5. 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, stero 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

4.1. MODELIRANJE OBJEKATA I SCENE

Modelirani objekti:

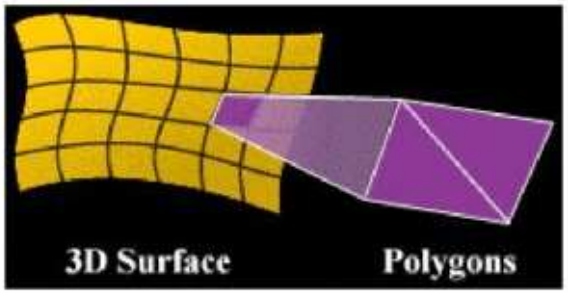
- modeliranje **površine**
  - definirana je vanjska ljuska objekta (“koža”)
    - *poligonima* (*trokuti*), *prednja i stražnja strana*,
    - *parametarska površina*
    - *elementima površine* (engl. surfel) – PBG (point base graphics)
- modeliranje **volumena** tijela
  - definirana je unutrašnjost objekta (*implicitno* definirano)
    - *implicitnim funkcijama*
    - *elementima volumena* (voxel)

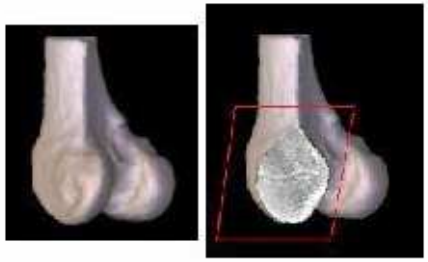
  možemo iz jednog oblika načiniti drugi (nije uvijek jednostavno)

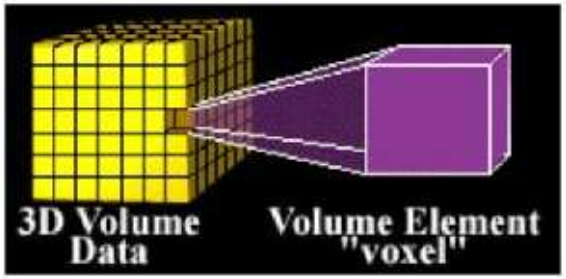
Zapisivanje scene (strukture više razine):

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

površina i volumen







bitna razlika implicitnog i eksplicitnog oblika

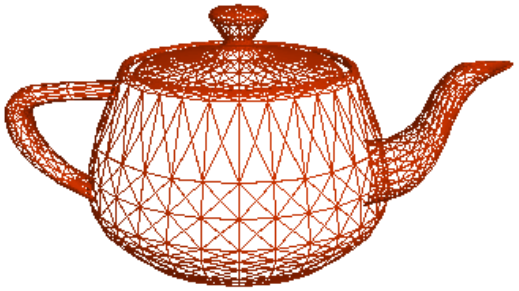
- implicitni
  - jednostavno možemo odrediti

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

    udaljenost od površine

    booleove operacije nad tijelima, detekcija sudara
- eksplicitni, parametarski
  - određivanje točaka površine, tangentnih ravnina

- 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
    - implicitni oblik jednadžbe ravnine

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

    - parametarski oblik jednadžbe ravnine

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

- površina – patametarski zadana

  modeliranje površine (engl. surface modelling)
  - površina je **glatka** i kontinuirano se može kontrolirati (obrada plohe)

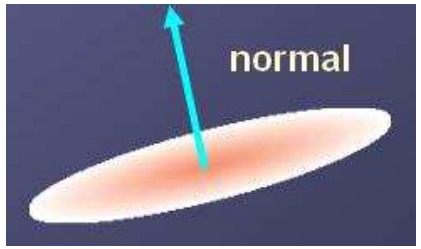
  definirana je površinska ljuska tijela (može biti zatvorena)
  - slobodno oblikovane površine (engl. Freeform surfaces**)**
    - NURBS (B-površine), Bezierove površine,

    - 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

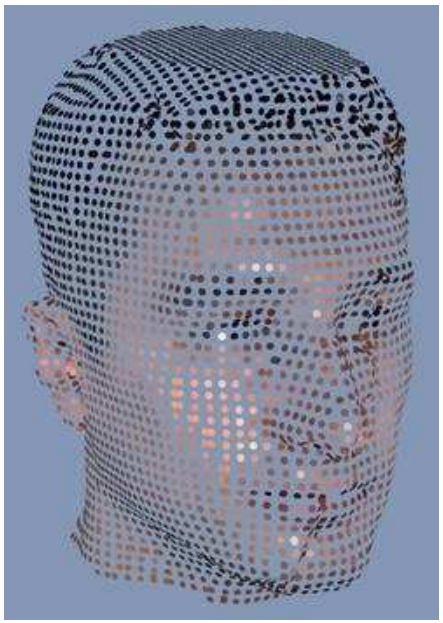
- površina objekta – točkama

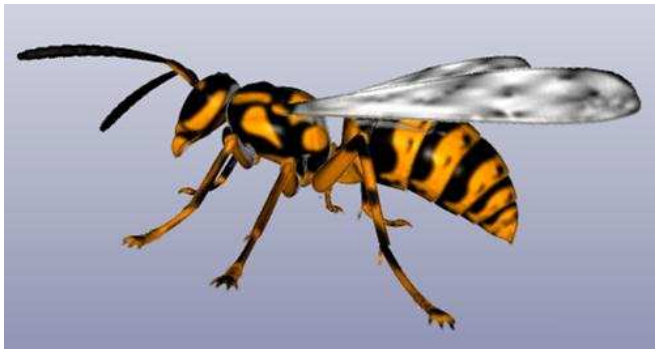
modeliranje elementima površine PBG (engl. point base graphics)

  - ‘+’ 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)

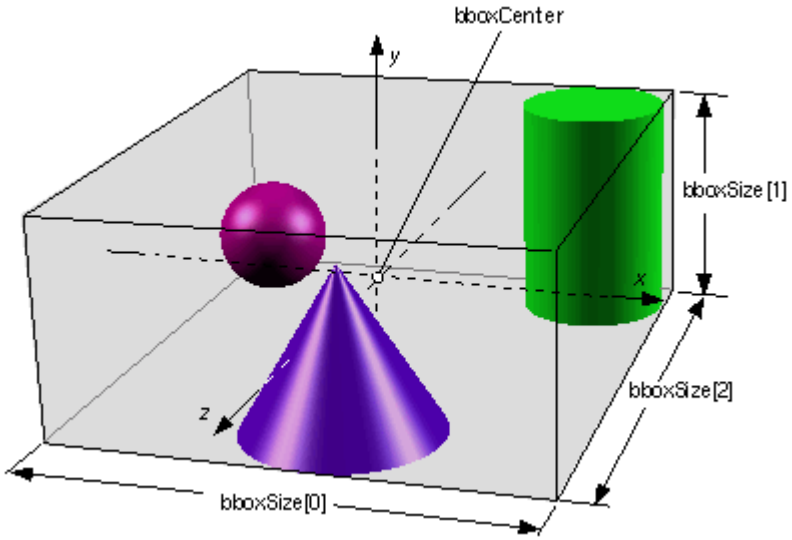
implicitno definirane površine

$$f(x, y, z) = const$$

definirana je unutrašnjost tijela npr:

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

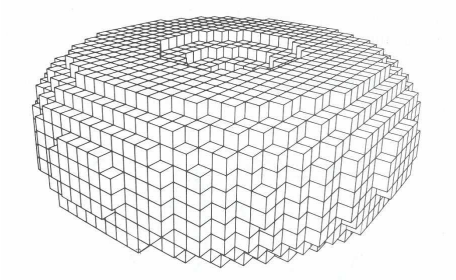
izvan tijela $f(x, y, z) > const.$



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

http://www.infogoaround.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 :

$$f(x, y, z) \leq con1,$$
$$f(x, y, z) \leq con2,$$
$$f(x, y, z) \leq con3, \ldots$$

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

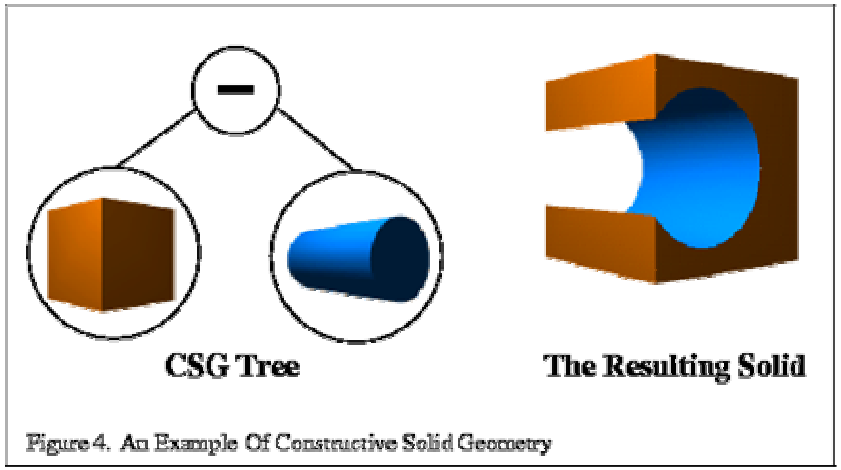
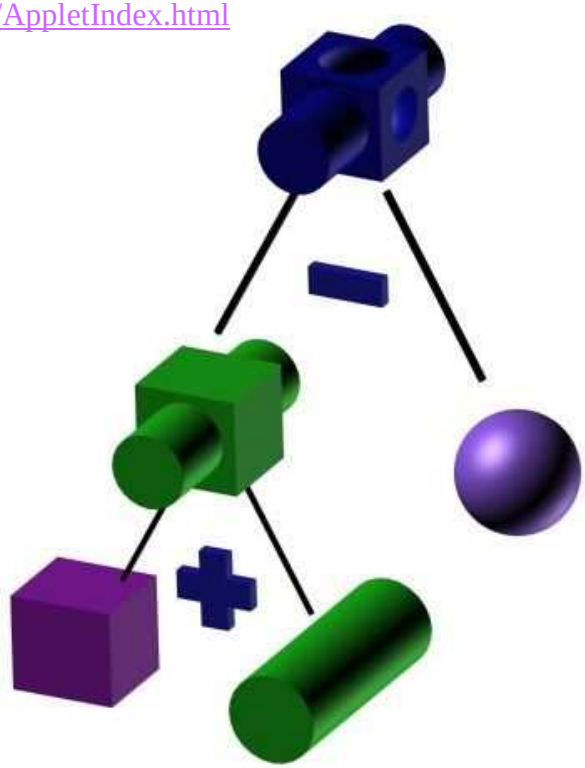


Figure 4. An Example Of Constructive Solid Geometry



– CSG stablo

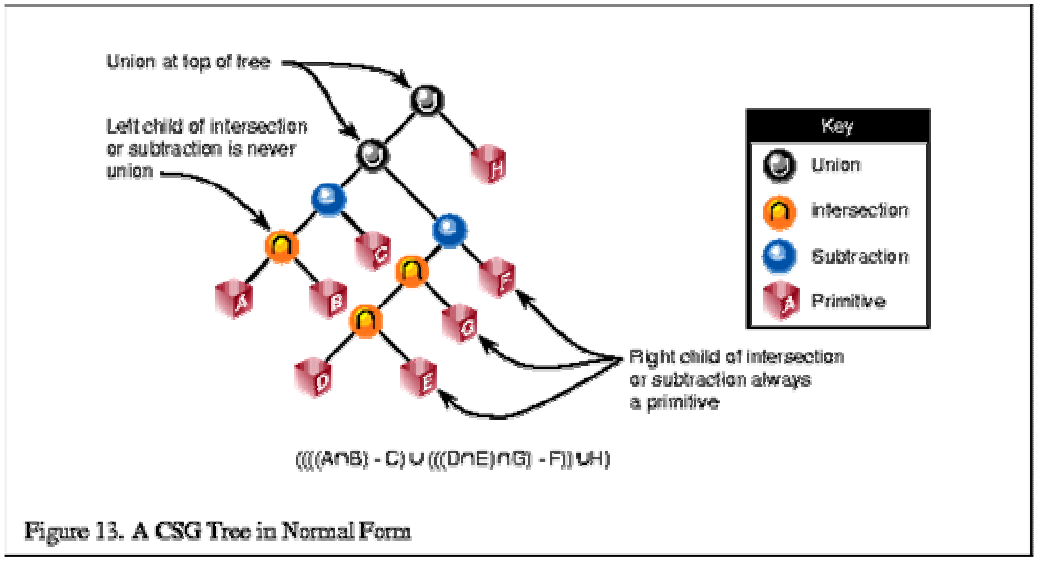
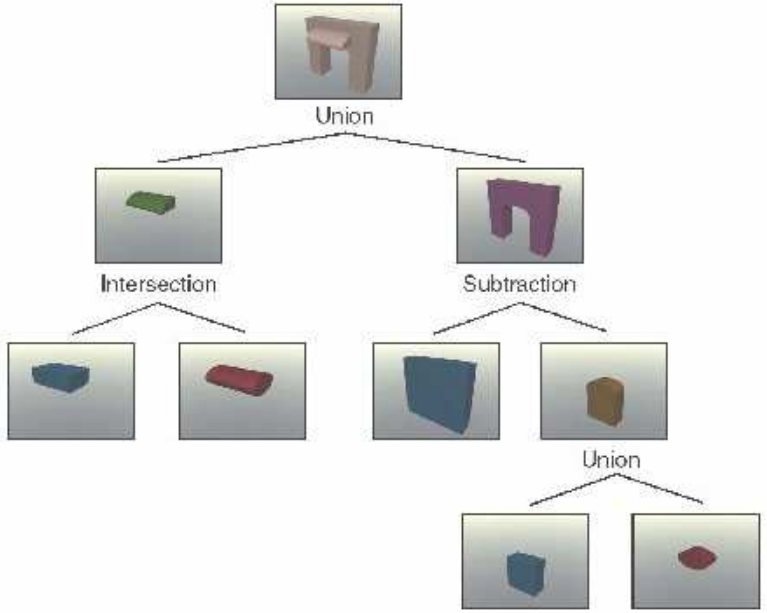
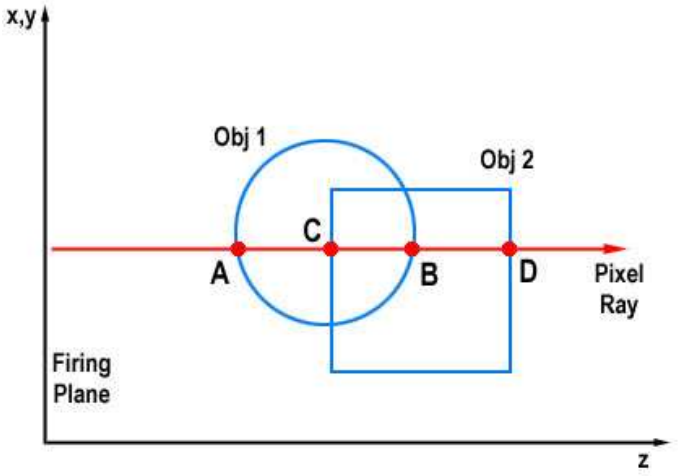


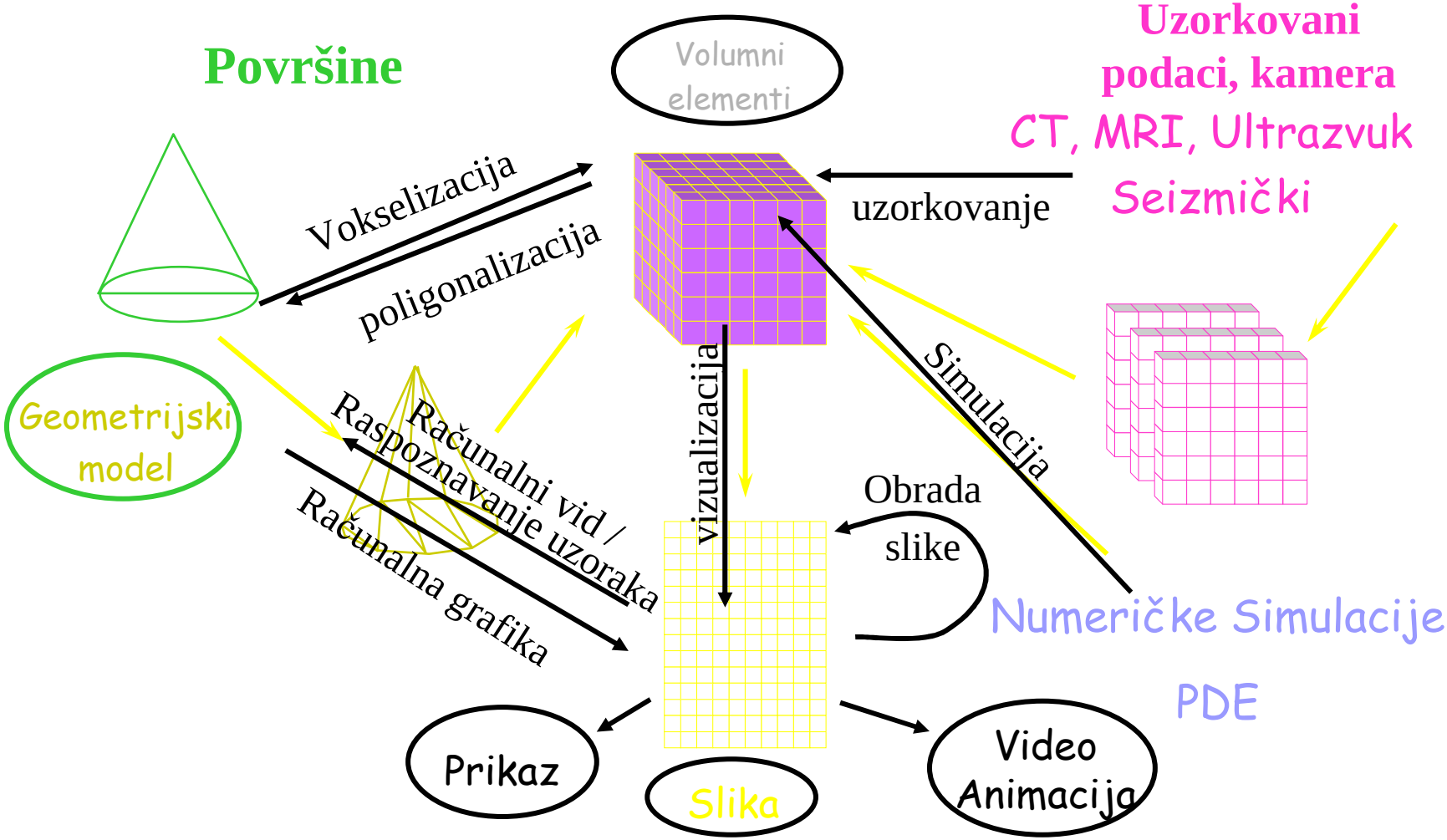
Figure 13. A CSG Tree in Normal Form



Booleove operacije



Objekti

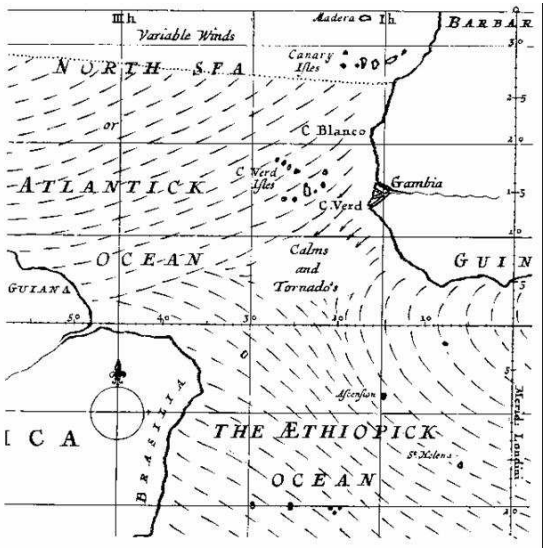

Površine
Volumni elementi
Uzorkovani podaci, kamera
CT, MRI, Ultrazvuk
Seizmički
Vokselizacija
poligonalizacija
uzorkovanje
Geometrijski model
Računalni vid / Raspoznavanje uzoraka
Računalna grafika
vizualizacija
Simulacija
Obrada slike
Numeričke Simulacije
PDE
Prikaz
Slika
Video Animacija

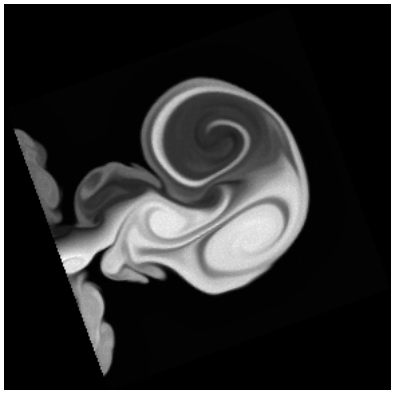
Ostvarivanje prikaza (rendering)

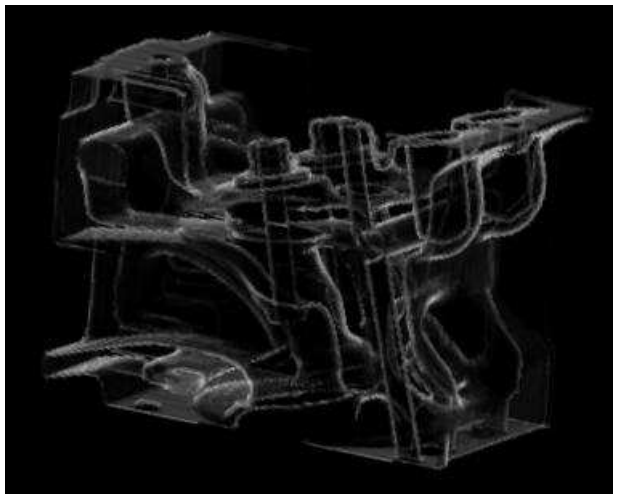
prikaz poligona – klasičan način – fotorealističan prikaz

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)

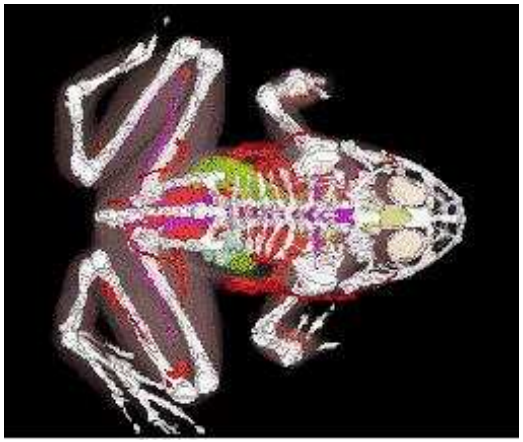
- određuje što će biti i na koji način preslikano u optičke parametre

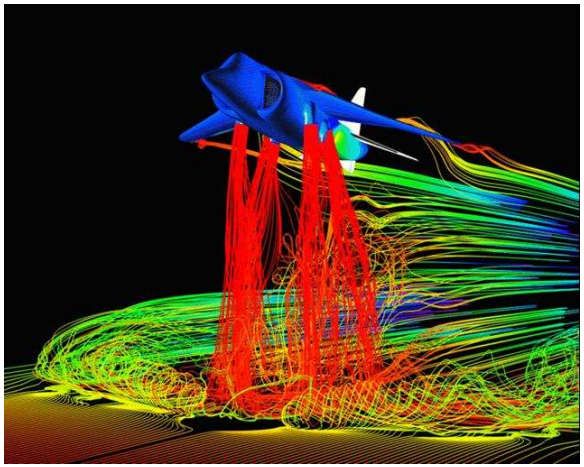
to može biti boja objekta izravno, no može biti i neka druga informacija

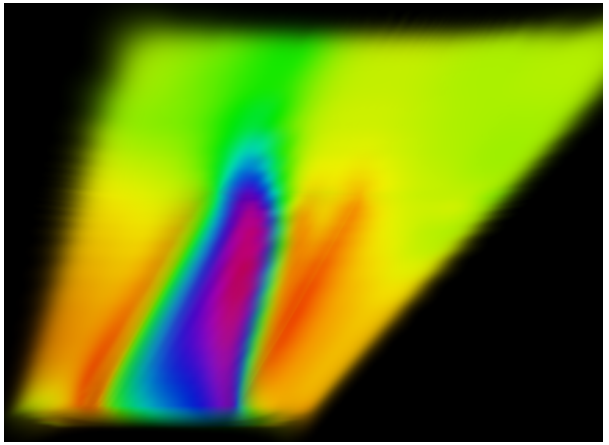
kod NPR tehnike to je mjesto gdje je normala na površinu okomita prema vektoru prema promatraču

upotreba boje za prokaz dodatne informacije

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



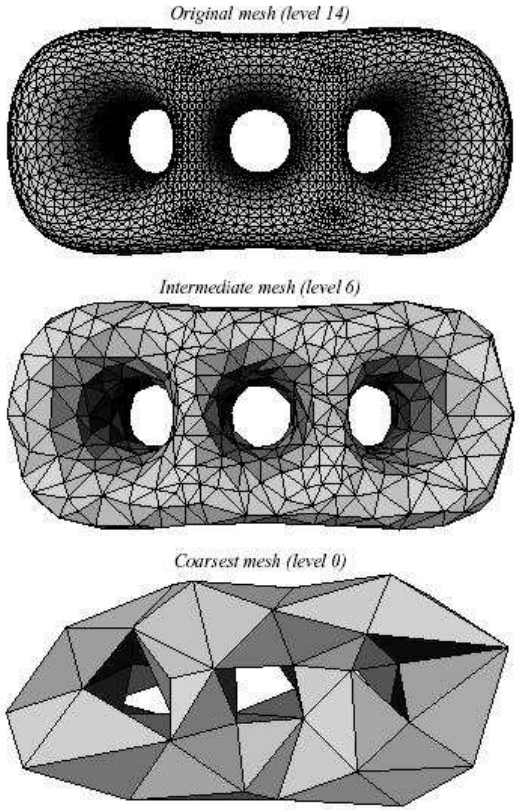




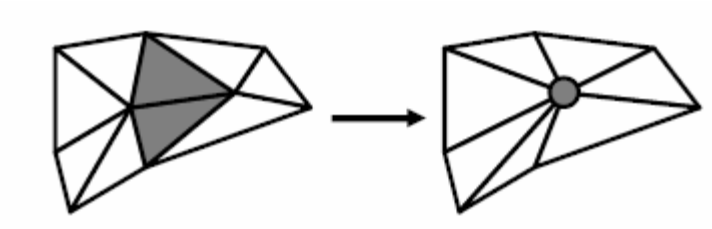
Složeni zapis objekta

- isti objekt nam je često potreban u različitim razinama složenosti (LOD)
  - prikaz objekta ovisno o udaljenosti i veličini prikaza
  - proračun sudara (kolizije)
- ugrubljivanje
  - krećemo od najsitnije podjele

broj vrhova 50 500 2 000



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

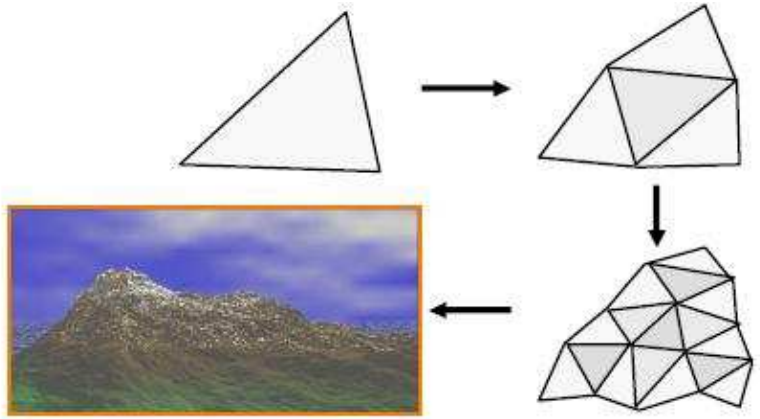


- usitnjavanje poligonalne mreže (engl. Subdivision)
  - dijelimo poligone najgrublje razine i novonastale vrhove pomičemo (tako da novi objekt bude gladak ili hrapav

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

Modeliranje:

http://www.mtl.t.u-tokyo.ac.jp/~takeo/teddy/teddy/ted



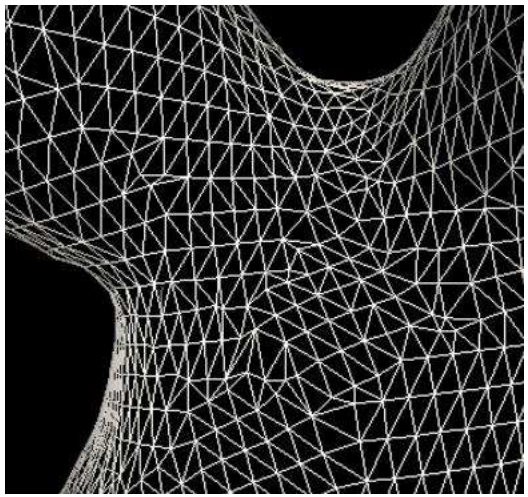
fraktalna podjela pri izradi planine

4.2 REPREZENTACIJA OBJEKATA

Površina objekta – zapis poligonima odnosno trokutima

- objekte najčešće predstavljamo mrežom poligona

(samo površina objekta)

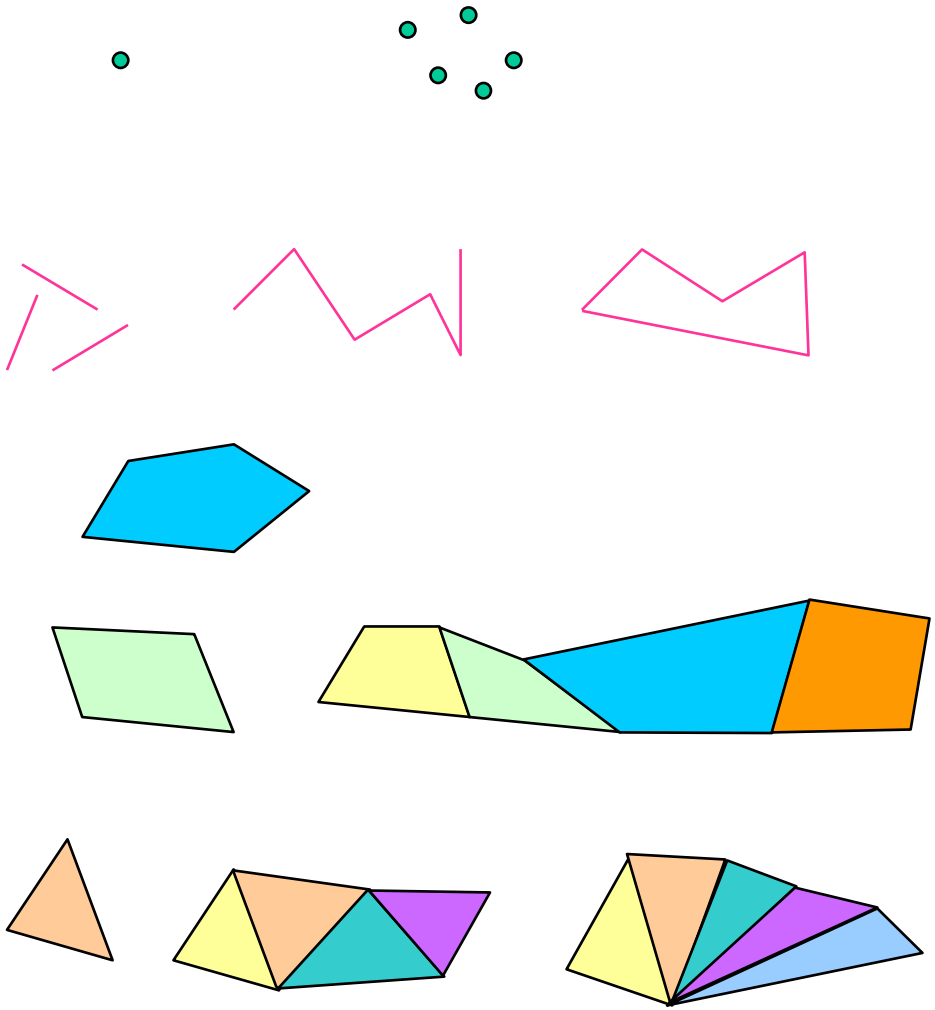


- trokuti – planarni su,
  - sklopovlje GPU podržava trokute
- kako načiniti strukture podataka
- osnovni elementi (poligonalna mreža)
  - **geometrijski podaci** – vrh (točka u prostoru)
  - **atributi** – boja, normala, koordinate teksture
  - **topološki podaci** – brid (povezuje 2 vrha)
    - poligon (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:

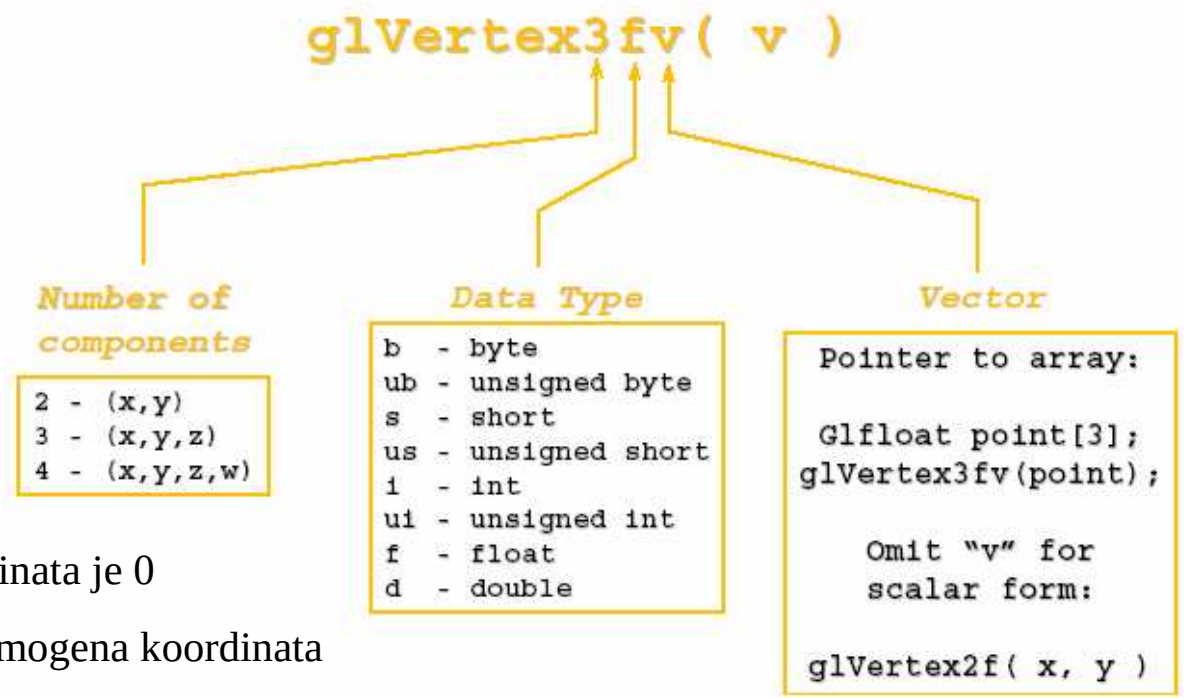
- točke
  - GL_POINTS
- dužina, niz dužina,
  - GL_LINES,
  - GL_LINE_STRIP,
  - GL_LINE_LOOP
- poligon
  - GL_POLYGON,
- četverokut, niz četverokuta
  - GL_QUADS,
  - GL_QUAD_STRIP,
- trokut, niz trokuta
  - GL_TRIANGLES
  - GL_TRIANGLE_STRIP
  - GL_TRIANGLE_FAN



- primjer naredbe u OpenGL-u primjer u Tutor – shapes.exe

```
GLfloat x1, y1;    // zbog različitih realizacija u OpenGL-u je definiran tip GLfloat, GLint …
                                        // ekvivalentan kao u C-u    - prenosivost
glBegin (Primitiva)   // Primitiva je npr. GL_LINES
        glVertex2f (x1, y1);
        glVertex2f (3,27, 5,226);
glEnd;
```

Primitiva – određuje na koji način će točke koje slijede biti kombinirane



2 - z koordinata je 0

4 - w je homogena koordinata

Izravan način:

- individualan prijenos podataka (ne uzimamo u obzir da su neki vrhovi zajednički)

```
glBegin(GL_TRIANGLES);       //slijede podaci - trokuti
      glVertex3f (1f, 2.2f, 3.4f);
      glVertex3f (x1, y1, z1);
      glVertex3f (x2, y2, z2);
glEnd();
```

- **indeksirani pristup** (polje podataka vrhova)

```
vrh[0] = {x0,y0,z0};
vrh[1] = {x1,y1,z1};
vrh[2] = {x2,y2,z2};

glBegin(GL_TRIANGLES);            // indeksirani vrhovi
   for (i = 0; i < 3; i++) {
       glNormal3fv (normala[i]);      // atributi za vrh i
       glVertex3fv (vrh[i]);
  }
glEnd();
```

- nije efikasno, za svaki vrh se poziva glVertex(),
  puno **funkcijskih poziva** i prijenosa podataka može biti **vrlo sporo**

Polje vrhova i primitiva GL_LINE_STRIP:

- primjer – spremnik vrhova

```
glVertex2fv();          // prenosi se polje vrhova, a ne vrh po vrh
```

- polje 100 točaka - 2D

```
GLfloat mojeTocke [100][2];     // obično dinamički alociramo koliko je potrebno
                                // (pročitamo iz datoteke koliko ima vrhova)
```

- definiranje tipa točke 2Df i polja od 3 točke

```
typedef GLfloat tocka2f[2];
tocka2f mojeTocke [3] = {{0.2, 2}, {1.3, 4}, {2.2, 1.5}};
```

mojeTocke [0] = ?, mojeTocke [2] [0] = ?

```
glBegin (GL_LINE_STRIP) ;
    glVertex2fv (&mojeTocke [0]);
glEnd;
```

- na ovaj način poslati ćemo odjednom niz točaka koje povezuje linija

- želimo jedan funkcijski poziv i odjednom poslati veću količinu podataka

```
glVertexPointer (3, GL_FLOAT, 0, &vertices[0]);  // (2,3,4) 3 – x, y, z, koordinate vrha
                // 0 – stride – razmak offset [byte] između uzastopnih vrhova, ako su npr vrhovi u strukturi
glEnableClientState(GL_VERTEX_ARRAY);
glDrawArrays(GL_TRIANGLES, 0, num_vertices);   // nema glBegin(), glEnd()
```

korištenja priručne memorije (engl. cache) vrhova

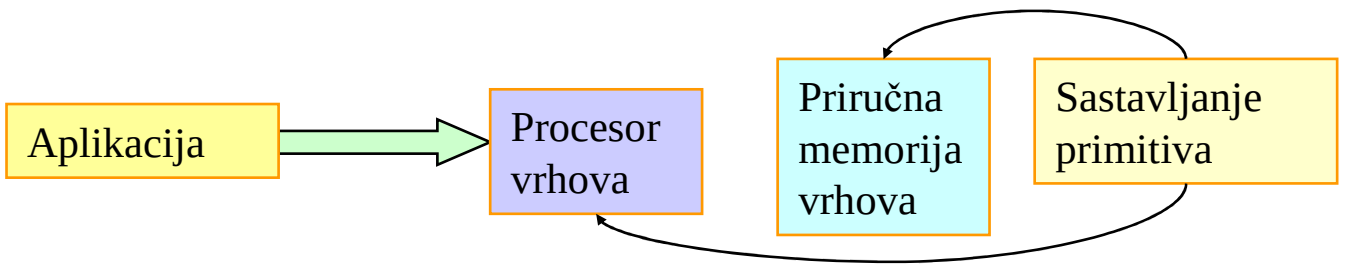
  – procesiranje vrhova je sekvencijsko

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

  – slučajni pristup vrhovima – omogućeno je dijeljenje vrhova

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

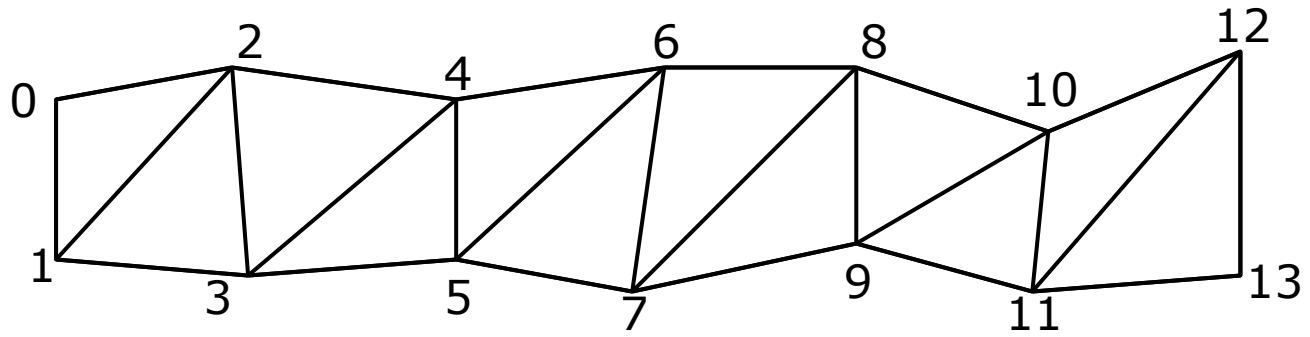
- u priručnoj memoriji su transformirani vrhovi
- neki vrhovi se višestruko ponavljaju, tj. za svaki trokut su vrhovi zasebno navedeni



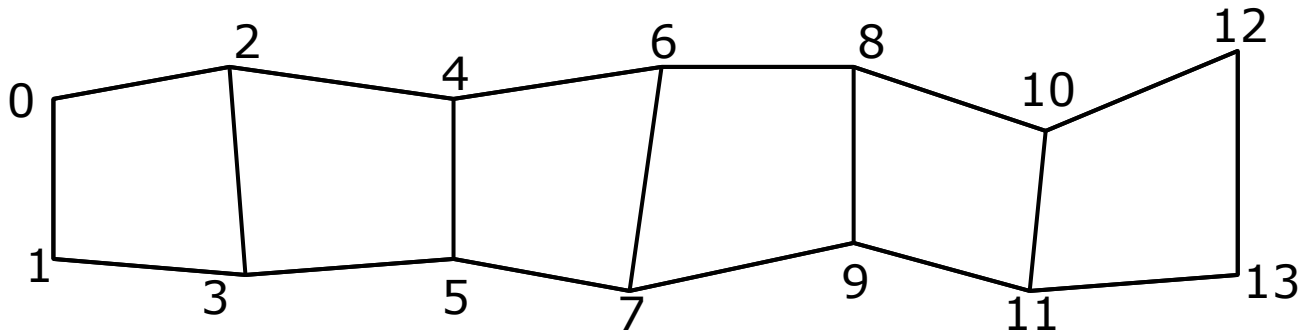
Niz trokuta (triangle strip):

- 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]);
```



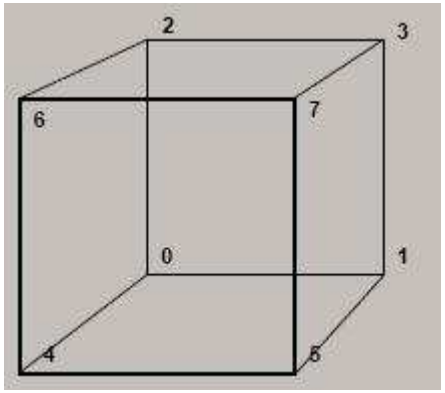
GL_QUAD_STRIP za n četverokuta imamo $2n + 2$ vrha

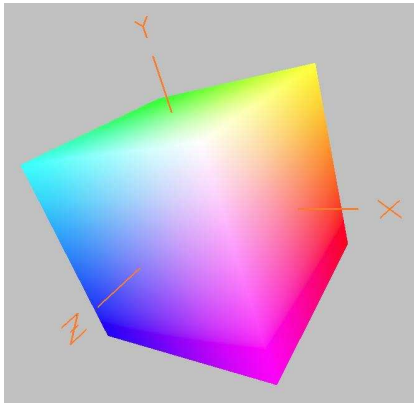


```
// KOORDINATE
static GLfloat CubeVertices[ ][3] =
{
        { -1., -1., -1. },
        { 1., -1., -1. },
        { -1., 1., -1. },
        { 1., 1., -1. },
        { -1., -1., 1. },
        { 1., -1., 1. },
        { -1., 1., 1. },
        { 1., 1., 1. }
};
```

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

Primitive –

GL_TRIANGLE_STRIP

Promjena stanja –

glPointSize(size);
glLineStipple(repeat, pattern);
glShadeModel(GL_SMOOTH);

Aktiviranje mogućnosti –

glEnable(GL_LIGHTING);
glDisable(GL_TEXTURE_2D);

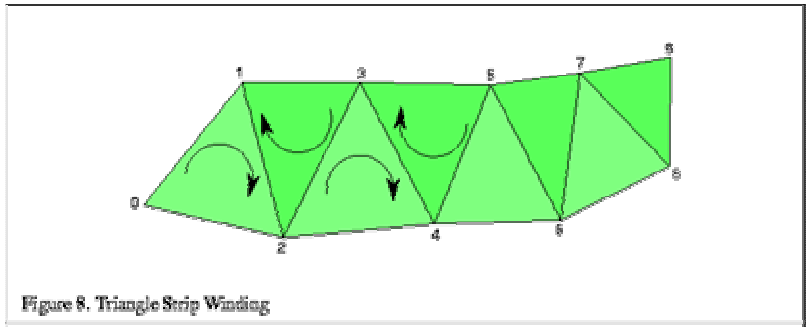


Figure 8. Triangle Strip Winding

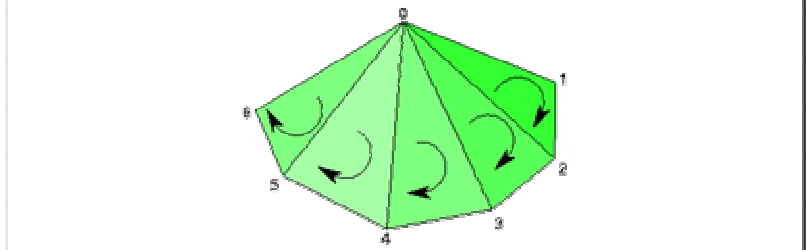


Figure 9. Triangle Fan Winding

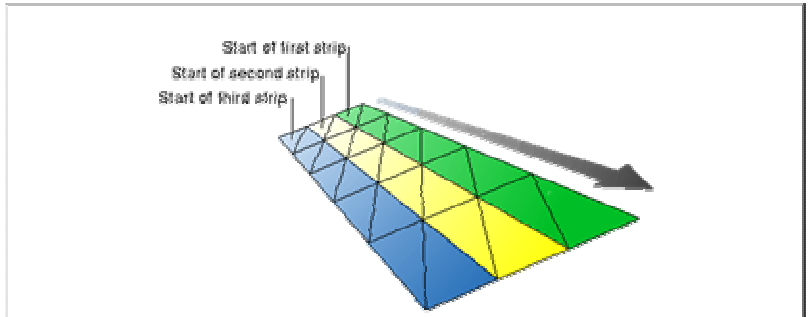
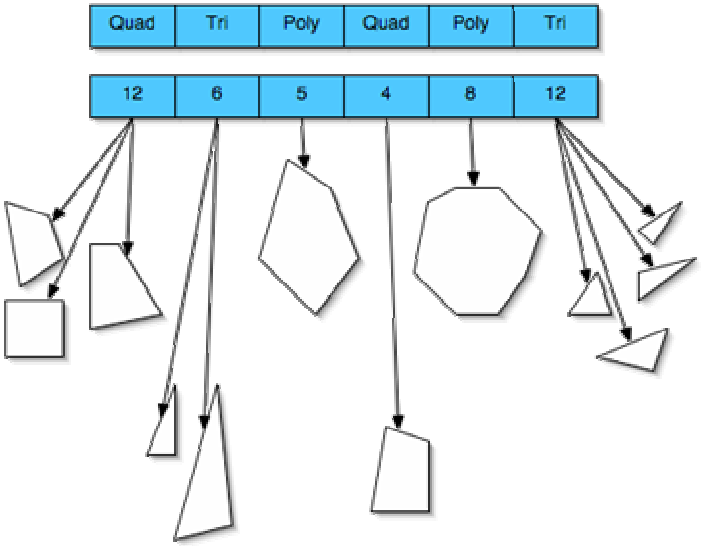
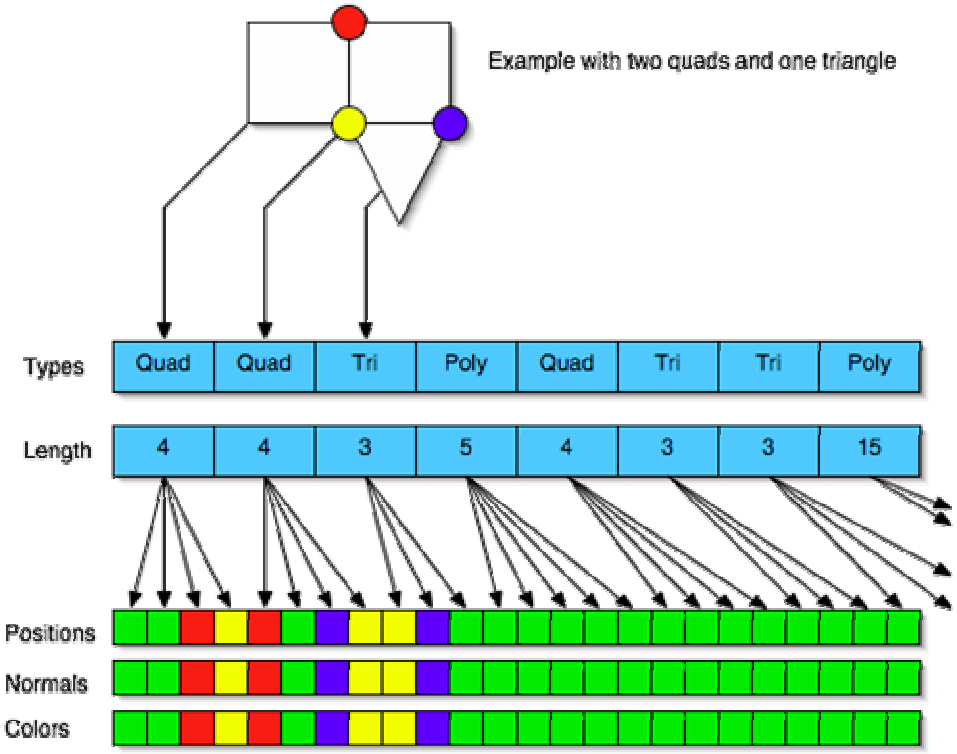


Figure 10. A Mesh Made up of Multiple Triangle Strips

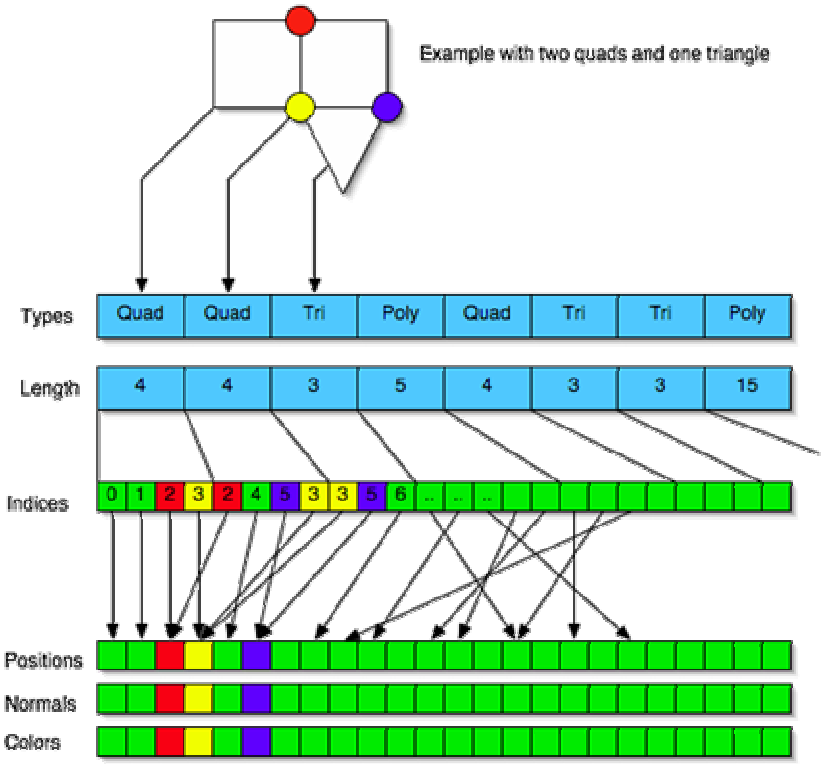
- Primjeri organizacije podataka i atributa:
  - http://www.gris.uni-tuebingen.de/grisalt/projects/grdev/doc/html/Overview.html

  - bez indeksiranja
    - višestruki podaci
    - npr. za eksploziju poligona

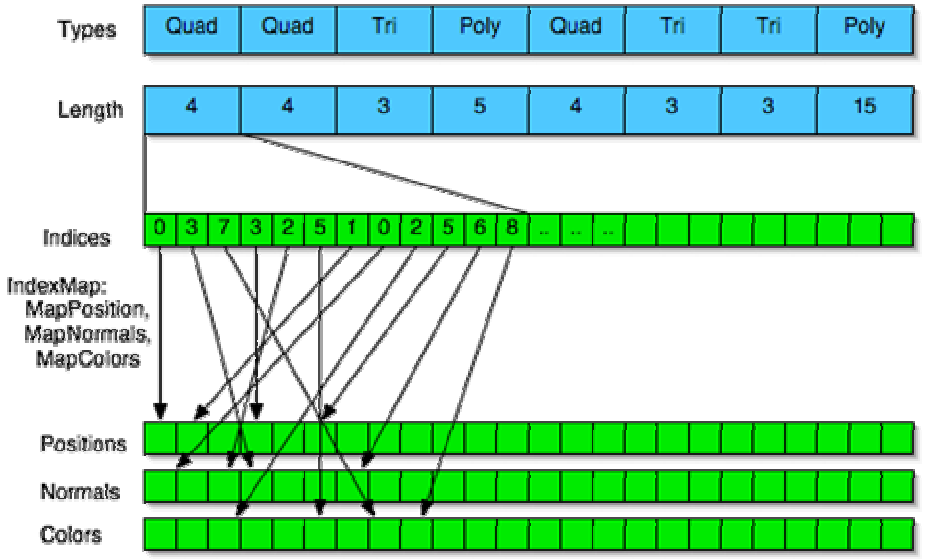


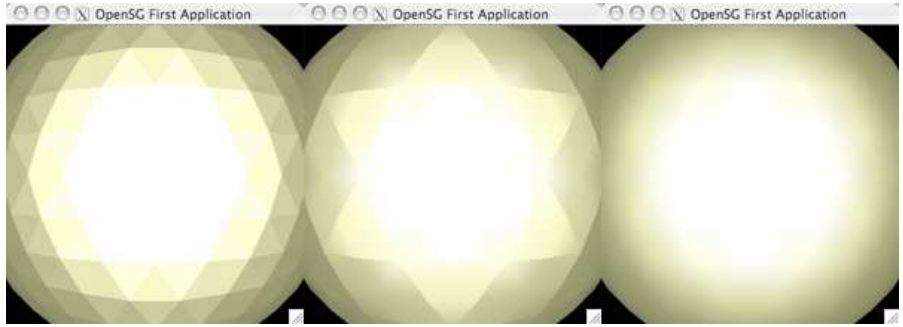


– indeksirani pristup
  - zajednički podaci za pojedini vrh
  - npr. sjenčanje Gouraud



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





4.3 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)

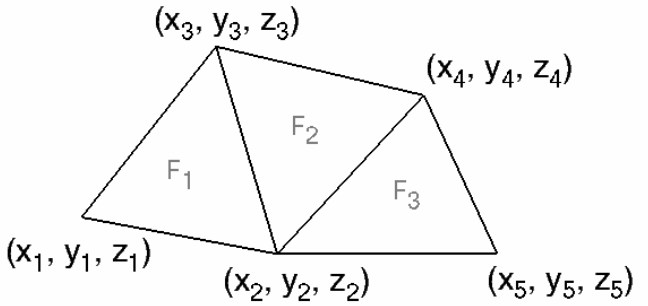
Strukture podataka

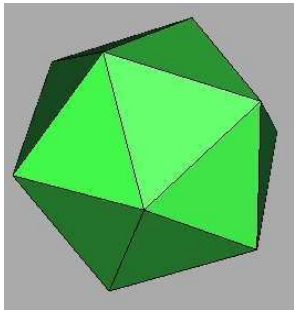
1. Tablica poligona
2. Tablice vrhova i poligona
3. Tablica bridova, vrhova i poligona
4. Liste susjednosti
5. Krilati brid

1. Tablica poligona

- nužno ako objekt “eksplodira” ili se poligoni raspršе, 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
- redoslijed vrhova je u primjeru suprotno smjeru kazaljke na satu gledano izvan tijela CCW (određuje redoslijed 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)



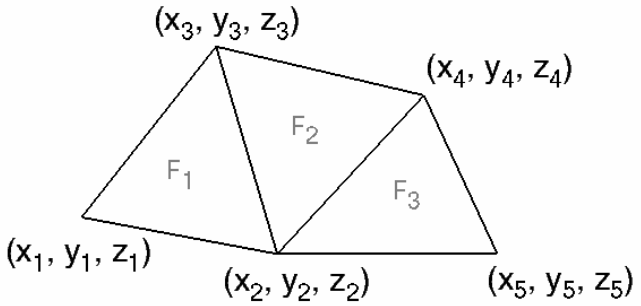


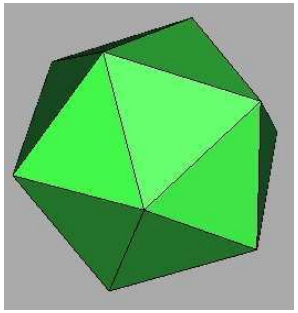
2. Tablice vrhova i poligona

- 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

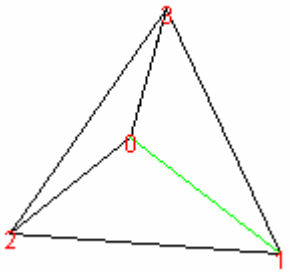
	X	Y	Z
0	0.453608	0.89043	0.037037
1	0.544331	-0.62854	-0.555555
2	-0.090722	-0.366647	0.925925
3	-0.907218	0.104757	-0.407407

EDGE-LIST

	P0	P1
0	0.0	2.0
1	2.0	1.0
2	1.0	0.0
3	1.0	3.0
4	3.0	0.0
5	3.0	2.0

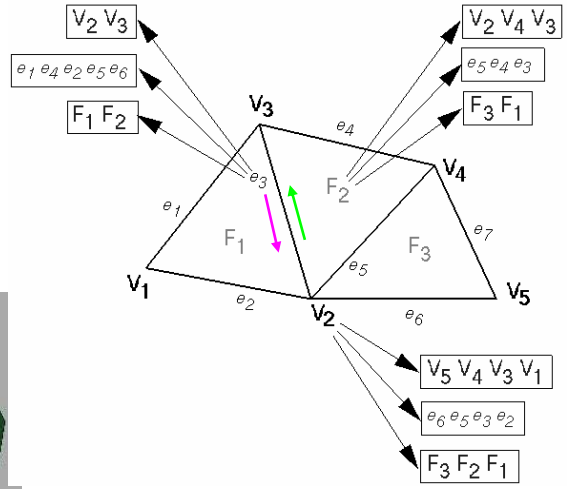
FACE-LIST

	K0	K1	K2
0	0.0	1.0	2.0
1	2.0	3.0	4.0
2	4.0	5.0	0.0
3	1.0	5.0	3.0

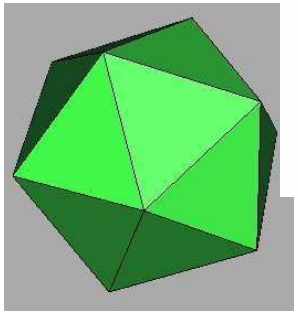


4. Liste susjednosti

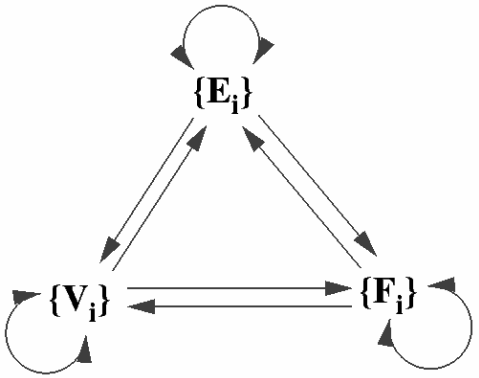
- tablica bridova
  - koji vrhovi čine brid (orijentacija brida V_2V_3, V_3V_2)
  - koji bridovi su susjedni (diraju prvi ili drugi vrh)
  - koji poligoni čine brid



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



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



- želimo imati informaciju o susljednosti ali

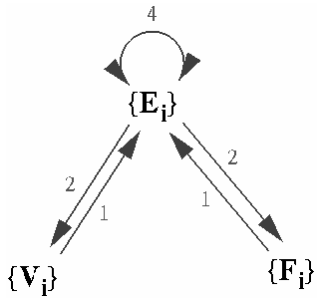
želimo pohranjivati manje podataka

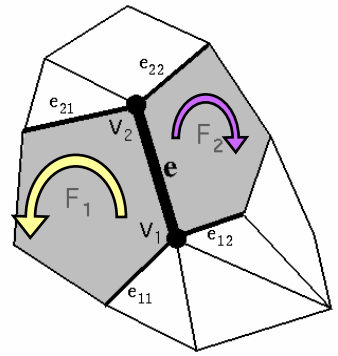
kompromis: potrebna memorija ↔ vrijeme potrebno za određivanje susjednosti

- 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)





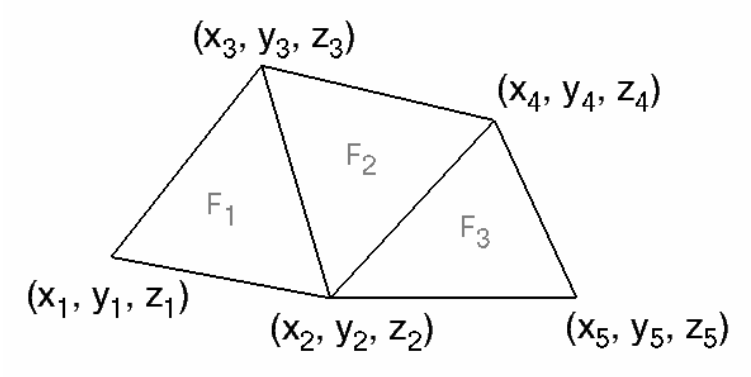
- krila brida **e** su e_{11} e_{21} e_{12} e_{22}

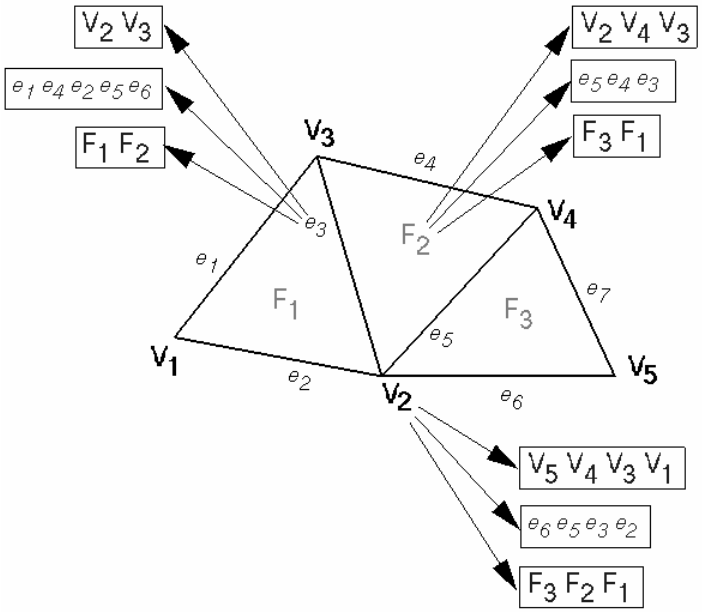
Različite varijante zapisivanja bridova

- 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)





- redoslijed 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	$V3$	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