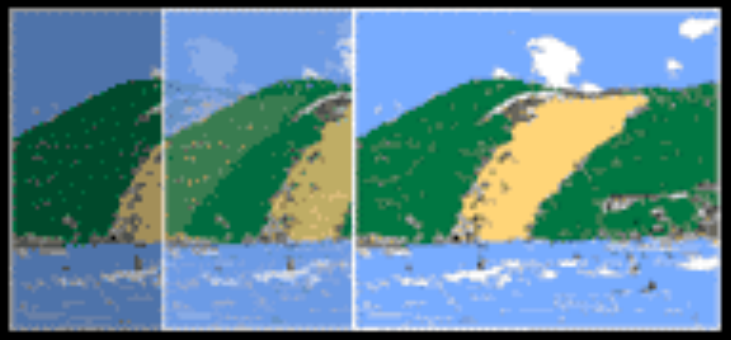
GPU-Based Volume Rendering of Unstructured Grids

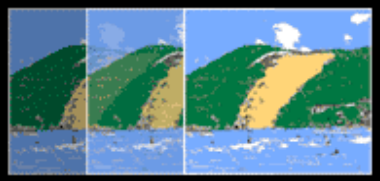
João L. D. Comba — UFRGS
Cláudio T. Silva — University of Utah
Steven P. Callahan — University of Utah
Fábio F. Bernardon — UFRGS



SIBGRAPI 2005

Natal - RN - Brazil

XVIII Brazilian Symposium on Computer Graphics and Image Processing



SIBGRAPI 2005

GPU-Based Volume Rendering of Unstructured Grids

Module 1: Graphics Hardware (GPUs)

Module 2: Projected Tetrahedra Techniques

Module 3: Isosurface Techniques

break

Module 4: Hw Assisted Visibility Sorting (HAVS)

Module 5: HW Ray-Casting

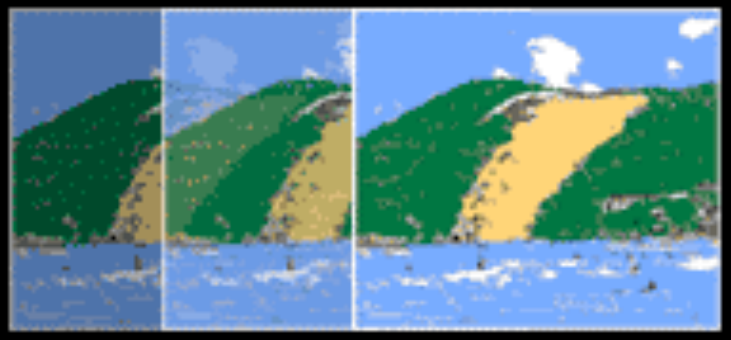
Module 6: Conclusion

GPU-Based Volume Rendering of Unstructured Grids

Module 1:
Graphics Hardware

João L. D. Comba

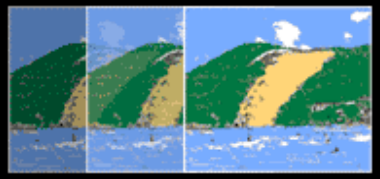
UFRGS



SIBGRAPI 2005

Natal - RN - Brazil

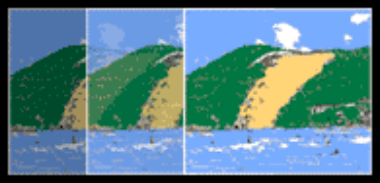
XVIII Brazilian Symposium on Computer Graphics and Image Processing



SIBGRAPI 2005

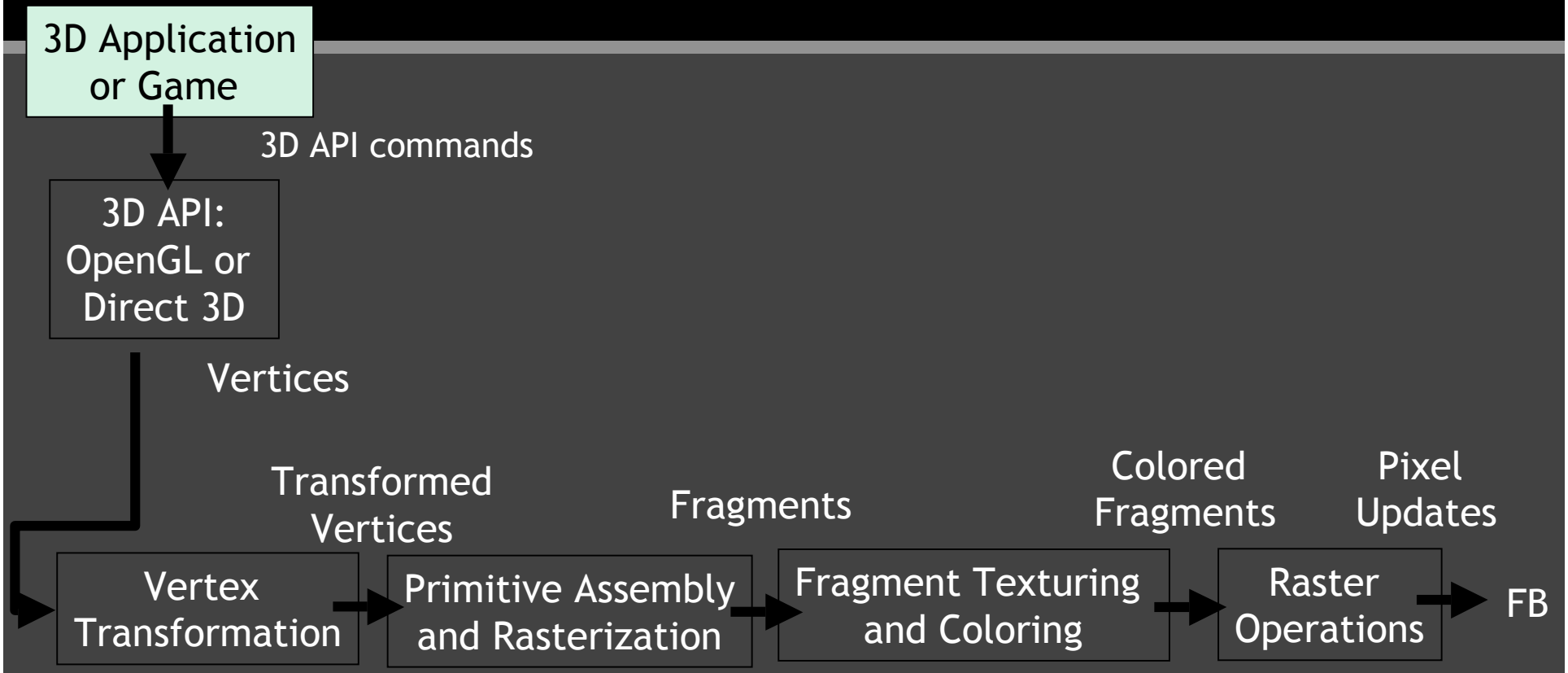
Pre-GPU Graphics Acceleration

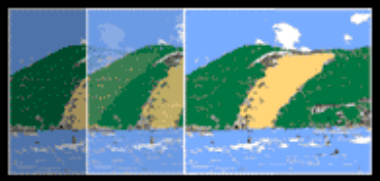
- Integrated Graphics Architecture
- Silicon Graphics
- Evans & Sutherland



SIBGRAPI 2005

Graphics Pipeline





SIBGRAPI 2005

Graphics Pipeline

3D Application or Game

3D API commands

```
glBegin(GL_TRIANGLES);
glVertex3f(0.0,0.0,0.0);
glVertex3f(1.0,0.0,0.0);
glVertex3f(0.5,1.0,0.0);
...
glEnd();
```

3D API: OpenGL or Direct 3D

Vertices

Vertex Transformation

Transformed Vertices

Primitive Assembly and Rasterization

Fragments

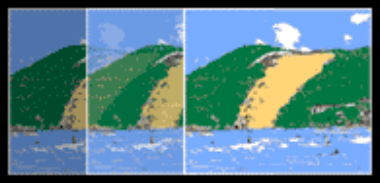
Fragment Texturing and Coloring

Colored Fragments

Raster Operations

Pixel Updates

FB



SIBGRAPI 2005

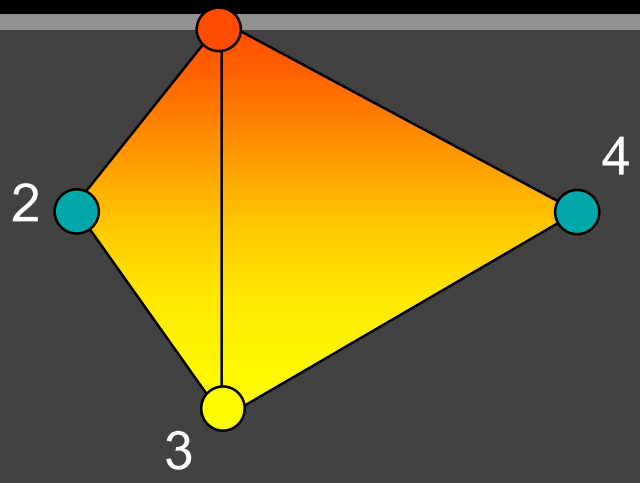
Graphics Pipeline

3D Application or Game

3D API commands

3D API: OpenGL or Direct 3D

Vertices



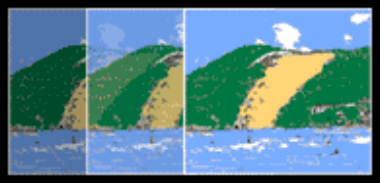
Transformed Vertices

Fragments

Colored Fragments

Pixel Updates

Vertex Transformation → Primitive Assembly and Rasterization → Fragment Texturing and Coloring → Raster Operations → FB



SIBGRAPI 2005

Graphics Pipeline

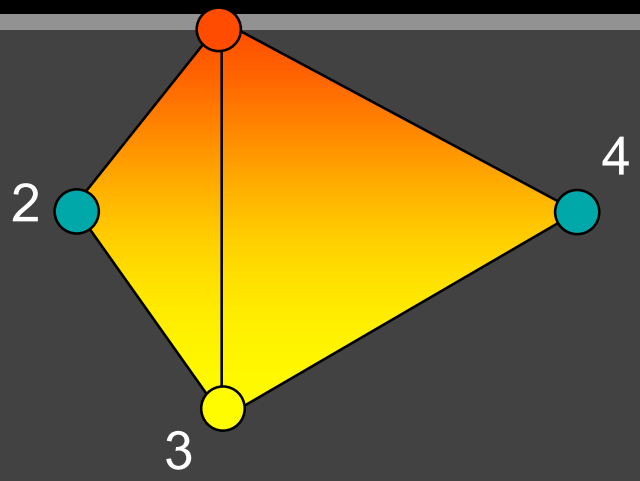
3D Application or Game

3D API commands

3D API: OpenGL or Direct 3D

4 3 1 Vertices

1 2 3



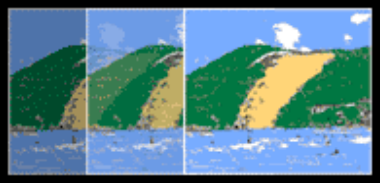
Transformed Vertices

Fragments

Colored Fragments

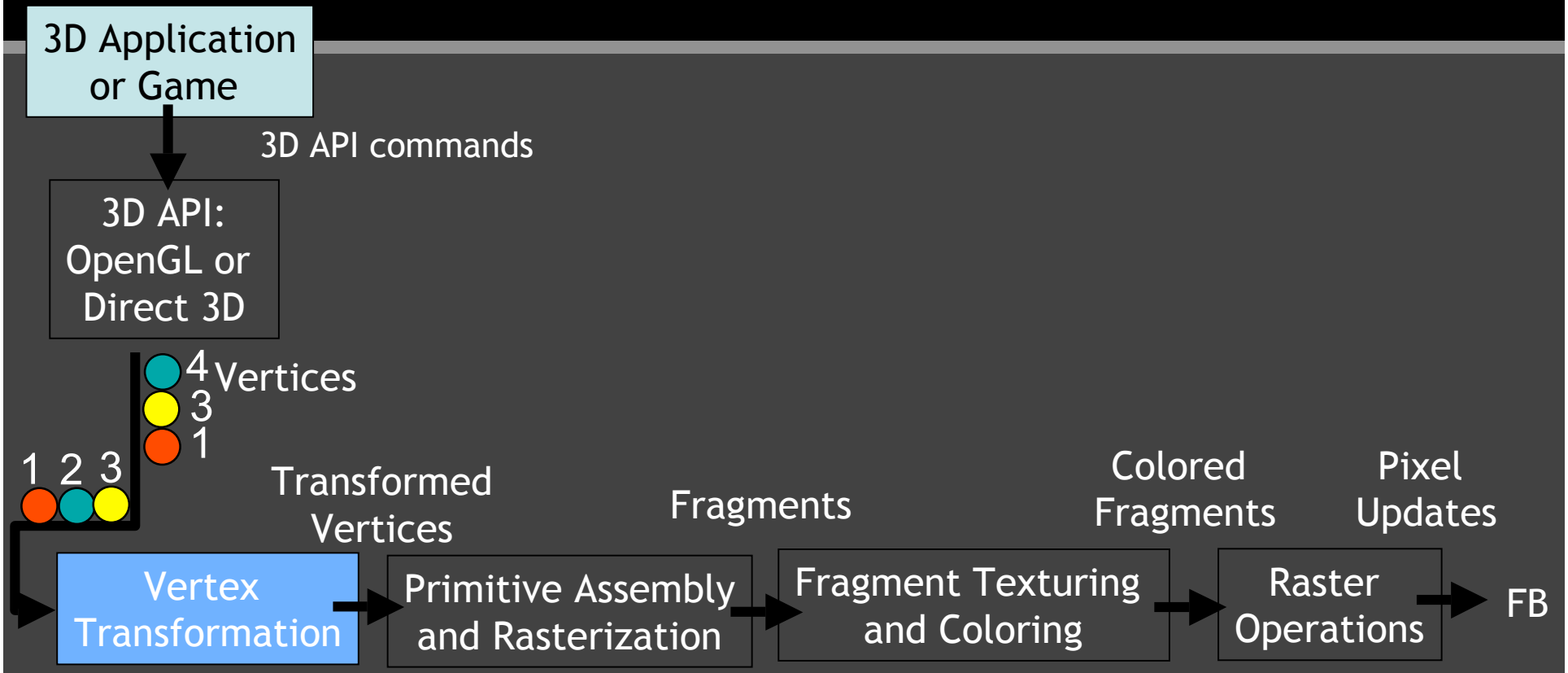
Pixel Updates

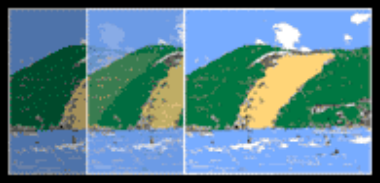
Vertex Transformation → Primitive Assembly and Rasterization → Fragment Texturing and Coloring → Raster Operations → FB



SIBGRAPI 2005

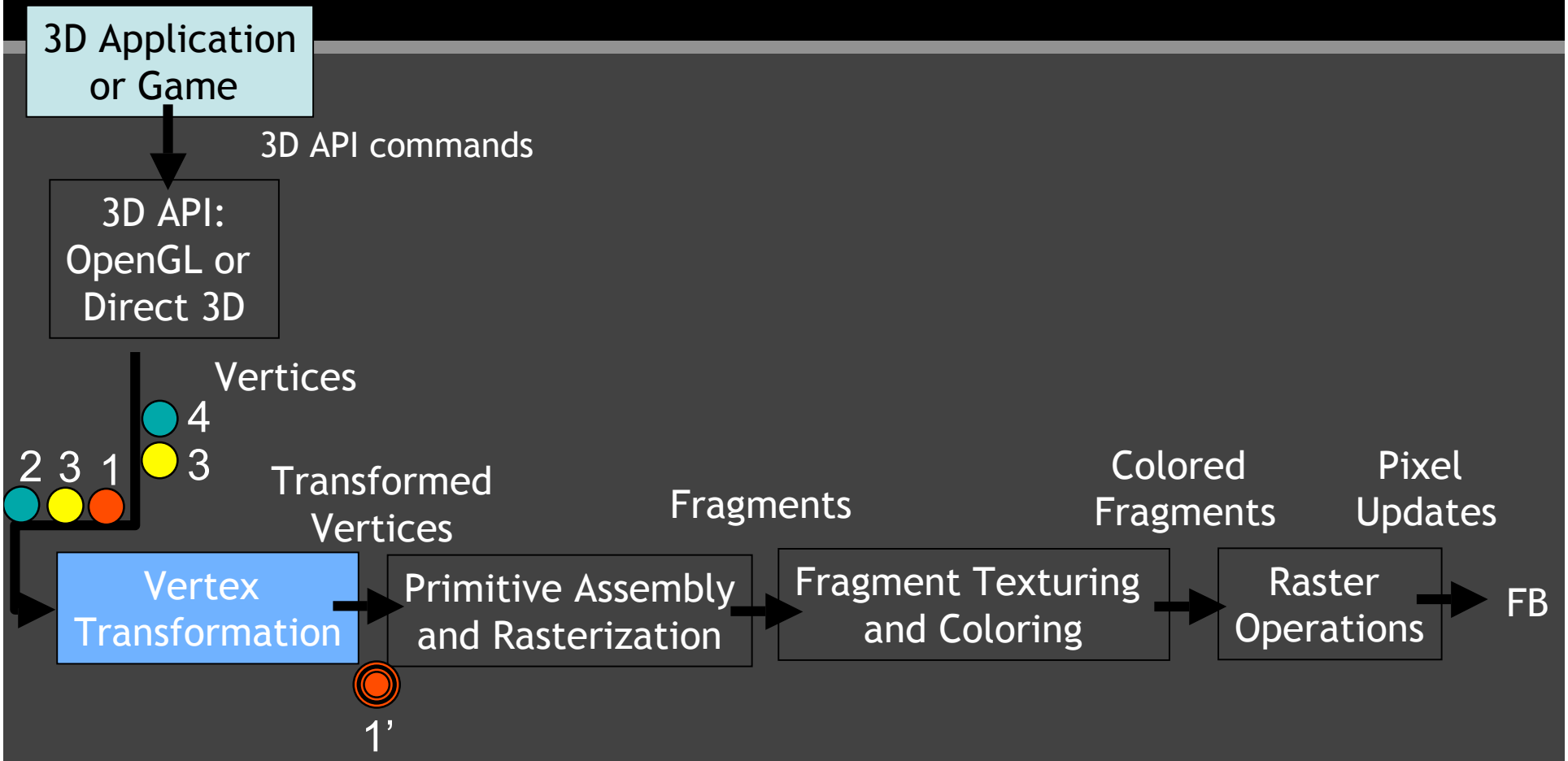
Graphics Pipeline

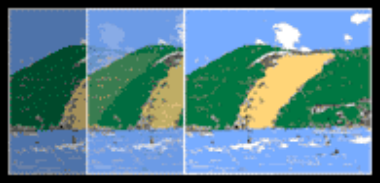




SIBGRAPI 2005

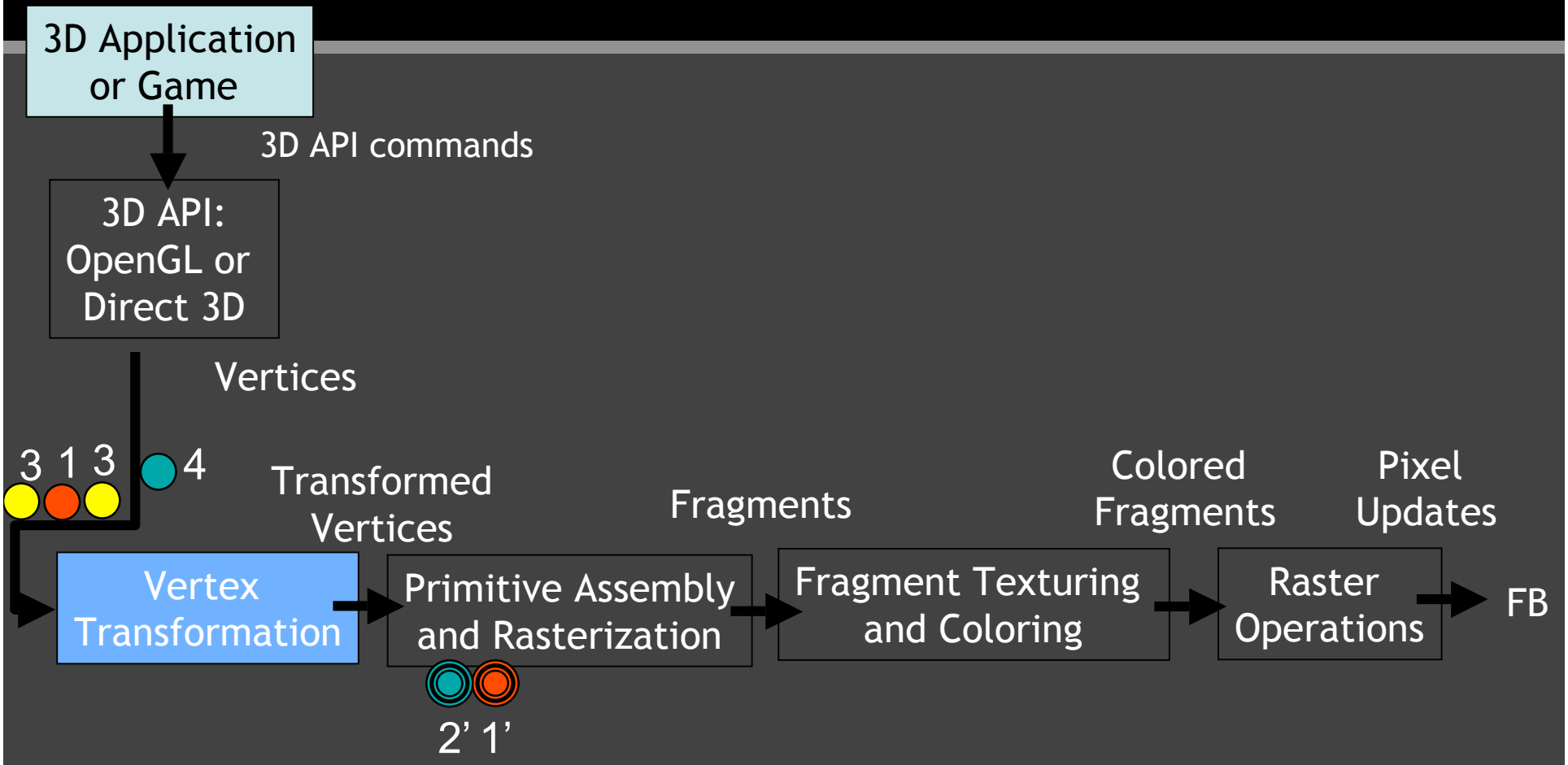
Graphics Pipeline

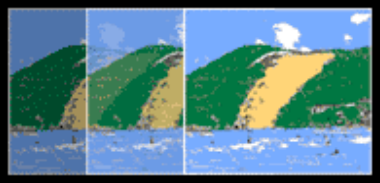




SIBGRAPI 2005

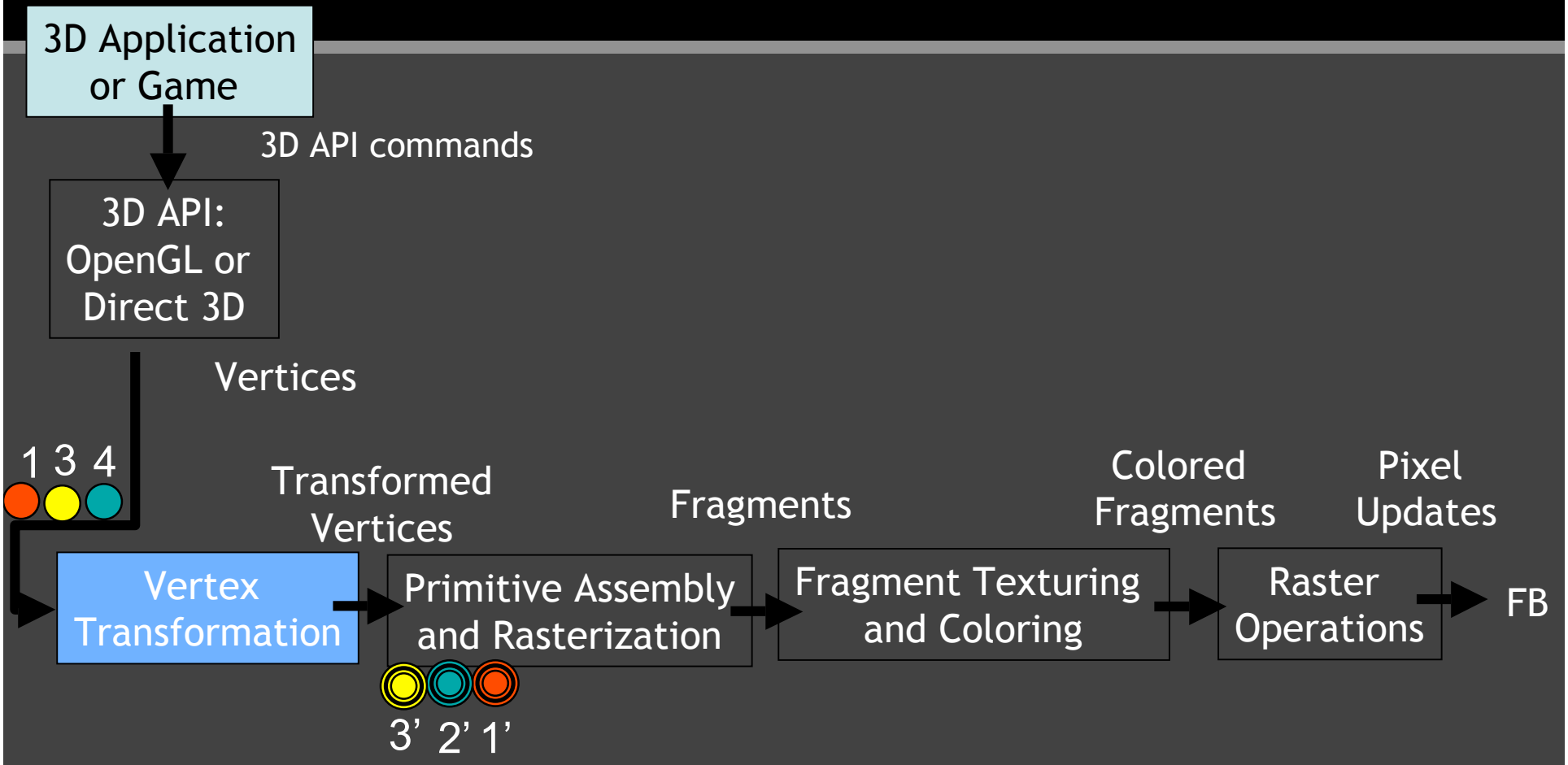
Graphics Pipeline

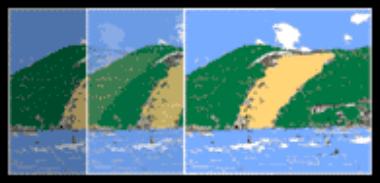




SIBGRAPI 2005

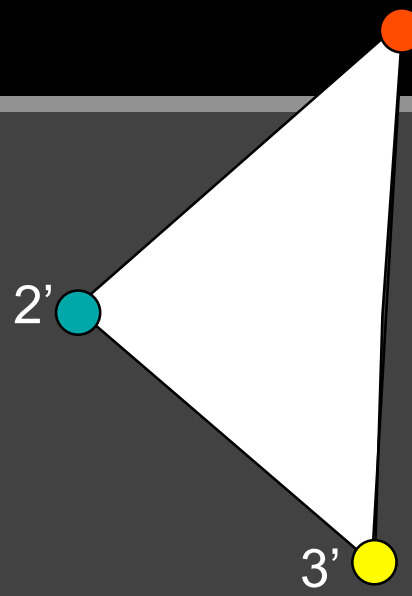
Graphics Pipeline





SIBGRAPI 2005

Graphics Pipeline



3D Application or Game

3D API commands

3D API: OpenGL or Direct 3D

Vertices

1 3 4

Transformed Vertices

Vertex Transformation

Primitive Assembly and Rasterization

Fragments

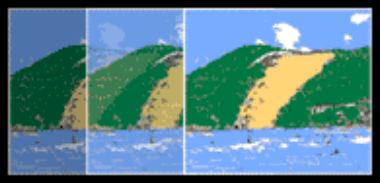
Fragment Texturing and Coloring

Colored Fragments

Raster Operations

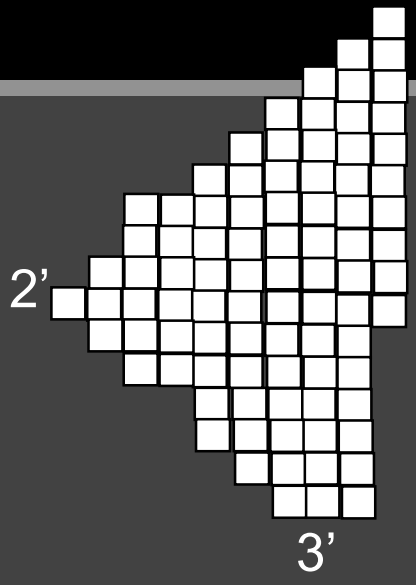
Pixel Updates

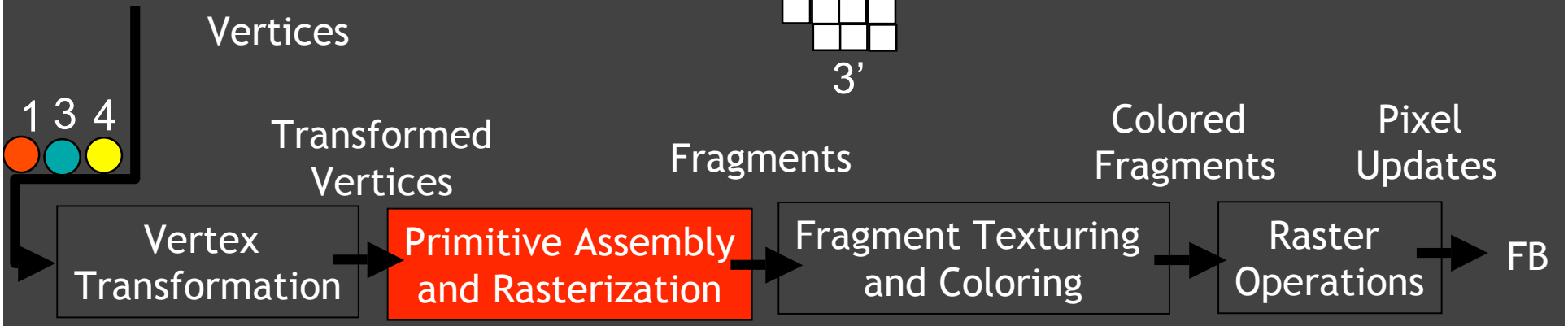
FB

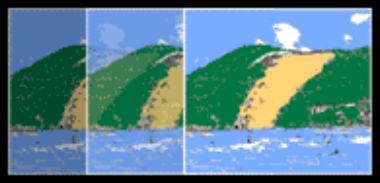


SIBGRAPI 2005

Graphics Pipeline

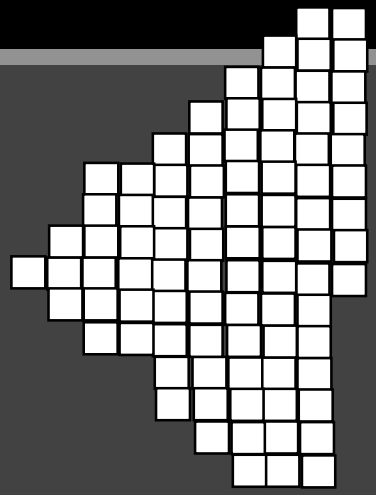


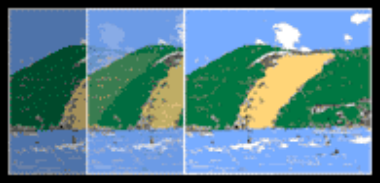




SIBGRAPI 2005

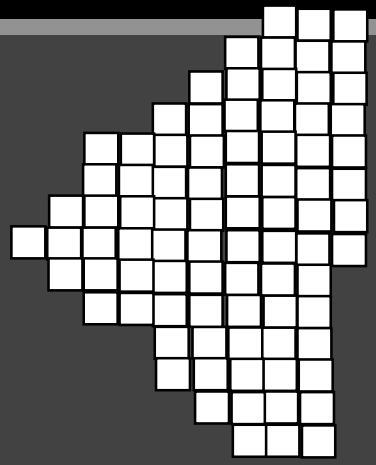
Graphics Pipeline

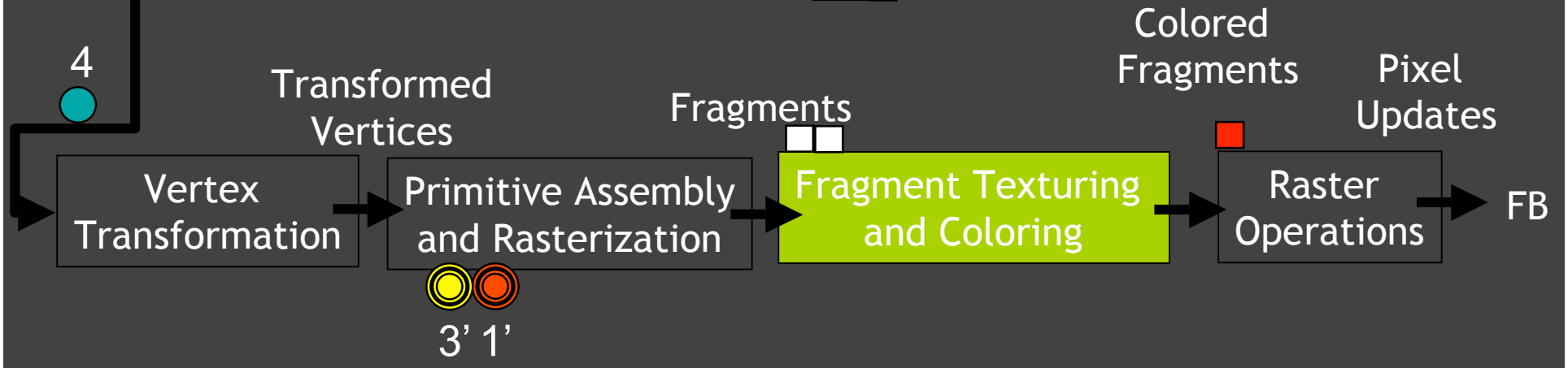


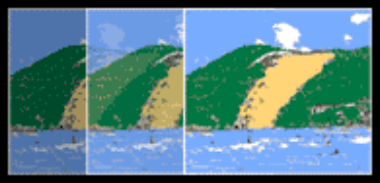


SIBGRAPI 2005

Graphics Pipeline







SIBGRAPI 2005

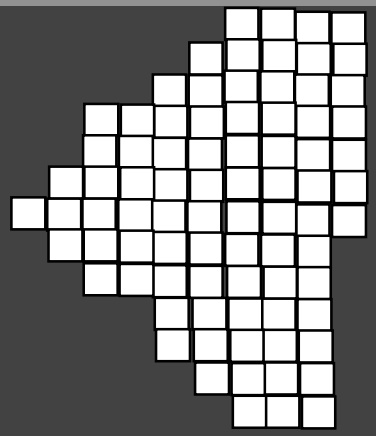
Graphics Pipeline

3D Application or Game

3D API commands

3D API: OpenGL or Direct 3D

Vertices



Transformed Vertices

Fragments

Colored Fragments

Pixel Updates

Vertex Transformation

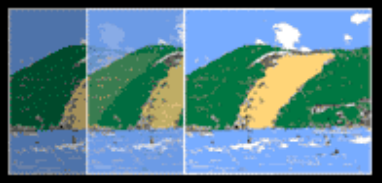
Primitive Assembly and Rasterization

Fragment Texturing and Coloring

Raster Operations

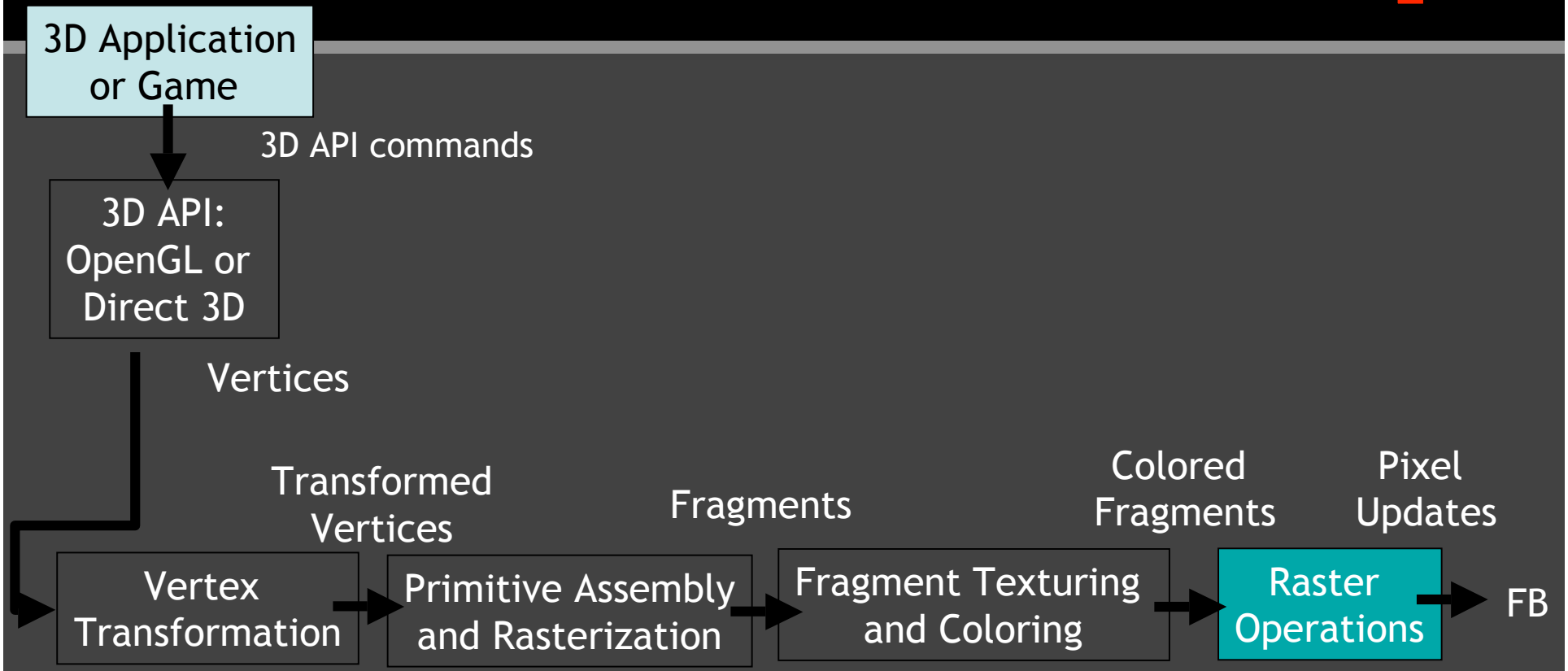
FB

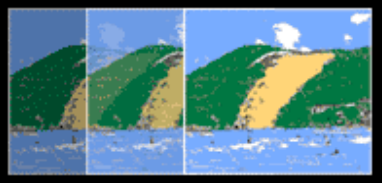
4’ 3’ 1’



SIBGRAPI 2005

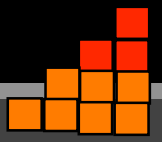
Graphics Pipeline

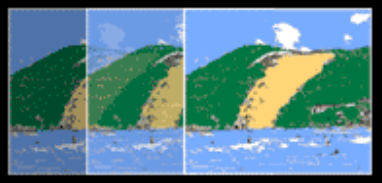




SIBGRAPI 2005

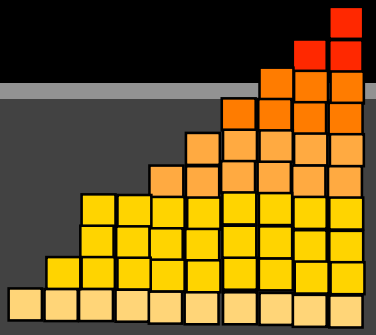
Graphics Pipeline





SIBGRAPI 2005

Graphics Pipeline



3D Application or Game

3D API commands

3D API: OpenGL or Direct 3D

Vertices

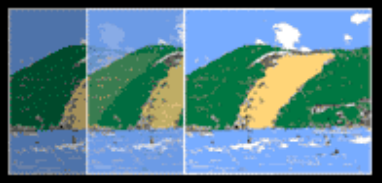
Transformed Vertices

Fragments

Colored Fragments

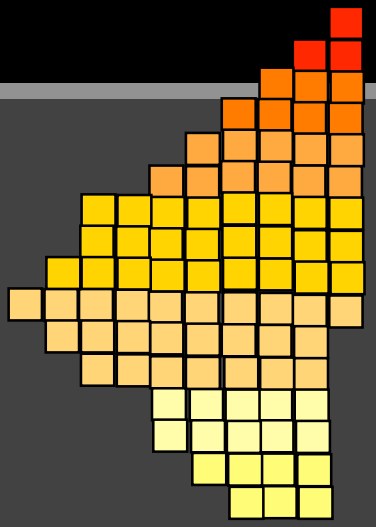
Pixel Updates

Vertex Transformation → Primitive Assembly and Rasterization → Fragment Texturing and Coloring → Raster Operations → FB



SIBGRAPI 2005

Graphics Pipeline



3D Application or Game

3D API commands

3D API: OpenGL or Direct 3D

Vertices

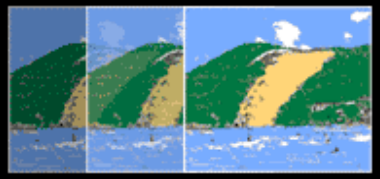
Transformed Vertices

Fragments

Colored Fragments

Pixel Updates

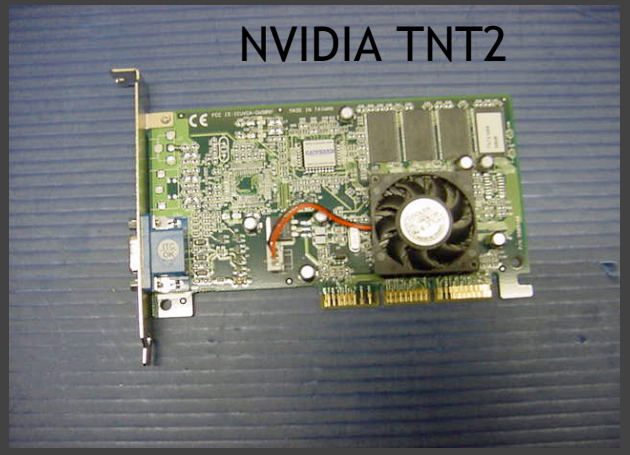
Vertex Transformation → Primitive Assembly and Rasterization → Fragment Texturing and Coloring → Raster Operations → FB

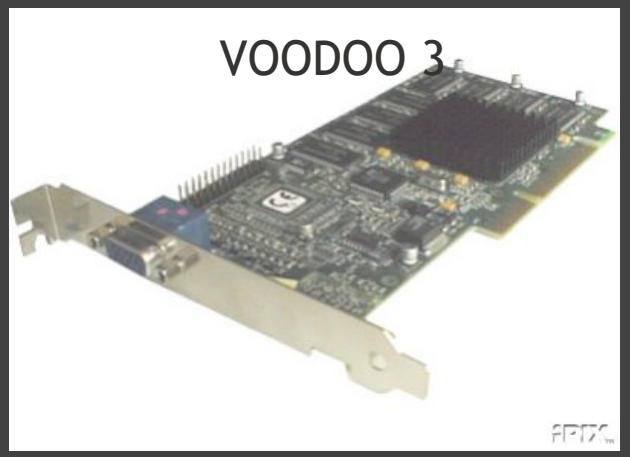


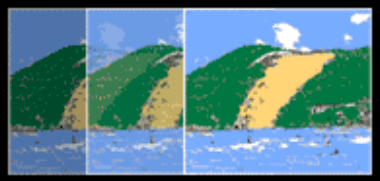
SIBGRAPI 2005

First-Generation GPUS (up to 1998)

- NVIDIA TNT2, ATI Rage, 3dfx Voodoo3
- Relevant tasks:
  - Rasterizing pre-transformed triangles
  - Applying one or two textures
  - Implement DirectX 6 feature set

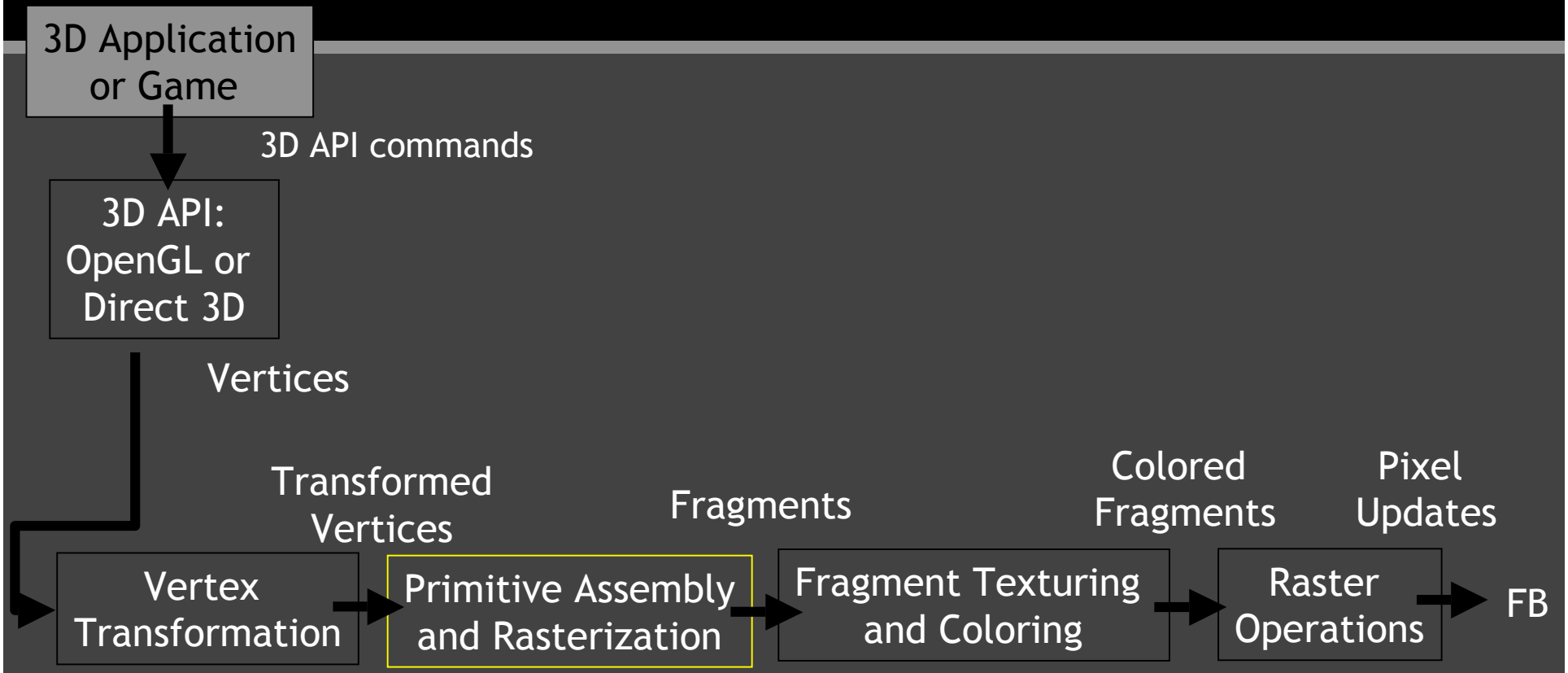


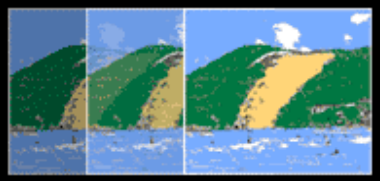




SIBGRAPI 2005

GPUs



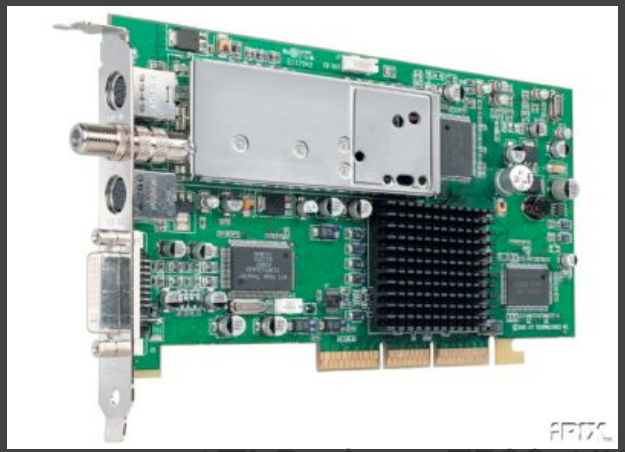


SIBGRAPI 2005

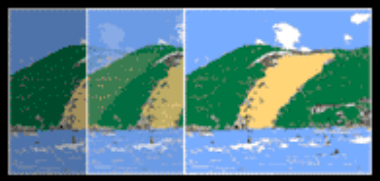
Second-Generation GPUS (1999-2000)

- NVIDIA GeForce 256, GeForce2, ATI Radeon 7500, S3 Savage 3D
- Relevant tasks:
  - Transformation & Lighting (T&L) em Hardware
  - Implement DirectX 7 feature set
  - Cube map textures
  - More math operations for combining textures (still limited)

GeForce2 GTS

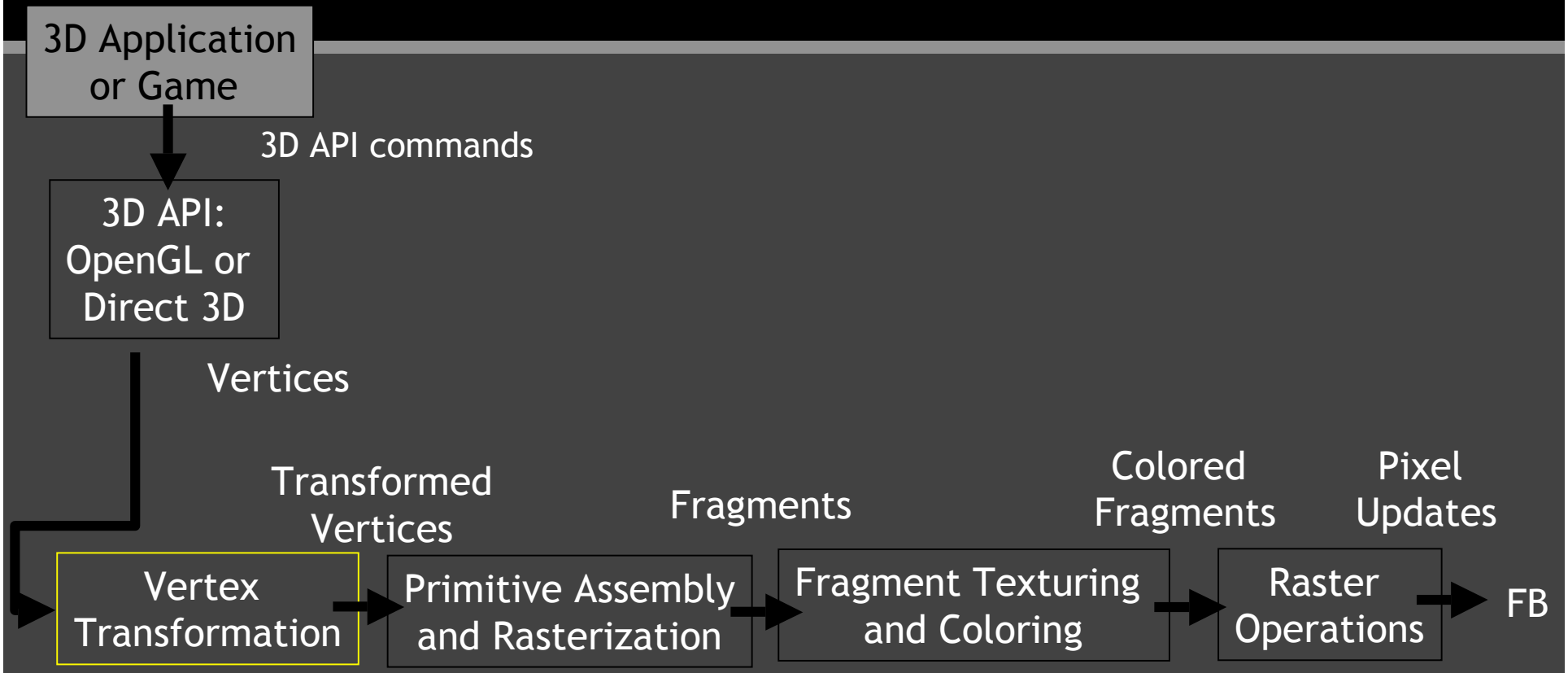


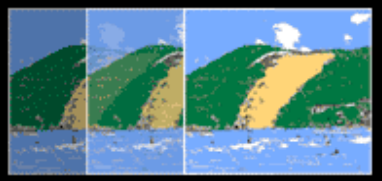
ATI Radeon 7500 AIW



SIBGRAPI 2005

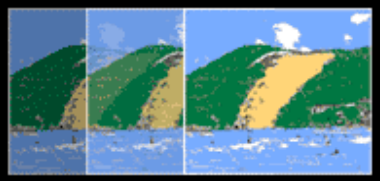
GPUs





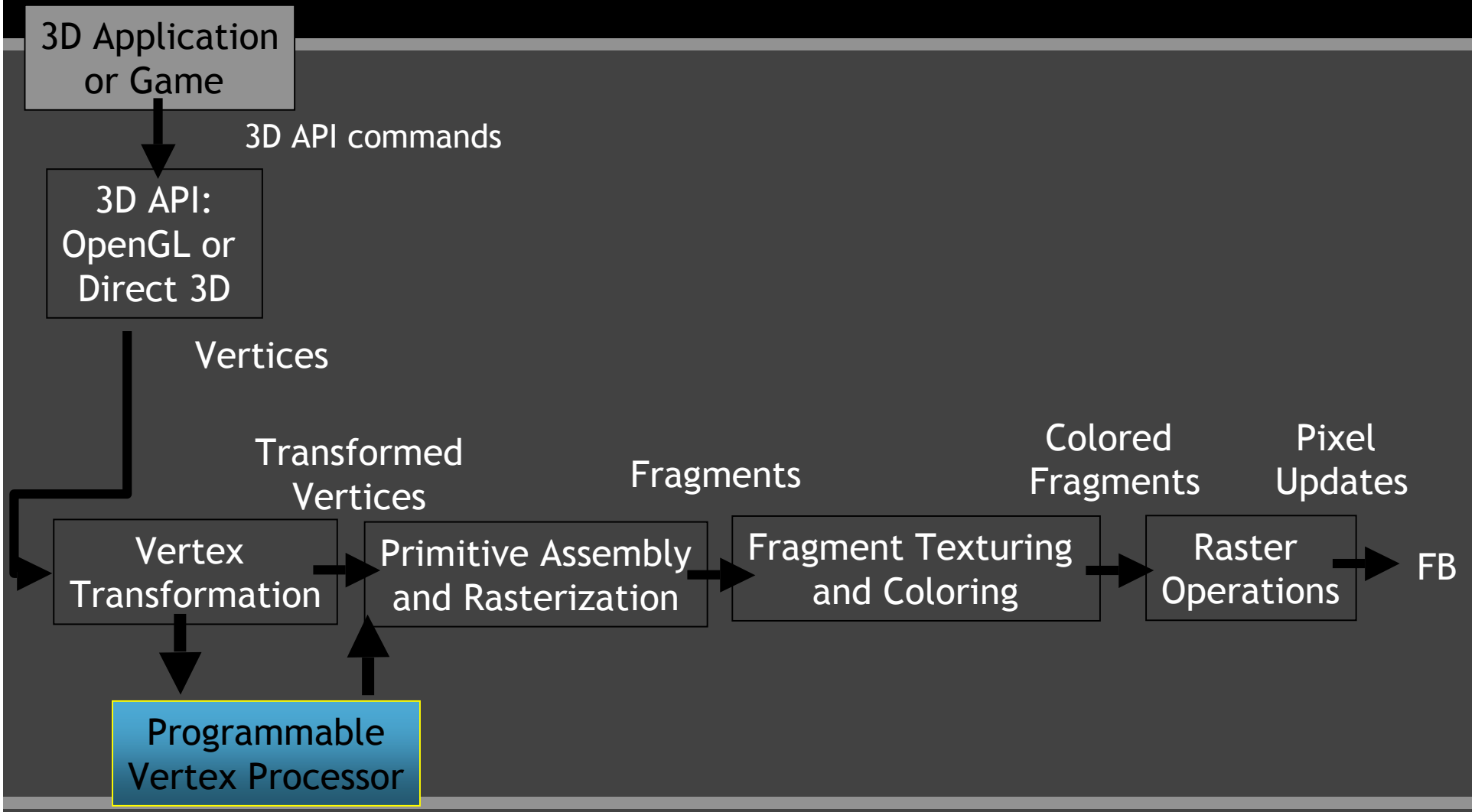
Third-Generation GPUS (2001)

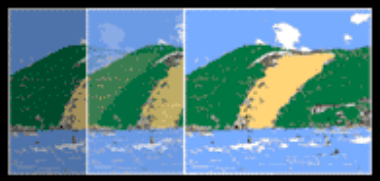
- NVIDIA GeForce 3, GeForce4 Ti, Microsoft XBox, ATI Radeon 8500, Quadro 4
- Relevant tasks:
  - Vertex programmability rather than more configurability
  - More pixel-level (fragment) configurability (not truly programmable)
  - 3D Textures
  - Shadow Maps



SIBGRAPI 2005

GPUs

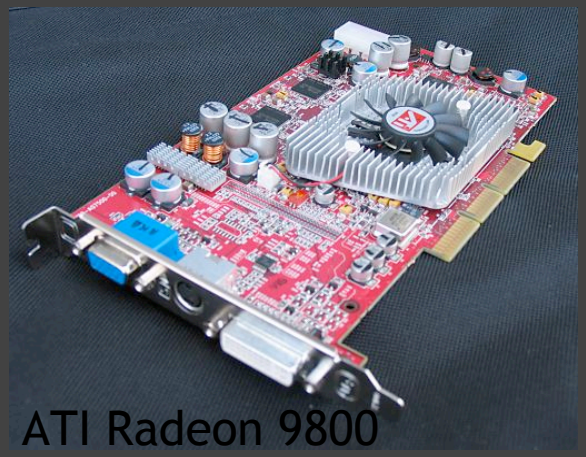




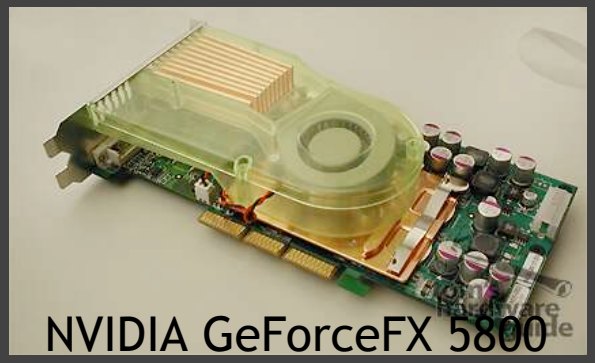
SIBGRAPI 2005

Fourth-Generation GPUS (2002-2003)

- NVIDIA GeForce FX family, ATI 9700, ATI 9800
- Relevant tasks:
  - Vertex and Fragment programmability
  - Implement DirectX 9 feature set
  - 32 Bit IEEE Floating Point per component (128-bit textures)



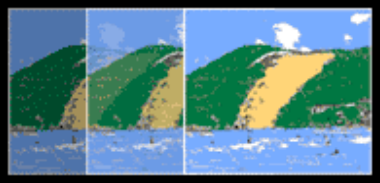
ATI Radeon 9800



NVIDIA GeForceFX 5800

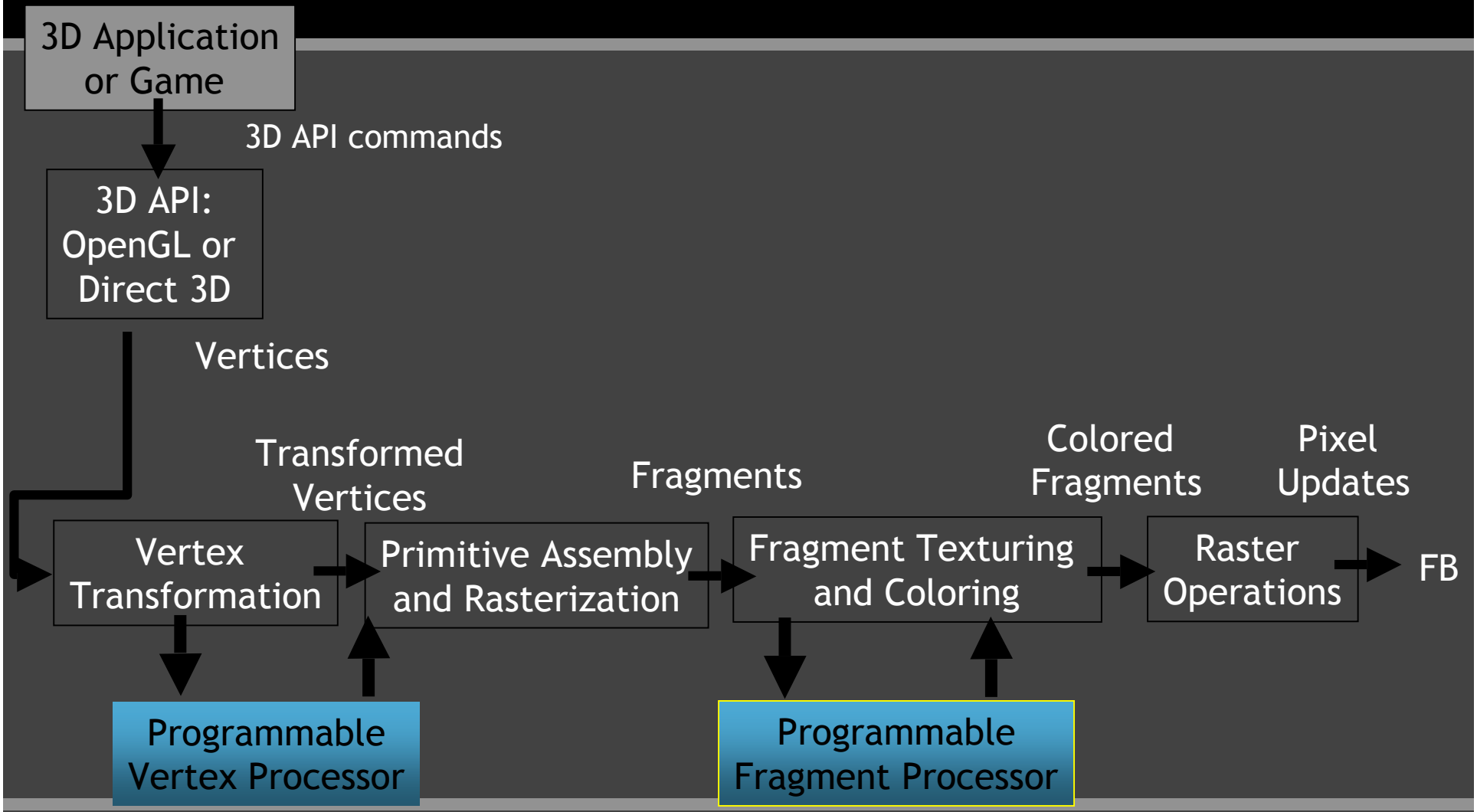
Intel Pentium 4 2.4 Ghz = 55 million transistors
NVIDIA GeForce FX 5800 = 125 million transistor

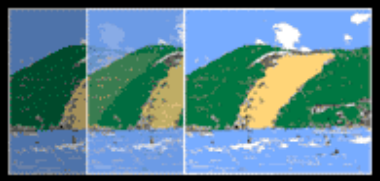
2.27



SIBGRAPI 2005

GPUs

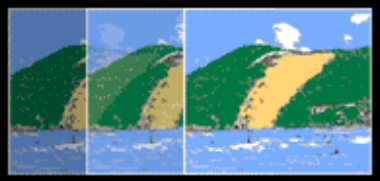




SIBGRAPI 2005

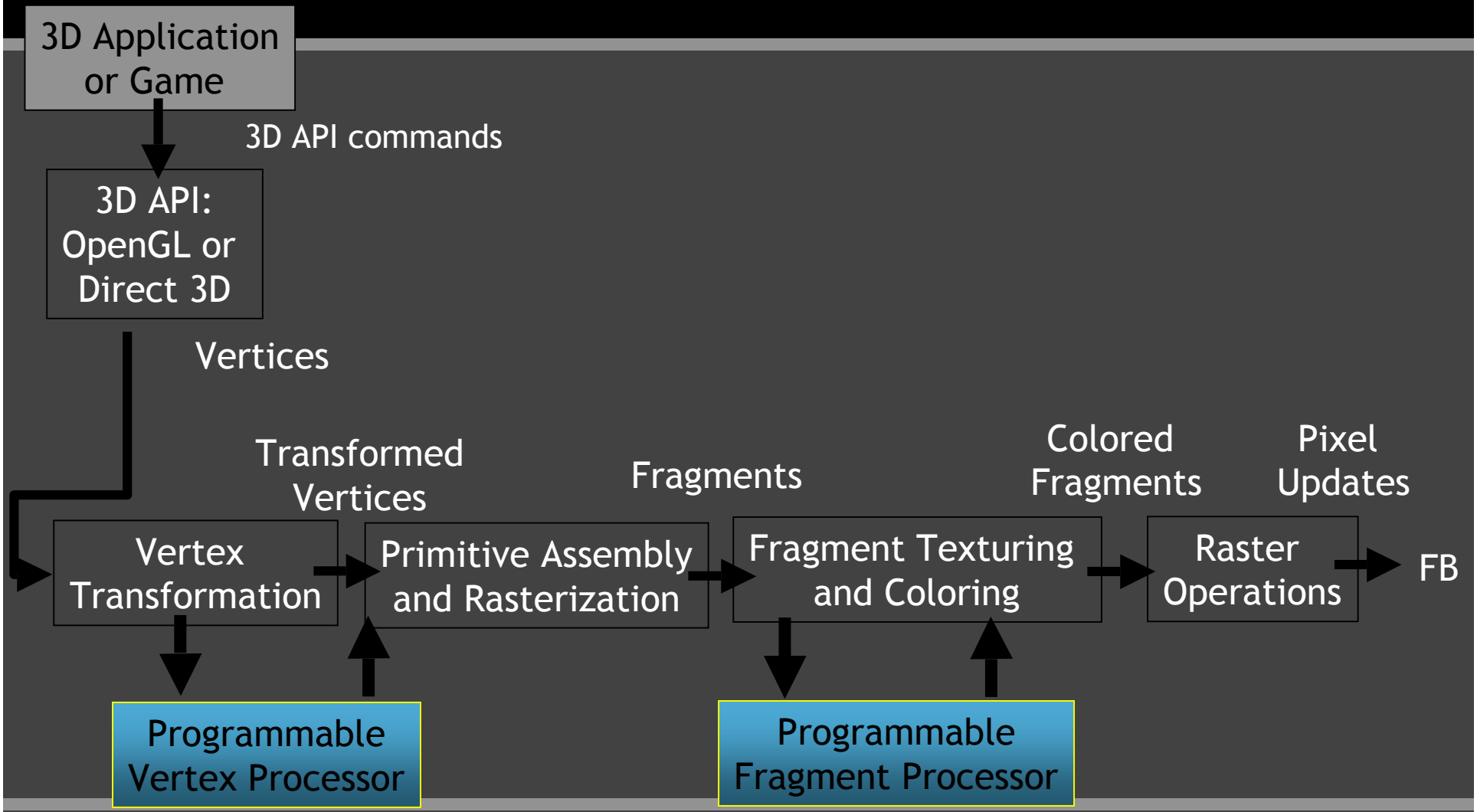
Fifth Generation (04-05)

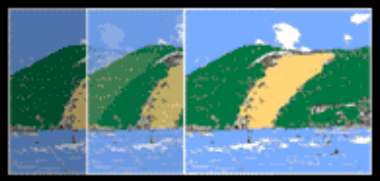
- NVIDIA 6800
  - 16 pipes fragment shader
- NVIDIA 7800:
  - 24 pipes fragment shader
- ATI X1800 (released 4 days ago)
- What is has to offer:
  - Memory access for vertex programs
  - Branches in fragment program
  - Longer fragment programs



SIBGRAPI 2005

GPUs

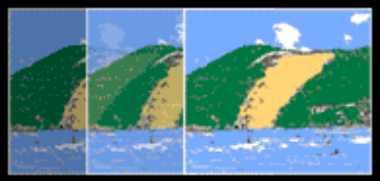




SIBGRAPI 2005

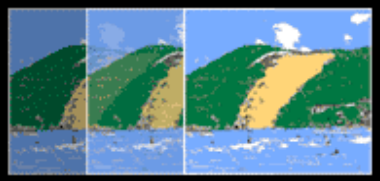
Sixth Generation (2006)

- Unified Architecture:
  - Same Instruction Set (Vertex and Fragment Processors)
  - Geometry Processor:
    - Ability to program how vertices can be combined to form new primitives
    - Ability to create vertices
  - Generalized Output Buffers
  - More ?



What the GPU offers ?

- Streaming processors
- Vector operations:
  - 4-component instructions with 32-bit IEEE floating point operations
- GPU Memory Accesses through Textures:
  - 1-,2-,3-D tables
  - no read-write textures
- Multiple Render Targets
- Limited branching/loops



SIBGRAPI 2005

What we will see today

- Summary of solutions for Volume Rendering problems using GPUs
- Span different architectures
- Open exciting perspectives for other interesting problems:

Dynamic or Time-Varying Problems