

## **1. PIKSELI, REZOLUCIJA**

Monitor je rasterski uđaj, a slika na njemu se sastoji iz redova i kolona tačaka, koje se nazivaju pikseli. Piksel je standardizovana jedinica za meru, isto kao metar ili inč. Svaki piksel na slici je savršen kvadrat na savršenoj mreži, koji je ravnomerno obojen jednom jedinom bojom. Slika ima tri atributa: -velčinu (datoteke) -mera koja pokazuje koliko piksela sadrži slika, -rezoluciju (broj piksela koji se prikazuju ili štampaju po linearnom inču ppi), -dimenziju (fizička širina i visina slike). Pojam rezolucije se u vezi sa monitorima tumači dvojako: \_rezolucija ili rezolucija slike je gustina piksela na ekranu. Ona može varirati od 50 piksela/inču do 85 piksela/inču mada je opšte prihvaćeni standard 72 piksela/inču. Postoje monitori sa veoma velikim rezolucijama od 100, 150 pa čak i 200 piksela/inču ali je njihova cena veoma velika tako da su veoma retko u upotrebi. -Raster dimenzija ili rezolucija ekrana predstavlja veličinu slike koju monitor može da prikaže. Ona se izražava u broju piksela po širini i dužini koji mogu da stanu na ekran i zavisi na od monitora do monitora. Najčešća rezolucije koji standardni 14 «monitori pdražavaju su od 640\*480 DO 1024\*768. Veći monitori 17» i 21» podržavaju i veće rezolucije sve do 1600\*1200.

**2. RASTERSKA GRAFIKA** ili bitmap grafika, jednostavno rečeno, odnosi se na grafiku proizvedeno od velikog broja piksela. Rasterska slika se sastoji iz redova i kolona piksela složenih u matrici koji čine sliku. Pojmovi rezolucije koji su definisani kod monitora na sličan način se koriste i za bitmape: -Raster rezolucija se meri DPI jedinicama i govori nam koliki je kvalitet grafike. Što je veći broj piksela po inču to je slika kvalitetnija i obrnuto. -Sa druge strane, raster idmenzijom se određuje veličina grafike. Tu se misli na dužinu i širinu slike. -Treći bitan lement za određivanje grafike je broj boja koje nosi svaki piksel je dubina boje. Broj boja

je veoma bitan činilac za određivanje krnjeg kvaliteta slike i on se kreće od 256 do preko 32 miliona boja.

### **VEKTORSKA GRAFIKA**

se satoji iz serije proračunatih matematičkih jedanačina. Jednostavnije rečeno, sastoji se od poligon-objekata kao što su-tekst, - linije (krive i prave), -Pravogonici ili vkadrati, -Krugovi ili elipse. Boja se kod vektorske

grafike određuje po dimenzijama površina objekata iz kojih se sastoji grafika, tako što kod, koji ide sa slikom, govori programu koja boja se nalazi u kojim dimenzijama poligona. Prednost vektorske grafike je, što se veoma lako vrši promena njene veličine bez gubitka na kvalitetu. To je moguće pošto je vektor grafika, kao što je već rečeno, skup matematičkih jedanačina prenesenih na ekran, tako da preračunavanjem jedanačina po novim parametrima dobijamo efekat povećanja ili smanjenja

Neprotivurečne su (nekontradiktornost). Potrebno je iz sistema aksioma pokriti sve moguće posledice – potpunost. Lobačevski je umesto V postulata uveo sledeći (uopštenje): U ravni kroz datu tačku van date prave prolaze bar dve prave koje ne seku datu pravu.

Neprotivurečnost geometrije Lobačevskog posledica je neprotivurečnosti euklidske geometrije. Neprotivurečnost euklidske geometrije dokazao je Hilbert na osnovu protivurečnosti aritmetike. Drugačija neeuklidka geometrija se dobija ako se umesto V postulata uzme sledeća Rimanova aksioma paralelnosti: Kroz tačku A van prave a ne prolazi ni jedna prava koja ne seče pravu a.. Riman uvodi termin mnogostrukost ili mnogovrsnost. Element mnogovrsnosti je tačka, čiji se položaj karakteriše koordinatama  $x_1, x_2, \dots, x_n$ . Najprostiji slučajevi mnogovrsnosti su prostori konstantne krivine,  $K = \text{const}$ . Razlikuju se tri slučaja takvih prostora: K>

0, geometrija prostora pozitivne krivine - Rimanova geometrija u užem smislu ili eliptična geometrija.  $K=0$ , geometrija prostora nulte krivine - parabolična geometrija, koja se pod uslovom o pozitivnosti elemenata dužine poklapa sa geometrijom Eukl  $K<0$ , geometrija prostora negativne krivine - hiperbolička geometrija ili geometrija Lobačevskog..

**3. EUKLIDSKE I NEEUKLIDSKE GEOMETRIJE** – Euklid je u 4 veku pre n. ere svoje rezultate izložio u delu “Elementi” u 15 tomova. Po prvi put je izvršena sistematizacija. Aksiome su podeljene u grupe (Euklid): 1. Aksiome incidencije (9 aksioma), 2. Aksiome poretka (6 aksioma), 3. Aksiome podudarnosti (7 aksioma), 4. Aksiome neprekidnosti (2 aksiome), 5. Aksiome paralelnosti (1 aksioma) . Ove aksiome su nezavisne jedne od drugih.

**4. ELEMENTI ANALITICKE GEOMETRIJE** - Vektori su orijentisne duzi. Oni se mogu predstaviti svojim koordinatama. Def.  $A_1, a_2, \dots, a_n$ - vektori i  $k_1, k_2, \dots, k_n$  brojevi. Neka je bar jedan broj razlicit od 0, tada je linearna kombinacija  $k_1 a_1 + k_2 a_2 + \dots + k_n a_n = 0$  oni su linearno zavisni, a ako takav broj ne postoji onda su linearno nezavisni. T1: sistem vektora je linearno zavisan, akko se najmanje jedan vektor iz tog sistema moze izraziti preko ostalih vektora. T2: Vektori a i b su kolinearni, akko su linearno zavisni. T4: U  
prostoru  $E^3$  postoje 3 vektora koja su medjusobno linearno nezavisni, a svaka 4 vektora u ovom prostoru su linearno zavisni.  $E^n$  – n vektora su linearno nezavisni;  $E^{n+1}$

-  $n+1$  vektora su zavisni. Forme analitickog predstavljanja geometrijskih objekata: 1. eksplicitna forma; 2. implicitna forma; 3. parametarska forma.

**5. KOORDINATNI SISTEMI**-Položaj neke tačke na pravoj, u ravni, u prostoru može da se zapiše uz pomoć pogodno odabranih brojeva, tzv. Koordinata. Koordinate uspostavljaju most između geometrije i aritmetike i predstavljaju osnovu za rešavanje mnogih geometrijskih problema. Koordinatni sistemi: -homogeni-nehomogeni. Sistemi se dele: -Dekartov koo. Sistem-Polarni koo. Sist.-Cilindrični koo.sist.-Sferni koo.sistem

**6. KONAČNO PROJEKTNE RAVNI** Neka  $q$  označava broj veći od 1. Konačno projektne ravnine  $q$ -tog reda definiše se preko 4 aksioma: 1. za svake dve razne tačke postoji tačno jedna zajednička prava 2. Za svake dve razne prave tačno jedna zajednička tačka 3. Postoji potpun 4-temenik 4-Postoji prava koja ima  $q+1$  tačaka. U konačnoj projektnoj ravni važe sledeći stavovi:-  
svaka prava

**7. DIGITALNA GEOMETIJA** Kompujerska (računarska) geometrija je skup geometrijskih disciplina okrenut realizaciji na računaru. Glavni predmet izučavanja su geometrijski algoritmi, kod kojih se postavlja zahtev efikasnosti. Digitalna geometrija je podoblast kompujerske geometrije koja se ograničava na digitalne objekte-objekta celobrojnoj mreži. Sva razmatranja su ograničena na tačke sa celobrojnim koordinatama. Pošto se kompujerski ekran može shvatiti kao deo 2-dimenzionalne mreže sledi da će se rešavanje određenih problema u digitalnoj ravni koristiti u kompujerskoj grafici za prikaz slike na ekranu

**8. DIGITALIZACIJA** Neka je:  $A$ -euklidska konstantna slika,  $A'$ -unija svih piksela koji imaju neprazan presek sa  $A$ ,  $A''$ -skup svih tačaka pravougaone celobrojne mreže, koje su centri piksela iz  $A'$ .  $A''$  je digitalizacija od  $A$ . Digitalizacija se izvodi u dve faze. Problem digitalizacije: Za datu euklidsku sliku  $S$ , odrediti digitalno preslikavanje  $f$ , tako da se  $f(S)$  u određenom smislu ponaša što sličnije ulaznoj slici  $S$ . **PROSTA DIGITALIZACIJA DUŽI** Neka je u koordinatnom sistemu \*celobrojnoj kvadratnoj mreži, gridu, rasteru) data duž sa krajnjim tačkama  $(x_1, y_1)$  i  $(x_2, y_2)$ , pri čemu su koordinate  $x_1, y_1, x_2, y_2$  celobrojne. Neka je:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$y_1$

Digitalna duž  $(x_1, y_1), (x_2, y_2)$ , može se definisati sledećim algoritmom:  $x := x_1, y := y_1$ ; if

$x >= y$  then

for  $x := x_1 + 1$  to  $x_2$  do begin

algoritam digitalizacije prilagodi svakom položaju prave u ravni. **BRESENHAM-OV ALGORITAM** Prosta digitalizacija duži ima jednu dosta veliku manu. Iako je algoritam namenjen radu sa celobrojnim koordinatama, navedeni algoritam koristi i realnu aritmetiku, jer je korišćen nagib  $m$  realan broj (nastao deljenjem celih brojeva  $y$  i  $x$ , tj.  $y/x$ ). To se negativno odražava na efikasnost izvršavanja algoritma. Upotreba realne aritmetike potpuno je izbegnuta Bresenham-ovim algoritmom. Radi jednostavnosti razmatraće se slučaj:

$x \geq$

Kada je u pitanju neprekidna kriva, dve osnovne šeme digitalizacije su: Na svakoj celobrojnoj apscisi (x-koordinati) uzeti prvu celobrojnu tačku ispod krive najbližu celobrojnu tačku, iznad ili ispod krive. Ove dve šeme su ekvivalentne; jedna se iz druge može dobiti translacijom u vertikalnom pravcu za  $\frac{1}{2}$ . Po potrebi se kod strmo nagnutih krivih uloge x-koordinata i y-koordinata mogu zameniti.

$x := x + 1/m$ ; writeln (round(x), y) end; Primenjena je šema digitalizacije sa zaokruživanjem. Korišćenjem apsolutnih vrednosti za  $x$  i  $y$  obezbeđeno je da se jedinstveni

$y := y + 1/m$ ; writeln (x, round(y)) end else for  $y := y + 1$  to 2 do begin

ima  $q+1$  tačku - svaka tačka je incidentna sa  $q=1$  pravih/ravan ima  $q^2+q+1$  tačaka i pravih. Za sve stepene prostih brojeva  $q$  obezbeđena je geometrija Galoa-ove konačne projektivne ravni reda  $q$ , za koju postoji aritmetički model nad Galoa-ovim konačnim poljem  $GF(q)$ .

**9. ALGORITAM ISECANJA** Cilj operacije isecanja je prikazivanje samo one slike koja se nalazi unutar prozora. Kada je u pitanju isecanje teksta razlikuju se 3 nivoa: nivo stringa, nivo slova, detaljni nivo. Glavna ideja Koen-Saderlendovog algoritma isecanja s obzirom na prozor određen krajnjim tačkama ( $x_{min}, y_{min}$ ) i ( $x_{max}, y_{max}$ ) sadržana je u kodiranju 9 oblasti ravni s obzirom na prozor po sledećoj šemi (1,0,0,1) (1,0,0,0) (1,0,1,0)

(0,1,0,1) (0,1,0,0) (0,1,1,0). Unutrašnjost prozora je kodirana nula vektorom. Za svaku tačku ravni se formira 4-bitni kod (iznad, ispod, desno-od, levo-od) Algoritam testira da li se razmatrana duž treba u celosti da se prikaže ili ne. Ukoliko to nije slučaj prelazi se na postupno odsecanje delova duži koje štrče izvan prozora.

(0,0,0,1) (0,0,0,0) (0,0,1,0)

## **10. Algoritmi za popunjavanje mnogouglova**

Neke ideje za popunjavanje oivičenih površi, unutrašnjosti zadatih kontura u rasteru:

Algoritam poplave (flood fill) - rekurzivan postupak: PROCEDURE oboji (x,y: INTEGER); BEGIN

IF NOT (x,y) granični piksel THEN

BEGIN dodeli boju pikselu (x,y);

oboji (x+1, y);

oboji (x-1, y);

oboji (x, y-1);

oboji (x, y+1);

END END;

Ivični postupak - iterativni postupak : Zrak udesno se može realizovati petljom:

y := y + 1;

WHILE NOT (x,y) granični DO BEGIN

## **11.Kornjacina grafika**

Prelazak iz tekstualnog režima rada u grafički režim rada na Turbo PROLOG-u se vrši pomoću predikata graphics. Kada se pređe u grafički način rada ekran se obriše i "kornjača" (turtle) se postavlja u sredinu ekrana. Pomoću ugrađenih standardnih predikata može se odrediti pozicija kornjače, pravac i smer njenog pomeranja, kao i mogućnost crtanja, tj. ostavljanja traga njenog kretanja u određenoj boji.

U ovu grupu spadaju predikati: forward, back, right, left, pendown, penup, pencolour.

Prikazati na sredini ekrana kvadrat u koji je upisan drugi kvadrat čija temena polove stranice većeg kvadrata. Prema tome, ako se sa  $a$  obeleži dužina stranice većeg kvadrata, onda se dužina stranice manjeg kvadrata koja je obeležena sa  $b$ , dobija:  $b = a / \sqrt{2}$ .

## **12.Graficki formati datoteka**

Grafički formati predstavljaju način memorisanja neke slike. Postoji više vrsta formata od kojih svaki ima svoje prednosti i nedostatke. Svaki od formata je podesan za različitu primenu. Najpoznatiji formati su GIFF, JPEG, BMP i TIFF. Svaki od formata koristi različite metode kompresije radi smanjenja veličine fajla. Postoje dva načina kompresije: sa i bez gubitaka. Ovo

se pre svega odnosi na to da li kompresija utiče na gubljene podataka slike. Ako je kompresija sa gubicima tada se gubi i kvalitet slike. U zavisnosti od slike koristiće se jedna ili druga kompresija, na primer ako su gubici mali da ne utiču bitno na kvalitet slike može se koristiti i kompresija sa gubicima, prevashodno zbog toga što je tada stepen kompresije veći nego kod kompresije bez gubitaka.

GIF fajlovi se najčešće mogu naći na internetu, uglavnom na Web-u. To je format kreiran od strane CompuServe-a. GIF-ovi su 8-bitni, u boji ili kao slike u sivim nijansama.

JPEG je format fajla i metod za kompresiju koji omogućuje velike uštede u veličini fajla.

TIFF je najčešće korišćeni format za skenirane slike u pripremi štampe.

BMP Format podržan od strane većine Windows aplikacija. Razvijen od strane Microsofta. Podržava 24-bitne slike (16.7 miliona boja). Koristi se za razvoj Windows aplikacija.

### **13.GRAFIČKE METODE**

Metoda platna odnosno Canvas-a

Sva crtanja linija i iscrtavanja tekstova ostvaruju se odgovorajućim metodama klase TCanvas. U nasatavku navedene su neke važnije metode.

Za crtanje linija i za ivice zatvorenih površina uvek se koriste trenutni parametri u svojstvu Pen (pero), zatvorene površine se popunjavaju prema parametrima u svojstvu Brush (četkica), a tekstovi se ispisuju shodno parametrima u svojstvu Font (vrsta slova).

procedure LineTo(X, Y: Integer); - linija do

procedure MoveTo(X, Y: Integer); - pomeri se do

Prva metoda crta pravu liniju od trenutnog položaja pera (vrednost svojstva PenPos) do tačke sa koordinatama X i Y. Druga metoda samo pomera pero, ne ostavljajući traga na ekranu. Vrednost svojstva PenPos se menja da ukaže na novi položaj pera.

Metoda FillRect služi da oboji pravouganik trenutnom bojom četkice (brush) njena deklaracija je:  
procedure FillRect(const Rect: TRect);

Metode kao što su:

procedure PolyLine(Tacke: array of TPoint); - otvorena izlomljena linja

procedure Polygon(Tacke: array of TPoint); - zatvorena izlomljena linja

Prva metoda crta izlomljenu linju spajajući susedne tačke u nizu Tačke. Druga metoda radi isto,

ali na kraju spaja poslednju tačku sa prvom i boji unutrašnjost dobijene zatvorene površine.

Metode:

procedure      Rectangle(X1,Y1,X2,Y2: Integer); - crta pravouganik

procedure      RoundRect(X1,Y1,X2,Y2,X3,Y3:Integer);-zaobljeni pravouganik

Obe metode crtaju i popunjavaju pravougaonik sa gornjim levim uglom u tački (X1,Y1) i donjim levim uglom u tački (X2,Y2). Druga metoda temena pravouganika crta zaobljeno, kao četvtine elipse visine X3 i širine Y3.

Metoda

procedure      Ellipse (X1, Y1, X2, Y2:Integer); - elipsa

Ova metoda crta i popunjava elipsu koja je omeđana pravouganikom sa gornjim levim uglom u tački (X1,Y1) i donjim levim uglom u tački (X2,Y2).

Metode:

procedure Arc(X1,Y1,X2,Y2,X3,Y3,X4,Y4:LongInt); - crta luk

procedure Chord(X1,Y1,X2,Y2,X3,Y3,X4,Y4:LongInt); - odsečak

procedure Pie(X1,Y1,X2,Y2,X3,Y3,X4,Y4:LongInt); - crta isečak

Prva metoda crta luk kao deo elipse koja je omeđana pravouganikom sa gornjim levim uglom u tački (X1,Y1) i donjim levim uglom u tački (X2,Y2). Početna tačka luka je u preseku elipse i prave koja prolazi kroz centar elipse i kroz tačku (X3,Y3).

Krajnja tačka luka je u preseku elipse i prave koja prolazi kroz centar elipse i tačku (X4,Y4). Luk se crta u suprotnom smeru od kretanja kazaljki na časovniku.

Druga metoda crta odsečak elipse (zatvorena površina) koja se dobija spajanjem krajnjih tačaka luka, koji bi crtala prva metoda, pravom linijom.

Treća metoda crta isečak elipse (zatvorena površina) koja se dobija spajanjem krajnjih tačaka luka koji bi crtala prva metoda sa centrom elipse.

Metode:

procedure TextOut (X,Y: Integer; const: Tekst: string); - izlaz teksta

procedure TextRect (Pravoug: TRect; X,Y: Integer;

const: Tekst: string); - izlaz teksta unutar pravouganka

Obe metode iscrtavaju tekst u argumentu Teks sa gornjim levim uglom u tački (X,Y). Druga metoda odseca delove teksta koji bi bili izvan pravougaonika predstavljenog argumentom Pravoug.

Metode drugih komponentenata i crtanje

Sledeće tri metode, koje se tiču isrtavanja komponentenata na ekranu, poseduju sve komponente i obrasci. Ranije nisu bile objašnjene jer njihova upotreba do ovog trenutka nije bila neophodna. Sve tri metode su procedure bey argumenata.

Metoda Invalidate (učini nevažećim) označava da komponentu (uključujući i obrazac) za koju je pozvana treba iznova isrtavati čim to bude moguće (čim Windows ne bude imao nešto hitnije da radi). Kao posledica, pozvaće se rukovalac događajima OnPaint (prilikom isrtavanja). Ovu metodu treba pozivati u situacijama kad se proceni da komponenta treba ponovo da se isrtava, a Windows to ne bi uradio sam. Jedan primer takve situacije je promena veličine prozora

## **14.Jednostavne slike I grafikoni, krive linije**

Jednostavni likovi mogu se nacrtati na ekranu displeja razmeštajući neke simbole u skladu sa zahtevanim konturama. Na slici 2.1 prikazan je jednostavan lik, obrazovan pojedinačnim, međusobno nepovezanim simbolima. Obično se takvi

likovi

lako

formiraju

u programu na jeziku

BASIC

pomoću naredbe

PRINT , i u nekim

slučajevima takve slike mogu biti zaista korisne.

Za crtanje linije mo se koristiti naredba PSET . Slika odsečka prave između dve tačke na ekranu dobija se osvetljavanjem niza tačaka du prave, koja spaja dva kraja odsečk a (du).

Sve tačke prave koju crtamo nezavisne su međusobnoč prema tome, da

bi dobili, na primer, sliku vertikalne prave pomoću naredbe

PSET,

potrebno je povećavati Y sa korakom 1 pri konstantnom X . Analogno se crtaju horizontalne prave. (Programi 2.1.6

i

2.1.7). Ovi programi crtaju neprekidne linije, jer je

svaki

piksel

mali pravougaonik, koji se priključuje već

osvetljenom

pikselu.

(Ovo va za vertikalnu liniju samo ako se radi u remu najviše

rezolucije, na primer u

SCREEN 9

za

EGA

karticu.)

Pored toga grafikoni omogućavaju bržu ocenu informacije, koja se sadrži u skupu brojeva. Oni daju mogućnost da se različiti odnosi među podacima, koje je često teško uočiti iz jednostavnog skupa odštampanih cifara. Grafikoni se mogu konstruisati ili pomoću naredbe

PRINT

ili korišćenjem

specijalnih grafičkih naredbi (opisanih u prethodnom odeljku).

Mada je u nekim slučajevima i podesno koristiti naredbu

PRINT

grafička sredstva omogućavaju stvaranje detaljnijih i informativnijih grafikona.

Krive linije se mogu uključiti u crteže i grafikone pomoću standardnih metoda. Kako su mogući načini naredbe

PRINT

za crtanje krivih linija ograničene, razmotrimo uglavnom pikselske (tačkaste) grafičke

metode.

Pikselska grafika omogućava tačnije predstavljanje krivih. Deo krive

linije

može se aproksimirati odsečcima pravih ili blizu raspoređenim tačkama. Što je veća rezolucija korišćenog sistema, to je tačnija aproksimacija. Pri povećanju broja odsečaka koji vrše aproksimaciju, a koji se sadrže između dve tačke, kriva postaje glatkija. (Program 2.3.1) . U isto vreme, što se više

odsečaka

koristi,

to

je više vremena potrebno sistemu da formira sliku.

Kružnica je najčešće korišćena kriva. Ona se primenjuje kao jedna od osnovnih komponenti pri kreaciji crteža i kružnih dijagram

a. Da  
bi konstruisali kružnicu na ekranu displeja,

potrebno

je dati njen položaj i veličinu (Sl. 2.9). Položaj kružnice

određuje se koordinatama centra

$(X_C, Y_C)$

kružnice. Veličina kružnice zadaje se njenim poluprečnikom

$R$  . Pored toga, potrebno je imati jednačinu za određivanje koordinata svih tačaka duž tražene kružnice. Jednačina kružnice može biti napisana u nekoliko različitih

oblika. Koordinate niza tačaka

$(X, Y)$  , koje pripadaju kružnici,

najzgodnije je odrediti u zavisnosti od ugla

$A$  , koji se meri u smeru kazaljke na satu od horizontalne linije (Sl. 2.10). Vredosti

X

## Transformacija slika

Metode transformacija, koje se razmatraju u ovom odeljku, jesu osnovni instrument manipulisanja elementima slika. Shvatanje principa promene i transformacije elemenata slike u znatnoj meri povećava naše mogućnosti

pri

stvaranju

efektivnih

programskih sredstava mašinske grafike. Specijalno, pomoću metoda

transformacije mogu se: premeštati crteži i grafikoni s jednog mesta

ekrana na drugo, stvarati složeniji crtež od manjih

elemenata,

dodavati postojećem crtežu novi elementi, povećavati dimenzije

crteža

ili grafikona preglednosti radi ili zbog prikazivanja

sitnijih

detalja, smanjivati dimenzije crteža (zbog

unošenja

objašnjenja,

na primer, komentara ili dopunskih crteža), a takođe stvarati pokretne slike. Sve promene pojedinih crteža i grafikona vrše se

pomoću

tri osnovne operacije:

1. premeštanja prikazivanih objekata s jednog mesta ekrana na drugo (translacija);
2. povećavanja ili smanjivanja dimenzija prikazanog crteža ili grafikona (skaliranje);
3. promene orijentacije crteža (rotacija).

### **15.Transformacija slika(TRANSLACIJA)**

Promena položaja ili translacija. Metodi promene položaja elemenata slike na ekranu omogućavaju kreiranje crteža ili grafikona u proizvoljnoj oblasti ekrana i zatim njegovo premeštanje

na drugu stranu ekrana. Pri tome je

moguće

ili

jednostavno

ponovo sastaviti crtež, ili ga sastaviti po

delovima

od

skupa

gotovih elemenata.

Premeštanje crteža iz jedne oblasti ekrana u drugu ekvivalentno je premeštanju svih tačaka crteža i kasnijem ponovnom iscrtavanju svih sastavnih linija. Da bismo sprečili deformaciju oblika predmeta, sve tačke crteža treba premestiti na jednako rastojanje.

## **16.Transformacija slika(SKALIRANJE)**

. Promena veličine ili skaliranje. Posle stvaranja lika na ekranu displeja može nastati potreba da se promeni njegova veličina, da bi predstavljenu informaciju učinili jasnijom ili umetnuli ovaj lik u krupniju sliku. Veličina crteža ili grafikona može se

izmeniti ako se pomnože sva rastojanja među tačkama određenim koefici-

jentom skaliranja povećavanja ili smanjivanja lika. Na primer, ako hoćemo da povećamo (smanjimo) veličinu pravougaonika dva puta moramo dužine svih linija pomnožiti sa  $2$  ( $1/2$ ).

Pri skaliranju lika treba takođe odrediti njegov položaj posle povećavanja ili smanjivanja dimenzija. Možemo skalirati predmete u odnosu na neku centralnu tačku predmeta, tačku na granici ili proizvoljnu tačku van njega. Na primer, mogu se povećati dimenzije pravougaonika, fiksirajući položaj levog donjeg ugla pravougaonika ili fiksirajući njegov centar.

## **17.Transformacija slika(ROTACIJA)**

Promena orijentacije ili rotacija. U mnogim slučajevima javlja se potreba da se promeni orijentacija lika, na primer, histogram tako da su kolone smeštene horizontalno, a ne

vertikalno. Rotacija lika za

90

jeste dobar metod za rešavanje tog zadatka, koji ne iziskuje promenu strukture grafika. U

programima za modeliranje i igre često je potrebno nacrtati okrenute

predmete. Slika 3.5. ilustruje rotaciju tačke. Premeštanje se ostvaruje iz tačke

$(X, Y)$

u

tačku

$(X_R, Y_R)$

duž luka kružnice

sa

centrom rotacije

$(X_O, Y_O)$  . Ugao

A

na toj slici određuje veličinu obrtanja iz tačke

$(X, Y)$

u tačku

$(X_R, Y_R)$  . Koordinate tačke

$(X_R, Y_R)$

posle transformacije

izračunavaju

se

korišćenjem

veličina

A,  $(XO, YO)$

i

$(X, Y)$

prema formulama (12) Centar rotacije

$(XO, YO)$

može se

nalaziti

na

proizvoljnom pogodnom mestu na ekranu ili van njegovih granica. Ova tačka se ne osvetljava i koristi se samo kao

potporna

tačka

za

određivanje premeštanja po kružnici pri rotaciji (sl. 3.6). Centar rotacije

$(XO, YO)$

može se

nalaziti

na

proizvoljnom pogodnom mestu na ekranu ili van njegovih granica. Ova tačka se ne osvetljava i koristi se samo kao

potporna

tačka

za

određivanje premeštanja po kružnici pri rotaciji. Ugao obrtanja

A

meri se u smeru suprotnom od

kretanja

kazaljke na časovniku od početnog položaja tačke

(X,Y) .

Pri

tome se pretpostavlja da se koordinatni početak nalazi u gornjem

levom uglu ekrana. (Kod grafičkih sistema gde se koordinatni početak nalazi u levom donjem uglu ekrana, ugao

A

treba da se meri u smeru kretanja kazaljke na satu). Obično vrednost ovog ugla leži u dijapazonu od

0

do

$2 \cdot \pi$

(6.283185 rad).

I

druge

vrednosti

ugla obrtanja su dopuštene, međutim takva obrtanja ekvivalentna su obrtanju u gore navedenom dijapazonu.

Put, koji prelazi tačka po luku kružnice pri obrtanju za ugao

A , zavisi od rastojanja između tačke

(X,Y)

i

centra

rotacije

(XO,YO) . Što je centar rotacije

(XO,YO)

dalje od tačke

(X,Y) , to je duža trajektorija (putanja) kretanja

od

tačke

(X,Y)

do tačke

(XR,YR) . Na slici 3.7 pokazana je zavisnost među radijusom rotacije

R , uglom

A

i rastojanjem

D

od

tačke

(X,Y)

do tačke

(XR,YR) . Za male uglove obrtanja veličina

D

približno je jednaka proizvodu

$R \cdot A$  . Ako je

$R = 50$  , tačka se premešta za

1

korak rastra (u susedni piksel) pri uglu obrtanja

$A = 0.02$  rad .

### **18.Perspektiva I projekcija**

Za konstruisanje perspektivnih slika može se nacrtati skica na kockastoj hartiji, analogna slici 4.10, tako da se sve paralelne linije sastaju u jednoj udaljenoj tački, nazvanoj tačka

sastajanja. Zatim se određuju koordinate

X

i

Y

svih

tačaka,

posle čega se crta slika. Na takav način može se

konstruisati

proizvoljan perspektivni lik prema prethodnoj skici, nacrtanoj na kockastoj hartiji. Međutim, bilo koja izmena na

crtežu

ili

prelaz

na drugi crtež zahteva ponavljanje

potpunog

ciklusa,

počinjući

od skice na kockastoj hartiji.

Drugi način dobijanja perspektivnih slika sastoji se u korišćenju jednačina transformacija slika u programu, što nam omogućava veću gipkost. U ovom slučaju može se stvoriti

perspektivna predstava ponovljenih crteža ili crteža koji se

kreću

bez

ručne analize rasporeda i

dimenzija

pojedinih

fragmenata.

Koordinate stubova (Sl. 4.10) mogu biti određene samo jednom, a

zatim

višestruko konstruisani s umanjivanjem dimenzija proporcionalno dubini pozicija.

Projekcija predmeta može se dobiti zamenjivanjem jedne od koordinata  $X$  ili  $Y$  koordinatom  $Z$ , pri konstruisanju

slike

na ekranu. Tako, na primer, crtajući na ekranu sve tačke sa

vrednostima koordinata

$(X,Z)$ , dobijamo izgled predmeta odozgo ili odozdo, a crtajući tačke sa vrednostima

$(Z,Y)$ ,

dobijamo

izgled

sa strane, kao što je pokazano na slici 4.4. U prvom slučaju smatramo da su vrednosti

$Y$

pozitivne u dubini ekrana (inače dobijamo pogled odozdo). U drugom slučaju imamo pogled sleva, ako smatramo

da su vrednosti

$X$

pozitivne u dubini ekrana, i pogled

zdesna,

ako su vrednosti

$X$

pozitivne ispred ekrana. Ovakve projekcija

nazivaju se ortografske. Da bi nacrtali izgled predmeta odozgo, odozdo ili sa strane, moramo

uvesti

koordinate

$(X,Y,Z)$

svih

temena predmeta i konstruisati crtež prema odgovarajućem paru koordinata, isključujući treću koordinatu.

## 19. Vidljivost linija – poligona

Postoje dva osnovna prilaza odstranjivanju nevidljivih delova prikazivanog predmeta. Prvi prilaz sastoji

se

u

predstavljanju predmeta u obliku ukupnosti površi koje ga čine.

Pri

tome

određujemo i odstranjujemo one površi, koje se nalaze u zadnjem

planu ili su skrivene drugim površima. Drugi metod se

sastoji

u

predstavljanju slike predmeta u obliku ukupnosti linija. U

ovom

slučaju nalazimo i brišemo pojedine nevidljive linije, a ne površi. Metod odstranjivanja nevidljivih površi sastoji se u

prekrivanju bojom svake površi na ekranu,

počinjući

od

najudaljenije.

Prve se boje površi, koje imaju maksimalnu vrednost  $Z$ , tako da bliže površi zaklanjaju dalje

površ. Svaka površ može biti obojena određenom bojom, ili se sve površi boje

bojom

pozadine,

radi brisanja nevidljivih linija.

Program 9.1 ilustruje ovaj metod odstranjivanja nevidljivih linija. Omogućava se prikazivanje proizvoljnog broja pravougaonih površi.

Pretpostavlja se da je svaka ravna pravougaona površ paralelna ekranu displeja zato se za svaku površ određuje samo

jedna vrednost koordinate

Z . Površ se boje bojom pozadine posle

razmeštanja u niz prema smanjivanju koordinate

Z . Prekrivanje površi dovodi do brisanja nevidljivih linija daljih površi. Korišćenje različitih boja za različite površi dovodi, ne do brisanja, već do ponovnog bojenja površi.

## **20.Likovi koji se kreću(animacija)**

Možemo se koristiti osnovnim operacijama transformacije likova (translacijom, skaliranjem i rotacijom) da bi "oživeo" crtež. Pri tome se lik najpre crta zatim briše, vrše se transformacije i ponovo se crta transformisani lik. Ako ovu proceduru ponovimo nekoliko puta, dobijamo lik koji se kreće. Analogno se stvaraju

crtani filmovi: umetnik crta niz kadrova filmske trake s malim izmenama položaja ili dimenzija predmeta, a brzo smenjivanje pojedinih crteža stvara gledaocu iluziju neprekidnog kretanja.

Programi, koji omogućuju kretanje tačke po ekranu displeja, jesu modifikacije u 3.1 razmatra njih prog rama,

koji

su

vršili translacije lika. Umesto jednostrukog premeštanja tačke

omogućava se njeno kretanje, pri tome se može primorati (naterati) tačka

da se kreće kako po pravoj liniji, tako i po krivolinijskom putu

Kretanje tačke napred i nazad po pravoj liniji u vertikalnom pravcu znači da uvodimo horizontalne granice s vertikalnim koordinatama

(Y1

i

Y2)

i premeštamo tačku pri

fiksiranoj

vrednosti koordinate X. Da bi omogućili kretanje

u

proizvoljnom

pravcu, treba uvesti priraštaj istovremeno po obema koordinatama

X

i

Y. Ovaj proces je sličan

iscrtavanju

prave

linije,

sem

što

sada brišemo prethodnu tačku pre osvetljavanja nove.

Putanja

kretanja tačke može se odrediti na razne

načine

## **21.Trodimenzionalni likovi**

. Realni objekti postoje u trodimenzionalnom prostoru, zato pored dužine i visine imaju i debljinu. Kad se slika objekta prikazuje na ravnom (dvodimenzionalnom) ekranu, može se ili

ignorirati njegova debljina, ili projektovati objekat na

ekran

tako

da

se osjeti dubina slike. Predstavljanje dubine može

suštinski

poboljšati opažanje crteža i povećati njihovu informativnost. Niže

ćemo razmotriti nekoliko metoda konstruisanja

trodimenzionalnih

slika objekata.

Trodimenzionalni objekat, na primer kutija, može se predstaviti na ekranu displeja u obliku pravougaonika (Sl. 4.1,a). Iz ove dvodimenzionalne slike može se izvući informacija o dužini i visini kutije, ali o dubini se može samo nagađati, jer je kutija

vidljiva samo s jedne strane. Preglednije predstavljanje predmeta pokazano je na slici 4.1,b.

Sada se mogu

videti

i

druge

stranice predmeta i imati tačnija predstava o strukturi i dimenzijama predmeta. Analogno konstruisanje grafikona i dijagrama u tri dimenzije omogućava da se na ekranu prikaže dopunska informacija.

U nekim slučajevima potreban je opštiji metod konstruisanja trodimenzionalnih slika, koji bi omogućavao grafičkim programima da razlikuju prednju i zadnju stranu predmeta. Pri rotaciji

predmeta u vidno polje neprestano ulaze njegove nove strane

i

dešava se stalna smena položaja prednje i zadnje strane. Da bi se

razlikovala prednja i zadnja strana trodimenzionalnog objekta i

dobile informacije o njegovoj dubini koriste se različite metode. Jedan metod se sastoji u odstranjivanju

nevidljivih

elemenata

slike.

Osim toga, može se konstruisati slika predmeta tako, da se dubina predstavlja pomoću perspektive: bliski predmeti i površi crtaju se krupnije, nego udaljeni. Drugi metod izdvaja bliskije linije tako što prednje delove objekta crta sjajnije (blistavije) u

poređenju sa zadnjim. Mogu se takođe koristiti složene metode šrafiranja radi stvaranja

na površima objekata postepenog prelaza

od

svetlog ka tamnom. U svim nabrojanim metodama potrebno je da dubina elemenata crteža bude poznata programu koji formira sliku

## **22.Trodimenzionalne transformacije**

U ovom odeljku ćemo proširiti ranije izučene metode dvodimenzionalnih transformacija (translacije, skaliranja, rotacije) na slike trodimenzionalnih objekata. Ove metode transformacija bitno šire mogućnosti manipulisanja dvodimenzionalnim i trodimenzionalnim likovima.

Crteži i grafikoni mogu se premeštati ili postepeno kretati u prostoru, ako su za sve tačke predmeta

izabrane

jedne

te

iste vrednosti

H, V

i

D . Mogu se izabrati razne vrednosti

H, V

i

D

za različite tačke lika, međutim u tom slučaju prvobitni

oblik predmeta biće izobličen.

Primer takvog premeštanja u trodimenzionalnom prostoru dat je u programu 4.2.1: paralelopiped se premešta i projektuje na ekran.

Program uvodi veličine translacije po svakoj od tri ose, izračunava novi položaj, a zatim koristi jednačine perspektive, razmatrane u odeljku 4.1, radi crtanja premeštenog

lika

predmeta.

Pri velikim vrednostima

D

smanjujemo dimenzije lika paralelopipeda i približavamo ga tački sastajanja.

. Uopštavanje jednačina dvodimenzionalnog skaliranja za trodimenzionalan prostor ostvaruje se uvećavanjem ili smanjivanjem dimenzija po d

Z

osi , a ne samo po

X

i

Y

osi. Kao i kod

dvodimenzionalnog skaliranja, izabiramo fiksiranu tačku i

izračunavamo ponovo koordinate svih tačaka lika u odnosu na tu izabranu

tačku,

### **23. Teorija boja, digitalna estetika**

Boja ima dva značenja i predstavlja:

Subjektivni čovekov osećaj koji se javlja prilikom posmatranja svetlosti, odnosno supstancu koja sadrži pigmente.

Boje se mogu opisati objektivnim i subjektivnim parametrima.

Objektivni parametri boje:

Talasna dužina,

Čistoća,

Svetlosni fluks.

Talasna dužina i čistoća određuju hromatičnost svetlosti, tj. osećaj boje.

Objektivnim parametrima odgovaraju subjektivni parametri boje:

·Ton boje ili stepen obojenosti (Hue) odgovara talasnoj dužini,

·Veličina zasićenja (Saturation) odgovara čistoći boje,

·Svetlina boje (Brightness) odgovara svtlosnom fluksu.

§Metod opisivanja boje subjektivnim faktorima obeležava se sa HSB. Veličina zasićenja predstavlja stepen odstupanja neke boje od sive boje iste svetline. Zasićenje je obrnuto srazmerno udelu sivog u boji. Potpuno zasićena je ona boja koja ne sadrži sivu, takve su spektralne boje, dok su potpuno nezasićene: bela, crna, sivi tonovi.

Osnovne boje su:

·Crvena,

·Žuta i

·Plava.

Bela boja = crvena + žuta + plava.

Komplementarne boje:

·Crvena – zelena (= žuta + plava),

·Plava – narandžasta (= žuta + crvena),

·Žuta – ljubičasta (= crvena + plava).

Prilikom kreiranja slika na računaru potrebno je voditi računa o estetici i o:

·Prostornom efektu boja,

·Harmoniji i slaganju boja,

·Kontrastu boja:

·Kontrast boje prema boji,

·Svetlo-tamni kontrast,

·Hladno-topli kontrast,

·Komplementarni kontrast, ...

Izražavanje boja

Postoje dva osnovna načina izražavanja boja:

·Aditivni metod i ·Suptraktivni metod.

## **24.Primene racunarske grafike**

Sistemi računarske grafike koriste se na različite načine u zavisnosti od oblasti primene. Usko specijalizovani sistemi mogu izvršavati samo strogo određene funkcije č na primer, sistem imitacije letenja za treniranje pilota. Međutim sis temi

opšte

namene omogućavaju nam da stvaramo različite crteže (skice) i planove.

Obično se sistem računarske grafike opšte namene sastoji od monitora (displeja), tastature, i grafičkih programa

naravno, centralne jedinice (računara u užem smislu). Veliki broj grafičkih sistema omogućava priključivanje, osim

monitora  
i

tastature,

i drugih uređaja za ulaz i izlaz informacija kao što

su:

svetlosna olovka (light pen), miš (mouse), crtač (plotter) i štampač

(printer).

Sistem automatskog projektovanja i automatizovana proizvodnja bile su jedna od prvih primena i do sada ostaju osnovne sfere korišćenja računarske grafike. Displej računске mašine omogućava automatizaciju pripreme crteža, arhitektonskih pla  
nova,

ekonomskih dijagrama a takođe i šeme

proizvodnih

procesa.

Davši

računskom grafičkom sistemu dimenzije elementa

mašine

i

koristeći

metode konstruisanja crteža u mašinskom

projektovanju,

mogu

se

dobiti konture ili slika tog elementa pod proizvoljnim uglom posmatranja.

Inženjeri-konstruktori u automobilskoj, vazduhoplovnoj i aerokosmičkoj industriji koriste met

ode

računarskog

projektovanja pri konstrukciji oblika površi proizvoda.

Metode mašinskog projektovanja primenjuju se takođe pri razrađivanju električnih i elektronskih šema.

Arhitekta takođe koriste pri projektovanju crteže građevina, ostvarene pomoću računarske grafike. Ovi crteži mogu biti ostvarni u različitim oblicima.

Tehniku računarske grafike koriste umetnici.