog iskrivljenja u zavisnosti od veličine vidnog ugla (preuzeto sa slajdova Univerisity Brown).

Prethodno navedenim parametrima definiše se *zapremina pogleda* (engl. view volume, frustum) oblika četvorostране zarubljene piramide čiji je oblik definisan uglovima vidnog polja i prednjom i zadnjom ravni odsecanja, dok je njen položaj određen pozicijom kamere, vektorom pogleda i njenom orijentacijom (slika ??). Objekti koji se u potpunosti nalaze unutar zapremine pogleda biće razmatrani prilikom iscrtavanja slike, objekti koji su u potpunosti van nje se odbacuju, dok se kod objekata čiji se jedan deo nalazi van zapremine pogleda, a deo van zapremine pogleda vrši odsecanje u odnosu na zapreminu pogleda i u razmatranje se uzima samo deo koji se nalazi u unutrašnjosti. Ovo ima kao posledicu da se u nekim računarskim igrama dešava da objekti koji su daleko na sceni kada im se približimo odjednom iskoče na sceni, na primer u simulacijama vožnji.



Slika 1.12: Zapremina pogleda.

1.1.6 Boje

Veoma važan aspekt čovekovog doživljaja sveta koji ga okružuje se prenosi bojama. Boje predstavljaju i važnu komponentu vizuelne komunikacije u računarskoj grafici: kada se koriste na pravilan način mogu u velikoj meri da poboljšaju efikasnost poruke koju želimo da prenesemo.

Opažena boja nekog objekta zavisi od različitih faktora: od izvora svetla koji osvetljavaju objekat i njegovih karakteristika (poput intenziteta, nijanse, pozicije

i tipa izvora svetla), od geometrije samog objekta i svojstva materijala od kojeg je napravljen, od boje i svojstava okoline, ali i od čovekovog vizualnog sistema. Naime, bela sfera obasjana svetlošću reflektora crvene i plave boje će posmatraču u nekim svojim delovima delovati crveno, plavo, pa čak i ljubičasto. Dakle, boja objekta nije u samom objektu već u svetlosti koja se reflektuje ili prelama kroz objekat.

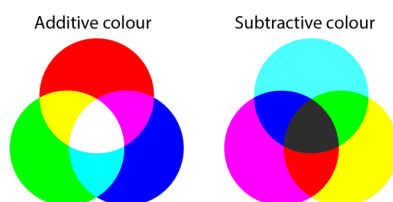
Važno je napraviti razliku u tome kako se meša svetlost, a kako pigmenti, tj. farbe. Naime, ako uperimo crveno i zeleno svetlo na ravnomerno reflektujući komad belog papira, reflektovana svetlost delovaće žuto. Za razliku od toga, ako na komad belog papira nanese crvenu temperu, pa preko nje nanese zelenu, došlo bi do mešanja crvene i zelene farbe i dobijena boja bi delovala braonkasto.

Mešanje boja

U skladu sa ovim razmatranjem razlikujemo dva koncepta mešanja boja: *aditivno* i *subtraktivno* (slika ??): aditivno mešanje boja odgovara mešanju svetlosti, a subtraktivno mešanju farbi.

- * U aditivnom mešanju boja se kreće od crne (odsustva svetla) i dodaju se različite boje svetlosti kako bi se dobile druge boje. Dakle, dodavanje više svetala rezultuje svetlijim bojama.
- * Kod subtraktivnog mešanja boja se kreće od svetlosti bele boje (koja sadrži sve boje) i iz nje oduzimamo određene talasne dužine korišćenjem pigmenata koji apsorbiraju neku svetlost, a ostalu reflektuju. Dakle u subtraktivnom mešanju boja se mešanjem većeg broja pigmenata dobija tamnija boja.

Aditivno mešanje boja igra važnu ulogu kod televizora i monitora u boji i za npr. osvetljenje scene u pozorištu, dok je subtraktivno mešanje boja važno za štampa.



Slika 1.13: Razlika između aditivnog i subtraktivnog mešanja boja (Image attribution: Additive and Subtractive colour primaries by Lisa Cianci, CC BY-NC-SA 4.0).

Postoji veliki broj načina na koje je moguće zadati boju i njih nazivamo *kolor modelima* (engl. color model). Kolor model za rastersku grafiku zadaje se trodimenzionim koordinatnim sistemom i podoblašću oblasti svih vidljivih boja. Svrha kolor modela je da omogući jednostavno adresiranje boja iz nekog opsega.

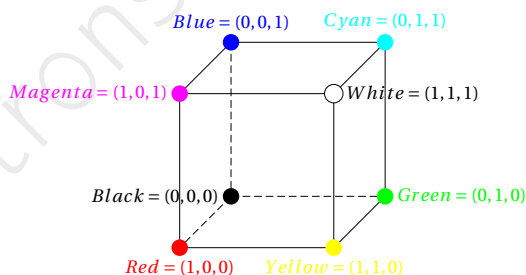
Kolor modeli se međusobno razlikuju po tome koji opseg boja mogu da predstavljaju, koliko je izabrani način zadavanja boje udoban za čoveka, od tehnologije u kojoj se primenjuje i sl. Naravno, svi kolor modeli su ograničeni, u smislu da mogu da opišu boje do određenog intenziteta: ovo često odgovara fizičkim karakteristikama uređaja. Najčešće korišćeni kolor modeli su RGB, HSV i HLS koji se koriste kod monitora i CMY kolor model koji se koristi u uređajima za štampanje.

RGB model

Opseg boja monitora može se zadati RGB kolor modelom. U RGB kolor modelu boje se predstavljaju kao kombinacija crvene (R), zelene (G) i plave (B) boje, pri čemu je vrednost svake boje iz opsega $[0,1]$. Na primer, crvena boja se u RGB modelu zadaje trojkom vrednosti $(1,0,0)$ gde crvena komponenta ima maksimalnu moguću vrednost, bez doprinosa plave i zelene komponente.

RGB kolor model se zadaje jediničnom kockom u trodimenzionom prostoru tako da se u temenu kocke sa koordinatama $(1,0,0)$ nalazi crvena boja, u temenu sa koordinatama $(0,1,0)$ zelena, a u $(0,0,1)$ plava boja. U koordinatnom početku nalazi se crna boja, dok se bela boja nalazi u temenu kocke sa koordinatama $(1,1,1)$. Ovo znači da kombinacija crvene, plave i zelene boje maksimalnog intenziteta proizvodi belu boju. Komplementarne boje se u RGB modelu nalaze jedna nasuprot druge. Nijanse sive se nalaze duž glavne dijagonale kocke, dok se sa većim udaljavanjem od glavne dijagonale dobijaju sve zasićenije boje (slika ??).

RGB kolor model se koristi za ekrane u boji. Primitive koje se koriste u ovom modelu su aditivne – pojedinačni doprinosi svake primitive se dodaju i daju konačni rezultat.

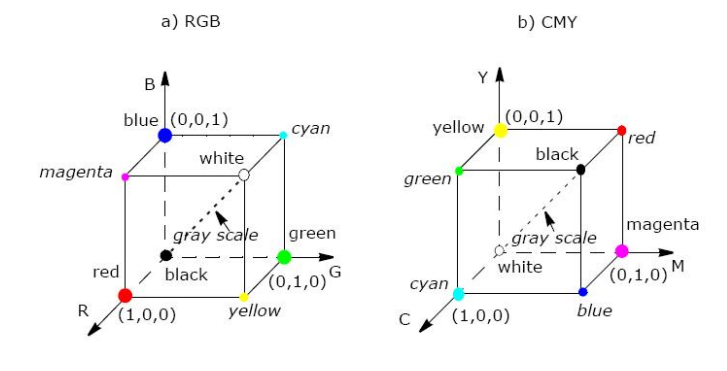


Slika 1.14: RGB kolor model.

CMY(K) model

U CMY kolor modelu boje se opisuju kao mešavine cijan (zeleno-plave), magenta (ljubičasto-crvene) i žute boje. Ovaj kolor model se koristi za štampače, jer mastilo reflektuje neki deo primljene svetlosti, a apsorbira drugi deo. Naime, kada

se površ prekrije cijan mastilom, iz reflektovane bele svetlosti (koja predstavlja sumu crvene, zelene i plave svetlosti) apsorbira se crveno svetlo, a reflektuje plavo i zeleno, magenta apsorbira zeleno, a žuta plavo svetlo. Kada se pomešaju dva mastila, svetlost koja se reflektuje je ona koju ne apsorbira nijedna od njih. Stoga, mešavina cijan i magenta mastila apsorbira i crvenu i zelenu boju, dajući pritom refleksiju plave svetlosti. Ako izmešamo cijan, magenta i žuto mastilo, apsorbiraće se crvena, zelena i plava svetlost i dobićemo crnu boju. U stvarnosti ta boja nije sasvim crna, jer mešavina cijan, magente i žute boje ne uspeva da apsorbira baš svu svetlost, stoga štampači često imaju i četvrto mastilo – crno (u oznaci K) koje se koristi da zameni tamnije mešavine cijan, magenta i žute boje.



Slika 1.15: Uporedni prikaz RGB i CMY kolor modela (preuzeto sa https://scc.ustc.edu.cn/zlsc/sugon/intel/ipp/ipp_manual/IPPI/ippi_ch6/ch6_color_models.htm).

CMY kolor model je, poput RGB kolor modela, određen jediničnom kockom u trodimenzionom prostoru, tako da se u koordinatnom početku umesto crne boje nalazi bela boja, cijan se nalazi u temenu kocke sa koordinatama (1,0,0), magenta u temenu sa koordinatama (0,1,0), a žuta boja u temenu sa koordinatama (0,0,1) (slika ??).

1.1.7 Teksture

Razmotrimo složenost mreže trouglova kojom bi se potpuno verno modelovala Velika Keopsova piramida. U osnovi njena geometrija je veoma jednostavna – u pitanju je tetraedar, međutim modelovanje svih njenih nepravilnosti, pukotina i izbočina bi iznedrilo mrežu sačinjenu od miliona trouglova, čime bi eksplodirali i memorijski i vremenski zahtevi obrade. Srećom, postoji način da se dobije odgovarajući prikaz, bez povećanja složenosti geometrijskog modela.

Preslikavanjem tekstura moguće je tokom procesa iscrtavanja (a ne modelovanja) povećati nivo detaljnosti scene bez uvećanja broja iscrtanih primitiva. Takođe,

tokom procesa iscrtavanja se površi mogu dodati fini detalji, čime se dodatno unapređuje realizam na slici. Na primer, moguće je na objektu prikazati neravnine, prljavštinu, ogrebotine, otiske ili stvoriti privid da je napravljen od nekog materijala, poput drveta, stakla ili mermera.

Nanošenje tekstura na trodimenzionu površ nalikuje prekrivanju objekta rastegljivim listom papira. Teksture preslikavamo na trougao tako što svakoj tački trougla dodeljujemo boju čitanjem odgovarajuće vrednosti sa slike teksture. Preciznije, za svaku tačku trougla potrebno je zadati koju tačku papira treba da „dotakne”. U praksi se ovo preslikavanje zadaje za svako teme trougla, dok se za unutrašnje tačke trougla koristi interpolacija.

Ovakav mehanizam nanošenja tekstura pretpostavlja postojanje koordinatnog sistema u kom možemo da referišemo na pozicije sa slike teksture. Prema dogovoru na tačke sa slike teksture referišemo korišćenjem *teksturnih koordinata* (engl. texture coordinates) u i v koje uzimaju vrednosti iz intervala $[0, 1]$.

Kao što smo već napomenuli, objekti se u računarskoj grafici često zadaju mrežom trouglova. Razmotrimo na koji način bismo teksturisali proizvoljnu mrežu trouglova.

Baricentrične koordinata

- * Najpre je potrebno svakom temenu mreže trouglova pridružiti teksturne koordinate u i v , nakon čega se jednostavno mogu odrediti (u, v) koordinate svake tačke date mreže trouglova, korišćenjem *baricentričnih koordinata* (engl. barycentric coordinates).
- * Ako temena trougla označimo sa A , B i C , tada proizvoljnu tačku P trougla ABC možemo predstaviti kao konveksnu linearnu kombinaciju temena trougla, odnosno u obliku $\alpha A + \beta B + \gamma C$, gde je $\alpha + \beta + \gamma = 1$ i $\alpha, \beta, \gamma \geq 0$.
- * Tačke kod kojih važi $\alpha = 0$ leže na stranici BC , tačke kod kojih važi $\beta = 0$ leže na stranici AC , dok tačke kod kojih važi $\gamma = 0$ leže na stranici AB trougla.
- * Brojeve α, β, γ nazivamo baricentričnim koordinatama tačke P u odnosu na trougao ABC i pišemo $P = \alpha A + \beta B + \gamma C$.

Stoga, ako su nam je poznata teksturna koordinata u temena trougla ABC , teksturnu koordinatu u tačke P trougla ABC možemo izračunati po formuli $u_P = \alpha u_A + \beta u_B + \gamma u_C$. Analogno možemo izraziti i v koordinatu tačke P preko koordinate v temena A , B i C . Na ovaj način se na jedinstven način određuju teksturne koordinate tačke P , interpolacijom teksturnih koordinata temena trougla ABC .

1.1.8 Algoritmi za utvrđivanje vidljivosti

Jedan od osnovnih zadataka u računarskoj grafici jeste utvrđivanje koji su objekti sa scene vidljivi iz pozicije kamere i u koje se piksele preslikavaju. Naime, nisu svi

objekti na sceni vidljivi iz ugla kamere: neki se nalaze van zapremine pogleda, neki su zaklonjeni drugim objektima. Nisu ni svi poligoni jednog istog objekta vidljivi: poligoni koji se nalaze sa zadnje strane neprovidnog objekta u odnosu na kameru neće biti vidljivi iz pozicije kamere. Razmotrimo primer objekta sa scene oblika kocke: najviše tri od šest njenih strana biće vidljive.

Od nastanka računarske grafike do danas razvijeni su različiti algoritmi za utvrđivanje vidljivih površi. Danas su najznačajniji algoritmi: *z-bafer algoritam* (engl. z-buffer algorithm) i *rej kasting algoritam* (engl. ray casting algorithm).

z-bafer algoritam

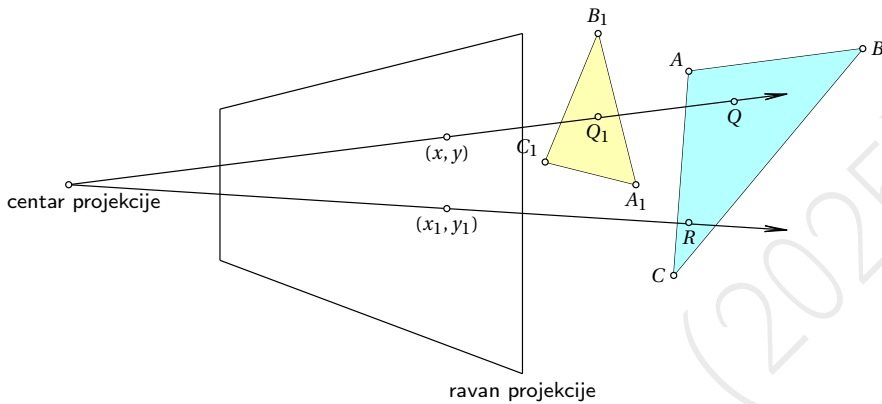
z-bafer algoritam predstavlja jedan od najjednostavnijih algoritama za određivanje vidljivih površi. Kao što i naziv algoritma sugerise, u algoritmu se pretpostavlja postojanje i intenzivno korišćenje z-bafera, odnosno bafera dubine. U z-baferu se tokom procesa rasterizacije trouglova čuvaju vrednosti dubine, tj. z-koordinate za svaki piksel slike. Na ovaj način se omogućava ispravno utvrđivanje da li se trouglovi međusobno zaklanjaju na poziciji nekog piksela.

Princip rada z-bafer algoritama

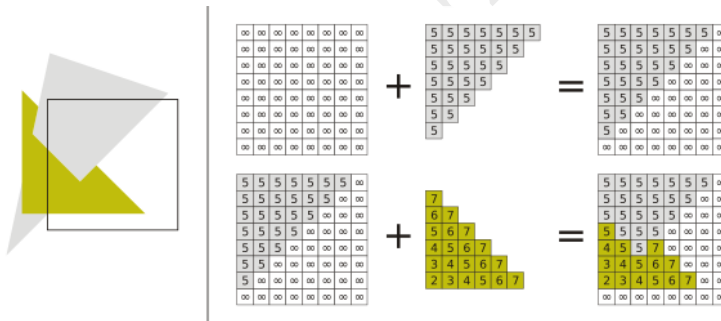
- * Na početku algoritma se sve vrednosti u kolor baferu inicijalizuju na boju pozadine, dok se sve vrednosti u z-baferu inicijalizuju na z-koordinatu zadnje ravni odsecanja (kao najudaljenije tačke prostora koju kamera razmatra).
- * Na sve trouglove kojima je predstavljena scena primenjuje se algoritam rasterizacije. Redosled u kome se trouglovi obrađuju je proizvoljan.
- * Prilikom procesa rasterizacije, ako je dubina d tačke trougla koja se projektuje u piksel (x, y) manja od vrednosti koja se trenutno nalazi u z-baferu za dati piksel (ako je nova vrednost bliža), onda se u z-bafer za piksel (x, y) upisuje dubina d , a u kolor bafer boja tačke poligona koji se rasterizuje u piksel (x, y) .

Razmotrimo primer ilustrovan na slici ???. U smeru zraka iz pozicije kamere kroz piksel (x, y) najbliži je trougao $A_1B_1C_1$, a u smeru zraka kroz piksel (x_1, y_1) trougao ABC . I trougao ABC i trougao $A_1B_1C_1$ se projektuju u piksel (x, y) , dok se samo trougao $A_1B_1C_1$ projektuje u piksel (x_1, y_1) . Ukoliko bi se prvo vršila rasterizacija trougla ABC , vrednosti dubine u z-baferu za piksel (x, z) i (x_1, y_1) bi se postavile na z-koordinatu tačaka Q i Q redom kao vrednosti dubine trougla ABC . Nakon toga, prilikom rasterizacije trougla $A_1B_1C_1$ zaključuje se da se da piksel (x, y) pripada njegovoj projekciji i da je tačka Q_1 bliža posmatraču od tačke Q te se vrednost u z-baferu na poziciji (x, y) menja vrednošću dubine trougla $A_1B_1C_1$.

Na slici ??? ilustrovano je izvršavanje z-algoritma na primeru dva trougla: sivog, koji je paralelan ravni projektovanja (dubine svih tačaka trougla su iste) i žutog



Slika 1.16: Ilustracija z-bafer algoritma.



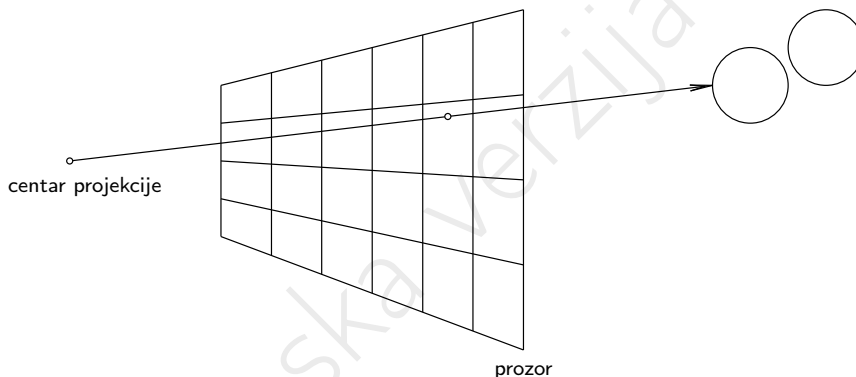
Slika 1.17: Ilustracija menjanja vrednosti u z-baferu.

trougla, koji nije paralelan ravni projektovanja. Inicijalno su sve vrednosti u z-baferu postavljene na vrednost ∞ (koja ovde simulira zadnju ravan odsecanja kamere), a vrednosti kolor bafera na belu, kao boju pozadine. Najpre se vrši rasterizacija sivog trougla i vrednosti z-bafera za sve piksele u unutrašnjosti trougla se ažuriraju i postavljaju na z-koordinatu sivog trougla, a odgovarajuće vrednosti u kolor baferu na sivu. Nakon toga vrši se rasterizacija žutog trougla i za svaki piksel u projekciji žutog trougla se vrši poređenje dubine tačke žutog trougla na poziciji tog piksela i tekuće vrednosti u z-baferu: ako je nova vrednost manja, vrši se ažuriranje vrednosti i z-bafera i kolor bafera.

z-bafer algoritam se veoma jednostavno implementira. Zbog svoje jednostavnosti, može se implementirati i u hardveru.

Rej kasting algoritam

Rej kasting algoritam (engl. ray casting algorithm) utvrđuje da li je neka površ vidljiva iz ugla kamere praćenjem imaginarnog zraka svetlosti od pozicije kamere do objekta na sceni. Pretpostavimo da je prozor proizvoljne ravni pogleda podeljen pravilnom mrežom tako da elementi mreže odgovaraju pikselima potrebne rezolucije. Zamislimo, takođe, da pustimo zrak svetlosti kroz centar piksela tog prozora – objekat koji zrak pogodi je onaj koji treba da bude prikazan na poziciji tog piksela. Dakle, za zadatu poziciju kamere i prozor ravni pogleda za svaki piksel se prati po jedan zrak od centra projekcije kroz sam piksel do najbližeg objekta na sceni (slika ??). Boja piksela se postavlja na boju najbližeg objekta u pravcu odgovarajućeg zraka.



Slika 1.18: Ilustracija rej kasting algoritma.

Naziv rej kasting se obično koristi za varijantu algoritma kojom se vrši samo određivanje vidljivosti, dok se za puni rekurzivni algoritam kojim se obrađuju i senke, refleksija i prelamanje svetlosti koristi naziv *rej trejsing*, odnosno *tehnika praćenja zraka* (engl. ray tracing).

Tehnika praćenja zraka je sporija od z-bafer algoritma, ali je njom lakše dobiti fotorealistične slike.

1.1.9 Osvetljenje i senčenje

Scena koju iscrtavamo u interaktivnoj računarskoj grafici osvetljena je izvorima svetla koji postoje na sceni.

Problem *osvetljenja* (engl. illumination) odnosi se na izračunavanje intenziteta i boje pojedinačne tačke sa površi na sceni onako kako je posmatrač doživljava,

simuliranjem atributa svetlosti. Problem *senčenja* (engl. shading) primenjuje model osvetljenja na skup tačaka i boji/osvetljava kompletnu površ.

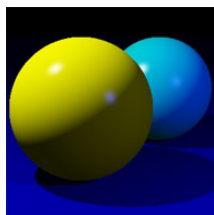
Odbijanje svetlosti od objekata na sceni određeno je karakteristikama izvora svetla, svojstvima materijala od koga je objekat i geometrijom tog objekta. Svaka tačka sa površine objekta prima svetlost *direktno* od izvora svetla koji nisu blokirani drugim objektima i *indirektno* od svetlosti koja dolazi od drugih objekata na sceni nakon refleksije i prelamanja svetlosti. Složeni algoritmi zasnovani na pravilima fizike koji modeluju rekursivnu prirodu međuobjektnih refleksija zahtevaju mnogo izračunavanja. Stoga se računarska grafika u realnom vremenu uglavnom zasniva na tehnikama aproksimacije. Sa današnjim grafičkim procesorima moguće je napraviti dovoljno dobre aproksimacije kojima je moguće zavarati čovekov vizuelni sistem.

Problem osvetljenosti

Razmotrimo na koji način možemo doći do osvetljenosti neke tačke objekta ukoliko se na sceni nalazi jedan tačkasti izvor svetla.

Udeo svetlosti koji dopire do objekta a koji će dalje biti reflektovan, kao i način na koji će ta svetlost biti reflektovana u mnogome zavisi od glatkosti površi. Kod glatkih površi poput ogledala ili mirne površine vode javlja se *spekularna refleksija* (engl. specular reflection), dok se kod hrapavijih površina poput glinene saksije i zida javlja *difuzna refleksija* (engl. diffuse reflection).

Efekat spekularne refleksije se javlja na svakom glatkom, sjajnom objektu. Naime, ako osvetlimo neki objekat jakom belom svetlošću, jedan deo objekta biće veoma svetao, gotovo beo (slika ??). Ako se pomerimo u odnosu na objekat, primetićemo da se taj sjajni deo objekta takođe izmešta. To je zato što sjajni objekti reflektuju svetlost različito u različitim pravcima. Efekat spekularne refleksije je najjači u pravcu koji je simetričan pravcu svetlosti u odnosu na normalu površi, i opada sa povećanjem ugla u odnosu na taj vektor. Stoga, efekat spekularne refleksije zavisi od pozicije kamere.

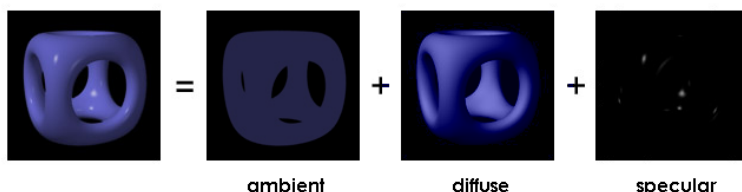


Slika 1.19: Efekat spekularne refleksije na sjajnim sferama (preuzeto sa Vikipedije).

Nasuprot tome, hrapave, matirane površine izgledaju jednako osvetljene iz svih uglova posmatranja i njihova osvetljenost ne zavisi od pozicije posmatrača (osvetljenost zida izgleda jednako iz svakog ugla posmatranja).

Ukoliko izvor svetla ne doprinosi direktno osvetljenosti neke primitive, intenzitet pridruženog osvetljenja bio bi jednak nuli. Međutim, želeli bismo da sve površi

vidljive iz pozicije kamere budu donekle osvetljene. Ovo se može postići dodavanjem ambijentalnog svetla na scenu značajno manjeg intenziteta, tj. dodavanjem ambijentalne komponente osvetljenosti ukupnoj vrednosti osvetljenosti.



Slika 1.20: Prikaz doprinosa ambijentalne, difuzne i spekularne komponente osvetljenosti ukupnoj osvetljenosti (preuzeto sa Vikipedije).

Dva najčešće korišćena modela osvetljenja u sebi uključuju difuznu, ambijentalnu i spekularnu komponentu osvetljenja i ukupna osvetljenost neke tačke se računa kao zbir ove tri komponente (slika ??).

Problem senčenja

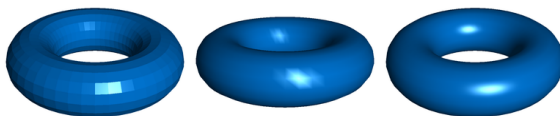
Kao što smo već pomenuli, problem senčenja se odnosi na proces bojenja kompletnog objekta, površi, odnosno poligona u trodimenzionoj sceni. Svaka površ se može senčiti tako što se u svakoj tački površi primeni odgovarajući model osvetljenja. Međutim, ovakav pristup bi bio jako skup i vremenski zahtevan.

Ravansko senčenje

Najjednostavniji i najefikasniji model senčenja jeste *ravansko (konstantno) senčenje* (engl. flat shading) kod koga se čitav poligon mreže trougla senči istom bojom (slika ?? levo). Na slici koja koristi ovaj vid senčenja uočljivi su pojedinačni poligoni mreže. Stoga ovaj model senčenja daje prihvatljiv prikaz samo ako mreža poligona potpuno verno opisuje sam model.

Interpolirano (Guroovo) senčenje

Osnovni nedostatak ravanskog senčenja može se otkloniti korišćenjem *interpoliranog (Guroovog) senčenja* (engl. Gouraud shading). U ovom modelu se za svako teme svakog trougla mreže računa njegova osvetljenost, dok se za tačke unutar trougla osvetljenost računa linearnom interpolacijom osvetljenosti njegovih temena (slika ?? sredina), korišćenjem baricentričnih koordinata.



Slika 1.21: Razlika između ravanskog, interpoliranog i Fongovog senčenja.

Fongov model senčenja

U slučaju obliha i glatkih objekata najbolji prikaz se dobija korišćenjem *Fongovog modela senčenja* (engl. Phong shading). U ovom modelu se vektor normale u tački unutar trougla računa interpolacijom vektora normala u temenima, a zatim se za svaki piksel računa njegova osvetljenost (slika ?? desno). Primetimo da je Fongov model senčenja računski mnogo zahtevniji od Guroovog jer se model osvetljenja primenjuje, umesto na svako teme, na svaku tačku unutar trougla.

Pitanja i zadaci za vežbu

Pitanje 1.1. Navesti bar tri različite primene računarske grafike.

Pitanje 1.2. Na koji način se pamti rasterska, a kako vektorska slika?

Pitanje 1.3. Navesti primer gde se u praksi koristi vektorska grafika, a gde rasterska.

Pitanje 1.4. Da li se kvalitet rasterske slike menja prilikom skaliranja slike? A vektorske?

Pitanje 1.5. Šta je rasterizacija?

Pitanje 1.6. Šta je alijasing efekat?

Pitanje 1.7. Objasniti razliku između 2D i 3D grafike.

Pitanje 1.8. Šta razlikuje interaktivnu od neinteraktivne računarske grafike?

Pitanje 1.9. Šta je zadatak faze modelovanja, a šta faze renderovanja?

Pitanje 1.10. Navesti bar tri parametra objekta koje treba modelovati?

Pitanje 1.11. Kojim se reprezentacijama najčešće modeluju pravouglo figure, a kojim glatke površi?

Pitanje 1.12. U čemu se razlikuju tačkasti izvor svetla i direkcioni izvor svetla?

Pitanje 1.13. Objasniti princip funkcionisanja kamere opskure.

Pitanje 1.14. Na koji način se dobija projekcija nekog objekta?

Pitanje 1.15. *Koja je razlika između perspektivne i projektivne projekcije?*

Pitanje 1.16. *Šta je zapremina pogleda? Čime je ograničena?*

Pitanje 1.17. *Koja su dva osnovna algoritma za utvrđivanje vidljivosti.*

Pitanje 1.18. *Kako funkcioniše z-bafer algoritam?*

Pitanje 1.19. *Na koji način se mešaju boje u aditivnim, a na koji način u subtraktivnim kolor modelima?*

Pitanje 1.20. *Koji se kolor model koristi kod monitora, a koji kod štampača?*

Pitanje 1.21. *Čemu služi tehnika preslikavanja tekstura?*

Pitanje 1.22. *Šta su baricentrične koordinate? Od kog su značaja u računarskoj grafici?*

1.2 Računarska vizija

Računarska vizija (engl. computer vision) je oblast veštačke inteligencije koja se bavi razvojem tehnika koje omogućavaju računarima da „vide“, tj. razumeju i interpretiraju sadržaj slika i video snimaka. Ono može uključivati generisanje tekstualnog opisa slike, detekcija objekata, generisanje trodimenzionalnog modela itd.

Ovaj problem, na prvi pogled, deluje jednostavno jer je za ljude on gotovo trivijalan. Čak i kad malom detetu pokažemo neku sliku, ono bez problema ume da prepozna da li se na njoj nalazi mačka, pas ili zec. Međutim, pokazuje se da za računar ovaj problem nije nimalo lak. To je zato nije u potpunosti izučen način funkcionisanja čovekovog vizualnog sistema i moždanih procesa kojima se tumači ono što čovek vidi. Ovaj problem je težak za računar i zato što je svet koji nas okružuje složen i potrebno je objekat prepoznati iz bilo kog ugla posmatranja, u svim uslovima osvetljenja i u situacijama kada je delimično zaklonjen drugim objektima.

Napomenimo i to da se računarska vizija razlikuje od obrade slika. Naime, obrada slike se svodi na digitalnu obradu signala i ne bavi se razumevanjem sadržaja slike.

1.2.1 Centralni problemi računarske vizije

Problemi kojima se bavi računarska vizija danas sve više dobijaju na značaju, jer smo u svakodnevnom životu gotovo preplavljeni slikama. Naime, pametnim telefonima se svakog momenta napravi ogroman broj fotografija i video snimaka,

a njihovo deljenje nikada nije bilo lakše. Samo na YouTube-u se svakog minuta postavlja na stotine sati novog materijala.

Neki od najznačajnijih zadataka računarske vizije su:

- * klasifikacija slika: prepoznavanje specifičnih pojmova na slici, poput lica, saobraćajnih znakova i sl;
- * detekcija objekata na slici: lociranje objekata unutar slike ili video materijala;
- * segmentacija slike: podela slike na veći broj segmenata, pri čemu svaki segment odgovara drugom objektu ili regionu.

1.2.2 Algoritam zasnovan na dubokom učenju

Postavlja se pitanje na koji način možemo postići da računar „vidi“ stvari na način na koji to vidi čovek. Kako bi to bilo moguće, neophodno je obučiti ga na velikom korpusu vizuelnog materijala: slika ili video materijala. Na primer, ako želimo da obučimo računar da prepoznaje lica, potrebno je da mu unapred prosledimo brojne relevantne slike, posebno one snimljene pri slabom osvetljenju ili pod različitim uglovima. Nakon toga se iz prosleđenih podataka vrši ekstrakcija korisnih informacija.

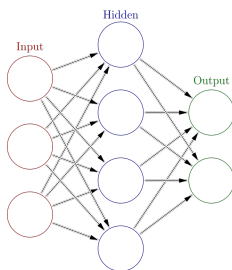
Ljudima nije teško da razumeju kontekst ili semantičko značenje slike, ali za računar slika nije ništa drugo do ogromna kolekcija piksela. Međutim, računari se mogu obučiti da otkriju i nauče ove informacije, algoritmima tzv. *dubokog učenja* (engl. deep learning) zasnovanim na *konvolucijskim neuronskim mrežama* (engl. convolutional neural network, CNN). Neuronske mreže se sastoje od određenog broja međusobno povezanih procesora (koje nazivamo *neuronima*) koji su organizovani u veći broj slojeva. Slojevi međusobno komuniciraju tako što se izlaz svakog od procesora iz prethodnog sloja povezuje sa ulazima svih procesora narednog sloja (slika ??). Konvolucijske neuronske mreže su posebno dizajnirane za obradu i analizu vizuelnih podataka, kao što su slike. Svaki od slojeva mreže izdvaja različite karakteristike iz piksela slike, počevši od nekih osnovnih, poput ivica, a postepeno prelazeći u sve složenije karakteristike kao što su oblici na slici. Poslednji slojevi mreže su sposobni da identifikuju specifične objekte na slici kao što su lica ljudi.

Konačno, CNN model formira matricu numeričkih vrednosti koje predstavljaju verovatnoću da je određeni objekat prisutan na slici: na primer verovatnoću da je na slici pas, verovatnoću da je u pitanju mačka i verovatnoću da je zec.

1.2.3 Primene računarske vizije

U poslednje vreme ostvaren je veliki progres na polju računarske vizije. Pokazuje se da tehnike računarskog vida mogu na različite načine unaprediti život čoveka.

Današnji svet se ne može zamisliti bez upotrebe interneta. Internet sadrži ogromne količine teksta i slika i njegov neraskidiv deo predstavljaju pretraživači



Slika 1.22: Primer neuronske mreže koja se sastoji iz tri nivoa: ulaznog, jednog skrivenog unutrašnjeg sloja i izlaznog sloja.

kojima se može doći do relevantnih informacija. Relativno je jednostavno indeksirati i pretraživati tekst, ali kako bi indeksirale i pretraživale slike, neophodno je da algoritmi prepoznaju šta se na njima nalazi.

Aplikacija Zoom koristi računarsku viziju za identifikaciju i maksimizovanje rezolucije najrelevantnijih delova videa, prevashodno ljudskih lica, a ne objekata u pozadini. Računarska vizija se koristi u robotici za razumevanje okruženja u cilju izvršavanja zadataka poput navigacije robota, hvatanje objekata i njihovog rukovanja. Koristi se i u autonomnim vozilima da omogući zadatke kao što su otkrivanje objekata (automobila, ljudi i drugih objekata na putu), zadržavanje trake pri vožnji i prepoznavanje saobraćajnih znakova. Računarska vizija je našla primenu i u medicinskom snimanju radi ranog otkrivanja bolesti, postavljanja dijagnoze i planiranja lečenja. Važna primena je i u sistemima za nadzor, za otkrivanje i praćenje ljudi ili vozila, kao i za prepoznavanje lica ili registarskih tablica. Naravno, računarska vizija je pronašla primenu i u računarskim igrama, za realizovanje zadataka kao što su prepoznavanje pokreta i prepoznavanje izraza lica radi poboljšanja interaktivnosti igara.

Ostvaren je veliki pomak kada su u pitanju sistemi za optičko prepoznavanje karaktera (engl. optical character recognition, OCR) i detekcija lica na kamerama i pametnim telefonima. Naime, do pre deset godina bi verovatno zvučalo nestvarno da će ljudi jednog dana otključavati svoje mobilne telefone na osnovu prepoznavanja lica.

Kao što vidimo, računarska vizija već ima širok spektar upotrebe u različitim oblastima. Računari su u stanju da obrađuju mnogo veći obim podataka mnogo većom brzinom nego što bi to čovek mogao. Ovo s jedne strane vodi poboljšanju efikasnosti i produktivnosti, a s druge strane omogućava automatizaciju zadataka koji bi inače morali da se obavljaju ručno. Računarska vizija može poboljšati bezbednost, korišćenjem sistema koji automatski nadgledaju javne prostore radi identifikovanja potencijalnih opasnosti po bezbednost ljudi.

Naravno, ostaje doza zabrinutosti oko primena računarske vizije. Naime, tehnologija prepoznavanja lica je predmet mnogih debata zbog zabrinutosti oko njene upotrebe od strane organa za sprovođenje zakona. Sistemi računarske vizije mogu

biti obučeni na pristrasnim podacima, što može rezultirati pristrasnim i diskriminatorskim ishodom. Sisteme zasnovane na računarskoj viziji je često teško interpretirati ili protumačiti. Kako bi se povećalo poverenje u njih, neophodno je omogućiti objašnjivost modela, tako da zainteresovane strane mogu da razumeju na koji način on funkcioniše i kako se donose odluke.

1.3 Računarska geometrija

Računarska geometrija je oblast računarstva koja se bavi dizajnom i analizom efikasnih algoritama i struktura podataka za rešavanje geometrijskih problema, uobičajeno u ravni ili trodimenzionalnom prostoru.

Za razliku od obrade slika, računarska geometrija se ne bavi samim slikama, već apstraktnim geometrijskim konfiguracijama, koje obuhvataju skupove objekata kao što su tačke, prave i ostale geometrijske figure.

Računarska geometrija se bavi problemima koji se često mogu jednostavno rešiti korišćenjem naivnog pristupa, međutim naivni algoritmi često nisu dovoljno efikasni na ulazima većih dimenzija. Stoga se računarska geometrija fokusira na razvoj efikasnih algoritama za rešavanje ovih problema.

Osnovni element u računarskoj geometriji je *tačka*: nju predstavljamo uređenim parom realnih brojeva ako je reč o geometriji ravni, odnosno trojkom realnih brojeva, ako je reč o geometriji 3D prostora. Putem tačaka predstavljamo druge osnovne objekte, npr. pravu kao par različitih tačaka koje joj pripadaju, duž parom njenih krajnjih tačaka, trougao njegovim temenim, mnogougao cikličnim poretком njegovih temena.

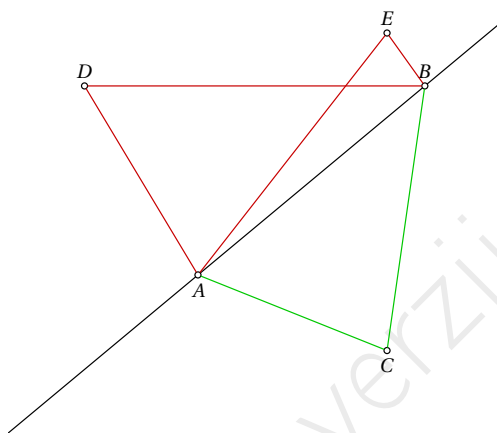
Drugi osnovni element računarske geometrije čine *vektori*, koje, kao i tačke, predstavljamo uređenim parom, odnosno trojkom realnih brojeva, koje odgovaraju njegovim koordinatama. Pokazuje se da se da skalarni i vektorski proizvod služe kao gradivni blok za efikasno rešavanje širokog spektra geometrijskih problema u računarskoj geometriji. Naime, skalarni proizvod se koristi za računanje ugla između dva vektora (samim tim i utvrđivanje da li su vektori ortogonalni) i za računanje projekcija jednog vektora na drugi, dok vektorski proizvod omogućava računanje površine paralelograma koji vektori razapinju, računanje orijentacije trojke tačaka, kao i utvrđivanje različitih odnosa geometrijskih primitiva, poput ispitivanja da li je tačka u poligonu, da li se duži seku i sl.

Za potrebe rešavanja problema iz ove oblasti razvijene su namenske, prostorne strukture podataka, poput *kd-stabala* (engl. kd-tree), koja pomažu u rešavanju problema određivanja najbližeg suseda ili upita opsega, *kvodtri* (engl. quadtree) i *oktri* (engl. octree) za particionisanje prostora, *binarna stabla prostornog particionisanja* (engl. binary space partition tree) za rešavanje problema iscrtavanja i vidljivosti.

1.3.1 Centralni problemi računarske geometrije

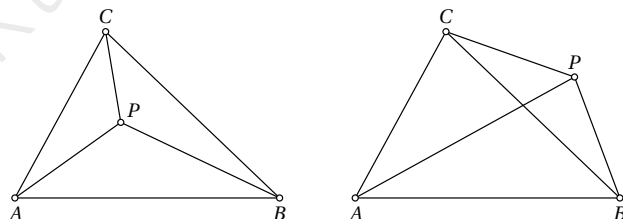
Računarska geometrija uključuje rešavanje različitih algoritamskih problema nad geometrijskim objektima poput tačaka, pravih, mnogouglova i drugih geometrijskih objekata.

Jedan od elementarnih zadataka u računarskoj geometriji i važan element složenijih algoritama jeste ispitivanje da li se neka tačka nalazi levo ili desno u odnosu na dati zrak (slika ??).



Slika 1.23: Tačke D i E se nalaze sa leve strane zraka AB , dok se tačka C nalazi sa desne strane.

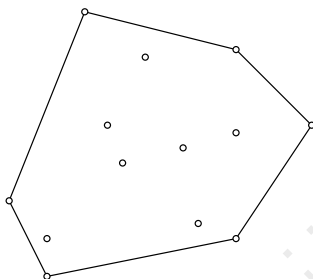
Često razmatran problem je i ispitivanje da li se tačka nalazi unutar neke geometrijske figure, npr. konveksnog ili nekonveksnog prostog mnogougla. Na slici ?? dat je prikaz slučaja kada se tačka nalazi i kada se ne nalazi u trouglu.



Slika 1.24: Situacija kada tačka P pripada i kada ne pripada trouglu ABC .

Jedan od osnovnih problema računarske geometrije je računanje *konveksnog omotača* (engl. convex hull) skupa tačaka, čiji je zadatak da za dati skup tačaka

odredi najmanji konveksni mnogougao kojem sve tačke pripadaju (slika ??). Pokazuje se da je konveksni omotač skupa tačaka jedinstven i da su mu temena neka od tačaka skupa. Za problem računanja konveksnog omotača se nekad kaže da je „problem sortiranja” u računarskoj geometriji, u smislu da je ovo jedan od fundamentalnih problema, da postoji veliki broj algoritama za njegovo rešavanje, a neki čak analogni algoritmima sortiranja.



Slika 1.25: Konveksni omotač datog skupa tačaka.

Još neki bazični problemi računarske grafike su: utvrđivanje dve najbliže i dve najudaljenije tačke iz skupa od n datih tačaka, ispitivanje da li se dva data poligona seku i računanje njihovog preseka i da se za dati skup tačaka odrede one koje se nalaze unutar nekog opsega (npr. pravougaonika ili kruga).

Osnovni princip kojim računari barataju složenim objektima jeste da ih razlože na veliki broj jednostavnijih objekata. Najjednostavniji objekat u ravni jeste trougao. Pokazuje se da podela mnogougla na trougle, tzv. *triangulacija* (engl. *triangulation*) može pomoći u rešavanju različitih problema.

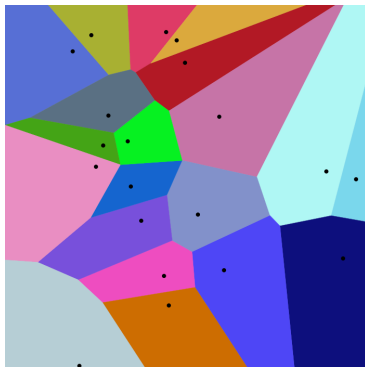
Iako se računarska geometrija često fokusira na optimizaciju rešenja za naizgled jednostavne probleme, ona se takođe bavi i izazovnijim problemima, koji zahtevaju sofisticirane algoritme i inovativne tehnike za efikasno rešavanje. Jedan od takvih problema je *Voronoi dijagram* (engl. *Voronoi diagram*, slika ??) u kome je potrebno podeliti prostor na regione na osnovu rastojanja do skupa datih tačaka, tako da svaki region sadrži sve tačke koje su bliže jednoj konkretnoj tački od ostalih.

Algoritmi računarske geometrije su podložni greškama zaokruživanja. Naime, algoritmi koji mogu biti jednostavni za opisivanje na papiru mogu biti veoma suptilne detalje i zbog toga biti teški za implementaciju.

1.3.2 Primene računarske geometrije

Računarska geometrija ima primenu u raznim oblastima.

Ona igra fundamentalnu ulogu u računarskoj grafici, omogućavajući kreiranje, manipulaciju i prikazivanje 2D i 3D scena. Na primer, utvrđivanje preseka zraka sa geometrijskim objektima igra važnu ulogu u renderovanju realističnog osvetljenja,



Slika 1.26: Voronoi dijagram datog skupa tačaka (preuzeto sa https://en.wikipedia.org/wiki/Voronoi_diagram).

refleksija i senki. Procesom triangulacije se proizvoljne geometrije mogu transformisati u mreže trouglova, koje su osnova reprezentacija objekata u grafici. Algoritmi računarske geometrije igraju važnu ulogu i u animacijama, npr. za detekciju kolizije.

U robotici ovi algoritmi imaju primenu u planiranju kretanja robota i za ispitivanje vidljivosti. Tipičan primer primene je sledeći: dat je skup prepreka oblika mnogougla u ravni i potrebno je pronaći najkraći put od startne do ciljne pozicije koja izbegava sve prepreke.

Računarska geometrija ima primenu i u geografskim informacionim sistemima (engl. geographic information system, GIS) za efikasno lociranje objekata na mapi, prikazivanje odgovarajućeg dela mape i planiranje trasa kretanja. Ona je integralni deo dizajna i proizvodnje vođene računarima (engl. CAD, CAM), omogućavajući precizno modelovanje i fabrikaciju složenih objekata.

Ona nalazi primenu i u naučnim izračunavanjima, na primer simulacije čestica: za modelovanje gravitacionih i elektromagnetnih interakcija, u oblasti biologije i hemije za analizu prostornog rasporeda i savijanja biomolekula, u dinamici fluida za simulaciju kretanja fluida i interakciju fluida sa površima.

Mnoge algoritamske tehnike su osmišljene za rešavanje geometrijskih problema, a ovi problemi imaju primenu u mnogim drugim oblastima. Na primer, *tehnika pokretne (brišuće) prave* (engl. sweeping line) kojom se pregleda cela ravan je jedna od uobičajenih tehnika u računarskoj grafici kojom se vrši pregledanje neke geometrijske figure (mnogougla, skupa tačaka ili cele ravni) tako što je prava prebriše.

1.4 Robotika

Robotika (engl. robotics) je naučna disciplina koja se bavi dizajnom, konstrukcijom i primenom robota, kao i razvojem računarskih sistema za njihovu kontrolu i obradu prikupljenih podataka.

Roboti su mašine programirane da samostalno ili uz malu pomoć čoveka izvršavaju različite zadatke, kao što su usisavanje kuće ili snimanje okoline iz vazduha.

Svet robotike je istovremeno raznolik i fascinant. Postoje različite vrste robota koje igraju važne uloge u našem svetu. Od humanoidnih robota koji podsećaju na ljude, do autonomnih robota koji se samostalno kreću, roboti dolaze u različitim oblicima, a svaki je osmišljen da obavlja određeni zadatak.

Prevažni cilj robotike je dizajn mašina koje mogu pomoći ljudima u izvršavanju nekih praktičnih zadataka. Neki roboti su napravljeni da obavljaju poslove koji su opasni za ljude, kao što je pronalaženje preživelih u nestabilnim ruševinama i istraživanje svemira, drugi zamenjuju ljude na poslovima koji su dosadni, naporni ili repetitivni, kao što su čišćenje, nadgledanje, montaža. Robotika je oblast koja se razvija uporedo sa razvojem tehnologije.

Inteligencija koju robot poseduje zapravo je skup programa koji mu omogućava da prepozna određene situacije i da se u njima snađe, tj. da na odgovarajući način reaguje. Štaviše, robot može da uči i iz sopstvenog iskustva, prilagođavajući se novim situacijama i uspešno rešavajući nove probleme.

Roboti se mogu podeliti prema različitim kriterijumima. Prema stepenu samostalnosti oni se dele na *industrijske robote*, koji se kreću po skupu diskretnih lokacija i *autonomne robote*, koji mogu da interaguju sa okruženjem.

Roboti koji po izgledu i ponašanju oponašaju ljude nazivaju se *humanoidni roboti*. Oni se često koriste u istraživanju i razvoju, za korisničke servise i u industriji zabave. *Kolaborativni roboti* ili *koboti* su dizajnirani da bezbedno rade direktno zajedno sa ljudima u proizvodnim procesima, bez potrebe za zaštitnim barijerama. U mnogim sektorima, tradicionalni industrijski roboti se dopunjuju ili zamenjuju kobotima. Umesto da zamene ljudske poslove, koboti ih unapređuju omogućavajući radnicima da preuzmu nove odgovornosti poput kontrole kvaliteta na kraju automatizovanih proizvodnih procesa.

Naravno, roboti nisu nužno humanoidni, već mogu da imaju različite oblike i funkcionalnosti, u zavisnosti od njihove namene. Na primer, roboti su dizajnirani za istraživanje svemira, dronovi se koriste za istraživanja iz vazduha, softverski botovi obavljaju zadatke u virtuelnom okruženju. Naime, roboti nisu definisani njihovim izgledom, već njihovom sposobnošću da obavljaju zadatke autonomno ili poluautonomno koristeći računarsku inteligenciju.



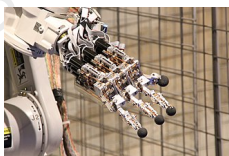
Slika 1.27: Humanoidni robot TOPIO igra stoni tenis na internacionalnoj izložbi robota IREX 2009. (preuzeto sa <https://en.wikipedia.org/wiki/Robotics>).

1.4.1 Komponente robota

Robotika je interdisciplinarna oblast koja u sebi objedinjuje razne discipline kao što su računarstvo, mašinstvo, elektronika, mehatronika i dr. Tri ključna elementa u razvoju robotskih sistema su mehanički dizajn, koji robotu daje oblik i strukturu; električne komponente, koje ga pokreću i omogućavaju mu da radi; i softver, koji mu daje „mozak” za obavljanje zadataka.

Mehanička konstrukcija robota uključuje dizajniranje okvira, odnosno strukture prilagođene za obavljanje određenog zadatka, u skladu sa fizičkim karakteristikama okruženja (slika ??).

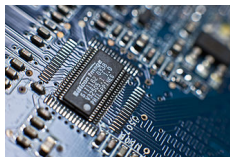
Na primer, robot koji je dizajniran da se kreće po neravnom terenu može biti opremljen gusenicama.



Slika 1.28: Mehnička šaka robota (preuzeto sa <https://en.wikipedia.org/wiki/Robotics>).

Električne komponente obezbeđuju snagu i kontrolu koje su neophodne za funkcionisanje robota (slika ??).

Prethodno pomenutom robotu sa gusenicama je potrebna električna energija da pomera svoje gazišta. Električni aspekt igra ključnu ulogu u kretanju robota (putem motora), njegovom opažanju okoline (detekcijom toplote, zvuka, položaja i dr.) i funkcionisanju (snabdevanjem motora i senzora energijom potrebnom za aktivaciju i izvršavanje osnovnih zadataka).



Slika 1.29: Integrirano električno kolo (preuzeto sa <https://en.wikipedia.org/wiki/Robotics>).

Konačno, robot izvršava neki program, kojim se definiše niz odluka kada ili kako treba da učini nešto.

Softver predstavlja srž robota koji obezbeđuje da on ispravno autonomno ili poluautonomno izvršava zadatak za koji je predviđen. On integriše različite tehnologije i kombinuje algoritme veštačke inteligencije, računarske geometrije i specijalizovane algoritme.

Veštačka inteligencija omogućava robotima da uče, da se prilagođavaju i donose odluke. Geometrija igra ključnu ulogu u razumevanju okruženja i komunikacije sa njom, poput određivanja optimalne putanje, za detekciju kolizije, modelovanje konstrukcije robota, i slično.

Robot je sastavljen od komponenti koje omogućavaju njegovo funkcionisanje i obavljanje zadatka za koji je predviđen. Jedan od osnovnih elemenata robota su *senzori* koji mu omogućavaju da prikupi informacije o svom okruženju, poput količine svetlosti, temperature, zvuka. Druga važna komponenta su *aktuatori* koji predstavljaju pokretačku snagu robota, pretvarajući električnu ili mehaničku energiju u kretanje, silu ili zvuk. Aktuatori omogućavaju izvođenje predviđenih radnji i komunikaciju sa spoljašnjim svetom. Veza između senzora i aktuatora se ostvaruje putem kontrolne jedinice, koja tumači podatke prikupljene putem senzora, donosi odgovarajuće odluke i prosleđuje ih aktuatorima. Konačno, za rad robota potrebno je napajanje, koje robotu obezbeđuje potrebnu energiju za rad. On pretvara električnu energiju u napon, tj. struju koja je robotu potrebna.

1.4.2 Primene robotike

Primene robotike su brojne i raznovrsne.

Roboti se sve više koriste u industriji, a takvi roboti poznati su kao *industrijski roboti*. Oni se primenjuju za sastavljanje i obradu proizvoda u proizvodnim okruženjima. Većina industrijskih robota sastoji se od robotske ruke, hvataljke, senzora i kontrolne jedinice. Kako roboti postaju sve češći deo radne snage, mnoge fabrike postaju potpuno automatizovane.

Roboti danas postaju sve popularniji i u domaćinstvu jer mogu da obave neki od svakodnevnih kućnih poslova i na taj način uštede vreme i energiju ljudi, na primer usisavanje i košenje travnjaka.

Roboti se koriste danas i u zdravstvu za izvršavanje različitih zadataka, kao što je pomoć prilikom izvođenja operacionih zahvata, davanje lekova, transport pacijenata u bolnicama ili u rehabilitaciji.

Roboti se sve više koriste u autonomnom transportu, gde preuzimaju ključnu ulogu u povećanju efikasnosti i bezbednosti saobraćaja. Na primer, samovozeći automobili koriste sofisticirane senzore, kamere i algoritme veštačke inteligencije kako bi detektovali okolinu i doneli odluke u realnom vremenu, omogućavajući vozilima da se kreću bez ljudske intervencije. Autopiloti aviona omogućavaju automatsko upravljanje letom, od polaska do sletanja, smanjujući mogućnost ljudske greške. Ovakvi robotski sistemi ne samo da smanjuju potrebu za intervencijom ljudi, već i unapređuju sigurnost, smanjuju troškove i povećavaju efikasnost u transportu i logistici.

Postoje i različiti tipovi građevinskih robota koji se koriste za unapređenje efikasnosti i bezbednosti u građevinskoj industriji. Među njima su tradicionalni roboti, robotske ruke i robotski egzoskeleti, svaki sa specifičnim zadacima i primenama. Tradicionalni roboti se obično koriste za obavljanje teških fizičkih poslova, kao što su podizanje, premeštanje i transport materijala, čime se smanjuje potreba za ljudskom radnom snagom u opasnim ili napornim zadacima. Robotske ruke u građevini služe za izvođenje preciznih operacija, poput montaže delova, zavrtnja i lepljenja. Robotski egzoskeleti predstavljaju nosive robote koji omogućavaju radnicima da podnose veće težine i manje se umaraju tokom napornih fizičkih aktivnosti.

Roboti se široko koriste i za istraživanje svemira, a jedan od najpoznatijih primera su roveri na Marsu (slika ??). Ovi roboti omogućavaju istraživanje planeta i meseca na način koji je nemoguć za ljudske misije zbog ekstremnih uslova, kao što su visoke temperature, velika radijacija i nedostatak atmosfere.



Slika 1.30: Robotičari sa tri rovera za istraživanje na Marsu (preuzeto sa <https://en.wikipedia.org/wiki/Robotics>).