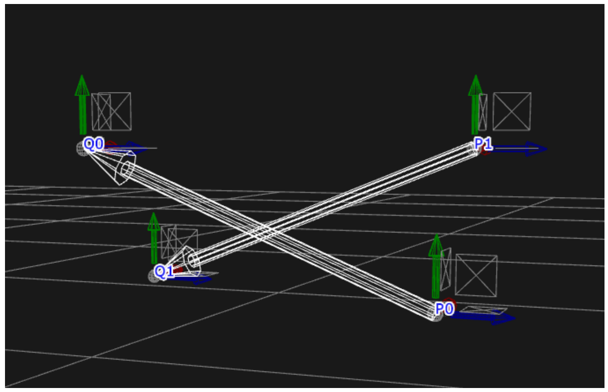
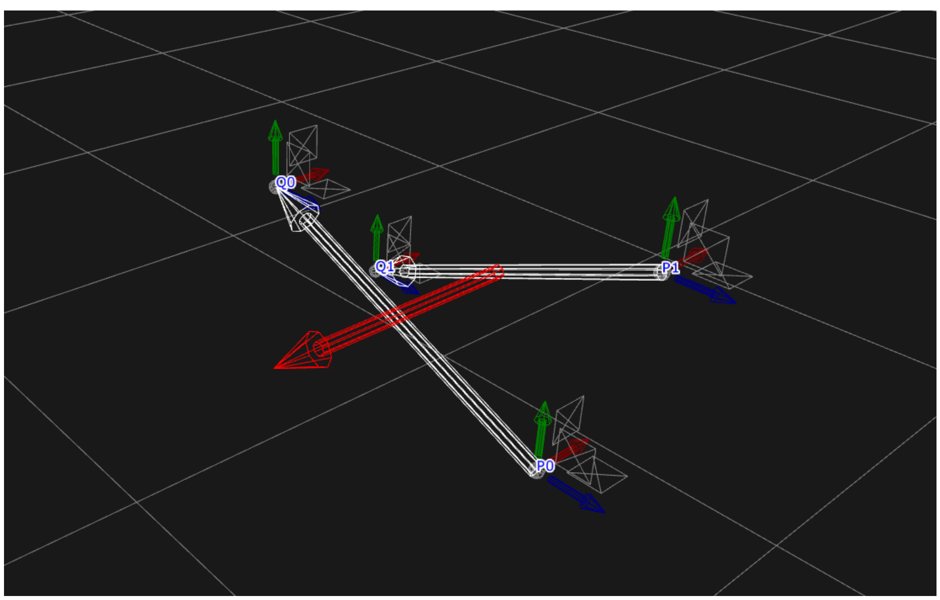
1. Line definition



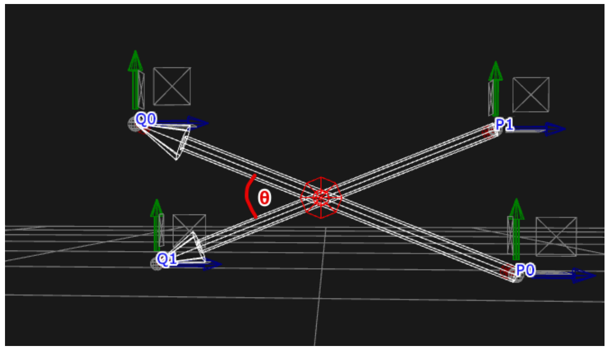
$$d_0 = Q_0 - P_0$$
$$d_1 = Q_1 - P_1$$
$$L_0(u) = P_0 + ud_0$$
$$L_1(v) = P_1 + vd_1$$

2. Step by Step derivation

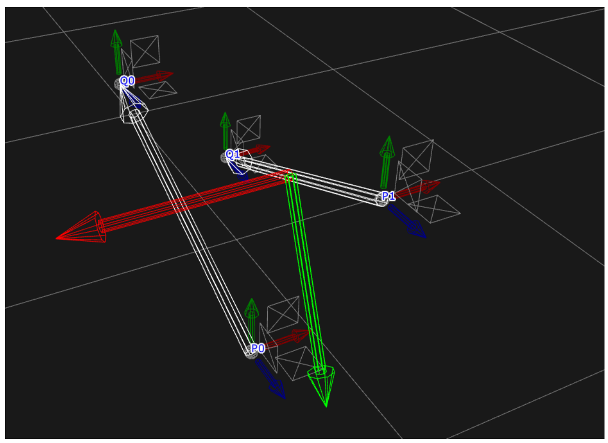
First, let's get n perpendicular against d_1 and d_2



$$n = d_0 \times d_1$$
$$|n| = |d_0||d_1|\sin\theta$$

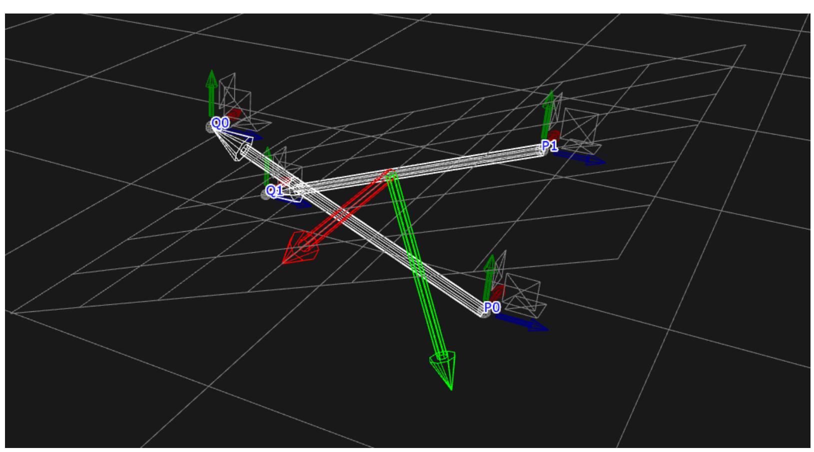


and we can define another green vector n_g that is orthonogal against d_2

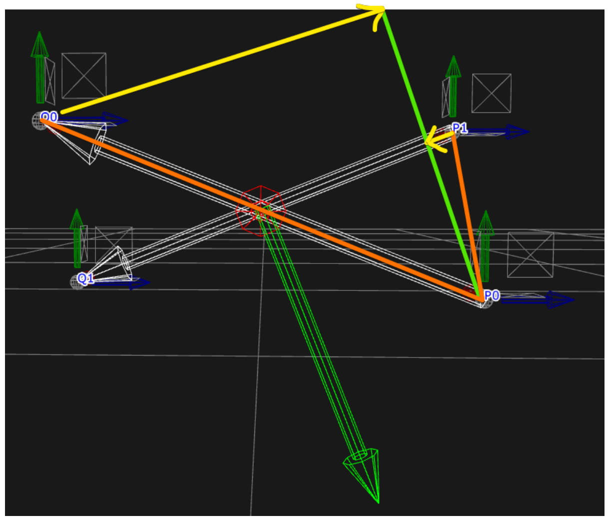


$$n_g = n \times d_1$$

A possible idea is to consider n_g as normal vector that along with L_1



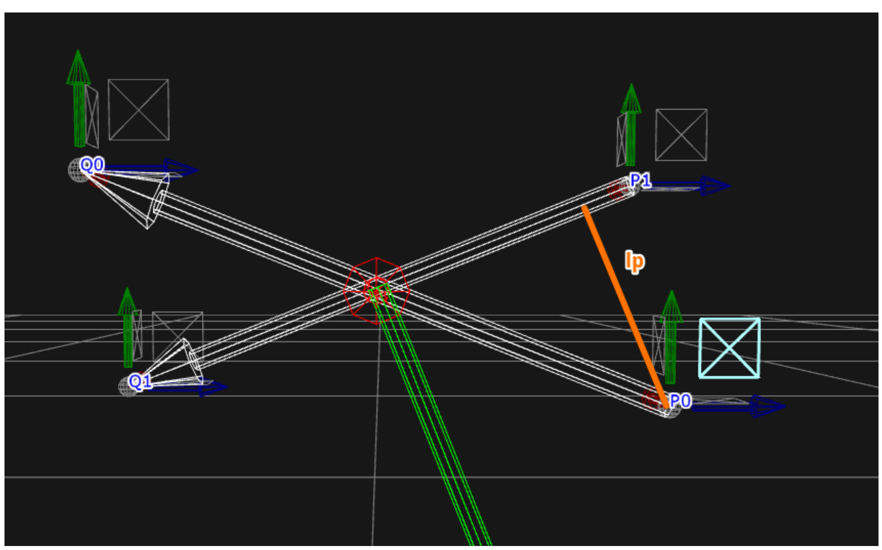
So we can project d_1 and $(P_0 - P_1)$ into n_g to calculate u



$$u = \frac{\frac{n_g}{|n_g|} \cdot (P_1 - P_0)}{\frac{n_g}{|n_g|} \cdot d_1} = \frac{n_g \cdot (P_1 - P_0)}{n_g \cdot d_1}$$

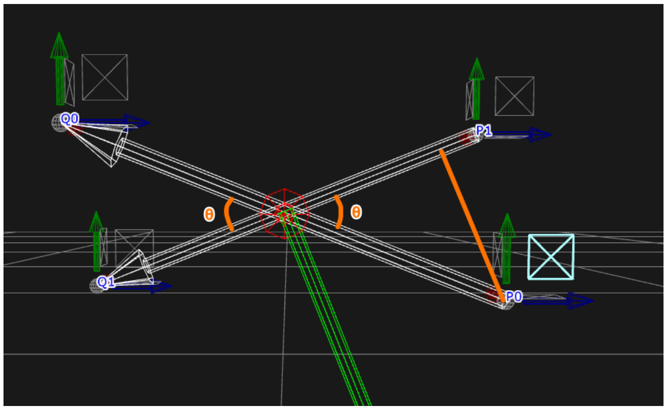
This idea is described here https://en.m.wikipedia.org/wiki/Skew_lines#Nearest_Points and it works well. But there is something we can do.

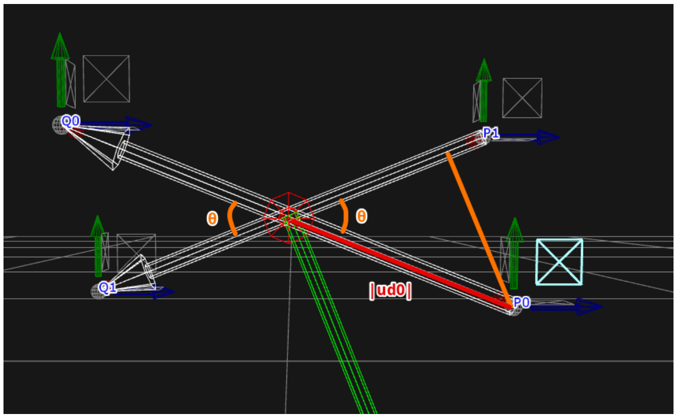
First, get projected length of $(P_0 - P_1)$ to n_g this is a similar operation above



$$l_p = \frac{n_g}{|n_g|} \cdot (P_0 - P_1)$$

And we calculate $|ud_0|$





$$|ud_0| = \frac{l_p}{\sin\theta}$$

Thus, u is

$$u = \frac{l_p}{|d_0|\sin\theta}$$

Let's simplize the formula

$$u = \frac{l_p}{|d_0|\sin\theta}$$

$$= \frac{1}{|d_0|\sin\theta}\left(\frac{n_g}{|n_g|}\cdot(P_0 - P_1)\right)$$
$$= \frac{1}{|d_0|\sin\theta}\left(\frac{n\times d_1}{|n\times d_1|}\cdot(P_0 - P_1)\right)$$

we already know $n \perp d_1$. so

$$u = \frac{1}{|d_0|\sin\theta}\left(\frac{n\times d_1}{|n||d_1|}\cdot(P_0 - P_1)\right)$$
$$= \frac{1}{|d_0|\sin\theta}\left(\frac{n\times d_1}{|n||d_1|}\cdot(P_0 - P_1)\right)$$

also we know

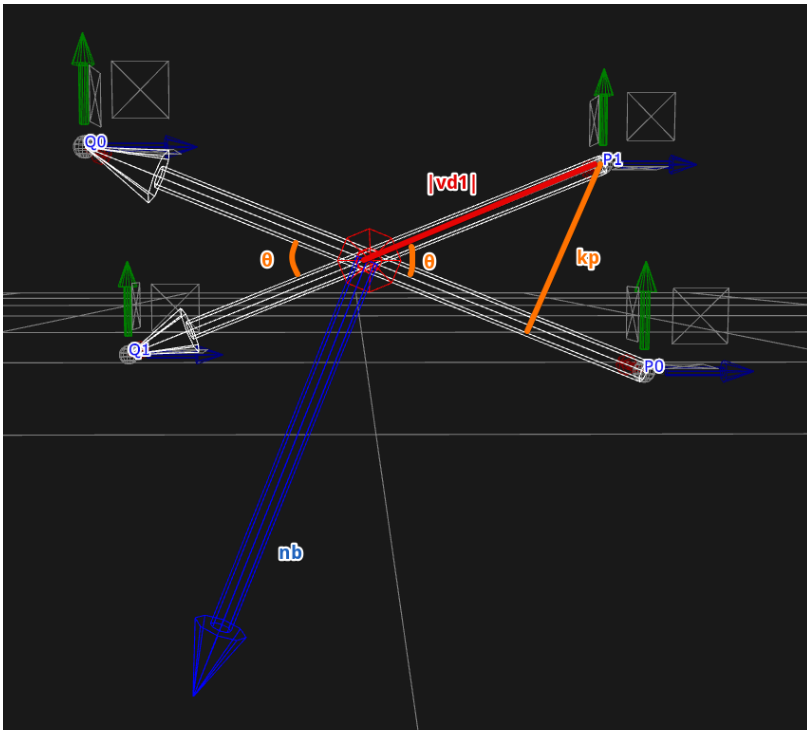
$$n = d_0 \times d_1$$
$$|n| = |d_0||d_1|\sin\theta$$

Thus

$$u = (n\times d_1)\cdot(P_0 - P_1)\frac{1}{|n|^2}$$
$$u = (n\times d_1)\cdot(P_0 - P_1)\frac{1}{n\cdot n}$$

What simple!

we can apply this idea to v as well.



$$n_b = n \times d_2$$

$$k_p = \frac{n_b}{|n_b|} \cdot (P_0 - P_1)$$

$$|vd_1| = \frac{k_p}{\sin\theta}$$

$$v = \frac{k_p}{|d_1|\sin\theta}$$

$$= \frac{1}{|d_1|\sin\theta}\left(\frac{n_b}{|n_b|} \cdot (P_0 - P_1)\right)$$

$$= \frac{1}{|d_1|\sin\theta}\left(\frac{n \times d_2}{|n \times d_2|} \cdot (P_0 - P_1)\right)$$

$$= \frac{1}{|d_1|\sin\theta}\left(\frac{n \times d_2}{|n||d_2|} \cdot (P_0 - P_1)\right)$$

$$= (n \times d_2) \cdot (P_0 - P_1)\frac{1}{|n|^2}$$

$$= (n \times d_2) \cdot (P_0 - P_1)\frac{1}{n \cdot n}$$

Now, we have two formulas and these two are pretty similar.

$$u = (n \times d_1) \cdot (P_0 - P_1) \frac{1}{n \cdot n}$$
$$v = (n \times d_2) \cdot (P_0 - P_1) \frac{1}{n \cdot n}$$

And also we can apply a property of "Scalar triple product".

$$a \cdot (b \times c) = b \cdot (c \times a) = c \cdot (a \times b)$$

So,

$$\begin{aligned} u &= (n \times d_1) \cdot (P_0 - P_1) \frac{1}{n \cdot n} \\ &= (P_0 - P_1) \cdot (n \times d_1) \frac{1}{n \cdot n} \\ &= d_1 \cdot ((P_0 - P_1) \times n) \frac{1}{n \cdot n} \end{aligned}$$

$$\begin{aligned} v &= (n \times d_2) \cdot (P_0 - P_1) \frac{1}{n \cdot n} \\ &= (P_0 - P_1) \cdot (n \times d_2) \frac{1}{n \cdot n} \\ &= d_2 \cdot ((P_0 - P_1) \times n) \frac{1}{n \cdot n} \end{aligned}$$

Then, we got two formulas that can share some terms.

$$u = d_1 \cdot ((P_0 - P_1) \times n) \frac{1}{n \cdot n}$$
$$v = d_2 \cdot ((P_0 - P_1) \times n) \frac{1}{n \cdot n}$$

Finally we got a quite simple and efficient code to calculate u and v

```
glm::vec3 n = glm::cross( d0, d1 );
float length2n = glm::dot( n, n );
glm::vec3 n2 = glm::cross( P0 - P1, n );
float u = glm::dot(n2, d1) / length2n;
float v = glm::dot(n2, d0) / length2n;
```

References

Skew lines, https://en.m.wikipedia.org/wiki/Skew_lines#Nearest_Points

Cool Patches: A Geometric Approach to Ray/Bilinear Patch Intersections, https://research.nvidia.com/publication/2019-03_Cool-Patches%3A-A