Derivation

Hwang Chan IL

c.hwang

5415279

1. Preparation
   1. Identify top, middle, bottom vertices

Top vertex ( xt, yt ), ( rt, gt, bt )

Middle vertex ( xm, ym ), ( rm, gm, bm )

Bottom vertex ( xb, yb ), ( rb, gb, bb )

* 1. Identify leading and trailing edges

Leading : the line that starts at xt and ends at xb

Trailing : the line that starts at xt and ends at xm

and starts at xm and ends at xb

* 1. Identify if triangle is left or right

Left : if( xm < xb )

Right : if ( xm > xb )

* 1. Identify left and right edges

(Left triangle)

Left edge = trailing edge

Right edge = leading edge

(Right triangle)

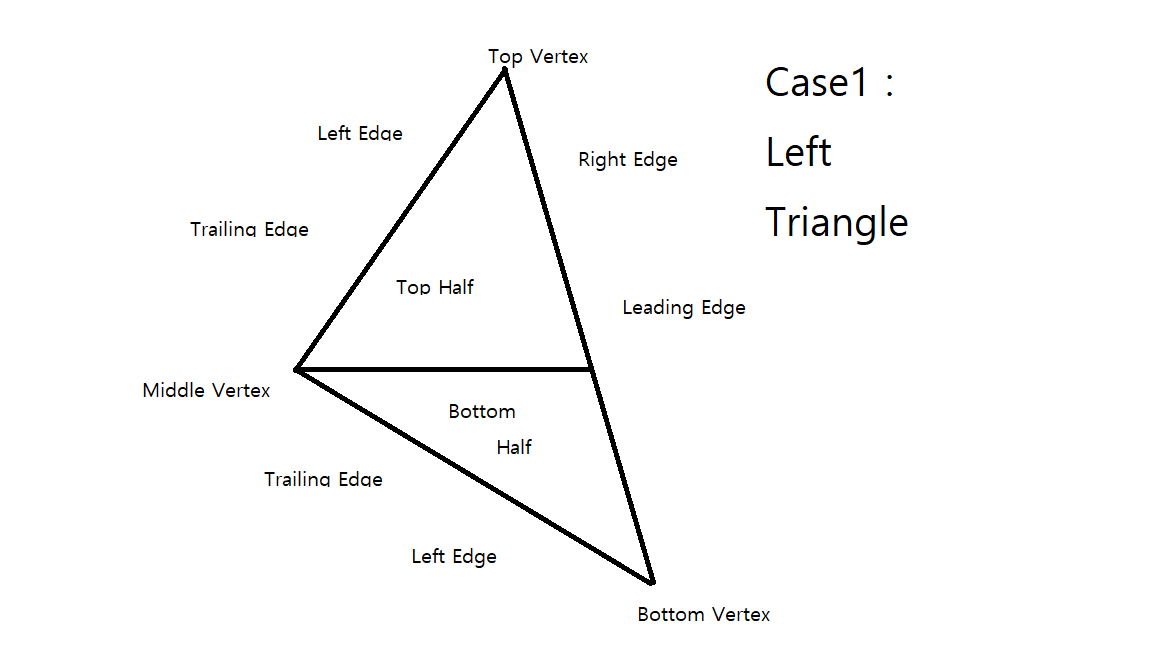
Left edge = leading edge

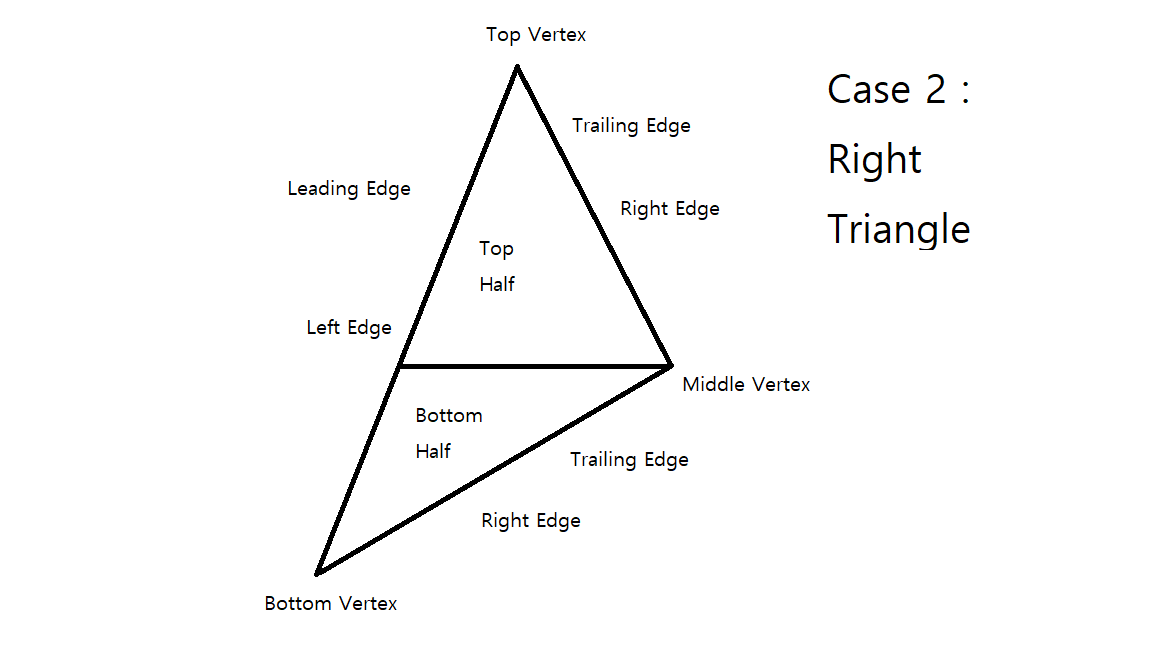
Right edge = trailing edge

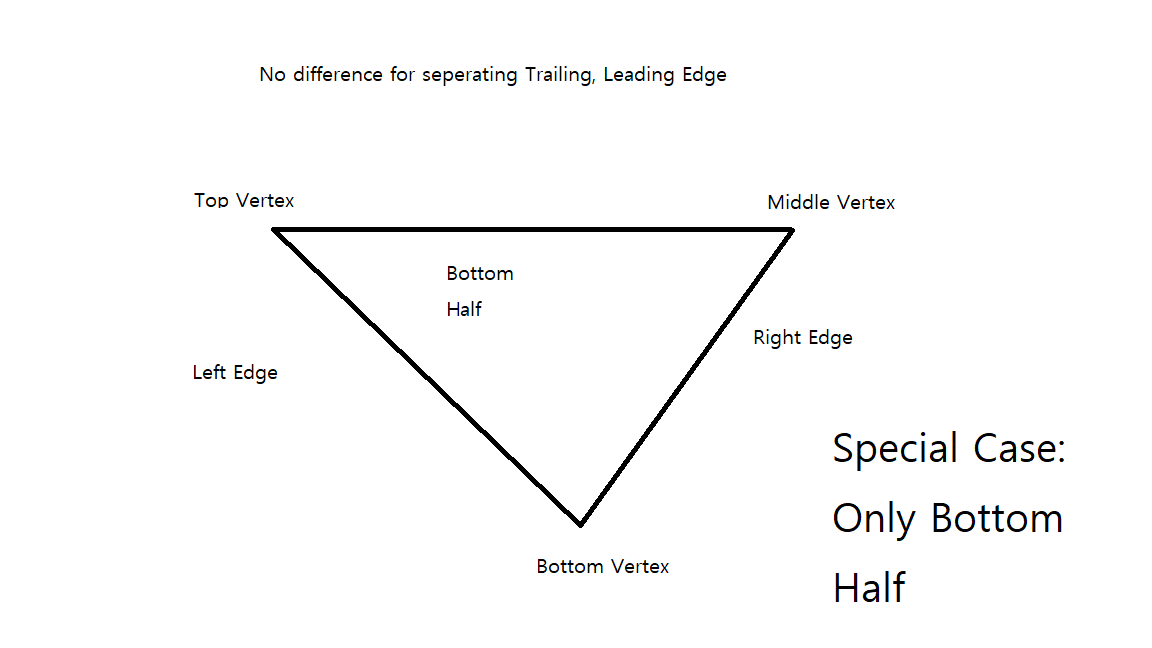
* 1. Special Cases

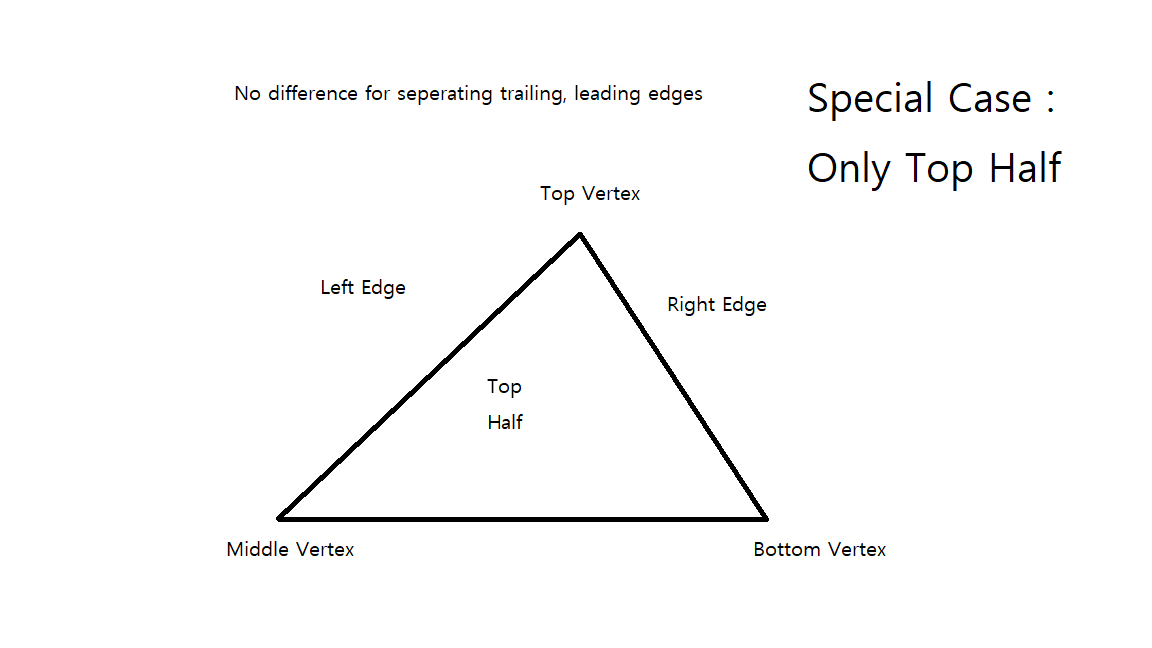
Only Top Half : if( xm == xb )

Only Bottom Half : if( xt == xm )





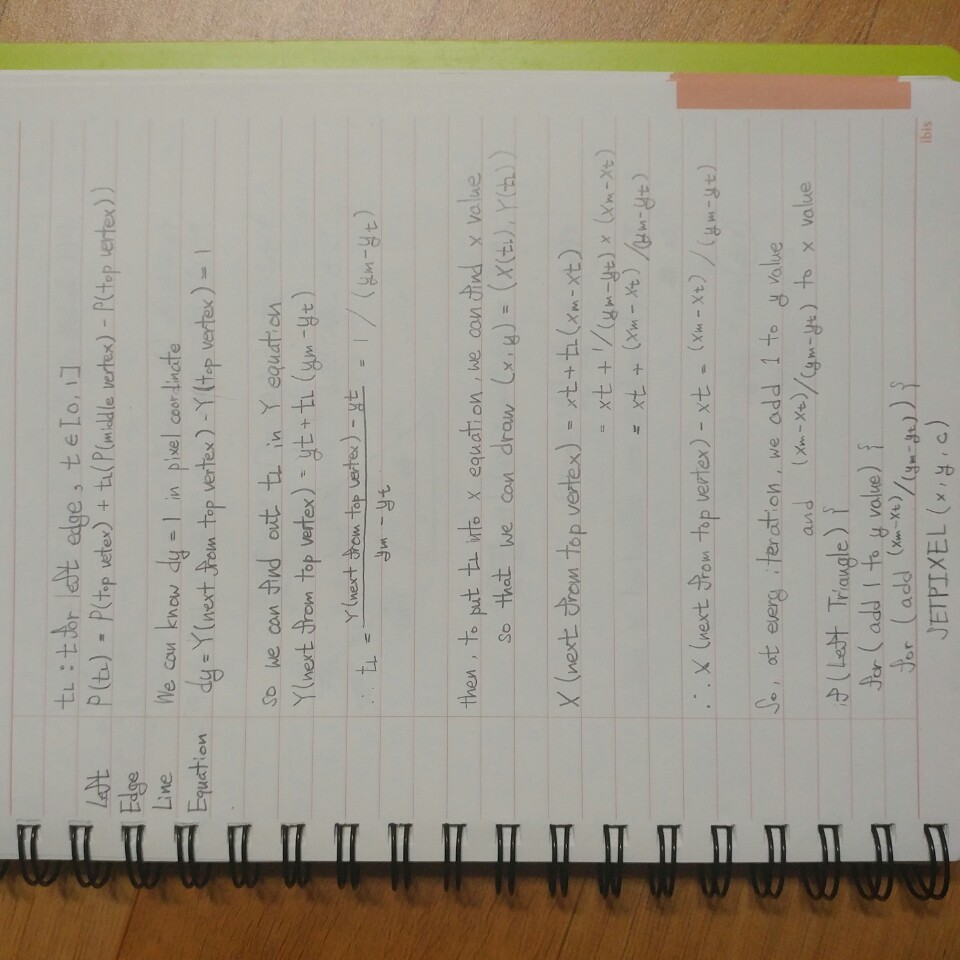




1. Edge coherence with bi-linear attribute interpolation

Assume that triangle is left triangle and it is not a special case. If triangle is right triangle, we just changed right edge and left edge.

I will use edge walking and attribute interpolation that compute attributes at each span extent.



* 1. Computing Left Edge

( TL = t for left edge, TR = t for right edge )

Left edge and right edge are different whether the triangle is right or left. ( See above pictures for the details )

So, the assumed triangle is left, left edge is trailing edge.

(If triangle is right, just switch left edge and right edge)

* + 1. **Trailing Edge**(Left edge)

For Trailing Edge, We should divide into two parts :

Top to Middle line / Middle to Bottom line

So, at first, derivate the attribute of top-middle line’s x and y values. ( And also rgb values )

Top – Middle line

This is the parametric line equation. We will use this equation to calculate attributes

P(TL)= P(Top vertex)+ (TL) \* ( P( Middle vertex ) – P( Top vertex ))

We know because y is on scan line(span). So we use equation of Y to get TL.

Y( next span ) = yt + TL \* (ym – yt)

TL = ( Y(next from top vertex) – yt )/(ym-yt)

Y(next span) – yt =1

∴TL = 1 / (ym – yt)

Then put TL into equation of any value to find out how much that value increases(attribute of that value) when y increased by one(attribute of y = 1).

X(next span) = xt + TL \* (xm – xt)

TL = 1 / (ym – yt)

∴X(next span)- xt = (xm – xt)/(ym – yt)

dX =

r(next span) = rt + TL \* ( rm – rt )

TL = 1 / (ym – yt)

∴r(next span)- rt = (rm – rt)/(ym – yt)

(Reapting same procedure…)

g(next span)- gt = (gm – gt)/(ym – yt)

b(next span)- bt = (bm – bt)/(ym – yt)

Now, we got all values we need to draw left edge, we can draw a line using that values.

X += (xm – xt)/(ym – yt)

Y += 1

R += (rm – rt) / (ym – yt)

G += (gm – gt) / (ym – yt)

B += (bm – bt) / (ym – yt)

Middle - Bottom line

If we repeat same procedure above, then

X += (xm – xb)/(ym – yb)

Y += 1

R += (rm – rb) / (ym – yb)

G += (gm – gb) / (ym – yb)

B += (bm – bb) / (ym – yb)

* + 1. **Leading Edge**(right edge)

For Leading Edge, there is only one line :

Top to Bottom line.

So we just calculate attributes using the above method only once.

X += (xb – xt)/(yb – yt)

Y += 1

R += (rb – rt) / (yb – yt)

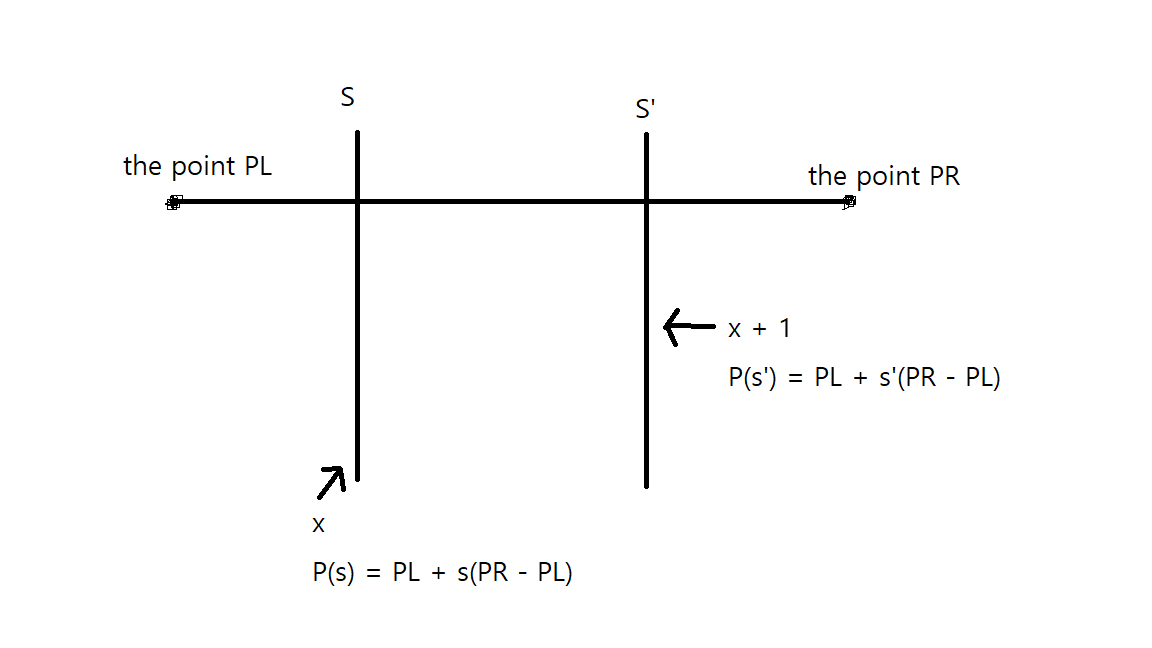
G += (gb – gt) / (yb – yt)

B += (bb – bt) / (yb – yt)

1. Span coherence

In the above case, the t parameter increased.

In this case, we will use s to compute attributes of r,g,b that is changed as the x value increases along the span.



Once again, through the above interpolations procedure, we can get these attributes.

S = (X – X(PL)) / (X(PR) – X(PL))

S’ = (X + 1 – X(PL)) / (X(PR) – X(PL))

S’ - S = 1 / (X(PR) – X(PL))

So,

X += 1

R += (R(PL) –R(PR)) / (X(PL) – X(PR))

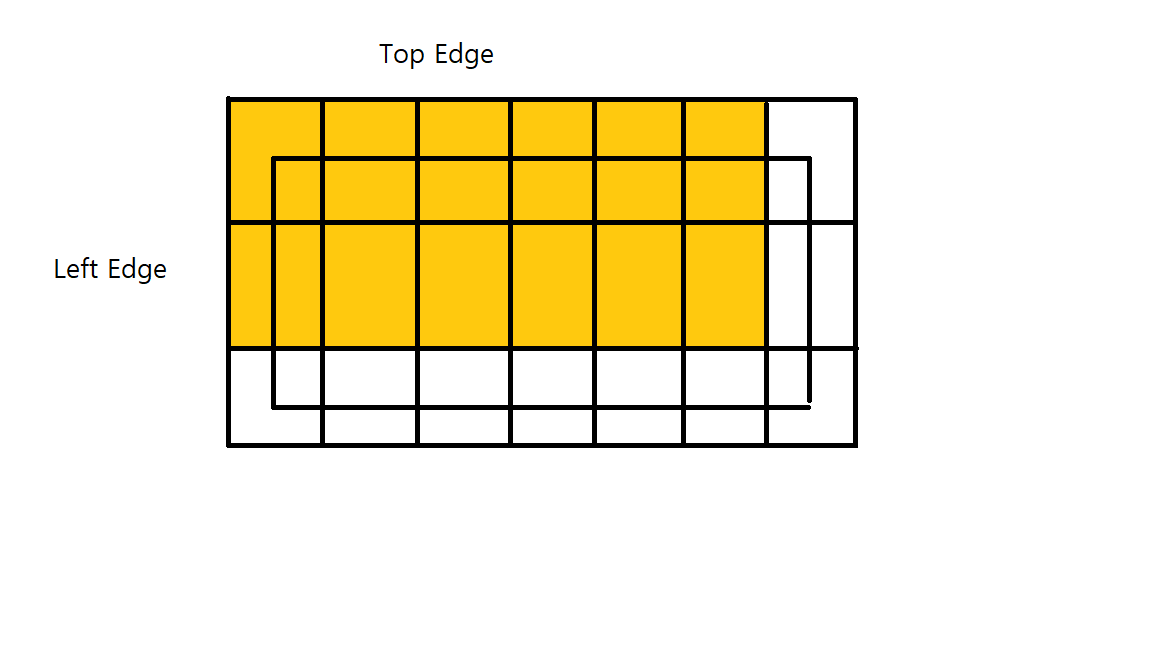
G += (G(PL) –G(PR)) / (X(PL) – X(PR))

B += (B(PL) –B(PR)) / (X(PL) – X(PR))

1. Tie-breaking rule

To distinguish which pixels of triangles are drawn when the triangles have shared edge, we have to use tie-breaking rule.

If the rule is top-left, then the pixels lies exactly on the left and top edges of triangles are drawn.



1. Special cases
   1. Vertices are not in the desired order.

If the triangle’s vertices are in clockwise order, this triangle should not be drawn.

To check this case, we use cross product between

<xm – xt, ym – yt, 0> X <xb – xt, yb – yt, 0>

Then the z of the vecter product is positive, it means this triangle is on counter - clockwise order. Otherwise, this triangle is on clockwise order.

* 1. Degenerate triangles

If xt, xm, xb are all same, or If yt, ym, yb are all same,

We call it degenerate triangles.

I decided not to draw in this case.