Faculty of Informatics, USI
Autumn term 22

Arnaud Fauconnet & Francesco Costa
Prof. Piotr Didyk

Computer Graphics
Assignement Set 7

Exercise 1 [7 points]

To get the projected point p_1', p_2', we can use a transformation matrix T similar to the one found in exercise 4 of assignment 4.

In our case, we want to project the point on the plane $z = 1$, therefore our matrix would be

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

So

$$p_1' = T \cdot p_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ -1 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} -0.5 \\ -0.5 \\ 1 \\ 1 \end{pmatrix} = (-0.5, -0.5, 1)$$

$$p_2' = T \cdot p_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ 1 \\ 10 \\ 1 \end{pmatrix} = \begin{pmatrix} 7 \\ 1 \\ 10 \\ 10 \end{pmatrix} = \begin{pmatrix} 0.7 \\ 0.1 \\ 1 \\ 1 \end{pmatrix} = (0.7, 0.1, 1)$$

The size of the single pixel is

$$s = \frac{\tan^{-1}\left(\frac{\pi}{4}\right)}{5} = \frac{1}{5}$$

The indices i_1, i_2 which represent the index of p_1' and p_2' respectively, can be calculated by expressing the point relative to the top-left corner of the image and then dividing the coordinates by s. We then take the absolute value of the coordinates, since indices are positive.

$$p_1'' = (-0.5, -0.5) - (-1, 1) = (0.5, -1.5)$$
$$i_1 = \text{ceil}(p_1'' \cdot 5) = (3, 8)$$

$$p_2'' = (0.7, 0.1) - (-1, 1) = (1.7, -0.9)$$
$$i_2 = \text{ceil}(p_2'' \cdot 5) = (9, 5)$$

We define the implicit line equation as

$$\begin{aligned} F(x,y) &= ydx - xdy + x_1dy - y_1dx = \\ &= 6y - (-3)x + 3\cdot(-3) - 8\cdot 6 = \\ &= 6y + 3x - 57 \end{aligned}$$

where

$$dx = x_2 - x_1 = 6, \qquad dy = y_2 - y_1 = -3$$

To decide which pixel to color, we look at the sign of $F(M)$, where M is the midpoint between the two possible pixels.

We repeat this process for all six pixels separating p_1' and p_2'

$$F(4, 7.5) = 0$$

We thus color the pixel $(4, 7)$

$$F(5, 6.5) = -3 < 0$$

We thus color the pixel $(5, 7)$

$$F(6, 6.5) = 0$$

We thus color the pixel $(6, 6)$

$$F(7, 5.5) = -3 < 0$$

We thus color the pixel $(7, 6)$

$$F(8, 5.5) = 0$$

We thus color the pixel $(8, 5)$

Finally, we color $(9, 5)$ because it is p_2'.

- The z-value of the horizontal midpoint $(6, 6)$ has vertical barycentric coordinate $\lambda = \frac{1}{2}$ (formula taken from slide 9 of "11 - Perspective Interpolation")

$$\begin{aligned} p^z &= \frac{1}{(1-\lambda)\frac{1}{p_1^z} + \lambda\frac{1}{p_2^z}} \\ &= \frac{1}{(1-\frac{1}{2})\frac{1}{2} + \frac{1}{2}\frac{1}{10}} \\ &= \frac{1}{\frac{6}{20}} \\ &= \frac{10}{3} \end{aligned}$$

- The color of the horizontal midpoint can be calculated with the attribute formula (formula taken from slide 10 of "11 - Perspective Interpolation")

$$A = \left((1-\lambda)\frac{A_1}{p_1^z} + \lambda\frac{A_2}{p_2^z}\right)\cdot p^z$$

where A is the attribute (in our case the color of the pixel) of p. Thus

$$\begin{aligned}
A &= \left(\left(1-\frac{1}{2}\right)\cdot\begin{pmatrix}1\\0\\0\end{pmatrix}\cdot\frac{1}{2}+\frac{1}{2}\cdot\begin{pmatrix}0\\1\\0\end{pmatrix}\cdot\frac{1}{10}\right)\cdot\frac{10}{3} = \\
&= \left(\begin{pmatrix}\frac{1}{4}\\0\\0\end{pmatrix}+\begin{pmatrix}0\\\frac{1}{20}\\0\end{pmatrix}\right)\cdot\frac{10}{3} \\
&= \begin{pmatrix}1/4\\1/20\\0\end{pmatrix}\cdot\frac{10}{3} \\
&= \begin{pmatrix}5/6\\1/6\\0\end{pmatrix}
\end{aligned}$$

Exercise 2 [8 points]

We remind the definition of barycentric coordinates and it's relation to the areas of the subtriangles. We have that

$$\lambda_i = \frac{w_i}{A}$$

where w_i is the signed area of the subtriangle opposite to vertex p_i and A the total area of the triangle.

We can now express the signed areas w_i as a function of $p = (x, y)$

$$\begin{aligned}
2\cdot w_1(x,y) &= \|(p_2-p)\times(p_3-p)\| \\
&= \det\begin{pmatrix}p_{2x}-x & p_{3x}-x\\ p_{2y}-y & p_{3y}-y\end{pmatrix} \\
&= (p_{2x}-x)(p_{3y}-y)-(p_{2y}-y)(p_{3x}-x) \\
&= p_{2x}p_{3y}-yp_{2x}-xp_{3y}+xy-p_{3x}p_{2y}+yp_{3x}+xp_{2y}-xy \\
&= x(p_{2y}-p_{3y})+y(p_{3x}-p_{2x})+p_{2x}p_{3y}-p_{3x}p_{2y}
\end{aligned}$$

We have therefore the final formula

$$w_1(x,y) = \frac{1}{2}\big(x(p_{2y}-p_{3y})+y(p_{3x}-p_{2x})+p_{2x}p_{3y}-p_{3x}p_{2y}\big)$$

The formulas for w_2 and w_3 can easily be found if we look at the indexes present in the formula for w_1 since

$$w_i(x,y) = \|(p_{i+1}-p)\times(p_{i-1}-p)\|$$

hence

$$\begin{aligned}
w_2(x,y) &= \frac{1}{2}\big(x(p_{3y}-p_{1y})+y(p_{1x}-p_{3x})+p_{3x}p_{1y}-p_{1x}p_{3y}\big) \\
w_3(x,y) &= \frac{1}{2}\big(x(p_{1y}-p_{2y})+y(p_{2x}-p_{1x})+p_{1x}p_{2y}-p_{2x}p_{1y}\big)
\end{aligned}$$

If we want to find the barycentric coordinates of point $(x+1, y)$, we see that

$$\lambda_i(x+1,y) = \frac{w_i(x+1,y)}{A}$$

We notice easily that

$$\begin{aligned} w_1(x+1,y) &= \frac{1}{2}\big((x+1)(p_{2y}-p_{3y}) + y(p_{3x}-p_{2x}) + p_{2x}p_{3y} - p_{3x}p_{2y}\big) \\ &= w_1(x,y) + \frac{1}{2}(p_{2y}-p_{3y}) \end{aligned}$$

and similarly

$$w_2(x+1,y) = w_2(x,y) + \frac{1}{2}(p_{3y}-p_{1y})$$

$$w_3(x+1,y) = w_3(x,y) + \frac{1}{2}(p_{1y}-p_{2y})$$

Finally, the barycentric coordinates of point $(x+1,y)$ are

$$\lambda_i(x+1,y) = \lambda_i(x,y) + \frac{1}{2A}\left(p_{(i+1)y} - p_{(i-1)y}\right)$$

which is pretty good since we already computed $\lambda_i(x,y)$

The same reasoning can be applied to find the barycentric coordinates of point $(x, y+1)$, since

$$\begin{aligned} w_1(x,y+1) &= \frac{1}{2}\big(x(p_{2y}-p_{3y}) + (y+1)(p_{3x}-p_{2x}) + p_{2x}p_{3y} - p_{3x}p_{2y}\big) \\ &= w_1(x,y) + \frac{1}{2}(p_{3x}-p_{2x}) \end{aligned}$$

and similarly

$$w_2(x,y+1) = w_2(x,y) + \frac{1}{2}(p_{1x}-p_{3x})$$

$$w_3(x,y+1) = w_3(x,y) + \frac{1}{2}(p_{2x}-p_{1x})$$

So, the barycentric coordinates of point $(x, y+1)$ are

$$\lambda_i(x,y+1) = \lambda_i(x,y) + \frac{1}{2A}\left(p_{(i-1)x} - p_{(i+1)x}\right)$$

which is pretty good since we already computed $\lambda_i(x,y)$

```
function BoundingBox(p1, p2, p3)
    x_min = min(p1.x, p2.x, p3.x)
    y_min = min(p1.y, p2.y, p3.y)
    x_max = max(p1.x, p2.x, p3.x)
    y_max = max(p1.y, p2.y, p3.y)
    return Point(x_min, y_min), Point(x_max, y_max)
end


function RasterizeTriangle(p1, p2, p3)
    p_min, p_max = BoundingBox(p1, p2, p3)
    A = Area(p1, p2, p3)

    triangle_pixels = []
```

```
    // start with pixel (x_min - 1, y_min - 1)
    init_lambda_1, init_lambda_2, init_lambda_3 = Baryncentric(
        p_min.x - 1,
        p_min.y - 1,
        p1, p2, p3
    )

    for y = p_min.y to p_max.y
        // from (x, y) -> (x, y + 1)
        lambda_1 = init_lambda_1 + 1/(2*A)* (p3.x - p2.x)
        lambda_2 = init_lambda_2 + 1/(2*A)* (p1.x - p3.x)
        lambda_3 = init_lambda_3 + 1/(2*A)* (p2.x - p1.x)

        // save the coordinates of pixel (x_min - 1, y)
        init_lambda_1 = lambda_1
        init_lambda_2 = lambda_2
        init_lambda_3 = lambda_3

        for x = p_min.x to p_max.x
            // from (x, y) -> (x + 1, y)
            lambda_1 = lambda_1 + 1/(2*A)* (p2.y - p3.y)
            lambda_2 = lambda_2 + 1/(2*A)* (p3.y - p1.y)
            lambda_3 = lambda_3 + 1/(2*A)* (p1.y - p2.y)

            // save the pixel if it is inside the triangle
            if All_Non_Negative(lambda_1, lambda_2, lambda_3)
                triangle_pixels.push( (x, y) )
            end
        end
    end

    return triangle_pixels
end
```