



OpenGL Shading Language

Igor Rosandić



Sadržaj

- ❖ OpenGLSL Uvod
- ❖ Mogućnosti GLSL-a
- ❖ Pipeline
- ❖ Vertex processor
- ❖ Fragment processor
- ❖ Prikaz programa
 - ◆ Demo i projekt
- ❖ Zaključak



OpenGLSL Uvod

- ❖ High-level grafički jezik
- ❖ Postao sastavni dio OpenGL 2.0
- ❖ Nagli razvoj OpenGL-a i GLSL-a
- ❖ Zajmena područja ustaljenih funkcija s programibilnima
- ❖ Dva osnovna područja:
 - ◆ Vertex processing
 - ◆ Fragment processing



OpenGLSL Uvod

- ❖ Shader – GLSL kod za procesuiranje
 - ◆ Vertex shader
 - ◆ Fragment shader
- ❖ Program – mehanizam za kompajliranje i povezivanje shadera u izvršni kod
- ❖ Baziran na C i C++ sintaksi



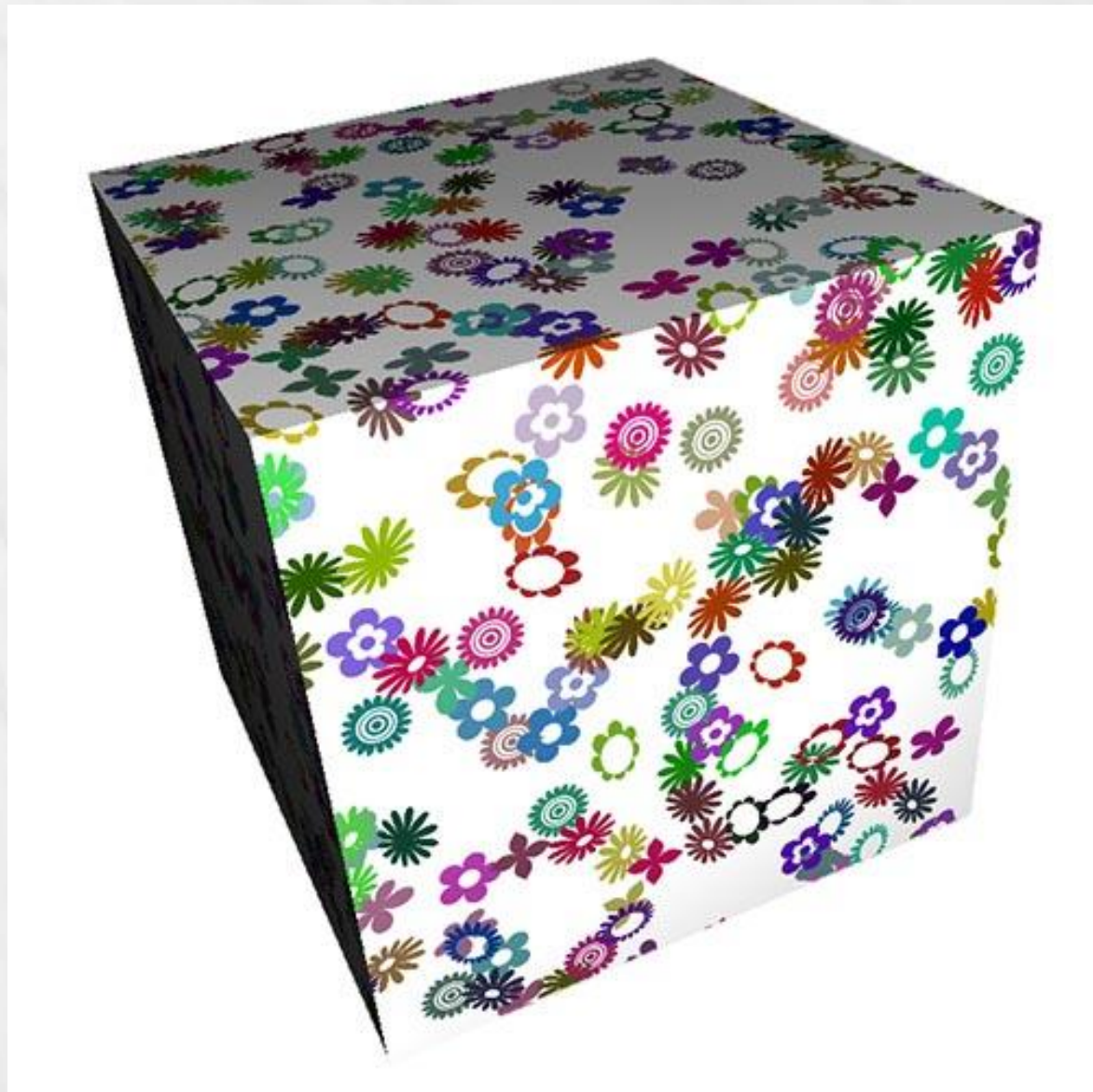
Mogućnosti GLSL-a

- ❖ Povećanje realističnosti materijala i efekata
 - ◆ Voda, vatra, obalici, dim, metali, drvo ...
- ❖ Unapređeni efekti rendiranja
- ❖ Novi način korištenja tekstura
- ❖ Dinamičko generiranje tekstura
- ❖ Procesuiranje slika
- ❖ Animacijski efekti

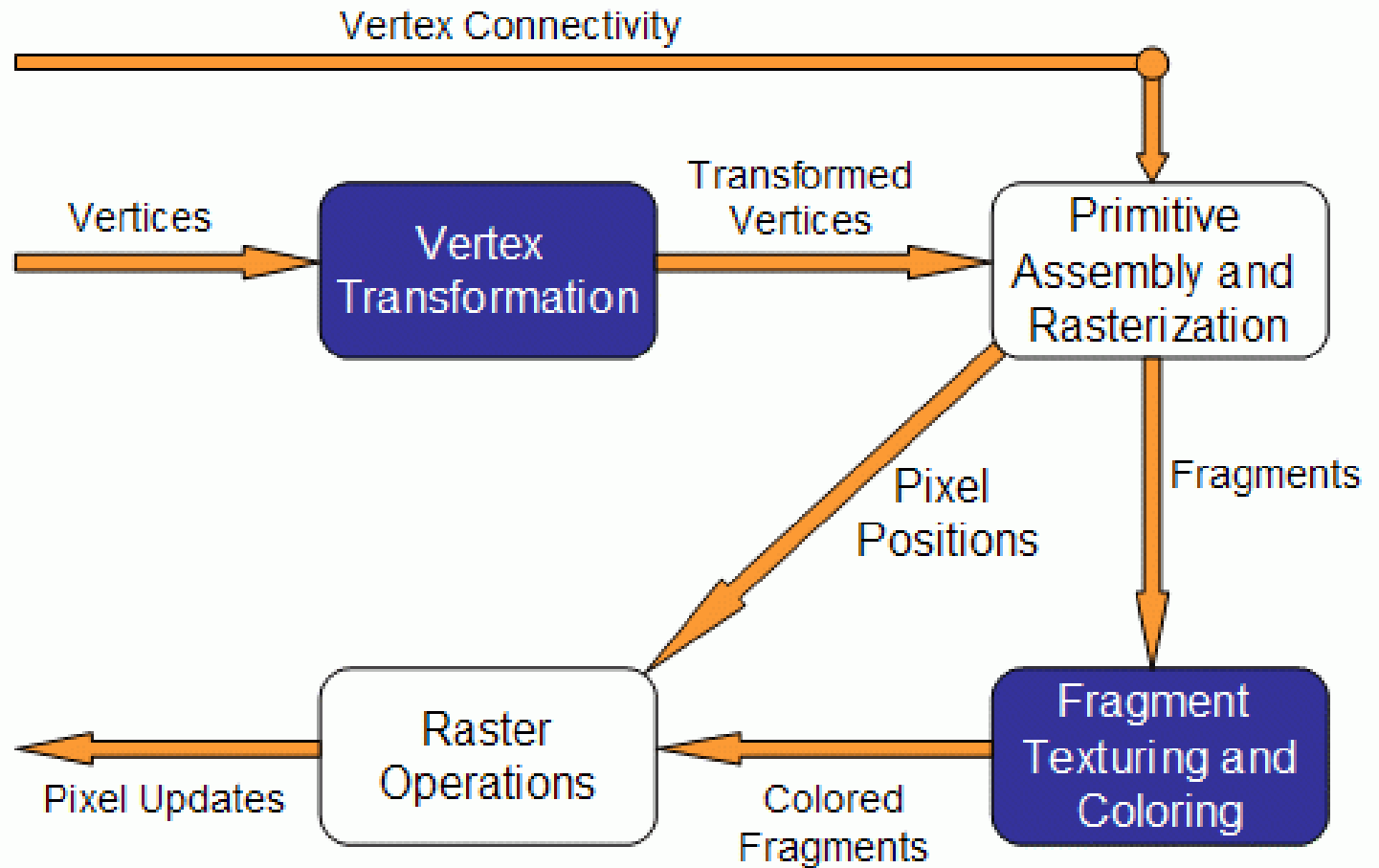




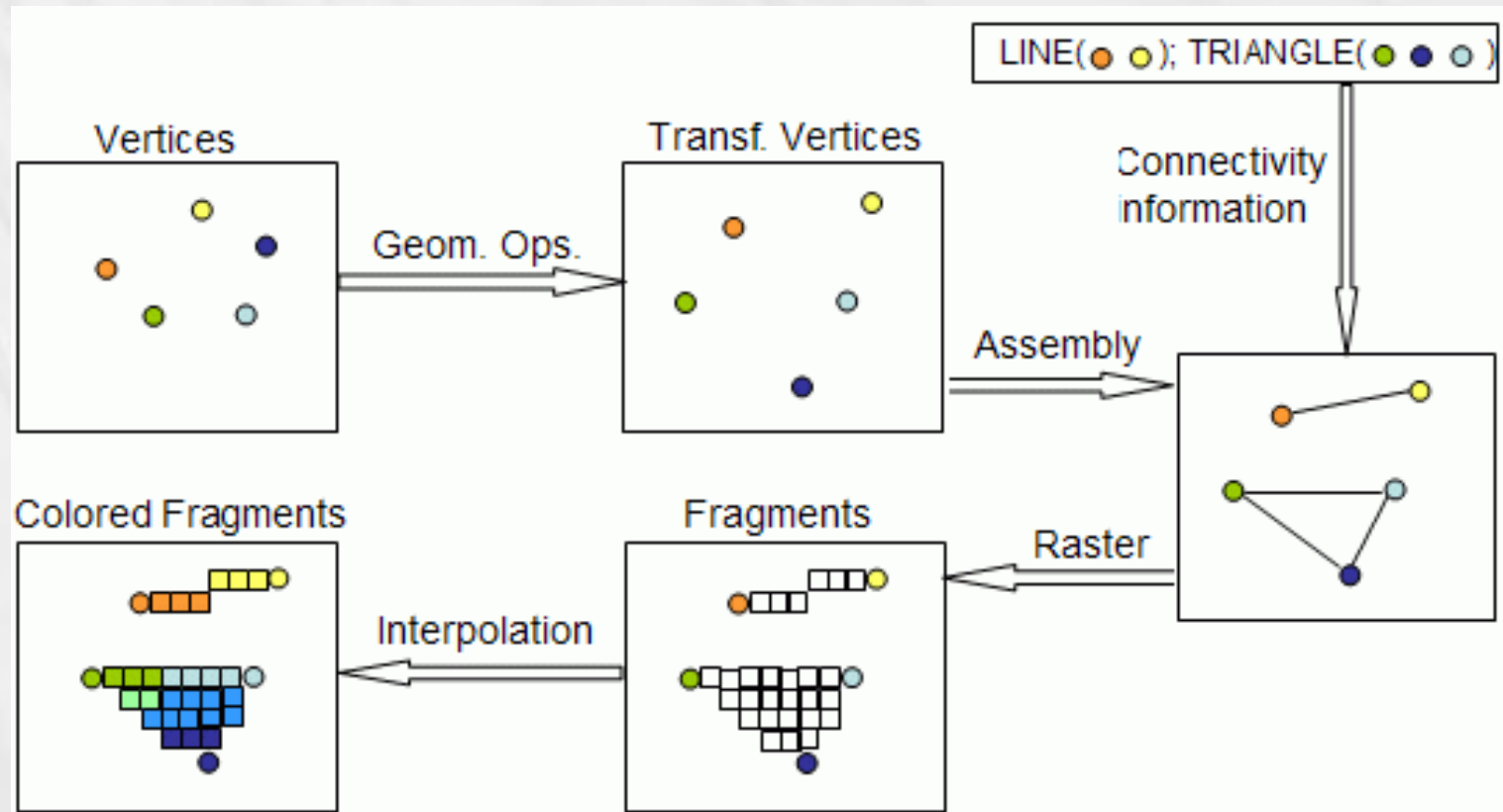




OpenGLSL pipeline



Fiksna funkcionalnost GLa



Vertex processor

- ❖ Programibilna jedinica čija svrha jest djelovanje nad vrijednostima vrhova te njihovim povezanim podacima.
- ❖ Grafičke operacije:
 - ◆ Transformacija vrhova
 - ◆ Normalna transformacija i normalizacija
 - ◆ Generiranje koordinata teksture
 - ◆ Transformacija koordinata teksture
 - ◆ Osvijetljenje
 - ◆ Boja

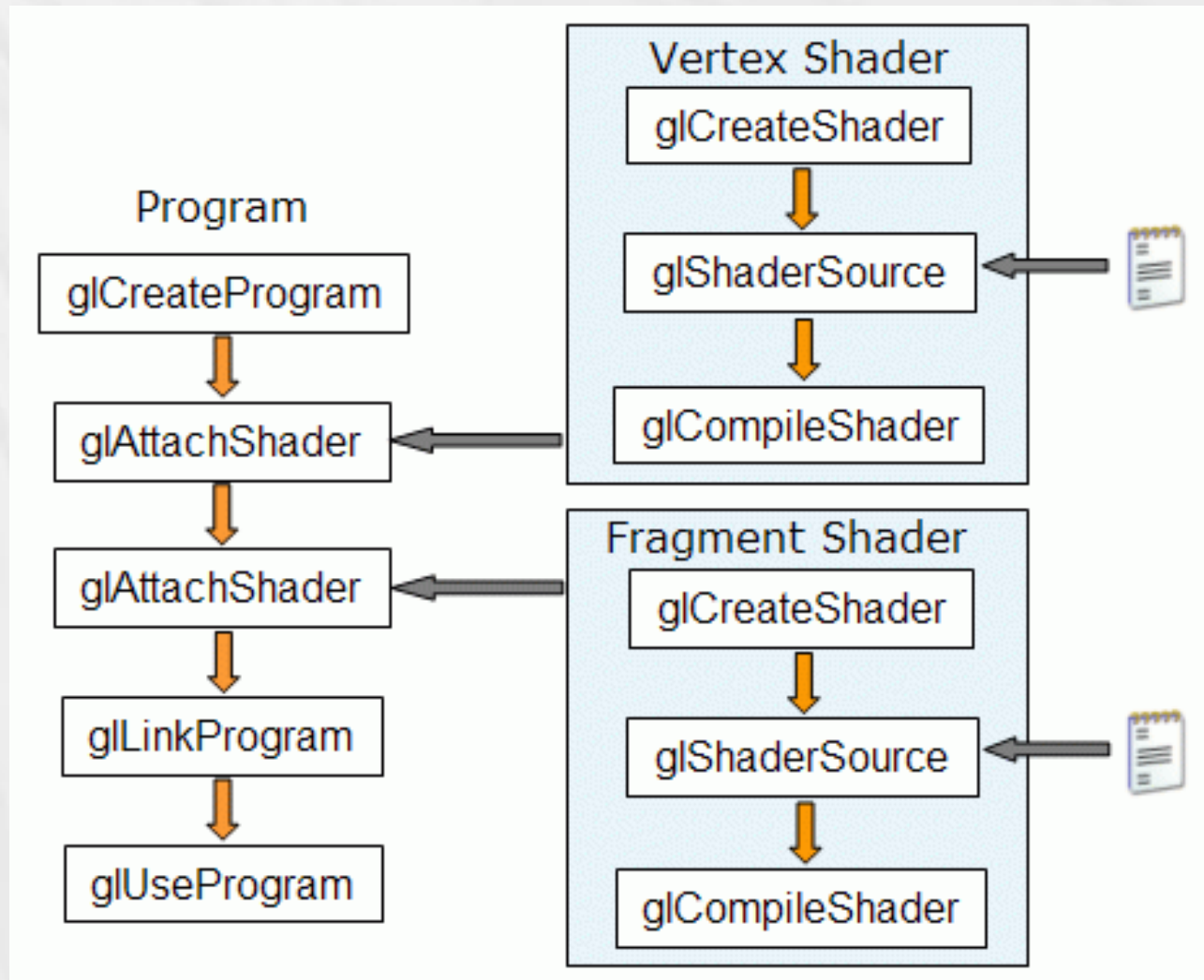


Fragment Processor

- ❖ Programibilna jedinica čija svrha jest djelovanje nad vrijednostima fragmenata i njihovima povezanima podacima.
- ❖ Grafičke operacije:
 - ◆ Operacije nad interpoliranim vrijednostima
 - ◆ Pristup teksturama
 - ◆ Magla
 - ◆ Zbir boja
 - ◆ Alpha testiranje



Prikaz korištenja programa





Primjer programa

```
void setShaders() {  
    char *vs,*fs;  
  
    v = glCreateShader(GL_VERTEX_SHADER);  
    f = glCreateShader(GL_FRAGMENT_SHADER);  
    vs = textFileRead("toon.vert"); fs = textFileRead("toon.frag");  
  
    const char * vv = vs; const char * ff = fs;  
  
    glShaderSource(v, 1, &vv,NULL);  
    glShaderSource(f, 1, &ff,NULL);  
  
    free(vs); free(fs);  
    glCompileShader(v); glCompileShader(f);  
    p = glCreateProgram();  
  
    glAttachShader(p,v); glAttachShader(p,f);  
  
    glLinkProgram(p);  
    glUseProgram(p);  
}
```

Zaključak

- ❖ High-level grafički jezik
- ❖ Razvija se naglom brzinom
- ❖ Očekuje se sve veći broj mogućnosti te daljnji “brzi” razvoj OpenGL-a kao i Shading Language-a

