

RAČUNARSKA GRAFIKA 2023/2024  
Prof. dr Dragan Ivetić 

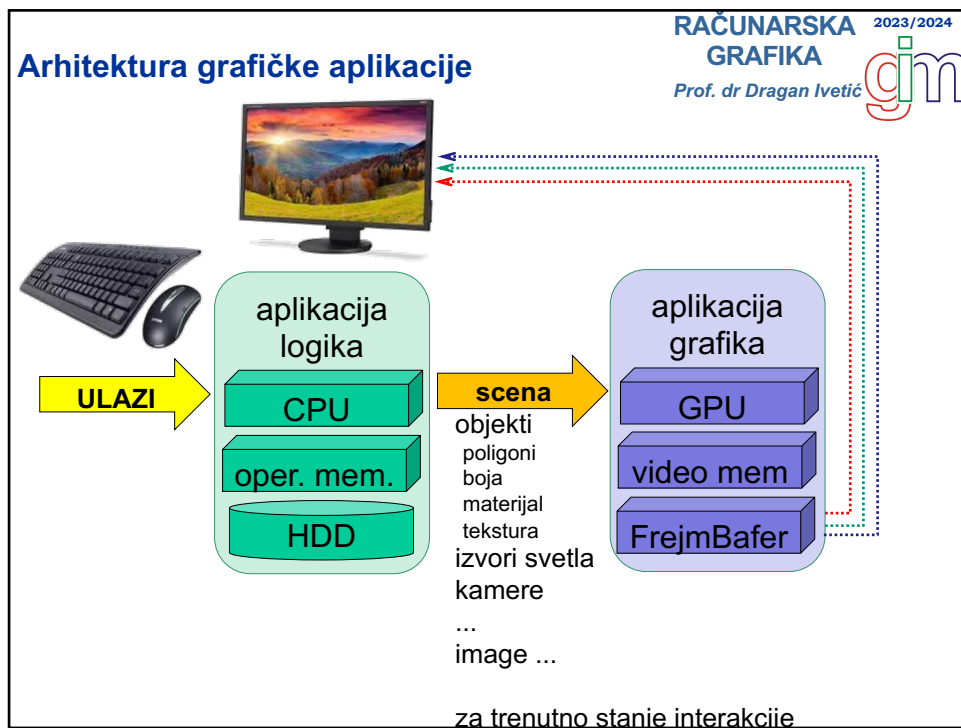
## Osnovni pojmovi i procesi

Samo 3D esencijalni i perceptivni:

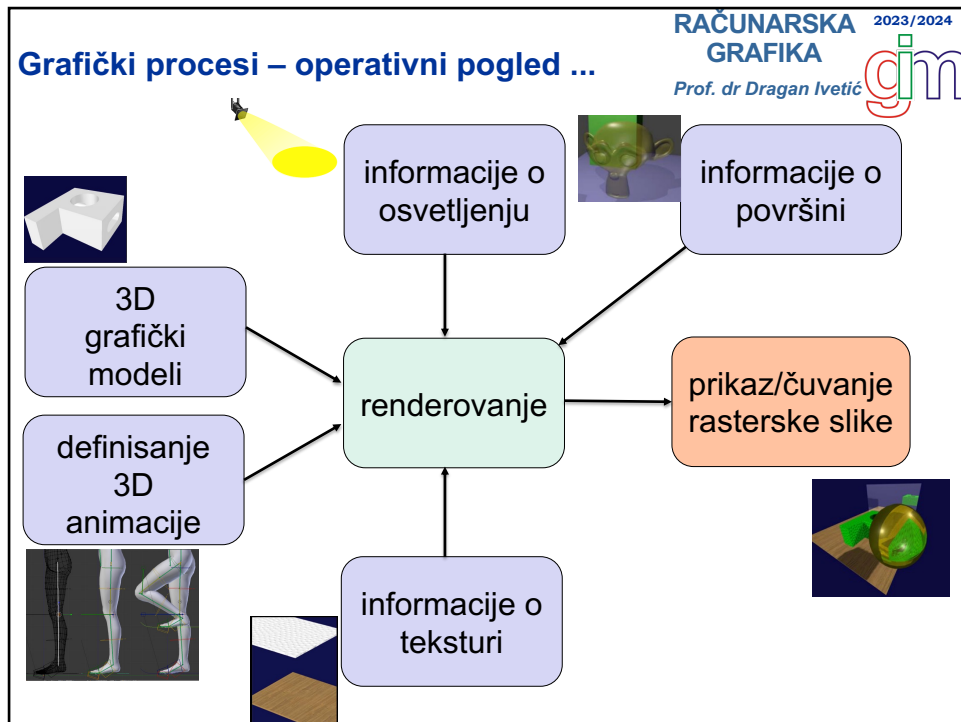
- Tačka, **vertex**, **vertices**
- Linija/linije, **edge**
- Trouglovi,
- Transformacije, 
- 3D/2D projekcija/skrivanje,
- Kliping,
- Bojenje,
- Senčenje,
- Teksture,
- Spajanje.



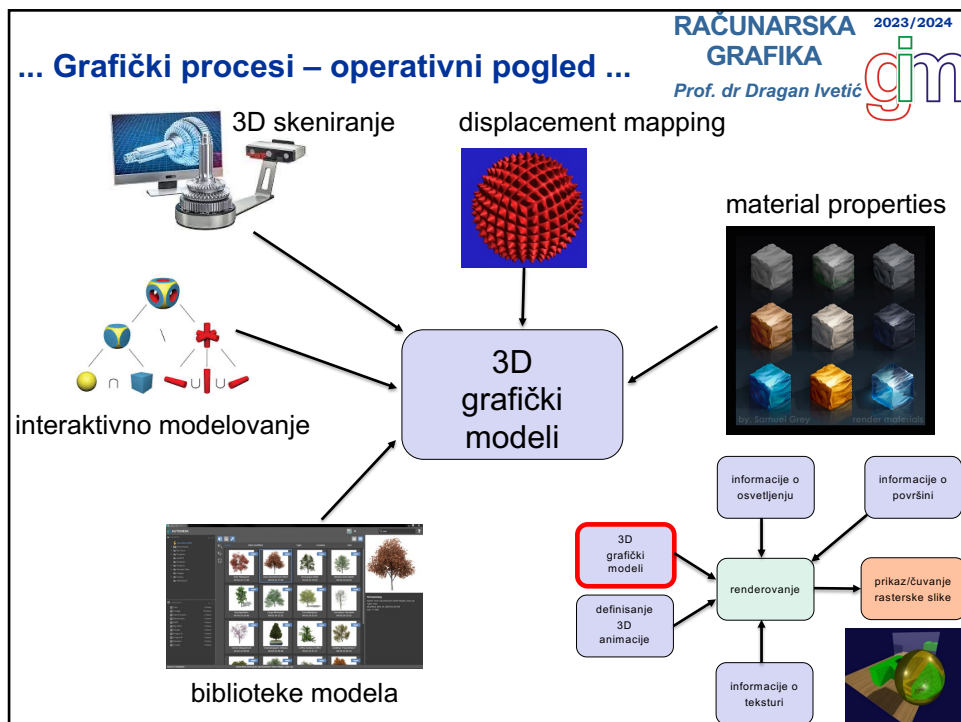

1



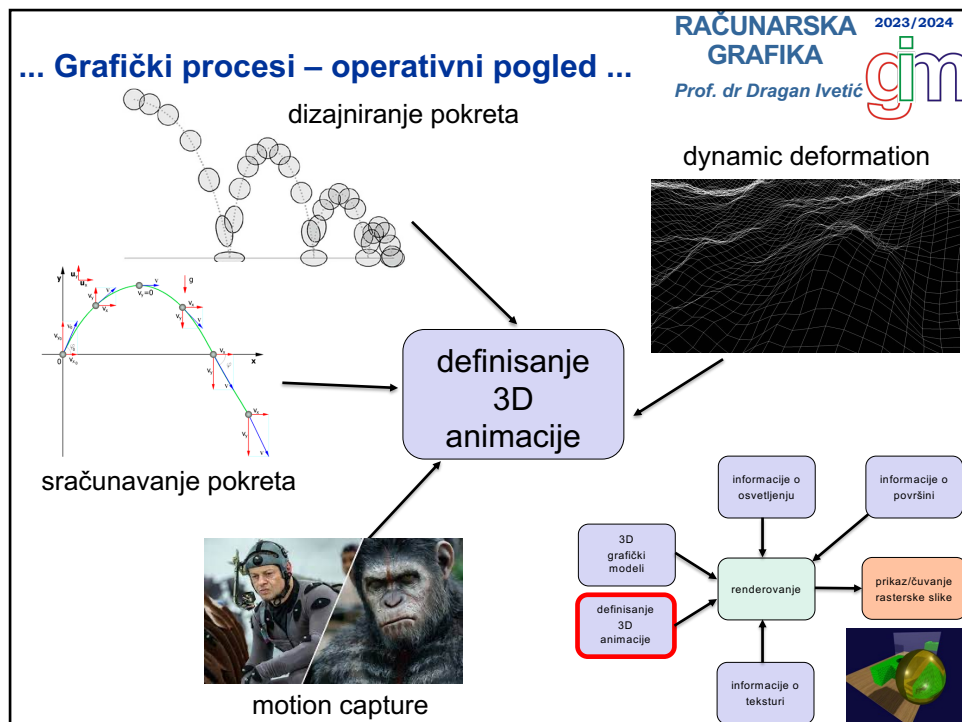
2



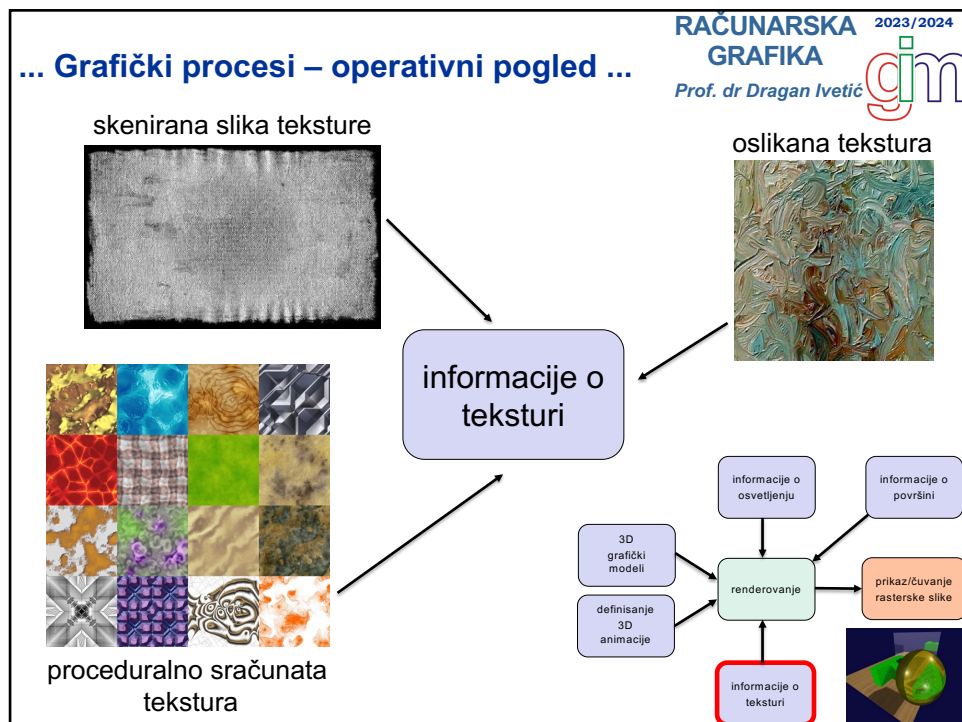
3



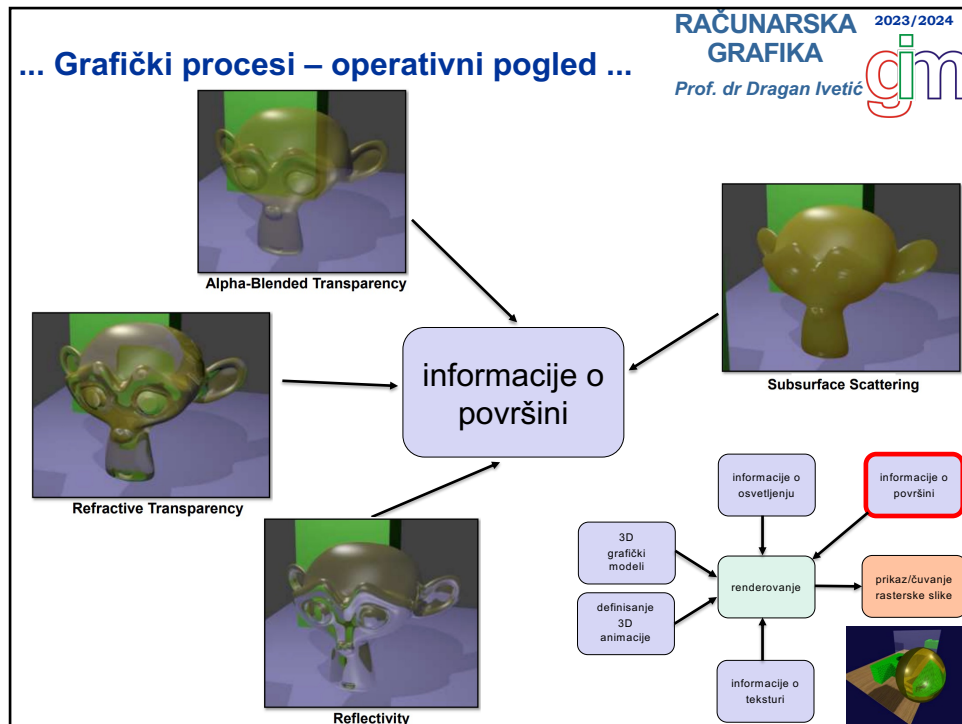
4



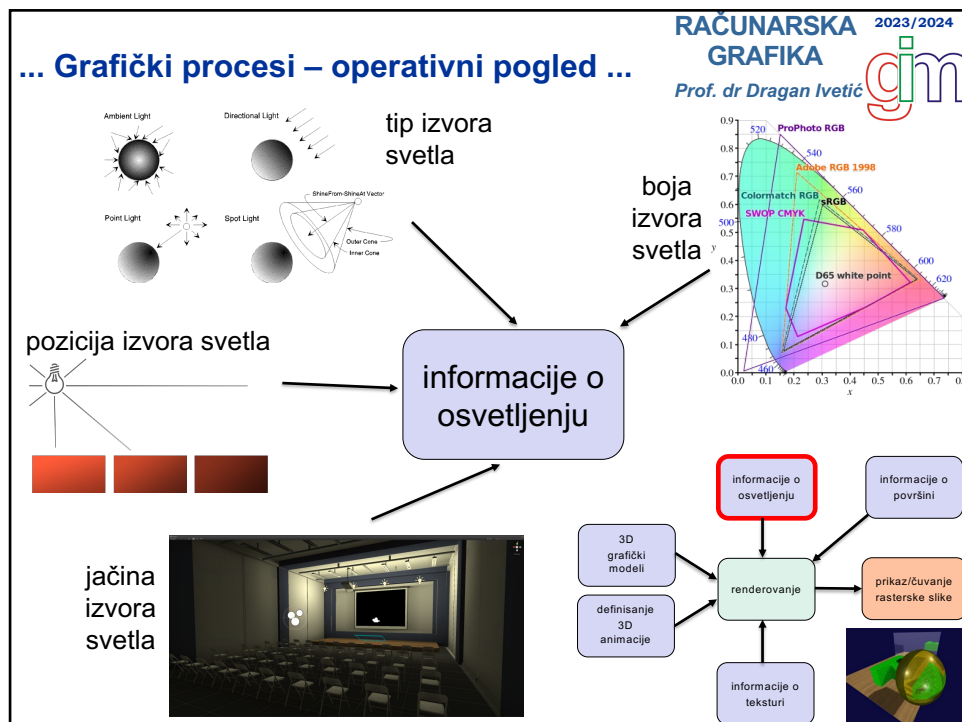
5



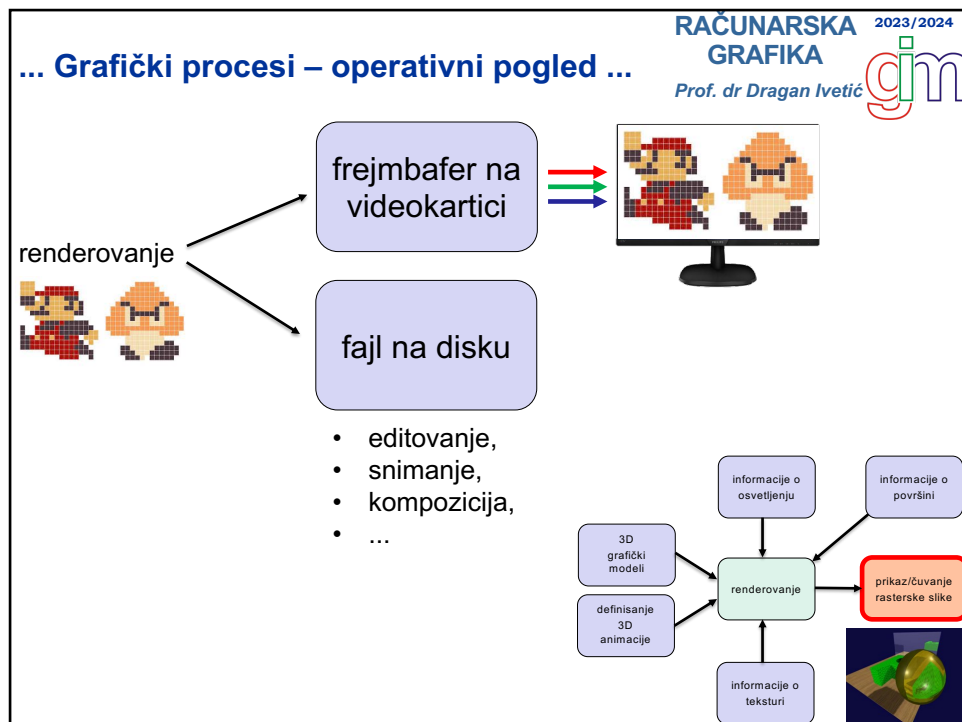
6



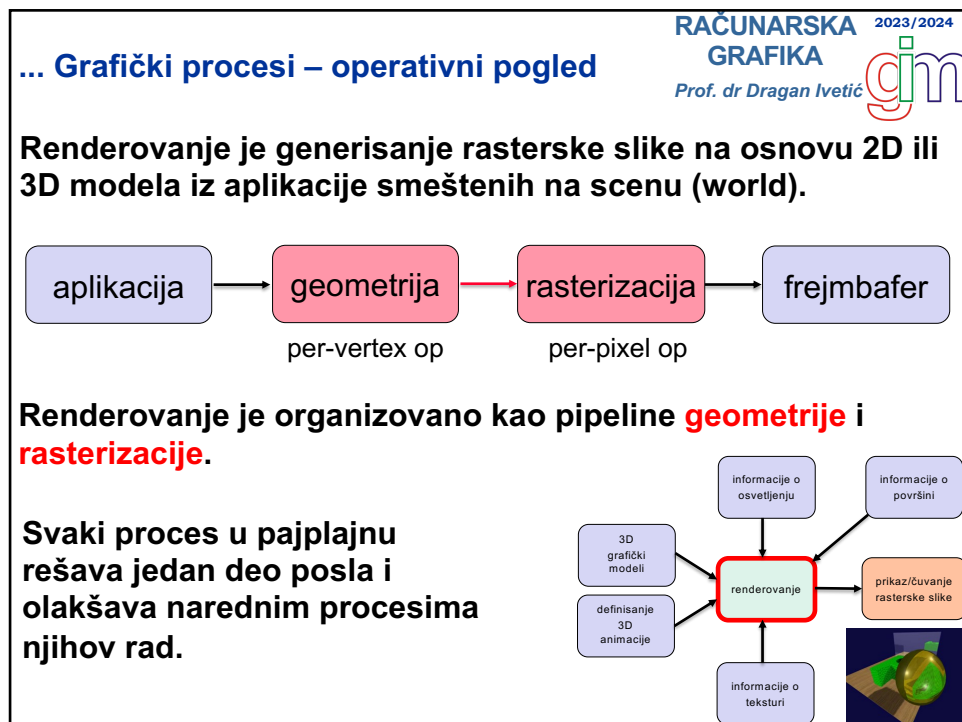
7




8



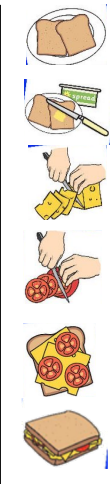
9



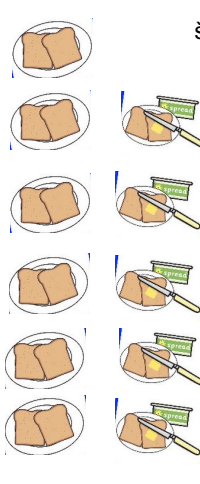
10

RAČUNARSKA GRAFIKA 2023/2024  
Prof. dr Dragan Ivetić 


### Šta je pajplajn?



jedan kuvar, 6 x 20s,  
novi sendvič svaka 2 minuta



šest kuvara




novi sendvič svakih 20s,  
osim prvog koji za 120s

Šta ako treba mazati obe kriške hleba?  
Koraci 3-6 dangube po 20s!

Zaposliti dodatnog kuvara za korak 2 da istovremeno  
mažu obe kriške hleba!

11

RAČUNARSKA GRAFIKA 2023/2024  
Prof. dr Dragan Ivetić 

### Osnovni grafički pajplajn – GEOMETRIJA ...

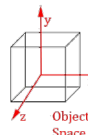
Model&View Transform

Vertex Shading

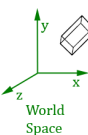
Projection

Clipping

Screen Mapping



Object Space



World Space

- 3D scena komponovana instanciranjem modela (objekata)
- svaki model sa svojim atributima i **object koord. sistem**,
- **ModelTransform** postavlja model na željenu poziciju u **world coordinates (space)** – scena,
- npr. kako kocku u svom model (object) space (-5-55, 5-55, 555, -555, -5-5-5, 5-5-5, 55-5, -55-5) postavimo na scenu tako da njen centar bude na (a,b,c)? Ali zarotiranu za α po Z osi?

**u homogenizovanom 3D prostoru**

$$\begin{bmatrix} X_n \\ Y_n \\ Z_n \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} + \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$\begin{bmatrix} X_n \\ Y_n \\ Z_n \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix}$$

12



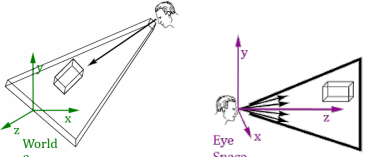
RAČUNARSKA 2023/2024  
GRAFIKA  
Prof. dr Dragan Ivetić

**...Osnovni grafički pajplajn – GEOMETRIJA...**

**GEOMETRIJA**

Model&View Transform → Vertex Shading → Projection → Clipping → Screen Mapping

- ViewTransform mapira **world coordinates** temena svakog objekta u **koord. system kamere** da bi olakšala naredne aktivnosti...
- na koji korisnik gleda iz neke tačke scene “kroz kameru” što se zove camera (**eye**) **space**,
- ako je aktivna kamera u WC na (a,b,c) onda je ViewTransformacija



$$\begin{bmatrix} X_n \\ Y_n \\ Z_n \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -a \\ 0 & 1 & 0 & -b \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix}$$

13

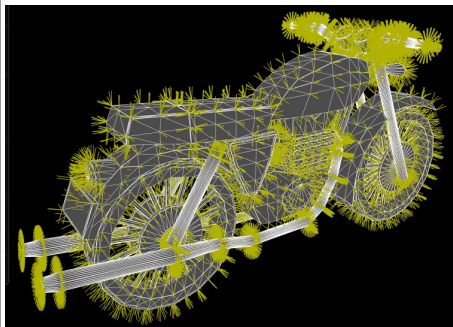
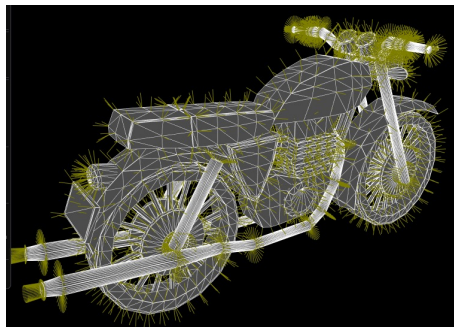
RAČUNARSKA 2023/2024  
GRAFIKA  
Prof. dr Dragan Ivetić

**...Osnovni grafički pajplajn – GEOMETRIJA...**

**GEOMETRIJA**

Model&View Transform → Vertex Shading → Projection → Clipping → Screen Mapping

- nemoguće kreirati ili ukloniti teme, ali može podatke koji su pridruženi temenu,
- sigurno sračunava srednju (usrednjenu) normalu,
- a može i boju u temenu.

14

RAČUNARSKA GRAFIKA 2023/2024  
Prof. dr Dragan Ivetić

### ...Osnovni grafički pajplajn – GEOMETRIJA...

**GEOMETRIJA**

Model&View Transform → Vertex Shading → **Projection** → Clipping → Screen Mapping

- paralelna ili perspektivna projekcija na view volume (unit cube), Matricom projekcije
- često je “uparalelizovana” sa narednom fazom,
- model je sada definisan u normalized device coordinates – **image space**,

Orthographic [Song Ho Ahn] Perspective Image Space

15

RAČUNARSKA GRAFIKA 2023/2024  
Prof. dr Dragan Ivetić

### ...Osnovni grafički pajplajn – GEOMETRIJA...

**GEOMETRIJA**

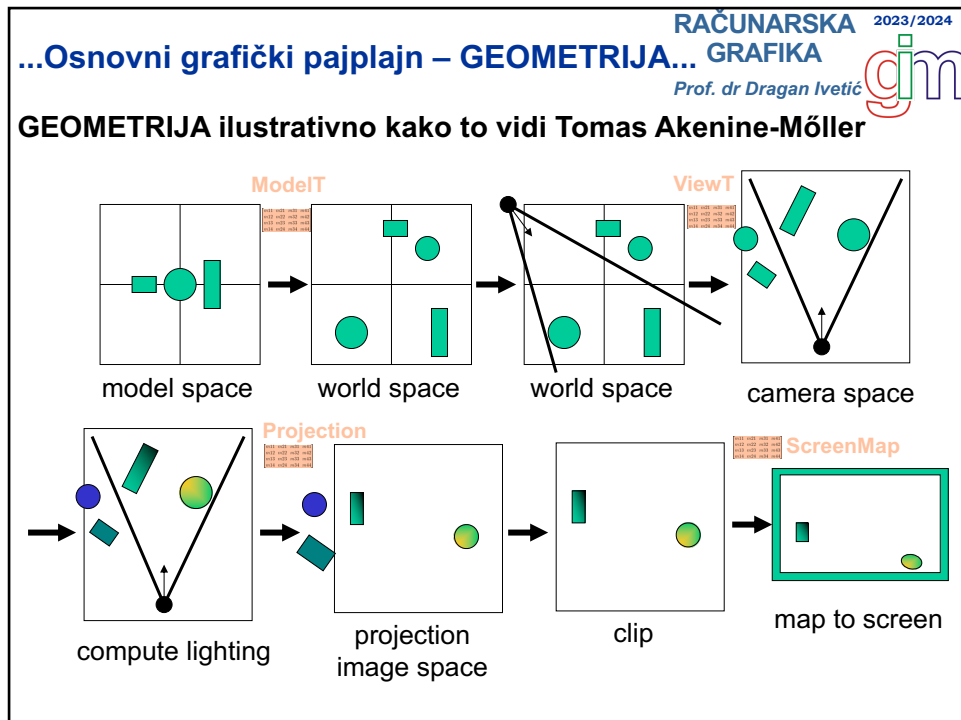
Model&View Transform → Vertex Shading → Projection → **Clipping** → **Screen Mapping**

- veličina 3D scene postaje invarijantna na mogućnosti prikaza,
- u unit cube stoje zahvaćena temena i novonastala temena koja su nastala isecanjem primitiva.
- x i y koordinate temena u/na unit cube, preslikavaju se u koordinate ekrana (int tip), a z se prosleđuje kao (0 ... 1).

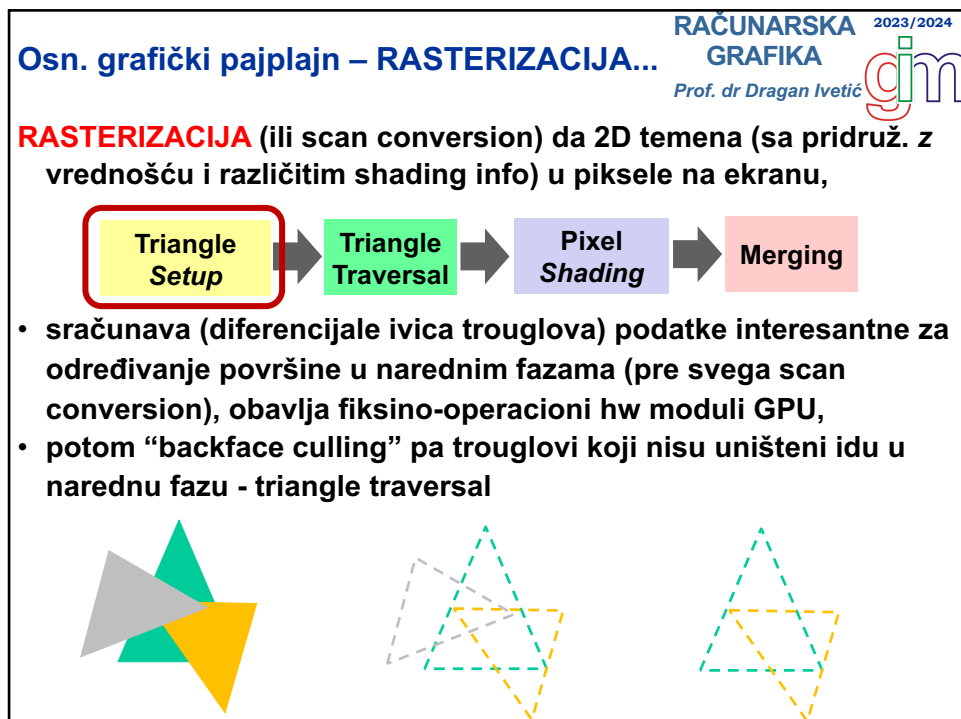
unit-cube Clipping new vertices new vertex Screen mapping  $(x_1, y_1)$   $(x_2, y_2)$

16





17



18

RAČUNARSKA 2023/2024  
GRAFIKA  
Prof. dr Dragan Ivetić

**... Osn. grafički pajplajn – RASTERIZACIJA...**

**RASTERIZACIJA**

- ili scan conversion - formira fragmente (set of samples/pixels) od kojih se sastoji površina trougla,
- održava sve ostale podatke o fragmentu iz geometrije (Z, boje...),

Example: Coverage sample point at pixel center  
 $(x+0.5, y+0.5)$   
Pixel  $(x,y)$

= triangle covers sample, fragment generated for pixel  
 = triangle does not cover sample, no fragment generated

19


RAČUNARSKA 2023/2024  
GRAFIKA  
Prof. dr Dragan Ivetić

**... Osn. grafički pajplajn – RASTERIZACIJA...**

**RASTERIZACIJA**

- sračunava konačnu boju fragmenta zavisno od izvora svetla, boje, materijala...
- izvršava (krajnje programabilna) jezgra GPU,
- komplikovano i zasnovano na mnoštvu ulaznih podataka,
- rezultat je samo vrednost boje,
- osnovne operacije su: texturing i per pixel lighting.

20

RAČUNARSKA GRAFIKA 2023/2024  
Prof. dr Dragan Ivetić 

## ... Osn. grafički pajplajn – RASTERIZACIJA

### RASTERIZACIJA

Triangle Setup

→

Triangle Traversal

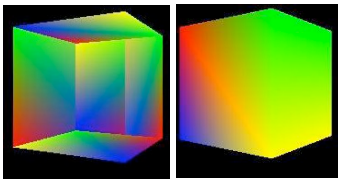
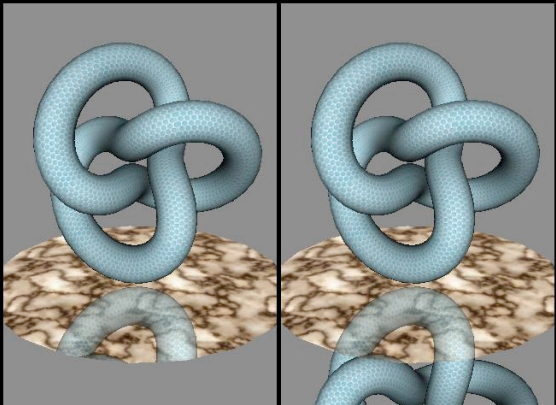
→

Pixel Shading


→

Merging


- konačno uređuje frejm bafer za prikaz kombinujući (blending) osnovni kolor bafer sa bojom fragmenta i Z baferom,

- ali i sa stencil baferom,
- i alpha baferom.



21

RAČUNARSKA GRAFIKA 2023/2024  
Prof. dr Dragan Ivetić 

**Models**

Selected model:

- Model 01: Sphere
- Model 02: Triangle
- Model 03: Cube
- Model 04: Plane with texture
- Model 05: All normals
- Model 06: Rotated normals
- Model 07: Shading
- Model 08: Texture

Selected shading type:

Flat

Smooth

Phong

**Material components**

Ambient:

- Ambient Red: 0.10
- Ambient Green: 0.10
- Ambient Blue: 0.10

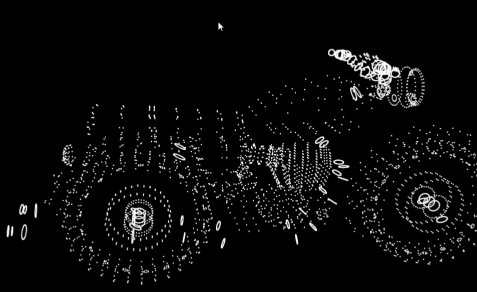
Diffuse:

- Diffuse Red: 0.10
- Diffuse Green: 0.10
- Diffuse Blue: 0.10

Specular:

- Specular Red: 0.10
- Specular Green: 0.10
- Specular Blue: 0.10

Shininess: 0.750 Strength



**Controls**

GUI toggle: G

Fullscreen: W A S D

Up/Down: Space/C

Camera: Mouse

Pause/Resume: F

Display mouse: E (hide)

Shading modes: 1 2 3 4 5 6 7 8

Selected shading type:

Flat: 1

Smooth: 2

Phong: 3

Jovan Srdanov RA 145/2019

[github.com/JovanSrdanov/OpenGL-Demo](https://github.com/JovanSrdanov/OpenGL-Demo)

22

RAČUNARSKA 2023/2024  
GRAFIKA  
Prof. dr Dragan Ivetić

## Programabilni grafički pajplajn

- OpenGL 3.x uvodi šejdere, kratke GPU programe pisane u GLSL koji daju slobodu programeru:
  - verteks**, obavlja transformaciju temena i normala, bojenje temena i rukovanje tekselima,
    - ostali poslovi fiksne funkcije (klipping, homogenizacija, viewport mapping, bekfejs kaling...)
  - tesalacioni**, opcioni, obično tesalacija konkavnih poligona na konveksne,
  - geomterijski**, opcioni, naprednije upravljanje topologijama: linijama (strips, loops), trouglovima (strips, fan) i quads (strips),
  - fragment/piksel**, sračunava svetlo i boju, teksturiše, fog,
    - ostali poslovi fiksne funkcije (Blending, Stencil test, Depth test, Scissor test).

Ed Angel University of New Mexico  
Dave Shreiner ARM, Inc.  
SIGGRAPH 2013

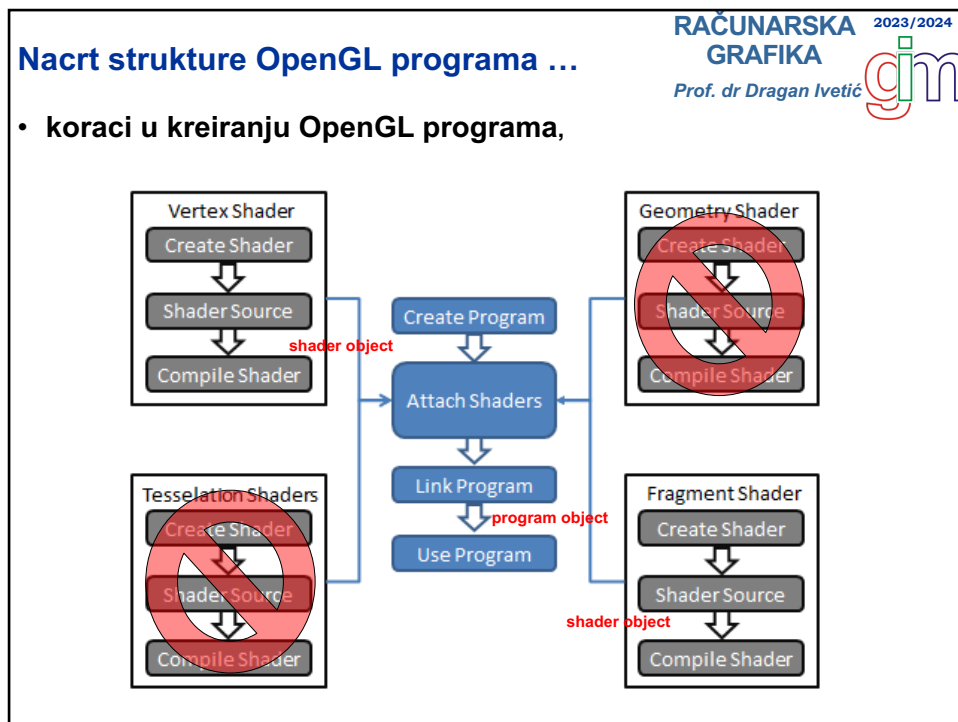
23

RAČUNARSKA 2023/2024  
GRAFIKA  
Prof. dr Dragan Ivetić

## OpenGL je konačni automat

- da ne bi svaki poziv OpenGL funkcije ima mnoštvo argumenata  
`glVertex3f( x, y, z, r, g, b, m00, ..., m33, s, t, nx, ny, nz, linewidth, ... );`
- stanje definisano skupom promenljivih (parametara) stanja
  - trenutna boja, veličina tačke i ostali atributi crtanja,
  - trenutne model&view transformacije,
  - trenutni model kamere i klipinga,
  - trenutni lighting&reflectance model,
  - trenutni viewport,...
- svaka promenljiva ima default vrednost koju zadržava dok je ne promenimo,
  - `glColor3f(1.0, 0.0, 0.0), glPointSize(2.5), gluOrtho2D(...),...`,
  - `glEnable(...), glDisable(...)`,
- renderovanje prikaza se obavlja shodno vrednostima promenljivih trenutnog stanja OpenGL.

24



25

**... Nacrt strukture OpenGL programa ...**

RAČUNARSKA GRAFIKA 2023/2024  
Prof. dr Dragan Ivetić

• izvršavanje:

- Vertex shader se izvršava za svaki verteks,
- Fragment shader se izvršava za svaki fragment,

• sadržaj:

- svaki shader mora imati main(),
- moguće klase promenljivih,
  - uniform – nema promena, read-only,
  - in – ulaz u shader, read-only,
  - out – izlaz (iz VS u FS, odnosno u frame bafer),

```

uniform mat4 projMatrix; // uniform input
in vec4 vertex; // attribut-input
out vec3 fragColor; // shader output
  
```

```

#version 330 core
layout(location = 0) in vec4 position;


void main()
{
    gl_Position = position;
}
  
```

```

#version 330 core
layout(location = 0) out vec4 color;

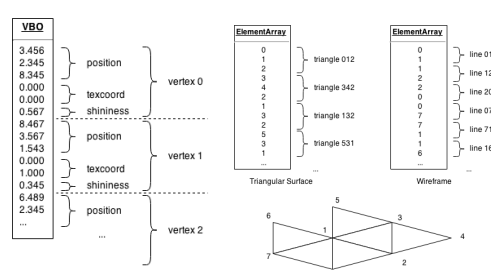
void main()
{
    color = vec4(1.0, 0.0, 0.0, 1.0);
}
  
```

26

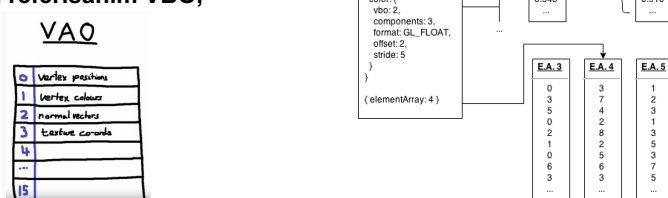
RAČUNARSKA 2023/2024  
GRAFIKA  
Prof. dr Dragan Ivetić 

### ... Nacrt strukture OpenGL programa ...

- vertex buffer objects (VBO):**
  - prenos podataka o verteksima u blokovima direktno u video memoriju,




- vertex array objects (VAO):**
  - niz do maksimalno 16 atributa o VBOs,
  - opisuje stanje meš mreže objekta opisanih u referisanim VBO,



**VAO**

0	Vertex positions
1	Vertex colors
2	Normal vectors
3	Texture coords
4	
...	
15	

27

RAČUNARSKA 2023/2024  
GRAFIKA  
Prof. dr Dragan Ivetić 

### ... Nacrt strukture OpenGL programa

- opšti template OpenGL C programa**
  - #include datoteka (paziti na redosled),
  - deklaracije podataka i prototipova funkcija (barem za kreiranje i linkovanje šejdera),
  - main (),
    - pokretanje GLFW za otvaranje prozora i hendlovanje događaja,
    - pokretanje GLEW biblioteke,
    - kreiranje buffer objekata i punjenje podacima,
    - kreiranje shader objekata i program objekta pomoću CreateShader,
    - kreiranje program objekta,
    - while (nije zatvoren prozor) {obriši prozor, glDrawArrays(...), glfwSwapBuffers(...)}
    - oslobodi program objekat
  - funkcija CompileShader ()
    - otvara i prevodi shader kodne datoteke,
  - funkcija CreateShader ()
    - formira objekte programa i šejdera, poziva kompajliranje pa ih potom linkuje u jedan program objekat.
  - ostale funkcije ...
- slično i C++, sem što su šejderi definisani globalno kao konstantni string.

28