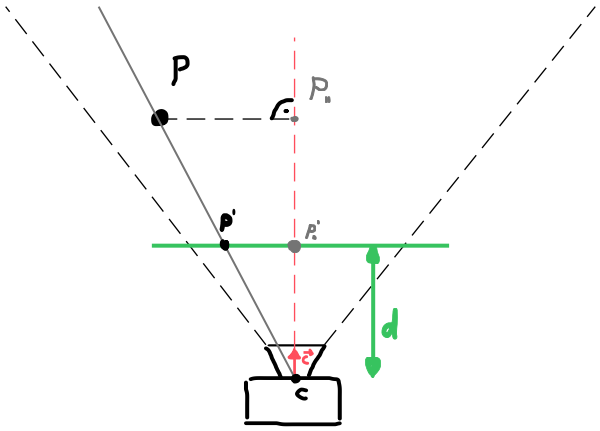
Um Perspektive zu behalten muss P um C zentrisch mit Faktor k gestreckt werden, sodass P immer auf dem Viewplane der Kamera landet



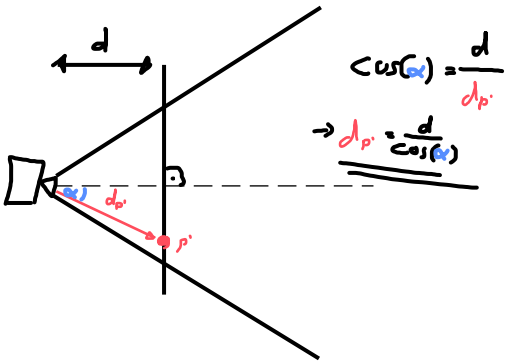
$$k \cdot \vec{P} = \vec{P'}$$

$$\rightarrow k \cdot |\vec{P}_{\perp}| = |\vec{P}'_{\perp}| \quad | \text{Ähnlichkeit}$$

$$= k \left(\frac{\vec{P} \cdot \vec{c}}{|\vec{c}|} \right) = |\vec{P}'_{\perp}|$$

(aus Formel Skalarprodukt, $\perp$; $|\vec{c}| = 1$; $|\vec{P}'_{\perp}| = d$)

$$k = \frac{d}{\vec{P} \cdot \vec{c}}$$



$$\cos(\alpha) = \frac{d}{d_{p'}}$$

$$\rightarrow d_{p'} = \frac{d}{\cos(\alpha)}$$

Grundsätzliches Vorgehen für die Projektion eines Punktes auf den Bildschirm:

Das Ziel ist es, den Punkt auf eine virtuelle Leinwand, die sich senkrecht in einer festgelegten Distanz vor der Kamera befindet, zu projezieren. Dabei soll die Perspektive beibehalten werden, das heisst, dass der Punkt um die Kamera mit einem bestimmten Faktor zentrisch gestreckt werden muss.
Erst wird der Punkt in den lokalen Koordinaten der Kamera angegeben. Das heisst, dass das ab jetzt die Position des Punktes so angegeben wird, wie es für jemanden wär, der die Kamera hält. Zum Beispiel würde dann ein Punkt gerade aus vor der Kamera nun die Koordinaten (0, 1) haben, egal wo er sich im globalen Koordinatensystem befindet, oder wie die Kamera um ihn gedreht ist.
Dann wird der Punkt so um die Kamera zentrisch gestreckt, dass er auf der virtuellen Leinwand landet. Da die Leinwand ja immer konstant vor der Kamera bleibt und sich diese im lokalen Kamera-Koordinatensystem weder bewegt noch dreht, ist diese Leinwand immer parallel zum Bildschirm und wir können nun einfach die z-Komponente der projezierten Punkte weglassen, da die ja immer einfach die z-Koordinate der Leinwand ist, und erhalten so die Bildschirm Koordinaten des Projezierten Punktes. Jetzt müssen wir sie nur noch so skalieren, dass die Koordinaten der Ecken der Leinwand mit den Koordinaten der Ecken des Bildschirms übereinstimmen und wir können nun Punkte Projezieren.