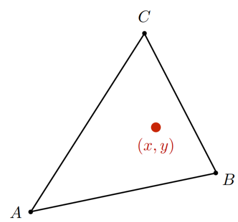
Interpolation

Steve Canves

yqykrhf@163.com

1 Interpolation across triangle – Barycentric Coordinates

Barycentric Coordinates uses (α, β, γ) to describe the point $(\mathbf{x}, \mathbf{y})$ inside the trianlge.



Barycentric Coordinates's mathematic expression is as follows:

$$(x, y) = \alpha A + \beta B + \gamma C$$
$$s.t. \quad \alpha + \beta + \gamma = 1$$

Point is inside the triangle if all three coordinates are non-negative and less than 1

There are three approaches to derive the α, β, γ

- solving equation
- signed distance function
- Geometric viewpoint – proportional area

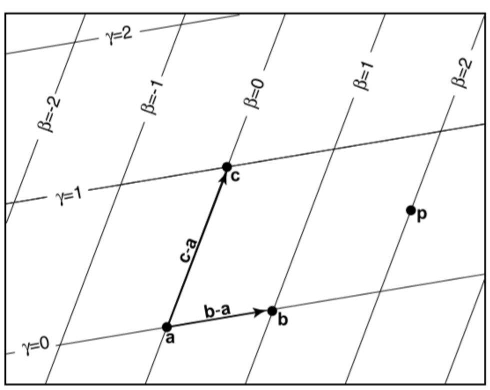


Figure 1: non-orthogonal coordinate

As shown in Figure 1, a point p can be written as

$$p = a + \beta(b - a) + \gamma(c - a)$$

Then solve the equation:

$$\begin{bmatrix} x_b - x_a & x_c - x_a \\ y_b - y_a & y_c - y_a \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} x_p - x_a \\ y_p - y_a \end{bmatrix} \tag{1}$$

Signed Distance Function

One geometric property of barycentric coordinates is that they are the signed scaled distance from the lines through the triangle sides, as is shown for β in Figure 2

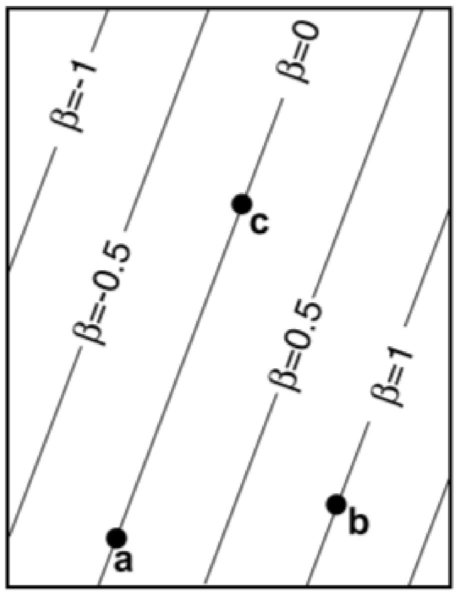


Figure 2: β is the signed distance scaled distance from the line through **a** and **c**

Also recall that if f(x, y)= 0 is the equation for a particular line, so is kf(x, y)= 0 for any non-zero k. Changing k scales the distance and controls which side of the line has positive signed distance, and which negative. We would like to choose k such that, for example, $kf(x, y) = \beta$. Since k is only one unknown, we can force this with one constraint, namely that at point **b** we know $\beta = 1$

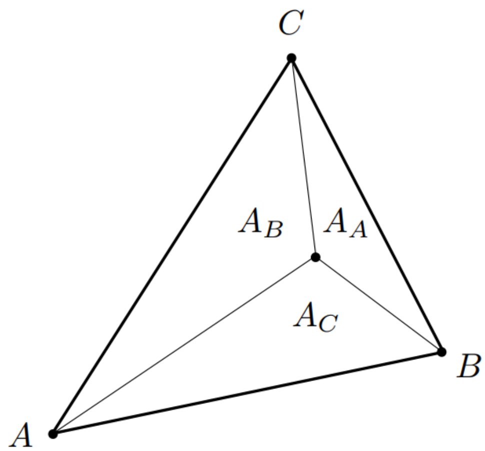
$$kf_{ac}(x_b, y_b) = 1 \tag{2}$$

For any points, they admit $kf_{ac}(x, y) = \beta$ Substituing it to Equation 2, we will get β

$$\beta = \frac{f_{ac}(x, y)}{f_{ac}(x_b, y_b)} \tag{3}$$

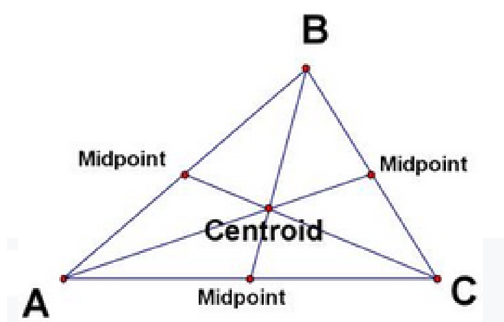
Equation $f_{ac}(x, y)$ can be simply written:

$$f_{ac}(x, y) = (y_c - y_a)x + (x_a - x_c)y + x_c y_a - x_a y_c = 0 \tag{4}$$



Geometric viewpoint

$$\alpha = \frac{A_A}{A_A + A_B + A_C}$$
$$\beta = \frac{A_B}{A_A + A_B + A_C}$$
$$\gamma = \frac{A_C}{A_A + A_B + A_C}$$



V_A can be positions, texture coordiantes, color, nomal, depth, matertial attributes...

$$V = \alpha V_A + \beta V_B + \gamma V_C$$

However, barycentric coordinates are not invariant under projection

The barycentric coordinates in view space is different from in screen space. If interpolating in screen space, perspective-correct interpolation is essential.

Referring to Figure 3, by similar triangle, we have

$$\frac{X_1}{Z_1} = \frac{u_1}{d} \Rightarrow X_1 = \frac{u_1 Z_1}{d} \tag{5}$$

$$\frac{X_2}{Z_2} = \frac{u_2}{d} \Rightarrow X_2 = \frac{u_2 Z_2}{d} \tag{6}$$

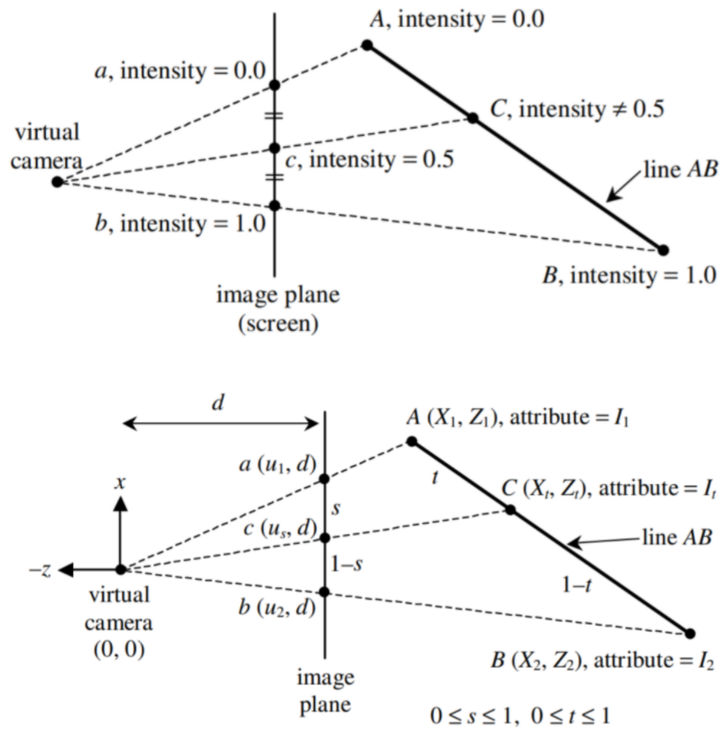


Figure 3: The virtual camera is looking in the +z direction in the camera coordinate system.

$$\frac{X_t}{Z_t} = \frac{u_s}{d} \Rightarrow Z_t = \frac{dX_t}{u_s} \tag{7}$$

By linearly interpolating in the image plane(or screen space), we have:

$$\frac{X_t}{Z_t} = \frac{u_s}{d} \Rightarrow Z_t = \frac{dX_t}{u_s} \tag{8}$$

Similiarly,

$$X_t = X_1 + t\left(X_2 - X_1\right) \tag{9}$$

$$Z_t = Z_1 + t\left(Z_2 - Z_1\right) \tag{10}$$

Substituing (4) and (5) into (3),

$$Z_t = \frac{d\left(X_1 + t\left(X_2 - X_1\right)\right)}{u_1 + s\left(u_2 - u_1\right)} \tag{11}$$

Substituing (1) and (2) into (7),

$$\begin{aligned} Z_t &= \frac{d\left(\frac{u_1 Z_1}{d} + t\left(\frac{u_2 Z_2}{d} - \frac{u_1 Z_1}{d}\right)\right)}{u_1 + s\left(u_2 - u_1\right)} \\ &= \frac{u_1 Z_1 + t\left(u_2 Z_2 - u_1 Z_1\right)}{u_1 + s\left(u_2 - u_1\right)} \end{aligned} \tag{12}$$

Substituing (6) into (8),

$$Z_1 + t\left(Z_2 - Z_1\right) = \frac{u_1 Z_1 + t\left(u_2 Z_2 - u_1 Z_1\right)}{u_1 + s\left(u_2 - u_1\right)} \tag{13}$$

which can be simplified into

$$t = \frac{sZ_1}{sZ_1 + (1-s)Z_2} \tag{14}$$

Substituing (10) into (6), we have

$$Z_t = Z_1 + \frac{sZ_1}{sZ_1 + (1-s)Z_2}(Z_2 - Z_1) \tag{15}$$

which can be simplified into

$$Z_t = \frac{1}{\frac{1}{Z_1} + s\left(\frac{1}{Z_2} - \frac{1}{Z_1}\right)} \tag{16}$$

Equation (12) tells us that the z -value at point c in the image plane can be correctly derived by just linearly interpolating between $1/Z_1$ and $1/Z_2$, and then compute the reciprocal of the interpolated result. For z -buffer purpose, the final reciprocal need not even be computed, because all we need is to reverse the comparison operation during z-value comparison.

And then derive formula to correctly interpolate,

$$I_t = I_1 + t(I_2 - I_1) \tag{17}$$

Substituing (10) into (13), we have

$$I_t = I_1 + \frac{sZ_1}{sZ_1 + (1-s)Z_2}(I_2 - I_1) \tag{18}$$

which can be rearranged into

$$I_t = \left(\frac{I_1}{Z_1} + s\left(\frac{I_2}{Z_2} - \frac{I_1}{Z_1}\right)\right) / \left(\frac{1}{Z_1} + s\left(\frac{1}{Z_2} - \frac{1}{Z_1}\right)\right) \tag{19}$$

$$I_t = \left(\frac{I_1}{Z_1} + s\left(\frac{I_2}{Z_2} - \frac{I_1}{Z_1}\right)\right) / \frac{1}{Z_t} \tag{20}$$

2 Linear Interpolation

Loot at Figure 4, Given two known points by the coordiantes (x_0, y_0), (x_1, y_1), For a value x in the interval (x_0, x_1), the value y along the straight line is given from the equation of slopes:

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$$

Solving this equation for y, which is the unknown value at x, gives

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0}$$

Rethink a way easy to understand called **weighted average**, The closer point has more influence than the farther point, which causes the it's weight is larger. At the same time, the weights should be sumed to 1. so the formula is derived as follows:

$$\begin{aligned}
y &= y_0\left(1 - \frac{x - x_0}{x_1 - x_0}\right) + y_1\left(1 - \frac{x_1 - x}{x_1 - x_0}\right) \\
&= y_0\left(1 - \frac{x - x_0}{x_1 - x_0}\right) + y_1\left(\frac{x - x_0}{x_1 - x_0}\right) \\
&= y_0\frac{x_1 - x}{x_1 - x_0} + y_1\frac{x - x_0}{x_1 - x_0}
\end{aligned}$$

The weight approach is essential for understanding other derived linear interpolation.

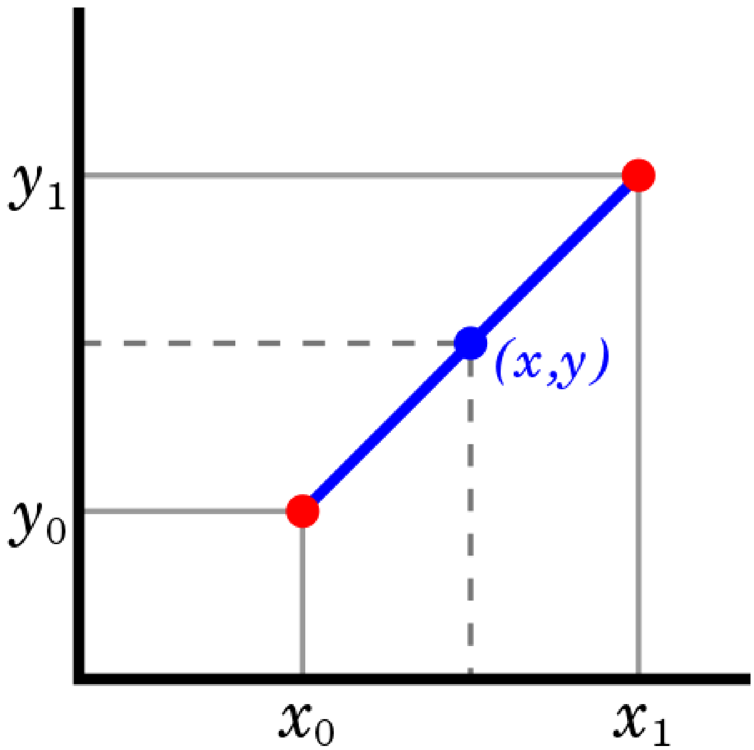


Figure 4: Given the two red points, the blue line is the linear interpolant between the points, and the value y at x may be found by linear interpolation

2.1 Bilinear Interpolation

Bilinear interpolation is performed using linear interpolation first in one direction, and then again in the other direction. Although each step is linear in the sampled values and in the position, the interpolation as a whole is not linear but rather quadratic in the sample location.

Firstly we do liner interpolation in the x-direction. This yields:

$$R_1 --f\left(x, y_1\right) \approx \frac{x_2 - x}{x_2 - x_1} f\left(Q_{11}\right) + \frac{x - x_1}{x_2 - x_1} f\left(Q_{21}\right)$$
$$R_2 --f\left(x, y_2\right) \approx \frac{x_2 - x}{x_2 - x_1} f\left(Q_{12}\right) + \frac{x - x_1}{x_2 - x_1} f\left(Q_{22}\right)$$

We proceed by interpolating in the y-direction to obtain the desired estimate:

$$\begin{aligned} f(x,y) &\approx \frac{y_2 - y}{y_2 - y_1} f\left(x, y_1\right) + \frac{y - y_1}{y_2 - y_1} f\left(x, y_2\right) \\ &= \frac{y_2 - y}{y_2 - y_1}\left(\frac{x_2 - x}{x_2 - x_1} f\left(Q_{11}\right) + \frac{x - x_1}{x_2 - x_1} f\left(Q_{21}\right)\right) + \frac{y - y_1}{y_2 - y_1}\left(\frac{x_2 - x}{x_2 - x_1} f\left(Q_{12}\right) + \frac{x - x_1}{x_2 - x_1} f\left(Q_{22}\right)\right) \\ &= \frac{1}{\left(x_2 - x_1\right)\left(y_2 - y_1\right)}\left[x_2 - x \quad x - x_1\right]\begin{bmatrix} f\left(Q_{11}\right) & f\left(Q_{12}\right) \\ f\left(Q_{21}\right) & f\left(Q_{22}\right) \end{bmatrix}\begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix} \end{aligned} \tag{21}$$

Note that we will arrive at the same result if the interpolation is done first along the y direction and then along the x direction.

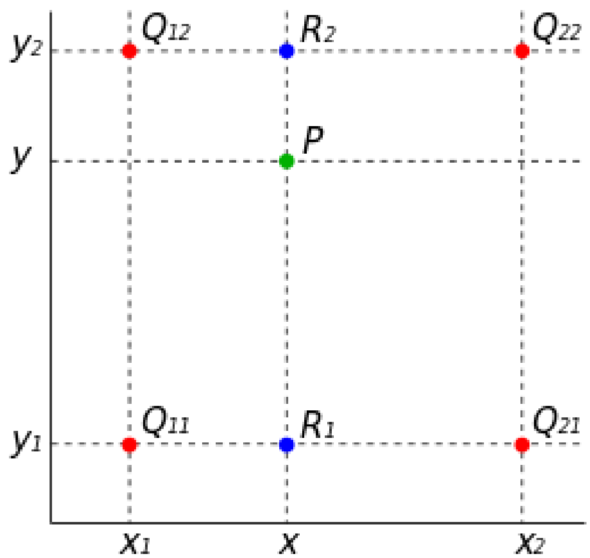


Figure 5: The four red dots show the data points and the green dot is the point at which we want to interpolate.

2.1.1 Alternative algorithm

An alternative way to write the solution to the interpolation problem is

$$f(x,y) \approx a_0 + a_1 x + a_2 y + a_3 xy$$

where the coefficients are found by solving the linear system

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_1 & y_2 & x_1 y_2 \\ 1 & x_2 & y_1 & x_2 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f(Q_{11}) \\ f(Q_{12}) \\ f(Q_{21}) \\ f(Q_{22}) \end{bmatrix}$$

yielding the result

$$\begin{aligned}
a_0 &= \frac{f(Q_{11})\, x_2 y_2}{(x_1 - x_2)(y_1 - y_2)} + \frac{f(Q_{12})\, x_2 y_1}{(x_1 - x_2)(y_2 - y_1)} + \frac{f(Q_{21})\, x_1 y_2}{(x_1 - x_2)(y_2 - y_1)} + \frac{f(Q_{22})\, x_1 y_1}{(x_1 - x_2)(y_1 - y_2)} \\
a_1 &= \frac{f(Q_{11})\, y_2}{(x_1 - x_2)(y_2 - y_1)} + \frac{f(Q_{12})\, y_1}{(x_1 - x_2)(y_1 - y_2)} + \frac{f(Q_{21})\, y_2}{(x_1 - x_2)(y_1 - y_2)} + \frac{f(Q_{22})\, y_1}{(x_1 - x_2)(y_2 - y_1)} \\
a_2 &= \frac{f(Q_{11})\, x_2}{(x_1 - x_2)(y_2 - y_1)} + \frac{f(Q_{12})\, x_2}{(x_1 - x_2)(y_1 - y_2)} + \frac{f(Q_{21})\, x_1}{(x_1 - x_2)(y_1 - y_2)} + \frac{f(Q_{22})\, x_1}{(x_1 - x_2)(y_2 - y_1)} \\
a_3 &= \frac{f(Q_{11})}{(x_1 - x_2)(y_1 - y_2)} + \frac{f(Q_{12})}{(x_1 - x_2)(y_2 - y_1)} + \frac{f(Q_{21})}{(x_1 - x_2)(y_2 - y_1)} + \frac{f(Q_{22})}{(x_1 - x_2)(y_1 - y_2)}
\end{aligned}$$

If a solution is preferred in terms of f(Q), then we can write

$$f(x,y) \approx b_{11} f(Q_{11}) + b_{12} f(Q_{12}) + b_{21} f(Q_{21}) + b_{22} f(Q_{22})$$

where the coefficients are found by solving

$$\begin{bmatrix} b_{11} \\ b_{12} \\ b_{21} \\ b_{22} \end{bmatrix} = \left(\begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_1 & y_2 & x_1 y_2 \\ 1 & x_2 & y_1 & x_2 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \end{bmatrix}^{-1} \right)^{\mathrm{T}} \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix}$$

2.1.2 Geometric Visualisation

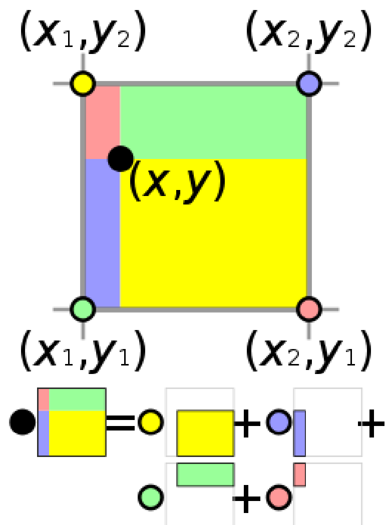


Figure 6: The product of the value at the desired point (black) and the entire area is equal to the sum of the products of the value at each corner and the partial area diagonally opposite the corner (corresponding colours).

From this viewpoint, we could write the Equation (1) down mentioned in the front of Chapter 2.1

2.2 Cubic Interpolation

her

2.2.1 Bicubic Interpolation

The interpolated surface by bicubic interpolation is smoother and have fewer artifaces than corresponding surfaces obtained by bilinear interpolation with the cost of speed. In contrast to bilinear interpolation, which only takes 4 pixels (2×2) in to account, bicubic interpolation considers 16 pixels (4×4).

3 Reference

There some references

- perspective correction
- Perspective-Correct interpolation Kok-Lim low