Computer Graphics Laboratory Exercise 4

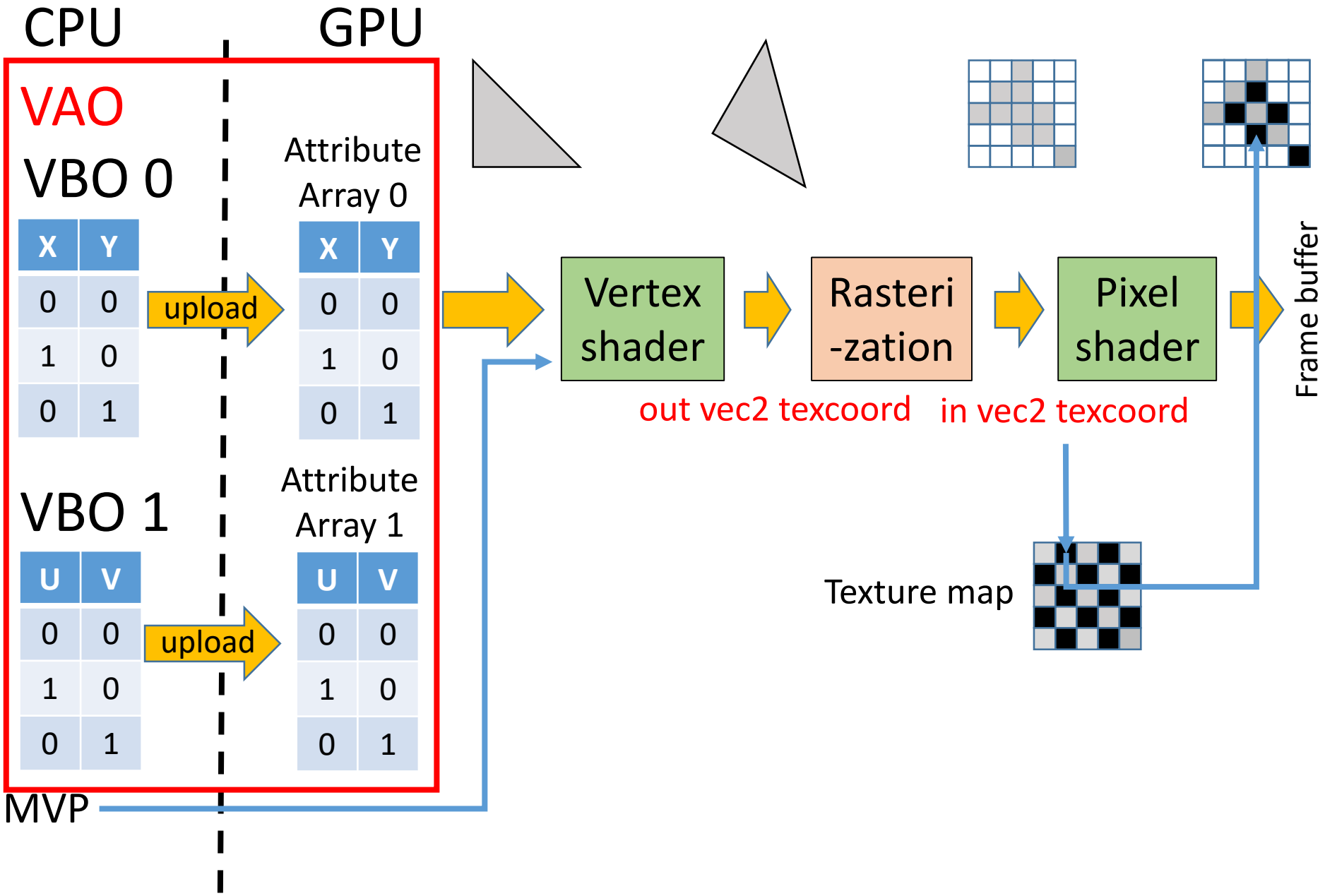
Balázs Csébfalvi

http://cg.iit.bme.hu/portal/en/cgbme

Textured Quad

- Implement a class that represents a textured quad
- Constructor
  - Upload two VBOs: one for the vertex coordinates, and one for the texture coordinates
  - Instantiate class Texture defined in framework.h
- Destructor
  - Delete the Texture instance created in the constructor
- Function Render()
  - Activate the texture using the corresponding setUniform function of the GPUProgram class

Define two VBOs



Class TexturedQuad

```
class TexturedQuad
{
    unsigned int vao, vbo[2];
    Texture* texture;
    vec2 vertices[4];
    int activeVertex;
public:
    TexturedQuad() {
        // ...
        // upload the VBOs here
        // ...
        texture = new Texture("brickwall.bmp");
    }
    ~TexturedQuad() {
        delete texture;
    }
    void Render() {
        gpuProgram.setUniform(*texture, "samplerUnit");
        glBindVertexArray(vao);
        glDrawArrays(GL_TRIANGLE_STRIP, 0, 4);
    }
};
```

Modify the GPU program

Vertex shader

```
#version 330
precision highp float;

uniform mat4 MVP;
layout(location = 0) in vec2 vp;
layout(location = 1) in vec2 vt;

out vec2 texcoord;

void main() {
    gl_Position = vec4(vp.x, vp.y, 0, 1) * MVP;
    texcoord = vt;
}
```

Pixel shader

```
#version 330
precision highp float;

uniform sampler2D samplerUnit;
in vec2 texcoord;
out vec4 outColor;

void main() {
    outColor = texture(samplerUnit, texcoord);
}
```

Rendering a quad with a brick wall texture

Implement drag-and-drop for the vertices of the textured quad