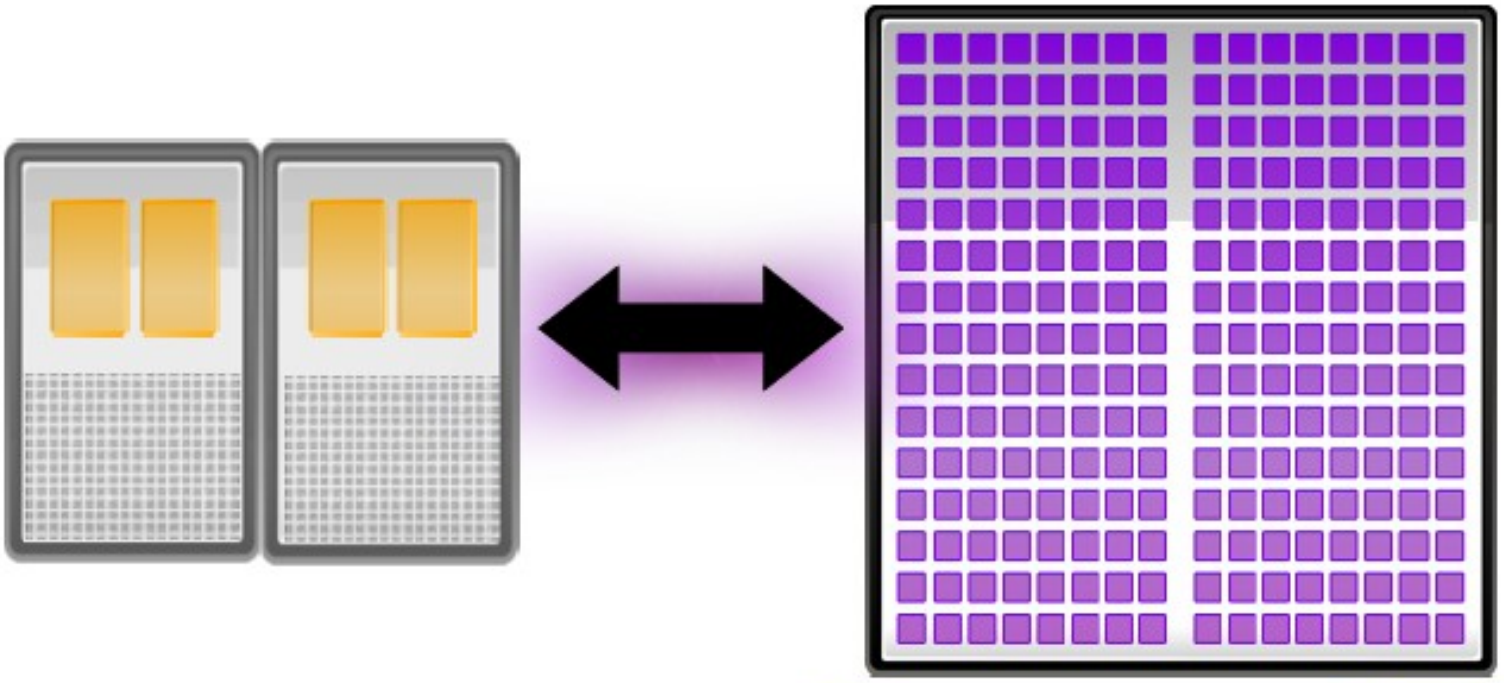
GSLS – Kaip veikia GPU?

A. Noreika

CPU ? GPU



Multicore CPU

Fast Serial Processing

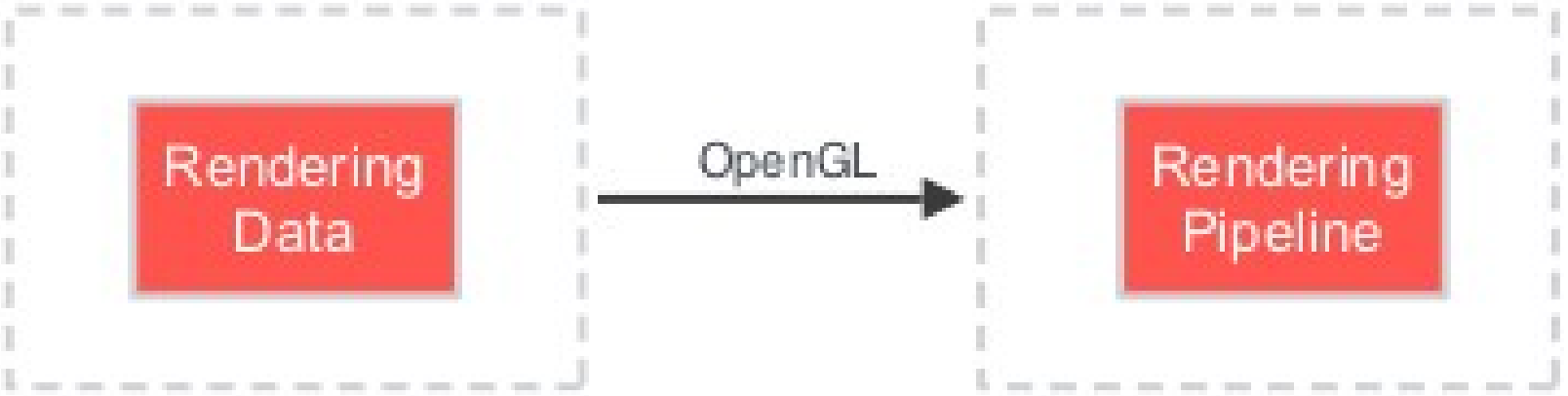
Manycore GPU

Scalable Parallel Processing

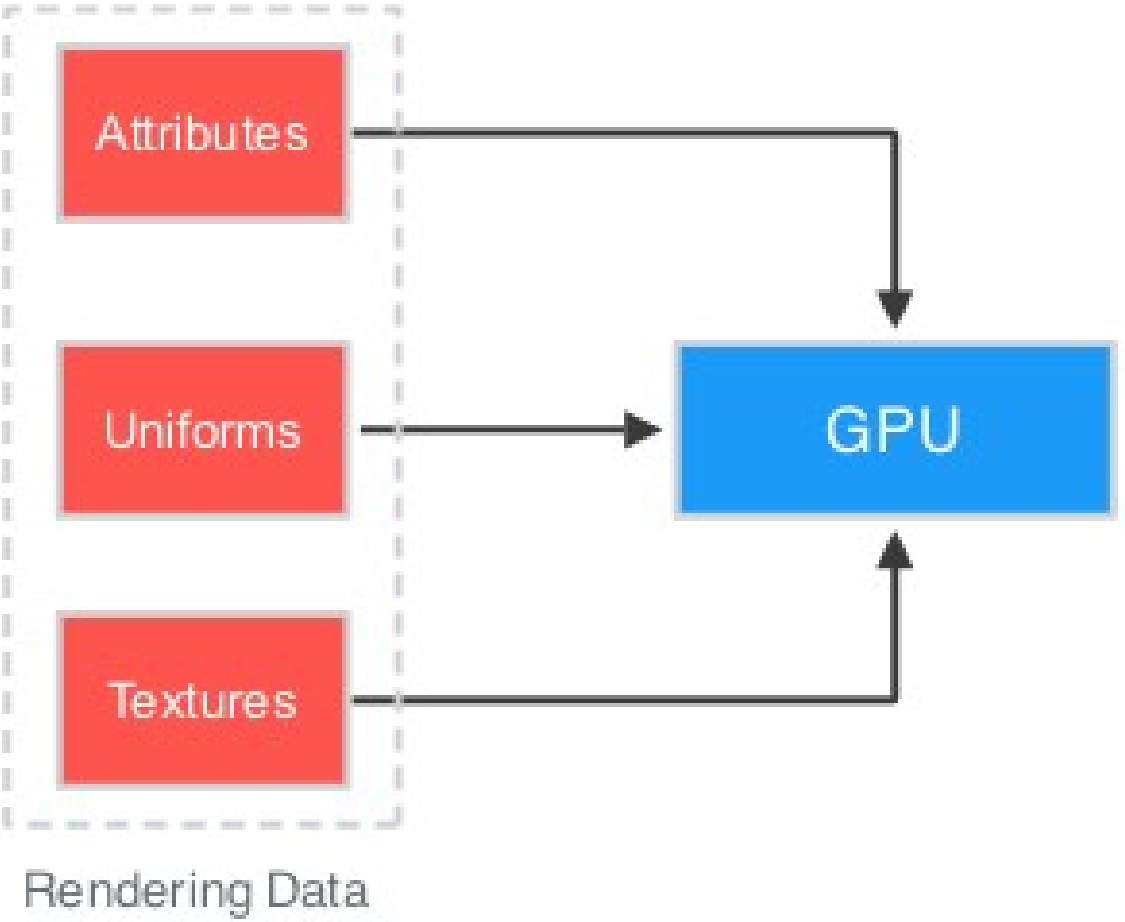
CPU → GPU


intel
Core™ i7


NVIDIA
GeForce GT
555M


Rendering Data
OpenGL
Rendering Pipeline
CPU
GPU

GPU Duomenys



“Attributes”

- Atributus GPU naudoja geometrijos konstravimui, apšvietimo ir tekstūrų skaičiavimams.
- Atributai priskiriami kiekvienai viršūnei
- Tipiniai atributai:
  - *Vertex positions* – geometrijos konstrukcijos skaičiavimai.
  - *Normals coordinates* – apšvietimas.
  - *U-V coordinates* – tekstūrų koordinatės.

“Uniforms”

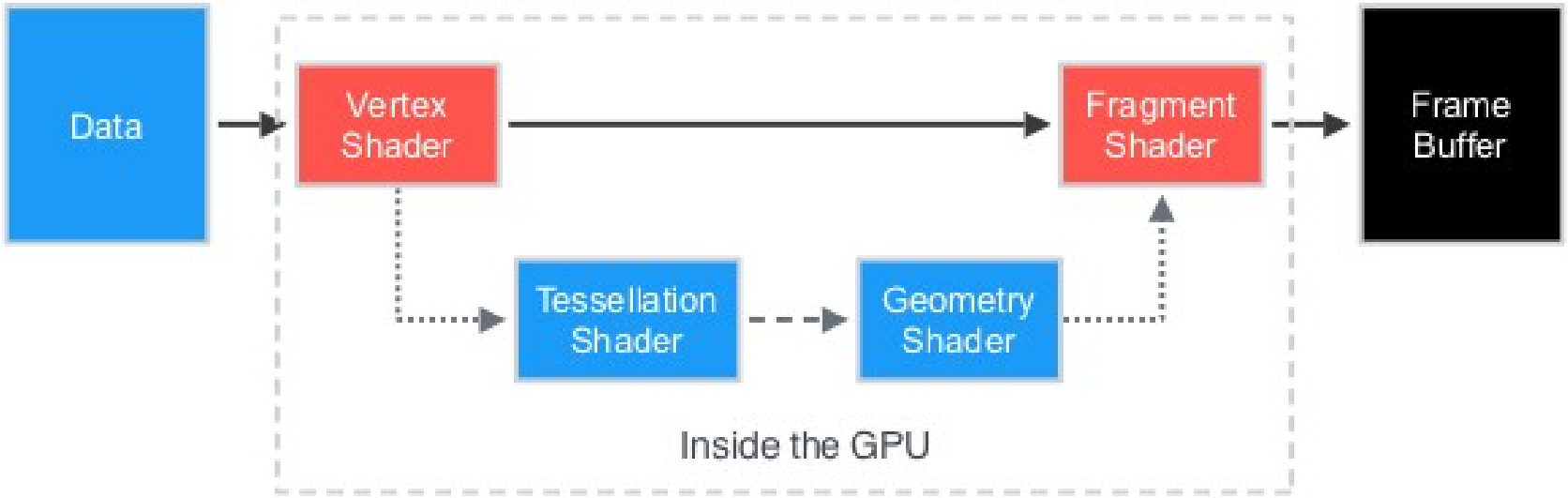
- “Uniforms” GPU perduoda erdvinius duomenų parametrus kaip:
  - Model Space
  - World Space
  - Camera Space
- Šie parametrai pozicionuoja objektus 3D ervėje ir ekrane.
- “Uniforms” taip pat naudojamas bendroms reikšmėms perduoti, kaip pvz.: temperatūra, pelės pozicija ir pan.

“Textures”

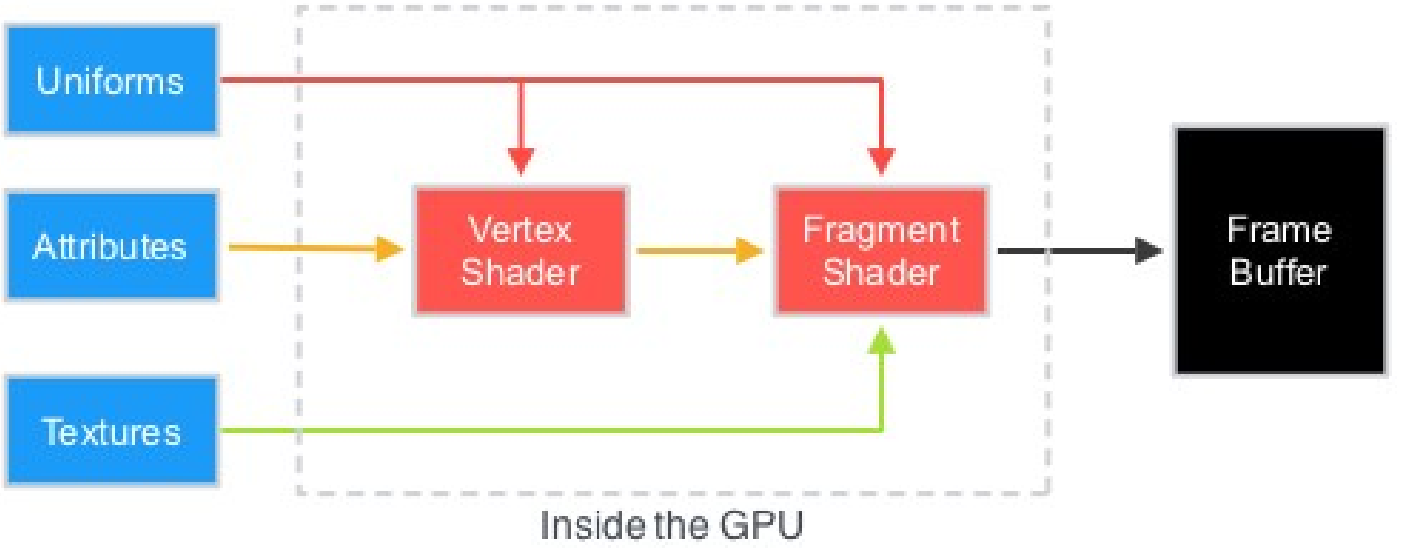
- Tekstūros iš principo yra duomenų matricos.
- Grafikoje tekstūromis perduodamos spalvos, apšvietimo žemėlapiai, “maskės”, filtrai ir t.t.
- Taikant ne grafikos uždaviniams – dažnai perduodami duomenys.

GPU: “Shader” tipai

- Vertex Shader
- Fragment Shader
- Tessellation Shader
- Geometry Shader



Kaip duomenys keliauja į GPU ?

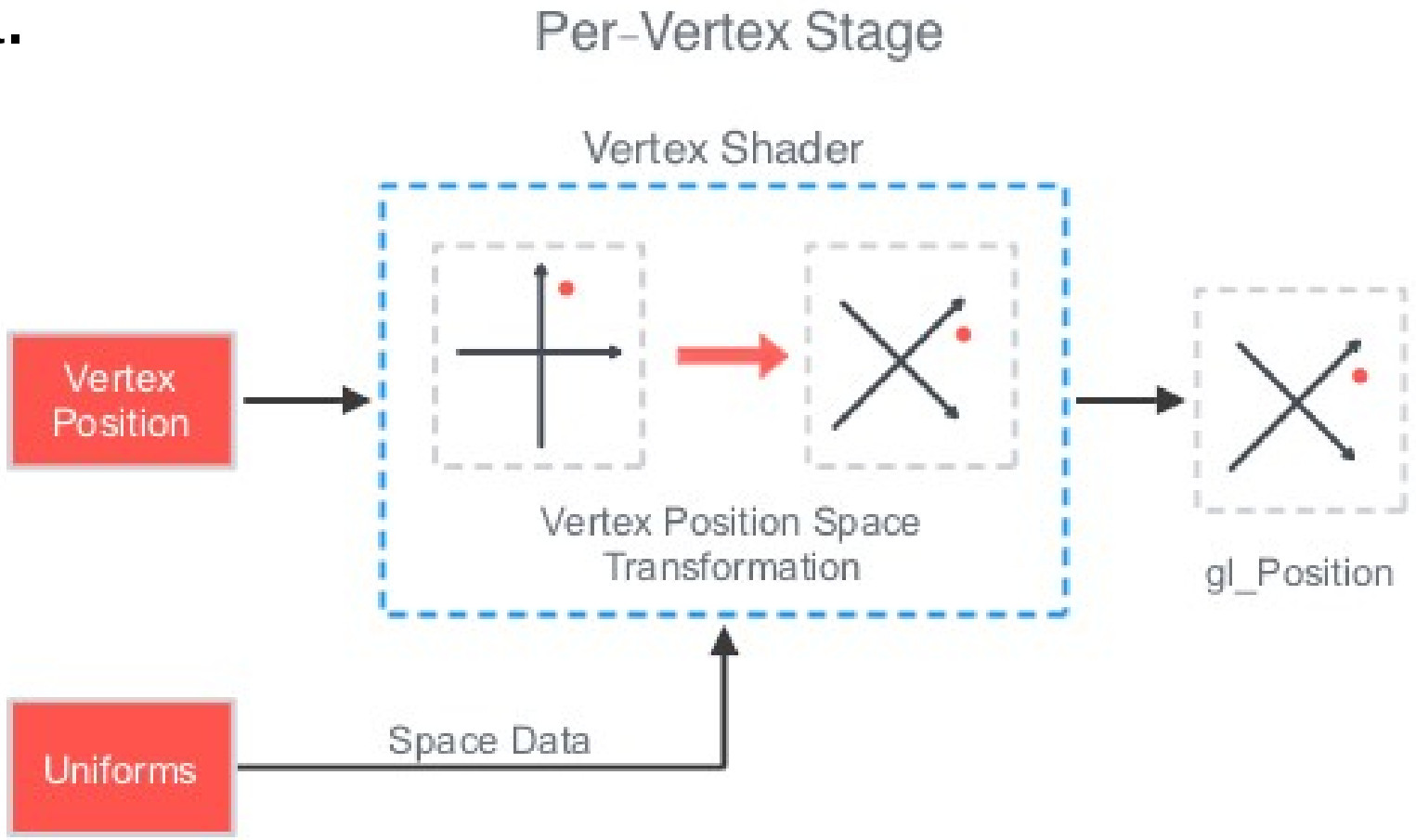


“Rendering pipeline”

- Per-Vertex Operation
- Primitive Assembly
- Primitive Processing
- Rasterization
- Fragment Processing
- Per-Fragment Operation

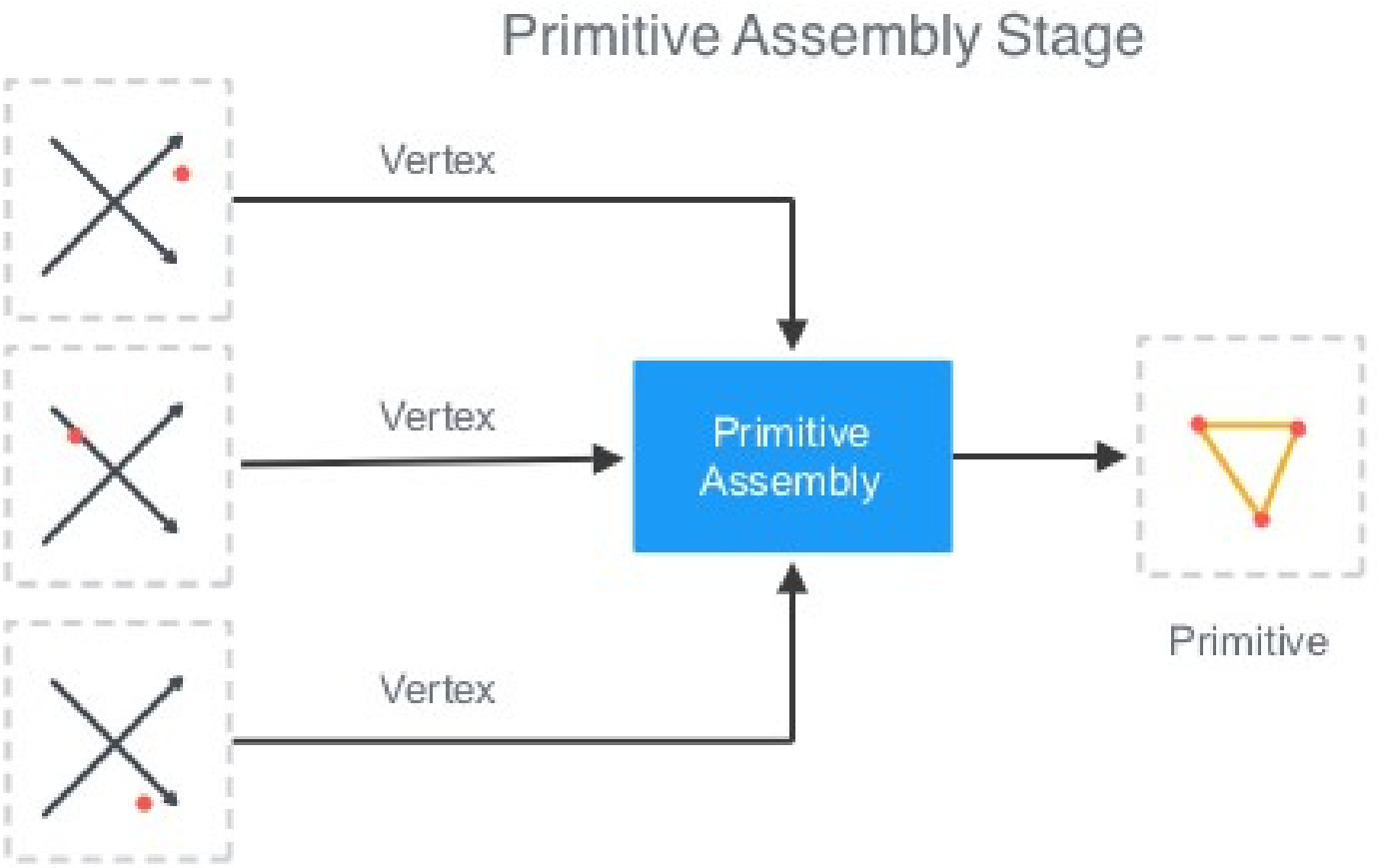
Per-vertex operation

- Viršūnės apdorojamos “Vertex Shader” - kiekviena viršūnė transformuojama erdvės matrica.



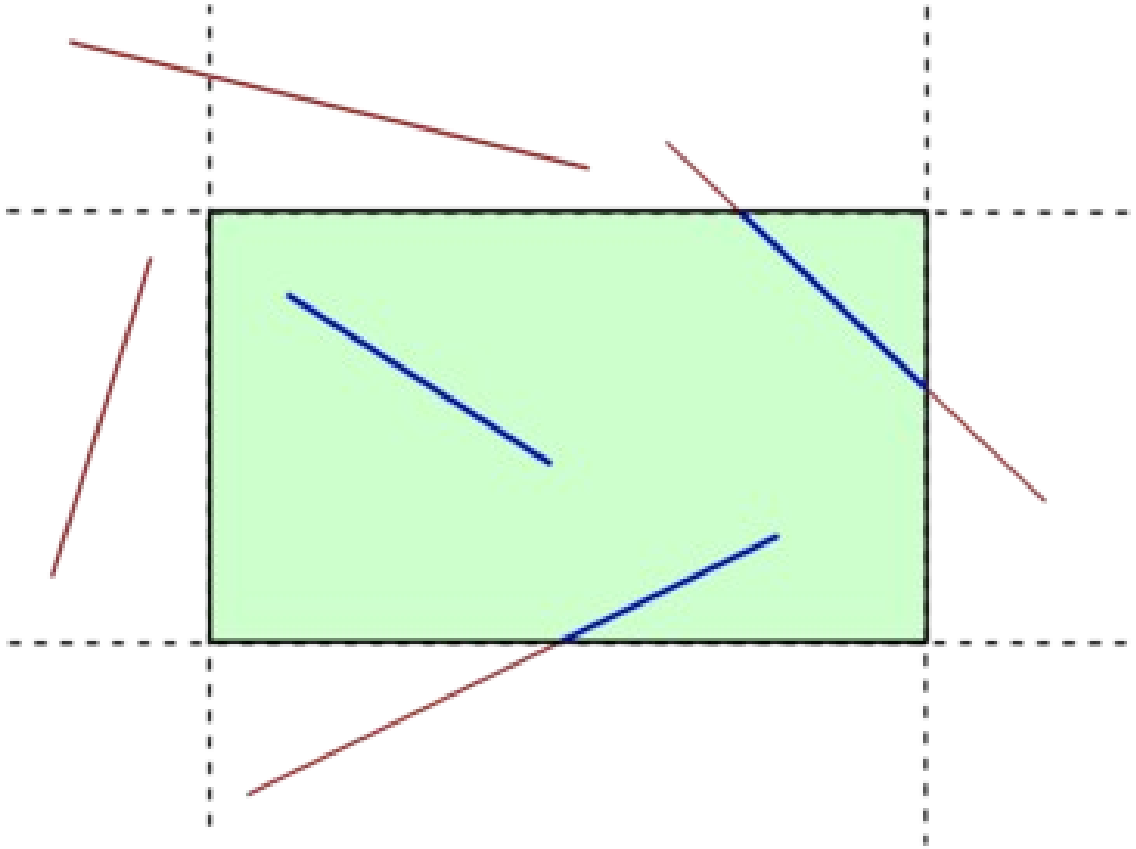
Primitive assembly

- Po “Vertex shader” - viršūnės sujungiamos į primityvus.



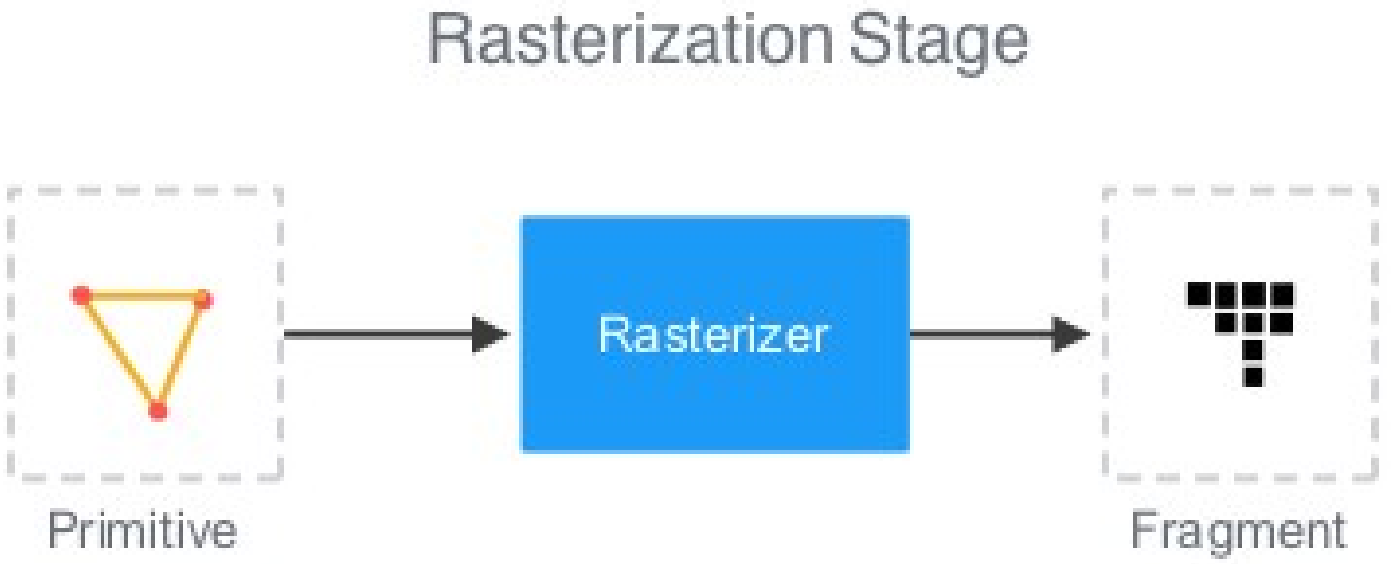
Primitive processing

- Kaip taisyklė šiame etape taikomas nematomų zonų nukirpimas - “clipping”



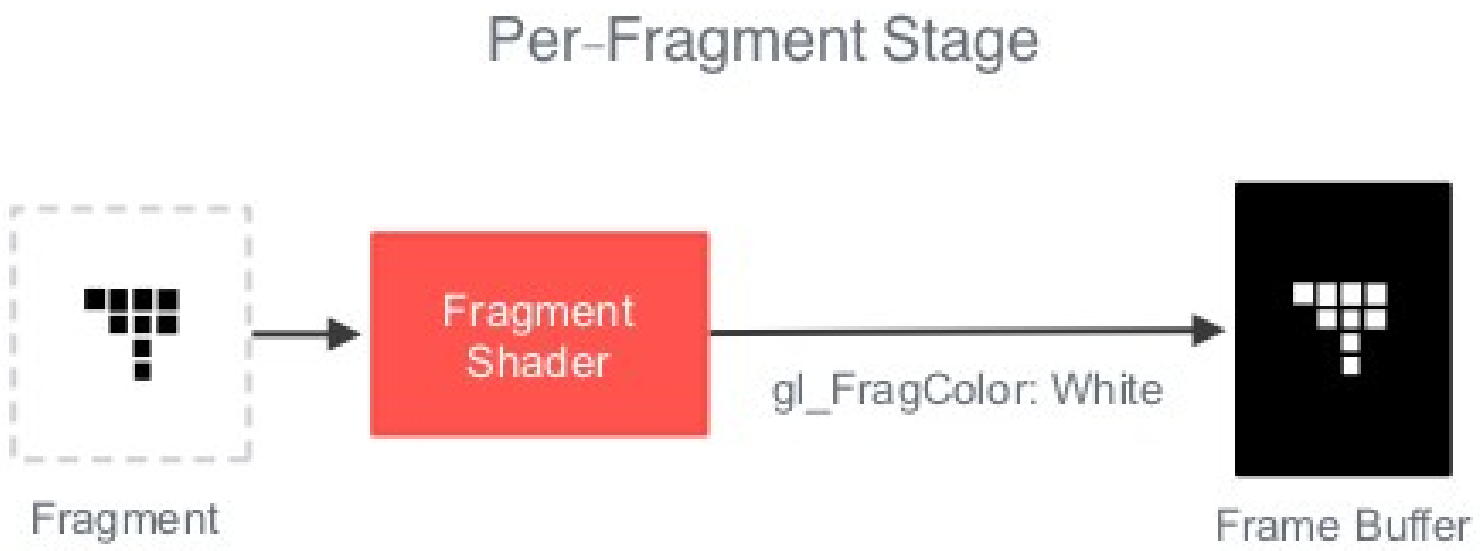
Rasterization

- Primityvai verčiami pikseliais.
- Pikseliai patenkantys į primityvo zoną vadinami Fragmentais



Fragment processing

- Fragmentai apdorojami “Fragment shader”
- Čia vykdomos pikselių (per-pixel) operacijos. Pritaikomos tokios savybės kaip spalva ar tekstūra.



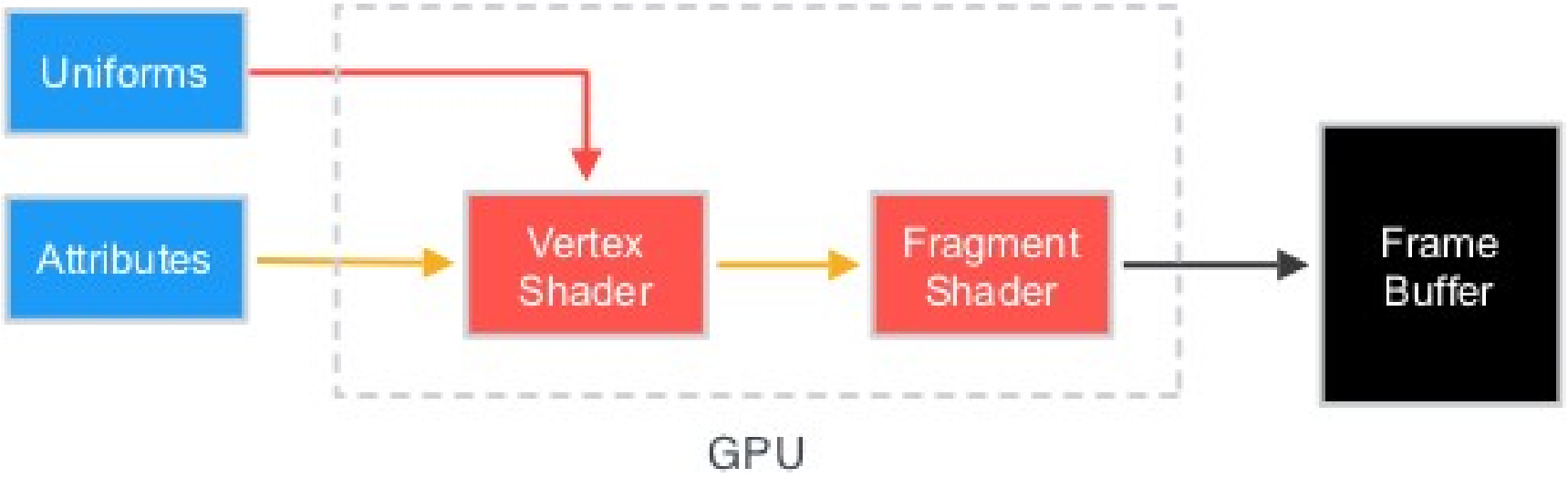
Per-Fragment Operation

- Galutinis fragmentų apdorojimas. Gali būti vykdoma visa eilė testų, pvz:
  - *Pixel Ownership test*
  - *Scissor test*
  - *Alpha test*
  - *Stencil test*
  - *Depth test*
- Pabaiga – suformuojamas galutinis vaizdas. Pikseliai išsaugomi vaizdo buferyje (ang. *Framebuffer*).
- Tam tikrai atvejais vaizdas gali būti saugomas tekstūroje, pvz veidrodžio efektui išgauti.

Apibendriname. Kaip veikia Shaderiai?

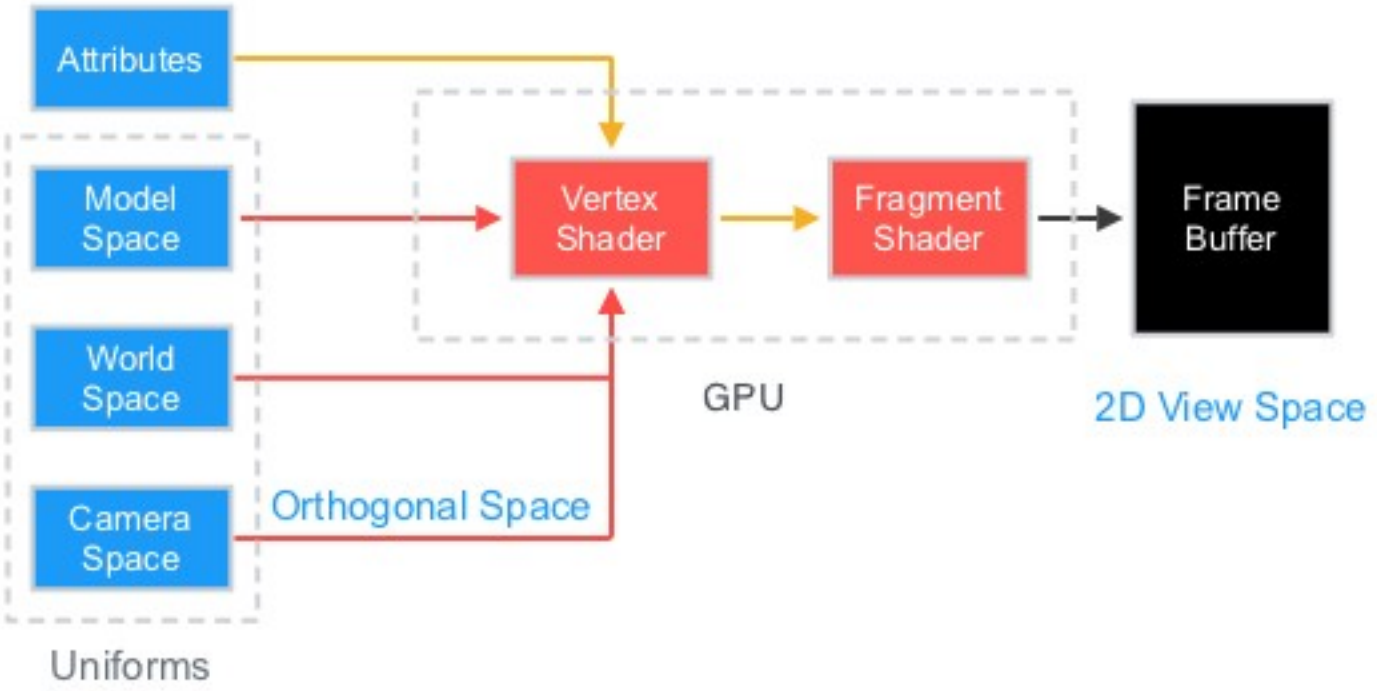
- Shaderiai GPU skaičiavimuose suteikia didelį lankstumą. Baziniai reikalingi dalykai:
  - Viršūnių pozicijos (Attributes)
  - Model-World-View Space transformacijos (Uniforms)
- “Vertex shader” transformuoja viršūnių pozicijas naudodamas Model-World-View space matricas ir suformuoja duomenis “Fragment shader”.

Apibendriname. Kaip veikia Shaderiai?



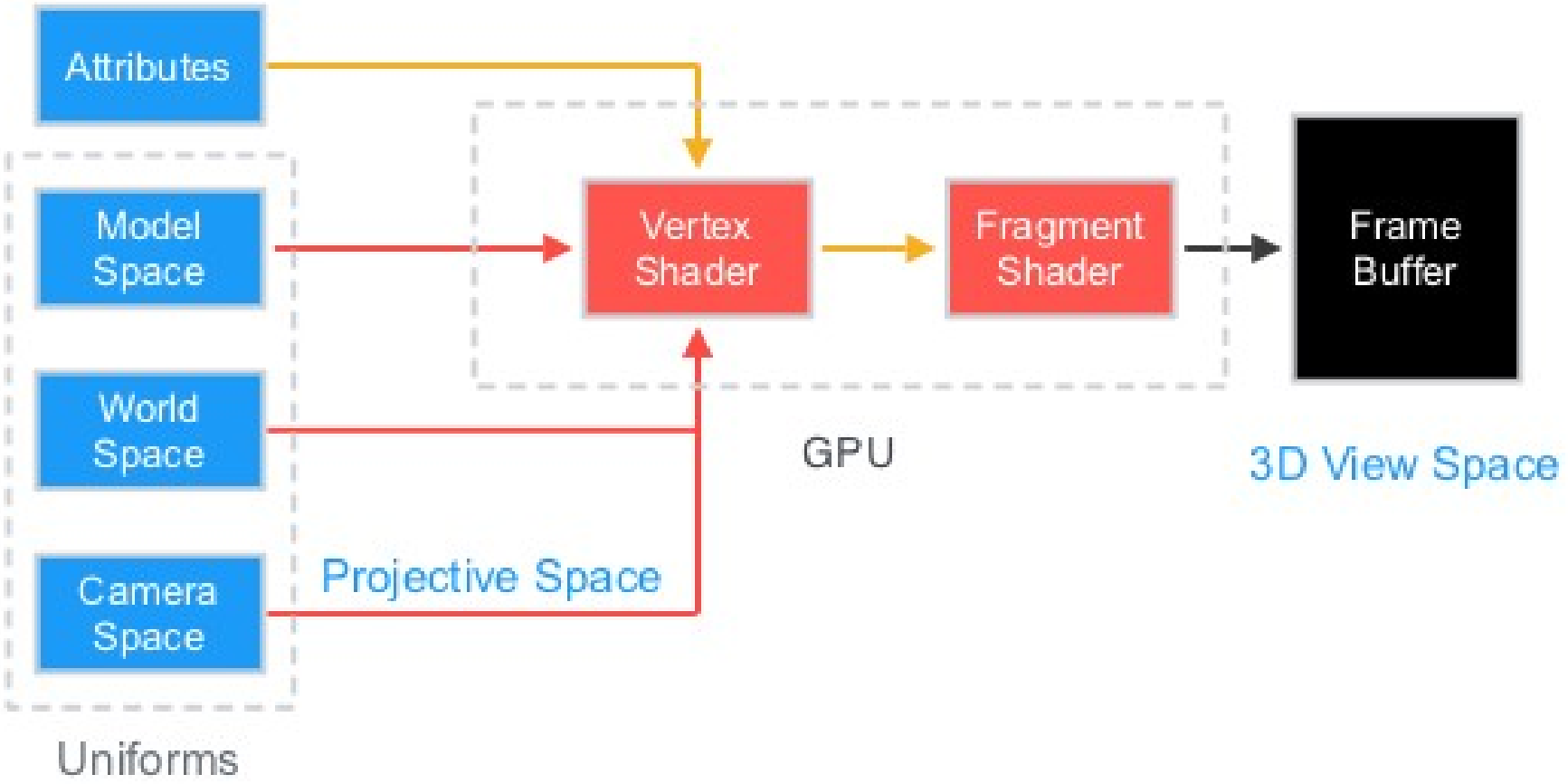
2D/3D vaizdo manipuliacija su Uniforms

- Manipuliodami View-space (uniform) kontroliuojame 2D/3D kameros perspektyvą.
- Ortogonali kamera sukuria 2D vaizdą.



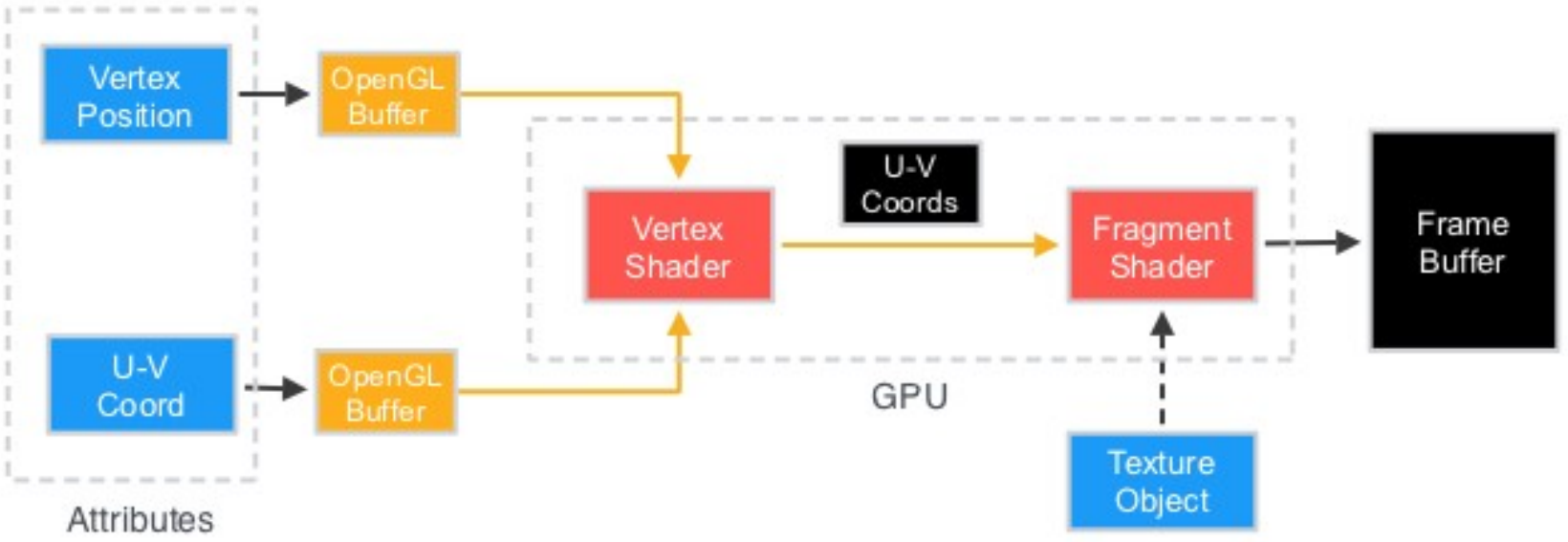
2D/3D vaizdo manipuliacija su Uniforms

- Perspektyvinė kamera sukuria 3D vaizdą.

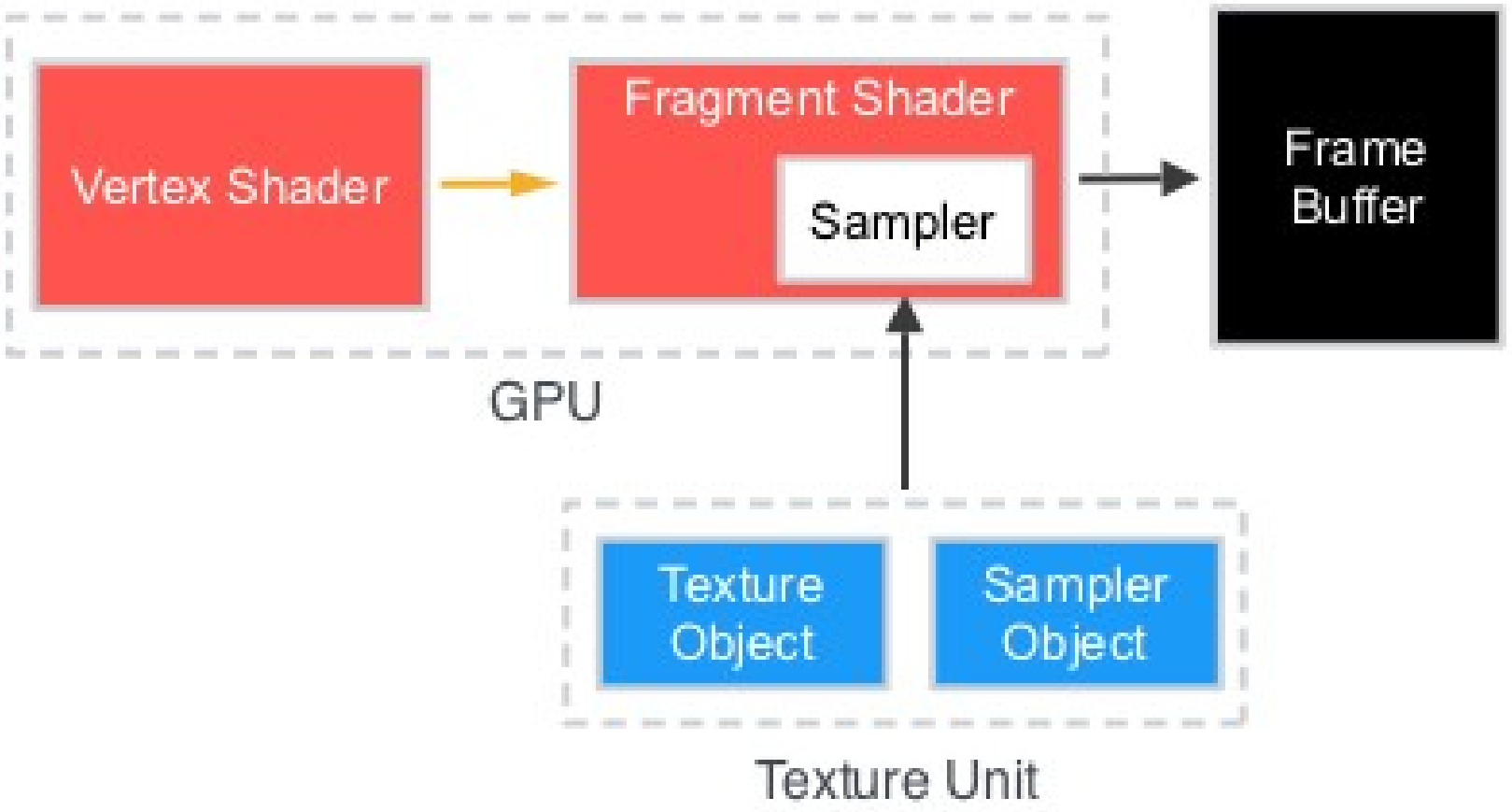


Tekstūravimas

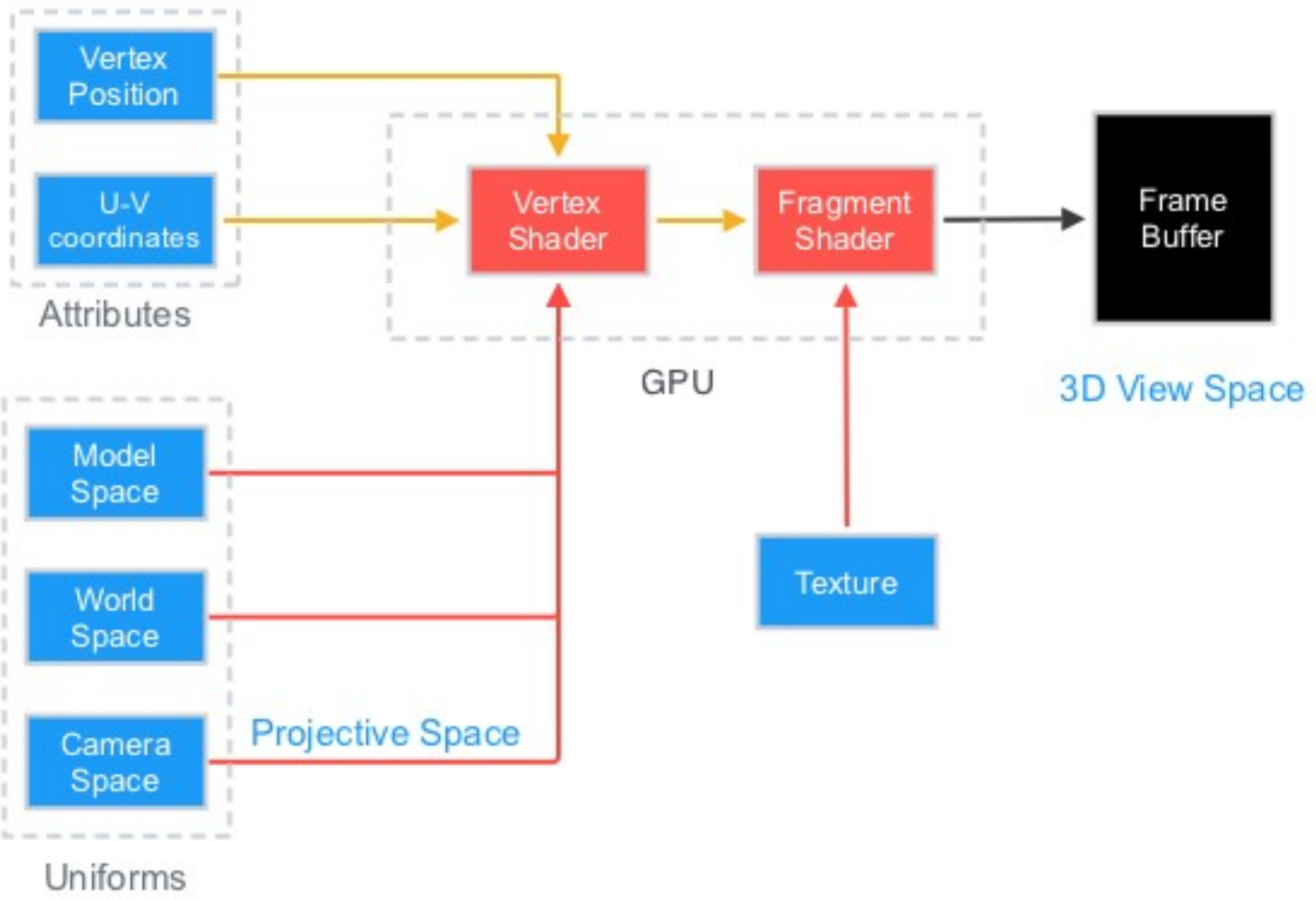
- Tekstūravimui reikalinga:
  - UV koordinatės - patenka per “vertex shader” kaip atributas, toliau perduodamos kaip “varying” tipas į “Fragment shader”
  - “Sampler” tipo tekstūra “Fragment” shaderiui.



Tekstūravimas



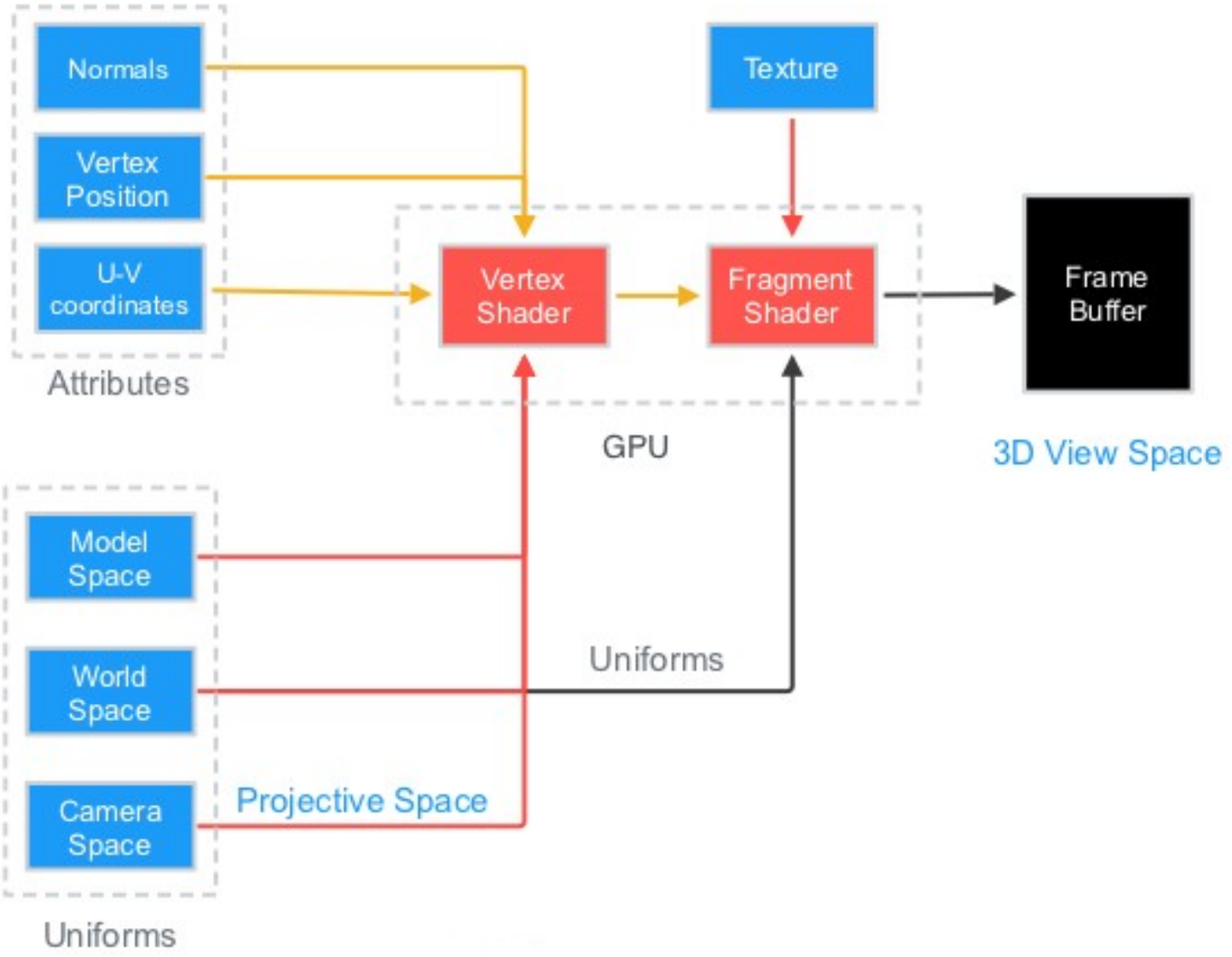
Bendras vaizdas



Apšvietimas

- Apšvietimas yra viskas. Be apšvietimo negalėtume suvokti 3D erdvės.
- Apšvietimo simuliacijai minimaliai reikalinga:
  - viršūnių pozicijos
  - normalių koordinatės.
- Apšvietimo simuliacija gali būti vykdoma tiek Vertex tiek Fragment (realistiškesnis apšvietimas) shaderiuose.

Apšvietimas



Nuorodos

- CPU vs GPU https://www.youtube.com/watch?v=-P28LKWTzrI
- Introduction to shaders https://www.youtube.com/watch?v=j_h3GdMtO0M
- How rendering pipeline works? http://www.haroldserrano.com/blog/how-to-develop-a-rendering-engine-an-overview
- Online GLSL editor http://shdr.bkcore.com/
- GLSL effects http://glslsandbox.com/, https://www.shadertoy.com