

KOMPJUTERSKA GRAFIKA

Prof.dr. Senad Rahimić

1. PRIMJENA I OSNOVNI KONCEPTI INŽINJERSKE GRAFIKE PUTEM RAČUNARA.....	7
□ Istoriski razvoj.....	7
□ Primjena.....	9
□ Osnovni koncepti grafičkog dizajniranja putem računara i podjela.....	10
➤ Osnovna koncepcija grafičkog protočnog sistema.....	11
• Geometrijski (vektorski) sistem.....	12
• Rasterski sistem.....	14
• Sistem prikaza.....	16
➤ Podjela.....	17
2. RAČUNARSKA GRAFIKA U INŽENJERSKIM PRIMJENAMA.....	19
□ Uvod.....	19
□ Oblasti primjene u inženjerstvu.....	19
3. GRAFIČKI SKLOPOVI I UREĐAJI.....	23
□ Sklopovska grafička oprema.....	24
➤ Grafički procesor.....	24
➤ Izlazni grafički uređaji.....	26
➤ Jedinice za prikaz.....	31
➤ Ulazni grafički uređaji.....	37
□ Programska grafička oprema.....	38

4. ORGANIZACIJA MIKRORAČUNARSKOG GRAFIČKOG SISTEMA	1
□ Gafička kartica.....	43
➤ Dijelovi grafičke kartice.....	49
➤ Tipovi grafičkih kartica.....	51
□ Grafičke stanice.....	53
5. GRAFIČKI SOFTVERSKI PAKETI I UREĐAJI	54
□ Opšte.....	54
□ Programska podrška projektovanju i dizajniranju.....	55
➤ Operativni sistem i grafički interface.....	57
➤ Programi za administraciju i vođenje poslova pri izvođenju projektovanja i dizajniranja.....	58
➤ Programi za projektovanje i 3D modeliranje.....	59
• CAD programi.....	60
• Programi za konceptualno modeliranje.....	65
• Modeliranje, renderiranje i animacija.....	67
➤ Programi za crtanje, obradu slika, grafički dizajn i stolno izdalaštvo...	71
• Programi za rastersku grafiku.....	73
• Programi za vektorsku grafiku.....	74
• Programi za stolno izdalaštvo.....	75
6. INTERAKTIVNA RAČUNARSKA GRAFIKA	76
□ Koncept interaktivne računarske grafike.....	76
□ Upravljanje interakcijom.....	79

7. GRAFIČKI TERMINAL I GRAFIČKI PROCESOR.....	80
□ Grafički terminal.....	80
□ Grafički procesor.....	80
8. RASTERSKI I VEKTORSKI GRAFIČKI SISTEMI.....	82
□ Grafički sistemi.....	82
➤ Vektorski grafički sistemi.....	82
➤ Rasterski grafički sistemi.....	89
• Arhitekture rasterskih prikaznih sistema.....	96
9. MATEMATSKE OSNOVE RAČUNARSKE GRAFIKE.....	102
□ Uvod.....	102
□ Grafičke transformacije.....	102
➤ Transformacija translacije.....	103
➤ Transformacija rotacije.....	104
➤ Transformacija promjene faktora proporcionalnosti.....	105
➤ Transformacija smicanja.....	106
➤ Složene transformacije.....	107
➤ Transformacije rotacije u 3D prostoru.....	108
□ Projekcije.....	111
➤ Ortogonalna projekcija.....	112
➤ Perspektivna projekcija.....	113

□ Matematičke osnove crtanja primitivnih oblika.....	114
➤ Rasterski prikaz ravnih linija.....	114
• Osnovni inkrementalni algoritam.....	115
• Digitalni diferencijalni analizator.....	116
• Algoritmi zasnovani na aritmetici cijelih brojeva.....	117
➤ Ispunjavanje primitivnih oblika.....	118
➤ Crtanje krivih linija.....	119
• Parimetarske krivulje trećeg reda.....	121
• Definicija odsječka krive linije $Q(t)$	123
• Načini crtanja parametarskih krivulja.....	127
□ Matematičke osnove crtanja površina.....	128
➤ Dvoparametarske površine trećeg reda.....	129
• Bezierove površine.....	130
• Načini crtanja dvoparametarskih površina trećeg reda.....	131
➤ Mreže mnogouglova.....	133
➤ Površine drugog reda.....	136
□ Crtanje trodimenzionalnih tijela.....	136
10. ANIMACIJA.....	138
□ Interpolacija.....	139
□ Animacijski jezici.....	141

11. BOJA U RAČUNARSKOJ GRAFICI.....	142
□ Akromatska svjetlost.....	143
➤ Polutonska aproksimacija.....	144
□ Kromatska svjetlost.....	145
□ Modeli boja u računarskoj grafici.....	150
➤ RGB model boje.....	151
➤ CMY model boje.....	152
➤ HSV model boje.....	153
➤ Primjena boje u grafici.....	154

1. PRIMJENA I OSNOVNI KONCEPTI GRAFIČKOG DIZAJNIRANJA PUTEM RAČUNARA

ISTORIJSKI RAZVOJ

- ❖ **1950. godine** na Massachusetts Institute of Technology (MIT) razvijen je prvi računar koji je za izlazni uređaj imao cijev sa katodnom zrakom (CRT - Cathode ray tube). Prikaz podataka na CRT pokazao je mnoge olakšice pri radu u raznim oblastima
 - Tako je *MIT Whirlwind* (vihor) Computer povezan sa CRT i pratilo je ponašanje vremena,a
 - protivvazdušna odbrana imala SAGE sistem koji je imao konzolu putem koje je CRT operator identificirao cilj svjetlosnim perom
- ❖ **Do sredine šezdesetih godina** prošlog stoljeća, sistemi koji su koristili računarom pomognuto projektovanje i proizvodnju (Computer Aided Design – CAD i Computer Aided Manufacturing - CAM) ušli su u primjenu npr. u auto industriji i grafika je imala važnu ulogu u ovim sistemima. *Većina uređaja za prikaz slike bili su uređaji sa vektorskim prikazom.*
- ❖ **1962. godine** Ivan Sutherland, kojeg neki nazivaju i ocem računarske grafike, razvio je prvi interaktivni grafički sistem Skatchpad, tj. postavlja hijerarhijski organizirane strukture podataka na osnovu standardnih komponenti, a postavlja i osnove interaktivnog rada (choice, point, tastatura i svjetlosno pero). Također je 1966. god. razvio i prvi sistem virtualne stvarnosti. Tih godina Doug Engelbart razvio je prvog miša
- ❖ Razvoj televizijske tehnologije **u ranim sedamdesetim godinama** uzrokovao je pojavu jeftinjih rasterskih displeja zasnovanih na poljima piksela. Otprilike u to doba , sistemi u boji sa tri zrake za osnovne boje (crvena, zelena i plava - RGB) postaju sve popularniji.
- ❖ **U ranim 1980-tim**, prvi personalni računari, sa ugrađenim mogućnostima rasterskog prikaza (Apple MAC, IBM PC) doveli su do širokog korištenja bitmap grafike i interaktivne grafike. Razvio se i grafički interface-i (GUI) koji omogućavaju novim korisnicima korištenje raznih vrsta računarskih programa.

- Display računara postaje elektronski radni sto („*desktop*“),
- Programi za upravljanje prozorima („*windows manager*“) koriste se za rad različitih programa.
- Direktno upravljanje objektima („*point and click*“) omogućilo je intuitivno upravljanje i oslobođilo je korisnika od zamornog i sporog upravljanja upisivanjem naredbi.
- Uparedo sa razvojem programske podrške, razvijaju se i novi ulazni uređaji koji olakšavaju navedene radnje (*miš, svjetlosna olovka...*).
- Ulazni uređaji koji omogućavaju unošenje trodimenzionalnih pa i višedimenzionalnih koordinata postaju uobičajeni u primjeni, a intenzivno se istražuju i mogućnosti audio komunikacije.
- Potreba za razvojem prenosivih programa koji ne zavise od specifičnosti pojedinog prikaznog uređaja dovela je do pisanja standarda za ovakvu vrstu grafičkih paketa.

Danas postoji i više standarda: *PHIGS (Programmer's Hierarchical Interactive Graphic System)*, *Open GL (Silicon Graphics)*, *X Windows System*, *Postscript (Adobe)*, *Direct 3D (Microsoft)*.

Ove grafičke biblioteke koriste se na gotovo svim nivoima i granama računarske grafike te na taj način oslobođaju programera od ponovnog izmišljanja kotača (tj. implementacije jednostavnih grafičkih funkcija) uz istovremeno rasterećenje rada centralne procesorske jedinice (CPU) s obzirom da se većina obrada može implementirati na čipu grafičke kartice.

PRIMJENA

Grafičko dizajniranje putem računara danas se koristi u različitim područjima ***privrede, administracije, edukacije, zabave i svakodnevnog kućnog života.***

Neki primjeri primjene grafičkog dizajniranja uključuju:

- **korisnički interface** (većina aplikacija na personalnim računarima i na radnim stanicama imaju grafički sistem prozora putem kojeg komuniciraju sa korisnicima. Primjeri takvih aplikacija uključuju obradu teksta, stolno izdalaštvo, proračunske tablice itd.);
- **interaktivno crtanje** (u poslovnim, naučnim i tehnološkim primjenama grafičko dizajniranje se koristi za prikazivanje funkcija, dijagrama, histograma i sličnih grafičkih prikaza sa svrhom jasnijeg sagledavanja složenih pojava i olakšavanja procesa odlučivanja);
- **uredska automatizacija i elektronsko izdalaštvo** (grafičko dizajniranje široko se koristi za izradu elektronskih i štampanih dokumenata);
- **projektovanje pomoću računara** (*Computer Aided Design – CAD* danas se standardno koristi za projektovanje sistema i komponenata u mašinstvu, elektrotehnici, elektronici, telekomunikacijama, računarstvu itd.);
- **trgovina** (grafičko dizajniranje se koristi za vizuelnu animaciju i elektronsku trgovinu);
- **upravljanje procesima** (podaci iz senzora dinamički se prikazuju u prikladnom grafičkom obliku);
- **geografski informacijski sistem** (grafičko dizajniranje se koristi za tačan prikaz geografsko rasprodjeljenih i rasprostranjenih sistema i mjernih podataka npr. u telekomunikacijama i telemetriji);
- **grafičko programiranje** (grafičko dizajniranje se koristi za automatizaciju procesa programiranja virtualnih sistema npr. u instrumentaciji);
- **medicina** (grafičko dizajniranje se koristi za dijagnosticiranje i planiranje operacija);
- **vojna industrija i istraživanje svemira** (grafičko dizajniranje se koristi za obuku i simuliranje situacija);
- **školstvo** (edukacija je nezamisliva bez korištenja grafičkog dizajniranja);
- **filmska industrija i reklame** (grafičko dizajniranje se koristi za stvaranje vizuelnih efekata);
- **igre** (grafičko dizajniranje se koristi za izgradnju prividnih svjetova);
- **meteorologija**
- **poslovanje** itd.

OSNOVNI KONCEPTI GRAFIČKOG DIZAJNIRANJA PUTEM RAČUNARA I PODJELA

Najjednostavnija definicija grafike je da je to prezentacija informacija pomoću *slika tj. boja i oblika* te shodno tome i **grafičko dizajniranje pomoću računara** je isto to, s tim što se za generisanje i prezentaciju slikovne informacije koristi računar.

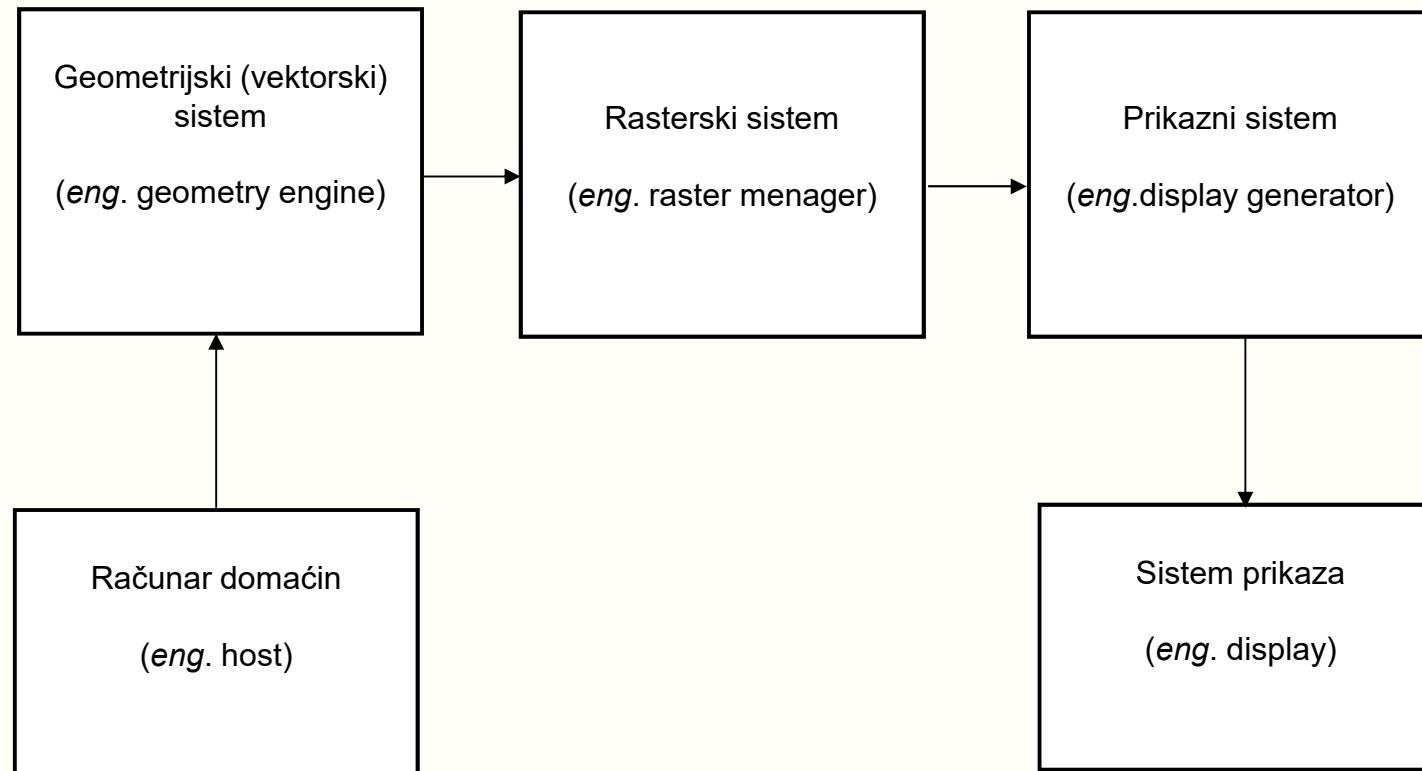
Grafičko dizajniranje putem računara možemo definisati i kao granu računarskih nauka koja se bavi izgradnjom višedimenzijskih modela objekata i njihovim prikazom uz upotrebu računara.

Grafičko dizajniranje putem računara nije potpuno originalna tema, jer za definisanje i rješavanje problema koristi neke već uspostavljenje tehnike kao što su **geometrija, algebra, optika i ljudska psihologija**.

- **Geometrija** se upotrebljava da osigura okvir (osnovu) za opisivanje dvo i trodimenzionalnog prostora,
- **Algebarska tehnika** se koristi za definisanje i evouliranje jednakosti vezanih za taj prostor.
- **Nauka o optici** omogućava modele za opisivanje ponašanja svjetla,
- **Ljudska psihologija** nudi modele za ljudsku viziju i percepciju boja.

OSNOVNA KONCEPCIJA GRAFIČKOG PROTOČNOG SISTEMA

Osnovni koncept zasniva se na tome da se logički nivo može preslikati u fizički nivo, odnosno oblik.

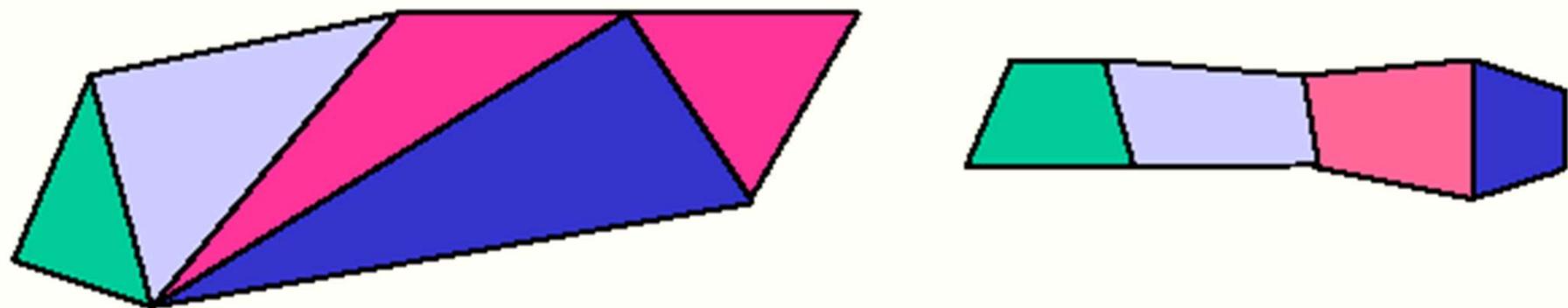


Slika 1- Koncept grafičkog sistema

Geometrijski (vektorski) sistem

Geometrijski sistem treba da obezbjedi slijedeće funkcije:

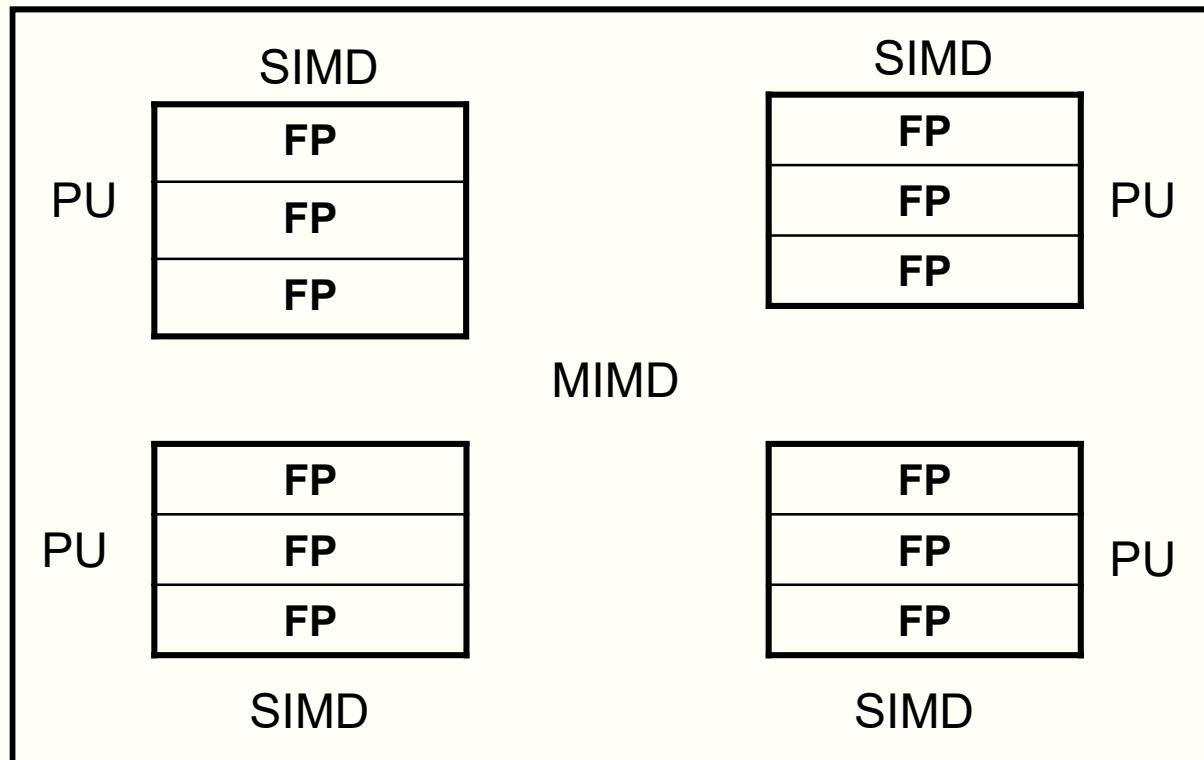
- geometrijske transformacije
 - ◆ translacija
 - ◆ rotacija
 - ◆ promjena mjerila (skaliranje)
- odsjecanje (eng. clipping)
- proračun osvjetljenja u trodimenzijskom prostoru i
- projekcije



Slika 2- Izgled geometrijskog oblika (pričazne liste)

Poželjna svojstva su mu:

- ◆ vezivanje s programskim standardima (pristup sklopovima kroz *OpenGL API-Application Program Interface*) parser (*analizator*) *OpenGL* naredbi
- ◆ ubrzano procesiranje prikaznih lista (*eng. polygon strips*)
- ◆ u interface-u prema računaru domaćinu poželjno je ostvariti *FIFO* (*first-in, first-out*) registre kako rad ne bi bio usporavan
- ◆ moguće ostvarivanje paralelizama npr.



PU – procesna jedinica
(*eng. geometry engine*)

FP – jedinica s
pomičnom tačkom za
obrađivanje
pojedinačne naredbe

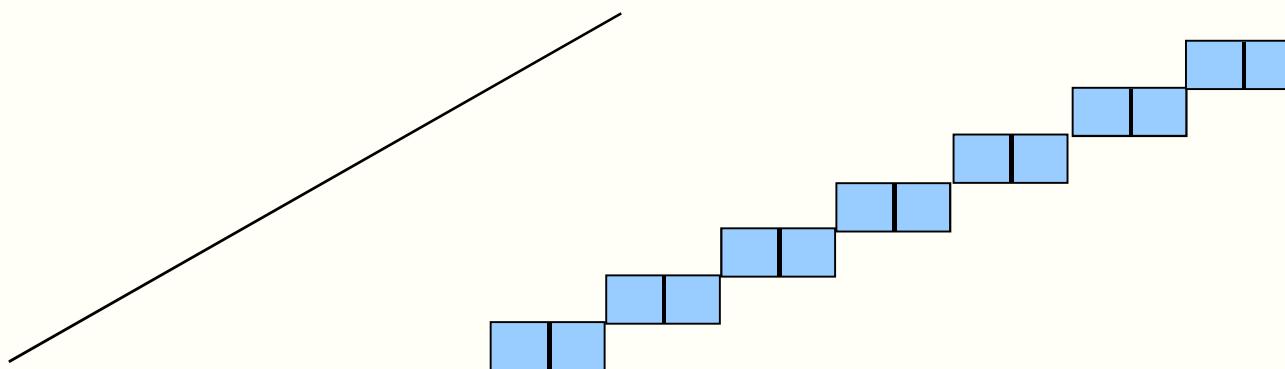
SIMD – jedna naredba,
više podataka (*eng.*
Single Instruction,
Multiple Data-SIMD)

Slika 3- Onyx2 Reality (*infinite Reality* tj. beskonačna stvarnost)

Rasterski sistem

Rasterski sistem treba da obezbjedi slijedeće funkcije:

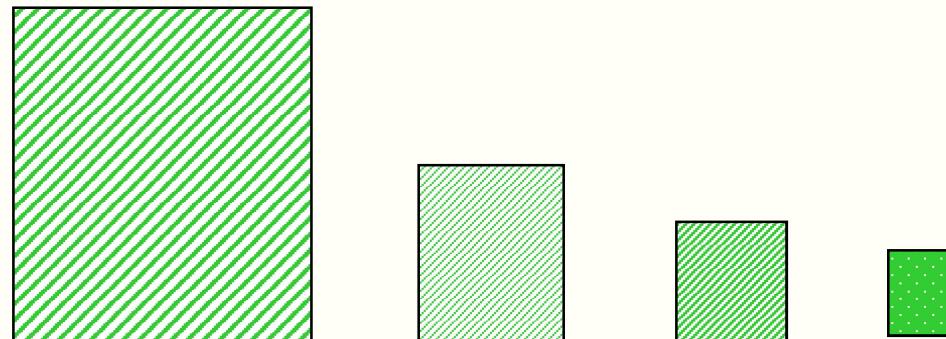
- ◆ pretvaranje iz kontinuiranih koordinata (nakon perspektivne projekcije uz 2D koordinatu čuva se i podatak o udaljenosti od početka) u diskretne koordinate u prikaznoj memoriji (*eng. scan-convert*)
- ◆ ispitivanje z-koordinate u Z-registru (uklanjanje skrivenih površina)
- ◆ preslikavanje teksture (na poligone)
- ◆ uklanjanje neželjenih učinaka uslijed diskretizacije (*eng. anti aliasing*)
- ◆ mješanje boja s različitim prozirnostima (*eng. blending*)
- ◆ omogućavanje prikaza atmosferskih učinaka (magla, dim))



Slika 4- Izgled linije geometrijskom i rasterskom sistemu

Poželjna svojstva su mu:

- ◆ Ostvarivanje funkcije u prikaznoj memoriji (eng. *frame buffer* 64MB-12GB). Ostvarivanje diskretizacije (eng. *anti-aliasing*) povećanim uzorkovanjem (eng. *multisampling*). Prikaz se ostvaruje u većoj efektivnoj jasnoći npr. 8 puta, od jasnoće konačnog prikaza.
- ◆ Upotreba memorije kombinovanjem među procesorima. Susjedni slikovni elementi se dohvataju u različitim procesnim jedinicama, tako da obrađivanju jednog poligona sudjeluje više procesnih jedinica.
- ◆ 10-bitni *RGB*, 12-bitni *RGBA* ($68 \cdot 109$ boja) (min. 128 bita-pixa)
- ◆ posebna jedinica za procesiranje teksture (16MB memorije- teksture)
 - perspektivna korekcija
 - MIP* preslikavanje (eng. *MIP mapping*), određivanje adreseodgovarajuće teksture (eng. *texel*)



Slika 5- Prikaz nekoliko izvedbi teksture

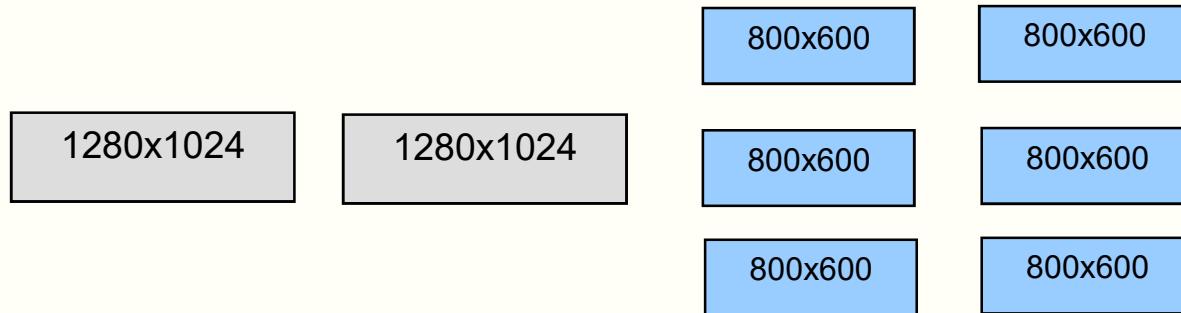
Sistem prikaza

Obezbjeđuje slijedeće funkcije:

- ◆ prijenos prikazane slike iz prikazne memorije *FB* (eng. *frame buffer*), preko *DAC* na prikaznu jedinicu
 - različite jasnoće i frekvencije osvježavanja
 - kombinovanje linije prikaza (eng. *interlace*)
- ◆ pretvaranje u analogni ili digitalni signal potreban za prikaznu jedinicu

Poželjna svojstva su mu:

- ◆ više kanala prikaza (2,4,6 ..) zbog mogućnosti izbora kanala prikaza (eng. *display channels*)
- ◆ različiti ulazno-izlazni video zapisi (*VHS, PAL, NTSC, HDTV 1920x1080*)

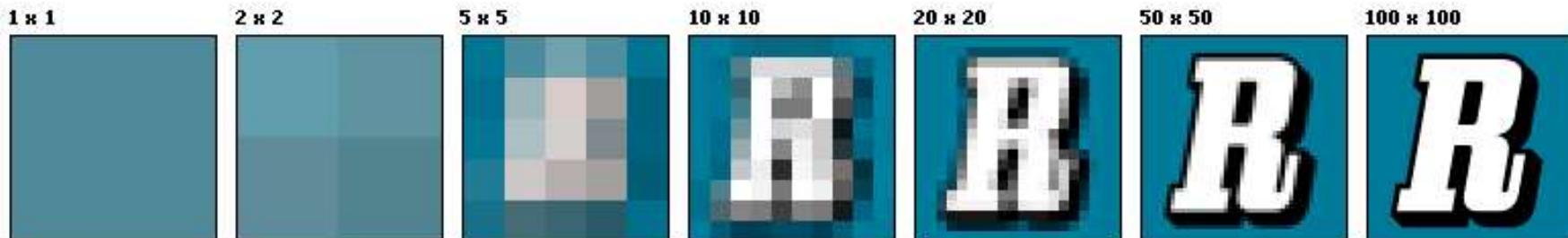


Slika 6- Prikaz mogućnosti višekanalnog prikaza

Rezolucija slike opisuje količinu detalja koju neka [slika](#) sadrži. Termin se koristi za [rasterske digitalne slike](#), filmske slike i ostale vrste slika. Veća rezolucija znači veći broj detalja koji se mogu vidjeti na slici.

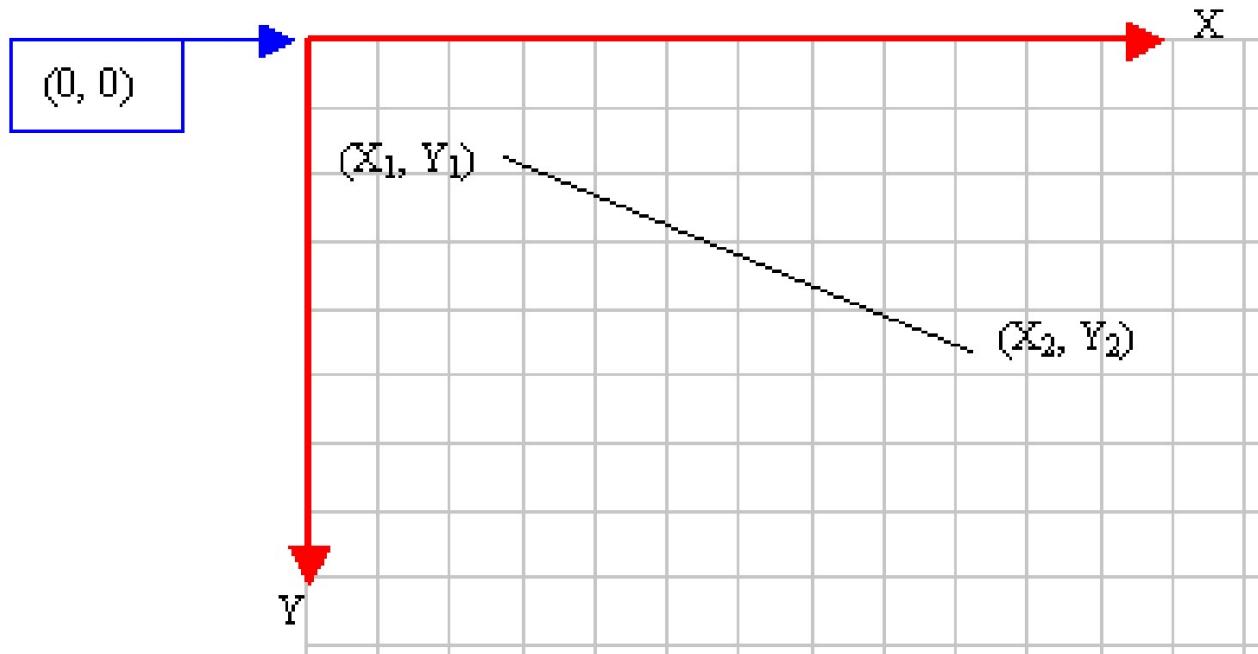
Najčešće uobičajene jedinice za opisivanje kvaliteta slike uključuju :

- PPI (pikseli po inču),
- DPI (tačke po inču),
- LPI (linije po inču) i
- SPI (uzorci po inču).



Basic shapes

- Lines
- Rectangles
- Ellipses
- Arcs
- Pies
- Polygons
- Cardinal Splines
- Bézier Splines
- Paths
- Open and Closed Curves
- Regions
- Clipping



```
public void DrawLine(Pen pen, Point pt1, Point pt2 );
```

Pen

Type: [System.Drawing.Pen](#)

[Pen](#) that determines the color, width, and style of the line.

Pt1

Type: [System.Drawing.Point](#)

[Point](#) structure that represents the first point to connect.

Pt2

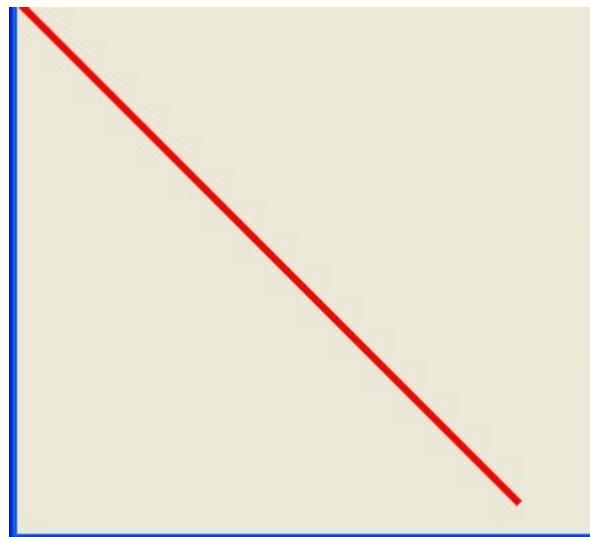
Type: [System.Drawing.Point](#)

[Point](#) structure that represents the second point to connect.

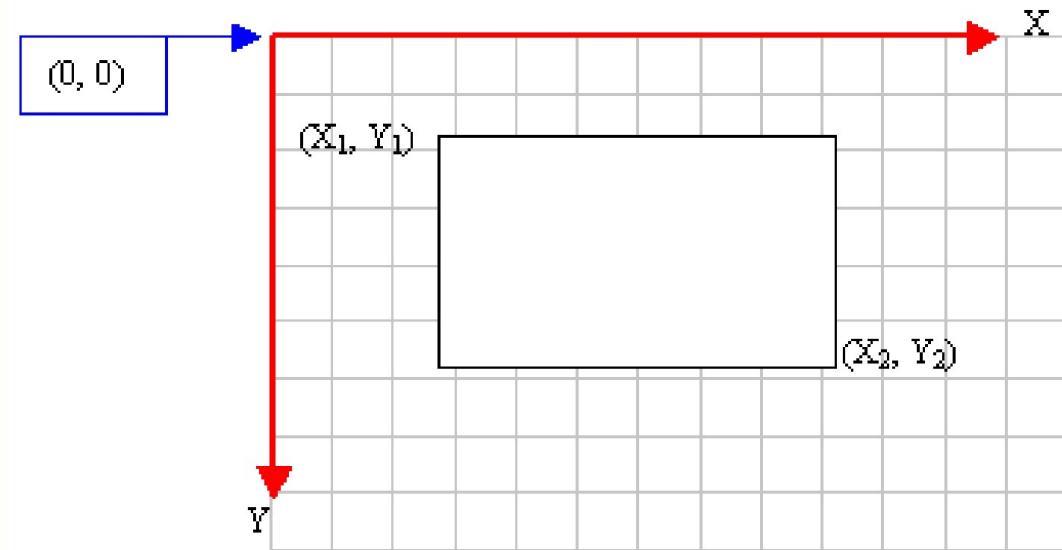
Lines

```
g.DrawLine(new Pen(Color.Red, 3.2f), 0, 0, 250, 250);
```

```
Point PocetnaTacka = new Point(0,0);
Point KrajnjaTacka = new Point(250,50);
Pen Olovka = new Pen(Color.GreenYellow, 3.2f);
g.DrawLine(Olovka, PocetnaTacka, KrajnjaTacka);
```



Rectangles



```
public void DrawRectangle( Pen pen, Rectangle rect )
```

Pen

Type: [System.Drawing.Pen](#)

A [Pen](#) that determines the color, width, and style of the rectangle.

Rect

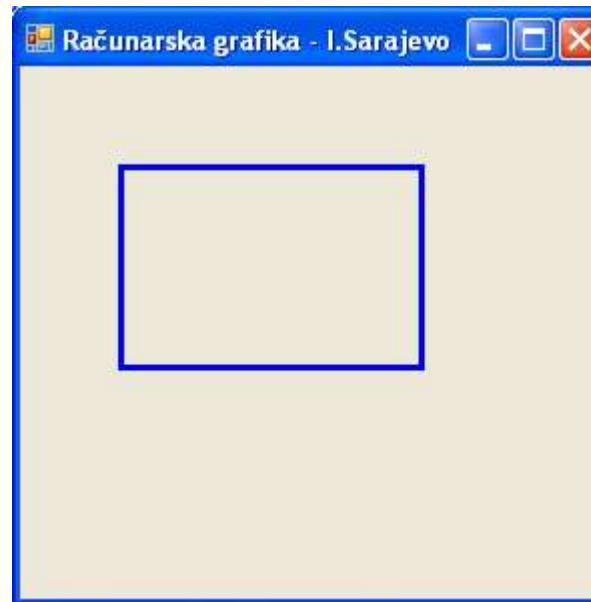
Type: [System.Drawing.Rectangle](#)

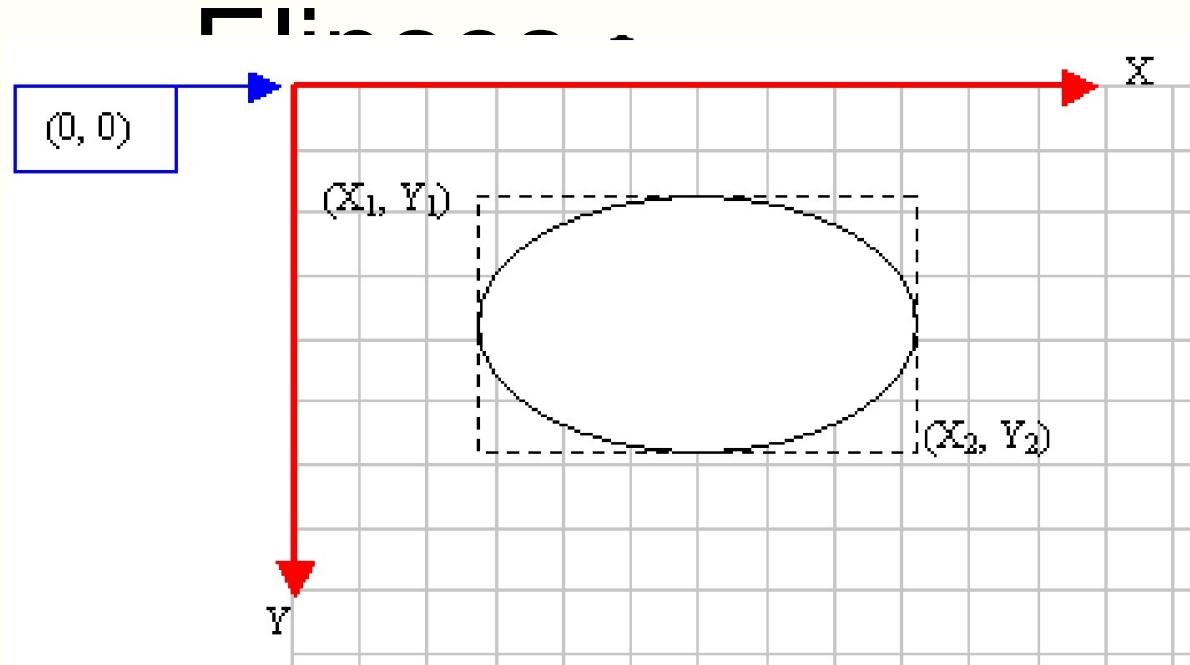
A [Rectangle](#) structure that represents the rectangle to draw.

Rectangles

```
g.DrawRectangle(new Pen(Color.Blue, 2.5f), 50, 50, 150, 100);
```

```
g.DrawRectangle(new Pen(Color.Blue, 2.5f),  
    new Rectangle(50, 50, 100, 50));
```





```
public void DrawEllipse( Pen pen, Rectangle rect )
```

Pen

Type: [System.Drawing.Pen](#)

A [Pen](#) that determines the color, width, and style of the rectangle.

Rect

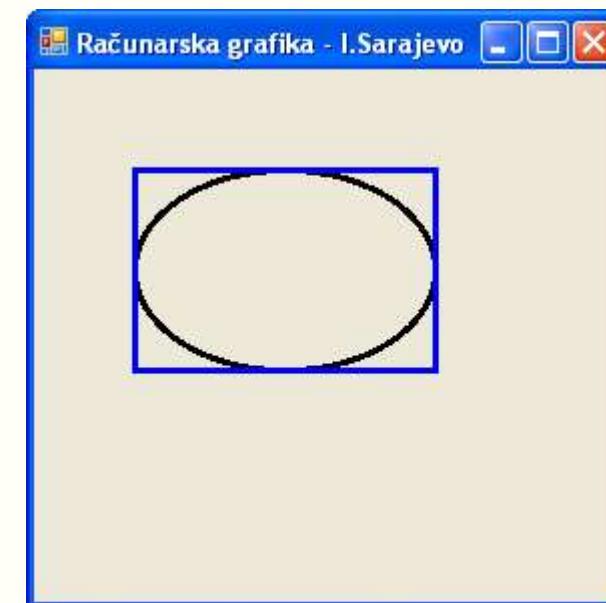
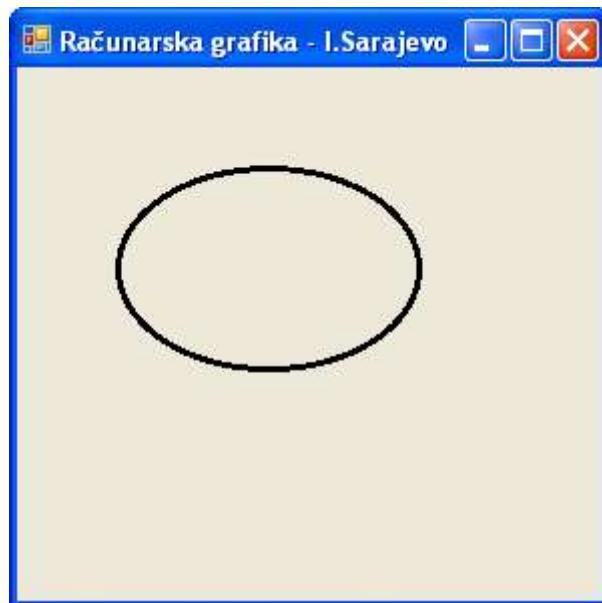
Type: [System.Drawing.Rectangle](#)

A [Rectangle](#) structure that represents the rectangle to draw.

Ellipses

```
g.DrawEllipse(new Pen(Color.Black, 3), 50, 50, 150, 100);
```

```
g.DrawEllipse(new Pen(Color.Blue, 2.5f), new  
Rectangle(50, 50, 100, 50));
```



Arcs

```
g.DrawArc(new Pen(Color.Black, 3), 50, 50, 150, 100, 0,  
140);
```



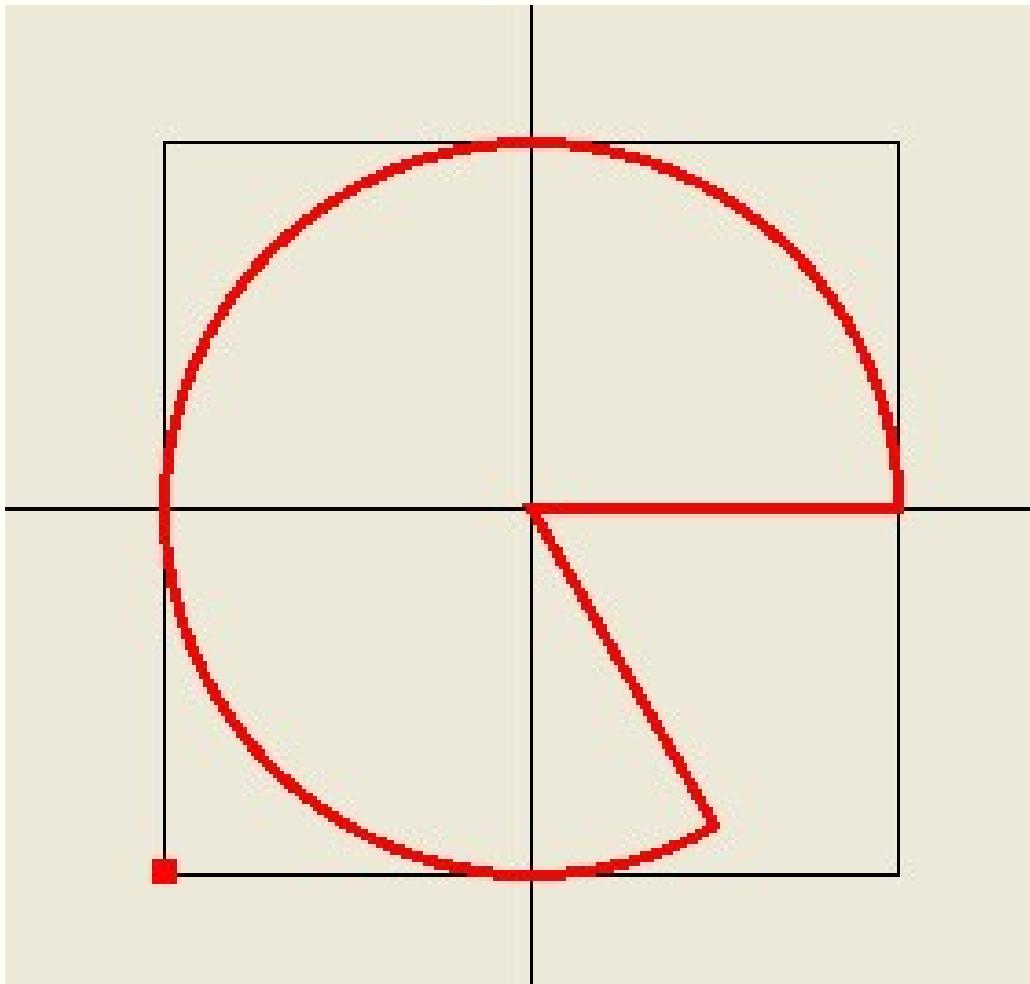
Pies

```
Graphics g = e.Graphics;
Matrix matricaTransformacija = new Matrix(1, 0, 0, -1, 0, 0);
matricaTransformacija.Translate(this.ClientRectangle.Width / 2,
    -this.ClientRectangle.Height / 2);
g.Transform = matricaTransformacija;

g.DrawLine(new Pen(Color.Black), -this.ClientRectangle.Width /
    2, 0, this.ClientRectangle.Width / 2, 0);
g.DrawLine(new Pen(Color.Black), 0,
    this.ClientRectangle.Height / 2, 0, -
    this.ClientRectangle.Height / 2);

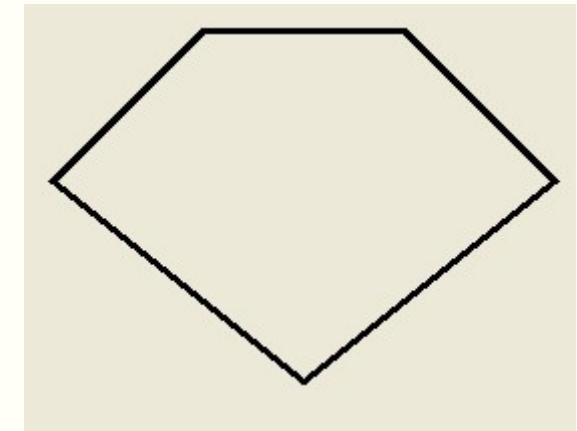
Pen Olovka = new Pen(Color.FromName("red"), 3);
g.DrawPie(Olovka, -100, -100, 200, 200, 0, 300);
```

Pies

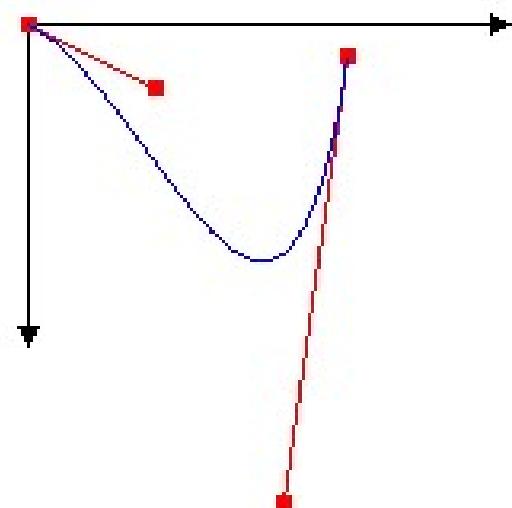
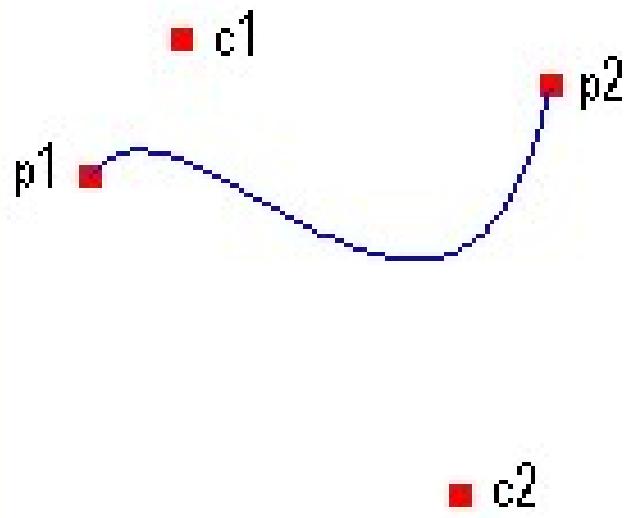
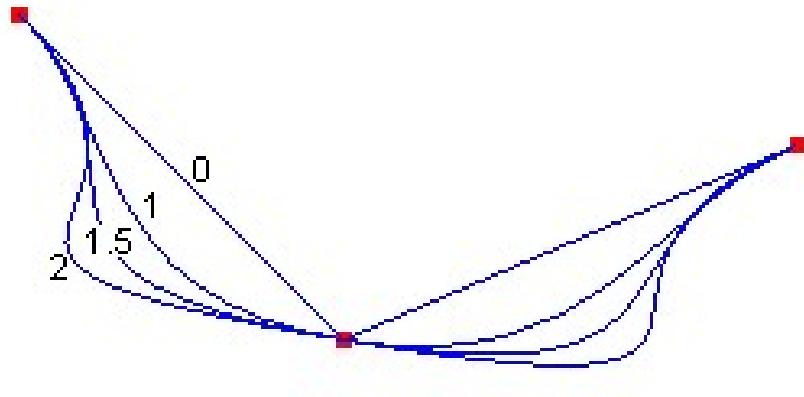
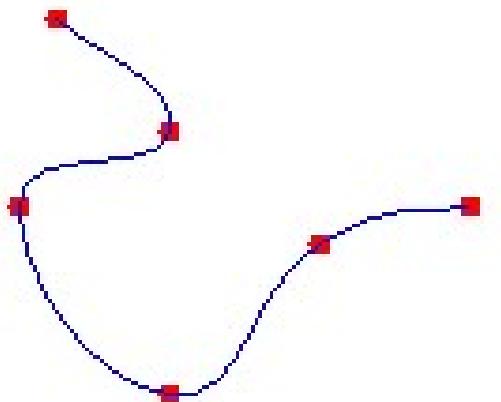


Polygons

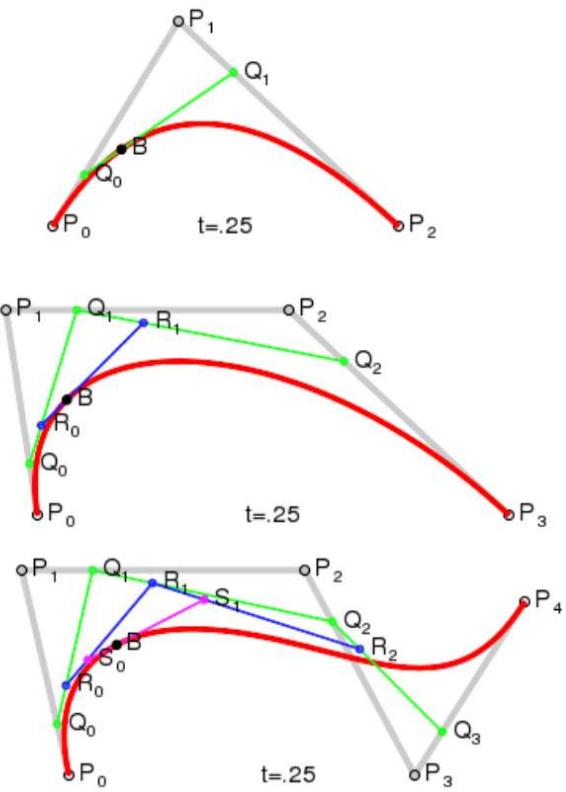
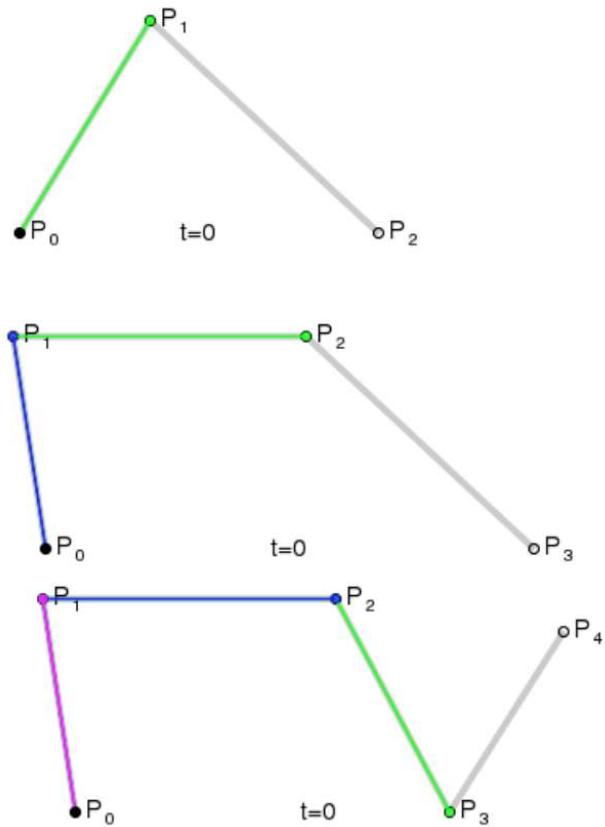
```
Point [] tacke = new Point[] {  
    new Point(100, 25),  
    new Point(25, 100),  
    new Point(150, 200),  
    new Point(275, 100),  
    new Point(200, 25)};  
Pen Olovka = new Pen(Color.Black, 3);  
g.DrawPolygon(Olovka, tacke);
```



Cardinal & Bézier Splines



Bézier Splines

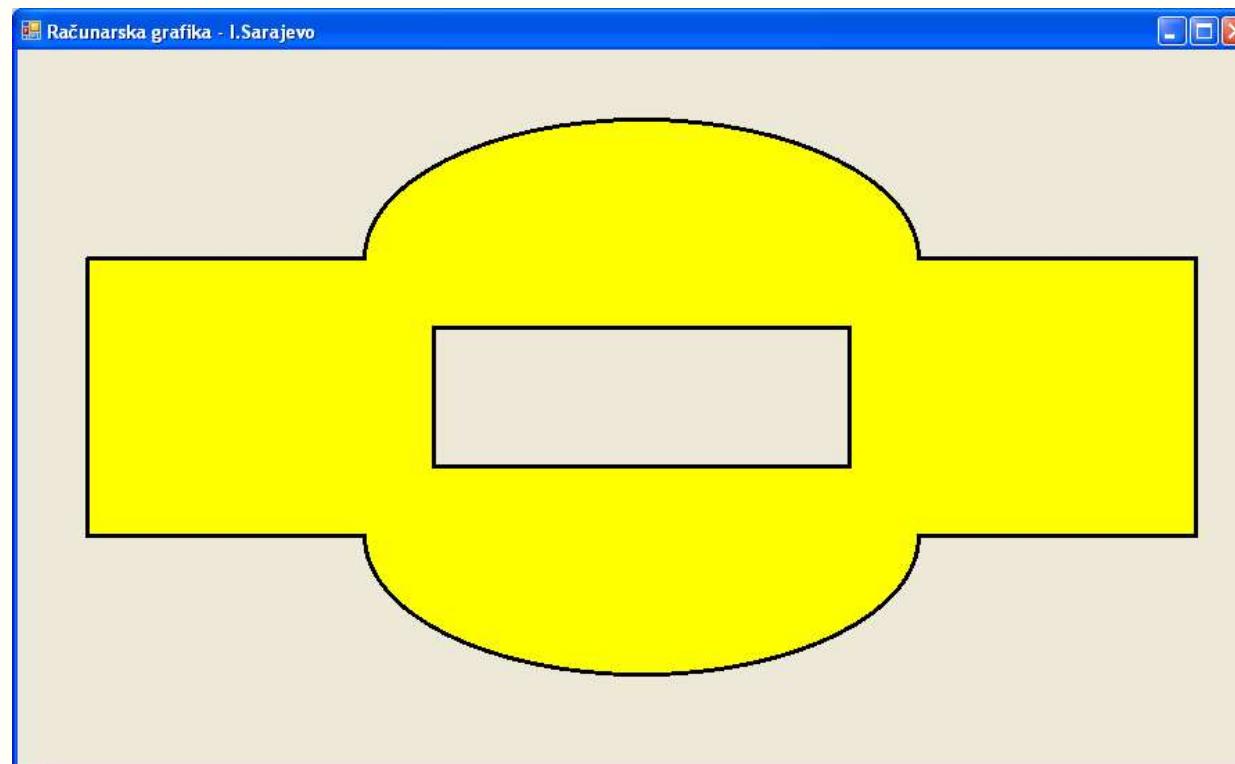


Paths

```
GraphicsPath gp = new GraphicsPath(FillMode.Alternate);
//gp.AddPolygon(tacke);
gp.AddLine(50, 150, 50, 350);
gp.AddLine(50, 350, 250, 350);
gp.AddArc(250, 250, 400, 200, 180, -180);
gp.AddLine(650, 350, 850, 350);
gp.AddLine(850, 350, 850, 150);
gp.AddLine(850, 150, 650, 150);
gp.Ad,dArc(250, 50, 400, 200, 0, -180);
//gp.CloseFigure();
gp.AddLine(250, 150, 50, 150);
gp.AddRectangle(new Rectangle(300, 200, 300, 100));
```

Paths

```
g.FillPath(new SolidBrush(Color.Yellow), gp);  
g.DrawPath(Olovka, gp);
```



Brushes:

- To fill a closed shape, you need a **Graphics** object and a **Brush** object.
 - Solid Brushes
 - Hatch Brushes
 - Texture Brushes
 - Gradient Brushes

Solid Brushes

```
public SolidBrush( Color color )
```

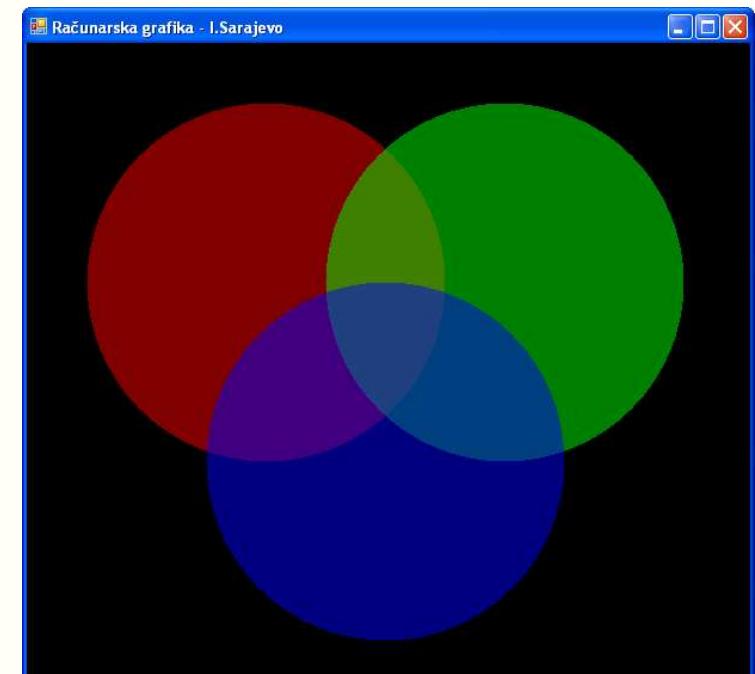
color

Type: [System.Drawing.Color](#)

A [Color](#) structure that represents the color of this brush.

Solid Brushes

```
g.Clear(Color.Black);
SolidBrush CetkaR = new SolidBrush(Color.FromArgb(128, 255, 0, 0));
SolidBrush CetkaG = new SolidBrush(Color.FromArgb(128, 0, 255, 0));
SolidBrush CetkaB = new SolidBrush(Color.FromArgb(128, 0, 0, 255));
g.FillEllipse(CetkaR, 50, 50, 300, 300);
g.FillEllipse(CetkaG, 250, 50, 300, 300);
g.FillEllipse(CetkaB, 150, 200, 300, 300);
```



Hatch Brushes

```
public HatchBrush( HatchStyle hatchstyle, Color foreColor, Color  
    backColor )
```

hatchstyle

Type: [System.Drawing.Drawing2D.HatchStyle](#)

One of the [HatchStyle](#) values that represents the pattern drawn by this [HatchBrush](#).

foreColor

Type: [System.Drawing.Color](#)

The [Color](#) structure that represents the color of lines drawn by this [HatchBrush](#).

backColor

Type: [System.Drawing.Color](#)

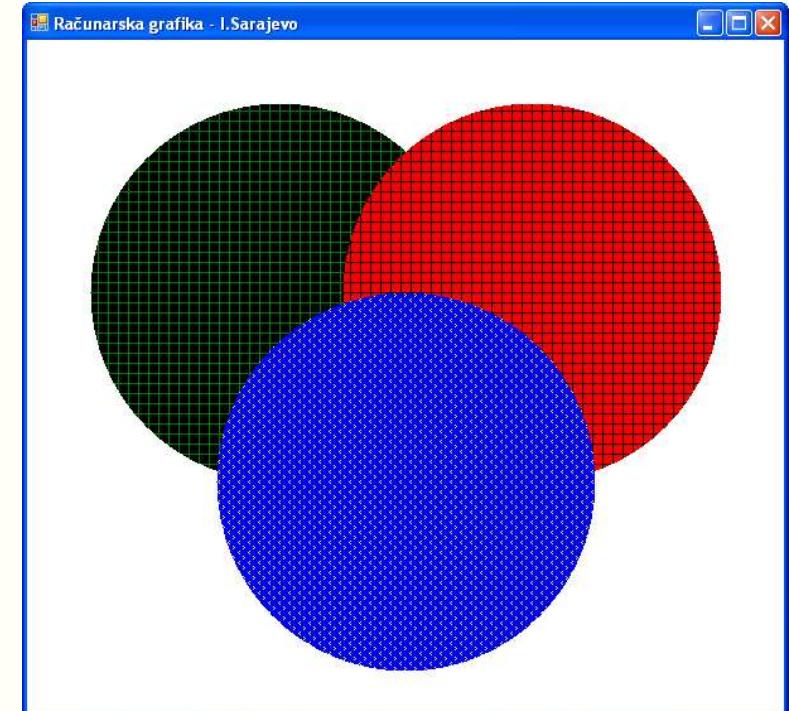
The [Color](#) structure that represents the color of spaces between the lines drawn by this [HatchBrush](#).

=====	Horizontal
=====	Min
	Vertical
/\	ForwardDiagonal
\/	BackwardDiagonal
	Cross
	LargeGrid
	Max
	DiagonalCross
.....	Percent05
.......	Percent10
.....	Percent20
.....	Percent25
.....	Percent30
.....	Percent40
.....	Percent50
.....	Percent60
.....	Percent70
.....	Percent75
.....	Percent80
.....	Percent90
/\	LightDownwardDiagonal
\/	LightUpwardDiagonal
/\	DarkDownwardDiagonal
\/	DarkUpwardDiagonal
/\	WideDownwardDiagonal
\/	WideUpwardDiagonal
	LightVertical
=====	LightHorizontal
	NarrowVertical
	NarrowHorizontal
	DarkVertical
=====	DarkHorizontal
.....	DashedDownwardDiagonal
.....	DashedUpwardDiagonal
.....	DashedHorizontal
.....	DashedVertical
.....	SmallConfetti
.....	LargeConfetti
.....	ZigZag
.....	Wave
/\	DiagonalBrick
====	HorizontalBrick
.....	Weave
.....	Plaid
.....	Divot
	DottedGrid
.....	DottedDiamond
.....	Shingle
.....	Trellis
.....	Sphere
	SmallGrid
.....	SmallCheckerBoard
.....	LargeCheckerBoard
.....	OutlinedDiamond
.....	SolidDiamond

Hatch Brushes

```
g.Clear(Color.White);
HatchBrush Cetka = new HatchBrush(HatchStyle.Cross, Color.Green);
HatchBrush Cetka1 = new HatchBrush(HatchStyle.Cross, Color.Black, Color.Red);
HatchBrush Cetka2 = new HatchBrush(HatchStyle.Divot, Color.Yellow, Color.Blue);

g.FillEllipse(Cetka, 50, 50, 300, 300);
g.FillEllipse(Cetka1, 250, 50, 300, 300);
g.FillEllipse(Cetka2, 150, 200, 300, 300);
```



Texture Brushes

```
g.Clear(Color.White);
Image slika = Image.FromFile("C:\\Documents and
    Settings\\Ognjen\\Desktop\\RG\\slike\\texture__.jpg");
Image slika1 = Image.FromFile("C:\\Documents and
    Settings\\Ognjen\\Desktop\\RG\\slike\\Y-YellowTexture.bmp");
Image slika2 = Image.FromFile("newwall.bmp");
```

```
TextureBrush Cetka = new TextureBrush(slika);
TextureBrush Cetka1 = new TextureBrush(slika1);
TextureBrush Cetka2 = new TextureBrush(slika2);
```



PODJELA

Više je načina da se podjeli ono što nazivamo "Grafičkim dizajniranjem putem računara".

- a) Prva osnovna podjela je na **interaktivno** i **neinteraktivno** grafičko dizajniranje.

◆ **Interaktivno** podrazumijeva dinamičan način prikazivanja slike na mediju (npr. monitoru) i, preko odgovarajućeg interfejsa, aktivno učešće čovjeka (dizajnera) u stvaranju i izmjeni slike, pri čemu su rezultati odmah vidljivi.

◆ **Neinteraktivnom** se smatra svako generisanje ili prezentiranje slikovnih informacija koje nezadovoljava prethodne uslove.

- b) Mnogo raširenija i češće upotrebljavana podjela grafičkog dizajniranja je podjela na **vektorsko** i **rastersko** grafičko dizajniranje. Ova podjela je izvršena prema osnovnim sastavnim (gradivim) elementima slike.

◆ **Vektorskog dizajniranja**

➤ Sastavni (gradivni) elementi su objekti (prave i krive linije, otvoreni i zatvoreni, ispunjeni i neispunjene geometrijski oblici) koji mogu da se preklapaju, prekrivaju ili uklapaju i tako tvore sliku.

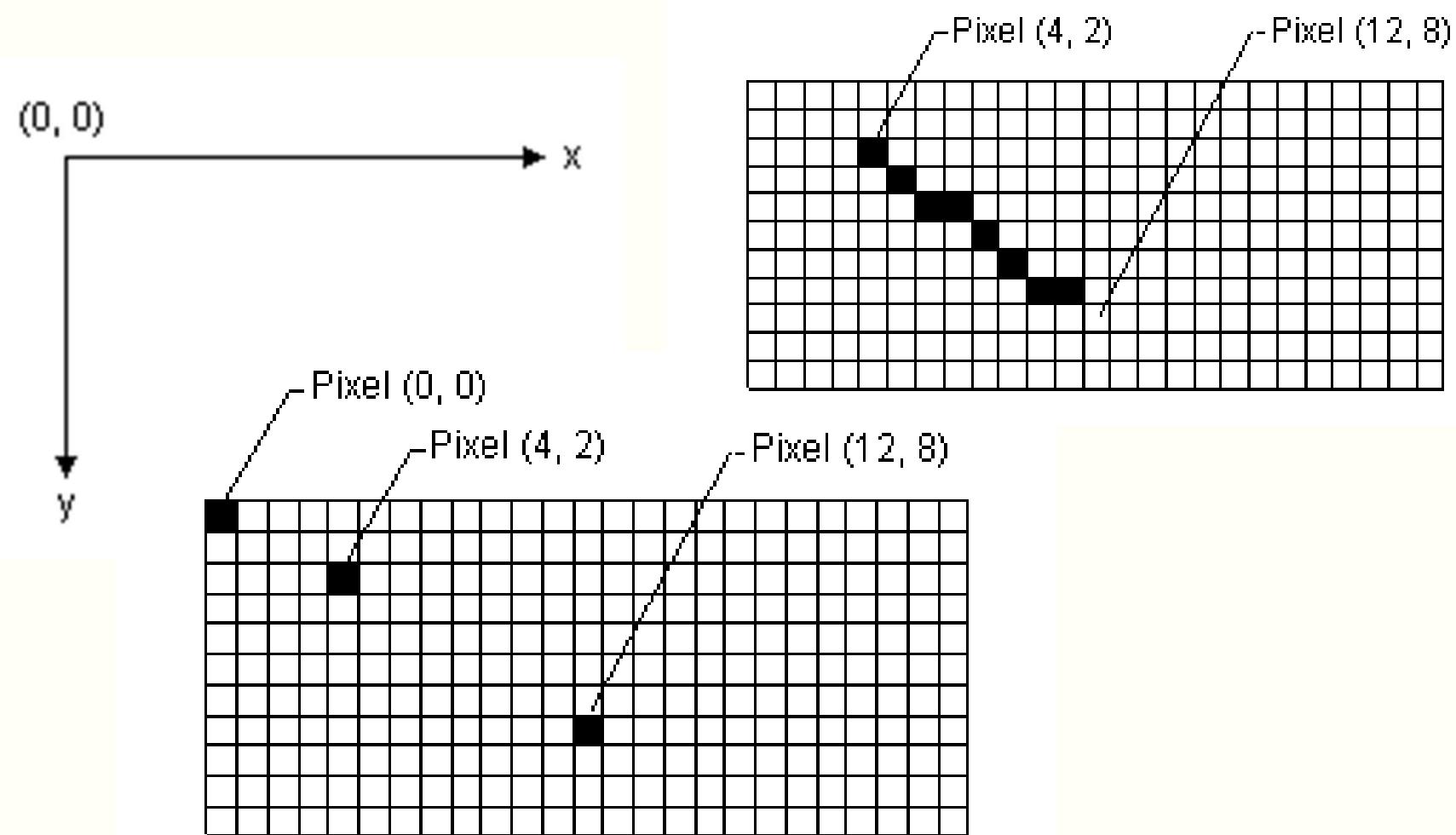
➤ Ovakve slike je lakše stvarati, mijenjati i kombinovati sa drugim slikama. Najadekvatnije ih je porediti sa kolažima od komadića raznobojnog papira.

➤ Računarska prezentacija ovakvih slika je niz matematskih vektorskih formula koje opisuju način i redoslijed iscrtavanja objekta. Zbog toga se ova vrsta grafičkog dizajniranja i zove **vektorsko**.

➤ Vektorsko grafičko dizajniranje se često naziva i **objektna grafika**.

➤ Vektorsko dizajniranje svoju primjenu nalazi u CAD programima i ono nema alternative ako želimo pomoći računara simulirati trodimenzionalni svijet.

➤ Za prikaz prizora koji se sastoji od jako mnogo detalja koji ne stoje u matematski opisivoj vezi (fotografije) vektorsko dizajniranje je potpuno nepodesno.



◆ **Rastersko dizajniranja**

- Kao osnovni sastavni (gradivni) element slike koristi tzv. **pixel** (naziv nastao od dvije engleske riječi: Picture i Element).
 - Pixel je najmanji dio slike koji ima jedinstvene vrijednosti boje i/ili intenziteta osvjetljenosti.
 - Kod nas se izraz pixsel često prevodi kao tačka.
 - Jednostavnije rečeno rasterska slika je slika sastavljena od tačaka različitog nivoa osvjetljenosti (monohromatske slike) ili različitih boja (kolor slike)
- c) Mnogo spominjana podjela je i podjela na **dvodimenzionalno (2D)** i **trodimenzionalno (3D)** grafičko dizajniranje. Međutim uz pojmove 2D i 3D vezuju se i neke zablude i zabune. Naime,
- ◆ U **2D** dizajniranju moguće je pomoći osvjetljenja i sijenki ili pomoći boja i oblika dočarati trodimenzionalni svijet. *Fotografije su, na primjer, dvodimenzionalne slike, ali vrlo vjerno prikazuju trodimenzionalni svijet.* U nekim programima za rad sa tekstrom postoji mogućnost prikazivanja ispuštenih slova. To ipak nije trodimenzionalno dizajniranje jer bilježi trodimenzionalni svijet u jednom trenutku tj. u jednom položaju. Na takvoj slici ne možemo pogledati neki objekat iz drugog ugla.
 - ◆ **3D** dizajniranje podrazumijeva da se "slike" sastoje od objekata u virtuelnom prostoru u memoriji računara. Slika koju ćemo vidjeti zavisi od prostornih odnosa između ovih objekata i od ugla posmatranja. U **3D** dizajniranju se zato ne govori o slikama nego o svjetovima. Naravno, svi danas široko rasprostranjeni uređaji za prezentaciju grafičkog dizajna stvorenog računarom su dvodimenzionalni što znači da se i **3D** svjetovi moraju prilagoditi **2D** prikazu.

2. RAČUNARSKA GRAFIKA U INŽENJERSKIM PRIMJENAMA

UVOD

- ◆ Računarska grafika se već razvila u tako moćno sredstvo da se njome koriste stručnjaci pri rješavanju vrlo složenih problema.
- ◆ Uz promoć program za računarsku interaktivnu grafiku inženjeri rješavaju prostorne probleme direktno u trodimenzionalnom ili virtualnom prostoru, analitičkim metodama ugrađenim u algoritme programa.
- ◆ Rezultati se prikazuju grafički, istovremeno u svim željenim projekcijama.

OBLASTI PRIMJENE U INŽENJERSTVU

U tehnici i inženjerstvu neke od najvažnijih oblasti primjene računarske grafike su:

- Projektovanje korištenjem CAD sistema
- Modeliranje i simulacija
- Inženjerske analize
- Ostale manje zastupljene oblasti

□ *Projektovanje korištenjem CAD sistema*

Projektovanje korištenjem CAD sistema je najrasprostranjenija oblast primjene i o njoj će biti govora u posebnom poglavlju

Modeliranje i simulacija

- ❖ **Modeli** se koriste za prikazivanje realnih ili apstraktnih objekata i pojava, ne samo s ciljem stvaranja slike nego i za prikaz njihove strukture ili svojstava.
- ❖ **Model je pojednostavljen opis objekta**, apstrakcija objekta, jer ne sadrži podatke koji za analizu nisu važni.
- ❖ On može biti *matematički* model, *geometrijski* ili *eksperimentalni*
- ❖ Oni omogućavaju *simulacije*, *analizu*, *ispitivanja* i *otkrivanje* svojstava objekta koji je modeliran.
- ❖ Mogu služiti za *razumijevanje*, *vizualizaciju*, *eksperimentisanje* ili *učenje*, a pomažu i u shvaćanju složenih sistema s međudjelovanjem brojnih komponenti, Omogućavaju također i predviđanje efekata pri variranju određenih ulaznih parametara.
- ❖ U mnogim slučajevima mnogo je jednostavnije (a nekad i jedino moguće) eksperimentisati s modelima umjesto sa stvarnim situacijama.
- ❖ Uglavnom možemo reći da je računaski model elektronska varijanta objekta.

Računasko modeliranje je mnogo šire područje, ali mi ćemo se osvrnuti samo na modele čiji je glavni cilj *grafička interpretacija*. Najčešći tipovi modela koji koriste kompjutersku grafiku su:

- organizacioni modeli (sheme knjigovodstvene klasifikacije, ...),
- kvantitativni modeli (financijski, demografski, klimatski,...),
- geometrijski modeli (arhitektonski, hemijski, u automobilskoj industriji,...).

Iz gore navedenih modela najbitnije je **geometrijsko modeliranje** koje je sastavni dio računarskog modeliranja.

- ❖ **Geometrijski model** prikazuje one objekte čija geometrijska svojstva prirodno traže grafičku reprezentaciju a sastoji se od geometrijskih oblika koji su izvedeni geometrijskim transformacijama
- ❖ Prvi korak i temelj čitavog procesa modeliranja je definisanje **geometrijskih svojstava i oblika** svih objekata na sceni.
- ❖ Nadalje, treba definisati **materijal objekta, boju i teksturu**.
- ❖ Problemi **svjetla** i **refleksije** rješeni su po zakonima geometrijske optike, kao i u prirodi. Postavljeni izvori svjetlosti odgovorni su za realističan prikaz objekta.
- ❖ Završni korak ka realističnosti je tzv. **renderiranje**, tj. **fino ili grubo sjenčenje**.
- ❖ Slijedeći korak je animacija.
- ❖ Većina CAD programa pruža mogućnost izrade geometrijskih modela od kojih je CAD opšte namjene najrašireniji. Najpoznatiji među CAD programima su:
 - **AutoCAD**
 - **3D Studio MAX** (*Autodesk Inc.*)
 - **Microstation** (*Intergraph*)
- ❖ Međutim, **CorelDraw** i slični nisu CAD program ali imaju veliku ulogu u grafičkom dizajniranju. Tako na primjer:
 - Program **Mathematica** (*Wolfram Research Inc*) omogućava prikaz ravanskih krivulja te prostornih krivulja i površina na vrlo jednostavan način.
 - **Rhinoceros** je jako dobar NURBS modeler
 - **LightScape** služi za animaciju i renderiranje
 - **CATIA** (*Dassault Systems*) je vrhunski CAD/CAM/CAE paket
- ❖ Razvijaju se novi modeleri kao **Softimage** (*Microsoft*) i drugi.

Inženjrske analize

Inženjerske analize mogu ponuditi širok spektar analiza fizičkih procesa u projektovanim sistemima a posebno na bazi virtualnog prototipa. Te analize ogledaju se uglavnom u:

- obavljanju standardnih mašinskih proračuna,
- analizi kinematike i dinamike mehanizama u nekom vremenskom opsegu (npr. $1/1000$ sec.),
- analizi naponskih stanja i sopstvenih frekvencija primjenom numeričkih metoda (metoda konačnih elemenata, konačnih volumena ili konačnih diferenci),
- analizi nelinearnih problema dinamike fluida i analizi termodinamike,
- elektrostatičkoj i elektromagnetskoj analizi,
- analizi mjernih signala fizičkih veličina,
- ostalo

3. GRAFIČKI SKLOPOVI I UREĐAJI

Svrstani su u dvije grupe opreme:

- sklopovska* grafička oprema i
- programska* grafička oprema

Sklopovsku grafičku opremu čine:

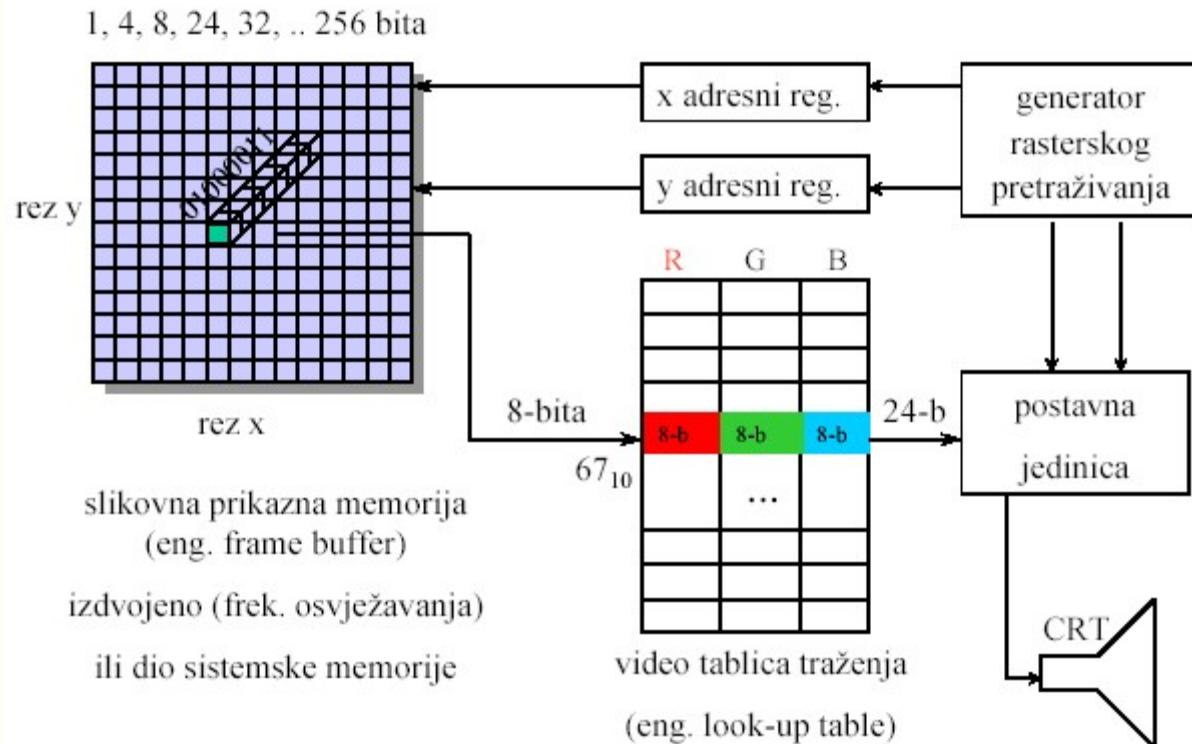
- grafički procesor *GPU* koji u sebi sadrži
 - vektorsku prikaznu procesnu jedinicu i
 - rastersku prikaznu procesnu jedinicu
- izlazne grafičke uređaje
- ulazne grafičke uređaje

Programsku grafičku opremu čine:

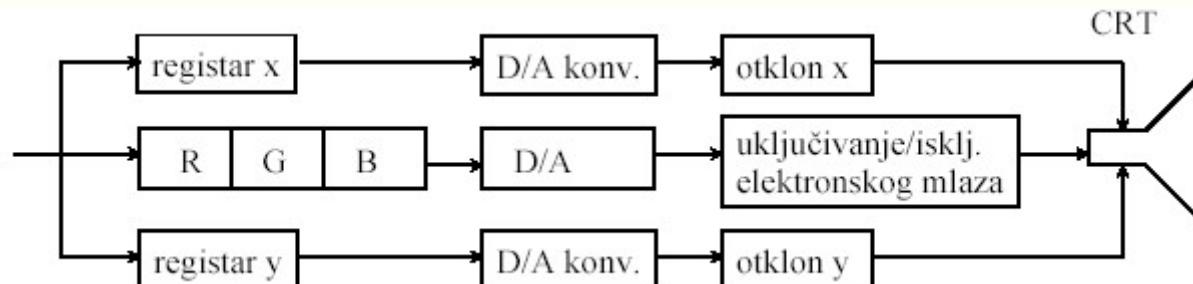
- baza grafičkih rutina
- grafička jezgra načinjena u okviru standarda (*API*)
- gotovi programski paketi za
 - crtanje – *CAD*, razne animacije i
 - prikaz podataka

SKLOPOVSKA GRAFIČKA OPREMA

GRAFIČKI PROCESOR

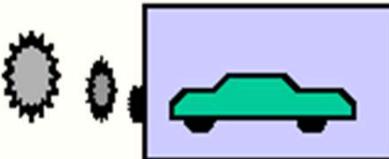
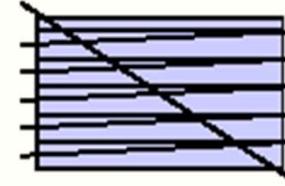


Slika 7- Rasterska prikazna procesna jedinica



Slika 8- Postavna jedinica

Upoređenje *vektorske* i *rasterske* prikazne procesne jedinice:

Vektorska	Rasterska
<p>-Nekadašnja izvedba(ograničena je količina memorije)</p> 	<p>- danas uobičajena</p> 
<p>prednosti:</p> <ul style="list-style-type: none">-tačnost prikaza (ploteri)-jednostavna promjena mjerila <p>nedostaci:</p> <ul style="list-style-type: none">-dugačka prikazna datoteka-popunjavanje poligona <p>=> postoji problem osvježavanja</p>	<p>prednosti:</p> <ul style="list-style-type: none">- veličina prikazne datoteke ne utiče na frekvenciju osvježavanja, odnosno rezoluciju slike (refresh rate) <p>- nedostaci:</p> <ul style="list-style-type: none">potreba pretvaranja u diskretnu prezentaciju <p>=> moguća pogreška diskretizacije</p>

IZLAZNI GRAFIČKI UREĐAJI

dijele se na:

- a) - emitirajuće (*CRT*, s plazmom, *LED* diode)
- ne emitirajuće (*LCD*, tekući kristali)
- b) - vektorske
- rasterske
- c) - osvježavajuće
- s pamćenjem
- Jednobojne
- d) - sivi klin
- višebojne

Neke od karakteristika jedinice za prikaz su:

- slika se pohranjuje u slikovnoj prikaznoj memoriji
- iz memorije podaci se prenose preko *DAC* do display-a puno puta u sekundi
- važna je *brzina* osvježavanja zbog eksponencijalnog slabljenja intenziteta svjetla koje emitira fosfor, više kvantnih nivoa:
 - florescencija dio (*ms*)
 - fosforoscencija 10-60 (*ms*)

Visoka *perzistencija* znači da svjetlu treba dugo da oslabi, te se tada može sporije osvježavati

➤ *dijagonala*

- npr. nazivna dijagonala 17"
- vidljiva dijagonala 15,6-16,2

➤ frekvencija osvježavanja

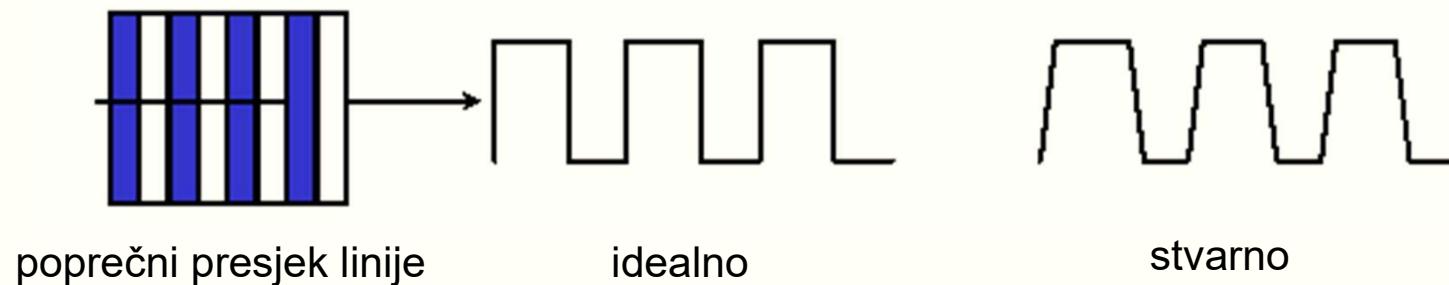
- vertikalna frekvencija (broj slika u sekundi) *60-160 Hz (85 Hz propisano VESA standardom)*
- horizontalna frekvencija (broj linija u sekundi) *30-100 KHz*
- frekvencija osvježavanja slikovnih elemenata (brzina paljenja i gašenja elektronskog snopa) *50-160 MHz*, odnosno širina pojasa (eng. pixel rate)

➤ geometrijska svojstva

- kada prikazujemo kružnicu želimo da nema oblik elepse

➤ jasnoća, zrnatost, rezolucija

- broj crnih/bijelih linija koje se mogu prikazati i odvojene su (obično se izražava po jedinici udaljenosti *dpi-dot pitch*)



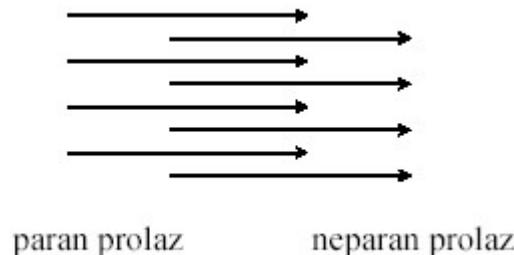
➤ Razmak

- između tačaka iste boje (eng. *dot pitch*-bojkano tačkama), dijagonalno razmak između tačaka *0,25-0,28 mm*, razmak rupica na sitastoj rešetki
- između pruga (eng. *stripe pitch*), horizontalna udaljenost *0,21-0,28 mm*, razmak na aperturnoj rešetki

- Veličina jedne tačke koja može biti načinjena (*eng. dot, spot size*)
- *adresibilnost*, tj. broj individualnih tačaka (po inču) koji može biti načinjen - obrnuto proporcionalno udaljenosti
 - poželjno je da veličina tačke bude veća od udaljenosti središta



- iscrtavanje s *preplitanjem*
 - (*eng. interlaced/noninterlaced*) ako sporije iscrtavamo možemo iscrtati veću sliku



- *toplina boje* – spektar zračenja (*eng. color temperature*)
 - kada se crno tijelo zagrije na temperaturu 9300 K ima identično zračenje monitoru (crna boja)
- *degauss*
 - uklanjanje statičkog naboja (demagnetizacija ekrana npr)

➤ različite karakteristike fosfora, *DAC*, elektronskog topa, sitaste maske, brzine i organizacije memorije utječu na konačne mogučnosti.

Na primjer:

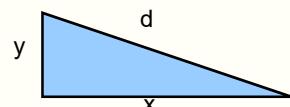
vertikalna frekvencija- 76 Hz, NI

oštrina- 1152x900

razmak pruga- 0,26mm=0,0103"

dijagonalna- $d=20''$

tipičan omjer slike- $y/x = 0,75$



$$\Rightarrow 20^2 = x^2 + 0,75^2 x^2 \Rightarrow \dots \dots \dots x = 16'' \Rightarrow 1550 \text{ slikovnih elemenata}$$

$$\Rightarrow \text{horizontalna frekvencija} \dots \dots \dots 900 \times 76 \text{ Hz} + 10\% \sim 76 \text{ kHz}$$

$$\Rightarrow \text{paljenje/gašenje elektronskog snopa} \dots \dots \dots 1152 \times 900 + 30\% \sim 100 \text{ MHz}$$

$$\Rightarrow \text{postavljanje zahtjeva na brzinu D/A pretvaranja}$$

$$3 \times 8 \text{ b i vrijeme pristupa memoriji} \dots \dots \dots \sim 10 \text{ ns}$$

➤ raspoložive memorije

- *SDRAM* (interno paralelna organizacija) ~ 60 ns, slijedeći ~10ns
- *SGRAM* (synchronous graphics RAM, ima dodatne grafičke mogućnosti, može biti i dvopristupni)
- *DRAM* (eng. dynamic)
- *VRAM* (eng. dual port)
- *EDO RAM*
- *RAMBUS*
- *WRAM* (eng. window)

NAPOMENA

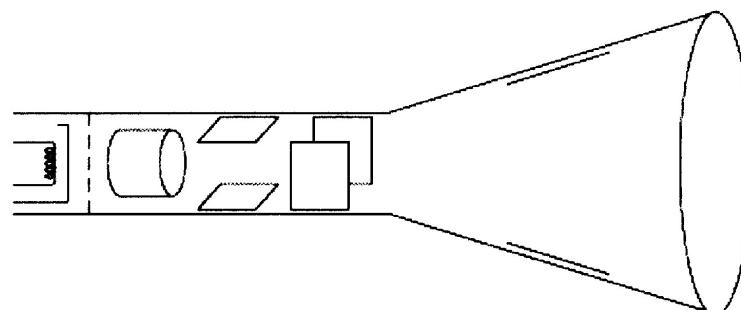
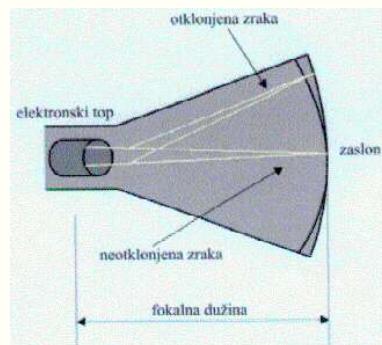
Pristup svakoj memorijskoj čeliji direktn je od strane mikroprocesora te se takva memorija naziva *RAM* (Random Access Memory - memorija s direktnim pristupom) i u njima se obavlja ČITANJE i UPISIVANJE podataka. Prema tome:

RAM - služi za upis i čitanje tekućih podataka i programa i može se vrlo lako i brzo mijenjati njen sadržaj. Sadržaj se gubi (zaboravlja) pri prestanku napajanja računara. Često se označavaju s *R/W* (read/write - čita/piše). Izrađuju se kao STATIČKE (*SRAM*) i DINAMIČKE (*DRAM*).

JEDINICE ZA PRIKAZ

• CRT (princip rada)

- u elektronskom topu *žarna nit* grijе *katodu* koja emitira snop elektrona, *kontrolna mrežica* određuje količinu elektrona koja će proći dalje i na taj način određuje osvjetljenje sistem za *fokusiranje* elektronskog snopa, dinamički fokusira snop ovisno o položaju na zaslonu (defokusiran-mutna slika). Teži se ravnom zaslonu (horizontalno, vertikalno)
- horizontalni i vertikalni *otklonski sistem* otklanjaju snop
- visoko pozitivna *metalizacija* (anoda) 15.000-20.000V ubrzava elektrone
- sitasta *maska* ili aperturna rešetka
- *fosfor* naparen na staklo- prelazak u više kvantno energetsko stanje a prilikom povratka elektrona emitira se energija u obliku *svjetla* određene talasne dužine R.G.B. Obično postoje razlike u fosforu tako da ista slika izgleda različito na različitim monitorima
- miješanje talasnih dužina => princip *oka čovjeka*
- uticaj ambijentnog svjetla na osvjetljenje i kontrast
- kalibriranje boja
- na elektronski snop (elektro) magnetska polja imaju uticaj

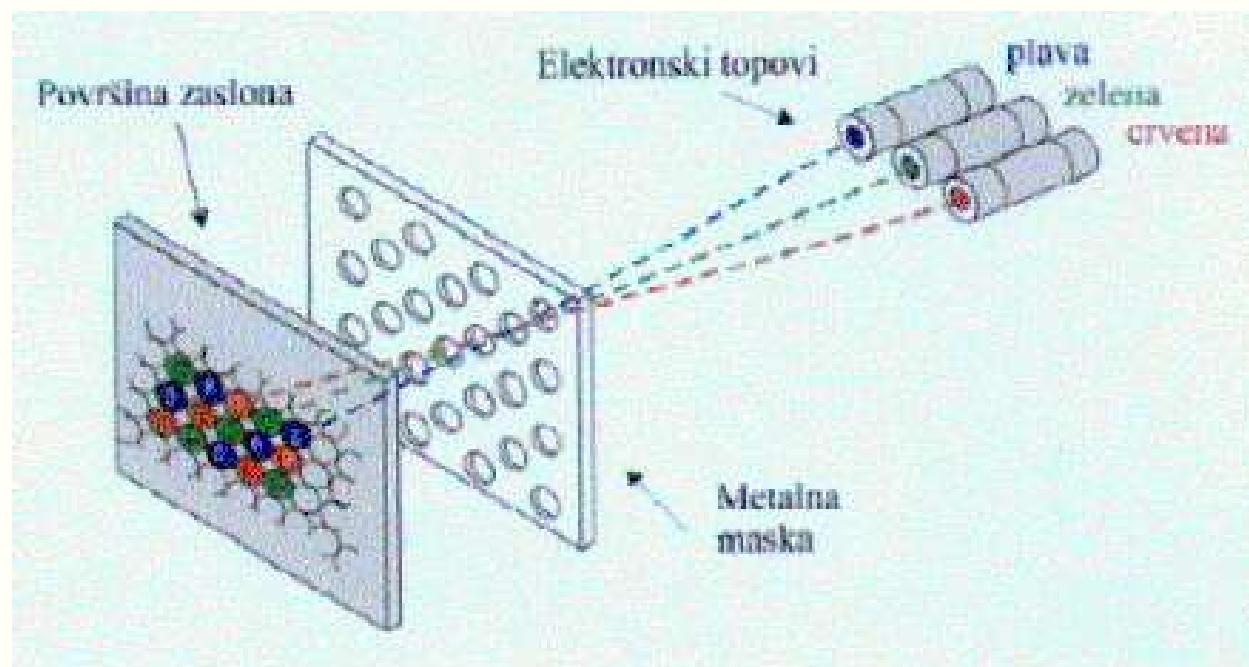


•CRT - FST ravna cijev sa sitastom maskom (eng. shadow mask)

- delta raspored fosfornih cijevi (topova) u nizu (eng. *in line*)
- sitasta maska se radi od legure invar kod koje je prisutan problem grijanja i naprezanja
- problem je također vrlo precizno fokusiranje snopa na pripadajući fosfor (neujednačena slika po površini zaslona u vidu crveno-plavih tragova)
- veliki dio površine je zaklonjen maskom (~20% elektrona pogodi fosfor) pa je zbog toga smanjeno osvjetljenje



Sitasta maska



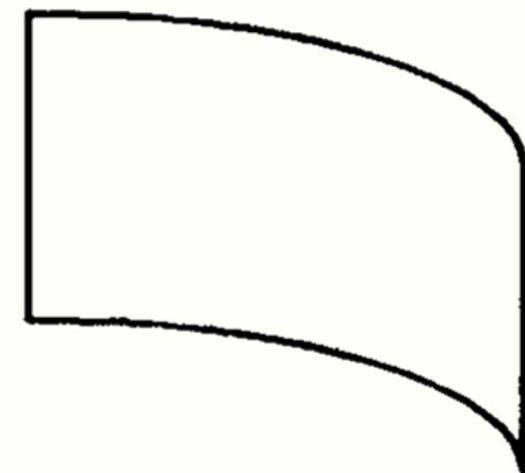
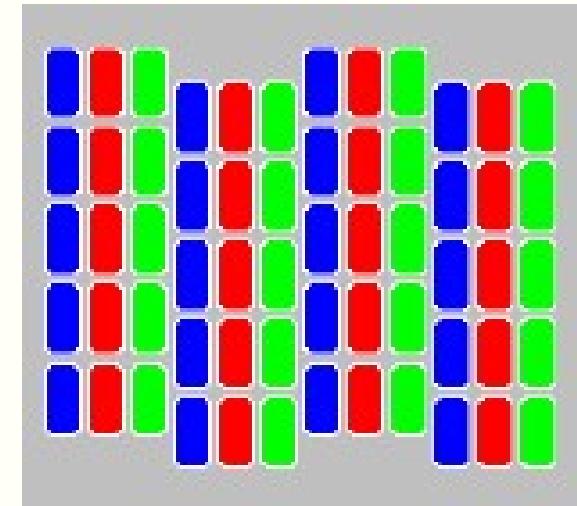
•CRT – trinitron cijev sa aperturnom (rupičastom) rešetkom (eng. aperture grill)

- posjeduje niz vertikalnih traka (žica),
- zauzimaju manju površinu pa je slika svjetlijia, kontrasnija, vjernije su boje
- prilikom rada rešetka se ugrije pa se javlja problem deformacije i vibracije. Stoga se dodaju dvije žice od volframa za učvršćivanje (obično su teži zbog problema učvršćivanja i osjetljiviji su na transport),
- ima izvjesnu horizontalnu zakrivljenost,
- skuplji su 30-50% od prethodnih.

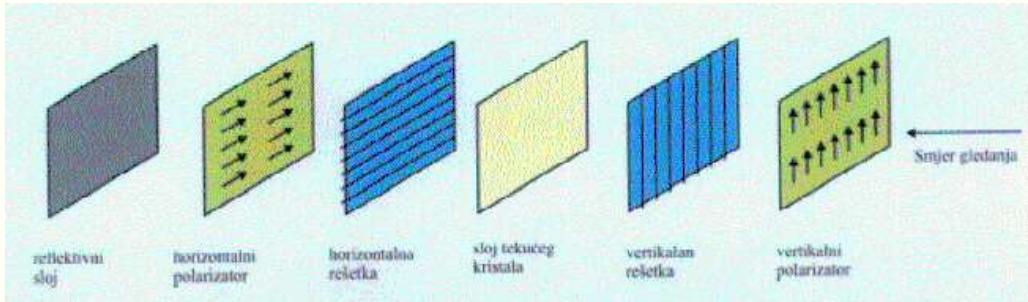
Na primjer:

- ❖ Sony-64 je koristio cijev chromatron za prvi TV u boji i imao je niz vertikalnih žica, a
- ❖ Sony 67 posjedovao je cijev sa tri elektronska topa i jednim sistemom za fokusiranje

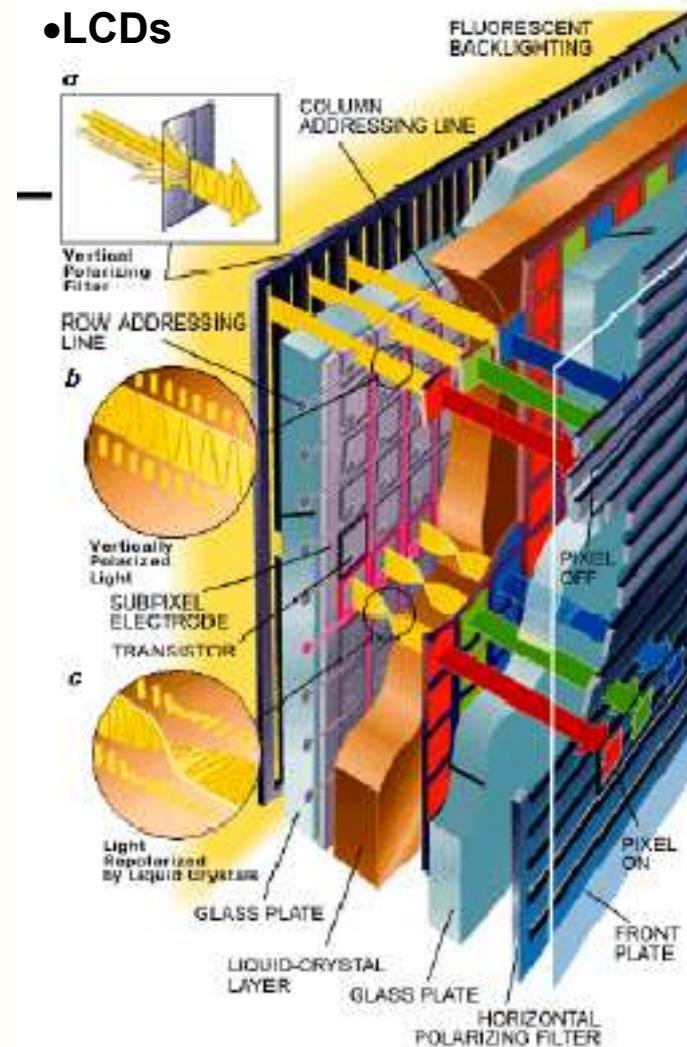
•CRT – kombinacija ravne cijevi i trinitron



•LCD – prikazna jedinica s tekućim kristalima

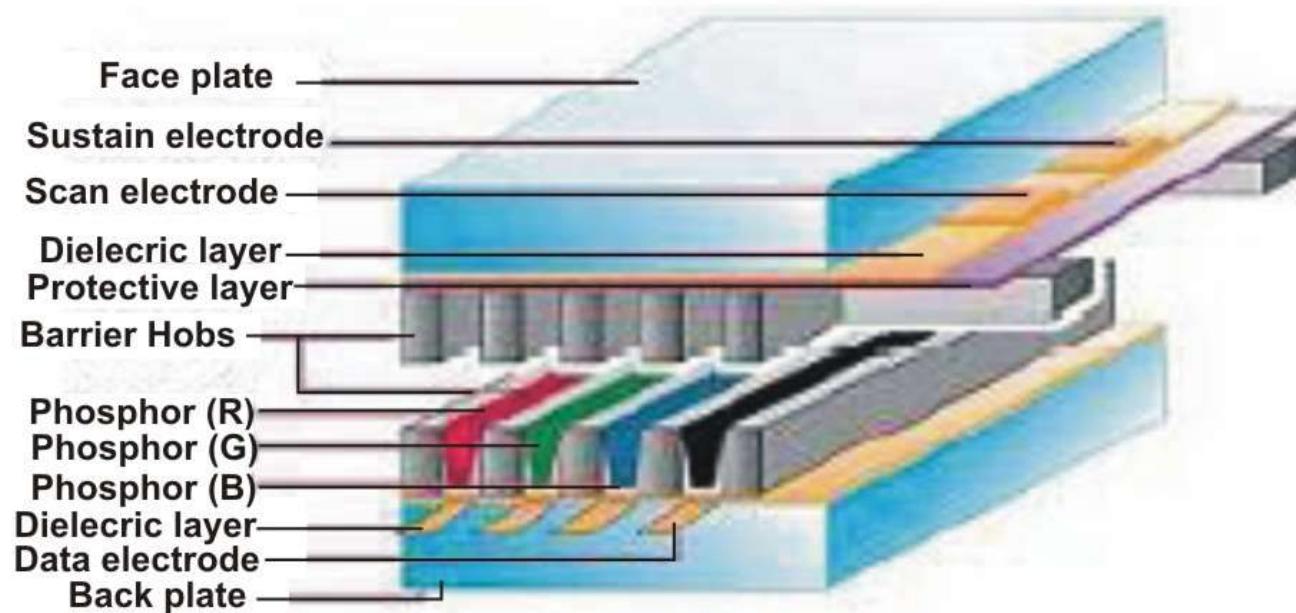


- materijal tekućih kristala je načinjen od dugačkih molekula
 - ❖ kada je kristal u *električnom polju* nema polarizirajuća svojstva na svjetlo koje dolazi pa svjetlo ostaje vertikalno polarizirano i *ne prolazi* kroz horizontalnu polarizaciju
 - ❖ kada kristal nije u električnom polju zakreće ravninu polarizacije za 90° iz vertikalne u horizontalnu
- *TFT* (eng. *thin film tranzistor*) na svakom (x,y) ima tranzistore koji služe kao aktivna memorija dok se stanje ne promijeni
- Prednosti* su mu ti što su lagani, mala potrošnja, mali su po z-osi, povoljna im je cijena
- Nedostaci* su mu što nisu izvor svjetlosti pa se ne može koristiti stražnje osvjetljenje za projekcije, spora im je promjena slike, ugao gledanja im je ograničen
- upotrebljavaju se kao:
 - ❖ prijenosni računari
 - ❖ projektori
 - ❖ *HMD* (*pokazivači koji se nose na glavi*-engl. *head-mounted display*)



•Prikazna jedinica s plazmom

- kod *CRT* prikaznih jedinica velika je dubina po z-osi i tehnološki je ograničena veličina a i na mjestu ukrštanja elektroda (kod *CRT*) je adresirano mjesto, gdje će doći do ionizacije xenon/neon (*Xe/Ne*) plina što izaziva ultravioletno zračenje koje aktivira fosfor, tj. svjetlo (znači da uređaj nije pasivan), a
- kod *LCD* prikazne jedinice također je problem načiniti veće od 20" bez vizualnih efekata, ali zato prikazne jedinice s plazmom mogu imati veličinu ~40" (100kg), odnosno i 55"-100"



•Ostali izlazni grafički uređaji

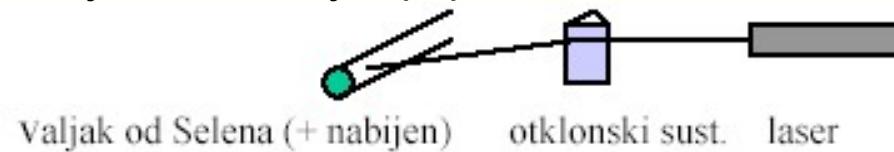
□ *Pisači*

❖ matrični

- glave (7-24 iglica) više glava u boji (ili ribon u boji)
- rasterski uređaj (konverzija)
- povećanje adresibilnosti (2 pomaknuta reda, 2 prolaza)

❖ laserski

- naboј ostaje tamo gdje treba biti crna boja (gdje laser ne izbije naboј)
- negativno nabijeni toner-valjak-papir



❖ ink-jet

- raspršuje boju Cyan, Magenta, Yellow, Black u jednom prolazu



❖ termo

- ugrađena pera prenose pigmente s voštanog papira CMYK (u više prolaza)



❖ 3D pisači (Zprinter)

❖ uređaji za stereolitografiju

- važno za brzu izradu prototipa

Crtači (ploteri)

s pisaljkom (PEN)

- optimiranje praznog hoda, akceleracija
- prikaz karata

elektrostatski

- negativno nabijeni papir, pozitivno nabijena tinta
- brži je i manje kontrasni od plotera sa pisaljkom

Tokarilice, glodalice

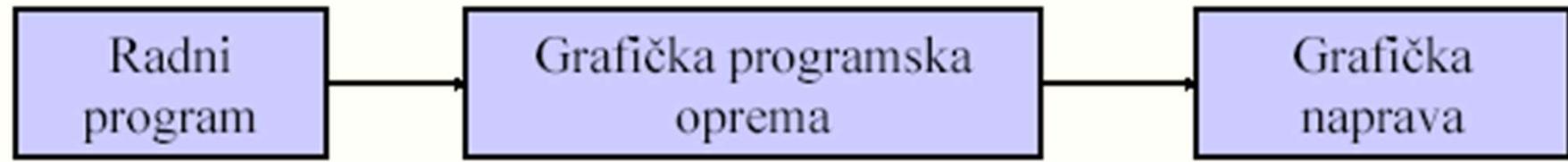
- služe za izradu trodimenzijskih objekata

ULAZNI GRAFIČKI UREĐAJI

- ❖ tablice (eng. tablet) s pisaljkom, putem dodira na osjetljivu ploču
 - kapacitivna sprega, elektromagnetska, zvučna, naponski gradijent
- ❖ miš
 - mehanički, mehaničko optički, optički
- ❖ 3D zvučno pero, svjetlosno pero
- ❖ sistem za 3D uzorkovanje
 - ultrazvučno
 - CT, PET (računarska tomografija)
 - MR (magnetska rezonancija)
 - laserskim snopom, mogućnost uzorkovanja boje i temperature

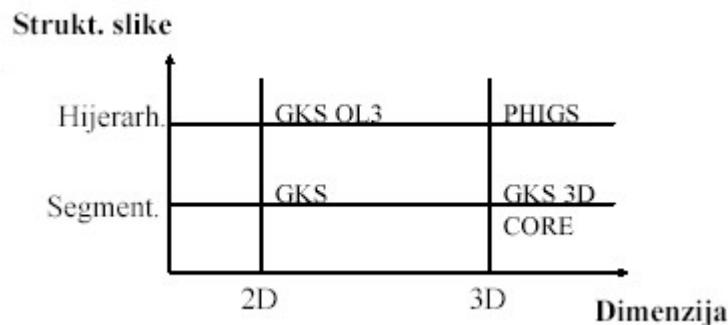


PROGRAMSKA GRAFIČKA OPREMA



- Baza grafičkih rutina (grafičke rutine koje se pozivaju iz nekog višeg programskog jezika s atributima C, C++). Teži se da ova baza bude načinjena prema specifikaciji *API* standarda, ili prema nekom drugom standardu
 - “+” znači neovisnost radnog programa o sklopovskoj opremi
 - “-” znači da se obično može ostvariti potpuna iskorištenost sklopovske opreme
- Standardima su propisani
 - API (*Application Program Interface*), prema kojima se prave grafičke baze npr.
 - ❖ *OpenGL-mesa, DirectX, Direct3D, Phigs-PEX*
 - zapisi
 - ❖ slika *TIF, GIF, BMP, JPG, HPGL, PS* (rasterski, vektorski)
 - ❖ niz slika *GIF, MPEG, AVI, DVX*

- Grafičke jezgre načinjene u okviru standarda
 - 3D CORE (*Core Graphics System*)
 - ❖ 1979. ACM SIGGRAPH (*Association for Computing Machinery Special Interest Group on Graphics*)
 - GKS (*Graphics Kernel System*)
 - ❖ ISO 88, 94, 97, 98, 99 (*International Standards Organization*)
 - ❖ ANSI 85 (*American National Standards Institute*)
 - PHIGS (*Programmer's Hierarchical Interactive Graphics System*), PHIGS+ za pseudorealističan prikaz
 - ❖ ISO 90, 97
 - ❖ ANSI 88
 - VRML (*Virtual Reality Modelling Language*)
 - ❖ ISO 97, 98, 99



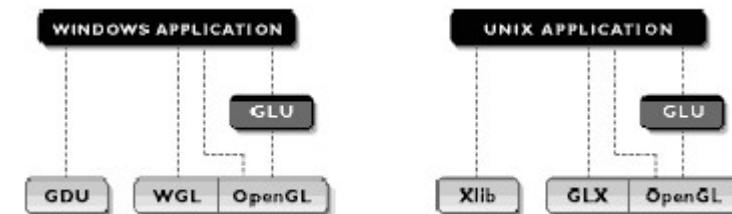
- Osim službenih standarda postoje “*de facto*” ili industrijski standardi

– <i>GL</i> , <i>OpenGL</i>	<i>SGI</i>
– <i>X Window System</i> ’s <i>Xlib</i>	<i>MIT</i> <i>PEX</i>
– <i>RenderMan</i>	<i>Pixar</i>
– <i>PostScript</i>	<i>Adobe</i>
– <i>OpenFlight</i>	

Komercijalno su ovi standardi značajniji od oficijelnih standarda jer se jednostavnije mogu mijenjati

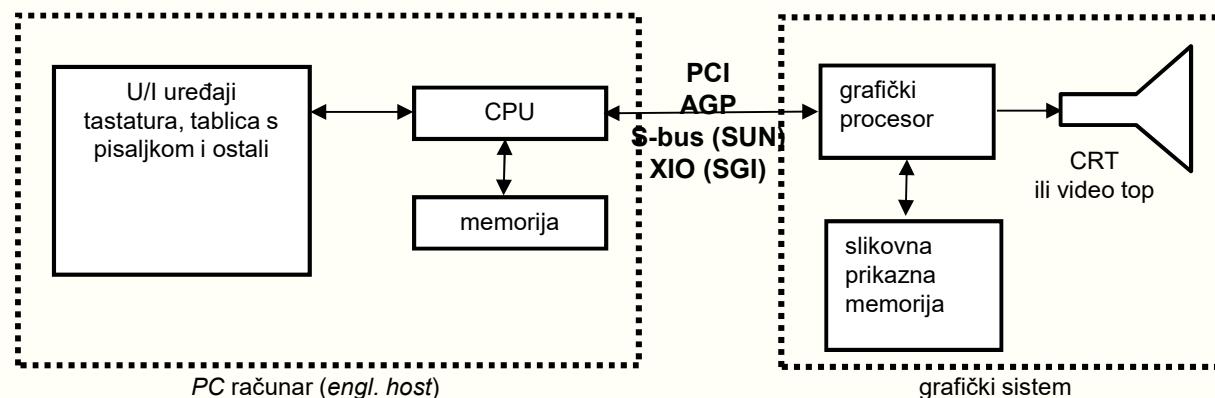
- *OpenGL* (*eng. Open Graphics Library*)

- *IrisGL* je temelj za *OpenGL*
- 1992 nastaje industrijski standard
- nadalje se pojavljuje programsko sučelje prema grafičkom sklopolju, neovisan o platformi
- *eng. state machine* kontrolišu skup specifičnih operacija crtanja 2D/3D (definiše kontekst za prikaz)
- *GLU Utility Library* (pomaže u modeliranju i nekim operacijama s prozorima) *OpenGL* se temelji na *Frame Buffer*-u međutim u svakom konceptu ne podržava grafičke ulazno izlazne uređaje kao što su miš i tastatura
- *GLX X-server extension API* definiše mrežni protokol za *OpenGL* naredbe za prikazivanje (enkapsulirane u *X* protokol)



4. ORGANIZACIJA MIKRORAČUNARSKOG GRAFIČKOG SISTEMA

Organizovanost mikroračunarskog grafičkog sistema kod *računara* zasniva se na uzajamnoj interakciji, tj. komunikaciji između dijelova koji su u funkciji stvaranja grafičkog prikaza a putem određenih veza koje ih povezuju (sabirnice- unutarnje i vanjske).

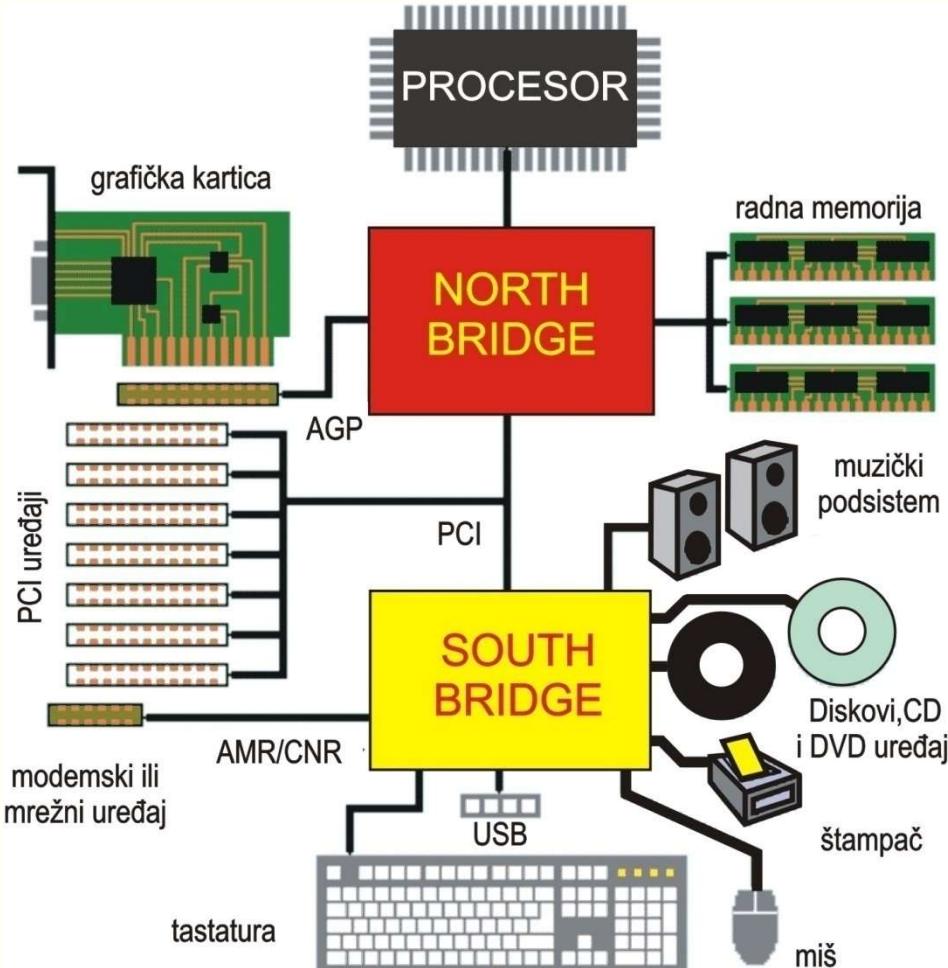


Organizacija mikroračunarskog grafičkog sistema

Da bi sve komponente računara i priključeni mu uređaji mogli uspješno funkcionirati kao jedna jedinstvena zajednička cjelina brine se skup elektronskih logičkih komponenti objedinjenih u jednom ili više integriranih krugova nazvanih **CHIPSET**, kojima je zadatak uspješno upravljanje i razmjena **podataka** između pojedinih uređaja unutar računara i dodatnih priključenih uređaja

Vodeći proizvođači u *PC* svijetu su **INTEL**, **AMD** (*Advanced Micro Devices*), **VIA**, **ALI** (*Acer Labs Inc.*) i **SIS** (*Silicon Integrated Systems*). Ostali proizvođači kao **SUN** ili **SILICON GRAPHICS** dizajniraju, proizvode ili naručuju chipset-ove prema vlastitim potrebama.

U osnovi chipset se sastoji od integriranih krugova sa sistemima za podršku **grafičkim resursima** i **perifernim uređajima**. Promatrano ka dvije cijeline može se prikazati kao na slici.

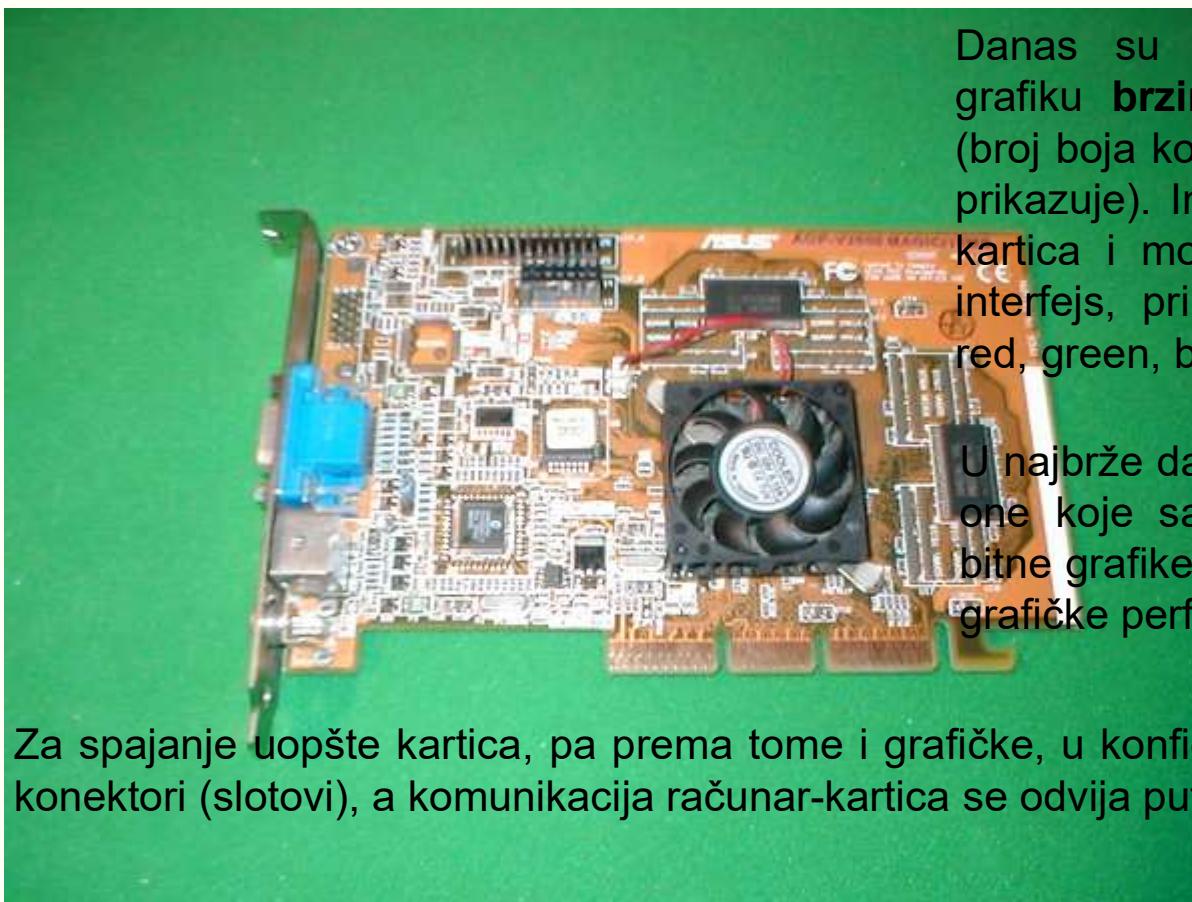


Blok shema chipset-a

Važan element grafičkog sistema kod savremenih računara je poseban elektronički sklop - **grafička kartica** - SVGA ili VGA ili neku drugu.

GRAFIČKA KARTICA

Grafička kartica ili video-adapter (VGA, SVGA kartica) posreduje između dijela računarske radne memorije namijenjene ispisu na ekran monitora, te prevodi digitalne podatke iz memorije u analogne video signale, tj. u čovjeku razumljive znakove: slova, interpunkciju, linije i dr. Postoji više različitih video-adaptera, kao i više kakvoća slike na ekranu monitora. Kvalitet slike može biti standardan, visok ili veoma visok. Za uobičajene obrade s koriste video-adapteri sa standardnom kakvoćom slike. Za ocjenu njihove kvalitete potrebno je poznavati veličinu njihove memorije (standardno 8, 16, 32, 64 Mb, itd.).



Danas su najvažniji pojmovi vezani za grafiku **brzina**, **rezolucija** i **dubina boje** (broj boja koje sistem može istovremeno da prikazuje). Interfejs između većine grafičkih kartica i monitora zove se analogni *RGB* interfejs, pri čemu je *RGB* skraćenica za red, green, blue (crveno, zeleno, plavo).

U najbrže današnje grafičke kartice spadaju one koje sadrže jedinicu za obradu 256-bitne grafike kojom se postižu optimalne *3D* grafičke performanse.

Za spajanje uopšte kartica, pa prema tome i grafičke, u konfiguraciju PC računara koriste se konektori (slotovi), a komunikacija računar-kartica se odvija putem sistema sabirnica.

Kada se govori o *grafičkoj kartci* i uopšte o njenoj ulozi u *organizaciji mikroračunarskog grafičkog sistema* potrebno je u vezi s tim povezati slijedeće pojmove:

- Centralni procesor
- Sistemska sabirnica i njezin interfejs za grafičku karticu
- Video memorija
- Grafički čip na kartici
- Digitalno analogni konvertor (DAC)

a) *Centralni procesor i grafika*

Centralni procesor obrađuje grafičke podatke i putem grafičke kartice šalje ih na monitor. Kad neki program želi prikazati podatke, on to čini na način da *naređuje centralnom procesoru da ih smjesti na grafičku karticu*. Načini na koje to procesor radi se razlikuju, budući da postoje dvije osnovne vrste grafičkih kartica: *prosti baferi slike* i *koprocesorske/akceleratorske ploče*. To znači da centralni procesor kontroliše grafičku karticu.

b) *Sistemska sabirnica*

Centralni procesor je povezan s grafičkom karticom preko sistemske sabirnice.

U PC svijetu često se susrećemo sa sabirnicama pod nazivom:

- **ISA (Industry Standard Architecture)** - 16 bit-na sabirnica,
- **EISA (Enhanced Industry Standard Architecture)** - 32 bit-na sab,
- **MCA (Mikro Channel Architecture)** - u verziji 16 i 32 bit-a,
- **VLB (VESA Local Bus)** - u verziji 32 bit-a,
- **PCI (Peripheral Component Interconnection)** - 32 bit-a,
- **AGP (Accelerated Graphics Port)** - 32 bit-a.

Broj *bit*-a sabirnice odnosi se na sabirnicu podataka. Brzina prijenosa podataka (propusnost) nije ista za sve navedene sabirnice a uobičajena brzina za pojedine iz popisa je $8MB/s$, $32MB/s$, , $132MB/s$ i $524MB/s$ redoslijedno.

Posebna sabirnica zamišljena za grafičku memoriju je *AGP sabirnica*.

Navedena sabirnica ima direktnu vezu s procesorom na većem taktu od *PCI* sabirnice te se u slučaju nedostatka grafičke memorije njezin upravljački sklop posluži resursima "klasične" memorije. Na taj način osjetno se ubrzavaju grafičke aplikacije pisane upravo za ovakav tip računara. Osim toga interno je i povećan takt *PCI* sabirnice.

Kako noviji mikroprocesori rade na nekoliko puta većim frekvencijama od mogućnosti sabirnice, poseban upravljač sabirnice vodi brigu o prijenosu podataka između procesora i uređaja računara. Stoga je potrebno da brži procesori uz svoju sabirnicu imaju međumemoriju (cache) za internu upotrebu, te za vrijeme jednog taka sabirnice obavljaju više internih radnji mikroprocesora.

Visokokvalitetne grafičke kartice imaju 64 ili više *RAM* memorije, tako da je maksimalna količina podataka kojom bi grafička kartica mogla operirati bila 72×64 odnosno $4608 MB$ u sekundi.

c) Video memorija

Slika se potom smješta u video memoriju. Kao i kod standardne memorije, i ovdje vrijedi pravilo: "što brže i više to bolje", iz jednostavnog razloga, što više boja i tačaka na ekranu želimo, to je grafičkoj kartici potrebna veća memorija. Ali video memorija ima i neke posebne potrebe, jer je obično adresirana po blokovima i istovremeno joj pristupa nekoliko čipova, tačnije, centralni procesor i grafički čip. Postoji:

- Dvoulazna memorija –VRAM*
- Memorija s dva ulaza i adresiranjem blokava – WRAM*
- Multibank DRAM (MDRAM)*
- Synchronous Graphics RAM (SGRAM)*
- Double Data Rate RAM (DDR SDRAM)*

Dvoulazna memorija –VRAM

Čipu *RAM* memorije pristupa isključivo centralni procesor i niti jedan drugi čip, osim ako se ne odvija *DMA* operacija, u kojem slučaju centralni procesor ne pristupa memoriji. S druge strane, memorija na grafičkom čipu "komunicira" s dva čipa: centralni procesor puni podatke u video memoriju, dok ih grafički čip izvlači iz nje.

Na grafičkim karticama sa *običnim RAM*-om to nije moguće istovremeno. Stoga se centralni procesor i grafički čip izmjenjuju u pristupu.

Na nekim grafičkim karticama postoji posebna vrsta memorije čiji je originalni naziv "memorija s dva ulaza" (*Dual Ported RAM*), a koja je poznata kao *VRAM (Video RAM)*

Memorija s dva ulaza i adresiranjem blokava – WRAM

Važna karakteristika memorije jest njena "blokovska" priroda. Moderni grafički interfejsi adresiraju memoriju u velikim blokovima umjesto bajt po bajt. *RAM* memorija s prozorom (*window RAM-WRAM*) omogućava da se pomoću samo nekoliko naredbi adresiraju blokovi, odnosno "prozori" memorije.

MDRAM, SGRAM i DDR

Multibank DRAM (MDRAM) je namjenjena za niže troškove i dobre performanse. Ova tehnologija omogućava da se memorija dodaje u "komadima" od 32 KB, tj upravo onoliko za koliko se smatra da je dovoljno. *MDRAM* je puno brža od *VRAM* i *WRAM*.

Synchronous Graphics RAM (SGRAM) radi na većim brzinama od ostalih video memorija, između 66 i 80 MHz.

Double Data Rate RAM (DDR SDRAM) ne povećava učestalost na kojoj radi memorija, već pomicće dva puta više memorije po jednom ciklusu, tako da udvostručuje brzinu kojom se podaci premještaju.

Video prolaz

Neki video akceleratori (ubrzivači visokih performansi) brzinu postižu tako što mapiraju svoju video memoriju direktno u adresni prostor računarske memorije. Taj adresni prostor se naziva *video prolaz* (eng. *aperture*).

e) Grafički čip

Kad se slika konačno nađe u *RAM* memoriji mora se pretvoriti u format digitalnog videa. To se obavlja pomoću "čipa za prikaz", "grafičkog čipa" ili "čipa za sliku" ovisno kako tko zove. Tokom godina pojavilo se puno različitih grafičkih čipova, međutim, danas je kod grafičkih čipova najvažnije da li je riječ o jednostavnom *baferu slike* ili o nekoj vrsti *akceleratorskoko procesorskog čipa*.

□ Baferi slike

Većina grafičkih čipova napravljenih prije 1992. su bili *baferi slike*. Bafer slike znači da je kartica "*naseljena*" memorijskim čipovima u koje se smješta slika, vrlo slična onoj koja se pojavljuje na ekranu. Svakoj tački (*pikselu*) na ekranu odgovara jedna lokacija u video memoriji, odnosno svakoj boji odgovara numerička vrijednost u memoriji. Nevolja s baferima slike proizlazi otuda što svaku od tih tačaka mora rasporediti centralni procesor.

□ Koprocesori i akceleratori

I video *koprocesori* su jedna vrsta centralnih procesora posebne namjene. Njihov je zadatak da brzo prebacuju tačke. Dok je matematički koprocesor samo jedan čip, video koprocesori su čitave ploče. Koprocesori mogu biti vrlo brzi, ali jednakо tako i skupi.

Koprocesori se povezuju sa normalnim grafičkim karticama putem konektora za proširenja.

Koprocesor je kompletni mikroprocesor koji se može programirati tako da obavlja bilo koji zadatak iz područja rada centralnog procesora.

Akcelerator nije procesor opće namjene, već čip koji može brzo obaviti određene grafičke zadaće. Mnogi akceleratori vrlo brzo prenose blokove bit mape. Korisnici grafičkog interfejsa vrlo često slike nazivaju *bit mapama*. Sporost Windowsa je u velikoj mjeri prouzročena smještanjem bit mapa na ekran, ili pomicanjem bit mapa s jednog na drugi dio ekrana. pomicanje bit mapa se naziva *prijenos bloka bit mape*.

Mnogi jednostavniji Windows akceleratori nisu ništa drugo do *VGA* kartice sa čipom za prijenos bloka bit mape.

f) Digitalno-analogni konvertor (DAC)

Nakon što je grafički čip napravio digitalnu sliku ostaje da se digitalna slika pretvori u analognu. Taj posao obavlja posebni čip *digitalno-analogni konvertor* (*Digital-to-analog Converter, DAC*).

Konvertori se razlikuju po sposobnosti prikazivanja boja. petnaestobitni konvertor daje 32768 boja, šesnaestobitni 65536, osamnaestobitni 262144. Dvadesetčetverobitni i tridesetobitni digitalno-analogni konvertori daju 16 000 000, odnosno milijardu boja.

Konvertor se obično ne može nadograđivati.

DIJELOVI GRAFIČKE KARTICE

Osnovni dijelovi grafičke kartice su:

- video upravljač (procesor)
- veza prema U/I sabirnici
- ROM karakter generator
- Video RAM memorija
- Generator video-*RGBI* signala (signali slike i inteziteta)
- Generator signala sinkronizacije (*HSYN* i *VSYN*) monitora

- Generator *RGBI* i generator sinkronizacijskog signala tvore sklop upravljanja i monitorske kontrole i u sebi pored ostalog sadrži sklopove koji pretvaraju digitalne podataka u analogne signale (*D/A* pretvaranje - *DAC*) i tablicu boja (palette) na temelju koje se iz digitalnog zapisa određuje boja. Podaci o tablici boja i broju elemenata slike (jasnoća) čuvaju se u video BIOS-u.

- Sve sklopove video upravljač povezuje preko interne video sabirnice. Skuplje kartice mogu sadržavati i druge uređaje kao TV birač kanala, VHS i SVHS video izlaz i slično. Naravno sve su to port-ovi koji pripadaju video kartici.

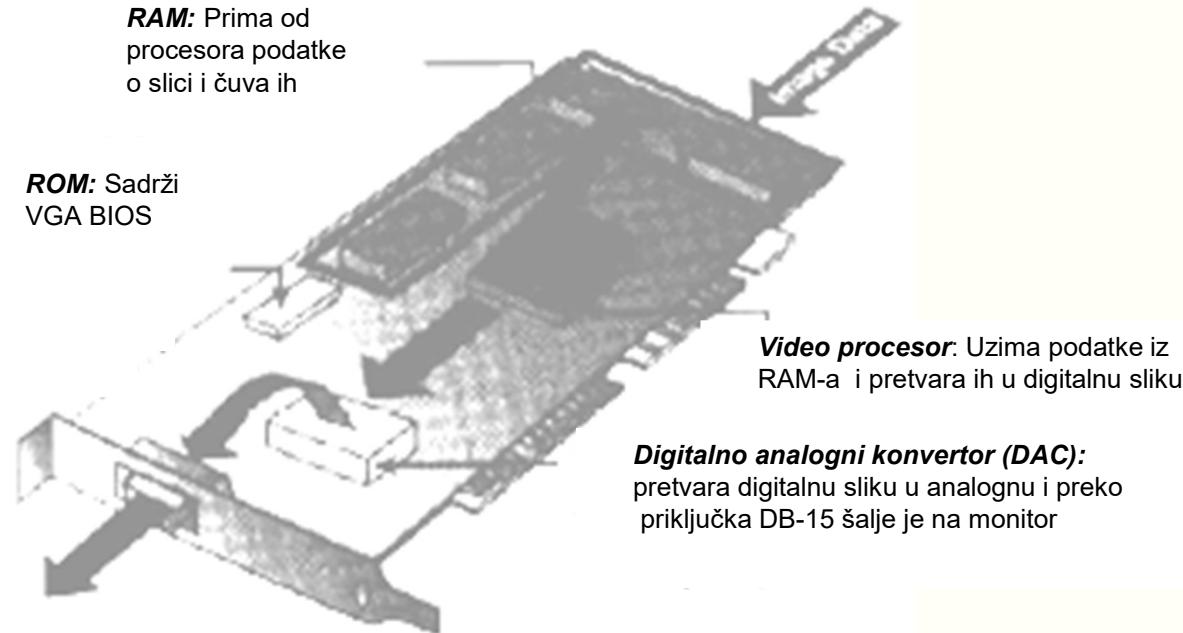
U osnovi mogu da rade u dva moda rada:

- grafički mod
- tekst mod

Tekst mod prikazuje samo karaktere koje daje karakter generator i obično je to prikaz karaktera u mreži od 80 stupaca i 25 redova. Grupa karaktera istog stila naziva se *FONT*. Broj stupaca i redaka može se povećati ili smanjiti u ovisnosti o mogućnostima kartice.

Grafički mod je zahtijevniji, jer finije dijeli prikaz na ekranu (gustoće 320X200 elemenata slike i naviše) i može praviti tekst karaktere promjenjive po obliku i veličini (vektorski-skalabilni fontovi). Može se prihvati tumačenje da je prikaz na ekranu podijeljen na grafičku i tekstualnu sliku koje se mogu po potrebi i istovremeno prikazati. Današnje kartice u suštini nemaju odvojene tekstualne i grafičke funkcije, već se tekst mod rada emulira uz pomoć sofisticiranih video-elektronskih sklopova. Otuda i mogućnost skalabilnosti fontova, tj. izmjena veličine slova po nahođenju.

- Kratkospojnicima na kartici određuju se početni parametri kartice i usklađuju s tipom monitora, te broj prekida i veličina memorije kartice. Uz karticu se isporučuje i programska podrška orijentisana prema važnijim svjetskim software-skim produktima (*driver-i*) koji uz karticu omogućuju lakše korištenje DOS i Windows aplikacija.



Shematski prikaz VGA kartice sa osnovnim dijelovima

TIPOVI GRAFIČKIH KARTICA

U dosadašnjem razvoju poznate su sljedeće značajne grafičke kartice:

- *MDA (Monochrome Display Adapter)* monokromatski adapter,
- *CGA (Colour Graphic Adapter)* grafički adapter s mogućnošću prikaza u boji,
- *HGC (Hercules Graphic Card)*,
- *EGA (Enhanced Graphic Adapter)*,
- *VGA (Video Graphic Adapter)*,
- *SVGA (Super Video Graphic Adapter)* i
- ostale

Glavne karakteristike nekih grafičkih kartica:

STANDARD	Godina	Jasnoča	Tekst/grafika	Boje	Polje znaka	Kompatibilni modovi	Vert. frekv.(Hz)	Hor. frekv. (kHz)
MDA (Monochro-me Display Adapter)	1981.	720x350	T	1	9x14	-	50	18.43
CGA (Color Graphic Adapter)	1981.	640x200	T	16	8X8	-	60	15.75
		320x200	T	16	8X8	-	60	15.75
		160x200	G	16	-	-	60	15.75
		320x200	G	4	-	-	60	15.75
		640x200	G	2	-	-	60	15.75
MGA (Hercules Monochrome Graphic Adapter)	1982.	720x350	T	1	9X14	MDA	50	18.10
		720x348	G	1		MDA	50	18.10
EGA (Enhanced Graphic Adapter)	1984.	640x350	T	16	8X14	CGA,MDA	60	21.85
		720x350	T	4	9X14	CGA,MDA	60	21.85
		640x350	G	16	-	CGA,MDA	60	21.85
		320x200	G	16	-	CGA,MDA	60	21.85
		640x200	G	16	-	CGA,MDA	60	21.85
PGA (Profess. Graphic Array)	1987.	720x400	T	16	9X16	CGA,EGA	70	31.50
		720x400	T	16	9X16	CGA,EGA	70	31.50
		360x400	T	16	9X16	CGA,EGA	70	31.50
		640x400	G	16	-	CGA,EGA	60	31.50
		640x480	G	2	-	CGA,EGA	60	31.50
		320x200	G	256	-	CGA,EGA	70	31.50
MCGA (Memory Controller Gate Array)	1987.	320x400	T	4	8X16	CGA,EGA	70	31.50
		640x400	T	2	8X16	CGA,EGA	70	31.50
		640x400	G	2	-	CGA,EGA	60	31.50
		320x200	G	256	-	CGA,EGA	70	31.50
SUPER VGA (VESA Standard)	1989.	800x600	G	16	-	VGA,CGA,EGA	56, 60,72	35,37,6,48

GRAFIČKE STANICE

Grafičke stanice se koriste za izradu i unos slika i crteža u memoriju ili vanjsku memoriju za složenije grafičke zahtjeve. Imaju dva načina prikazivanja slike na monitoru: *storage* i *refresh*. *Storage* način zadržava sliku, a *refresh* ponavlja sliku brzinom 30-60 puta u sekundi.

Grafičke stanice omogućavaju vektorsko i rastersko prikazivanje slika. Kod vektorskog prikazivanja linije prikazujemo koordinatama tačaka linija, a kod rasterskog prikazivanja prikazujemo sliku nizom tačkica (piksela) kojima definišemo prostor slike. Gustoća ovih točkica tj. piksela predstavlja kvalitet slike koju nazivamo *rezolucijom*.

Komunikacija se obavlja specijalnim grafičkim protokolom putem: *tastature*, *palice (joystick)*, *miša*, *pomične kugle (trackball)*, *grafičke table* i *svjetlosne olovke*.

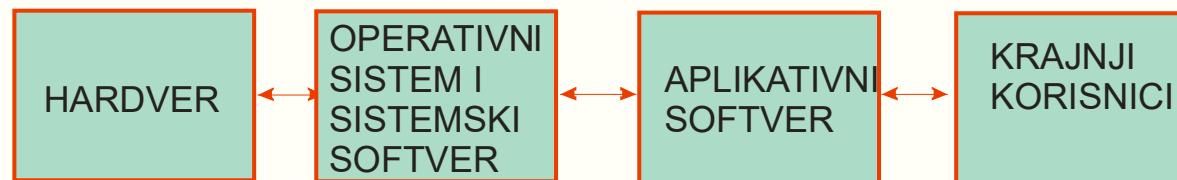
5. GRAFIČKI SOFTVERSki PAKETI I UREĐAJI

OPŠTE

Aplikativne grafički softveri čine programi koji su izrađeni da razriješe mnoge konkretnе probleme u raznim oblastima grafičkog dizajniranja.

Oni služe za izradu i obradu slika na računaru. Pod slikom se podrazumijeva svaki grafički objekt kao što je fotografija , ilustracija, nacrt , grafikon i sl.

Aplikativni programi mogu biti i nezavisni od hardvera, tj. mogu se pisati tako da komuniciraju sa operativnim sistemom, a ne sa hardverom. Na taj način je omogućeno izvršavanje programa na različitim vrstama računara, sve dok je operativni sistem isti.



PROGRAMSKA PODRŠKA PROJEKTOVANJU I DIZAJNIRANJU

Sve što se događa u računaru proizvod je korisničke interakcije s programskom podrškom instaliranom u računaru. Sve što radimo ili planiramo raditi pomoču računara, zavisiće o programskoj podršci koja nam je na raspolaganju.

Također prije nego krenemo u bilo kakve odluke potrebno je znati što želimo raditi pomoču računara i koji nam je krajnji cilj (2D dokumentacija, 3D modeliranje, animacija itd.). Koliko će raditi ljudi, koju organizaciju u sklopu izvođenja projekta očekujemo i sl. Sve to u jednom birou predstavlja proces koji zahtjeva ozbiljan projekt informatizacije. Ovdje nije cilj tako detaljno obuhvatiti ovaj problem i zato čemo se orijentisati samo na prikaz *programske podrške* koja se u sklopu izrade dokumentacije može koristiti u projektnim biroima.

Kod izbora programskih paketa važno je obratiti pažnju na neke momente koji nam u dalnjem korištenju mogu olakšati ili zakomplikovati daljnji rad.

Početak odlučivanja treba bazirati na provjerjenim i ispitanim programima kod zastupnika programskih paketa. Kod njih treba provjeriti da li postoje sve mogućnosti koje su nam potrebne od programa. Jedan od momenta u odlučivanju su saradnici s kojima radimo. Ukoliko oni rade s nekim od programske alata, dobro je provjeriti kako ti alati mogu razmjenjivati podatke s programom s kojim smo odlučili raditi u birou.

Također je dobro uzeti u obzir prethodno znanje osoba koje trebaju raditi na računaru.

Programe o kojima treba razmišljati i koji će se sigurno koristiti u projektnim biroima u inženjerskom domenu možemo podijeliti u nekoliko grupa:

- a) *operativni sistem i grafički interface* su nezaobilazni dio programske podrške svakog računara,
- b) *programi za vođenje poslova u birou*, koji predstavlja osnovne alate za unos i obradu teksta, osnovne tablične kalkulacija, i vođenje baze podataka,
- c) *programi poznati kao CAD (Computer Aided Design)*, programi za dvodimenzionalno tehničko crtanje i trodimenzionalno inženjersko modeliranje sa mogućnostima osnovne prostorne vizualizacije biti će korišteni u većini projektnih biroa,
- d) *programi za 3D modeliranje*, koji kao nadogradnja i u uskoj vezi s CAD programima omogućuju modeliranje na bazi elementarnih tijela (solid modelling), doradu modela do fotorealističnog kvaliteta i animacije, te na taj način nude vrlo kvalitetnu prezentaciju projekta,
- e) *programi za crtanje i grafički dizajn (CorelDraw, Designer, Freehand, Illustrator), obradu slika (Photoshop, Photopaint, Picture Publisher), različite grafičke efekte i stolno izdalaštvo i pripremu materijala za štamparske tehnike*, je grupa programa koji će nam omogućiti da se dodatno pripremi i upotpuni dokumentacija, te pripremi za ispis i po potrebi za štampu
- f) *programi za razne proračune i izradu troškovnika*, koji su danas djelomično integrirani u CAD programe, ali u posebnim slučajevima mogu biti osnova rada u birou koji se bavi upravo tim poslovima.

Svaka od ovih grupa programa karakteristična je za određeni segment rada na računaru, a u svakoj grupi programa ističu se pojedini paketi kao standard u korištenju, što olakšava razmjenu podataka s radnom okolinom.

OPERATIVNI SISTEM I GRAFIČKI INTERFACE

Operativni sistem je grupa kompjuterskih programa koji sadrži instrukcije koje povezuju komponente u računaru i usklađuju njihov rad, što uključuje rad s memorijom, sistemske pozive i uputstva procesoru, rad s periferijama, kontrolu eksternih memorija (diskova). Izbor operativnog sistema znači definisanje kompjuterske platforme koja će se koristiti i čitavog sklopa programske podrške koja postoji za taj sistem.

Budući da su operativni sistemi vezani za računar, najvažnije je odabratи *primarni grafički program* i provjeriti na kojem/kojim operativnim sistemima radi.

Između više sistema koji danas postoje dominantni su ***Microsoft Windows, Macintosh, System 7, DOS, Unix itd.. Apple***. *Apple* se prvi pojavio sa operativnim sistemom koji je posjedovao grafički način komunikacije između korisnika i računara i grafički interface. Lakoča takve komunikacije navela je neke kompanije da stvore grafička okruženja za *Unix* operativni sistem, jedan od prvih je ***X-Windows***, ali i za ***DOS*** pod nazivom *Windows*. Budući je ovime tehnika korištenja različitih računara postala slična mnogi su se aplikativni programi u posljednje vrijeme počeli prilagođavati različitim platformama omogućujući im što širi krug korisnika. Prednost ovih promjena je u tom što su se smanjile barijere između operativnih sistema i omogučilo lakše prilagođavanje i rad na različitim računarima.

Dinamičnim razvojem *Windows* grafičkog interfacea, Microsoft je znatno uzdrmao primat *Apple* računara kao grafičkih računara.

Korištenje grafičkog okruženja *Windows*, ima niz prednosti u odnosu na druge operativne sisteme. Pored jednostavnijeg rada u odnosu na "goli" *DOS* npr. i korištenjem "drag and drop, tj. vuci i ispusti" tehnologije, *Windowsi* omogućuju i multi-tasking, odnosno rad sa više programa u isto vrijeme. To se postiže otvaranjem prozora na ekranu (otuda dolazi ime *Windows*) koji su promjenjive veličine tako da se različite aplikacije mogu koristiti paralelno ovisno koji je prozor aktiviran. Tako se može u isto vrijeme pisati tekst, crtati shemu ili tlocrt, organizirati svoje datoteke i direktorije, pratiti vrijeme na satu itd.

PROGRAMI ZA ADMINISTRACIJU I VOĐENJE POSLOVA PRI IZVOĐENJU PROJEKTOVANJA I DIZAJNIRANJA

Ovi programi su u velikom dijelu zajednički za većinu poslova tako da ih nije potrebno posebno obrazlagati. Oni su karakteristični u toliko što su nam neophodni u svakodnevnom radu u birou, te s njima radimo osnovne poslove unosa teksta, ispisivanja, popunjavanja raznih obrazaca, prezentiranja informacija kolegama i sl.

Ovu grupu programa možemo podijeliti na osnovne programe, a to su:

- a) programi za obradu teksta,
- b) programi za tablično računanje
- c) programi za baze podataka
- d) programi za poslovne prezentacije
- e) programi za planiranje i organizaciju rada i personalne informacije

Iako svaki od ovih programa postoji kao zasebna aplikacija, i razni proizvođači nude razna programska rješenja, za Windows operativni sistem *Microsoft* je ponudion integrисани paket pod nazivom ***Microsoft Office***, koji sadrži većinu gore spomenutih programa, i predstavlja idealno rješenje za većinu "običnih" poslova u birou, iako svojim mogućnostima nudi zaista impresivna rješenja, pogotovo kada je više računara vezano u lokalnu mrežu.

PROGRAMI ZA PROJEKTOVANJE I 3D MODELIRANJE

Ovo su programi koji su u središtu interesa raznih inženjerskih struka (mašinski, građevinski, grafički inženjer itd. te arhitekti i slični) kada se govori o primjeni računara u njima. Pošto svi ovi programi imaju zahtjev sa snažnim resursima, na osnovu ovih programa definišu se zahtjevi prema **hardwareu**. Njih ćemo također podijeliti u nekoliko grupa radi bolje preglednosti:

- a) programi za tehničko crtanje, projektovanje i modeliranje - CAD programi
- b) programi za vizualizaciju - modeliranje, rendering i animaciju
- c) programi za urbanističko projektiranje i prostorno planiranje u djelokrugu urbanizma – G/S
- d) programi za dinamičke simulacije - FEM, energetske simulacije
- e) programi za različite proračune i troškovnike
- f) programi za analize pri projektiranju - ekspertni sistemi i neuronske mreže

Mnogi tipova programa nisu uopšte strogo vezani za pojedinu inženjersku oblast, te je realno preklapanje mogućnosti koje nude pojedini programi. Zato će se za svaku navedenu grupaciju dati osnovne smjernice korištenja, ali ozbiljan rez među kategorijama programa teško možemo odrediti.

CAD programi

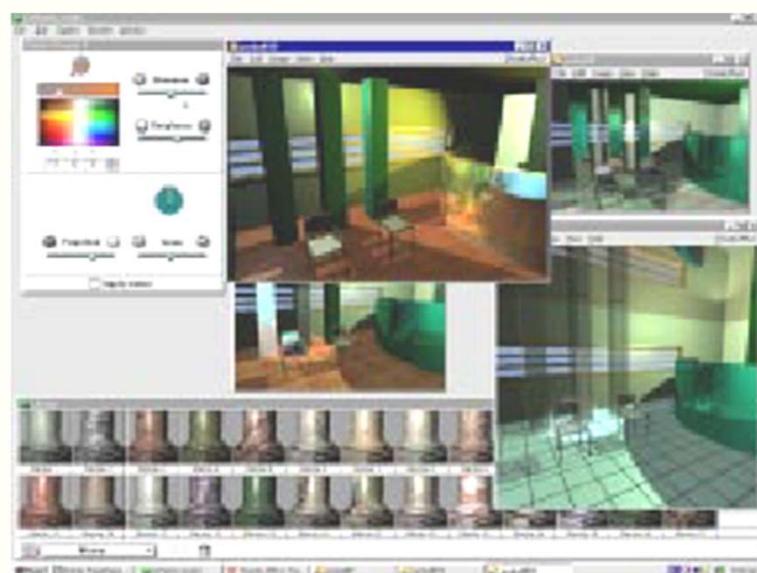
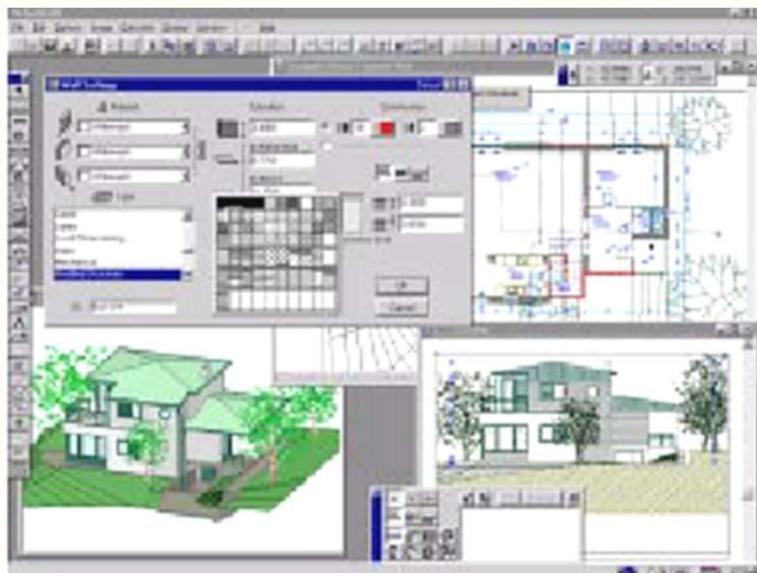
Ova grupa programa osnovna je za bilo koji biro koji se bavi projektiranjem. Karakteristično za CAD programe je da se mogu primjeniti na tzv. "2D drafting" odnosno dvodimenzionalno tehničko crtanje i 3D modeliranje, većina programa podržava dodjelu materijala i renderiranje. Međutim u posljednje vrijeme pojedini dodatni programi koji su se zasebno prodavali profilirali su posebne programe za određena područja primjene.

Danas dvodimenzionalno crtanje predstavlja podskup mogućnosti koje nude programi za 3D modeliranje. No ako su nečiji zahtjevi isključivo orijentisani na izradi tehničke dokumentacije tada će vam 2D CAD programi biti dovoljni, a investicija manja.

Da navedemo samo neke od ovih programa koji su i osnova za današnje mnogobrojne moderne CAD aplikacije: **Minicad**, **TopCAD**, **ClarisCAD**, **Vellum** postoje za Maca, a za PC-e **Autosketch**, **Drafix**, **Vellum**, **CorelCAD** za Windows. Verzije Autodesk AutoCADA, AutoCAD12 i ostale Auto CAD verzije, Bently Microstationa, postoje za sve tri platforme, Graphisoft ArchiCAD za MAC i Windows okruženje, dok Architron postoji samo za Maca, a Arris, Star i Sonata samo za Unix.

Iako postoji određeni trend prema specijalizaciji za određena područja, u području npr. arhitekture možemo razlikovati dva osnovna koncepta i njihove predvodnike: **Graphisoft ArchiCAD** i **Autodesk AutoCAD**.

ArchiCAD firme Graphisoft je program isključivo namjenjen arhitektonskom projektiranju. U njemu umjesto alata sa znakom linije stoji alat sa znakom zida, međspratne konstrukcije, vrata, prozora, krova itd. Ako želite nacrtati vanjski zid nećete morati vući nekoliko paralelnih razmakačnih linija, već će vam se pojaviti podmeniji sa karakteristikama zida koje ćete izabrati



Grafičke izvedbe u ArchiCAD aplikaciji

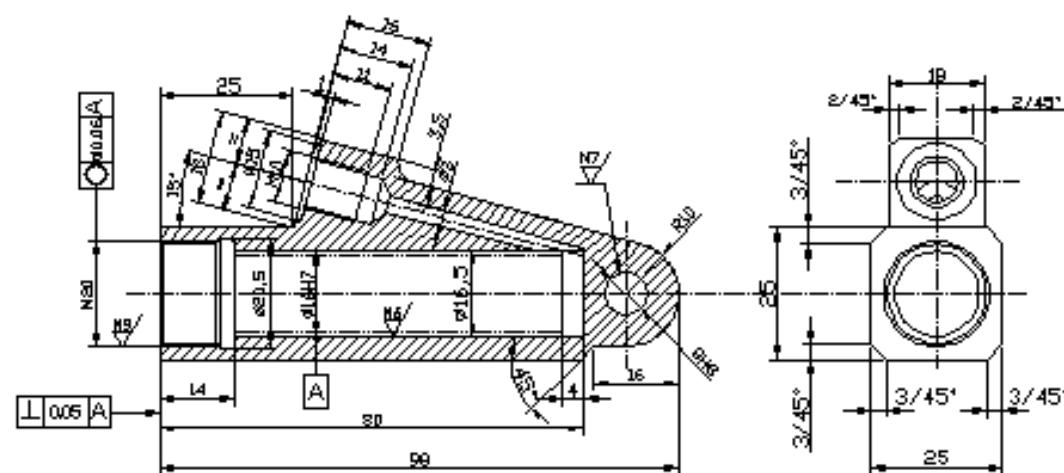
AutoCAD i njemu slični programi nudi drugi pristup. To je program za tehničko crtanje i projektovanje odnosno modeliranje opšte namjene. Modeli se grade od linija tako da im se definiše visina (“*polylines sa thicknessom*”) ili da se likove zatvorene linijama na žičanom modelu pretvori u plohe uz pomoć 3D faceova (*poligona*). Crtačka fleksibilnost je veća ali je i složenost korištenja i dužina savladavanja veća. Kad ste izgradili model i zatvorili plohe objekt možete pogledati u aksonometrijskom ili perspektivnom prikazu kao žičani model, model sa skrivenim nevidljivim linijama ili jednostavno osjenčeni model. Tada se često uvidi da sve plohe nisu zatvorene i da su nužne korekcije.

Kako bi se pojednostavnilo crtanje jednostavnih tijela ponuđen je opcionalno i dodatni modul nazvan *AME (Advanced Modelling Extension)* za oblikovanje modela uz pomoć primitivnih tijela sa kojima se mogu vršiti prodori, zasjecanja, unije, računanje volumena, težišta itd.). U novim varijantama ovog programa dodane su nove mogućnosti za 3D solid modeling.

Da bi pojednostavio crtački posao raznim profilima inženjera, Autodesk, proizvođač AutoCAD-a, nudi i razne programske pakete. Takav je npr. AEC koji se nadograđuje na AutoCAD i koristi inženjerima kao poseban program za crtanje, prostorno modeliranje i vizualizaciju.

AutoCAD se približio načinu i logici rada projektanata. Na tržište je sredinom 1993 izašao AutoCAD 12 za Windows okruženje. Pored veće brzine u radu on donosi niz novih opcija u AutoCAD-u standardnih za Windows. Dalje se razvija AutoCAD 14....., AutoCAD 2000 itd.

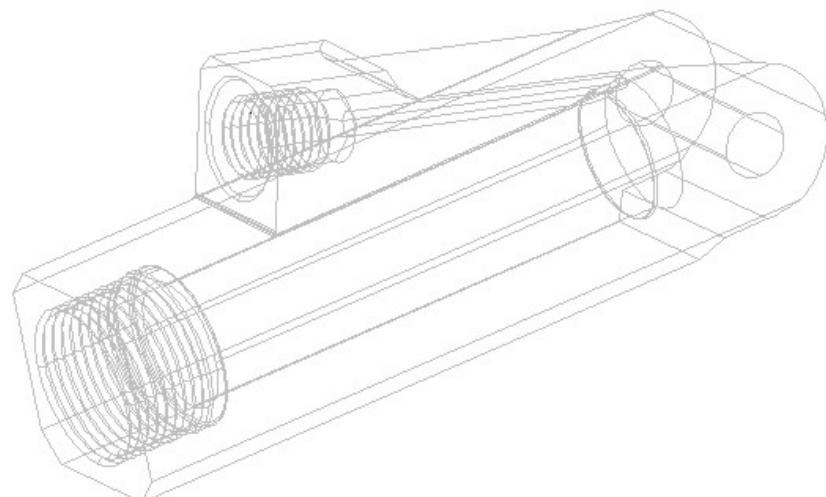
#8H8	+0.022
	0
#16H7	+0.018
	0



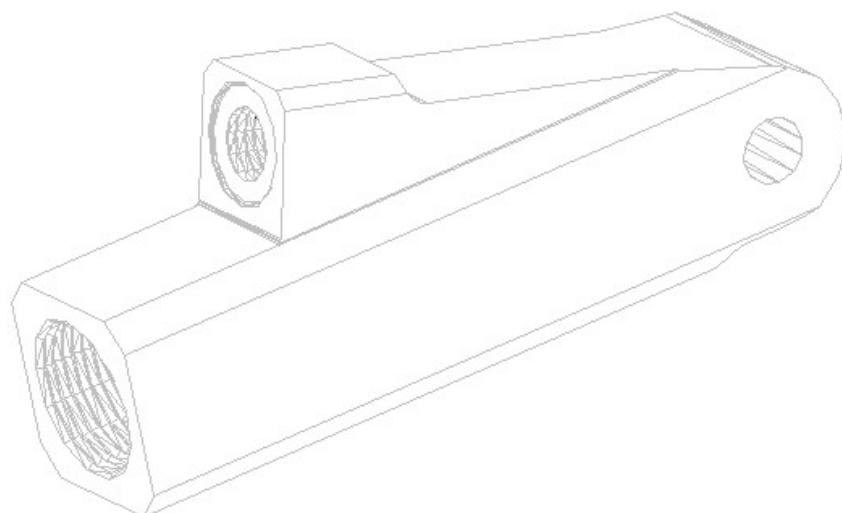
NAPOMENA

- fuchs sind die alten -

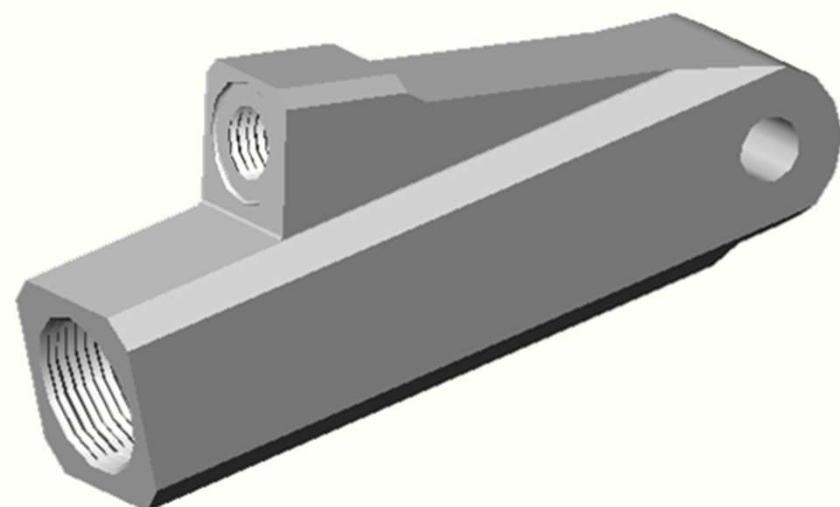
Tehničko crtanje i projektovanje odnosno modeliranje opšte namjene ~2D prikaz konstruktivne razrade tijela cilindra~



Žičani model tijela cilindra



Model skrivenih linija tijela cilindra



Osjenčeni model tijela cilindra

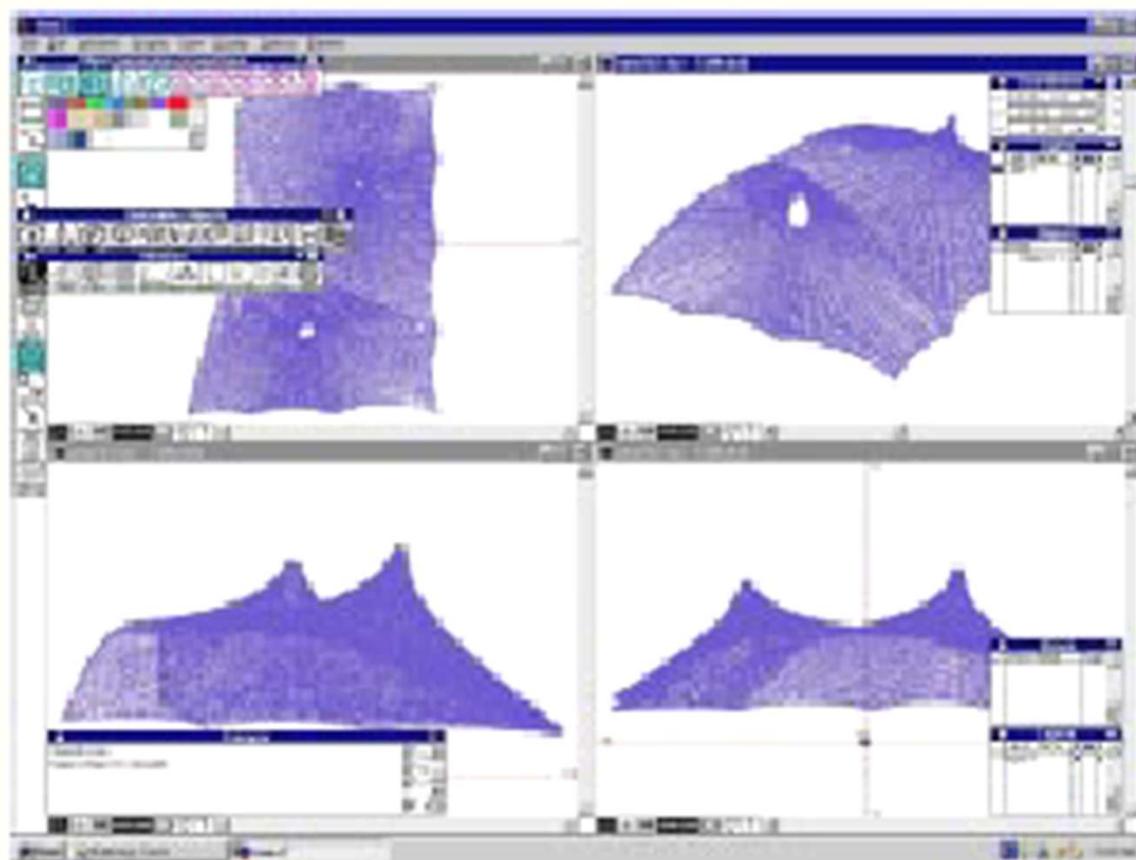
Programi za konceptualno modeliranje

Proces projektiranja sastoji se od više faza u kojima su zahtjevi za grafičkim mogućnostima računara, naravno ako ih uopšte postavljate, različiti. U konceptualnoj fazi projektant nema potrebe za detaljnim kotiranjem ili šrafiranjem crteža. Crtež, odnosno skica, služi za provjeru nekih osnovnih gabaritnih i volumenskih rješenja.

Popularna slika većine projektanata u toj početnoj fazi rada je kako uglavnom crta prostom rukom po papiru.

Upravo za ovu fazu rada određene softwareske kuće namjenile su programe za konceptualno trodimenzionalno modeliranje kao npr., softwareska kuća *Alias* ima program ***Upfront***, Autodeskov je program ***3D Concept***, a popularan je i program ***FormZ*** i mnogi drugi novijih generacija.

Mogućnosti ovih programa nisu male. Bitna njihova karakteristika je da omogućuju jednostavno modeliranje u tri dimenzije, korištenje paleta za dodjelu boje tijelima, jednostavno definisanje perspektivnih pogleda, mogućnost osjenčanja tijela ali i bacanja sjene, koja se može definisati preko geografske širine, doba dana i godine. Programi dozvoljavaju (pogotovo *FormZ*) postavu dodatnih izvora osvjetljenja, a pogodni su za izradu fotomontaže. Modele sagrađene uz pomoć ovih programa moguće je prebaciti i učitati u složene *CAD* programe preko *DXF* formata. Koriste se na mnogim školama, visokoškolskim ustanovama i u biroima.



Izgled konceptualnog prikaza

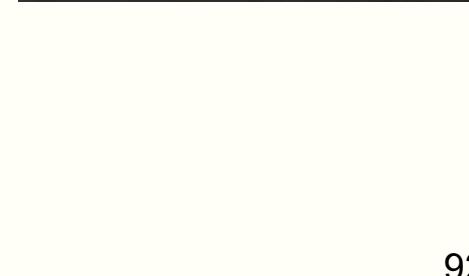
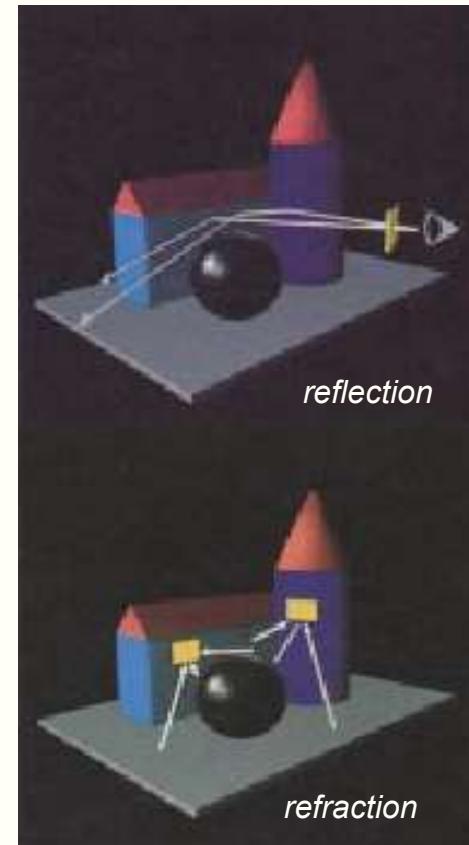
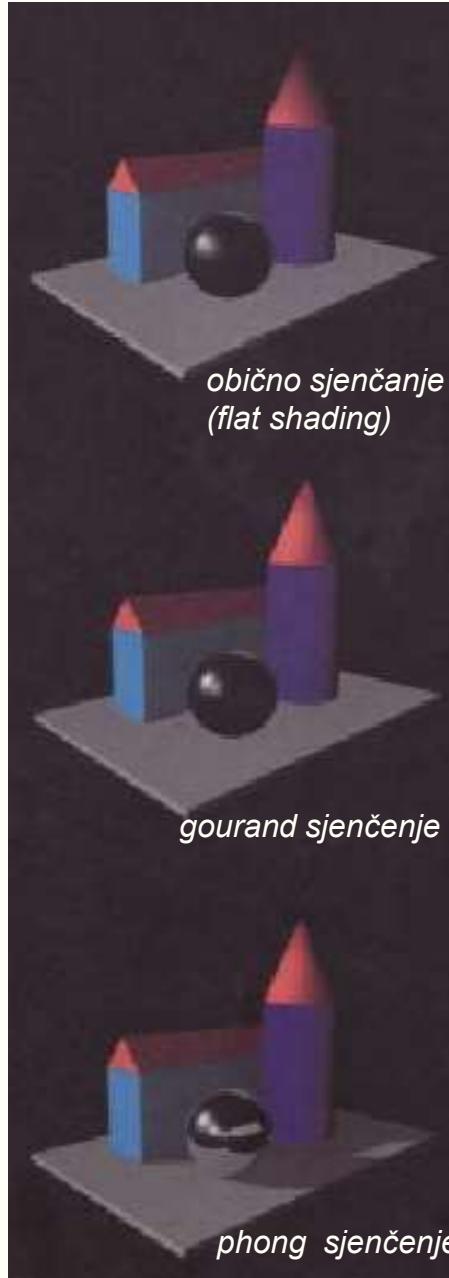
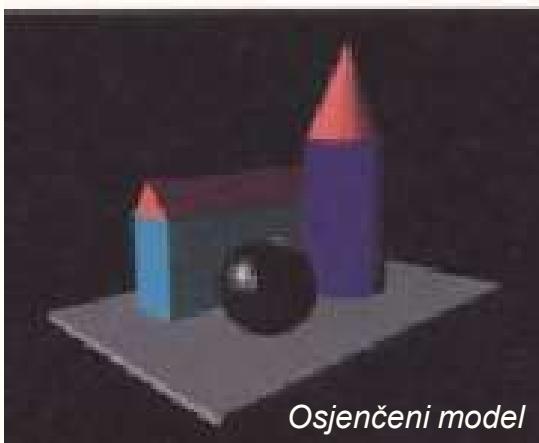
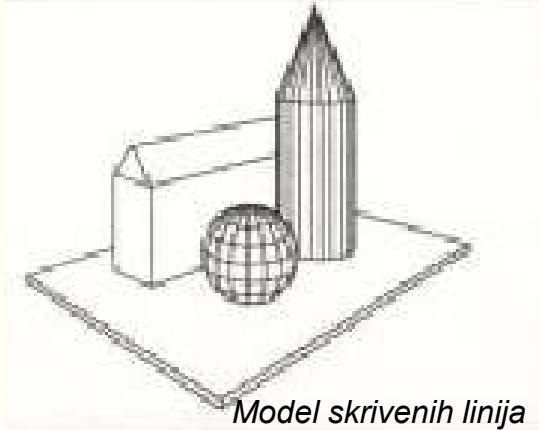
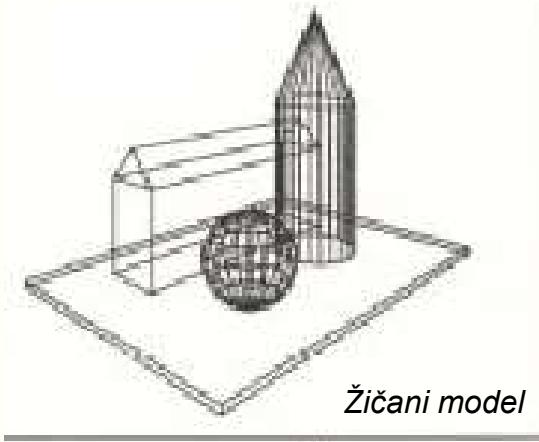
Modeliranje, renderiranje i animacija

Sve je veći interes projektanata koji dijelove svojih projekata modeliraju na računaru u tri dimenzije, da te modele dodatno obrade, odnosno renderiraju, dodajući plohamu atributu boje, teksture, refleksije, prozirnosti i određujući višestruke izvore svjetla, težeći tako ka fotorealističnom prikazu svojeg projekta.

Programa koji pokrivaju ovo područje ima vrlo mnogo. Mnogi su se razvili na platformama *MAC* i *Amiga*, ali i *PC-a*, a trenutno većina postoji za različite platforme, a svi postoje za *Windows* okruženje.

Prije nego nešto kažemo o programima za renderiranje, moramo nešto reći o samom terminu **renderiranja**, i tehnikama koje razni programi koriste i osnovnim terminima koji se u tim programima koriste.

U svim *CAD* paketima, kako je naprijed rečeno, postoje, barem dva, ali i tri osnovna prikaza *3D* modela (detaljnije o njima bit će rečeno u kasnijem izlaganju). Prvi i najjednostavniji je **žičani model**. Kao i pri aksonometrijama i perspektivama urađenim rukom, računar može izračunati model, koji je prikazan od linija. Ukoliko tom prozirnom žičanom modelu, zadamo da sakrije nevidljive linije, dobivamo novi prikaz pod tim nazivom, tj. model **sakrivenih linija**. Treći nivo prikaza također je postao standardan u svim modelima, ali i *CAD* paketima. To je tzv. **osjenačni model**, a u programima ćete to pronaći pod nazivom "*shading model*". Taj nivo prikazuje model kao obojeni puni model. Pri tome određenim plohamu možemo mijenjati karakteristiku boje. Ti modeli iako se zovu osjenčani ne moraju bacati sjene, ali većina programa ima i tu mogućnost. Sve do ovog nivoa je relativno jednostavno. Međutim svi programi za modeliranje, rendering i animaciju, nude mnogo više.



Osim karakteristike boje, svi programi nude i tri osnovne karakteristike ponašanja materijala, koja je u načelu definisana bojom. To su:

- ambientalna* karakteristika (kakva će biti boja u dijelu koji nije direktno osvjetljen)
- difuznu* karakteristiku (odnosi se na osvjetljeni dio plohe, tj. koja će biti boja na plohi tamo gdje je najveći udar svjetla “*highlite*”.) i
- *Intenzitet boje* koji postoji za svaku karakteristiku boje.

Svako od navedenih osobina je različita za materijale koji nas okružuju i dovoljne su za sve naprednije modele renderiranja.

Kako bi smo se upoznali sa renderiranjem kao osnovno potrebno je poznavati sjenčenje. Osnovni model sjenčenja je obično sjenčanje (*flat shading*) u kojima zakrivljene plohe nisu zaobljene nego su napravljene od ravnih ploha, a postoji i *phong* (kontrast iz pozadine i blaga površina) i *gourand* (*linearna interpolacija boja ili sjenka kroz poligon*) model sjenčanja gdje svaki od njih uvažava više karakteristika pri računanju modela.

Neki od programa ili standardno ili kao dodatak nude metal model koji nam omogućju da vrlo kvalitetno dobivamo *metallic* karakteristike materijala.

Uz ove osnovne karakteristike materijala postoje i druge karakteristike kao što su npr. *prozirnost* (*transparency*). Međutim niti svi prozirni materijali se ne ponašaju jednako, jer postoji različita karakteristika loma svijetlosti (*refraction*) i refleksije (*reflection*).

Osim već spomenutih fizikalnih karakteristika materijala, postoje i drugi važni opisi materijala. Jedan od važnijih je *tekstura*, za npr. drvo, kamen i slične materijale, za koje postoji nekoliko modela preslikavanja tekture na plohu, ovisno o obliku plohe, npr, ravninski, kubusni, sferni, i sl. Jedan od opisa plohe je i tzv. *bump* odnosno *reljefnost* plohe, koji je važan kod realističnih prikaza.

Nadalje, pored karakteristika materijala, postoje i karakteristike *osvjetljenje* i *utjecaj atmosfere*. Kod svjetla možemo spomenuti osnovne tipove, *kuglasto*, *paralelno*, i *reflektorsko* svjetlo.

Poznavajući naprijed navedene karakteristike moguće je sada i objasniti pojam *renderinga*. *To je proces računanja prikaza 3D modela bojenjem ploha prema karakteristikama materijala u vidu, boje, prozirnosti, grubosti, refleksije, refrakcije, teksture, reljefnosti i ostalih karakteristika, osvjetljavanjem pojedinim izvorima svjetlosti.*

Nije se lako snaći među ovim programima kojih je na tržištu sve veći broj a čije se mogućnosti i cijene znatno razlikuju. Među njima postoji podjela na *low-end* programe za personalne platforme (**Lightwave**, **3Dmax/3Dstudio**, **Stratavision**, **Infini D**, **Ray Dream Designer**, **Sculpt 4D**, **Imagine**) i *high-end* programe za radne stanice (programi firmi **Alias**, **Softimage**, **Wavefront** i **ElectroGIG**).

Svi navedeni programi posjeduju vlastiti modul za 3D modeliranje koji grade model od primitivnih geometrijskih i rotacionih tijela. Kako je obavezni standard kod svih programa za rendering filter za *DXF* fileove oni mogu očitati modele načinjene u *CAD* programima.

PROGRAMI ZA CRTANJE, OBRADU SLIKA, GRAFIČKI DIZAJN I STOLNO IZDALAŠTVO

Promjene koje je donijela kompjutorizacija ogleda se u brisanju tradicionalnih granica među profesijama tako da kada su već savladali programe za pisanje teksta i crtanje, projektantima i dizajnerima nije bilo teško napraviti dodatni napor i ovladati programima za stolno izdalaštvo koji im nude da svoje elaborate dovedu na grafički nivo pravih publikacija.

Programe za stolno izdalaštvo, obradu slike i grafički dizajn možemo podjeliti u nekoliko osnovnih grupa:

- programi za grafički dizan, bazirani na rasterskoj grafici
- programi za grafički dizajn, bazirani na vektorskoj grafici
- programi za obradu slika i fotografija
- programi za posebne efekte
- programi za prijelom i stolno izdalaštvo
- uslužni programi, za konverzije, skeniranje i sl.

Kako je osnovna podjela grafičkih programa na one za **vektorskú** i **rasterskú** grafiku to je bitno pobliže poznavati ova dva pojma, mada će o njima kasnije biti detaljno riječi.

- **Rastersku grafiku** (bitmap) koriste tzv. Paint programi poput **Paintbrusha**, **MacPainta**, **Paintera** i drugih. U njima se crtež stvara "aktiviranjem" pixela u rasteru kojima se može dodjeliti boja. Ova tehnika ima prednosti i mana u odnosu na vektorsku grafiku, a sličnija je crtaju olovkom po papiru. Mnogi programi su izuzetno sofisticirani programski alati koji pružaju korisniku simulaciju crtanja sa velikim brojem slikarskih tehnika, odabirom različitih podloga za crtanje i nizom dodatnih efekata. Novi slojevi boje ne prekrivaju prethodni, kao u standardnim paint programima, već sa njime stvaraju novu boju vrlo slično tradicionalnim slikarskim tehnikama.
- **Vektorska grafika** definisana je koordinatama. Linija je crta između dvije tačke, likovi su omeđeni linijama. Prikaze gradimo od osnovnih likova. Kad izaberemo neki lik za obrisati moramo brisati čitav osnovni lik. Ne možemo brisati samo njegov dio, osim ako za to ne postoje alati. Vektorska grafika, osim što je preciznija i brža za crtanje, pruža daleko veće mogućnosti u manipulisanju likovima i tijelima, te primjeni efekata koje nije moguće izvesti sa rasterskom grafikom. Sa trodimenzionalnom vektorskog grafikom moguće je izračunavanjem koordinata dobivati perspektivne prikaze. Nove generacije programa poput serije iz programa **Corel-a**, ili **Fractal Design Expressiona** smanjuju granicu između rasterske i vektorske grafike. **Corel** ima mnogo programskih mogućnosti za rastersku grafiku unutar vektorske strukture samog programa, dok **Expression** koristi vektorskog grafiku kao bazu za rasterske efekte. Na tim se temeljima baziraju današnji programi za grafički dizajn.

Programi za rastersku grafiku

U programe bazirane na rasterskoj grafici spadaju:

programi za **crtanje** (npr. Canvasa),
-programi za **posebne ilustracije** (npr. *Fractal Design Paintera*) i
-programi za **obradu fotografije** (image processing). Tu se svakako ističe **Adobe PhotoShop**, **Corel PhotoPaint**, **Macromedia XRES**, i drugi. Uz pomoć njih se obrađuju fotografije unesene preko skenera u kompjutor (kao raster pixela) ili izgenerisani fotorealistični prikazi koji su spremljeni u programima za vektorsku grafiku kao rasterski prikazi. Moguće su korekcije svjetloće i kontrasta, boje, specijalni dvodimenzionalni i trodimenzionalni efekti, retuširanje i docrtavanje. Programi se mogu koristiti za retuširanje, za pripremu pozadine kod fotomontaže i za dorađivanje fotorealističnih prikaza koji su dobiveni programima za vizualizaciju trodimenzionalnih kompjutorskih modela.

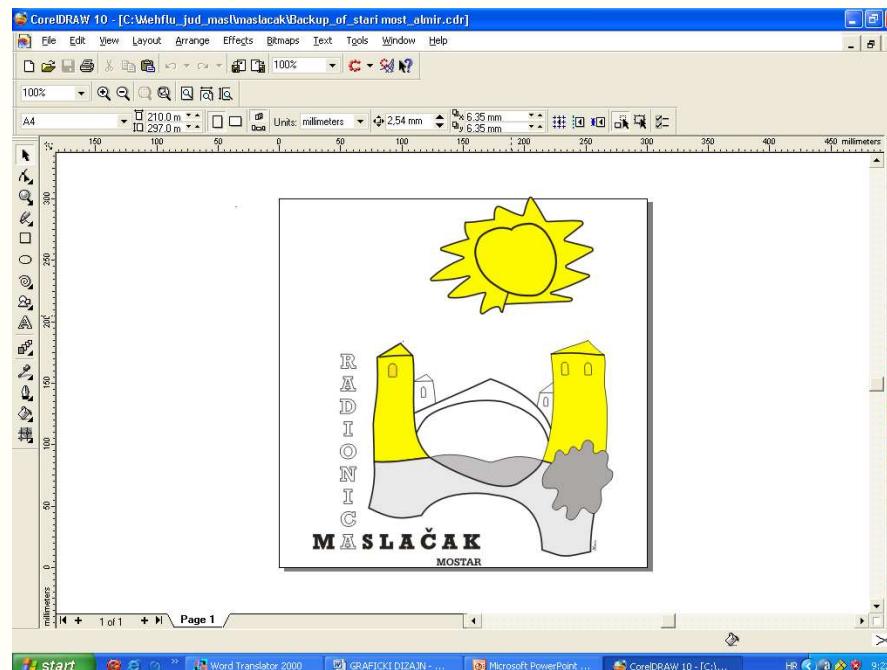
Pored navedenih programa postoje i programi odnosno potprogrami (*plugin*) za izvođenje različitih efekata na prikazima. *Adobe Gallery Effects* npr. omogućuje nam da na fotografiju primjenimo **slikarske efekte** koji će simulirati kao da je prikaz načinjen vodenim bojama, ugljenom, kredom, grafičkim perom, da izgleda poput mozaika itd. Postoji veliki broj “*pluginova*” za programe poput *Photoshopa*, *Xresa*, *Paintera* i drugih.

Kao posebni programi koji koriste rastersku tehnologiju postoje i programi za **morphing (metamorfoze)** koji nam omogućuju da uzmemo dva različita prikaza i da na njima odredimo korespondirajuće tačke ili linije na temelju kojih će računar izgenerirati zadani broj pretapanja tj. metamorfoza. Rezultati mogu biti vrlo efektni a mogu se koristiti kao pojedinačni prikazi ili kao kratke animacije.

Programi za vektorsku grafiku

Programi za grafičke ilustracije, poput **CorelDrawa**, **CorelXare**, **Adobe Illustratora** i **Macromedia Freehanda**, ili **Micrographics Designer**a za Windows odnosno **Adobe Illustratora** i **Macromedia Freehanda** za Apple, da navedemo samo one najpopularnije, i mnogi drugi novijih generacija, baziraju se na dvodimenzionalnoj vektorskoj grafici a nude mnogobrojne mogućnosti za stvaranje grafičkih ilustracija, crteža, efektnih naslova i sl. koji se mogu prenjeti u **DTP (Desk Top Publishing)** program, tj program za štampu.

Mogućnost da se izabere neki od hiljade fontova ili nacrta neki crtež i da se sa njima rade različite manipulacije (povećavanje, upisivanje teksta po nekoj nepravilnoj liniji, davanje dubine ili stvaranje perspektive, itd.) koje su gotovo neizvedive tradicionalnim načinom, su brojne i jednostavne za izvesti.



Corel draw

Nove varijante programa nude i razne trodimenzionalne efekte..

Programi za stolno izdalaštvo

Programi za stolno izdalaštvo (**Adobe PageMaker**, **Quark Xpress**, **Corel Ventura**, **MsPublisher itd.**) omogućuju veliku fleksibilnost u sređivanju i kombinovanju tekstualnih podataka i ilustracija (crteža, fotografija, table, grafovi i sl.).

Razmaci među slovima, riječima, redovima ili paragrafima se mogu precizno definisati, tekst se može rasporediti oko slike i/ili u različite stupce, a prisutne su funkcije za niz drugih podešavanja i oblikovanja publikacije.

Važno je reći da uz manje ili više problema može se kompletna priprema za štampu brošure ili kataloga napraviti na računarima unutar biroa pomoću već spomenutih programskih alata.



Adobe PageMaker

6. INTERAKTIVNA RAČUNARSKA GRAFIKA

Pojam interaktivnosti u računarskoj grafici podrazumijeva **interakciju između korisnika i sistema** na način da korisnik upravlja sadržajem, strukturom i pojmom objekta i njegovih predočenih slika upotrebom ulaznih uređaja (tastatura, miš, zaslon osjetljiv na dodir...).

KONCEPT INTERAKTIVNE RAČUNARSKE GRAFIKE

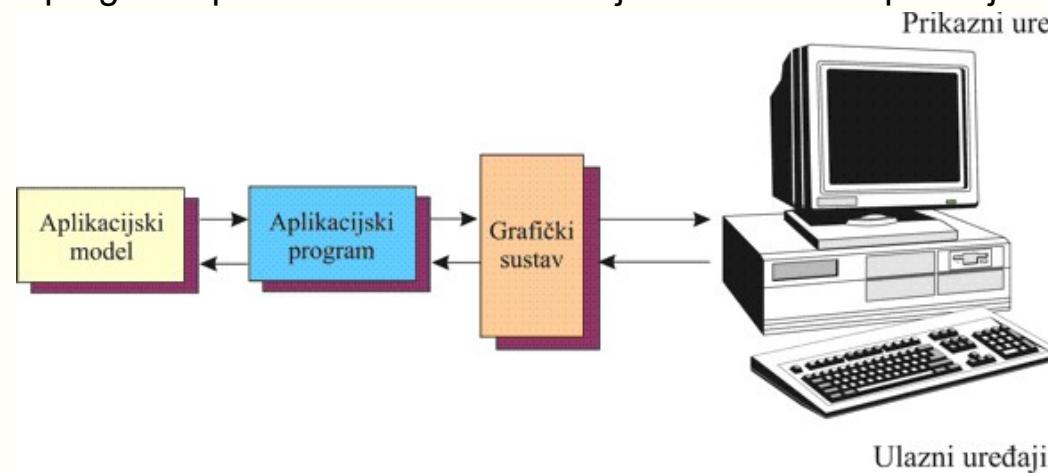
Koncept interaktivne računarske grafike podrazumijeva dva nivoa: **sklopovalski i programski** (softverski).

Sklopovalski nivo koncepta interaktivne računarske grafike može se sažeto opisati na sljedeći način:

- **računar** prima ulazne informacije od **interakcijskih uređaja** i prenosi slike **prikaznom uređaju**.

Programski nivo koncepta interaktivne računarske grafike može se sažeto opisati na sljedeći način:

- **aplikacijski model** predstavlja podatke ili objekte koji se prikazuju na zaslonu;
- **aplikacijski program** prima i obrađuje informacije od aplikacijskog modela i korisnika, razvija aplikacijski model, generira skup grafičkih izlaznih naredbi koje sadrže detaljan geometrijski opis onoga što treba prikazati kao i načina na koji se pojedini objekti prikazuju;
- **grafički sistem** proizvodi sliku na temelju detaljnog opisa u obliku grafičkih naredbi koji generira aplikacijski program i prenosi ulaznu informaciju od korisnika aplikacijskom programu na obradu.



Programski nivo koncepta interaktivne računarske grafike

➤ Aplikacijski model

Funkcije aplikacijskog modela su sljedeće:

- aplikacijski model sadrži sve podatke, objekte i odnose među njima koje koriste prikazni i interakcijski dio aplikacijskog programa ili negrafički moduli za obradu podataka;
- aplikacijski model predstavlja objekte kombinacijom podataka i proceduralnih opisa neovisnih o prikaznom uređaju.

Aplikacijski model sadrži sljedeće:

- primitivne oblike (tačka, crta, višeugaoni likovi u 2D ili 3D, različitih prostornih ploha u 3D...) od kojih je sastavljen model objekta,
- atribut objekata (vrsta linije, boja, struktura površine...),
- odnose među objektima i dijelovima objekata (povezivanje, spajanje...),
- podatke o položaju objekata i dijelova objekata.

Aplikacijski model sadrži sljedeće vrste podataka:

- geometrijske podatke,
- negeometrijske podatke (tekstualni i brojčani podatci).

Podatci u aplikacijskom modelu mogu biti organizirani kao:

- jednostavni niz podataka o koordinatama točaka,
- povezane liste koje predstavljaju umrežene strukture podataka,
- relacijske baze podataka.

➤ Aplikacijski program

Aplikacijski program pretvara opis dijela modela koji treba prikazati u pozive procedura ili naredbi grafičkog sistema koji se koristi za stvaranje slikovnog prikaza u dva koraka:

1. pretraživanje aplikacijske baze podataka i odabiranje podataka nužnih za prikaz odabranog dijela modela,
2. pretvaranje podataka u format prikladan za ulaz u grafički sistem.

U slučaju da aplikacijski model sadrži geometrijske primitivne oblike koji nisu podržani u grafičkom sistemu aplikacijski program ih mora svesti na one koje grafički sistem podržava.

➤ Grafički sistem

Grafički sistem posreduje između aplikacijskog programa i prikaznog uređaja. Zadaci grafičkog sistema su:

- izlazna transformacija (transformira objekt u aplikacijskom modelu u slikovni prikaz modela);
- ulazna transformacija (transformira korisničko djelovanje u ulaznu informaciju za aplikacijski program na temelju kojih aplikacijski program djeluje na promjenu modela i/ili slike).

Osnovni zadatak dizajnera interaktivnog grafičkog aplikacijskog programa je specifikiranje podataka i objekata koje treba grafički prikazati i načina odvijanja interakcije između korisnika i aplikacijskog programa s ciljem kreiranja i modificiranja modela i njegove vizualne predodžbe.

Grafički sistem sastoji se od skupa izlaznih **podprograma** koji odgovaraju različitim primitivnim oblicima, atributima i drugim elementima. Ovi podprogrami tvore biblioteku grafičkih podprograma ili grafički paket i mogu se pozivati iz programskih jezika visokog nivoa (*C, Pascal, LISP*).

UPRAVLJANJE INTERAKCIJOM

Logički ulazni uređaji omogućavaju grafičkom programeru da stvarne ulazne uređaje (miš, palica, digitalizator...) tretira kao lokacijski logički uređaj koji generira koordinate lokacije na zaslonu.

Aplikacijski program može od grafičkog sistema zatražiti:

- periodičko uzorkovanje ulaznih uređaja,
- čekanje u određenom stanju na pokretački događaj.

Tipičan oblik interakcije definisane aplikacijskim programom naziva se ***petlja pokretana događajima***

- Petlja predstavlja sekvencijalni automat s konačnim brojem stanja sa središnjim stanjem čekanja.
- Prijelazi u druga stanja uzrokovani korisničkim ulaznim događajima.

Ovaj model interakcije ograničen je na sekvencijalni dijalog između korisnika i sistema (ping-pong model izmjeničnih korisničkih pitanja i odgovora računara).

Složeniji koncepti interakcije uključuju više paralelnih komunikacija.

Pseudokod petlje pokretane događajima može se napisati na sljedeći način:

stvari početni prikaz na temelju aplikacijskog modela!

while(!kraj) { /* korisnik nije odabrao opciju kraj */

omogući izbor naredbenih objekata!

/* program čeka beskonačno na korisnikovo djelovanje */

čekaj korisnikov izbor!

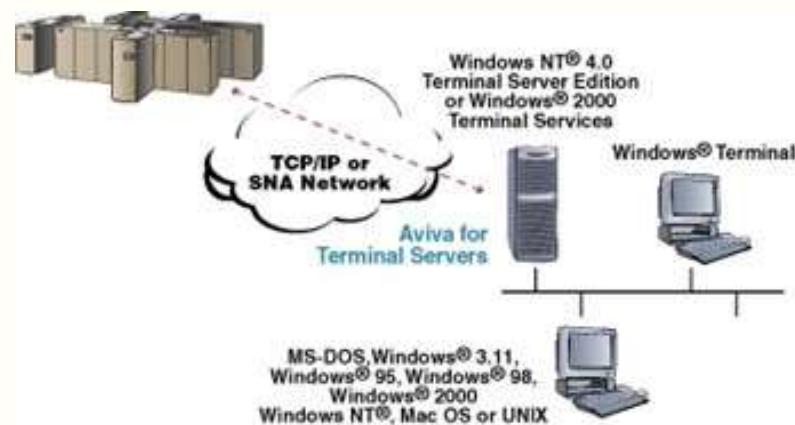
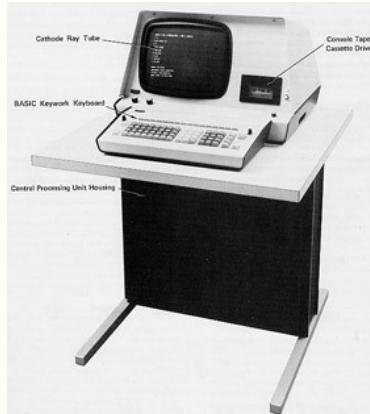
switch(izbor) { obradi izbor i upotpuni naredbu ili izvrši upotpunjenu naredbu

te promijeni model ili prikaz na zaslonu po zahtjevu}

7. GRAFIČKI TERMINAL I GRAFIČKI PROCESOR

GRAFIČKI TERMINAL

- Terminali spadaju u grupu ulazno izlaznih uređaja.
- Oni omogućavaju slanje naredbi na računar koji je negdje drugdje u smislu daljinskog primanja ili davanja podataka, odnosno izvođenja obrada.
- Ostvarivaju tzv. *on-line* veze s računaram.
- Minimalno terminal mora da sadrži tastaturu i monitor.



Obrane se vrše putem većeg broja terminala različitih mogućnosti za interaktivni rad, za rad u distribuiranoj obradi i dr.

Korištenjem terminala obrada se vrši u pravilu sa mesta gdje se neki proizvodni ili poslovni proces obavlja.

Terminali i PC računari koriste se danas najviše od svih jedinica za unos podataka.

Unos se obavlja u interaktivnom radu putem tastature i prikaza dokumenta na display-u u koji unosimo podatke putujući display-em monitora

Kao nositelji podataka koriste se diskete, rjeđe kasete, te vanjske memorije - diskovi i trake.

GRAFIČKI PROCESOR

Grafički prikaz je osnovni oblik prezentacije korisniku te za kvalitetan grafički prikaz nisu dovoljni procesori opšte namjene, pa se krenulo na razvoj specijaliziranih koprocesorskih čipova koji su omogućili obradu ogromnog broja podataka sa svrhom što vjernijeg grafičkog prikaza na display-u monitora. Na temelju njih nastale su razne grafičke kartice, grafički ubrzivači, grafički kontroleri i *grafički procesori*, a niz proizvođača uvidjelo je njihovu isplativost razvoja i prodaje.

Znači, da bi se mikroprocesor oslobodio poslova oko prikazivanja slike, o tome se brini u tu svrhu dizajnirani *grafički procesori* ili ubrzivači kao *S3801*, *TRIO64*, *ATI*, *MATROX* i drugi.

U PC sistemima u pravilu su kao zasebna elektronska kartica umetnuta u jedan od slot-ova, s vlastitom grafičkom memorijom tipa *DRAM* ili *VRAM (VideoRAM)*.

Moguće rješenje je i da grafički procesor koristi resurse radne memorije. Općenito, zadatku kartice je da osigura grafičko interface *GUI* (Graphic User Interface) primjereno programskoj podršci koja se koristi.

Tehnologija koju koriste *grafički procesori* u današnjim karticama napredna je kao i ona u najbržim *Intelovim* procesorima *Pentium* i *AMD-ovim Athlonima*.

Tokom godina, *grafički procesori* su od sistemskog procesora preuzimali sve više zadataka.

Grafički procesori na današnjim karticama mogu efikasno da obrade neverovatno složene *3D* pokrete koji bi zadali muke i najmoćnijem sistemskom procesoru.

8. RASTERSKI I VEKTORSKI GRAFIČKI SISTEMI

GRAFIČKI SISTEMI

Grafički sistema je *sistem koji proizvodi sliku na temelju grafičkih naredbi koji generira aplikacijski program i prenosi ulaznu informaciju od korisnika aplikacijskog programa*).

Razlikujemo dvije osnovne skupine grafičkih sistema

- **vektorski** grafički sistem (sistemi sa proizvoljnom putanjom otklonjene zrake),
- **rasterski** grafički sistemi (sistemi sa sekvencijalnom putanjom otklonjene zrake)

VEKTORSKI GRAFIČKI SISTEMI

Vektorski grafički sistemi razvijeni su još od sredine 60-ih godina i u standardnoj upotrebi bili su do sredine 80-ih godina.

Pojam “**vektor**” ovdje označava liniju.

Linija koja povezuje dvije (prizvoljno) odabrane tačke na display-u osnovni je element grafičkog prikaza.

Dijelovi vektorskog grafičkog sistema su:

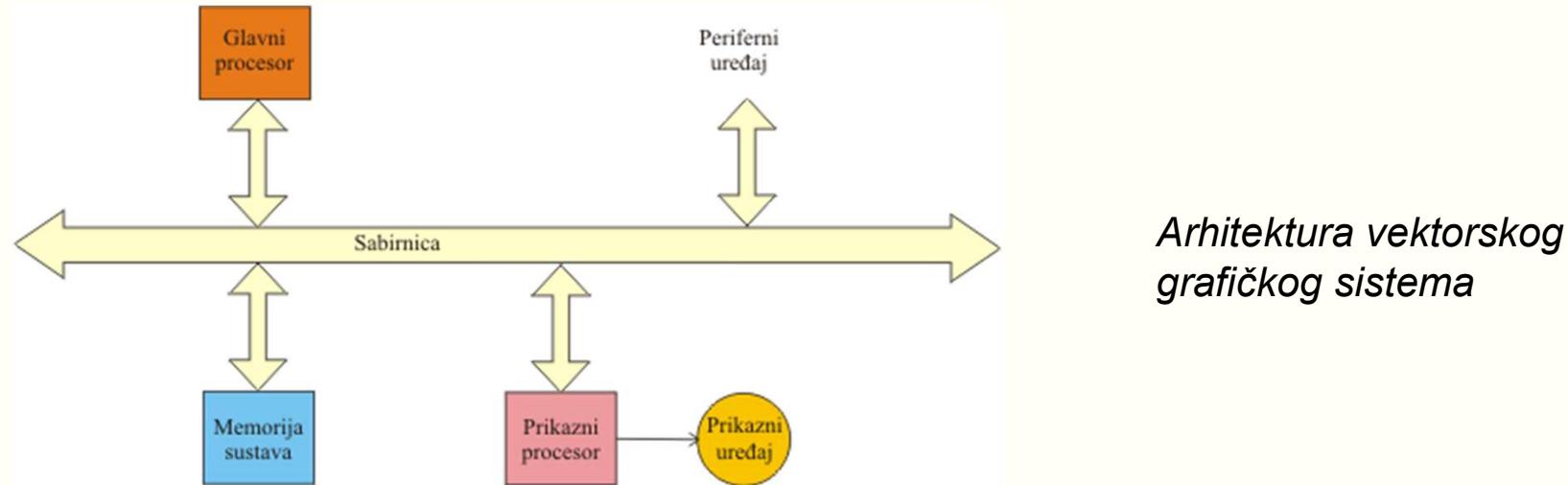
- **prikazni procesor** priključen kao *UI* uređaj na glavni procesor (interpretira grafičke naredbe i proslijeđuje koordinate tačaka vektorskog generatoru),
- **prikazna privremena memorija** (sadrži prikaznu listu ili prikazni program),
- **vektorski generator** (pretvara digitalne koordinate u analogne vrijednosti napona za otklonski sistem),
- **prikazni uređaj**.

```
:  
:  
MOVE  
10  
15  
LINE  
400  
300  
CHAR  
a  
b  
LINE  
. .  
JMP
```



Arhitektura grafičkog sistema s vektorskim prikaznim uređajem

Prikazna lista ili **prikazni program** u okviru *prikazne memorije* sadrži niz grafičkih naredbi (npr. za crtanje tačaka, linija, znakova). Na kraju je naredba JMP (skok) koja upućuje procesor na početak liste. *Procesor ciklički ponavlja naredbe iz prikazne liste frekvencijom od najmanje 30 puta u sekundi i na taj način stvara privid mirne slike jednolikog intenziteta na fosfornom zaslonu* koji sadržava osvijetljenost u desetcima ili stotinama mikrosekundi.



Glavni procesor izvodi aplikacijski program i grafički paket koji su pohranjeni u memoriji sistema.

Grafički paket kreira prikaznu listu i ukazuje na početnu naredbu.

Memoriji sistema pristupaju glavni procesor i grafički prikazni procesor (*graphic display processor*) koji pretvara izlazne primitivne oblike u bitovnu matricu, te izvodi transformacije i izmjene u rasterskim prikaznim sistemima (prikazna procesorska jedinica, grafički kontroler).

Grafički prikazni procesor dohvata, dekodira i izvodi naredbe iz prikazne liste. Procesor u pravilu ima X i Y registre i brojač naredbi. Naredbe su definisane kao 16-bitovne riječi. Tipična je naredba LD sa inačicama M, P, L (load and move; load and point; load and line) za pomak, crtanje tačke i crtanje ravne linije. Parametar R/A određuje da li se radi o relativnoj ili absolutnoj adresi.

LD {X/Y} {R/A} M	[000 X/Y R/A (X/Y/dX/dY)]
LD {X/Y} {R/A} P	[001 X/Y R/A (X/Y/dX/dY)]
LD {X/Y} {R/A} L	[010 X/Y R/A (X/Y/dX/dY)]
{X/Y}: 0 => X, 1 => Y	
{R/A}: 0 => dX/dY, 1 => X/Y	

Struktura naredbi LD i značenje vrijednosti parametara {X/Y} i {R/A}

POČETAK:	LDXA 100	
LDYAM	100	Pomak na (100,100)
LDXRL	400	linija do (500,100)
LDYRL	400	linija do (500,500)
LDXRL	-400	linija do (100,500)
LDYRL	-400	linija do (100,100)
JUMPR	POČETAK	

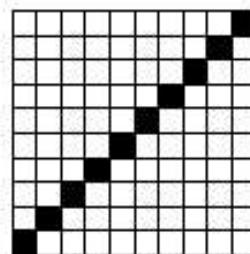
Primjer prikazne liste koja prikazuje kvadrat stranice dužine 400 s lijevim donjim vrhom u tački (100,100)

Vektorski grafički sistemi nemaju mogućnost prikaza ispunjenih površina, manipulaciju bitovima i tablicama, ali mogu ostvariti veće rezolucije od rasterskih sistema i prikazivati glatke kose linije.

- Vektorski zapis, engl. *vector format, vector graphic*.
- Kod vektorskog zapisa slika se pohranjuje kao matematička prikaz objekata na slici.
- Objekt je dio slike predložen jednim matematičkim izrazom.
- Moguće je matematički analizirati sliku, npr. izračunati površinu, težište, polovište i sl.
- Moguće su i jednostavno izvedive transformacije slike, npr. povećanje i smanjenje, preslikavanje, ukošenje, rotacija i sl.
- Objedinjavanje objekata: više objekata moguće je objediniti u skupni objekt (grupisati).
- Sve transformacije na skupnom objektu odnose se istovremeno na sve objekte od kojih je sastavljen, npr. povećanje, zakretanje, bojenje.

Usporedba (2)

Vektorski zapis:



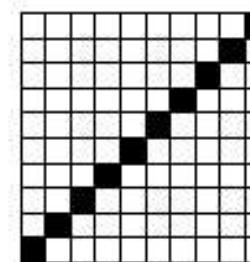
A0099

A
rijed je o točki
0 0 početna točka (0,0)
9 9 savršna točka (9,9)

24

Usporedba (1)

Točkasti zapis:

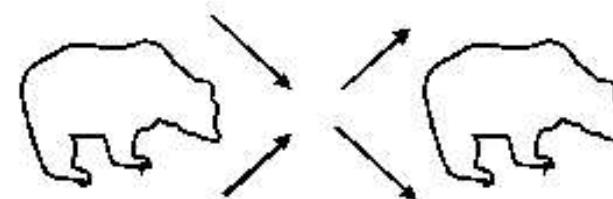


0000000001
0000000010
0000000100
00000010000
00000100000
00010000000
00100000000
01000000000
10000000000

25

Vektorski zapis - primjer

■ Primjer smanjenja i ponovnog povećanja vektorskog zapisa.



26

110

Karakteristike vektorskog zapisa

- Veličina vektorskog zapisa u načelu ne zavisi o veličini prikaza (broju tačaka slike na display-u).
- Veličina vektorskog zapisa u načelu zavisi o broju i složenosti objekata u slici.
- Bez obzira na format zapisa, slika se na display-u monitora mora prikazati pomoću tačaka display-a.
- Pri svakom prikazu na display monitora potrebno je matematički prikaz pretvoriti u raspored tačaka display-a.
- Pretvaranje traje određeno vrijeme.
- Trajanje pretvaranja zavisi o složenosti slike.

Prikaz objekata vektorskog zapisa

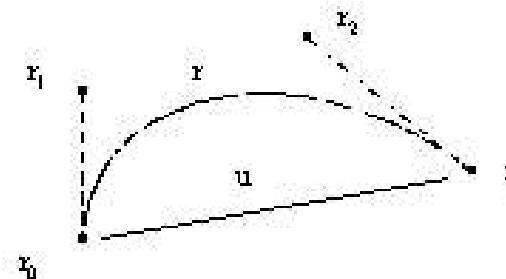
- Jednostavniji objekti: dužina (početna i završna točka), kružnica (središte i poluprečnik), pravougaonik (dvije dijagonalne točke) itd.
- Krivulje se prikazuju Bezier krivama koje su određene kontrolnim tačkama.

Bezier krivulje - karakteristike

- Krivulju je moguće prikazati (zapisati) npr. s četiri tačke, pa zapis zauzima malo memorije.
- Složene krivulje se mogu sastaviti od više Bezier krivulja definisanih s po četiri tačke.
- Krivulju je jednostavno oblikovati pri izradi crteža pomicanjem kontrolnih točaka.

Bezier krivulje - definicija

$$r = (1-u)^3 r_0 + 3u(1-u)^2 r_1 + 3u^2(1-u)r_2 + u^3 r_3$$



Prednosti vektorskih zapisa

- Moguće matematičke transformacije objekata slike i cijele slike, npr. povećanje i smanjenje, zakretanje.
- Moguće matematičke obrade objekata i cijele slike, npr. izračunavanje površine, težišta.

Nedostaci vektorskih zapisa

- Otežano crtanje i prikaz nepravilnih crteža (npr. crteži nejasnih obrisa, prelijevanja boja, fotografije).
- Relativno dugo iscrtavanje slike pri prikazu na display monitora.

Formati vektorskog zapisa

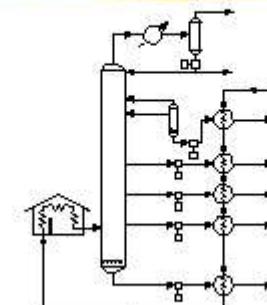
Rasprostranjeni formati vektorskih zapisa:

- dwg (AutoCAD),
- cdr (CorelDraw!),
- dxf- Drawing Exchange Format (format zapisa za razmjenu između podataka između različitih vektorski orientiranih programa).

Vektorski zapis - primjer (1)



Vektorski zapis - primjer (2)



RASTERSKI GRAFIČKI SISTEMI

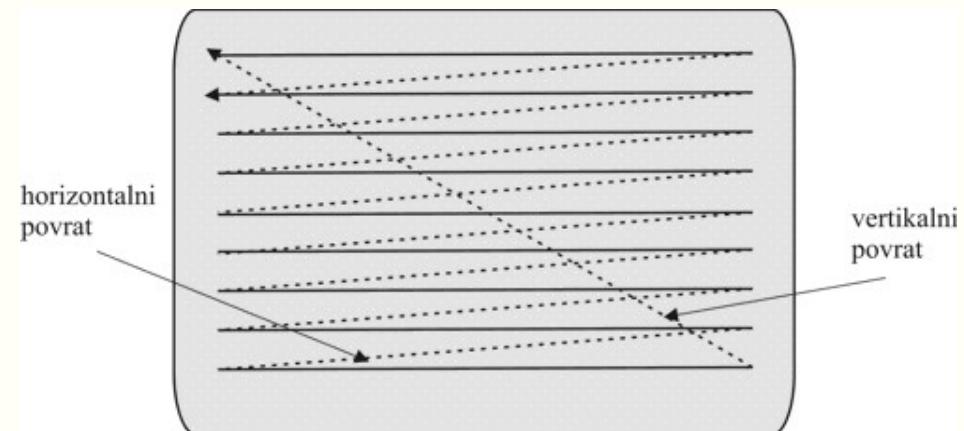
Rasterska grafika (*bitmap graphics, raster graphics*) razvila se ranih 70-ih godina na temelju jeftine televizijske tehnologije.

Rasterski prikazni uređaji pohranjuju primitivne oblike (kao što su linije, alfanumerički znakovi, ispunjene površine) u memoriju u obliku njihovih osnovnih sastavnih slikovnih elemenata-piksela.

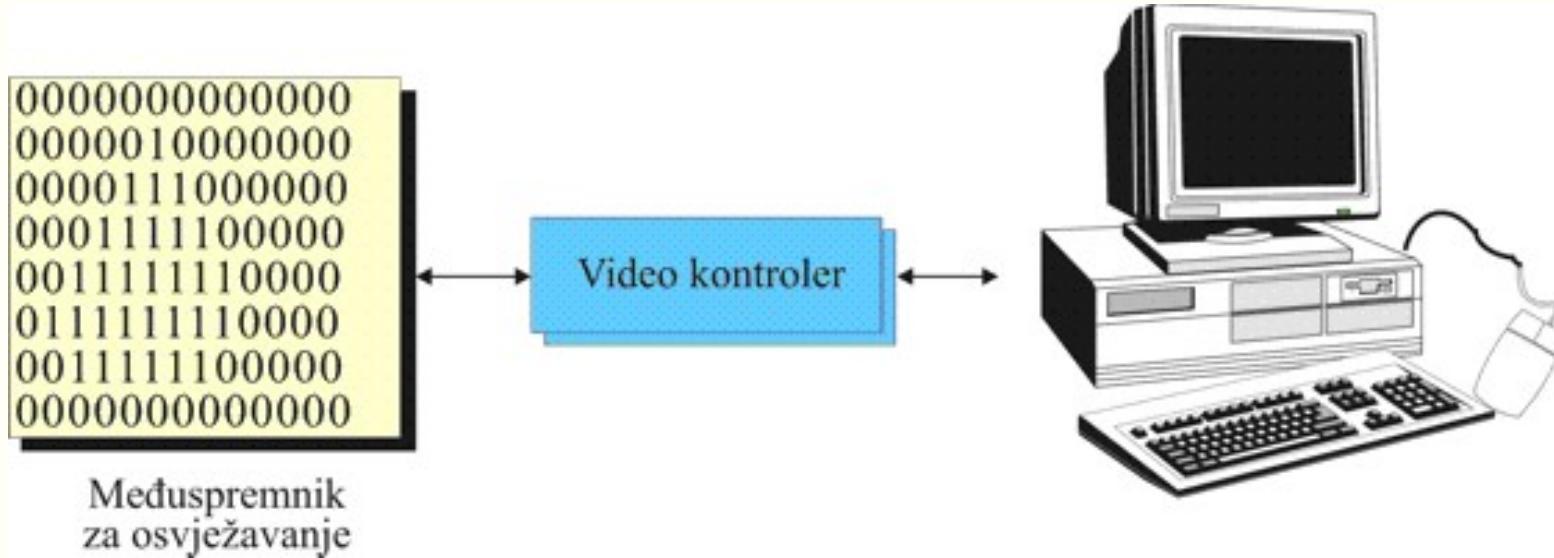
Cijelovita slika prikazuje se na rasteru koji predstavlja niz paralelnih horizontalnih redova slikovnih elemenata (ili pravougaonu matricu slikovnih elemenata), koji prekrivaju čitavu površinu zaslona.

Pri kreiranju prikaza zraka prolazi preko svih piksela uvijek istim slijedom po svim horizontalnim redovima piksela s lijeva na desno od gornjeg do donjeg horizontalnog reda piksela.

U sistemu sa dva nivoa intenzitet zrake pri prolasku preko pojedinog piksela određuje njegovu svjetloću odnosno boju (jednu od dvije moguće). U sistemima sa prikazom u boji koriste se tri zrake (crvena-*R*, zelena-*G* i plava-*B*), a kombinacija njihovih intenziteta određuje boju piksela. Pri povratku na početak sljedećeg reda piksela zraka se zatamnjuje (ne izaziva vidljivu promjenu intenziteta ili boje piksela).



Putanja zrake (raster scan) pri kreiranju prikaza na rasterskom prikaznom uređaju



Arhitektura grafičkog sistema s rasterskim prikaznim uređajem

Osnovni pojnovi rasterske grafike su:

- **slikovni element** – elementarna površina na zaslonu čijom svjetloćom (*ili bojom*) je moguće upravljati (*piksel, pel – picture element*)
- **raster** - niz paralelnih horizontalnih redova slikovnih elemenata, odnosno pravougaona matrica slikovnih elemenata koja prekriva čitavu površinu zaslona
- **bitovna matrica (bitmap)** – matrica čiji elementi (1,0) predstavljaju svjetloću (*ili boju*) odgovarajućih elemenata pravougaonog rasporeda osvjetljivih tačaka zaslona (*slikovnih elemenata*) u sistemu sa dva nivoa (*informacijski kapacitet je 1bit/piksel*).
- **matrica slikovnih elemenata (pixmap – pixel map)** – matrica čiji elementi predstavljaju boju odgovarajućih elemenata pravougaonog rasporeda osvjetljivih tačaka zaslona (*slikovnih elemenata*) u sistemu sa više nivoa (*informacijski kapacitet n bit/piksel*)

Da bi se ubrzao proces pretvaranja modela slike u rasterski prikaz često se koriste posebni namjenski procesori (*raster image processor*) koji imaju ulogu koprocesora ili ubrzivača (*accelerator*).

NAPOMENA:

Mjera količine informacije vjerovatnog događaja naziva se **BIT** (**BInary digiT** = binarni broj) .

Količina informacije računa se izrazom:

$$I = \log_2 \frac{1}{p}$$

I Količina informacije
p vjerovatnost zbijanja
događaja

Na jednom mjestu u isto vrijeme može i ne mora da postoji impuls (I), odnosno da bude pozitivan ili negativan što simbolički predstavlja stanja "1" ili "0", te je količina informacije za jedno mjesto u slijedu:

I = 1 b (jedan bit)

BIT je kao jedinica bio dosta neprikladan za praktičnu upotrebu te je uveden pojam **BYTE** (**BAJT**), složenica od izraza 'BinarY TERm' kao oznaka koja se odnosi na radnje povezane uz jedan znak. U začecima digitalne računarske ere broj bit-a za jedan znak nije bio isti kod svih vrsta računara, no danas je za taj pojam opće prihvaćeno da je to skup od 8 bit-a.

DOGOVOR:

8 b (bit-a) = 1 B (jedan BAJT)

16 b = 2 B = RIJEČ

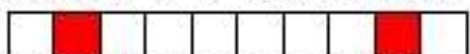
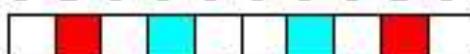
32 b = 4 B = DUGA RIJEČ

4 b = 1/2 B = POLUBAJT

• Primjer rasterskog zapisa

- Vrijednost bajta označava boju (npr. 0=bijela boja, 1=tamnoplava boja, 2=crvena boja, 3=svjetloplava boja, 4=žuta boja, 5=zelena boja).
- Za zapis prikazane slike potrebno je 100 bajtova (za svaku tačku jedan bajt).

Primjer točkastog zapisa (3)

1 red	
	0 0 0 0 1 0 0 0 0 0
2 red	
	0 0 1 1 0 1 1 1 0 0
3 red	
	0 2 0 0 0 0 0 0 2 0
4 red	
	0 2 0 3 0 0 3 0 2 0

Primjer točkastog zapisa (4)

```
0000100000  
0011011100  
0200000020  
0203003020  
2200400022  
0200000022  
0200555020  
0020000020  
0002222200  
000000000000
```

21

22

Karakteristike rasterskog zapisa

- Jedna zaslonska tačka predočena je s jednim ili više bajta (s jednim bajtom moguće je prikazati najviše 256 boja).
- Veličina zapisa u načelu direktno zavisi o veličini slike tj. broju zaslonskih tačaka potrebnih za prikaz slike.
- Zapis se obično sumira (veličina objedinjenog zapisa ovisi o karakteristikama slike)
- Složenost slike ne utiče na veličinu neobjedinjenog zapisa.
- Vrijeme iscrtavanja na zaslonu monitora ne ovisi o složenosti slike.
- Otežani ili nemogući neki postupci obrade slike poput: povećanja i smanjenja slike, ukošenja i sl.

Prednosti rasterskog zapisa

- Jednostavna izrada nepravilnih i šarenih uzoraka,
- relativno brz prikaz na zaslonu bez obzira na karakteristike crteža,
- jednostavna upotreba.

Nedostaci tačkastog zapisa

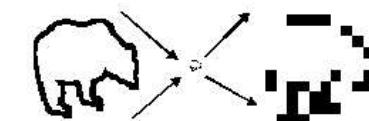
- Otežane ili nemoguće transformacije crteža (povećanje, zakretanje i sl.),
- otežana ili nemoguća matematička analiza crteža (pronaći težište, izračunati površinu i sl.),
- otežana izrada crteža crtalom,
- otežano otiskivanje pisačima različite rezolucije.

Primjena tačkastih zapisa

- Pogodno za likovne crteže nepravilnih i raznobojsnih oblika kod kojih nije potrebna transformacija i matematička analiza crteža.
- Komercijalni tačkasto orijentisani programi obično u svom nazivu imaju riječ "paint" (npr. Corel Photo Paint, Paintbrush)

Točkasti zapis - primjer

■ Primjer smanjenja i ponovnog povećanja točkastog zapisa.



23

Primjena tačkastih zapisa

- Pogodno za likovne crteže nepravilnih i raznobojnih oblika kod kojih nije potrebna transformacija i matematička analiza crteža.
- Komercijalni tačkasto orijentisani programi obično u svom nazivu imaju riječ "paint" (npr. **Corel Photo Paint**, Paintbrush)

Točkasti zapis - primjer



20

Formati tačkastog zapisa

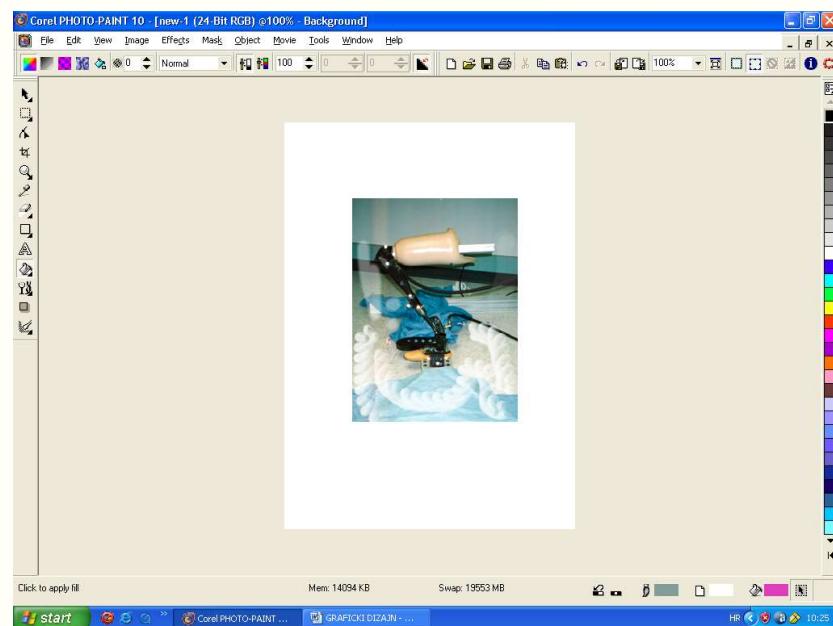
Rasprostranjeni formati tačkastih zapisa (zapis ima navedeni produžetak):

- bmp-Basic Multilingual Plane
- gif -Graphics Interchange Format
- tiff-Tag Image File Format
- pcx-Paintbrush exchange format
- jpg-Joint Photographic Experts Group
- npr. pijetao.bmp, cvijet.pcx, dijete.jpg

Adobe photo shop



Corel photo paint



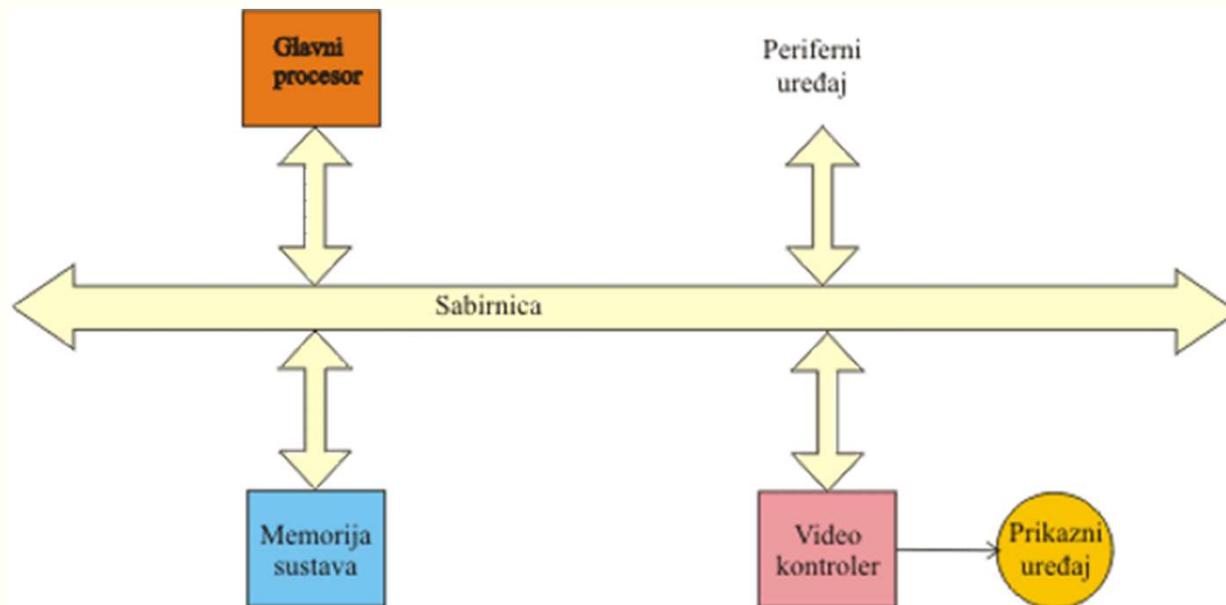
Arhitekture rasterskih prikaznih sistema

U rasterskim prikaznim sistemima grafički prikazni procesor (grafički kontroler, prikazni koprocesor) pretvara izlazne primitivne oblike u bitovnu matricu, te izvodi transformacije i izmjene.

Sistemi se razlikuju s obzirom na raspodjelu poslova između glavnog procesora i grafičkog prikaznog procesora te na način pohrane bitovne matrice tako da postoje:

- jednostavni rasterski prikazni sistemni,
- rasterski prikazni sistemi sa zasebnim prikaznim procesorom i
- rasterski prikazni sistemi sa jedinstvenim memorijskim prostorom

□ Jednostavni rasterski prikazni sistem



Arhitektura jednostavnog rasterskog prikaznog sistema

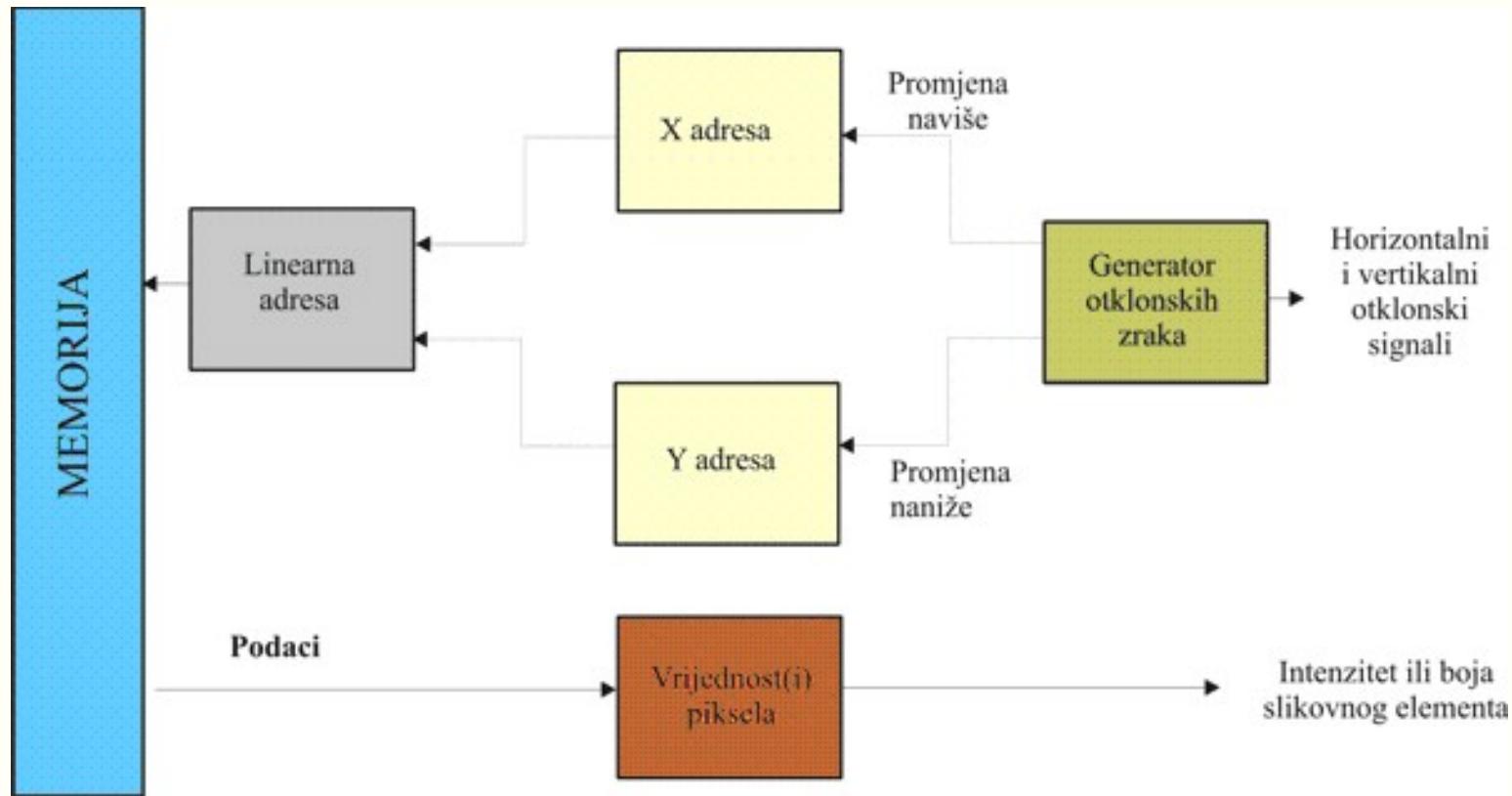
Osnovna svojstva ove arhitekture su:

- odnos memorije i glavnog procesora je isti kao kod negrafičkih sistema,
- dio memorije služi kao bitovna matrica (fiksni dio, stranice, proizvoljni dio),
- aplikacijski program i grafički paket pohranjeni su u memoriji sistema, a izvodi ih glavni procesor,
- video kontroler prikazuje slike pohranjene u okvirnom međuspremniku,
- video kontroler koji sadrži generator adresa i otklonskih signala adresira lokacije međuspremnika okvira u memoriji, a podaci određuju intenzitet ili boju slikovnih elemenata,
- nedostatak ovakve arhitekture je sporost i veliki broj pristupa memoriji.

Video kontroler ima zadatak stalnog osvježavanja prikaza. Da bi se izbjegao efekt treperenja osvježavanje se treba obavljati frekvencijom od minimalno *60 Hz*.

Ako se svi retci piksela u rasteru osvježavaju istovremeno radi se o **sistemu bez prepletanja** (*noninterlaced*).

Ako se izmjenično osvježavaju parni i neparni retci u rasteru (u pravilu upola manjom frekvencijom) radi se o **sistemu s prepletanjem** (*interlaced*).



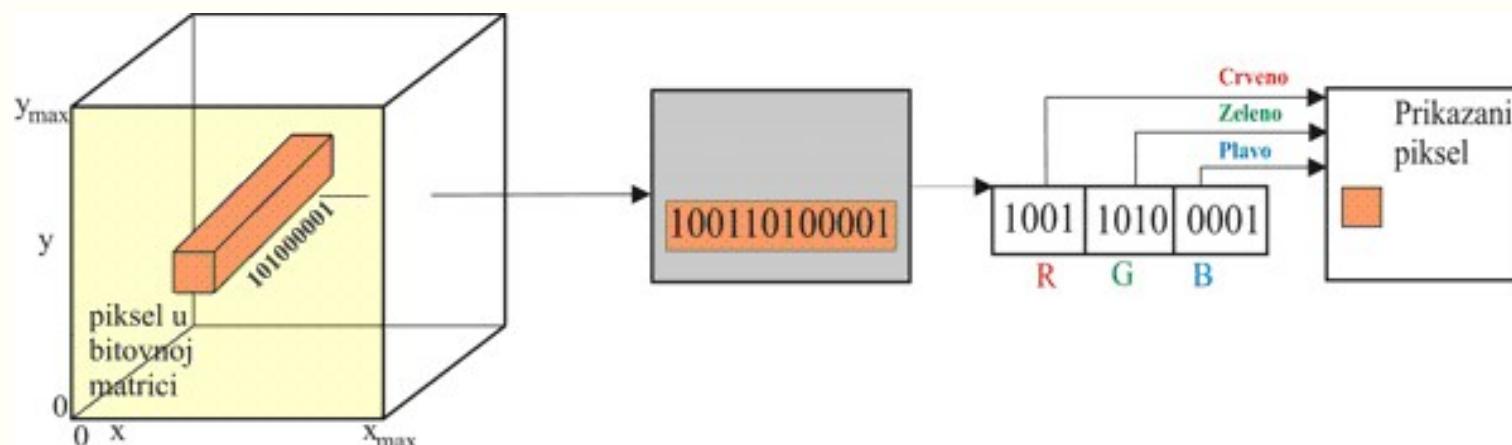
Logička organizacija video kontrolera.

Generator horizontalnih i vertikalnih otklonskih signala usmjerava zraku na određeni slikovni element na zaslovu. Istovremeno generira i horizontalnu i vertikalnu adresu piksela u koordinatnom sistemu zaslona.

Na osnovu tih koordinata izračunava se linearna adresa (adresa pripadajuće memorijske lokacije) na kojoj se nalaze podaci o svjetloći ili boji slikovnog elementa. Na osnovu tih podataka postavljaju se parametri svjetlosnih izvora koji određuju svjetloću ili boju slikovnog elementa.

Zbog uštede memorijskog prostora video kontroler često sadrži **pretvaračku tablicu** (*look-up table, LUT*). Pretvaračka tablica ima onoliko elemenata koliko ima različitih vrijednosti piksela. Vrijednost piksela ne koristi se direktno za upravljanje bojom piksela nego predstavlja pokazivač u pretvaračkoj tablici.

Pokazana vrijednost iz pretvaračke tablice upravlja zrakama koje određuju boju piksela na zaslonu.



Logička organizacija pretvaračke tablice s 256 elemenata koja je povezana s međuspremnikom s 8 bit/piksel.

Rasterski prikazni sistem sa zasebnim prikaznim procesorom

Uvođenjem **zasebnog prikaznog procesora** koji preuzima grafičke funkcije kao što je pretvaranje modela u rasterski prikaz mogu se poboljšati osobine sistema u odnosu na jednostavni rasterski prikazni sistem.

U tom slučaju u sistemu će procesorske funkcije biti raspodijeljene između dva procesora: **glavnog procesora opšte namjene i specijaliziranog grafičkog prikaznog procesora**.

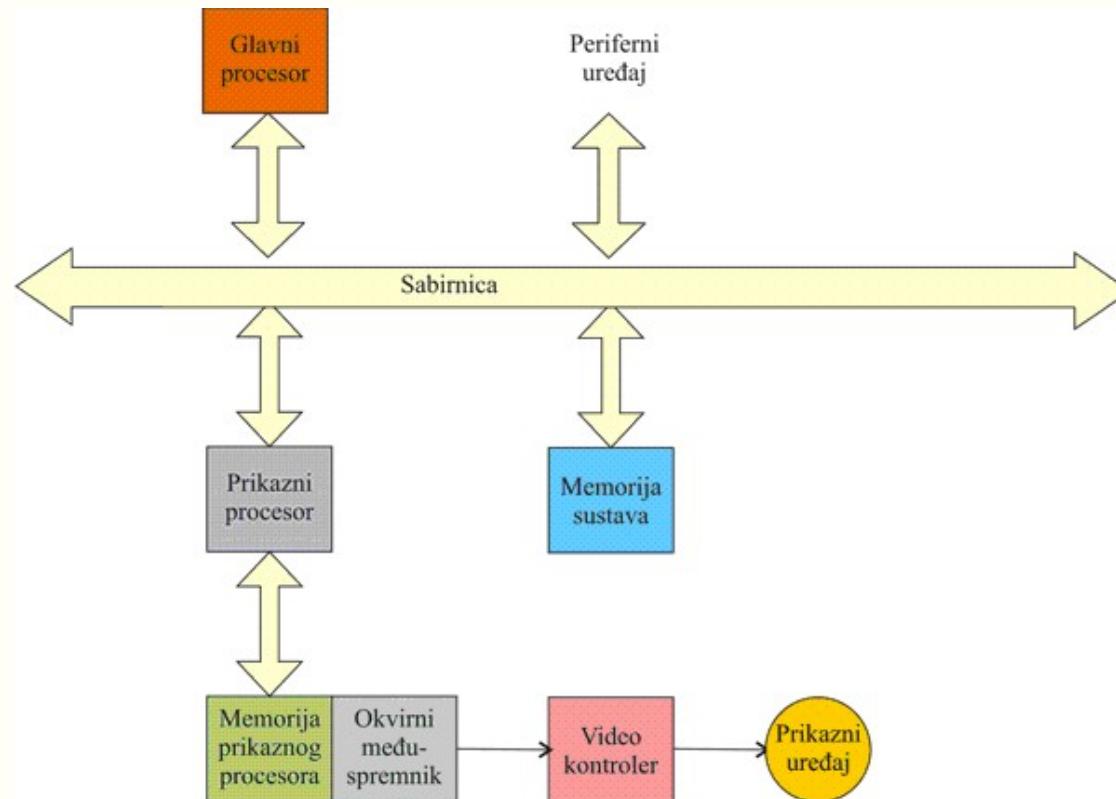
Pored toga u sistemu su definisana tri odvojena memoriska područja: **glavna memorija sistema, memorija prikaznog procesora i zasebni međuspremnik okvira.**

Memorija sistema sadrži aplikacijski program, grafički paket i operacijski sistem.

Memorija prikaznog procesora sadrži podatke i programe za stvaranje slike.

Slika se pohranjuje u okvirni međuspremnik.

Ovakvom organizacijom memorije omogućava se istovremeni pristup odgovarajućim dijelovima memorije od strane dvaju procesora i video kontrolera.

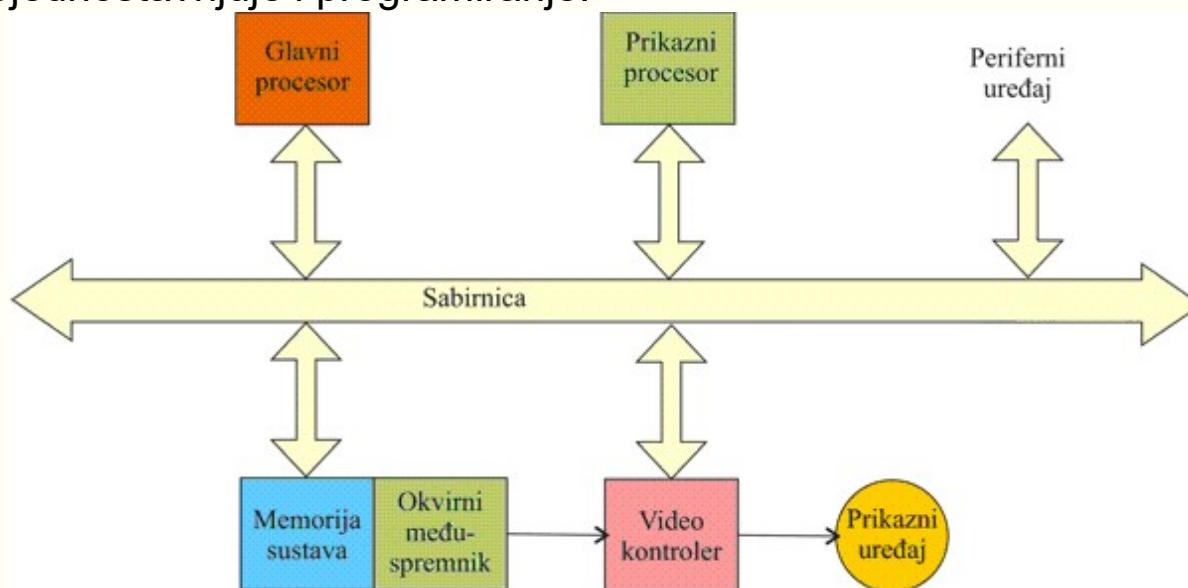


Arhitektura rasterskog prikaznog sistema sa zasebnim prikaznim procesorom. 124

Rasterski prikazni sistem s jedinstvenim memorijskim prostorom

Nedostatci arhitekture rasterskog sistema sa zasebnim prikaznim procesorom, a oni proizlaze iz potrebe za međusobnom komunikacijom dvaju procesora i prijenosom sadržaja između različitih memorijskih prostora, mogu se izbjegći **objedinjavanjem memorijskog adresnog prostora**.

U tom slučaju moguća je fleksibilna dodjela memorijskih prostora i pristupa što pojednostavljuje i programiranje.



Arhitektura rasterskog prikaznog sistema s integriranim prikaznim procesorom.

Nedostatci ove arhitekture proizlaze iz potrebe za istovremenim pristupom memoriji od strane dvaju procesora. Jedno rješenje je u korištenju priručne memorije u samom procesoru opšte namjene (**cache memorija**). Time se smanjuje broj i učestalost pristupa memoriji od strane glavnog procesora.

Jedan način poboljšanja sistema je i korištenje posebne vrste memorije kao što je video RAM (VRAM)

9. MATEMATSKE OSNOVE RAČUNARSKE GRAFIKE

UVOD

U procesima transformacije ulaznih podataka u modele objekata u grafičke prikaze na prikaznim uređajima neophodna je primjena niz elementarnih i složenih matematičkih transformacija.

Matematički opis objekata u n-domenzionalnom prostoru zasniva se na primjeni **analitičke geometrije u prostoru i ravni**.

GRAFIČKE TRANSFORMACIJE (*Matematički oblik transformacije grafičkih objekata*)

Transformacije grafičkog objekta mogu se prikazati u **matematičkom obliku** na sažet i jednostavan način.

Ako je uopštena transformacija homogene tačke u 3D prostoru definisana matricom **M** tada se njena transformacija može opisati na sljedeći način:

- a) homogena predstava tačke (x, y, z) u homogenom prostoru je homogena tačka (x_1, x_2, x_3, x_4)
- b) koordinate tačke transformisane homogene tačke $[x'_1, x'_2, x'_3, x'_4]$ određene su slijedećom jednačinom: $[x'_1, x'_2, x'_3, x'_4] = [x_1 \ x_2 \ x_3 \ x_4] M$

Transformacija objekta obavlja se tako da se ista transformacija primjeni na sve njegove tačke.

TRANSFORMACIJA TRANSLACIJE

Transformacijom translacije obavlja se pomak tačke V u tačku V' za iznos (T_x, T_y) .

Koordinate transformirane tačke mogu se odrediti slijedećim dvijema jednačinama:

$$\begin{aligned}x' &= x + T_x \\y' &= y + T_y\end{aligned}$$

Sažetiji način zapisa ovoga proračuna u homogenom prostoru može se izvesti ako definišemo matricu translacije \mathbf{T} na slijedeći način:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

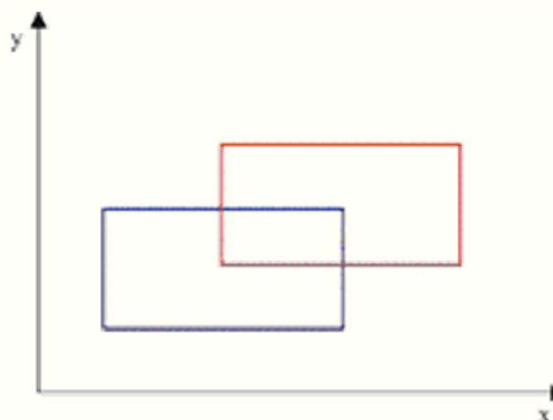
U homogenom prostoru možemo odabratи vrijednosti dodatne koordinate npr. iznosa 1 pa matrična jednačina za proračun koordinata transformirane tačke ima slijedeći oblik:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

Ovakva jednačina može se napisati u opštem obliku na slijedeći način: $\mathbf{V}' = \mathbf{V}\mathbf{T}$

Translacija nekog objekta obavit će se na način da se primjenom gornje jednačine translatiraju sve njegove tačke.

Primjer transformacije translacije



TRANSFORMACIJA ROTACIJE

Transformacijom rotacije definisana je rotacija tačke V oko ishodišta za ugao Θ

Koordinate tačke V' koja nastaje rotacijom tačke V oko ishodišta za ugao Θ mogu se izračunati na slijedeći način:

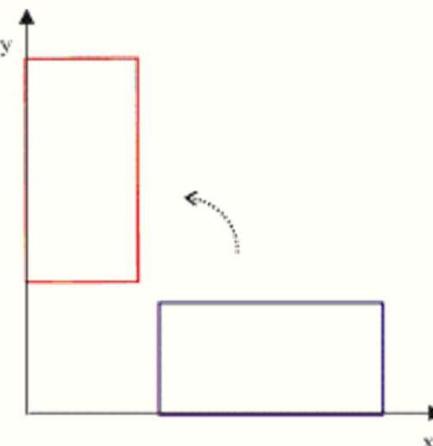
Sažetiji način zapisa ovog proračuna u homogenom prostoru može se izvesti ako definišemo matricu rotacije R na slijedeći način:

U homogenom prostoru možemo odabratи vrijednosti dodatne koordinate npr. iznosa 1 pa matrična jednačina za proračun koordinata transformisane tačke ima slijedeći oblik:

Ovakva jednačina može se napisati u opštem obliku na slijedeći način: $\mathbf{V}' = \mathbf{VR}$

Translacija nekoga objekta obavit će se na način da se primjenom gornje jednačine translatiraju sve njegove tačke.

Primjer transformacije rotacije



$$x' = x \cos \Theta + y \sin \Theta$$

$$y' = -x \sin \Theta + y \cos \Theta$$

$$R = \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 \\ \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 \\ \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

TRANSFORMACIJA PROMJENE FAKTORA PROPORCIONALNOSTI

Transformacija promjene faktora proporcionalnosti za faktor S_x u smijeru koordinatne ose x i za faktor S_y u smijeru koordinatne ose y definisana je slijedećim jednačinama:

Sažetiji način zapisa ovog proračuna u homogenom prostoru može se izvesti ako definišemo matricu promjene faktora proporcionalnosti S na slijedeći način:

U homogenom prostoru možemo odabrati vrijednost dodatne koordinate npr. iznosa 1 pa matrična jednačina za proračun koordinata transformisane tačke ima slijedeći oblik

Ovakva jednačina može se napisati i u opštem obliku na slijedeći način:

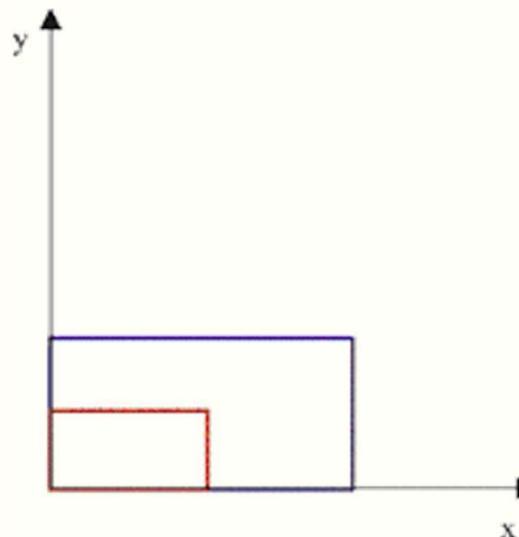
$$\begin{aligned}x' &= xS_x \\y' &= yS_y\end{aligned}$$

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{V}' = \mathbf{VS}$$

Primjer transformacije promjene faktora proporcionalnosti



TRANSFORMACIJA SMICANJA

Transformacija smicanja za ugao α u odnosu na koordinatnu osu x i ugao β u odnosu na koordinatnu osu y definisana je slijedećim jednačinama

$$x' = x + y \operatorname{tg} \beta$$

$$y' = y + x \operatorname{tg} \alpha$$

Sažetiji način zapisa ovog proračuna u homogenom prostoru može se izvesti ako definišemo matricu promjene faktora proporcionalnosti D na slijedeći način:

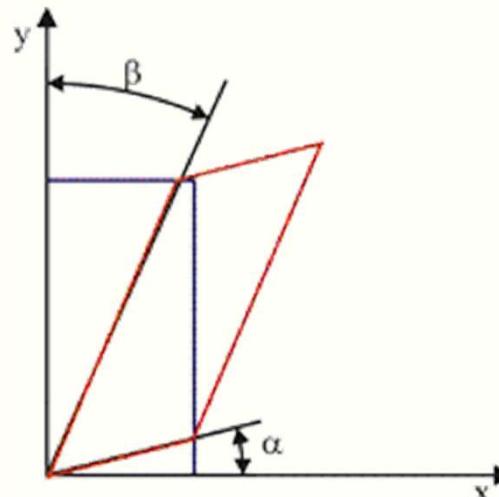
U homogenom prostoru možemo odabrati vrijednost dodatne koordinate npr. iznosa 1 pa matrična jednačina za proračun koordinata transformisane tačke ima slijedeći oblik:

Ovakva jednačina može se napisati i u opštem obliku na slijedeći način:

$$D = \begin{bmatrix} 1 & \operatorname{tg} \alpha & 0 \\ \operatorname{tg} \beta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & \operatorname{tg} \alpha & 0 \\ \operatorname{tg} \beta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{V}' = \mathbf{VD}$$



Primjer transformacije smicanja

SLOŽENE TRANSFORMACIJE

U slučajevima kada je potrebno izvesti **složene transformacije** npr. kombinaciju translacije i rotacije moguće je pristupiti na način da se obavljaju redom pojedine elementarne transformacije.

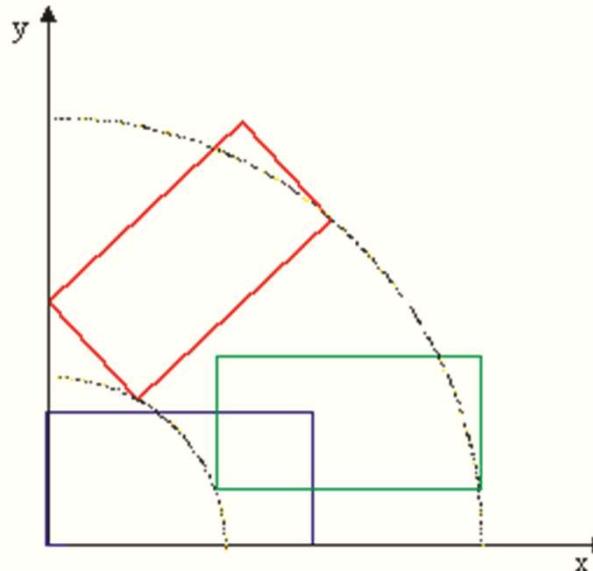
Međutim ovaj način zahtijeva velik broj matričnih množenja

Taj broj se može smanjiti ako se definiše matrica složene transformacije **M** množenjem matrica pojedinih elementarnih transformacija **M₁**, **M₂**,**M_n**. Nakon toga se obavlja množenje vektora pojedinih tačaka s matricom složene transformacije.

$$V' = V \begin{pmatrix} M_1 & M_2 & \dots & M_n \end{pmatrix}$$

$$V' = VM$$

Primjer složene transformacije



TRANSFORMACIJE ROTACIJE U 3D PROSTORU

U **3D prostoru** moguće je definisati rotaciju u odnosu na ishodište oko svake pojedine koordinatne ose.

Tačka s koordinatama (x, y, z) iz $3D$ prostora u homogenom prostoru predstavljena je vektorom $[x_1 \ x_2 \ x_3 \ x_4]$.

Rotacija tačke oko ose x za ugao Θ (ugao njihanja ili ugao nutacije) definisana je u homogenom prostoru slijedećom transformacijskom matricom:

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \Theta & -\sin \Theta & 0 \\ 0 & \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

U homogenom prostoru možemo odabrati vrijednost dodatne koordinate npr. iznosa 1 pa matrična jednačina za proračun koordinata transformisane tačke ima slijedeći oblik:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \Theta & -\sin \Theta & 0 \\ 0 & \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ovakva jednačina može se napisati u opštem obliku na slijedeći način:

$$V' = VR_x$$

Rotacija tačke oko ose y za ugao Θ (ugao vlastitog okretanja) definisana je u homogenom prostoru slijedećom transformacijskom matricom

$$R_y = \begin{bmatrix} \cos \Theta & 0 & \sin \Theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \Theta & 0 & \cos \Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

U homogenom prostoru možemo odabratи vrijednost dodatne koordinate npr. iznosa 1 pa matrična jednačina za proračun koordinata transformirane tačke ima slijedeći oblik:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \cos \Theta & 0 & \sin \Theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \Theta & 0 & \cos \Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ovakva jednačina može se napisati u opštem obliku na slijedeći način:

$$V' = VR_y$$

Rotacija tačke oko ose z za ugao Θ (ugao precesije) definisana je u homogenom prostoru slijedećom transformacijskom matricom

$$R_z = \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 & 0 \\ \sin \Theta & \cos \Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

U homogenom prostoru možemo odabratи vrijednost dodatne koordinate npr. iznosa 1 pa matrična jednačina za proračun koordinata transformirane tačke ima slijedeći oblik:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 & 0 \\ \sin \Theta & \cos \Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ovakva jednačina može se napisati u opštem obliku na slijedeći način:

$$V' = VR_z$$

PROJEKCIJE

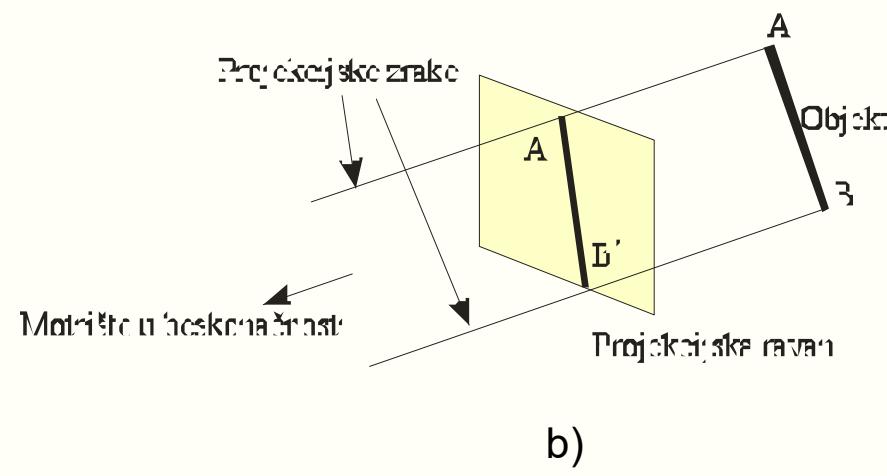
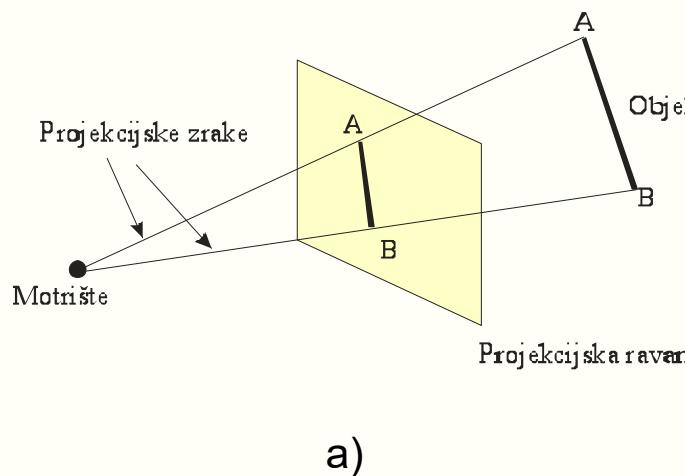
U računarskoj grafici često je potrebno **objekte iz prostora višeg reda prikazati u prostoru nižeg reda**.

Primjer takvog slučaja je prikaz scene definisane u 3D prostoru na dvodimenzionalnom prikaznom uređaju kao što je zaslon računara.

Ravninske geometrijske projekcije se u načelu odvijaju na način da projekcijske zrake koje izlaze iz projekcijskog središta prolaze kroz sve točke objekta i presjecaju projekcijsku ravninu tvoreći u njoj oblik projekcije.

Osnovne dvije skupine projekcija su **perspektivne projekcije i paralelne projekcije**.

Kod paralelnih projekcija udaljenost projekcijskog središta i projekcijske ravnine je beskonačna, dok je kod perspektivnih projekcija ta udaljenost konačna.



Primjer projekcije dužine: a) perspektivna projekcija, b) paralelna projekcija

ORTOGONALNA PROJEKCIJA

Ortogonalna projekcija preslikava sve tačke objekta paralelnim zrakama na određenu ravan.

Kod ortogonalne projekcije na xy -ravan $z=0$.

Matematički se ova operacija može definisati kao množenje vektora s dijagonalnom matricom čiji su svi elementi jednaki jedinici osim trećeg koji je jednak nuli.

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Na sličan način ortogonalna projekcija na xz -ravan odgovara izboru konstantne vrijednosti y koordinate iznosa 0 tj. $y=0$.

Matematički se ova operacija može definisati kao množenje vektora s dijagonalnom matricom čiji su svi elementi jednaki jedinici osim drugog koji je jednak nuli.

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ortogonalna projekcija na yz -ravan odgovara odabiru konstantne vrijednosti x koordinate iznosa 0 tj. $x=0$.

Matematički se ova operacija može definisati kao množenje vektora s dijagonalnom matricom čiji su svi elementi jednaki jedinici osim prvog koji je jednak nuli.

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Kosa projekcija ima sličnost s ortogonalnom u tome što su projekcijske zrake međusobno paralelne. Razlika je tome što nisu ortogonalne na projekcijsku ravan nego na nju upadaju pod nekim uglom.

PERSPEKTIVNA PROJEKCIJA

U praksi je često zanimljiva vrsta projekcije kod koje projekcijske zrake nisu paralelne.

Takav je slučaj kod fotografije i ljudskog vida

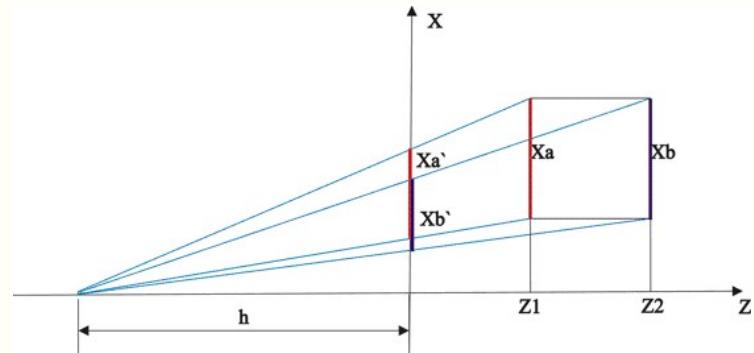
Fotografija predstavlja projekciju scene iz 3D prostora na dvodimenzionalnu ravnicu.

Projekcijske zrake u tom slučaju izviru iz jedne tačke na konačnoj udaljenosti od projekcijske ravni. Takva projekcija naziva se **perspektivna ili fotografkska**.

Promatrač se nalazi na udaljenosti h od projekcijske ravnine. Bridovi x_a i x_b preslikavaju se u dužine x'_a i x'_b .

Zbog sličnosti trokuta može se pisati

$$\frac{x'}{h} = \frac{x}{z+h} \quad \text{odnosno} \quad x' = \frac{x}{1 + \frac{z}{h}} \quad y' = \frac{y}{1 + \frac{z}{h}}$$



Ilustracija postupka perspektivne projekcije.

Na temelju ovih dvaju izraza može se definisati **transformacijska matrica perspektivne projekcije** te se postupak svodi na množenje vektora pojedine tačke s tom matricom.

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/h \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ 0 \ 1+z/h]$$

MATEMATIČKE OSNOVE CRTANJA PRIMITIVNIH OBLIKA

Prikazivanje slika objekata na rasterskim prikaznim uređajima zahtijeva pretvaranje idealnih **slika u rasterski prikaz** (*scan conversion*) s pikselima kao osnovnim elementima prikaza.

U našim razmatranjima možemo priхватiti aproksimaciju oblika piksela kao kružića sa središtem na sjecištima horizontalnih i vertikalnih pravaca koji prave rešetku

Postupak pretvaranja idealne slike u rasterski prikaz podrazumijeva određivanje nivoa svjetloće i/ili boju piksela u rasteru.

RASTERSKI PRIKAZ RAVNIH LINIJA

Algoritmi za rasterski prikaz ravnih linija proračunavaju koordinate piksela na 2D rasteru koji su najbliži idealnoj, beskonačnoj uskoj ravnoj liniji

Poželjne karakteristike prikazane ravne linije uključuju

- jednolik intenzitet linije čitavom dužinom,
- jednolik intenzitet linije neovisno o nagibu,
- mogućnost proizvoljnog izbora širine linije,
- mogućnost proizvoljnog oblikovanja krajeva.

Osnovni je problem crtanja kose ravne linije širine jednog piksela na rasterskom prikaznom uređaju sa dva nivoa.

Za tu svrhu razvijen je niz algoritama s ciljem pojednostavljenja i ubrzanja postupka.

Osnovni inkrementalni algoritam

Najjednostavniji pristup zasniva se na proračunu piksel po piksel na osnovu eksplisitnog oblika jednačine pravca.

Algoritmi se sastoje od slijedećih koraka:

- proračun nagiba pravca $m = Dy / Dx \quad (|m| < 1)$
- proračun nove vrijednosti x_i povećanjem prethodne vrijednosti za 1,
- proračun vrijednosti $y_i = mx_i + B$,
- sticanje piksela $(x_i, Round(y_i))$ gdje je $Round(y_i) = Floor(0.5 + y_i)$

Funkcija *Round* zaokružuje vrijednost realnog broja na najbliži cijeli broj. Ona se može realizirati pomoću funkcije *Floor* koja pretvara realni broj u cijeli broj odbacivanjem decimalnog dijela.

Osnovni inkrementalni algoritam odabire tačke koje su najmanje udaljene od idealne linije.

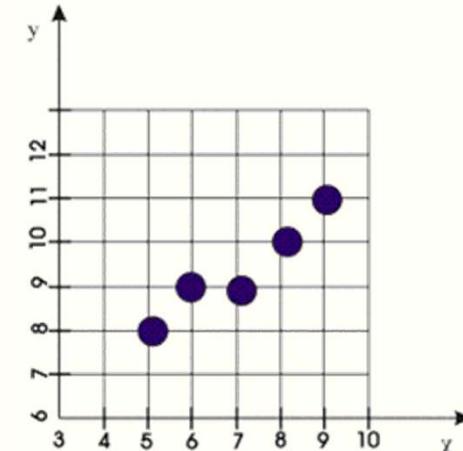
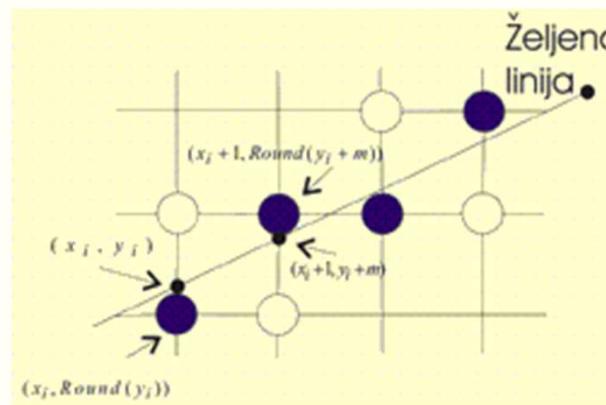
NAPOMENA

Osnovni **inkrementalni algoritam** predstavlja najjednostavniji pristup crtajući ravne linije na resterskom prikaznom uređaju, a temelji se na proračunu piksel po piksel na osnovi eksplisitnog oblika jednačine pravca. **Algoritam** odabire tačke koje su najmanje udaljene od idealne linije. Ovaj **algoritam** nije optimalan u smislu računske složenosti jer se u svakoj iteraciji obavljaju operacije množenja i sabiranja s realnim ili (racionalnim) brojevima kao i poziv funkcije *Floor*.

Ovaj interaktivni primjer omogućava promatranje rezultata primjene algoritma za crtajući zadane ravne crte. Ulazni podaci zadaju se u obliku koordinata početne i krajnje tačke: početne tačke $T_1(x_1, y_1)$ i krajnje tačke $T_2(x_2, y_2)$. Rezultat primjene algoritma prikazuje se u grafičkom obliku kao niz istaknutih piksela koji predstavljaju aproksimaciju idealne ravne linije koja spaja zadane krajnje tačke. Ulazni podaci ograničeni su na interval $[0, 10]$.

Primjer:

Primjenom osnovnog inkrementalnog algoritma odredite niz piksela koji aproksimiraju ravnu liniju koja povezuje tačke (5,8) i (10,11).



Primjer primjene osnovnog inkrementalnog algoritma

Digitalni diferencijalni analizator

Nedostaci osnovnog inkrementalnog algoritma mogu se velikim dijelom otkloniti ako se algoritam izmjeni na način da se izbjegne množenje realnih brojeva (koje predstavljaju vrlo zahjevnu operaciju za procesor).

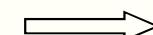
Množenje se može izbjegići ako se izraz za proračun vrijednosti koordinate y napiše na slijedeći način:

mijenjaju se uloge x i y koordinata u algoritmu:

$$Dy=1, \quad Dx=Dy/m=1/m$$

Na taj način vrijednosti x i y koordinata se povećavaju u malim koracima za vrijednost 1 ili $-1 < m < 1$.

Ovakav algoritam naziva se **digitalni diferencijalni analizator (DDA)**. DDA u osnovi predstavlja algoritam za numeričko rješavanje diferencijalnih jednačina istovremenim uvećavanjem koordinata x i y za vrijednosti proporcionalne njihovim prvim derivacijama.



$$y_{i+1} = mx_{i+1} + B = m(x_i + Dx) + B = y_i + mDx$$

Uz izbor da je $Dx=1$ slijedi:

$$y_{i+1} = y_i + m$$

Ako je $|m| > 1$

Algoritmi zasnovani na aritmetici cijelih brojeva

Računska složenost DDA algoritma, iako je značajno smanjena u odnosu na osnovni inkrementalni algoritam, još uvijek nije optimalna jer sadrži operacije s realnim brojevima.

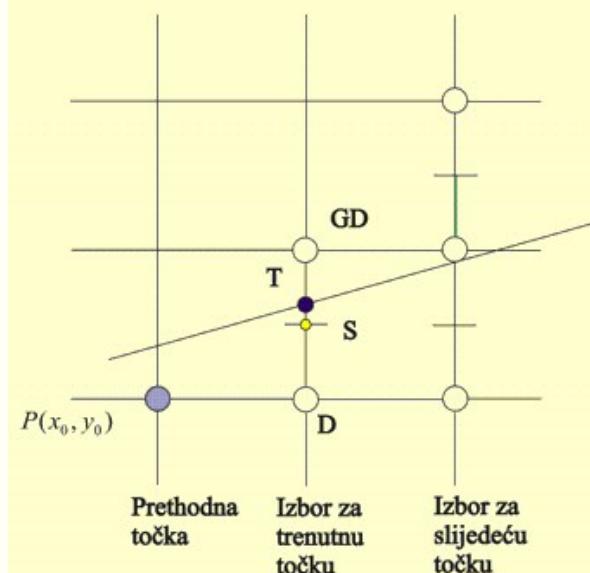
Daljnji napredak na ovom području ostvaren je razvojem **algoritama koji su zasnovani na aritmetici cijelih brojeva**.

Bresenhamov algoritam baziran na inkrementalnoj tehnici daje najbolju aproksimaciju idealnih linija u smislu minimizacije grešaka (odносно udaljenosti odabralih tačaka od idealnih linija), a može se primjeniti i na crtanje kružnica.

Poboljšane varijante ovoga algoritma baziraju se na primjeni **tehnike središnje tačke (Pitteway, Van Aken)**.

Primjer:

Primjenom algoritma središnje tačke odrediti niz piksela koji aproksimiraju ravnu liniju koja povezuje tačke (5,8) i (10,11).



Primjer primjene algoritma središnje tačke za crtanje ravne linije

Prvo se definiše središnja tačka $S(x_p+1, y_p+1/2)$ kao tačka polovišta dužine koja povezuje tačke D i GD. Odluka se donosi na osnovu informacije o položaju središnje tačke u odnosu na idealni pravac. Ako je središnja tačka S ispod idealnog pravca onda je očigledno da je tačka GD bliža idealnom pravcu. Ako je središnja tačka S iznad idealnog pravca onda je tačka D bliža idealnom pravcu.

Da bi se ovaj postupak odvijao na efikasan način definiše se varijabila odluke d ($d=F(S)=F(x_p+1,$

$y_p+1/2)=a(x_p+1)+b(y_p+1/2)+c$) koja se rekursivno izračunava od koraka do koraka. Na osnovu predznaka varijabile odluke određuje se slijedeća tačka. U svakom koraku algoritam vrši izbor između dvije tačke na osnovu predznaka varijabile odluke te uvećava varjabilu odluke za jednu od dvije moguće vrijednosti u zavisnosti od izabrane tačke.

ISPUNJAVANJE PRIMITIVNIH OBLIKA

Mogućnost ispunjavanja, bojom ili određenim uzorkom, primitivnih oblika kao što su pravougaonici, kružnice, elipse i mnogougaonici značajno doprinosi vrijednosti i primjenjivosti grafičkih sistema.

Osnovni koraci u tom postupku su:

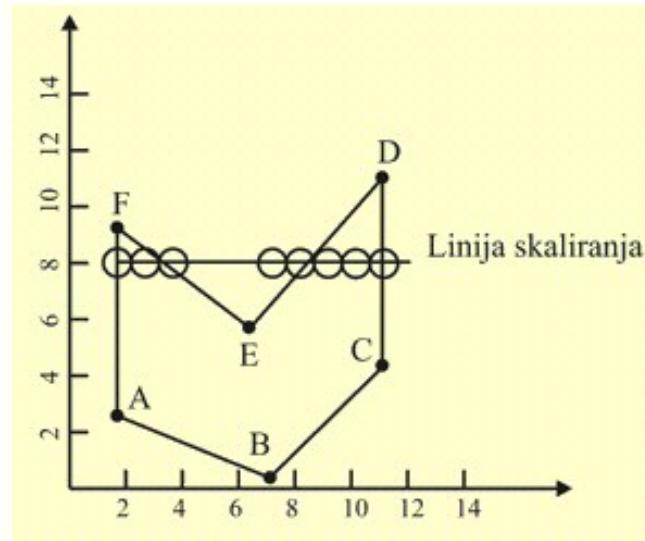
- izbor piksela koje treba ispuniti (zavisi od oblika),
- izbor vrijednosti koju treba pridijeliti odabranim pikselima (zavisi od načina ispunjavanja: bojom ili uzorkom).

Izbor piksela koje treba ispuniti odvija se u slijedećim koracima:

- skaliranje po horizontalnim linijama (retcima piksela),
- proračun presjecišta horizontalne linije i primitivnog oblika inkrementalnim algoritmom (odabiru se tačke koje leže unutar oblika),
- poredavanje presjecišta po veličini x koordinate,
- ispunjavanje odsječaka koji leže unutar primitivnog oblika (između neparnih i parnih presjecišta).

U slučaju ispunjavanja primitivnog oblika bojom odabranim pikselima pridjeljuje se prethodno odabrana konstantna vrijednost.

Primjer ispunjavanja mnogouglog oblika bojom.



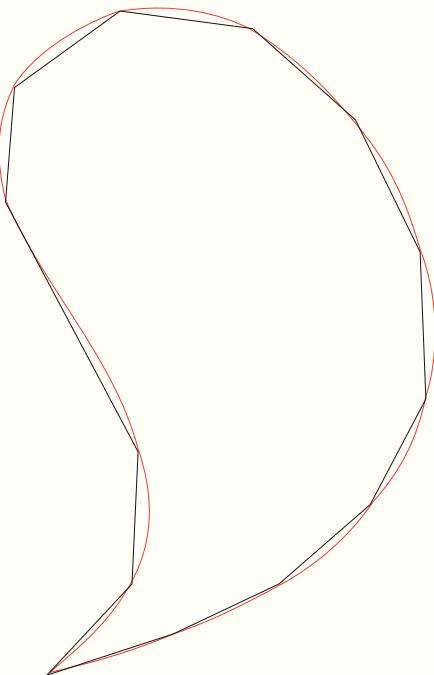
CRTANJE KRIVIH LINIJA

Mnogi objekti iz stvarnog svijeta, ali i iz virtualnih svjetova, koje se prikazuje primjenom računarske grafike omeđeni su glatkim krivim linijama ili plohami.

Najjednostavniji pristup modeliranju krive linije je **linearna aproksimacija** (prvog reda) **po dijelovima**

Kriva linija se aproksimira višestrukim linijama (niz povezanih ravnih linija) ili mnogouglovima. Tačnost aproksimacije određena je brojem linearnih segmenata kojima se aproksimira pojedini dio krive linije.

Za visok nivo podudarnosti linearne aproksimacijske modela i željene krive linije potreban je velik broj linearnih segmenata.



*Primjer modeliranja krive linije
linearnim segmentima.*

Veći nivo podudarnosti odnosno bolja aproksimacija uz manji broj pojedinačnih segmenata može se ostvariti primjenom **aproksimacija višeg reda**.

Postoji više oblika matematičkog prikaza krivih linija za aproksimacije višeg reda: **eksplicitni, implicitni i parametarski**.

U slučaju primjene **eksplicitnog oblika** koordinate y i z izražene su kao eksplisitne funkcije koordinate x : $y=y(x)$, $z=z(x)$. Nedostatci ovog oblika u primjenama računarske grafike su sljedeći:

- nisu moguće višestruke vrijednosti x (kao npr. kod kružnica),
- nije sačuvana rotacijska invarijantnost (nije jednostavno rotirati krivulju),
- teškoće s vertikalnim tangentama (zbog beskonačnog iznosa nagiba).

U slučaju primjene **implicitnog oblika** jednačina krive ima oblik $f(x,y,z)=0$. Nedostatci implicitnog oblika u primjenama računarske grafike su slijedeći:

- problem s višestrukim rješenjima (potrebno je postavljati dodatne uslove za izbor željenog rješenja),
- problem s kontinuitetom tangent u dodirnim tačkama različitih segmenata (podudarnost smjera).

U slučaju primjene **parametarskog oblika** jednačine krive sve tri koordinate izražene su kao funkcije parametra t : $x=x(t)$, $y=y(t)$, $z=z(t)$.

Parametarski oblik jednačine krive nema prethodne navedene nedostatke eksplisitnog i implicitnog oblika te je stoga najprikladniji za modeliranje krivih linija u računarskoj

Parametarske krivulje trećeg reda

Parametarske krive linije trećeg reda najčešće se koriste za modeliranje krivih linija u računarskoj grafici jer omogućavaju dovoljno fleksibilnosti za oblikovanje različitih krivih linija uz prihvatljiv nivo složenosti.

Model krive linije se specificira po odsjećima polinomima trećeg reda i svaki odsječak Q opisan je s tri funkcije (polinoma trećeg reda) x , y i z parametra t na sljedeći način: $Q(t) = [x(t) \ y(t) \ z(t)]$ gdje je:

Ako definišemo vektor potencija parametra t na sljedeći način:

te matricu koeficijenata triju polinoma na sljedeći način:

tada možemo pisati izraz za model odsječka krive linije u opštem obliku

Derivacijom prethodnog izraza dobit će se izraz za vektor smjera tangente

Cjeloviti model željene krive linije tvori se sastavljanjem modela pojedinih odsječaka. Nivo glatkoće krive na spoju dvaju odsječaka izražava se u smislu dviju vrsta kontinuiteta:

- geometrijskog kontinuiteta G ,
- parametarskog kontinuiteta C .

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z$$

$$T = \begin{bmatrix} t^3 & t^2 & t \end{bmatrix}$$

$$C = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}$$

$$\mathbf{Q}(t) = \mathbf{T} \mathbf{C}.$$

$$\frac{d}{dt} Q(t) = Q'(t) = \begin{bmatrix} 3t^2 & 2t & 1 & 0 \end{bmatrix}$$

Geometrijski kontinuitet definisan je na sljedeći način:

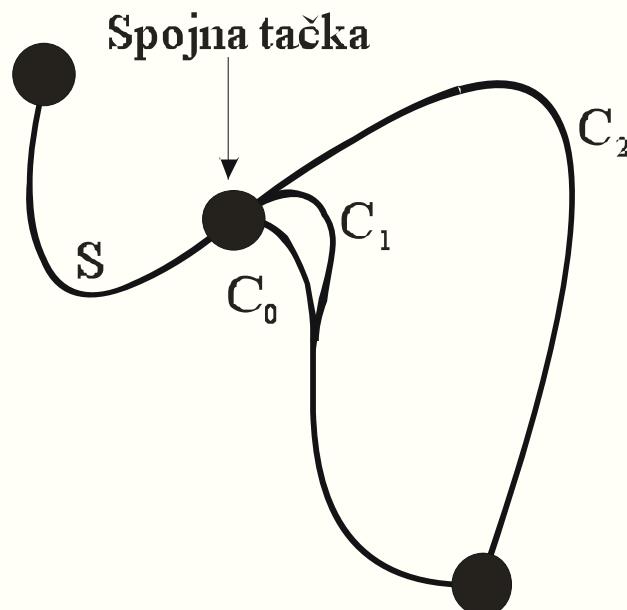
- geometrijski kontinuitet G^0 - neprekinutost krive u tački dodira odsječaka,
- geometrijski kontinuitet G^1 - jednakost vektora smjera tangente u tački dodira odsječaka.

Parametarski kontinuitet definisan je na sljedeći način:

- parametarski kontinuitet C^1 - jednakost parametara t u točki dodira odsječaka,
- parametarski kontinuitet C^n - jednakost n -te derivacije $Q(t)$ u točki dodira odsječaka.

Odnos parametarskog i geometrijskog kontinuiteta može se sažeti na sljedeći način:

$$C^1 \Rightarrow G^1$$



Primjeri krivih linija različitih nivoa glatkoće izraženih u smislu parametarskog kontinuiteta C^0 , C^1 , C^2 .

Definicija odsječka krive linije Q(t)

Polinom trećeg stepena kao model odsječka krive linije ima 4 nepoznata koeficijenta što zahtijeva 4 uslova za njihovo određivanje.

Na taj način dobija se sistem od ukupno 4 jednačine s 4 nepoznate.

Uslovi mogu biti: krajnje tačke, vektor smjera tangente ili kontinuitet u tačkama dodira pojedinih odsječaka.

S obzirom na izbor vrste uslova definisane su različite vrste krivih linija. Osnovne vrste krivih linija su;

- **Hermiteove krive** (*uslovi su:* dvije krajnje tačke i dva vektora smijera u krajnjim tačkama),
- **Bezierove krive** (*uslovi su:* dvije krajnje tačke i dvije dodatne tačke koje određuju vektore smijera u krajnjim tačkama),
- **B-krive i b-krive** (*uslovi su:* četiri kontrolne tačke)

Hermiteove krive i Bezierove krive zadovoljavaju kriterije G^1 i C^1 kontinuiteta uz određene uslove, dok B-krive i b-krive zadovoljavaju kriterije C^1 , C^2 kontinuiteta.

Matrica koeficijenata C može se izraziti kao umnožak bazne matrice M i geometrijskog vektora G koji sadrži zadate geometrijske uslove (npr. koordinate točaka):

$$\mathbf{C} = \mathbf{M}\mathbf{G}$$

Izraz za parametarski model odsječka krive tada možemo napisati na slijedeći način:

$$\mathbf{Q}(t) = \mathbf{T} \mathbf{M} \mathbf{G}$$

Posmatramo li samo jednu komponentu vektora $\mathbf{Q}(t)$ npr. $x(t)$ dobivamo slijedeći izraz:

$$x(t) = \mathbf{T} \mathbf{M} G_x$$

Onačimo li umnožak vektora T i matrice M s B ($B=TM$) možemo pisati :

$$Q(t) = B G$$

Elementi matrice B su polinomi trećeg reda parametra t . Na taj način vidimo da je kriva predstavljena kao težinski zbir elemenata geometrijskog vektora, gdje su težinski faktori polinomi parametra t .

Za svaku pojedinu vrstu krive definisana je bazna matrica M i geometrijski vektor G

Svaki skup krivih predstavljenih matričnim umnoškom $Q(t)=TMG$ moguće je međusobno pretvarati iz jednih u druge npr. krive predstavljene baznom matricom M , i geometrijskim vektorom G_1 , moguće je predstaviti drugom baznom maticom M_2 i odgovarajućim geometrijskim vektorom G_2 . Važno je da mora biti ispunjen uslov:

$$M_2 G_2 = M_1 G_1$$

Nepoznati geometrijski vektor G_2 može se izračunati na slijedeći način:

$$G_2 = M_2^{-1} M_1 G_1$$

Na taj način može se, npr. kriva predstavljena baznom matricom i geometrijskim vektorom za Hermiteove krive, transformirati u prikaz baznom matricom i geometrijskim vektorom za Bezierove krive.

Različite vrste krivih imaju različite prednosti za pojedine vrste primjena. Prednosti različitih vrsta krivih mogu se najbolje iskoristiti u kombiniranom načinu prikaza krivih.

Kriteriji za izbor vrste krive uključuju:

- prikladnost za interaktivnu manipulaciju,
- stepen kontinuiteta,
- opštenitost,
- brzinu proračuna.

Bezierove krive definisane su slijedećim geometrijskim uslovima:

- dvije krajnje točke P_1 i P_2 .
- dvije kontrolne točke P_3 i P_4 koje određuju vektore smjera u krajnjim tačkama R_1 i R_4 .

Pamoću dviju kontrolnih tačaka posredno su definisani vektori smjera tangenti R_1 i R_4 u dvjema krajnjim tačkama. Vektor smjera tangente u početnoj tačci odgovara derivaciji krive $\mathbf{Q}(t)$ za vrijednost parametra $t=0$, dok vektor smjera tangente u krajnjoj tačci odgovara derivaciji krive $\mathbf{Q}(t)$ za vrijednost parametra $t=1$:

$$R_1 = \mathbf{Q}'(0) = 3(P_2 - P_1)$$

$$R_4 = \mathbf{Q}'(1) = 3(P_4 - P_3)$$

Elementi geometrijskog vektora su četiri zadate tačke. Geometrijski vektor za Bezierove krive definisan je na slijedeći način:

$$\mathbf{G}_B = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

Bazna matrica za Bezierove krive definisana je na slijedeći način:

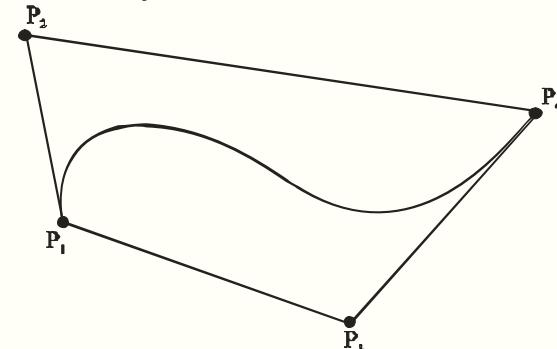
$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

imajući u vidu da je odsječak krive opisan izrazom: $Q(t) = TM_B G_B$

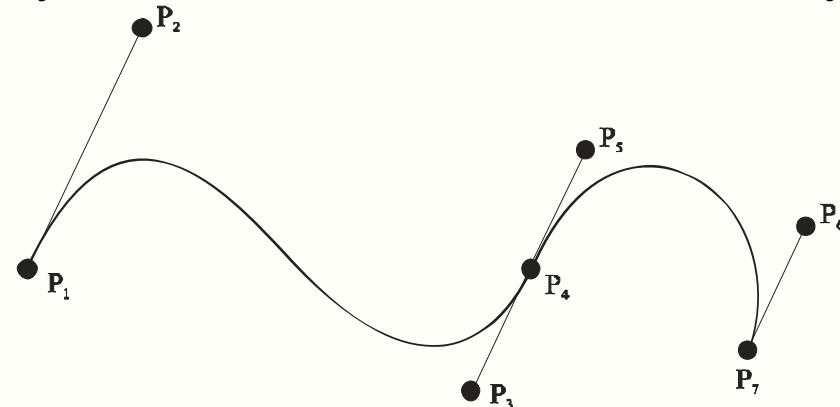
uvrštavanjem bazne matrice i geometrijskog vektora dobivamo slijedeći oblik jednačine odsječka Bezierove krive:

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4$$

Polinomi koji predstavljaju koeficijente pojedinih tačaka u ovom izrazu nazivaju se Bernsteinovi polinomi



a) Primjer crtanja odsječaka Bezierove krive



b) Primjer dvaju spojenih odsječaka Bezierove krive

Uslov za G_1 kontinuitet jest da tačke P_3 , P_4 i P_5 moraju biti različite i kolinearne

$$P_3 - P_4 = k(P_4 - P_5), \quad k > 0$$

Uslov za C^1 kontinuitet jest da je $k = 1$.

Načini crtanja parametarskih krivulja

Postupak crtanja parametarskih krivih uključuje određivanje diskretnog skupa tačaka na modelu krive i iscrtavanje ravnih linija koje ih povezuju. Pri tome su mogući različiti pristupi uključujući:

- iterativni proračun $x(t)$, $y(t)$ i $z(t)$ za niz bliskih rastućih vrijednosti parametra t ,
- rekurzivna podjela do zadovoljavajuće bliskosti kontrolnih tačaka na samoj krivoj,
- hibridni pristup kao kombinacija iterativnog i rekurzivnog načina.

Iterativni proračun odvija se na način da se vrijednosti koordinanata $x(t)$, $y(t)$, i $z(t)$ pojedinih tačaka izračunavaju za niz vrijednosti parametra t međusobno udaljenih za unaprijed određeni konstantan iznos d .

Proračunate tačke spajaju se ravnim linijama.

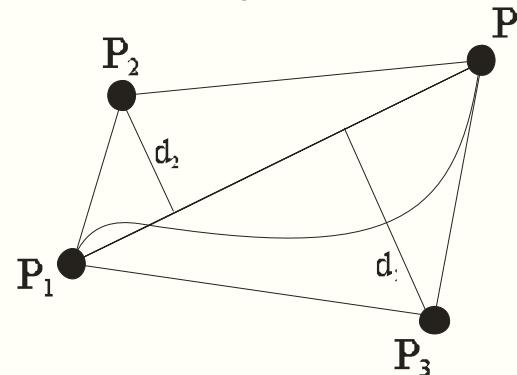
Problem ovakvog pristupa je u tome što se unaprijed treba odrediti razmak tačaka s obzirom na parametar t . Prevelik razmak rezultira slabim kvalitetom aproksimacije dok premalen razmak rezultira nepotrebnim proračunskim opterećenjem.

Rekurzivna podjela odvija se na način da se između dvije izračunate tačke umeće treća tačka.

Rekurzivna podjela zaustavlja se adaptivno kada odsječak krive postane dovoljno ravan da se može aproksimirati ravnom linijom. Pojedinosti postupka različite su za pojedine vrste krivih. Ovaj pristup je posebno zgodan za Bezierove krive.

Kriterij ravnoće je da udaljenosti unutrašnjih kontrolnih tačaka do spojnica krajnjih tačaka moraju biti manje od zadane vrijednosti

Hibridni pristup kombinuje najbolja svojstva iterativnog i rekurzivnog pristupa. U osnovi se može opisati kao iterativni proračun s adaptivnim korakom.



Primjer primjene kriterija ravnoće pri rekurzivnom pristupu crtanja Bezierovih

MATEMATIČKE OSNOVE CRTANJA POVRŠINA

U mnogim primjenama računarske grafike potrebno je prikazati različite vrste površina u trodimenzionalnom prostoru.

Najčešće korišteni načini modeliranje površina zasivaju se na primjeni:

- parametarskih površina,
- mreža mnogougla,
- površina drugog reda.

Parametarske površine predstavljaju generalizaciju parametarskih krivih na način da se umjesto jednog parametra koriste dva.

Mreža mnogougla predstavlja skup povezanih mnogougaonih ravnih površina kojima se aproksimira željena površina proizvoljnog oblika.

Površine drugog reda su površine definisane opštom jednačinom drugog reda u implicitnom obliku. U nastavku ćemo definisati ove načine modeliranja površina u trodimenzionalnom prostoru.

DVOPARAMETARSKE POVRŠINE TREĆEG REDA

Dvoparametarske površine trećeg reda predstavljaju generalizaciju parametarskih krivih trećeg reda.

U jednačini parametarske krive $Q(t)=TMG$ parametar t zamijenimo parametrom s , a zatim učinimo geometrijski vektor G promjenljivim u 3D prostoru duž nekog puta u zavisnosti od parametra t ,

$$Q(s,t) = SMG(t) = SM \begin{bmatrix} G_1(t) \\ G_2(t) \\ G_3(t) \\ G_4(t) \end{bmatrix}$$

Za izabranu vrijednost $t=t_1$, izraz $\mathbf{Q}(s,t_1)$ definiše krivu trećeg reda. Za niz bliskih vrijednosti parametra t izraz $\mathbf{Q}(s,t)$ definiše niz bliskih krivih trećeg reda koje definišu površinu. $G_i(t)$ su krive trećeg reda:

$$G_i(t) = TMG_i$$

gdje je:

$$G_i = [g_{i1} \quad g_{i2} \quad g_{i3} \quad g_{i4}]^T$$

Uvrštenjem ovih izraza za $G_i(t)$ i G_i u izraz za $Q(s,t)$ dobija se sljedeći oblik:

$$Q(s,t) = SM \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix} M^T T^T$$

koji se može sažeto napisati u slijedećem obliku:

$$Q(s,t) = SMGM^T T^T$$

Ova jednačina može se ispisati i posebno za svaku koordinatu:

$$x(s,t) = SMG_x M^T T^T$$

$$y(s,t) = SMG_y M^T T^T$$

$$z(s,t) = SMG_z M^T T^T$$

Bezierove površine

Ako u opštem izrazu za dvoparametarsku krivu trećeg reda uvrstimo Bezierovu baznu matricu M_B i Bezierovu geometrijsku matricu G_B dobivamo izraze za Bezierove površine:

Bezierova geometrijska matrica sastoji se od 16 kontrolnih tačaka.

16 kontrolnih tačaka Bezierove površine.

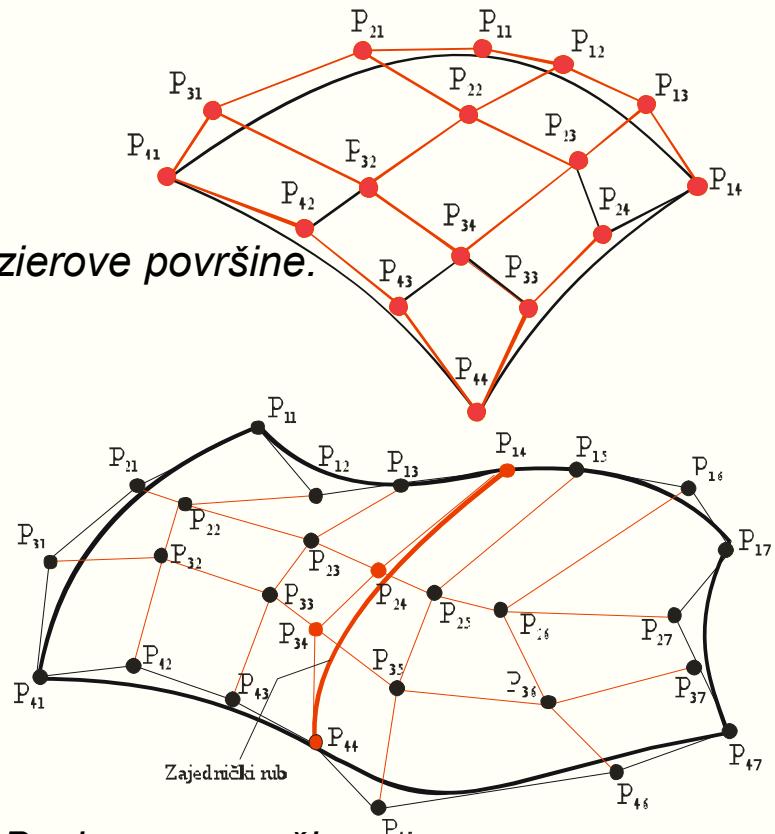
Bezierove površine su praktične za interaktivno crtanje (jer su vektori smjera tangenti u krajnjim tačkama eksplisitno izraženi). Uslov za C^0 i G^0 kontinuitet na spojevima dijelova površina je jednakost rubnih kontrolnih tačaka. Uslov za G^1 kontinuitet je kolinearnost četiri skupa tačaka koji uključuju rubne i susjedne kontrolne tačke.

Spoj dvaju segmenata Bezierove površine.

$$x(s, t) = S M_B G_{Bx} M_B^T T^T$$

$$y(s, t) = S M_B G_{By} M_B^T T^T$$

$$z(s, t) = S M_B G_{Bz} M_B^T T^T$$



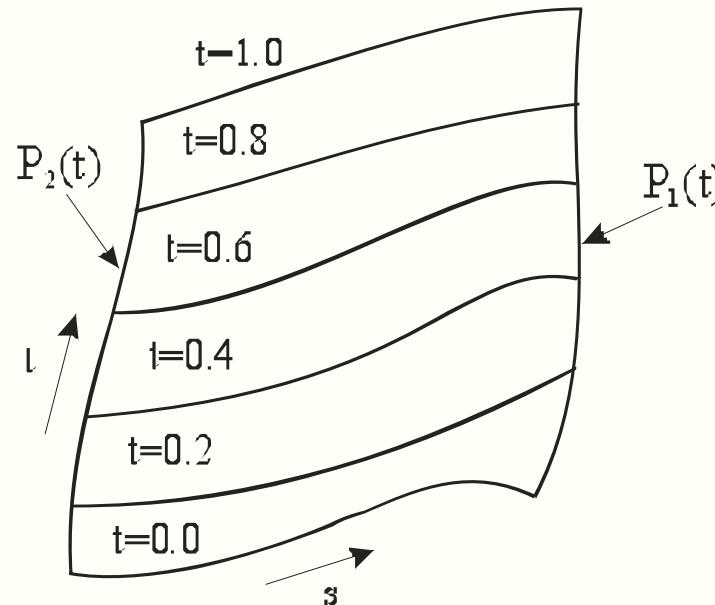
Načini crtanja dvoparametarskih površina trećeg reda

Kao i u slučaju crtanja parametarskih krivih, i osnovni pristupi crtaju parametarskih površina uključuje iterativni i rekurzivni postupak.

- **Iterativni postupak** podrazumijeva proračun polinoma trećeg reda za niz bliskih rastućih vrijednosti parametara t i s .

Mogući pristup je iscrtavanje niza krivih kao funkcija parametra t (dok se vrijednost parametra s drži konstantnom za pojedinu krivu, a mijenja se od krive do krive).

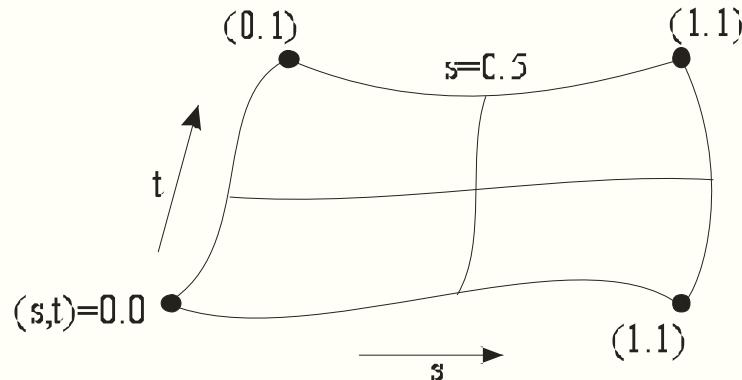
Zatim se iscrtava niz krivih kao funkcija parametra s (dok se vrijednost parametra t drži konstantnom za pojedinu krivu, a mijenja se od krivu do krive).



Postupak iterativnog pristupa crtaju površine.

□ **Postupak rekurzivne podjele** podrazumijeva dijeljenje površine na četverouglove (omeđene krivama) do zadovoljavajuće ravnoće rezultirajućih četveruglova.

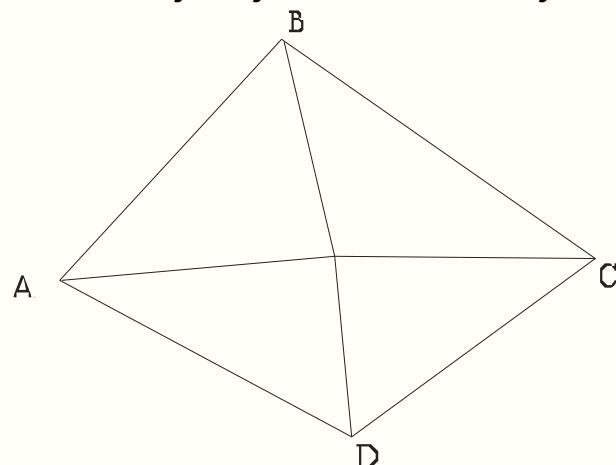
- Podjela se obavlja dijeljenjem prvo po jednom, a zatim po drugom parametru.
- Dijeljenje se obavlja crtanjem krive po jednom parametru uz konstantnu vrijednost drugog parametra.
- Konstantna vrijednost drugog parametra određuje se kao aritmetička sredina između vrijednosti tog parametra koje su korištene u prethodnoj podjeli.



*Postupak rekurzivnog pristupa
crtanju površine.*

Ovaj postupak najjednostavniji je za slučaj Bezierovih površina.

Četverougao zadovoljavajuće ravnoće dijeli se na četiri trougla koji se ispunjavaju

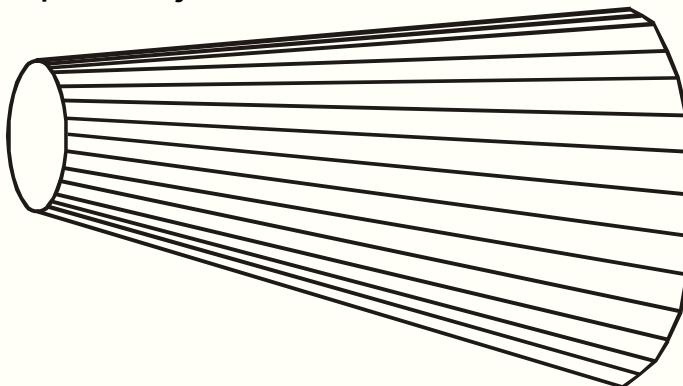


Četverougao se dijeli na četiri trougla određivanjem tačke P čije su koordinate aritmetičke sredine odgovarajućih koordinata vrhova A, B, C i D.

MREŽE MNOGOUGLOVA

Mreža mnogougla je skup bridova, vrhova i mnogougla povezanih tako da oblikuju površinu koja dovoljno dobro aproksimira željenu površinu.

- Mnogougao je zatvoreni skup povezanih bridova (stranica).
- Svaki brid zajednički je za najviše dva mnogougla, a svaki vrh zajednički je za najmanje dva brida.
- Svaki brid povezuje dva vrha.



Primjer izgleda trodimenzionalnog objekta pomoću mreže mnogougla.

Tipične vrste operacija koje je potrebno obavljati pri crtanjtu mreže mnogougla uključuju:

- određivanje svih bridova koji izlaze iz promatranoog vrha,
- određivanje svih mnogouglova kojima je promatrani brid ili vrh zajednički,
- određivanje svih vrhova kojima je promatrani brid zajednički i
- iscrtavanje mreže.

Najčešće korišteni načini prikazivanja modela mreže mnogougla su:

- eksplicitni način,
- struktura pokazivača na listu vrhova,
- struktura pokazivača na listu bridova.

U **eksplicitnom prikazu** svaki je mnogougao predstavljen listom koordinata vrhova:

$$P = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n))$$

Vrhovi su u ovom prikazu poređani redom iscrtavanja.

Nedostatak ovog pristupa je u tome što se vrhovi pojavljuju više puta jer ne postoji eksplisitna predstava zajedničkih vrhova.

Npr. za pomicanje (npr. pomoću miša) jednog vrha nužno je identificirati sve mnogouglove povezane s tim vrhom.

U tu svrhu nužno je uporediti sve koordinate vrhova u jednom mnogouglu s koordinatama vrhova u svim drugim mnogouglovima.

Pored toga i svi zajednički bridovi se iscrtavaju dva puta.

U prikazu zasnovanom na **strukturi pokazivača na listu vrhova** svaki vrh se pohranjuje samo jednom u listu vrhova.

$$V = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n))$$

Mnogougao se definiše listom pokazivača (indeksa) u listu vrhova (ova metoda koristi se u SPHIGS standardu).).

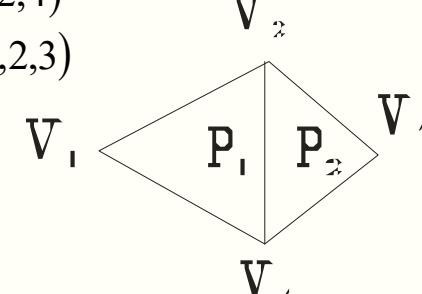
Memorijski zahtjevi kod ovog načina su znatno manji nego u eksplisitnom prikazu. Osim toga i promjena koordinata jednog vrha je jednostavna jer nije potrebno upoređivanje i pretraživanje.

Međutim, u ovom prikazu teško je odrediti mnogouglone koji imaju zajednički vrh, a i zajednički bridovi se iscrtavaju dva puta.

$$V = (V_1, V_2, V_3, V_4) = (x_1, y_1, z_1), \dots, (x_4, y_4, z_4)$$

$$P_1 = (1, 2, 4)$$

$$P_2 = (4, 2, 3)$$



Mreža mnogougla definisana pokazivačima u listu vrhova.

Navedeni problemi otklonjeni su u prikazu zasnovanom na **strukturi pokazivača u listu bridova**. U ovom prikazu svaki se brid unosi jednom.

$$V = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n))$$

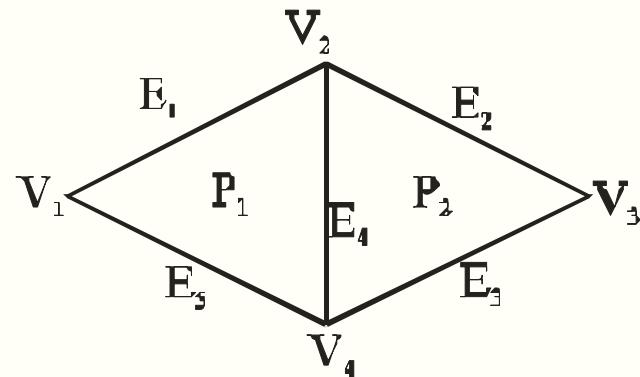
U ovom prikazu svaki se brid unosi jednom.

$$P = (E_1, E_2, E_3, \dots)$$

i to na sljedeći način:

$$E = (V_1, V_2, P_1, P_2, \dots)$$

gdje su prva dva elementa koordinate vrhova koji određuju brid, a druga dva elementa oznake mnogouglova kojima je taj brid zajednički.



Mreža mnogougla definisana pokazivačima u listu bridova.

$$V = (V_1, V_2, V_3) = (x_1, y_1, z_1) \dots (x_4, y_4, z_4)$$

$$E_1 = (V_1, V_2, P_1, \lambda)$$

$$E_2 = (V_2, V_3, P_2, \lambda)$$

$$E_3 = (V_3, V_4, P_2, \lambda)$$

$$E_4 = (V_4, V_2, P_1, \lambda)$$

$$E_5 = (V_4, V_1, P_1, \lambda)$$

$$P_1 = (E_1, E_4, E_5)$$

$$P_2 = (E_2, E_3, E_4)$$

POVRŠINE DRUGOG REDA

Površine drugog reda kao što su površina kugle, površina elipsoida ili cilindra mogu poslužiti kao elementarne površine za definisanje modela složenijih površina u trodimenzionalnom prostoru

Ova grupa površina definirana je implicitnom jednačinom oblika:

$$f(x, y, z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k = 0$$

Vrsta površine određena je vrijednostima parametara $a, b, c, d, e, f, g, h, i, j$ i k .

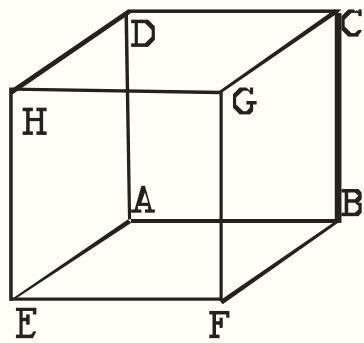
Prednosti ovog pristupa su prikladnost za niz proračuna kao što su *normale na površinu* u pojedinoj tačci, *određivanje presjeka*, te određivanje *pripadnosti tačke površini*. Ove prednosti zasnivaju se na jasnoj matematičkoj formi definicije površine.

CRTANJE TRODIMENZIONALNIH TIJELA

U mnogim primjenama važno je stvoriti **model trodimenzionalnog objekta** i grafički prikaz na kojem se jasno uočava razlika između unutrašnjosti i vanjštine tijela ili odnos između dvaju objekata u prostoru.

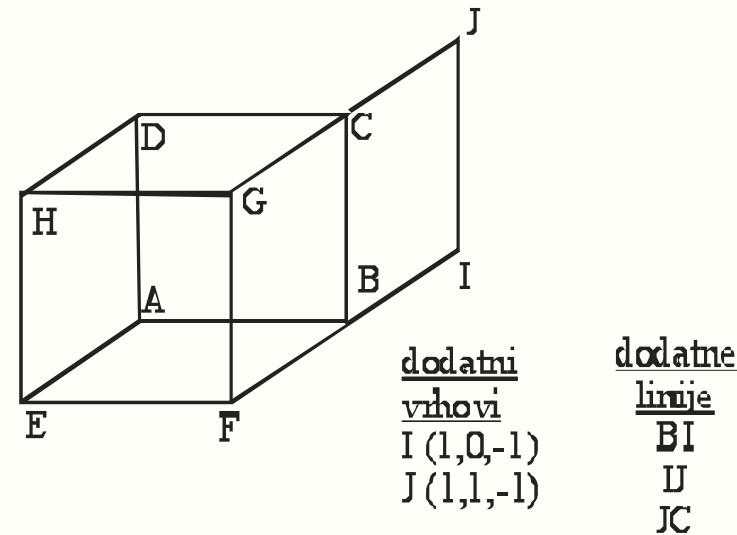
Elementi koje pri modeliranju i grafičkom prikazivanju tijela možemo koristiti su ravne linije, krive linije, mnogouglovi i površine.

Jedan od pristupa modeliranju tijela je primjena **žičanih modela** kod kojih se linijama modeliraju bridovi tijela.



a)

<u>Vrhovi</u>	<u>Linije</u>
A (0,0,0)	AB
B (1,0,0)	BC
C (1,1,0)	CD
D (0,1,0)	DA
E (0,0,1)	EF
F (1,0,1)	FG
G (1,1,1)	GH
H (0,1,1)	HE
	AE
	BF
	CG
	DH

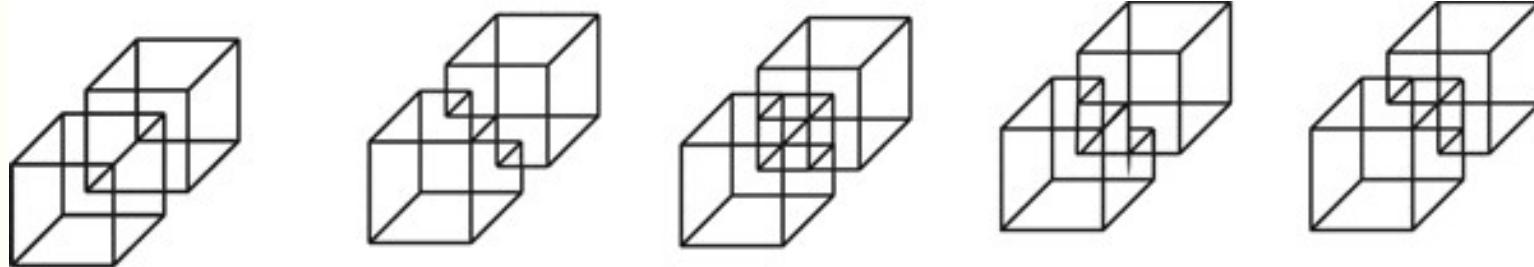


b)

Žičani model kocke: a) s prikazanim svim bridovima, b) s prikazanim samo vidljivim bridovima.

Neovisno o načinu prikazivanja trodimenzionalnih objekata potrebno je definisati mogućnost njihovog kombinovanja sa svrhom stvaranja novih oblika.

Jedna od najintuitivnijih i najpopularijih metoda za kombinovanje objekata je primjena Booleovih operacija uključujući: uniju, presjek i razliku.



Primjer Booleovih operacija nad trodimenzionalnim objektima

10. ANIMACIJA

Pojam animacija podrazumijeva sve promjene koje imaju vidljivi učinak, a to uključuje vremensku promjenu:

- položaja (kretanje),
- oblika,
- boje,
- transparentnosti,
- strukture,
- osvjetljenja,
- položaja gledišta,
- žarišta i drugo.

Animacija se primjenjuje u edukaciji, industriji zabave, industrijskim upravljačkim sistemima, naučnoistraživačkom radu i drugim područjima

Poseban naziv - **naučna predodžba** (*scientific visualization*) označava primjenu računarske grafike uključujući *animaciju*, *obradu signala*, *računarsku geometriju* i *teoriju baza podataka* za naučnoistraživačke namjene.

Složeni *simulacijski postupci*, *proračuni* i *mjerjenja* mogu rezultirati velikim količinama podataka čija preglednost se značajno može poboljšati primjenom računarske grafike.

Osnovni postupci u animaciji uključuju sljedeće korake:

- na osnovu zamisli i skice kreira se niz ključnih slika (key frames) u kojima su animirani entiteti u ekstremnim ili karakterističnim položajima,
- na temelju ključnih slika stvara se niz međuslika i tako upotpunjuje niz (ovaj postupak prikladan je za primjenu računara),
- prije primjene računara nužna je digitalizacija slike (optičkim skaniranjem, digitalizatorskom pločom ili crtanjem pomoću računara),

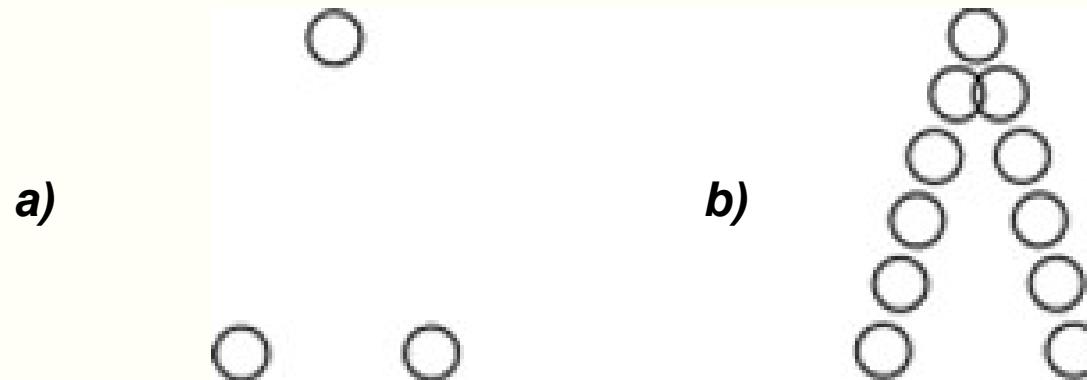
INTERPOLACIJA

Postupak stvaranja među slika pomoću računara provodi se primjenom tehnika **interpolacije**.

Najjednostavniji pristup stvaranju međuslika zasniva se na primjeni linearne interpolacije (ovaj postupak na engleskom se ponekad naziva **lerping - linear interpolation**).

Na osnovu poznatih vrijednosti nekog atributa u krajnjim tačkama v_p i v_k (položaj, boja, veličina) vrijednost atributa v_m u nizu međuslika može se izračunati na sljedeći način:

$$v_m = (1-t)v_p + tv_k \quad 0 < t < 1$$



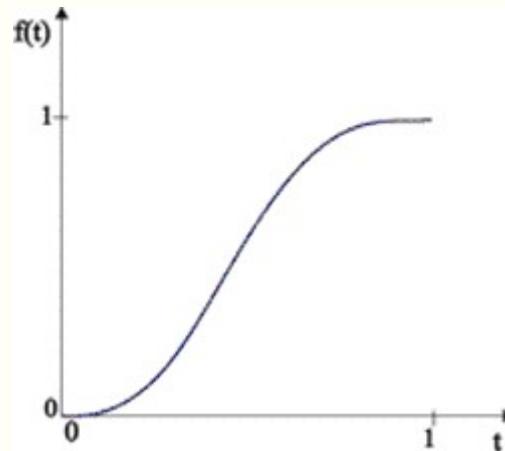
Neprirodnost animacije zasnovane na linearnoj interpolaciji posebno se ogledaju u najvišoj tačci putanje gdje dolazi do promjene smjera kretanja.

Ograničenja linearne interpolacije mogu se izbjegići primjenom **nelinearne interpolacije** zasnovane na nekoj krivoj $f(t)$ s konstantnom derivacijom u sredini i nultom derivacijom na krajevima.

Primjena takve krive osigurava kontinuitet derivacije u krajnjim tačkama putanje i znatno prirodniji izgled animiranog kretanja.

Vrijednost parametra u međuslici tada se izračunava u skladu sa sljedećim izrazom

$$v_m = (1 - f(t))v_p + f(t)v_k \quad 0 < t < 1$$



Oblik krive prikladan za nelinearnu interpolaciju

Poseban problem pojavljuje se pri animaciji orientacije trodimenzionalnog tijela.

Ako se orientacija specificira uglovima zakretanja oko tri glavne osi, poznatim kao **Eulerovi uglovi**, važan je redoslijed specifikacije pojedinih uglova, odnosno redoslijed zakretanja oko pojedinih osi.

Zamjenom redoslijeda zakretanja oko pojedinih osi dobiju se različite orientacije tijela u prostoru.

Zbog toga interpolacija Eulerovih uglova rezultira neprirodnim interpolacijama orientacija u prostoru.

Rješenje ovog problema je u primjeni **algebarske strukture kvaterniona** umjesto Eulerovih uglova.

Kvaternioni su simboli oblika:

$$a + bi + cj + dk,$$

gdje su a, b, c i d realni brojevi koji zadovoljavaju uslov: $a^2 + b^2 + c^2 + d^2 = 1$

ANIMACIJSKI JEZICI

Razvijen je veći broj animacijskih jezika koji se opštenito mogu svrstati u tri grupe:

- zapis u obliku linearnih lista,
- jezici opšte namjene s ugrađenim animacijskim naredbama,
- grafički jezici.

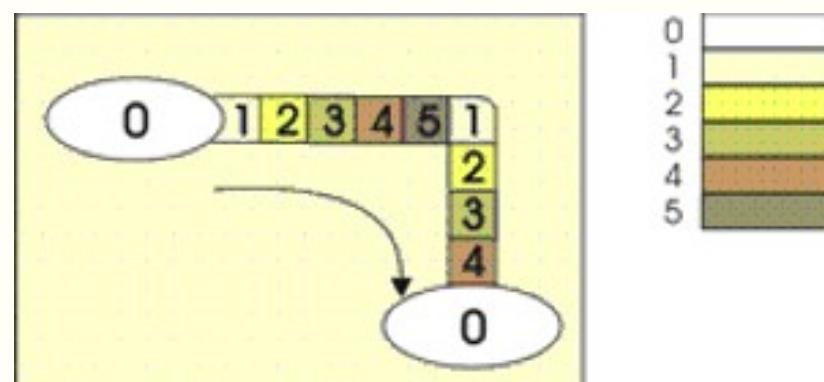
Zapis u obliku linearnih lista svaki događaj u animaciji opisuju rednim brojem početne i krajnje slike te djelovanjem npr.

42, 53, B ROTATE "SLIKA1", 1, 30

znači: između slika 42 i 53 zakrenuti objekt SLIKA1 oko osi 1 za 30 stepeni, pri čemu se zakretanje za svaku pojedinu sliku određuje iz tablice B.

Jezici opšte namjene s ugrađenim animacijskim naredbama imaju važnu prednost što se vrijednosti varijabli u jeziku koriste kao parametri za rutine koje generiraju animaciju. Posebno je zanimljiva mogućnost upotrebe jezika višeg nivoa za generiranje simulacije pri čemu se generira i animacija kao uporedni proces.

Grafički animacijski jezici zasnivaju se na vizualnom pristupu i direktnom stvaranju animacije umjesto pisanja niza naredaba. Prednost u odnosu na tekstualne jezike je to što animator može direktno promatrati učinak napisanih naredaba.



11. BOJA U RAČUNARSKOJ GRAFICI

Boja je vrlo kompleksan pojam koji se izučava s STANOVIŠTA fizike, psihologije, umjetnosti i grafičkog dizajna.

Primjena boje u računarskoj grafici značajno povećava mogućnosti i kvalitetu predstave složenijih objekata.

Boja nekog objekta ne zavisi isključivo o samom objektu nego i o izvoru svjetlosti, boji okoline i čovjekovog sistemu vida.

Neki objekti odbijaju svjetlost, a neki je propuštaju. Površina koja odbija samo plavu svjetlost, obasjana crvenom svjetlošću izgleda crna. Zelena svjetlost propuštena kroz staklo koje propušta samo crvenu svjetlost također izgleda crna.

Doživljaj boje povezan je s pojmom svjetlosti.

Razlikujemo dvije osnovne grupe svjetlosti:

- **akromatsku** (bezbojnu) i
- **kromatsku** (obojenu) svjetlost.

Primjena ovih dviju vrsta svjetlosti odnosno pripadnih boja na različit način se tehnološki podržava u računarskoj grafici, ali i u drugim područjima i medijima vizualne komunikacije kao što su televizija i štampa.

AKROMATSKA SVJETLOST

Akromatsku svjetlost doživljavamo kao **crnu, bijelu i sivu** boju.

Primjeri primjene akromatske svjetlosti su *crno-bijela televizijska slika, crno-bijela slika na monitoru računara i crno-bijela štampa*. Jedini atribut takve svjetlosti jest količina svjetlosti.

Količina svjetlosti u fizikalnom smislu odgovara energiji, a opisuje se veličinama **intenzitet i osvjetljenost** (luminance).

U psihološkom smislu opisuje se kao intenzitet osjeta i naziva se **sjajnost** (brightness).

Kao mjera intenziteta definisana je skalarna veličina:

- **0** koja odgovara crnoj svjetlosti,
- vrijednosti **1** bijela svjetlost,
- dok vrijednostima između **0** i **1** odgovaraju različiti nivoi sive boje.

Crno bijela televizija ili monitor može proizvesti različite nivoe sive boje na istom pikselu.

Crno-bijeli printeri i ploteri mogu proizvesti samo dva nivoa: crnu ili bijelu na jednom mjestu.

Privid različitih nivoa sive boje na takvim uređajima sa dva nivoa može se ostvariti primjenom tehnike polutoniranja (halftoning).

POLUTONSKA APROKSIMACIJA

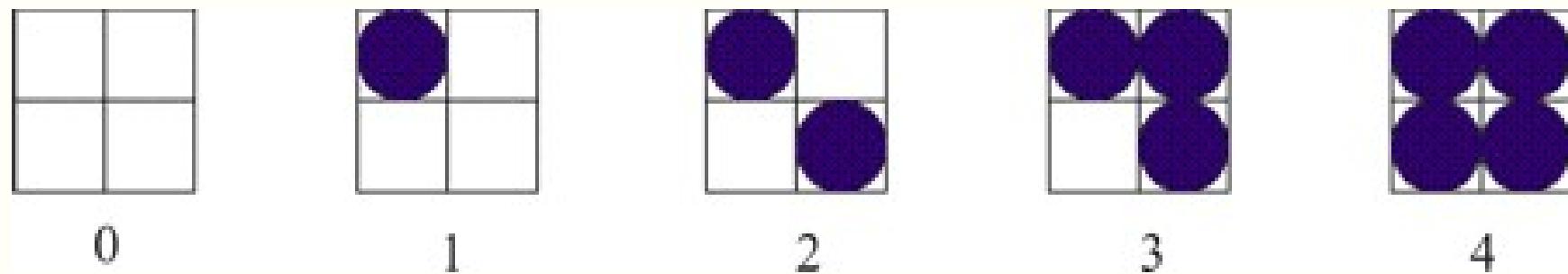
Polutonska aproksimacija omogućava prikazivanje nivoa sive boje na crno-bijelim uređajima sa dva nivoa.

Tehnika polutoniranja zasniva se na osobini prostorne integracije prirođenom ljudskom vidu.

Ako malu površinu gledamo iz dovoljno velike udaljenosti ljudske oči usrednjavaju boju pojedinih dijelova te površine i stvaraju doživljaj ukupnog intenziteta svjetlosti čitave površine.

U grafičkim uređajima ova se osobina koristi na način da se elementarnim djelovima prikazne površine (pikselima) pridjeljuje crna ili bijela boja tako da raspored crnih i bijelih elemenata stvara utisak određenog nivoa sive boje na površini određenoj grupom elemenata.

Grupa od $n \times n$ piksela sa dva nivoa može odražavati ukupno n^2+1 različitih nivoa sive boje (odnosno intenziteta). Tako površina određena s 2×2 piksela na taj način može odražavati 5 različitih nivoa sive boje



Pet nivoa sive boje izraženih tehnikom polutoniranja na površini od 2×2 piksela.

KROMATSKA SVJETLOST

Vizualni osjećaji koje izaziva kromatska svjetlost znatno su bogatiji i raznovrsniji od onih izazvanih akromatskom svjetlošću.

Doživljaj boje uobičajeno se opisuje s tri veličine:

- **nijansa** (hue), koja opisuje vrstu boje npr. crvena, zelena, žuta...
- **zasićenje** (saturation) koje opisuje udaljenost boje od sive boje istog intenziteta npr. crvena boja je vrlo zasićena, ružičasta je manje zasićena.
- **osvjetljenost** (lightness) koja opisuje intenzitet svjetlosti reflektirane od objekta.

Ponekad se umjesto osvjetljenosti koristi sjajnost (brightness), posebno kad se radi o objektima koji su izvori svjetlosti kao npr. sijalica.

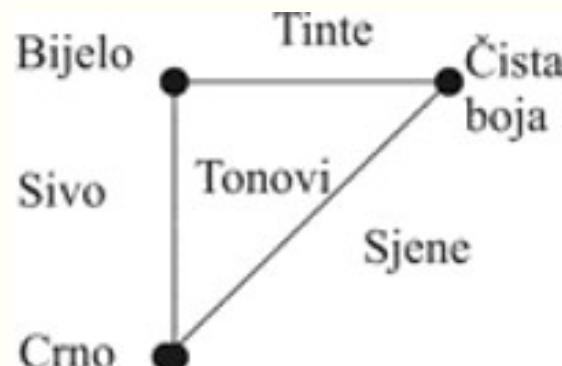
Umjetnici često specificiraju boje kao **tinte** (tint), **sjene** (shade) i **tonove** (tone) jako zasićenih čistih pigmenata

Tinta se dobija dodavanjem bijelog pigmenta čistom pigmentu čime se smanjuje zasićenost.

Sjena se dobija dodavanjem crnog pigmenta čistom pigmentu čime se smanjuje osvjetljenost.

Ton se dobija dodavanjem bijelog i crnog pigmenta čistom pigmentu.

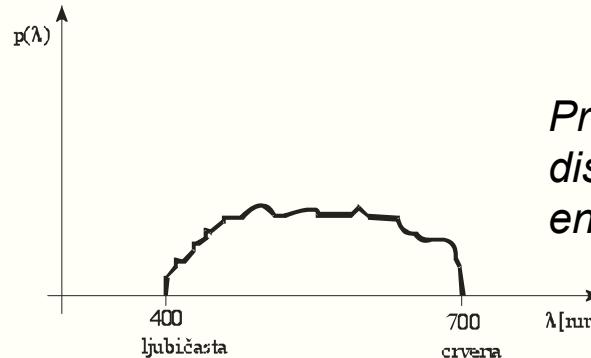
Ovi postupci proizvode različite boje iste nijanse, a različite zasićenosti i osvjetljenosti



Odnos među parametrima kojima se boja specificira u umjetnosti.

Objektivan način opisa boja zasniva se na grani fizike koja se naziva **kolorimetrija**.

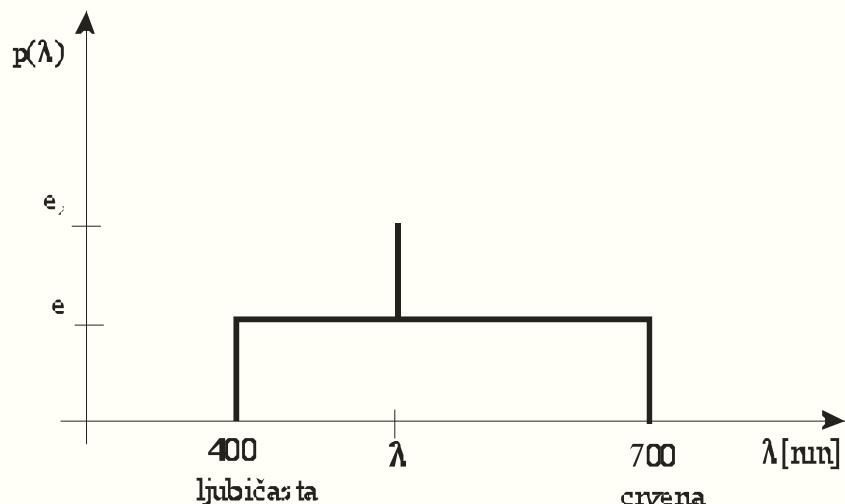
Fizikalno svjetlost predstavlja elektromagnetsku energiju u području talasnih dužina od 400 do 700 nm. Količina energije na pojedinoj talasnoj dužini opisuje se spektralnom distribucijom energije



Primjer spektralne distribucije energije svjetlosti.

Više različitih distribucija izazivaju percepciju iste boje. Spektralne distribucije koje izazivaju percepciju iste boje nazivaju se **metameri**.

Vizualni efekt spektralne distribucije se može opisati trima parametrima: **dominantnom talasnom dužinom, čistoćom pobude i količinom svjetlosti**



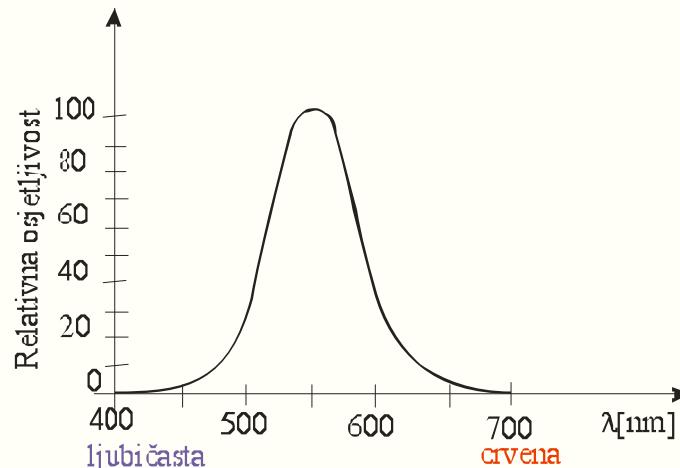
Ilustracija parametara spektralne distribucije svjetlosti

Na dominantnoj talasnoj dužini nalazi se izraženi vrh energije e_2 .

Čistoća pobude odgovara omjeru energije u području dominantne talasne dužine (e_2) i srednje vrijednosti spektralne gustoće energije (e_1).

Količina svjetlosti odgovara integralu umnoška spektralne distribucije i funkcije relativne spektralne osjetljivosti ljudskog oka.

Maksimum funkcije relativne spektralne osjetljivosti ljudskog oka nalazi se oko 550 nm (žuto-zelena boja)



Funkcija relativne spektralne osjetljivosti ljudskog oka.

Percepcijska veličina	Kolorimetrijska veličina
Nijansa	Dominantna valna duljina
Zasićenje	Čistoća pobude
Svjetloća	Intenzitet
Sjajnost	Intenzitet

Odnos između percepcijskih veličina koje opisuju svjetlost i kolorimetrijskih veličina

Miješanjem dviju boja dobija se nova boja. Međutim skup boja koje se mogu dobiti miješanjem dviju boja je ograničen.

Znatno zanimljiviji i širi skup boja može se dobiti miješanjem triju boja.

Uočeno je da se vrlo širok skup boja može dobiti ako se kao temeljne boje odaberu **crvena**, **zelena** i **plava** i taj pristup primijenjen je pri realizaciji *monitora u boji*.

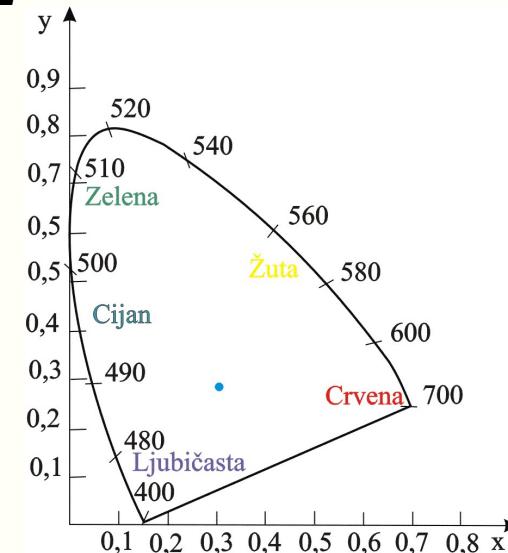
Upravljanjem intenzitetom svjetlosti triju komponenata (crvene, zelene i plave) određuje se boja svakog pojedinog piksela.

Međutim, na taj način ne mogu se prikazati sve vidljive boje jer bi za neke vidljive boje bilo potrebno koristiti negativne vrijednosti intenziteta crvene boje.

Mogućnost gradnje cjelovitog skupa vidljivih boja pomoću tri primarne komponente definisana je dijagramom kromaticiteta CIE (*Commision Internationale de l'Eclairage*). Primarne komponente označavaju se kao **X**, **Y** i **Z**, a njihovi udjeli u nekoj boji **C** kao X, Y i Z:

$$C = XX + YY + ZZ$$

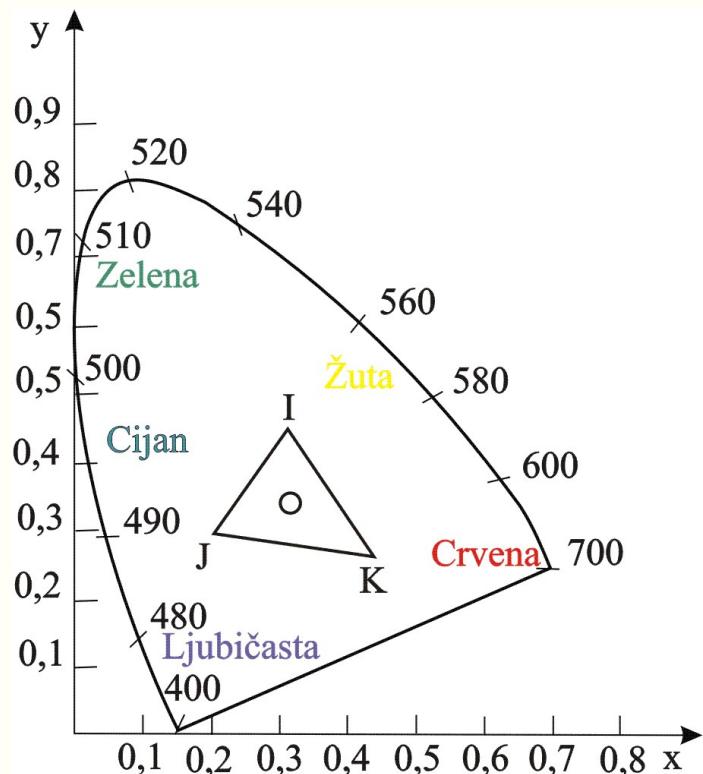
Dijagram kromaticiteta CIE dobija se normaliziranjem vrijednosti X, Y i Z dijeljenjem s njihovim zbirom ($X+Y+Z$), te projekcijom skupa tačaka vidljivih boja iz prostora xyz (gdje su x, y i z normalizirane vrijednosti veličina X, Y, i Z) na ravan xy.



CIE dijagram kromaticiteta

Sve vidljive boje smještene su na rubu i u unutrašnjosti zatvorene krive. Spektralno čiste boje, odnosno talasne dužine, odgovaraju tačkama smještenim na luku u smjeru kazaljke na satu. Bijela svjetlost odgovara tačci u sredini.

Područje boja (*color gamut*) koje se mogu dobiti miješanjem triju boja B_1 , B_2 i B_3 određeno je trokutom čiji su vrhovi u tačkama koje odgovaraju bojama B_1 , B_2 i B_3



Miješanjem boja B_1 , B_2 i B_3 mogu se dobiti sve boje iz trokuta $B_1 B_2$ i B_3 . Sve boje na crti $B_1 B_2$ mogu se dobiti miješanjem boja B_1 i B_2

MODELI BOJA U RAČUNARSKOJ GRAFICI

Model boje u računarskoj grafici odgovara specifikaciji trodimenzionalnog koordinatnog sistema i vidljivog podskupa u tom koordinatnom sistemu u kojem se nalaze sve boje iz određenog područja boja.

Svrha modela boje je prikidan način specifikacije boja iz određenog područja boja.

U računarskoj grafici u prvom redu su zanimljiva područja boja *CRT* (*Cathod Ray Tube*) monitora u boji koja su definisana primarnim bojama crvenom, zelenom i plavom.

Razvijene su dvije grupe modela boja.

- Prva grupa obuhvata modele koji su **sklopovski orijentisani**.
- Druga grupa modela obuhvata **korisnički orijentisane** modele

Primjeri sklopovski orijentisanih modela boje su **RGB** (*Red-Green-Blue*) za *CRT* monitore, te **CMY** (*cyan, magenta, yellow*) i **CMYK** (*cyan, magenta, yellow, black*) za tiskanje u boji.

Korisnički orijentisane modele su bliže načinu raspoznavanja svojstava boje od strane korisnika, a primjeri takvih modela su **HSV** (*nijansa-”hue”, zasićenost-”saturation”, sjajnost-”value”*), **HLS** (*hue, lightness, saturation*) i **HVC** (*hue, value, chromaticity*) modeli.

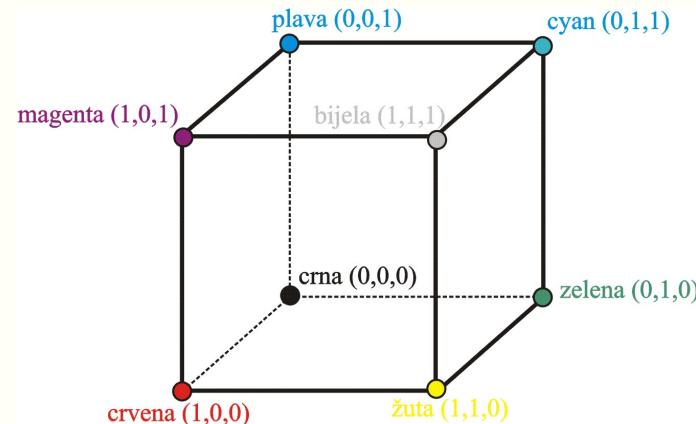
RGB MODEL BOJE

RGB model boje često se koristi u monitorima u boji i rasterskoj grafici.

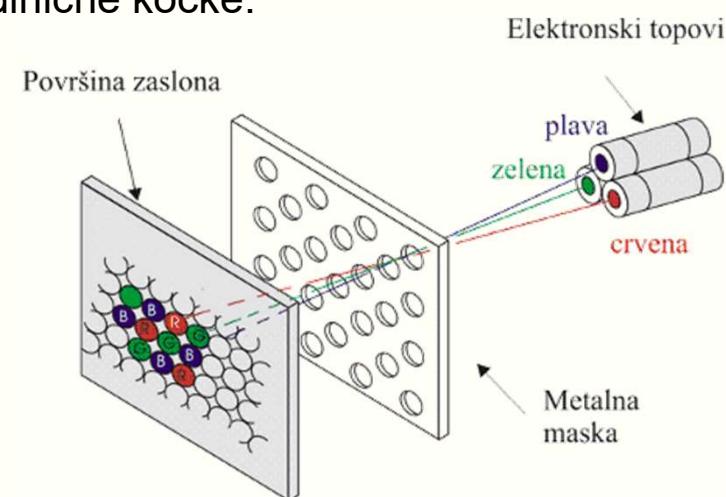
Definisan je u Kartezijevom koordinatnom sistemu.

Primarne veličine, **crvena, zelena i plava** boja, nazivaju se **aditivnim primarnim veličinama** jer se njihovi udjeli moraju sabrati da bi se dobila specificirana boja.

Područje u kojem su boje definisane je oblika jedinične kocke.



RGB model boje



Koncept RGB monitora.

U ishodištu je smještena crna boja, a na najudaljenijem vrhu bijela.

Prostorna dijagonala koja ih povezuje predstavlja nivo sivog.

Područje boja koje može prikazati CRT monitor određeno je svojstvima fosfora.

CRT monitori s različitim fosfornim slojem pokrivat će različita područja boja.

CMY MODEL BOJE

Cijan, magenta i žuta su komplementarne boje crvenoj, zelenoj i plavoj, respektivno.

Ove boje koriste se kao filtri za oduzimanje boja od bijele svjetlosti pa se nazivaju **subtraktivnim primarnim veličinama**. Na primjer, površina na koju je nanesen sloj cijan boje, ne reflektira crvenu boju.

Cijan oduzima crvenu boju od bijele svjetlosti.

Ovaj model primjenjuje se u uređajima za štampanje koji nanose pigmente boja na papir.

Veza između CMY modela i RGB modela matematički se definiše na sljedeći način:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Uvođenjem crne komponente K kao najmanjeg zajedničkog udjela cijana, magente i žute te izdvajanjem tog dijela iz C , M i Y komponenata na sljedeći način:

$$K = \min(C, M, Y)$$

$$C = C - K$$

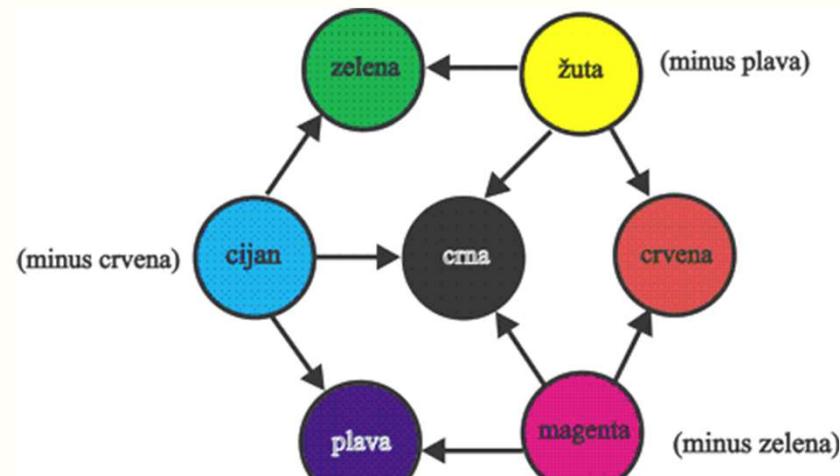
$$M = M - K$$

$$Y = Y - K,$$

definiše se model boje CMYK (cyan, magenta, yellow, black). Ovaj model se koristi u četverobojnoj štampi.

Subtraktivne primarne veličine i njihove kombinacije prikazane su na slici desno.

Npr. cijan i žuta daju zelenu.



CMYK model boje

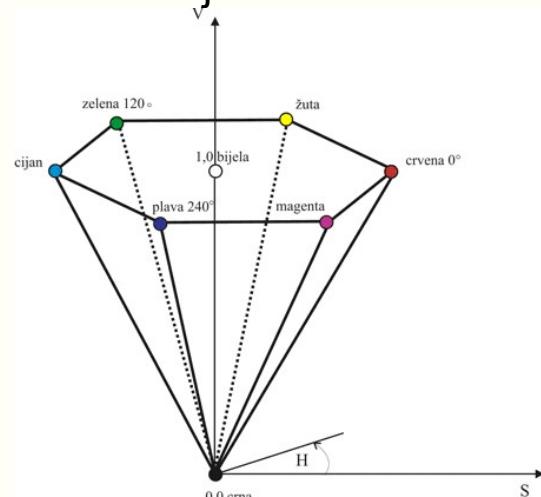
HSV MODEL BOJE

HSV (**hue**, **saturation**, **value**) model boje bliži je intuitivnom shvatanju boje i ovaj model se definiše u cilindričnom koordinatnom sistemu.

Boje su prikazane u podskupu prostora omeđenom šesterostranom piramidom a vrh piramide je u ishodištu i odgovara crnoj boji.

Vertikalna os V određuje sjajnost boje (**value**). Baza piramide je na nivou $V=1$ što odgovara skupu sjajnih boja.

Nijansa (**hue**) određena je uglom zakretanja H oko vertikalne osi V , a radijalna udaljenost od osi V određuje nivo zasićenosti boje (**saturation**).

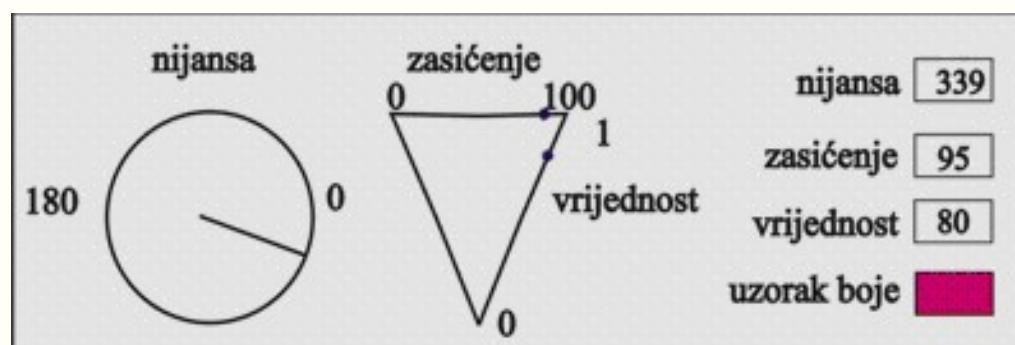


HSV model boje.

Primjer korisničkog sučelja za specificiranje boje u HSV prostoru

Mnogi aplikacijski programi omogućavaju korisniku specifikaciju boje različitih primarnih oblika uključujući linije, mnogouglove, tekst i druge, ali je problem u tome što je, zbog ograničenosti dimenzija zaslona, na takav način realno moguće prikazati relativno malen broj uzoraka boja.

Najprikladnije rješenje ovog problema je primjena sučelja koje omogućava interaktivnu promjenu uzorka boje pomicanjem odgovarajućih pokazivača ili numeričkom specifikacijom iznosa komponenata.



PRIMJENA BOJE U GRAFICI

Boja se koristi za različite namjene, npr. iz estetskih razloga, zbog postizanja utiska realističnosti ili za naglašavanje određenog dijela prikaza i sl..

Najčešće primjenjivana pravila koja je poželjno koristiti pri izboru boja :

- Prikladan način izbora boja je izbor tačaka s glatke putanje u podprostoru modela boje.
- Također dobar se učinak postiže ograničenjem na jednu plohu piramide u *HSV* modelu.
- Ako se na slici koristi samo nekoliko boja onda je prikladno komplementarnu boju jedne od njih odabrati kao boju pozadine.
- Neutralnu sivu boju prikladno je upotrijebiti kao boju pozadine u slučaju upotrebe većeg broja boja.
- Ako se koriste dvije boje koje nisu u skladnom odnosu prikladno je upotrijebiti tanki crni rub za razdvajanje.
- Plava i crna imaju sličan nivo sjajnosti pa se zbog toga slabo razlikuju. Sličan je odnos i bijele i žute.
- Žuto podcrtavanje crnog teksta na bijeloj pozadini vrlo je izražajno.
- Bijeli tekst na plavoj pozadini također se dobro ističe.
- Boja površine utiče na percepciju veličine, npr. crveni kvadrat izgleda veći od zelenog kvadarata istih dimenzija.
- Boja objekta utiče i na percepciju udaljenosti, npr. crveni objekt izgleda bliže od plavog objekta na istoj slici.