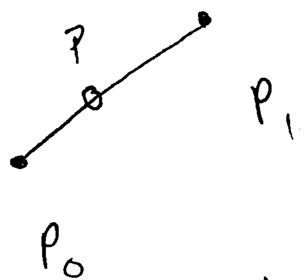
# Interpolation

"LERP" ("Linear Interpolation")



we use a parametric curve:

$$p = (1-t)P_0 + tP_1 \quad \text{(blending)}$$

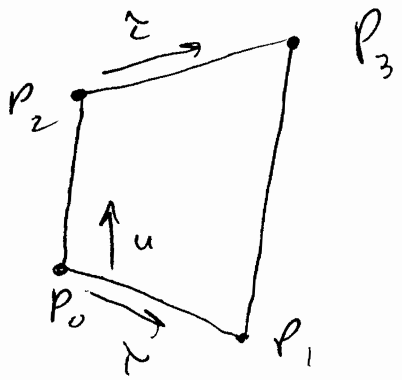
In 3D
$$\begin{cases} x_p = (1-t)\,x_{p0} + t x_{p1} \\ y_p = (1-t)\,y_{p0} + t y_{p1} \\ z_p = (1-t)\,z_{p0} + t z_{p1} \end{cases}$$

→ Typically written in the following more efficient way:

$$P = P_0 + t(P_1 - P_0)$$

- less computations (compute $P_1 - P_0$ once per pair of pts

"BERP" (Bilinear interpolation) (Proper BERPing) 



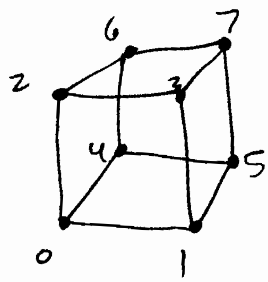

$$P_{01} = (1-\tau)P_0 + \tau P_1$$

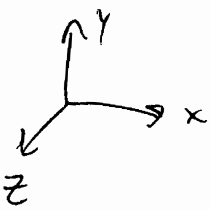
$$P_{23} = (1-\tau)P_2 + \tau P_3$$

$$P_{0123} = (1-u)P_{01} + uP_{23}$$

$$P_{0123} = (1-\tau)(1-u)P_0 + \tau(1-u)P_1 + (1-\tau)uP_2 + \tau u P_3$$

## mentioning of Trilinear Interpolation ("TERP")




- How could ~~you~~ derive this?
  - same as bilinear but another dimension

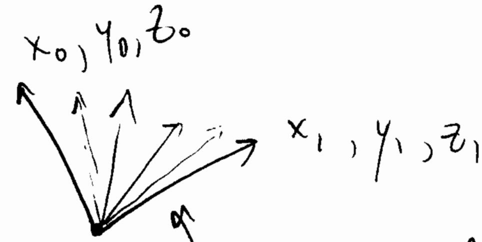
- ~~This~~ Interpolation takes us back to sub- and super-sampling
  - makes more sense to increase number of points this way
  - realise you aren't adding information from the measurements by interpolating

- matlab example shading interp

# Spherical Interpolation ("SLERP")



- we want to interpolate vectors




get wrong lengths using previous approach

$$( y_1 = (1-t)x_0 + t y_1$$


got 0 length vectors in case ~~[illegible]~~ $\theta = \pi$ !

we need a circle (in 2D) with center at origin that passes through both vectors


$$\cos\theta = \frac{\overline{V_0} \cdot \overline{V_1}}{\|V_0\| \, \|V_1\|}$$

if not unit vectors, make them!

Find $\theta$ by taking $\cos^{-1}(\ )$ of both sides:

$$\theta = \cos^{-1}\left( \frac{\overline{V_0} \cdot \overline{V_1}}{\|V_0\| \, \|V_1\|} \right)$$

then

$$\overline{V}(\tau) = \frac{\sin[(1-\tau)\theta]}{\sin\theta} \overline{V_0} + \frac{\sin(\tau\theta)}{\sin\theta} \overline{V_1}$$

- we start at $\overline{V_0}$, end at $\overline{V_1}$

- it's all about perspective