Hardware-driven Visibility Culling

Jeong Hyun Kim

KAIST (Korea Advanced Institute of Science and Technology)

KAIST

Contents

- Introduction
- Background
  - Clipping
  - Culling
    - Z-max
    - (Z-min)
    - Filter
    - Programmable culling unit
- Proposed Idea
- Conclusion

Introduction

- **The goal of 3D accelerator hardware is Real Time Rendering of Photorealistic Scene.**
  - **It needs more processing time, more memory bandwidth.**
  - **To reduce processing time**
    - **GPU**
  - **To reduce memory BW**
    - **R&D of faster memory → expand possible BW**
    - **Culling Schemes → reduced asked BW**

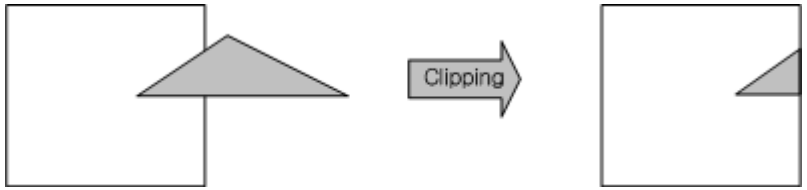
Introduction

- **Software-driven**
  - **Use CPU**
  - **Cons -- CPU is slow!**
  - **Pros -- Reduce data before GPU processing**

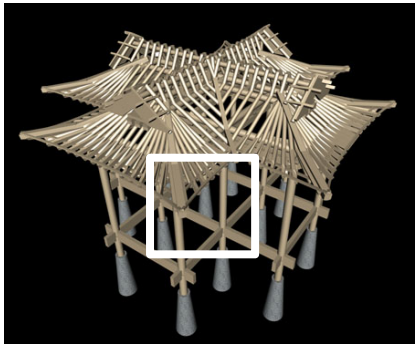
- **Hardware-driven**
  - **Means GPU culling.**
  - **Cannot reduce burden of Geometry processing**
    - **Only reduce burden of Fragment processing**
  - **But, Fragment processing is Bottleneck~!**
    - **Memory Access**
  - **Ex) R580(ATI, X1900) has 8 VS and 48 PS.**

Background_clipping

- Clipping
- What does clipping mean? (literally)



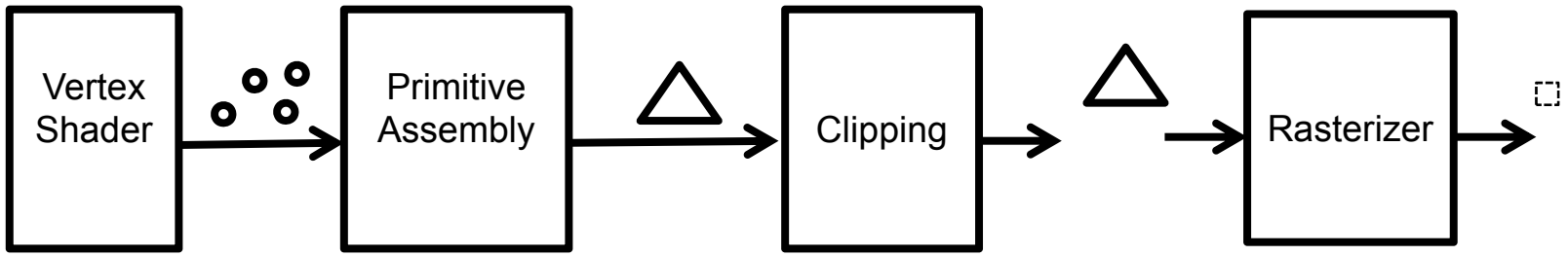
- In fact, clipping is not culling.
- In Hardware Implementation, clipping unit does view-frustum culling.



KAIST

Background_clipping

- **Clipping**

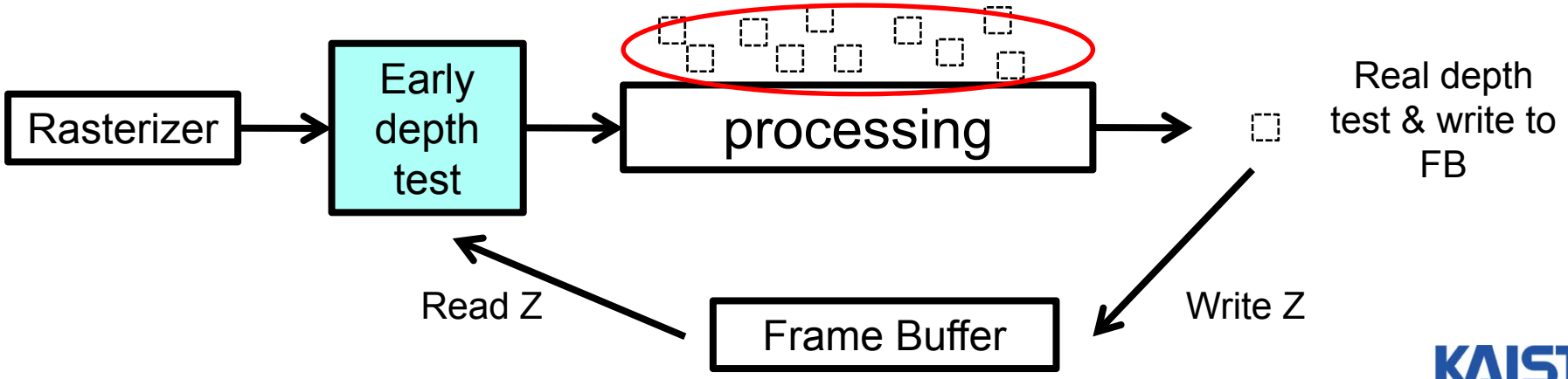


- **Reject or Pass or Clip&pass**
  - **Many triangles are rejected**
  - **A very simple primitive culling method.**

KAIST

Background_fragment culling

- **Fragment culling**
- **If a pixel is occluded something, we don't have to process that pixel.**
  - **Use depth info. in Frame Buffer?**
    - **Consistency problem.**
    - **Write FB is end of processing.**
    - **There is many pixels in processing.**



KAIST

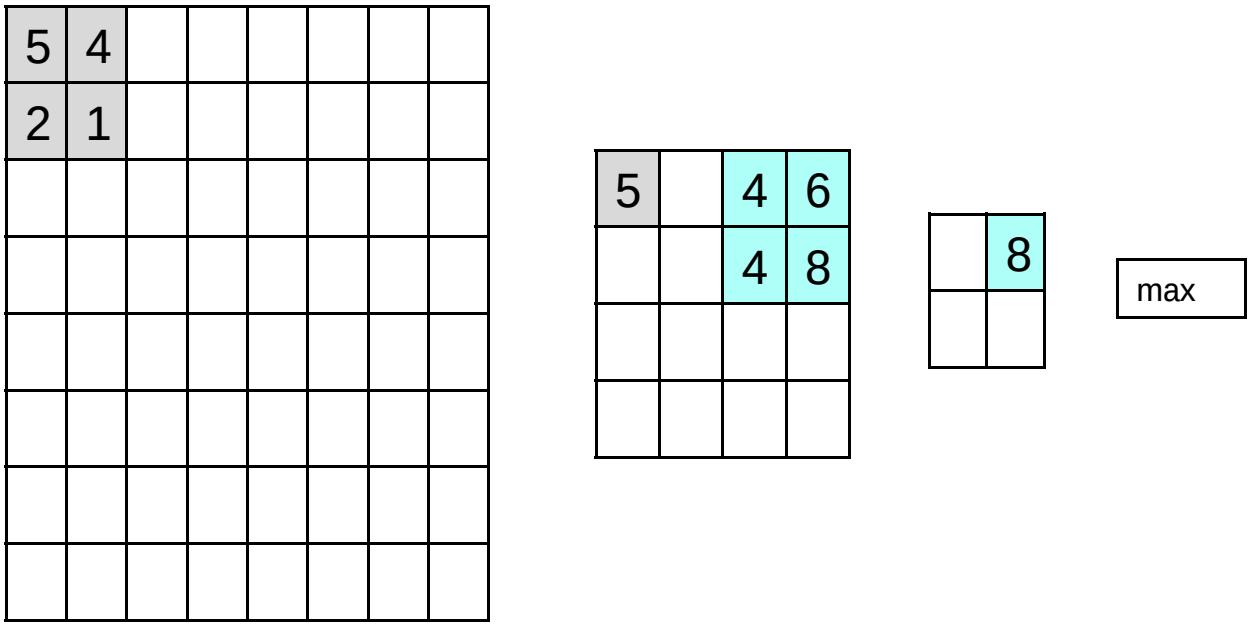
Background_fragment culling

- We have to use another information
- Hierarchical Z-buffer
  - Z-max algorithm
  - Have another z-cache, HZ-cache.
  - It needs much memory size.
  - Increase another memory BW.
  - High rejection ratio.

- Depth Filter
  - Z-max algorithm
  - Rough Culling
  - Need much smaller cache than HZ-buffer

Background_HZ-buffer

- **Hierarchical Z-buffer**
- **A value of level N takes the max value of four pixels of N-1 level.**



- **There is cache structure issue.**

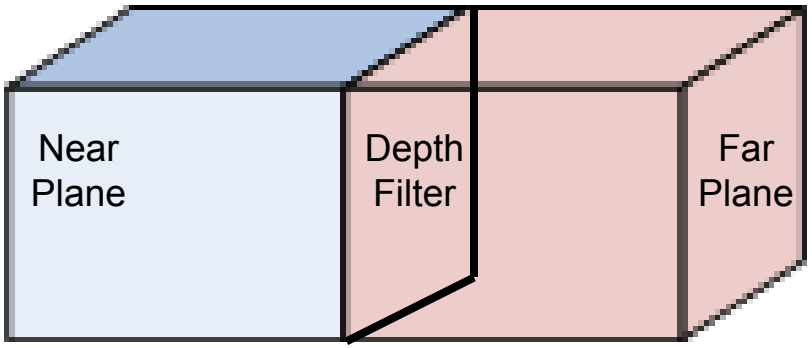
KAIST

Background_depth filter

- Depth Filter
- HZB needs another big size cache.
  - Depth is 24bit for each pixel.
  - Hierarchical map needs several maps for each level
- Instead of that, all that depth filter needs are only 1 or 2 bits per pixel.
  - Filter mask.
- Rough culling.

Background_depth filter

- Depth filter separates view volume.
- Pixels front of DF(Blue area) are passed to next and make high the filter mask.
- Pixels back of DF(Red area) are checked by filter mask.
  - If mask is high, the pixel is rejected.
  - If mask is low, the pixel is accepted.
    - No change in filter mask. Still low.

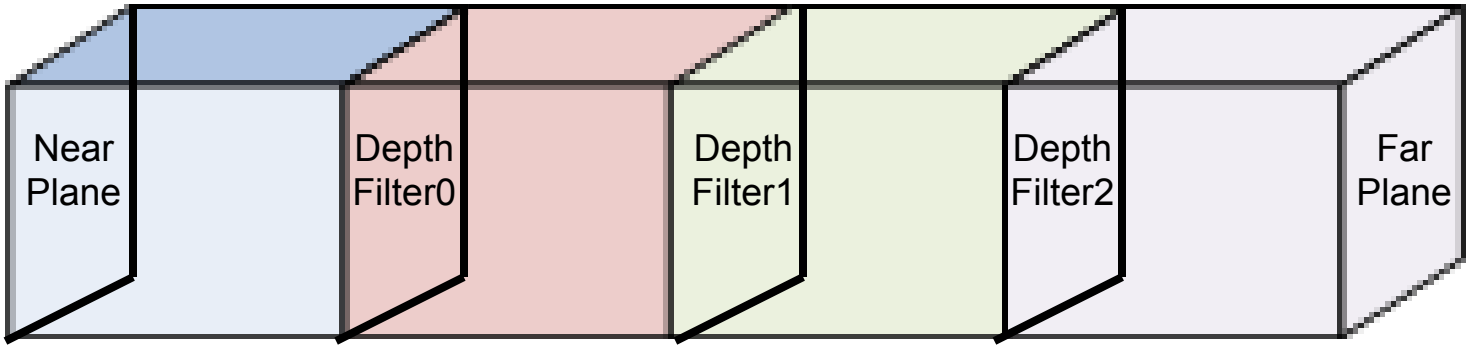


1 bit per pixel

KAIST

Background_depth filter

- 2 bits per pixel
  - We can separate view volume into 4 spaces.
  - → higher rejection ratio.



KAIST

Background_depth filter

- Adaptive modification of location of DF is possible.
  - Maximize rejection ratio.
    - When the # of pixels in front of DF and the # of survived pixels(back of DF and passed) are same.
  - Decrease memory BW proportionally to depth complexity.

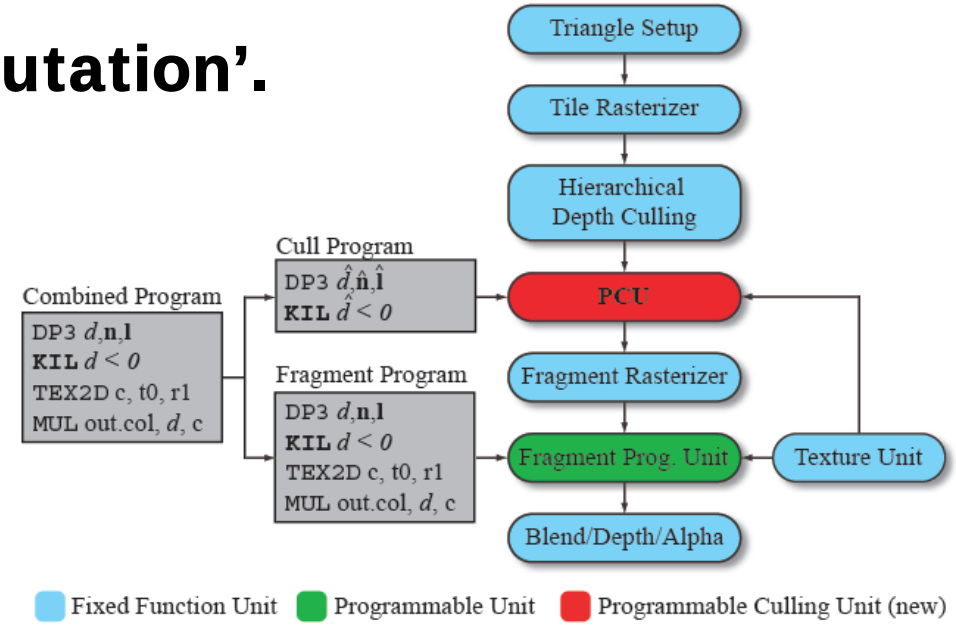
- DF method can be applied to primitive culling.

Background_PCU

- Programmable culling unit
- The PCU does tile based culling.
- The fundamental difference compared to Hierarchical Depth Culling is..
  - HDC does fixed function computations.
  - PCU bases its decision on the output from a shader program execution.

Background_PCU

- Cull program executes KIL instruction fast.
- PCU executes per-tile computation.
  - A tile can be killed only if all of fragments in the tile can be killed.
  - Use ‘interval computation’.



Background_PCU

- **Program Compilation and Separation**
  - **This PCU must be ease to use.**
    - **The programmer writes a combined program.**
    - **It is up to the driver or compiler to separate the cull and fragment program.**
  - **Use ‘Dead Code Elimination’ [Cytron 1991]**
    - **For fragment program**
      - Mark color outputs, depth outputs, all KIL statements
    - **For cull program**
      - All CUL and KIL statements
    - **Remove all code not contributing to the result**

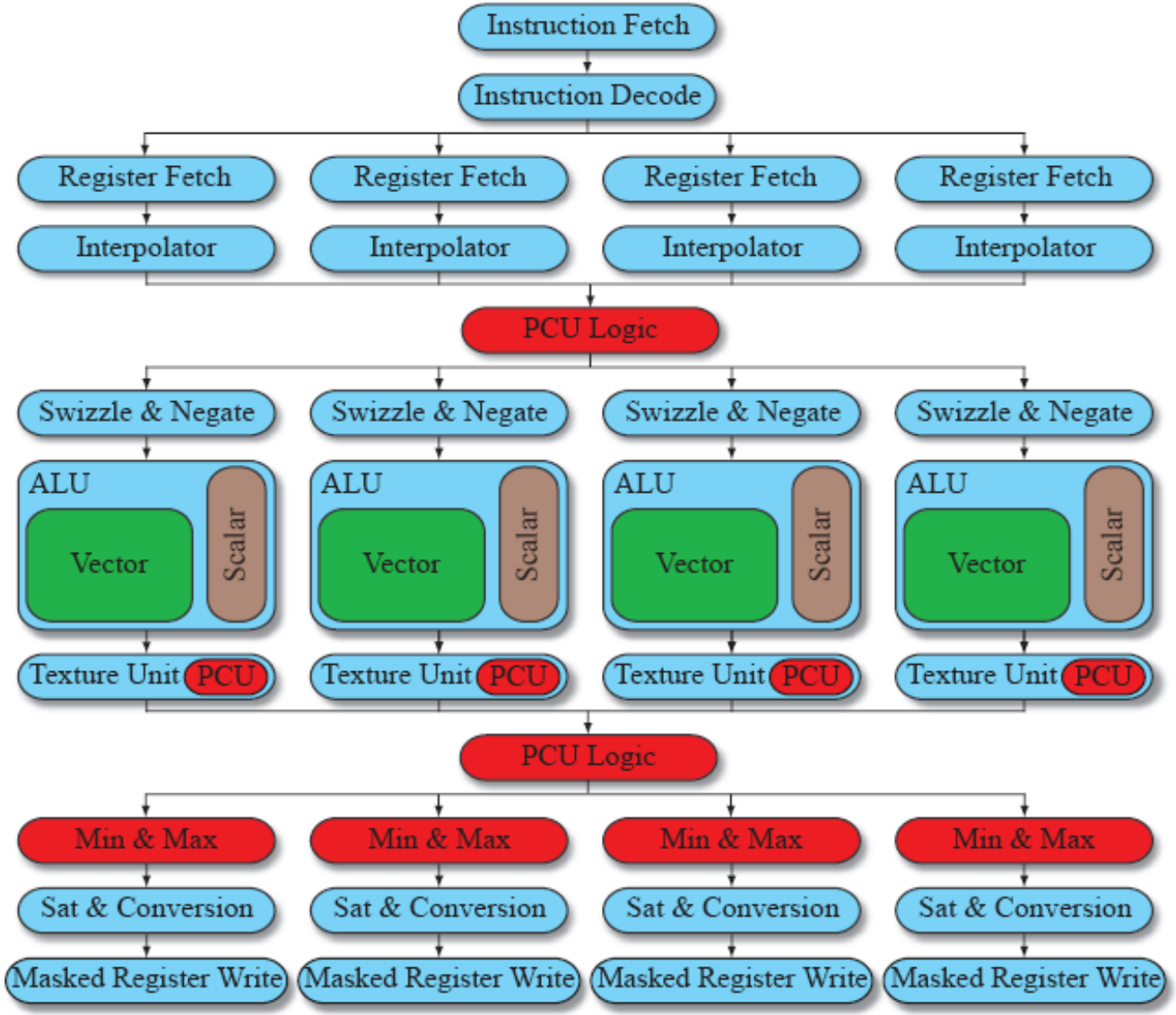
Background_PCU

- Higher Level Culling
  - PCU does culling in per-tile, per-triangle basis.
  - If triangles are very small. (even smaller than a pixel)
  - Than, per-tile check may be meaningless overhead.
  - Delay stream like unit [Aila 2003]
    - This unit receives triangles in the same order.
    - This unit groups triangles until the group will be larger than a tile.
    - If a group grows enough, execute cull program.

Background_PCU

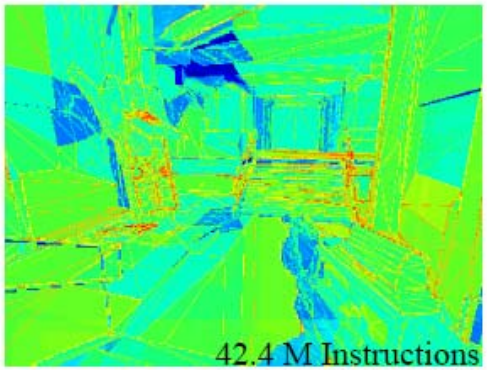
- Implementation of PCU

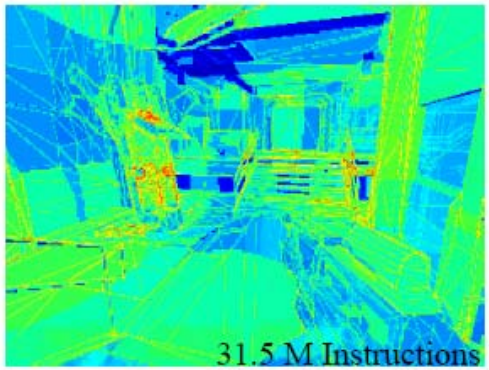
- PCU is combined with shader unit.
  - Reuse existing hardware.
- PCU Mainly consists of extra control logic before and after the ALU.
  - Role: value rerouting, detecting and handling special cases.
- Additional texture unit .
- Min & Max units to assemble the result.

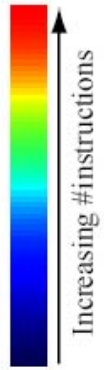

Instruction Fetch
Instruction Decode
Register Fetch
Register Fetch
Register Fetch
Register Fetch
Interpolator
Interpolator
Interpolator
Interpolator
PCU Logic
Swizzle & Negate
Swizzle & Negate
Swizzle & Negate
Swizzle & Negate
ALU
Vector
Scalar
ALU
Vector
Scalar
ALU
Vector
Scalar
ALU
Vector
Scalar
Texture Unit
PCU
Texture Unit
PCU
Texture Unit
PCU
Texture Unit
PCU
PCU Logic
Min & Max
Min & Max
Min & Max
Min & Max
Sat & Conversion
Sat & Conversion
Sat & Conversion
Sat & Conversion
Masked Register Write
Masked Register Write
Masked Register Write
Masked Register Write

KAIST

Background_PCU







KAIST

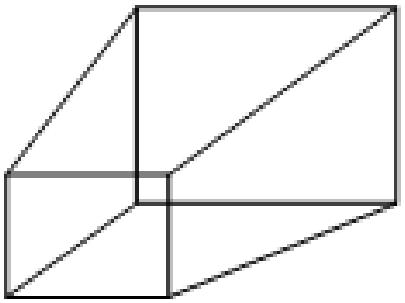
Proposed Idea

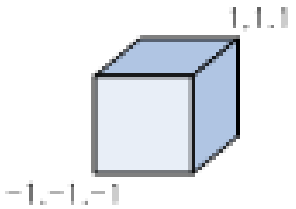
- Clipping unit is fixed functional culling unit.
- Every Graphics hardware implements Clipping unit.

- Apply exist Clipping unit to more culling.
  - If we use Clipping unit, it can be the simplest culling scheme.

Proposed Idea

- **Closer to Clipping unit.**
  - **If we use normalization system, VS outputs normalized vertices to view volume.**





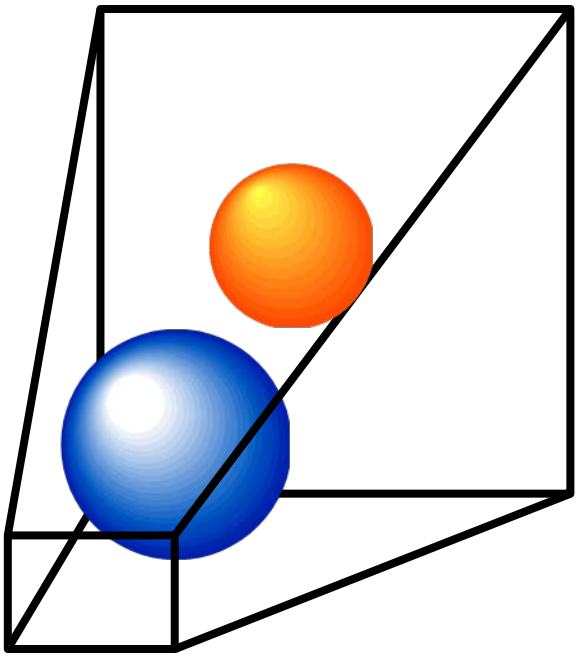
- **If the depth of input vertex is over 1, that means out of Far Plane.**
  - **We are using homogeneous coordinate system.**
  - **(w,y,z,w)**

KAIST

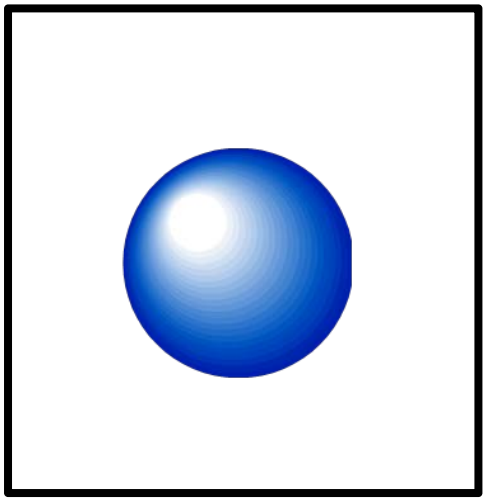
Proposed Idea

- When checking
  - ‘Z == w’ means the vertex is on the Far Plane.
  - We can simply modify the Far Plane by multiplying a value to W.
  - Lower W → closer Far Plane.
  - How about make closer the far plane at the value of Z-min of last frame?
    - We can obtain additional culling effect.

Proposed Idea



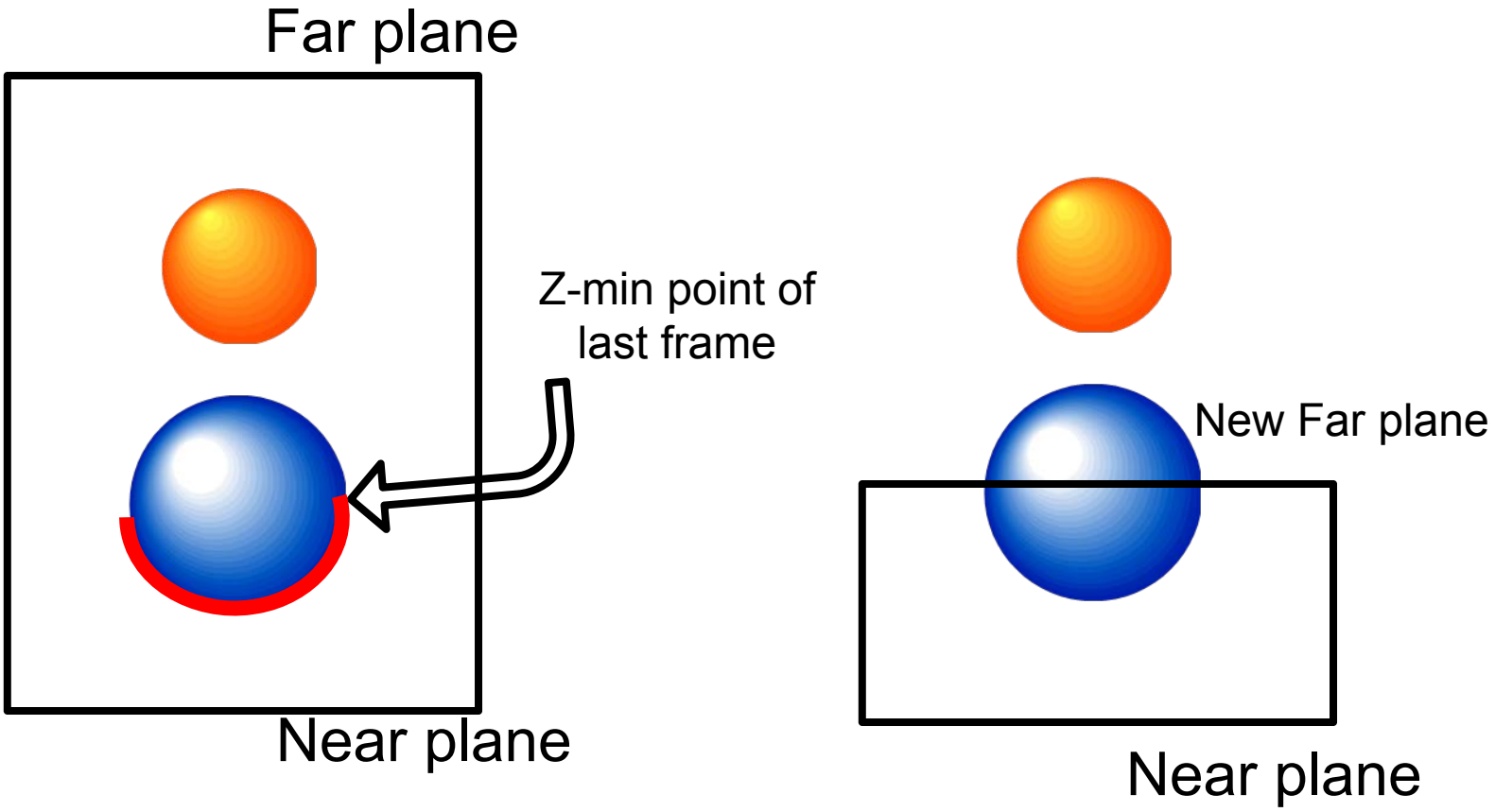
Result



KAIST

Proposed Idea

- **At Clipping Unit.**



Proposed Idea

- **There are many issues to solve.**
- **It will show improvement of performance only specified situations.**
  - **It's enough.**

Conclusion

- **Because Clipping Unit is implemented in graphics hardware and it handles all triangles, I tried to use this hardware more.**
  - **It can get additional primitive culling effect.**
- **It is need to try to cull more primitives or fragments.**
  - **Programmable or fixed functional or mixed?**
- **Programmable is always good?**
  - **PCU has not been implemented.**