

1. MODELIRANJE I REPREZENTACIJA OBJEKATA

Modeliranje - postupak izrade 3D objekata

- različiti postupci modeliranja objekata
- različiti zapisi podataka koji čine objekt (različiti postupci prikaza engl. rendering)

Postupci modeliranja objekata:

- pomoću programskih alata (CAD)
- na osnovi uzorkovanih podataka (medicinski podaci, strojarstvo, stereo slike)
- proceduralno modeliranje objekata (drveće, planine, oblaci, vatra)
- fizikalno temeljeno modeliranje (tkanina, kosa, tekućine, vatra)

Gotovi programski paketi:

- za crtanje - CAD, animacije
- za obradu i prikaz podataka, Matlab
- postupci uzorkovanja objekata, pripadni programski paketi

1.1. MODELIRANJE OBJEKATA I SCENE

Modelirani objekti:

- modeliranje **površine**

- definirana je vanjska ljuska objekta (“koža”)

- *poligonima (trokuti), prednja i stražnja strana,*

- *parametarska površina*

- *elementima površine* (engl. surfel) – PBG (point base graphics)

- modeliranje **volumena** tijela

- definirana je unutrašnjost objekta (*implicitno* definirano)

- *implicitnim funkcijama*

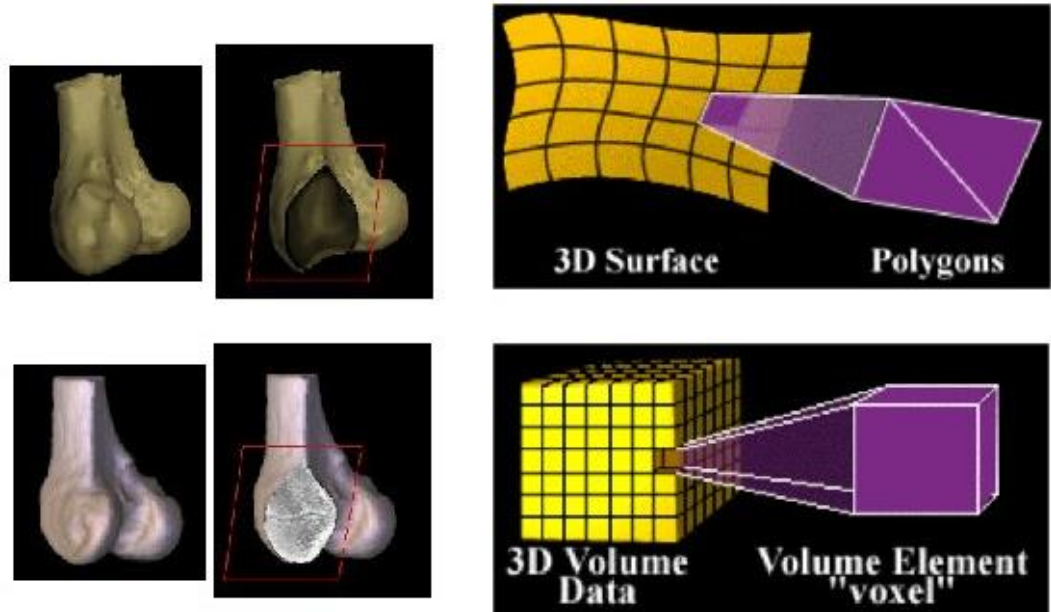
- *elementima volumena* (voxel)

možemo iz jednog oblika načiniti drugi (nije uvijek jednostavno)

Zapisivanje scene (strukture više razine):

- graf scene – podaci o izvorima, promatračima, animaciji,
- specifični podaci ovisni o aplikaciji – fizikalni elementi

površina i volumen



bitna razlika eksplicitnog i implicitnog oblika

– eksplicitni, parametarski

- određivanje točaka površine, tangenitnih ravnina

– implicitni

- jednostavno možemo odrediti

pripada li neka točka površini je li “iznad” ili “ispod”,
udaljenost od površine

booleove operacije nad tijelima, detekcija sudara

- površina objekta - poligonima
 - poligonalni model BREP (*boundary representation*)
 - žična forma objekta
 - geometrijski podaci (položaj vrhova)
 - topološki podaci (povezanost vrhova –poligoni)
 - poligon (trokut) definira jednadžbu ravnine
 - parametarski oblik jednadžbe ravnine
 - implicitni oblik jednadžbe ravnine



$$\mathbf{V} = [u \quad v \quad 1] \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix}$$

$$ax + by + cz + d = 0$$

- površina – parametarski zadana

modeliranje površine (engl. surface modelling)

- površina je **glatka** i kontinuirano se može kontrolirati (obrada plohe)
- definirana je površinska ljuska tijela (može biti zatvorena)
- slobodno oblikovane površine (engl. Freeform surfaces)
 - Bezierove površine, NURBS (B-površine),

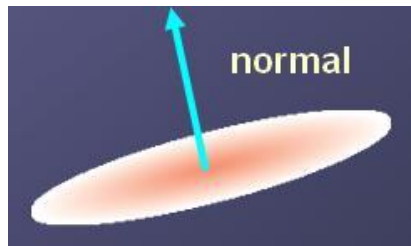


- modeliranje plohe <http://www.infogoaround.org/JBook/ShowRuleSurf.html>
<http://www.infogoaround.org/JBook/ShowSweptSurf.html>
- rotacione plohe <http://www.infogoaround.org/JBook/ShowRevSurf.html>

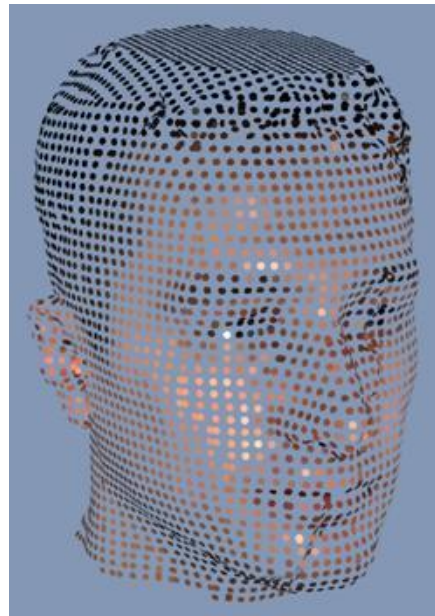
- površina objekta – točkama

modeliranje elementima površine PBG (engl. point base graphics)

- ‘+’ sklopovska podrška za brzu izradu prikaza
- ‘-’ pojava šupljina na rezultatu, alias
- surfeli se projiciraju na zaslon (splatting) s rekonstrukcijskom jezgrom ovisno o kutu između normale i promatrača
- <http://graphics.cs.cmu.edu/projects/objewa/>



surfel



273K surfel-a

- volumen tijela – implicitnim funkcijama

modeliranje volumena tijela (engl. volumetric modelling, solid modelling)

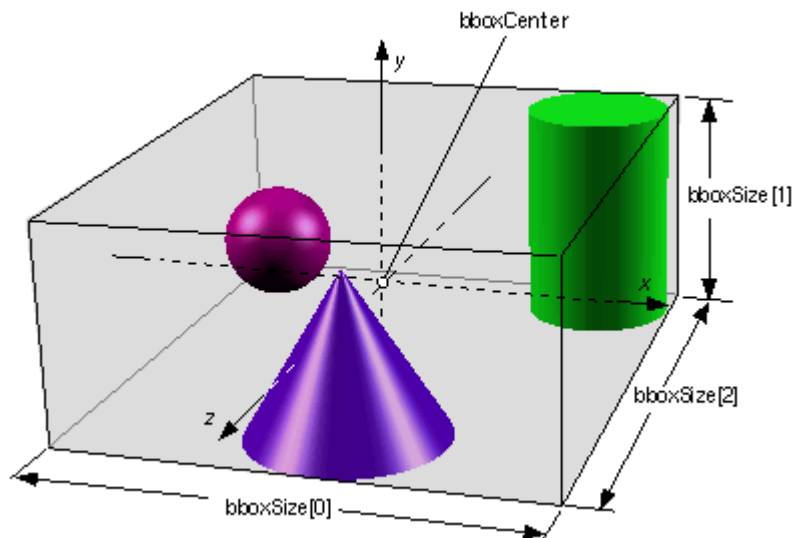
implicitno definirane površine

$$f(x, y, z) = \text{const}$$

definirana je unutrašnjost tijela npr:

unutar tijela $f(x, y, z) \leq \text{const}$,

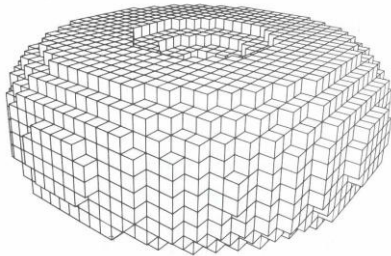
izvan tijela $f(x, y, z) > \text{const}$.



$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2$$

<http://www.infoaround.org/JBook/ShowTorus.html>

- volumen tijela – elementi volumena (engl. voxel)
 - po uzoru na slikovne elemente elementi volumena (vox-el) svakoj točki prostora (x, y, z) imaju pridruženu neku vrijednost



- podaci su obično ostvareni postupkom uzorkovanja (CT, MR) u unutrašnjosti je objekt slojevito predstavljen (kao luk)

izo - površine :

$$f(x, y, z) \leq con1,$$

$$f(x, y, z) \leq con2,$$

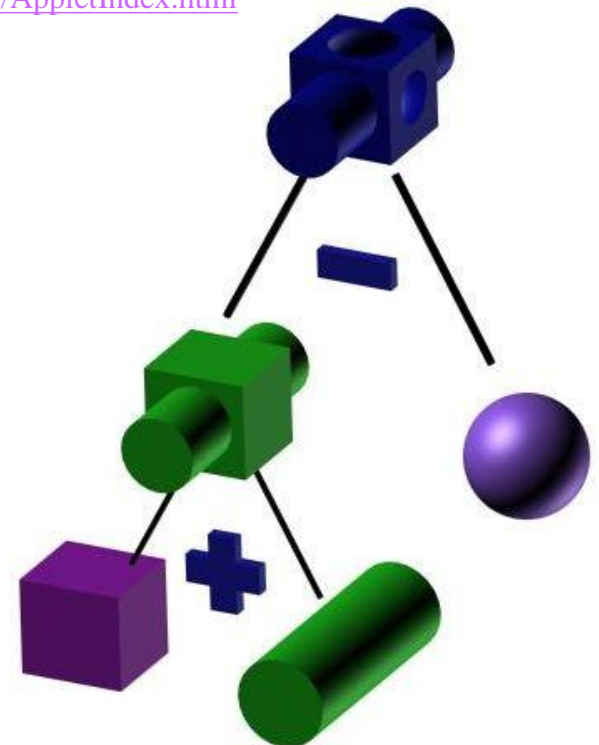
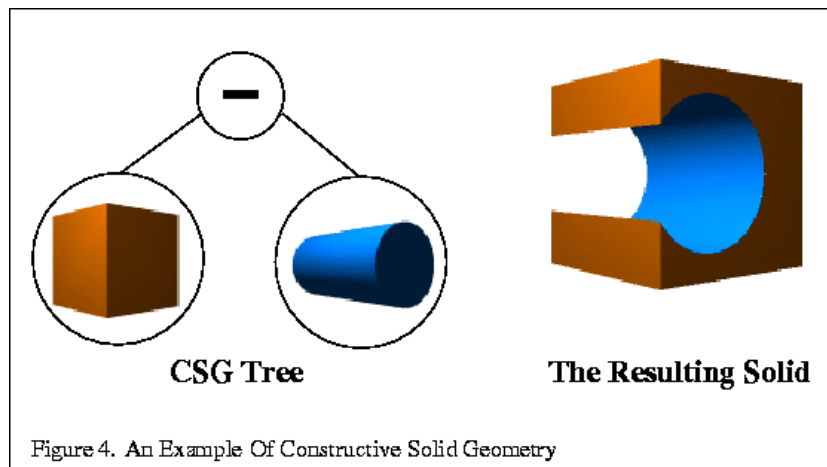
$$f(x, y, z) \leq con3, \dots$$

Konstruktivna geometrija tijela

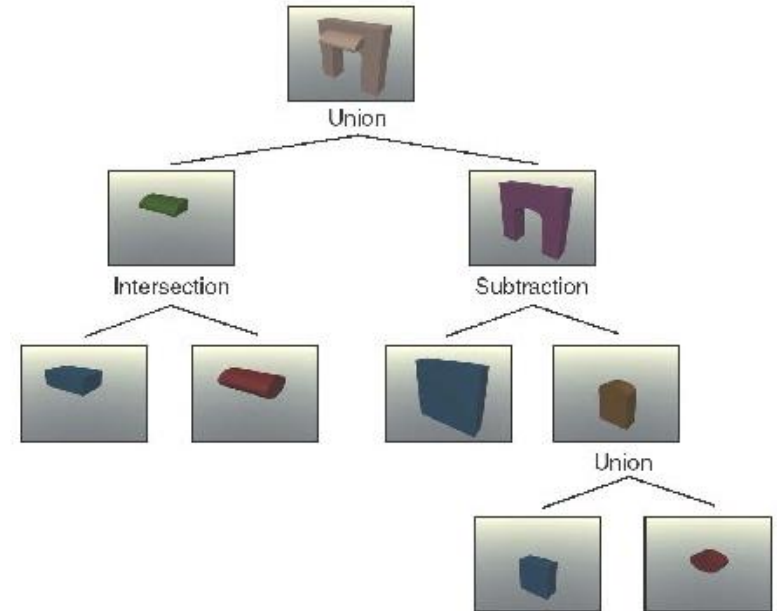
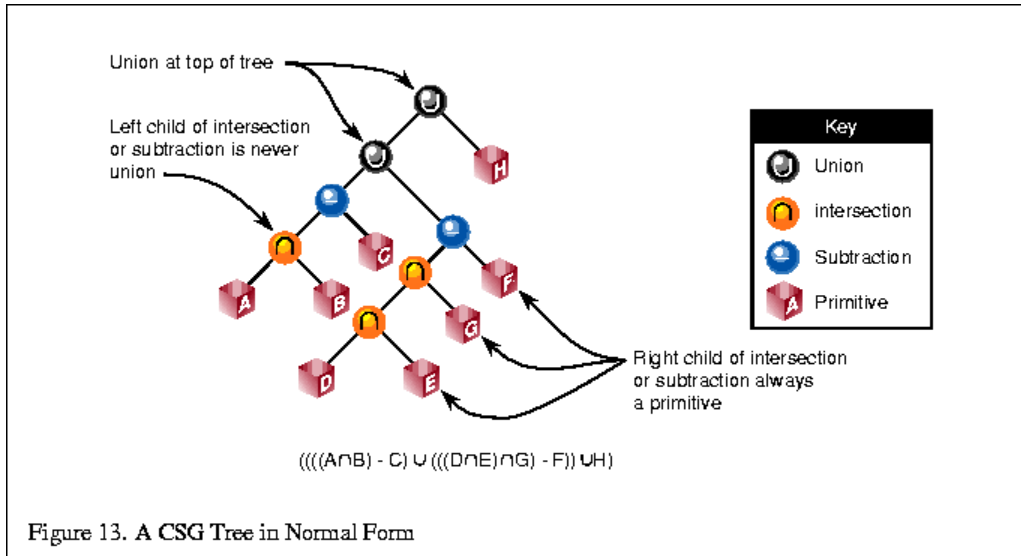
CSG (engl. Constructive solid geometry)

- geometrijska tijela (kugla, kocka, valjak, stožac ...)
- + Booleove operacije (unija, presjek, razlika)
- obično se koristi u CAD

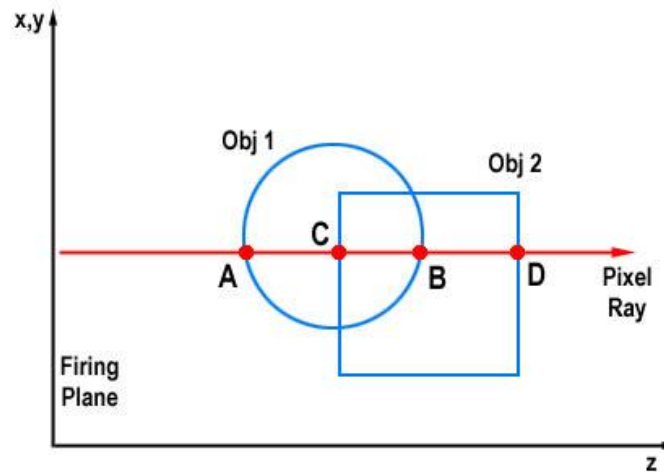
<http://www.cs.technion.ac.il/~cs234325/Applets/doc/html/etc/AppletIndex.html>



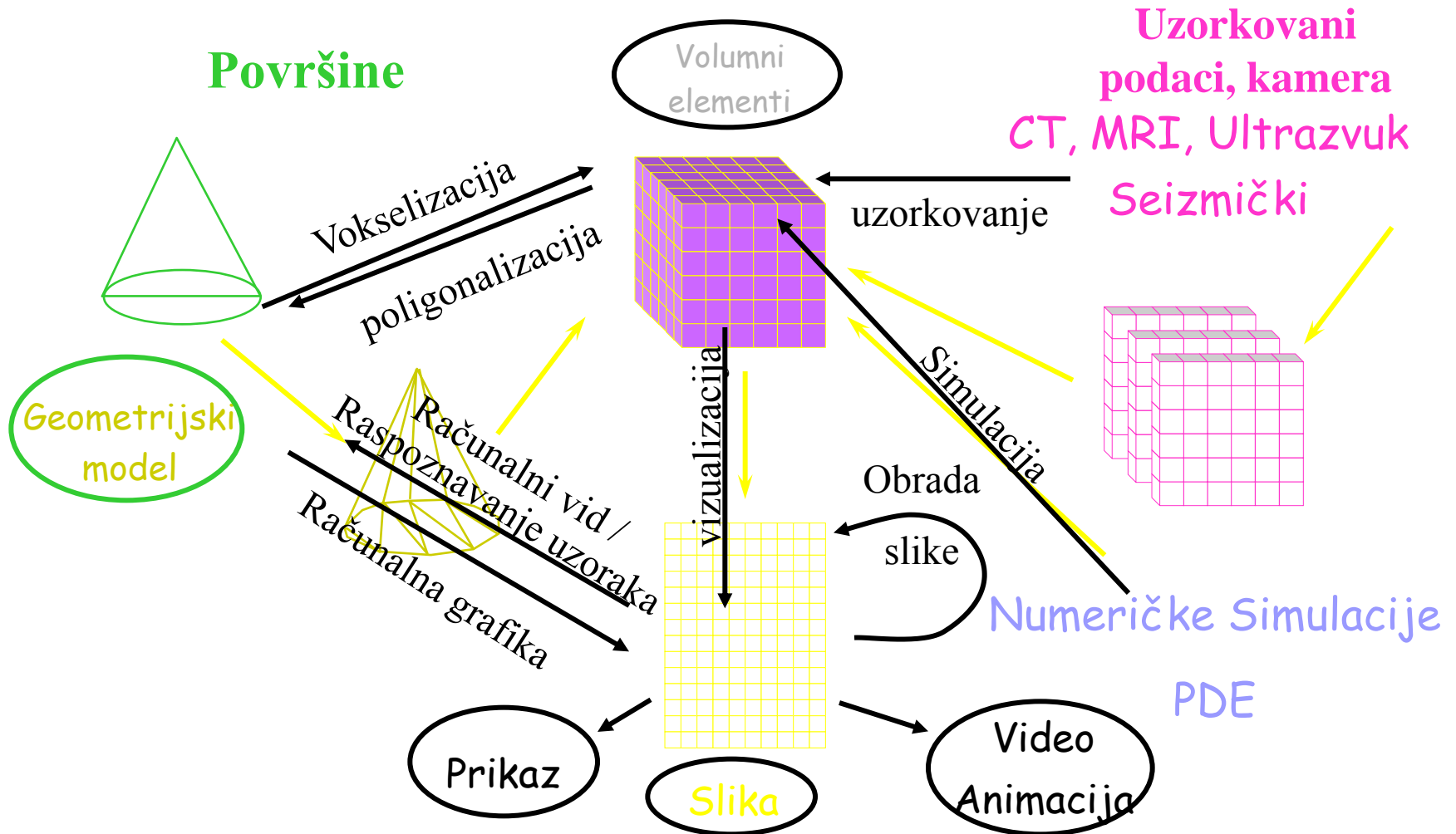
- CSG stablo



Booleove operacije



Objekti



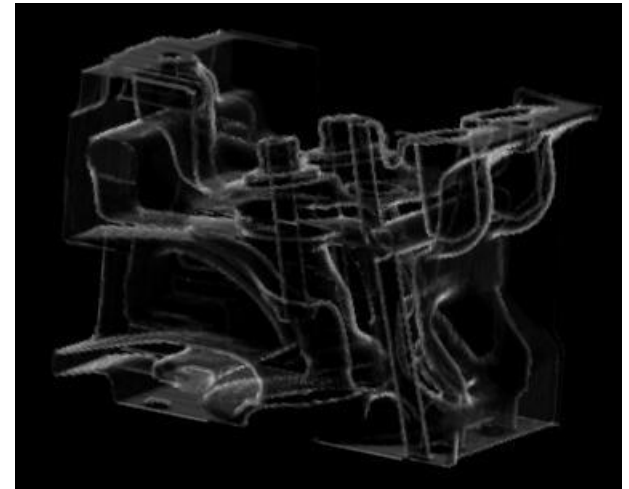
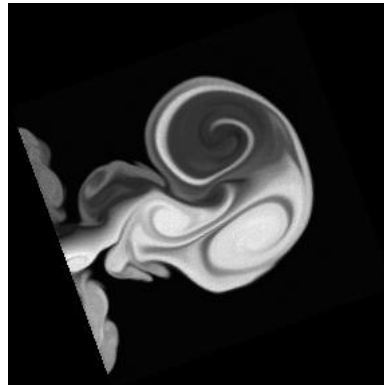
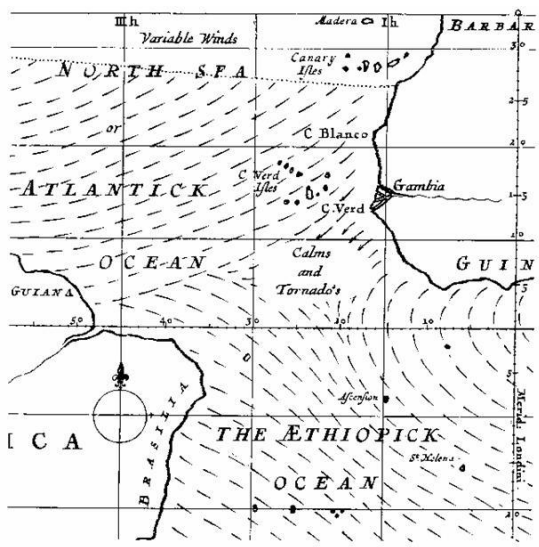
Ostvarivanje prikaza (rendering)

prikaz poligona – klasičan način – fotorealističan prikaz

NPR - ne fotorealističan prikaz (engl. Non-Photorealistic Rendering)

- ne želimo biti ograničeni samo na foto realističan prikaz
- skica objekta
- tehnika prikaza primjenjiva na objekte definirane volumno i površinom

<http://bandviz.cg.tuwien.ac.at/basinviz/compression/paperindex.html>

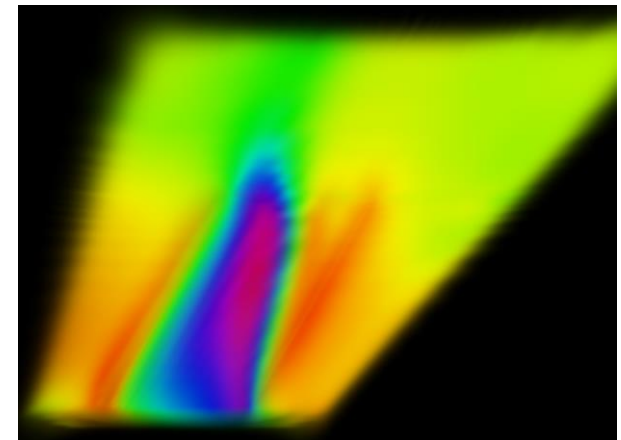
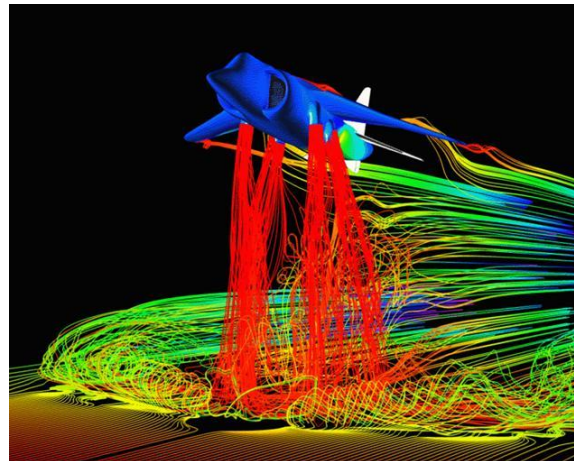
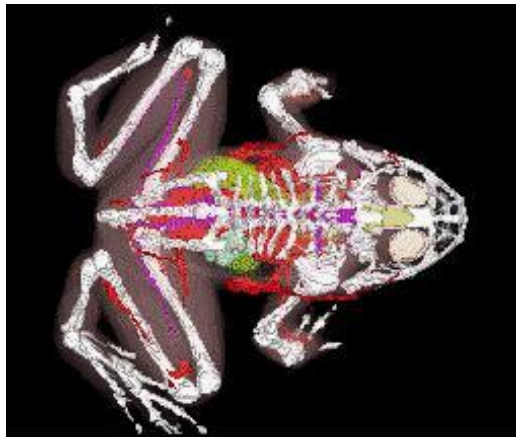
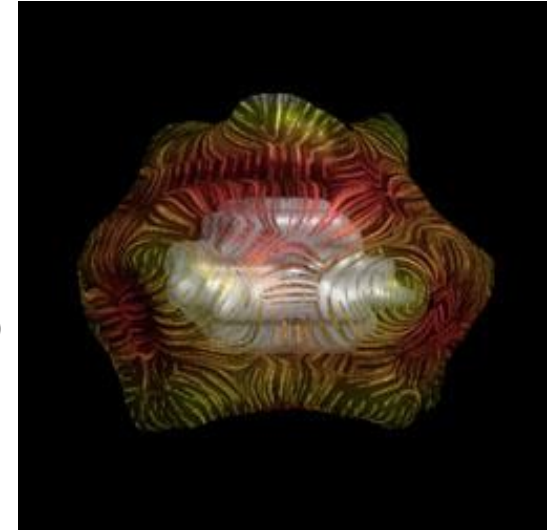


prijenosna funkcija (engl. transfer function)

- određuje što će biti i na koji način preslikano u optičke parametre to može biti boja objekta izravno, no može biti i neka druga informacija kod NPR tehnike to je mjesto gdje je normala na površinu okomita prema vektoru prema promatraču

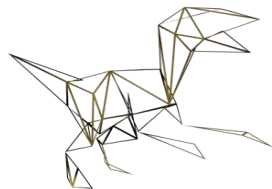
upotreba boje za prikaz dodatne informacije

- razlikovanje dijelova objekta – odjeljivanje cjelina
- razna svojstva objekta u pojedinoj točki (temperatura, brzina)



Složeni zapis objekta

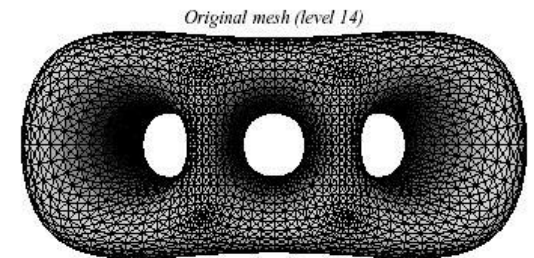
- postupci vizualizacije zahtijevaju interaktivni rad, pa je važna količina i organizacija podataka
- isti objekt nam je često potreban u različitim razinama složenosti (LOD)
 - prikaz objekta ovisno o udaljenosti i veličini prikaza
 - proračun sudara (kolizije)
- ugrublјivanje
 - krećemo od najsitnije podjele



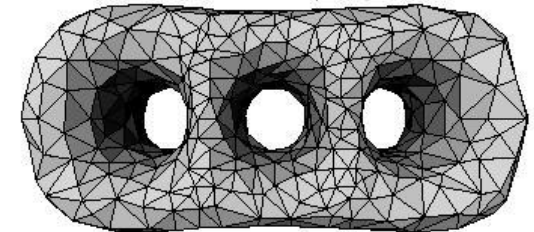
broj vrhova 50

500

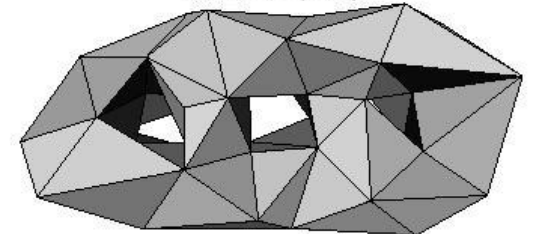
2 000



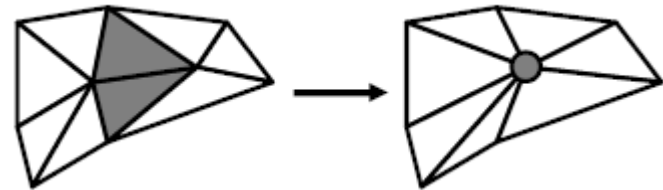
Intermediate mesh (level 6)



Coarsest mesh (level 0)

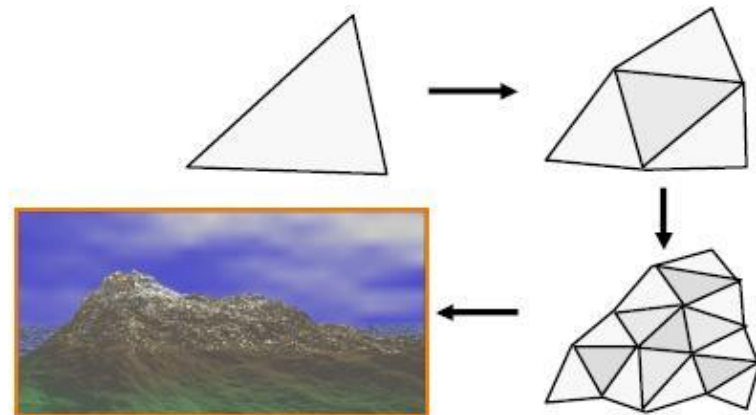


- ugrubljivanje poligonalne mreže
 - spajamo poligone u veće tako da važna obilježja objekta budu sačuvana
 - stapamo vrhove



- usitnjavanje poligonalne mreže (engl. Subdivision)
 - dijelimo poligone najgrublje razine i novonastale vrhove pomičemo (tako da novi objekt bude gladak ili hrapav)

<http://www.gvu.gatech.edu/~jarek/demos/polyEditor/>

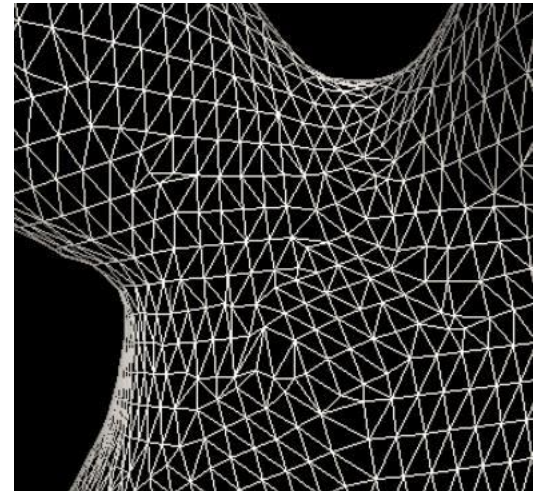


fraktalna podjela pri izradi planine

1.2 REPREZENTACIJA OBJEKATA

površina objekta – zapis poligonima odnosno trokutima

- objekte najčešće predstavljamo mrežom poligona (samo površina objekta)
- trokuti – planarni su,
 - sklopovlje GPU podržava trokute
- kako načiniti strukture podataka
- osnovni elementi (poligonalna mreža)
 - **geometrijski podaci** – vrh (točka u prostoru)
 - **atributi** – boja, normala, koordinate teksture
 - **topološki podaci** – brid (povezuje 2 vrha)
 - poligon (povezuje više vrhova)
- objekti
 - geometrijski podaci, atributi, topološki podaci
 - LOD – jedan objekt može imati više poligonalnih mreža različite složenosti – ovisno o udaljenosti prikazuju se različite mreže



Primitive u OpenGL-u:

– točke

- `GL_POINTS`



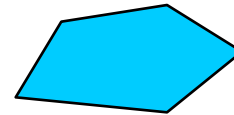
– dužina, niz dužina,

- `GL_LINES`,
- `GL_LINE_STRIP`,
- `GL_LINE_LOOP`



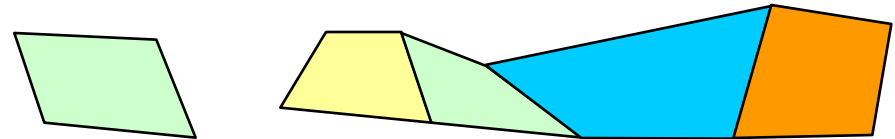
– poligon

- `GL_POLYGON`,



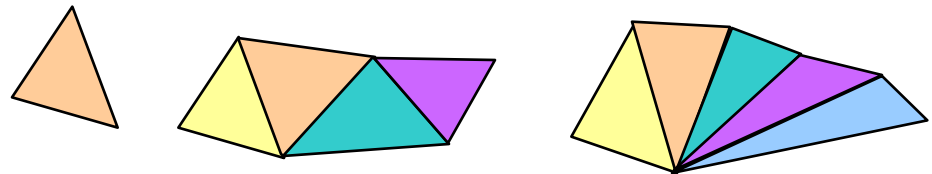
– četverokut, niz četverokuta

- `GL_QUADS`,
- `GL_QUAD_STRIP`,



– trokut, niz trokuta

- `GL_TRIANGLES`
- `GL_TRIANGLE_STRIP`
- `GL_TRIANGLE_FAN`



VTK (Visualization ToolKit)

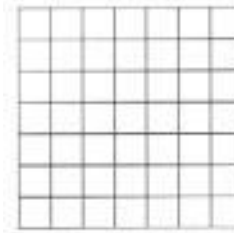
- alat za vizualizaciju, otvorenog koda (open source)
- procesiranje podataka i prikaz, (procesiranje slike), FFT, filtriranje, podrška za izradu sučelja
- mogućnost programiranja
 - C++ (biblioteka klasa)
 - Java
 - Python
 - interpreter Tcl/Tk
- “iznad” OpenGL-a, podržava skalare, vektore, tenzore, teksture, volumetrijske metode, implicitno modeliranje
- operacije nad podacima – odsijecanje, glađenje i redukciju poligonalne mreže, prikaz kontura, Delaunay trijangulaciju
- importiranje/eksportiranje različitih zapisa .obj, 3D studio, TIFF, BMP

Zapis objekata u VTK:

Zaglavlje (Header):

```
# vtk DataFile Version 2.0
Really cool data
ASCII | BINARY
DATASET type
...
POINT_DATA n
...
CELL_DATA n
...
```

(1)



(a) Structured Points

(4) 1D, 2D, 3D
DATASET STRUCTURED_POINTS
DIMENSIONS n_x n_y n_z
ORIGIN xyz
SPACING s_x s_y s_z

(1) prva linija je točno kako je navedena, uz verziju VTK

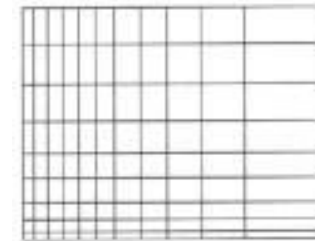
(2) 256 znakova proizvoljnog opisa

(3) podaci su ASCII ili binarni

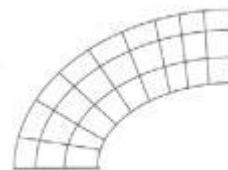
(4) ključna riječ DATASET i tip podataka

- STRUCTURED_POINTS
- RECTILINEAR_GRID
- STRUCTURED_GRID
- POLYDATA
- UNSTRUCTURED_GRID
- FIELD

(5) atributi – vektori normala, boja



(b) Rectilinear Grid



(c) Structured Grid

(4)
DATASET STRUCTURED_GRID
DIMENSIONS n_x n_y n_z
POINTS n $dataType$
 P_{0x} P_{0y} P_{0z}
 ...
 $P_{(n-1)x}$ $P_{(n-1)y}$ $P_{(n-1)z}$

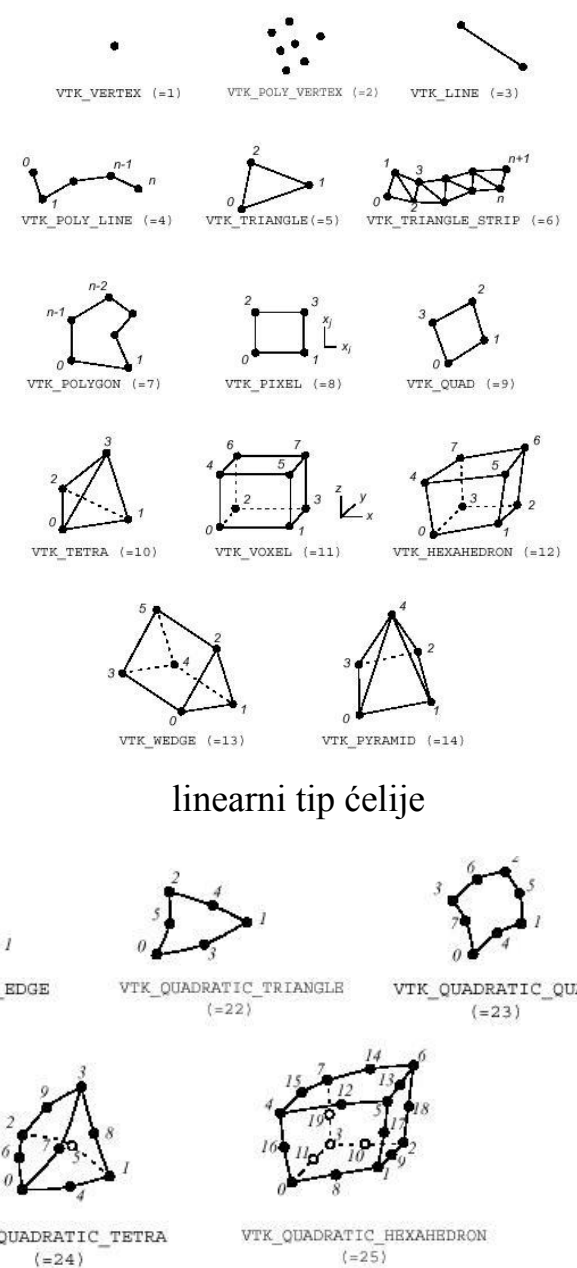
Zapis objekata u VTK:

(4) ključna riječ DATASET i tip podataka

- ...
- POLYDATA
 - POINTS (točke x,y,z)
 - VERTICES (skupovi točaka nD)
 - LINES
 - POLYGONS
 - TRIANGLE_STRIPS

(4)
 DATASET POLYDATA
 TRIANGLE_STRIPS n size
 numPoints₀, i₀, j₀, k₀,
 ...
 numPoints_{n-1}, i_{n-1}, j_{n-1}, k_{n-1}

- UNSTRUCTURED_GRID
 - POINTS
 - CELLS
 - CELL_TYPES (svaki tip ima svoj broj)



linearni tip ćelije

Figure 3 Non-linear cell types found in VTK.

Zapis objekata u VTK:

(5) atributi

- SCALARS, COLOR_SCALARS
- LOOKUP_TABLE [indeksirani pristup](#)
- VECTORS (npr. brzina)
- NORMALS
- [TEXTURE COORDINATES](#)
- TENSORS do 3×3
- FIELD

```
(5)
DATASET POLYDATA
TRIANGLE_STRIP n size
numPoints0, i0, j0, k0,
...
numPointsn-1, in-1, jn-1, kn-1
```

podržana je i XML sintaksa zapisa npr:

```
<VTKFile type="RectilinearGrid" ...>
  <RectilinearGrid WholeExtent="x1 x2 y1 y2 z1 z2">
    <Piece Extent="x1 x2 y1 y2 z1 z2">
      <PointData>...</PointData>
      <CellData>...</CellData>
      <Coordinates>...</Coordinates>
    </Piece>
  </RectilinearGrid>
</VTKFile>
```

- želimo jedan funkcijski poziv i odjednom poslati veću količinu podataka

korištenja priručne memorije (engl. cache) vrhova

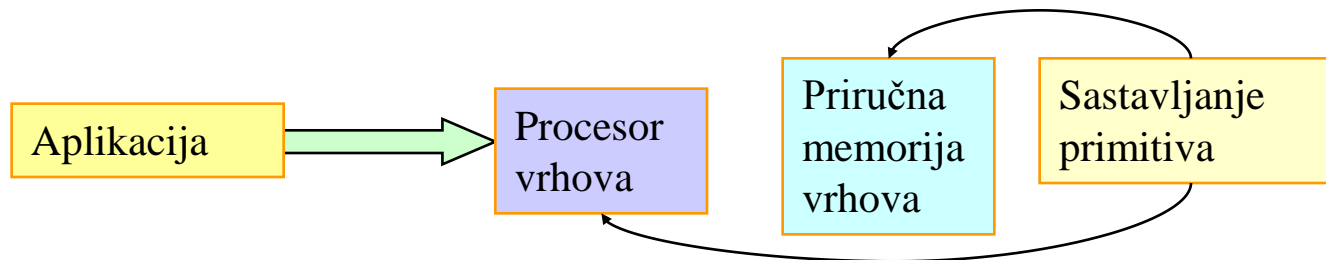
- procesiranje vrhova je sekvencijsko, nemamo zajedničke (dijeljene) vrhove

```
glDrawArrays(GL_TRIANGLES, 0, num_vertices);
```

- slučajni pristup vrhovima – omogućeno je dijeljenje vrhova

```
glDrawElements(GL_TRIANGLES, indices.size(), GL_UNSIGNED_INT, indices[0]);
```

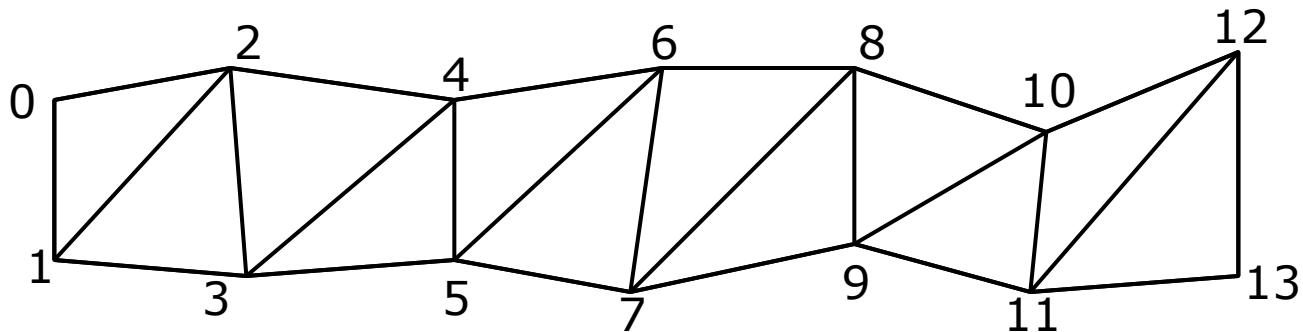
- u priručnoj memoriji su transformirani vrhovi
- neki vrhovi se višestruko ponavljaju, tj. za svaki trokut su vrhovi zasebno navedeni



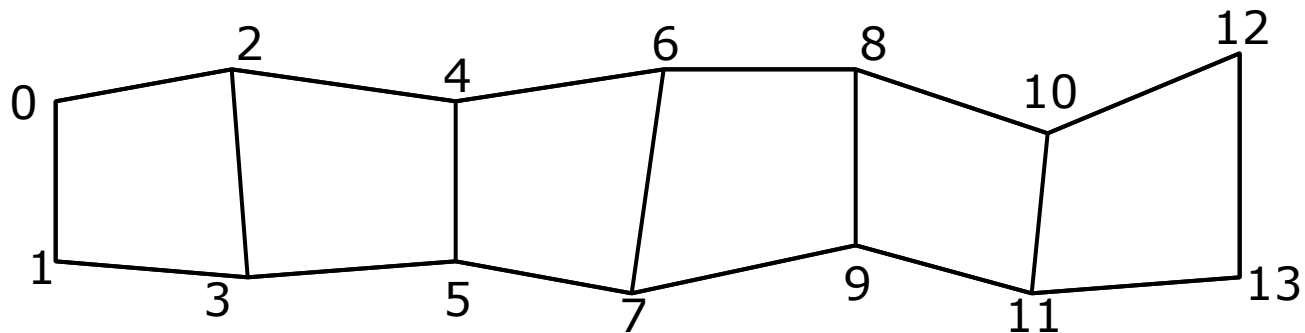
Niz trokuta (triangle strip):

- neki vrhovi zajednički trokutima (značajna ušteda)
- za n trokuta imamo $n + 2$ vrha umjesto $3 \times n$

```
glDrawElements(GL_TRIANGLE_STRIP, indices.size(), GL_UNSIGNED_SHORT, &indices[0]);
```



GL_QUAD_STRIP za n četverokuta imamo $2n + 2$ vrha



// KOORDINATE

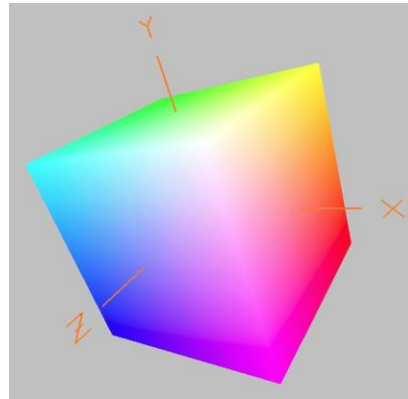
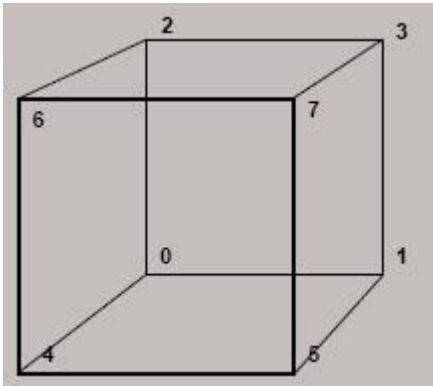
```
static GLfloat CubeVertices[ ][3] =  
{  
    { -1., -1., -1. },  
    { 1., -1., -1. },  
    { -1., 1., -1. },  
    { 1., 1., -1. },  
    { -1., -1., 1. },  
    { 1., -1., 1. },  
    { -1., 1., 1. },  
    { 1., 1., 1. }  
};
```

// ATRIBUTI - BOJA

```
static GLfloat CubeColors[ ][3] =  
{  
    { 0., 0., 0. },  
    { 1., 0., 0. },  
    { 0., 1., 0. },  
    { 1., 1., 0. },  
    { 0., 0., 1. },  
    { 1., 0., 1. },  
    { 0., 1., 1. },  
    { 1., 1., 1. }  
};
```

// INDEKSI VRHOVA

```
static GLuint CubeIndices[ ][4] =  
{  
    { 0, 2, 3, 1 },  
    { 4, 5, 7, 6 },  
    { 1, 3, 7, 5 },  
    { 0, 4, 6, 2 },  
    { 2, 6, 7, 3 },  
    { 0, 1, 5, 4 }  
};
```



```

// pozivi glVertexElement( )
glEnableClientState( GL_VERTEX_ARRAY );
glEnableClientState( GL_COLOR_ARRAY );
glVertexPointer( 3, GL_FLOAT, 0, CubeVertices );
glColorPointer( 3, GL_FLOAT, 0, CubeColors );
glBegin( GL_QUADS );
    glVertexElement( 0 );
    glVertexElement( 2 );
    glVertexElement( 3 );
    glVertexElement( 1 );
    glVertexElement( 4 );
    glVertexElement( 5 );
    glVertexElement( 7 );
    glVertexElement( 6 );
    glVertexElement( 1 );
    glVertexElement( 3 );
    glVertexElement( 7 );
    glVertexElement( 5 );
    glVertexElement( 0 );
    glVertexElement( 4 );
    glVertexElement( 6 );
    glVertexElement( 2 );
    glVertexElement( 2 );
    glVertexElement( 6 );
    glVertexElement( 7 );
    glVertexElement( 3 );
    glVertexElement( 0 );
    glVertexElement( 1 );
    glVertexElement( 5 );
    glVertexElement( 4 );
glEnd();

```

```

// poziv glDrawElements( )
glEnableClientState( GL_VERTEX_ARRAY );
glEnableClientState( GL_COLOR_ARRAY );
glVertexPointer( 3, GL_FLOAT, 0, CubeVertices );
glColorPointer( 3, GL_FLOAT, 0, CubeColors );
glDrawElements( GL_QUADS, 24, GL_UNSIGNED_INT,
                CubeIndices );

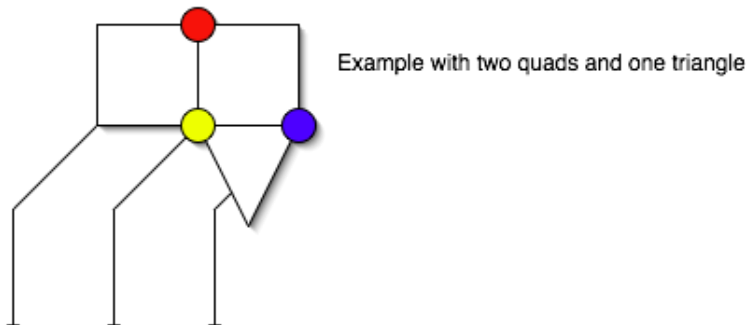
```

- Primjeri organizacije podataka i atributa:

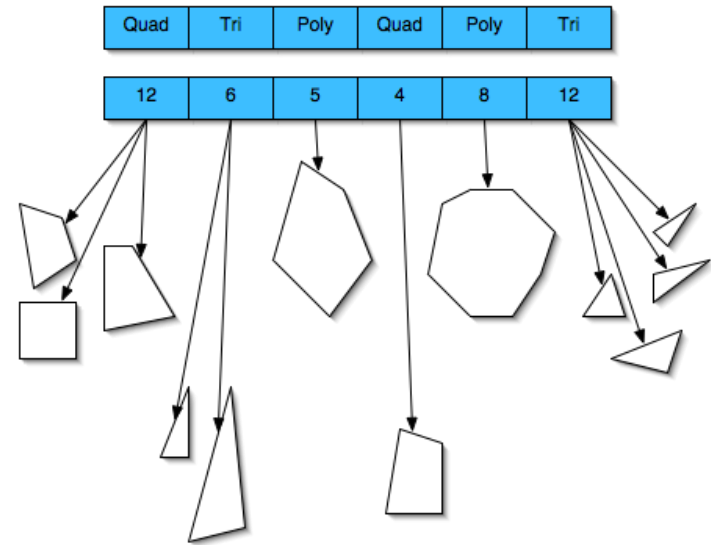
- <http://www.gris.uni-tuebingen.de/grisalt/projects/grdev/doc/html/Overview.html>

- bez indeksiranja

- višestruki podaci
 - npr. za eksploziju poligona

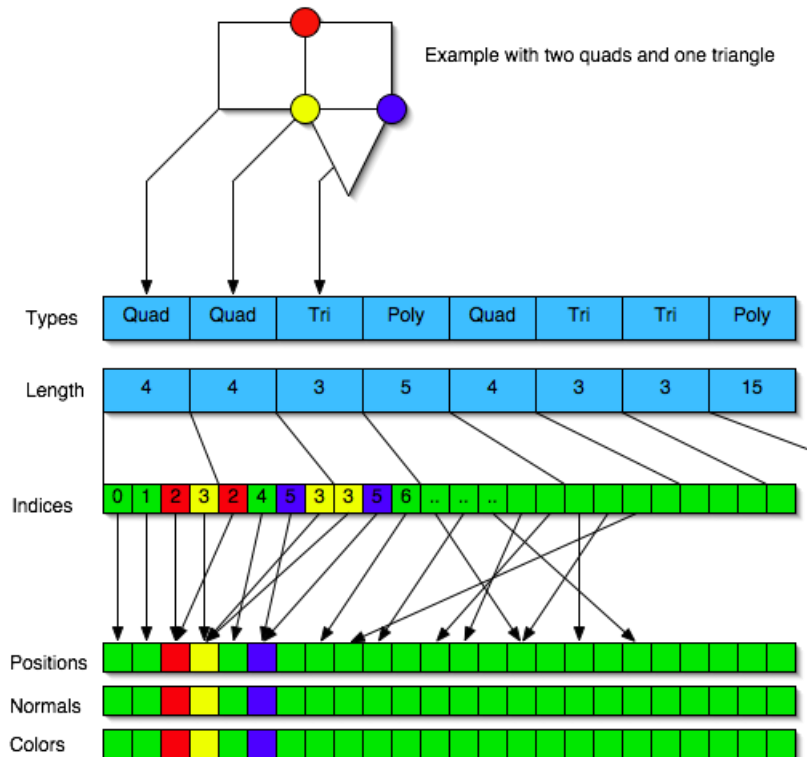


Types	Quad	Quad	Tri	Poly	Quad	Tri	Tri	Poly														
Length	4	4	3	5	4	3	3	15														
Positions	[Red, Yellow, Red, Green]				[Green, Yellow, Blue]			[Green]														
Normals	[Red, Yellow, Red, Green]				[Green, Yellow, Blue]			[Green]														
Colors	[Red, Yellow, Red, Green]				[Green, Yellow, Blue]			[Green]														

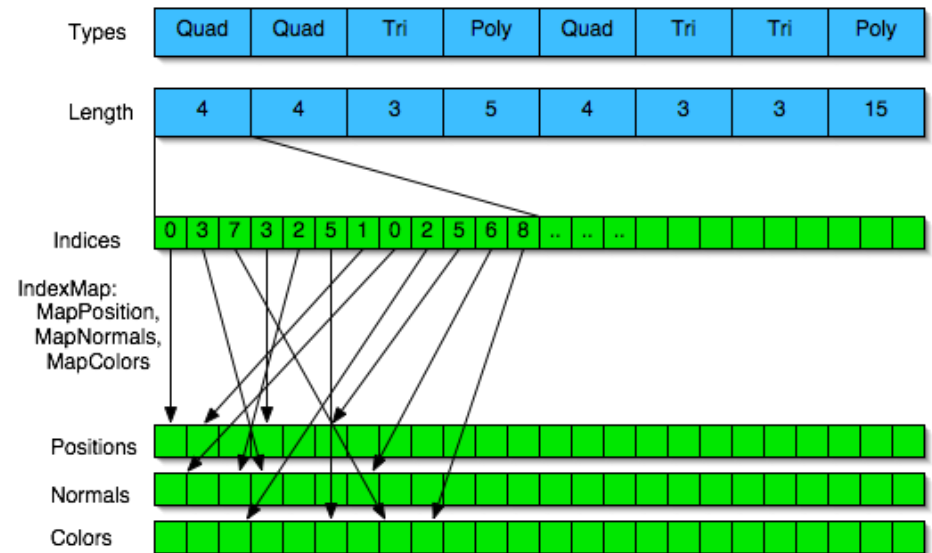


– indeksirani pristup

- zajednički podaci za pojedini vrh
- npr. sjenčanje Gouraud



- višestruko indeksirani pristup
 - poseban pristup normalama, boji i sl.
 - povećan broj indeksa ($\times 3$)
 - npr. normale poligona – konstantno



5.3 STRUKTURE PODATAKA ZA ZAPIS POLIGONALNIH OBJEKATA

Objekti zadani poligonima

- ovisno o tome za što je potrebno načiniti s objektima potrebno je formirati strukture podataka
 - samo prikaz i osnovne transformacije
 - modificiranje objekta (npr. izobličavanje, promjena broja poligona stapanjem vrhova)
 - ispitivanje kolizije (sudara) objekata
 - eksplozija objekta

Zapis površine objekta B-rep BREP (engl. boundary representation)

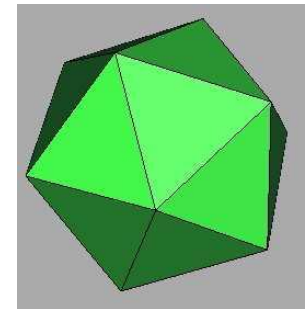
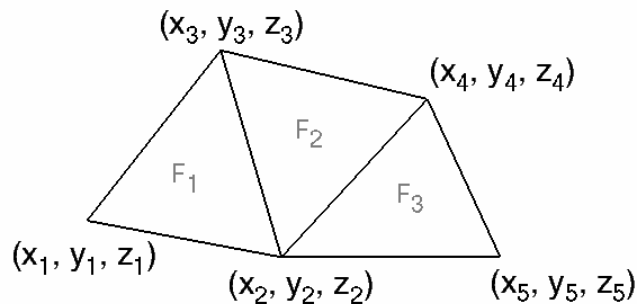
Strukture podataka

1. Tablica poligona
2. Tablice vrhova i poligona
3. Tablica bridova, vrhova i poligona
4. Liste susjednosti
5. Krilati brid

1. Tablica poligona

- nužno ako objekt “eksplođira” ili se poligoni rasprše, tada koordinate vrhova moraju biti posebno definirane iako su početno na istom mjestu
- nije efikasno ako objekt čini cjelinu, repliciramo podatke
- ako su vrhovi višestruko definirani može doći do pojave pukotina na spojevima poligona
- općeniti poligoni, nisu nužno trokuti
- redosljed vrhova je u primjeru suprotno smjeru kazaljke na satu gledano izvan tijela CCW (određuje redosljed bridova, određuje normalu po pravilu desne ruke)
- <http://www.gris.uni-tuebingen.de/edu/projects/grdev/doc/html/Overview.html>

Tablica poligona			
F ₁	(x ₁ , y ₁ , z ₁)	(x ₂ , y ₂ , z ₂)	(x ₃ , y ₃ , z ₃)
F ₂	(x ₂ , y ₂ , z ₂)	(x ₄ , y ₄ , z ₄)	(x ₃ , y ₃ , z ₃)
F ₃	(x ₂ , y ₂ , z ₂)	(x ₅ , y ₅ , z ₅)	(x ₄ , y ₄ , z ₄)

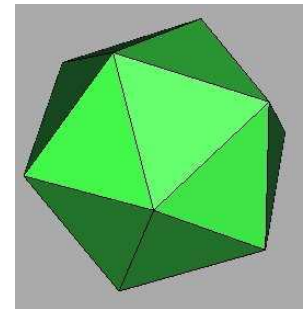
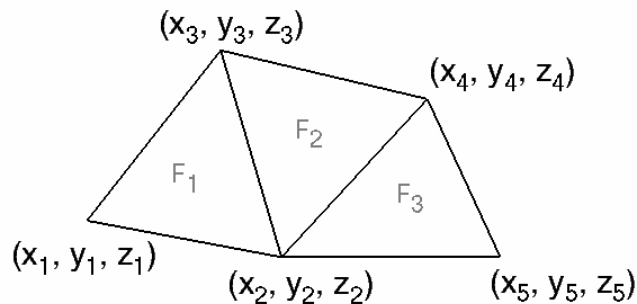


2. Tablice vrhova i poligona

- vrhovi su dijeljeni, zajednički za različite poligone, nisu replicirani,
- u tablici poligona su indeksi
- pomicanje jednog vrha izobličiti će sve poligone koji ga dijele
- razdvojena je informacija o geometriji (vrhovi) i topologiji (povezanosti - poligoni)
- nemamo informaciju o susjednosti
 - ako nas zanima za vrh V_2 koji poligoni sadrže taj vrh morati ćemo pretražiti sve poligone, ili koji poligoni čine brid B_{24}
- pogodno je što su istovrsni podaci (vrhovi) zajedno
- redoslijed vrhova u tablici poligona može biti upotrijebljen za određivanje normale

Tablica vrhova			
V_1	X_1	Y_1	Z_1
V_2	X_2	Y_2	Z_2
V_3	X_3	Y_3	Z_3
V_4	X_4	Y_4	Z_4
V_5	X_5	Y_5	Z_5

Tablica poligona			
F_1	V_1	V_2	V_3
F_2	V_2	V_4	V_3
F_3	V_2	V_5	V_4



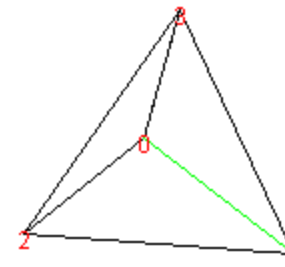
3. Tablice bridova, vrhova i poligona

- tablica poligona sadrži pokazivače na bridove,
- tablica bridova sadrži pokazivače na vrhove
- redoslijed bridova određuje orijentaciju poligona, redoslijed vrhova određuje orijentaciju bridova
- <http://www.gris.uni-tuebingen.de/edu/projects/grdev/doc/html/Overview.html>

npr: brid2 ide od V1 do V0

ako nam trebaju poligoni koji čine taj brid moramo pretražiti indekse vrhova u listi poligona

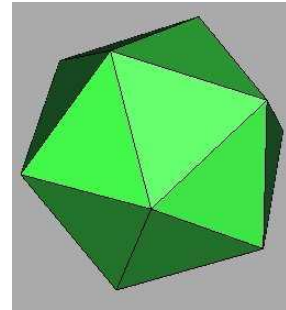
KNOT-LIST				EDGE-LIST			FACE-LIST			
	X	Y	Z		P0	P1		K0	K1	K2
0	0.453608	0.89043	0.037037	0	0.0	2.0	0	0.0	1.0	2.0
1	0.544331	-0.62854	-0.555555	1	2.0	1.0	1	2.0	3.0	4.0
2	-0.090722	-0.366647	0.925925	<input checked="" type="checkbox"/>	1.0	0.0	2	4.0	5.0	0.0
3	-0.907218	0.104757	-0.407407	3	1.0	3.0	3	1.0	5.0	3.0
---				4	3.0	0.0	---			
---				5	3.0	2.0	---			



4. Liste susjednosti

- tablica bridova
 - koji vrhovi čine brid (orijentacija brida V_2V_3 , V_3V_2)
 - koji bridovi su susjedni (diraju prvi ili drugi vrh)
 - koji poligoni čine brid

- tablica vrhova
 - susjedni vrhovi
 - incidentni bridovi
 - incidentni poligoni

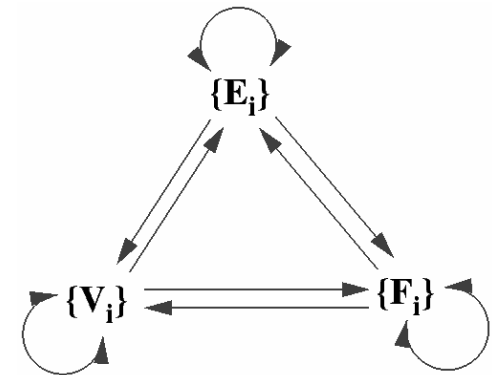
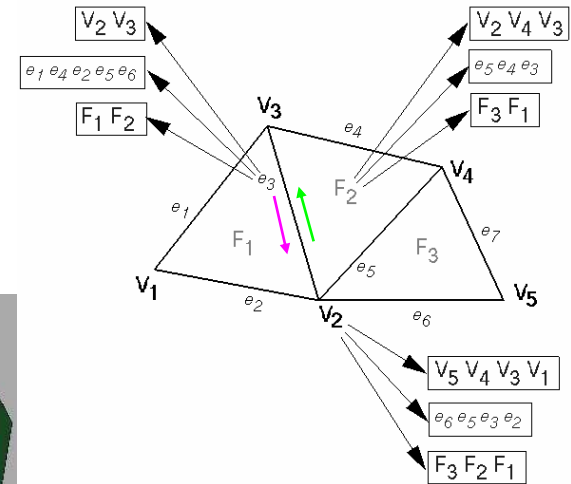


- tablica poligona
 - vrhovi
 - bridovi koji ga čine
 - susjedni poligoni

želimo imati informaciju o susljednosti ali želimo pohranjivati manje podataka

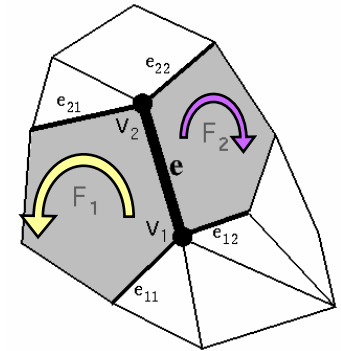
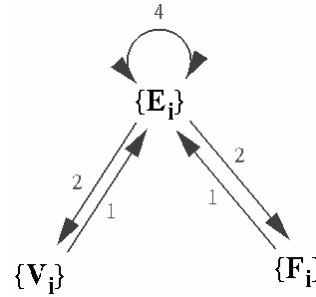
kompromis: potrebna memorija \leftrightarrow vrijeme potrebno za određivanje susjednosti

- 9 relacija susjednosti



5. Krilati brid (engl. winged edge)

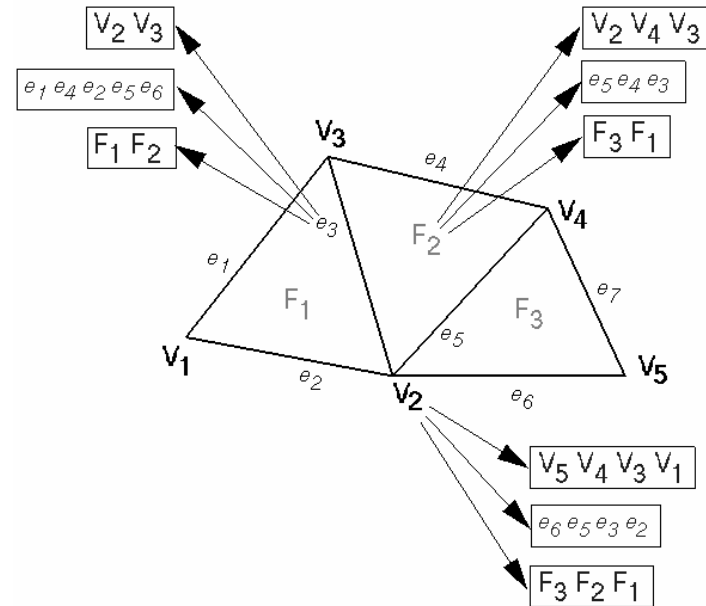
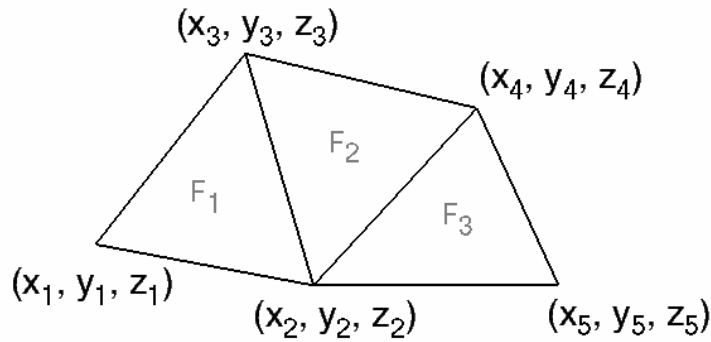
- tablica bridova
 - koji vrhovi čine brid (početni, završni) V_1 V_2
 - koji poligoni čine brid (lijevi, desni) F_1 F_2
 - bridovi lijevog poligona (brid koji prethodi, brid koji slijedi) e_{11} e_{21}
 - bridovi desnog (brid koji prethodi, brid koji slijedi) e_{12} e_{22}
- tablica vrhova
 - jedan brid (bilo koji)
- tablica poligona
 - jedan brid (bilo koji)
- krila brida e su e_{11} e_{21} e_{12} e_{22}



Različite varijante zapisivanja bridova

- orijentacija poligona može biti CW, CCW, određena bridom e ,
- zapis samo 2 krila
- ispitivanje **relacije susjednosti**:
 - da li je vrh V_1 susjedan poligonu F_3 ?
 - da li su poligoni F_1 i F_3 susjedni?
- proizvoljni poligoni (nisu nužno trokuti)

primjer: krilati brid (engl. winged edge)



- redosljed krila brida određen je bridom e

Tablica vrhova				
V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

Tablica bridova							
	11	12	21	22			
e_1	$V_1 V_3$	F_1	e_2	e_2	e_4	e_3	
e_2	$V_1 V_2$	F_1	e_1	e_1	e_3	e_6	
e_3	$V_2 V_3$	$F_1 F_2$	e_2	e_5	e_1	e_4	
e_4	$V_3 V_4$	F_2	e_1	e_3	e_7	e_5	
e_5	$V_2 V_4$	$F_2 F_3$	e_3	e_6	e_4	e_7	
e_6	$V_2 V_5$	F_3	e_5	e_2	e_7	e_7	
e_7	$V_4 V_5$	F_3	e_4	e_5	e_6	e_6	

Tablica poligona	
F_1	e_1
F_2	e_3
F_3	e_5