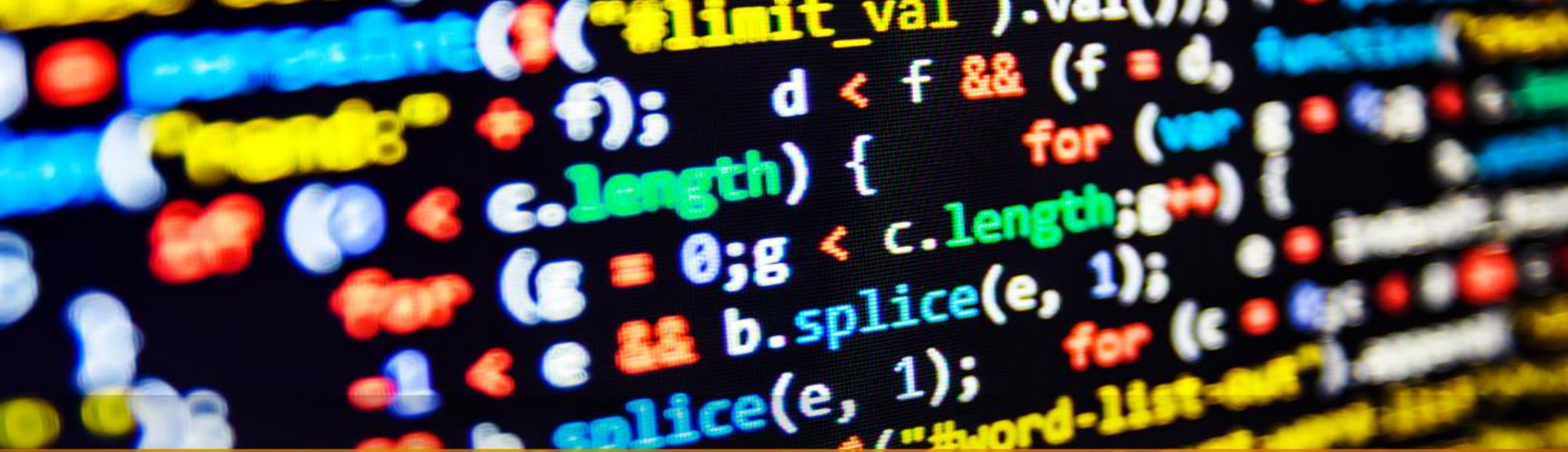
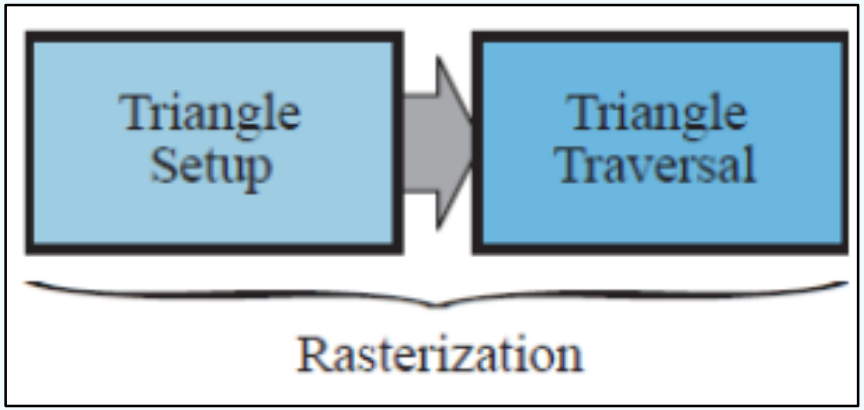


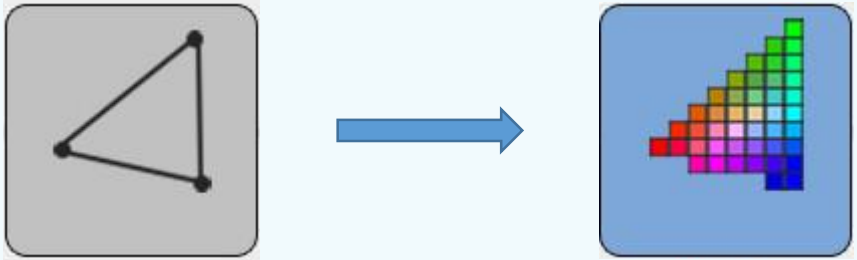
Lecture 07

The Rasterization Stage

Prepared by Ban Kar Weng (William)

Rasterization Stage





Purpose of Triangle Setup:

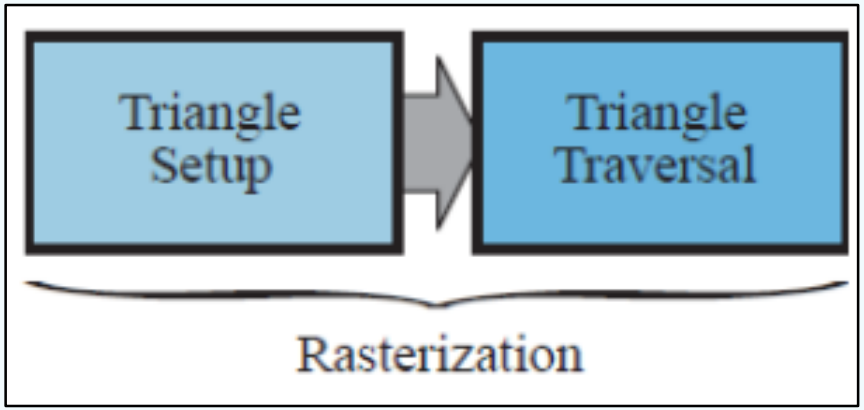
- Convert a **vertex stream** into a sequence of **base primitives**.
- Compute **factors** that are **constant** over the triangle so that triangle traversal can proceed efficiently.

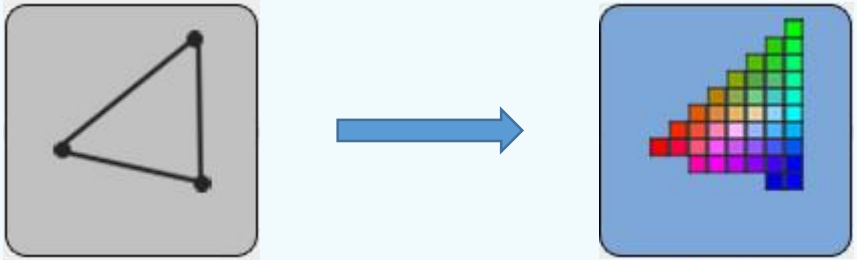
Purpose of Triangle Traversal:

- Generate a **fragment** for the part of the pixel that overlaps the triangle.
- Perform vertex attribute **interpolation**.

Except for the top image, which is obtained from Real-time Rendering (4th Edition), other images are taken online. To get the link, right click the image and select “Edit Hyperlink” in the context menu.

Rasterization Stage





Triangle rasterization takes **two steps**:

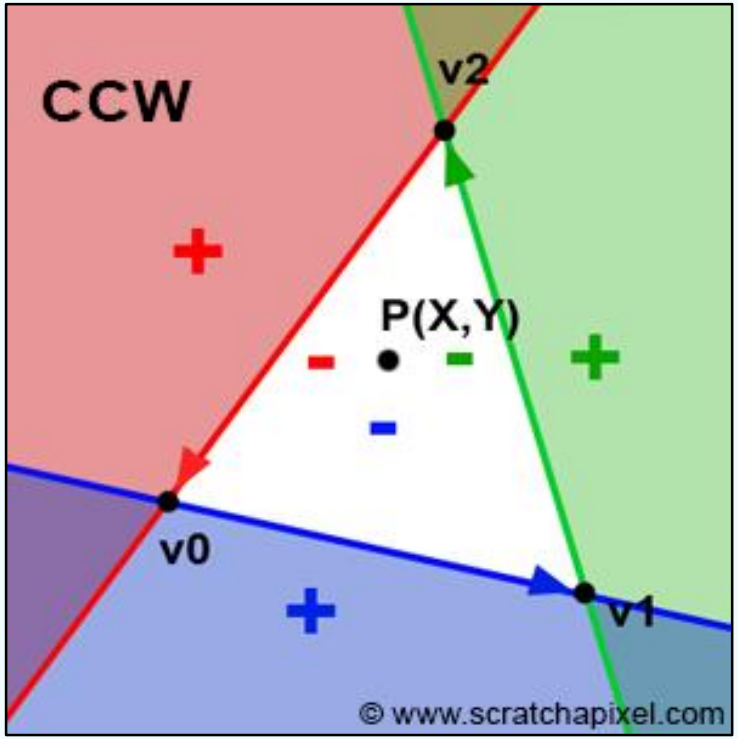
1. Find which pixels overlap the triangle
2. Define how to set the pixels' data (e.g. colour)

These can be achieved using **edge functions** and **barycentric coordinates**.

Except for the top image, which is obtained from Real-time Rendering (4th Edition), other images are taken online. To get the link, right click the image and select "Edit Hyperlink" in the context menu.

Edge Functions

Edge Functions | Definition (Part 1)



- Defined for each edge of the triangle.

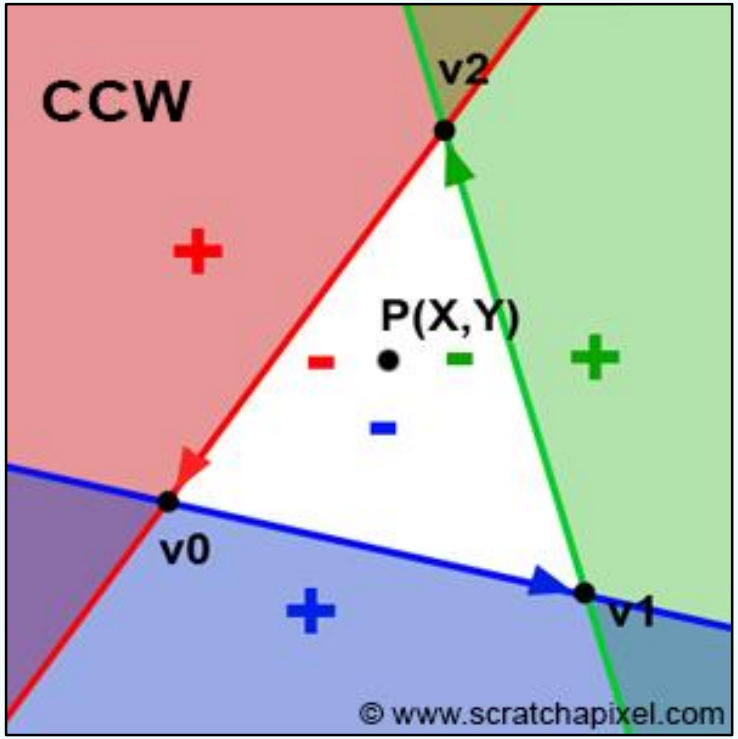
$$e_i(p) = n_i \cdot (p - p_i)$$

$$i = \{0,1,2\}$$

- **Input**: a 2D pixel coordinate p
- **Output**: a real number

All images are taken online. To get the link, right click the image and select “Edit Hyperlink” in the context menu.

Edge Functions | Definition (Part 2)



Returns:

1. **Positive number** - if input is on the right of the edge (i.e. outside triangle)
2. **Negative number** – if input is on the left of the edge (i.e. inside triangle)
3. **Zero** – if input is on the edge.

Find All Pixels Within A Triangle

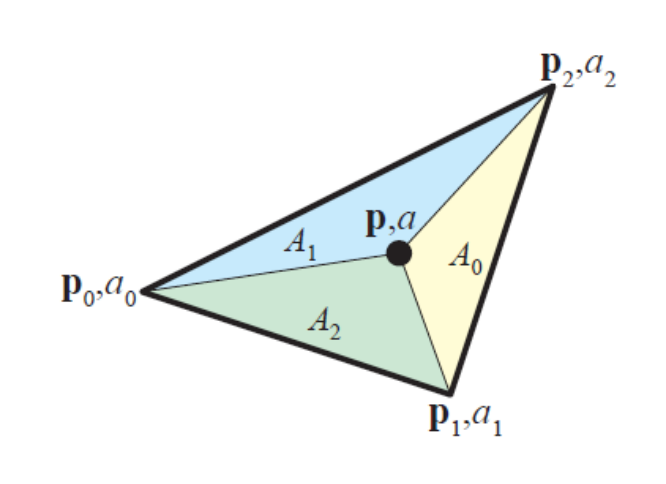
<u>One Simple Algorithm</u>:

1. Compute the bounding box that covers vertices v_0, v_1, v_2.
2. For each pixel p in the bounding box
    1. If $e_i(p) < 0$ for every i in $\{0,1,2\}$
        1. p is inside the triangle.

Note: This algorithm shows the basic idea and may not be the actual implementation in GPU.

Barycentric Coordinates

Barycentric Coordinates | Introduction

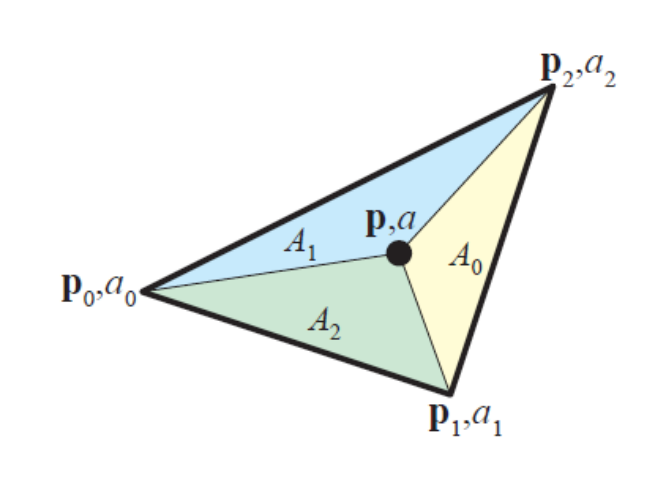


A point p on a triangle is given by the formula:

$$p = w_0 p_0 + w_1 p_1 + w_2 p_2$$

where (w_0, w_1, w_2) are the barycentric coordinates, which must fulfil $w_i \geq 0$ and $w_0 + w_1 + w_2 = 1$

Barycentric Coordinates | Introduction

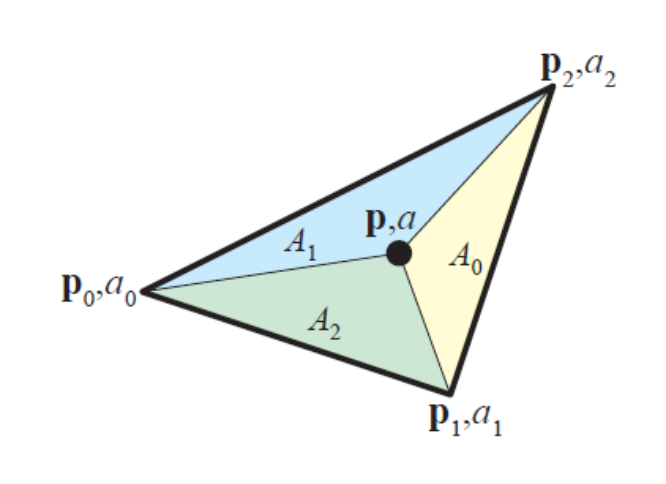


Barycentric coordinates are computed as follows:

$$w_i = \frac{A_i}{A_0 + A_1 + A_2}$$

Where A_i, the area of the sub-triangle,

Barycentric Coordinates | Introduction



A_i is the area of the sub-triangle for computing w_i

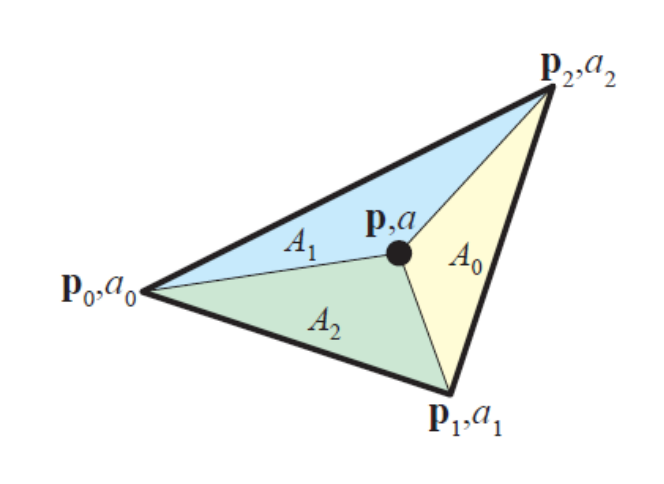
Let $q_0 = p_0 - p$, $q_1 = p_1 - p$, $q_2 = p_2 - p$

$$A_0 = \frac{1}{2}|q_1 \times q_2|$$

$$A_1 = \frac{1}{2}|q_2 \times q_0|$$

$$A_2 = \frac{1}{2}|q_0 \times q_1|$$

Barycentric Coordinates | Introduction



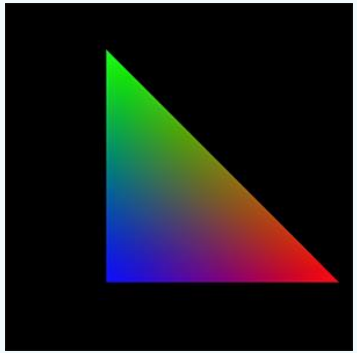
Barycentric coordinate can be used to interpolate vertex attributes

$$a = w_0 a_0 + w_1 a_1 + w_2 a_2$$

Find All Pixels Within A Triangle + Attribute Interpolation

<u>One Simple Algorithm</u>:

1. Compute the bounding box that covers vertices v_0, v_1, v_2.
2. Compute the area of the triangle A.
3. For each pixel p in the bounding box
    1. If $e_i(p) < 0$ for every i in $\{0,1,2\}$
        1. p is inside the triangle.
        2. Compute $A_0,\ A_1,\ A_2$.
        3. Compute the barycentric coordinates w_0, w_1, w_2.
        4. Compute attribute a at point p.



Note: This algorithm shows the basic idea and may not be the actual implementation in GPU.

Triangle Setup v.s. Triangle Traversal

Triangle Setup v.s. Triangle Traversal

One Simple Algorithm:

1. Compute the bounding box that covers vertices v_0, v_1, v_2.
2. Compute the area of the triangle A.
3. For each pixel p in the bounding box
    1. If $e_i(p) < 0$ for every i in $\{0,1,2\}$
        1. p is inside the triangle.
        2. Compute $A_0,\ A_1,\ A_2$.
        3. Compute the barycentric coordinates w_0, w_1, w_2.
        4. Compute attribute a at point p.

Triangle Setup

Triangle Traversal

Q & A

Acknowledgement

- This presentation has been designed using resources from PoweredTemplate.com