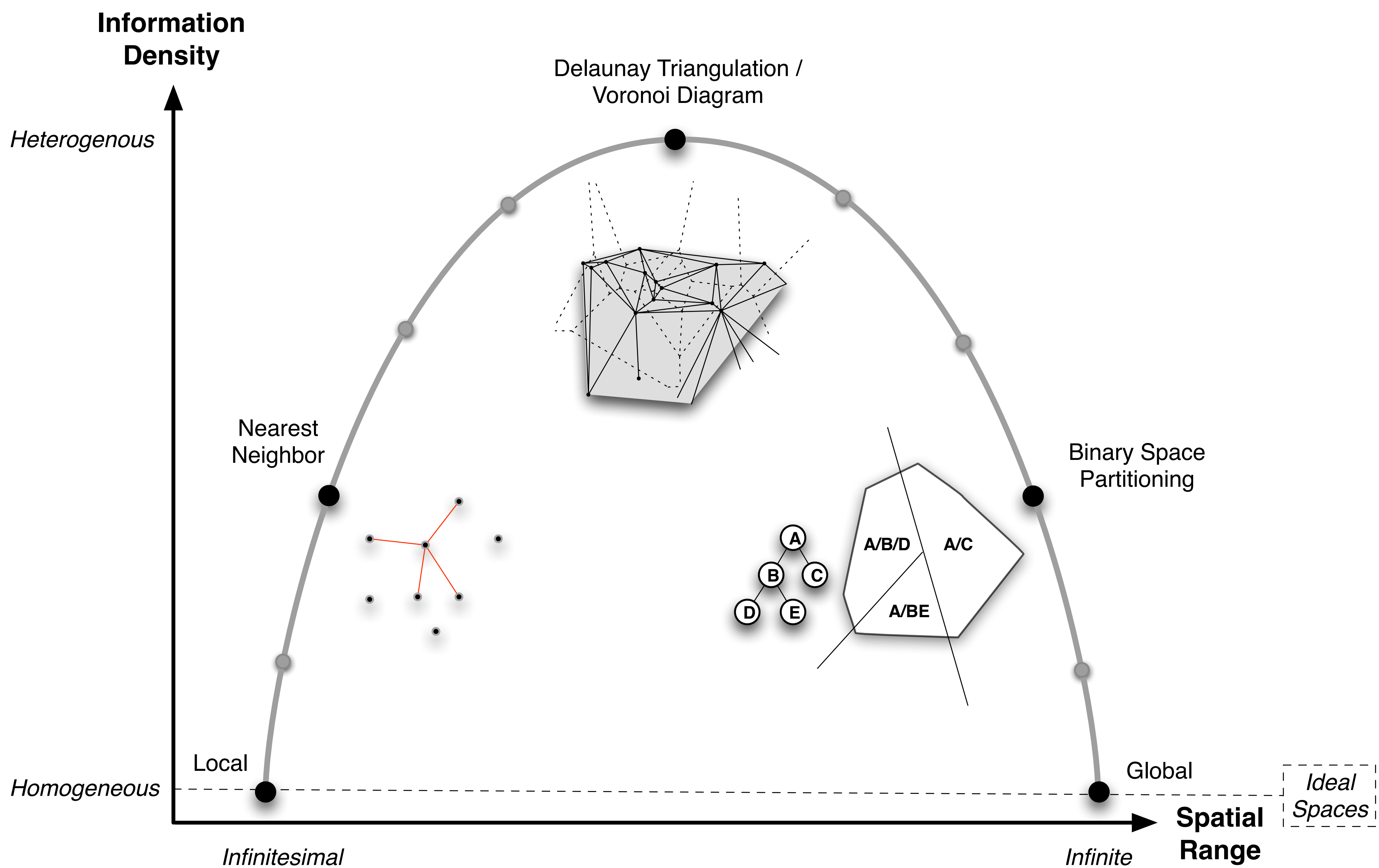
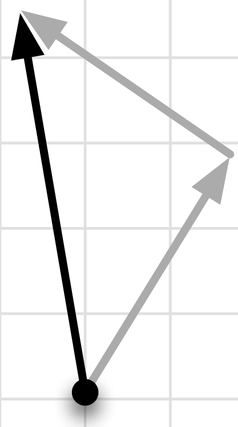
Spectrum of Computational Spaces

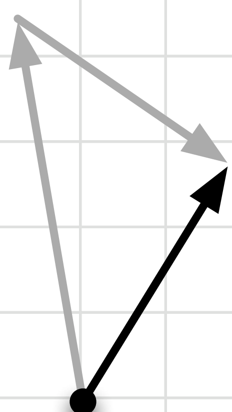


Coordinates and Vectors

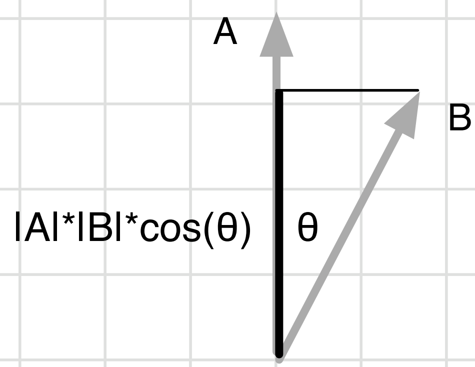
Addition



Subtraction



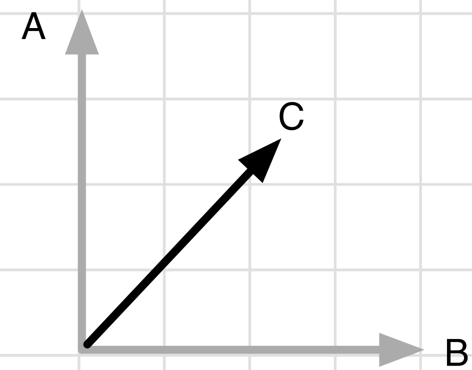
Dot Product



dot(A, B) = |A|*|B|*cos(θ)

<u>answers the question:</u>
How much is a vector aligned with another?

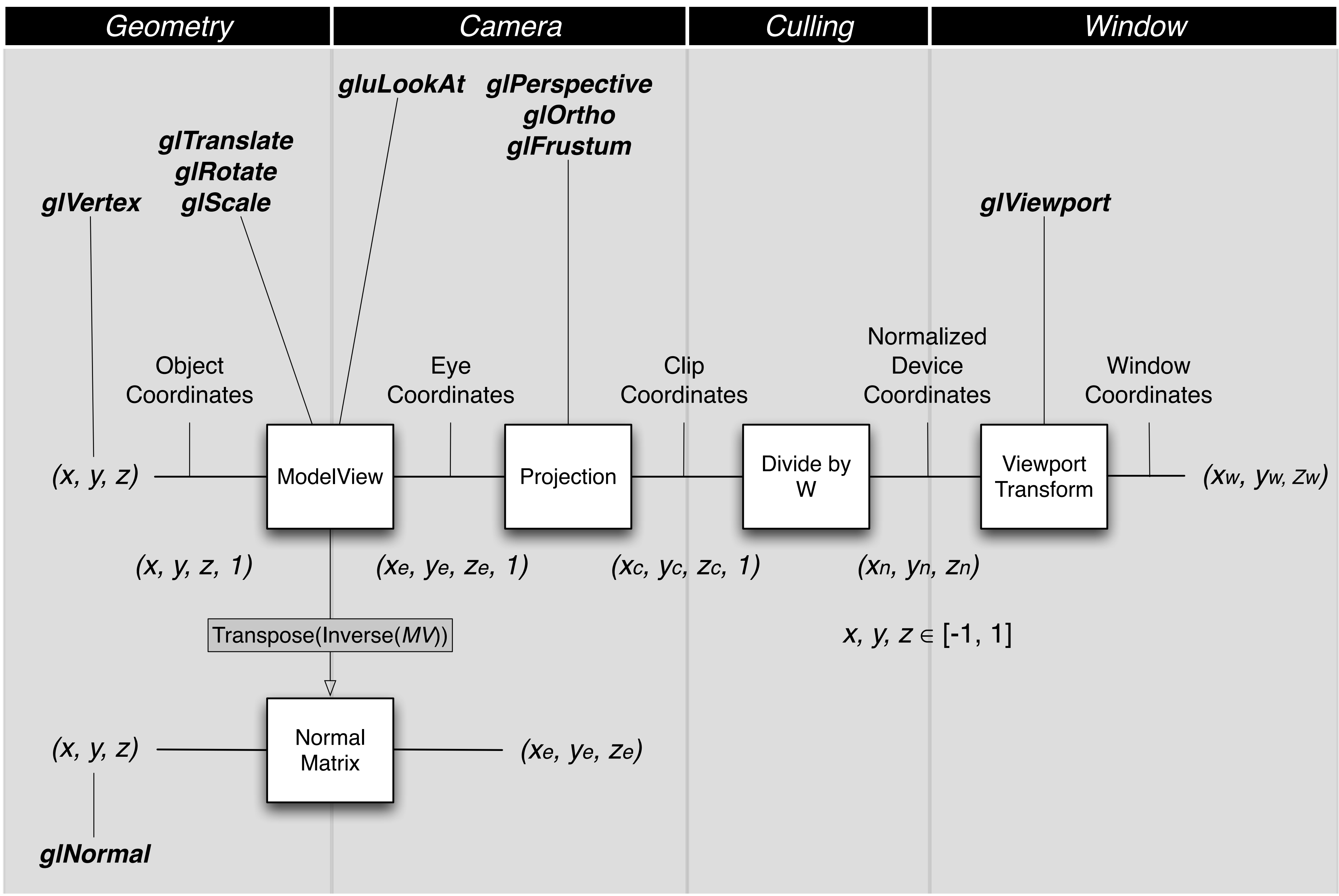
Cross Product



AxB = C, |C| = |A|*|B|*sin(θ)

<u>answers the questions:</u>
How orthogonal are two vectors?
What plane do two vectors define?

Rendering Pipeline



ModelView Matrix

$$\begin{bmatrix} X_{eye} \\ Y_{eye} \\ Z_{eye} \\ W_{eye} \end{bmatrix} = \begin{bmatrix} M_{modelview} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ W=1 \end{bmatrix} = \begin{bmatrix} M_{view} \end{bmatrix} \cdot \begin{bmatrix} M_{model} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ W=1 \end{bmatrix}$$

Homogeneous Coordinates

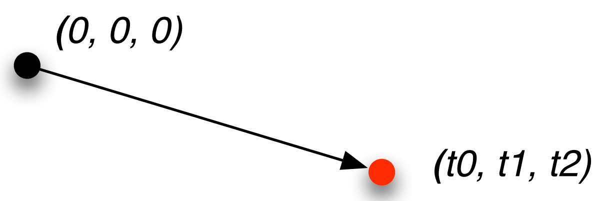
$$\begin{bmatrix} m00 & m10 & m20 & T0 \\ m01 & m11 & m21 & T1 \\ m02 & m12 & m22 & T2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\longleftrightarrow$$

Canonical Affine Transformation

$$\begin{bmatrix} m00 & m10 & m20 \\ m01 & m11 & m21 \\ m02 & m12 & m22 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T0 \\ T1 \\ T2 \end{bmatrix}$$

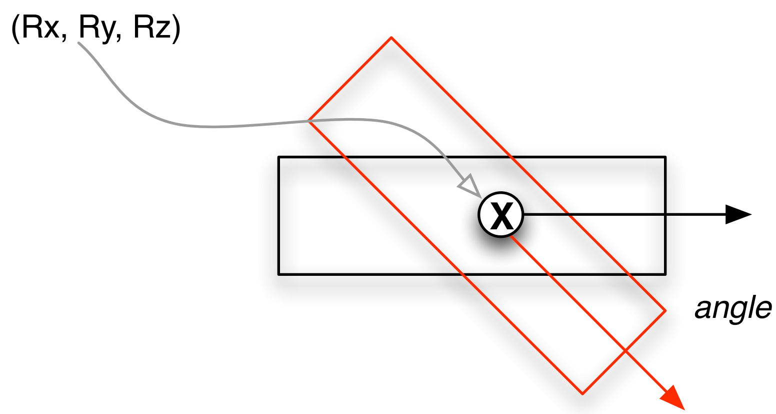
glTranslate(Tx, Ty, Tz)



1	0	0	Tx
0	1	0	Ty
0	0	1	Tz
0	0	0	1

Translation

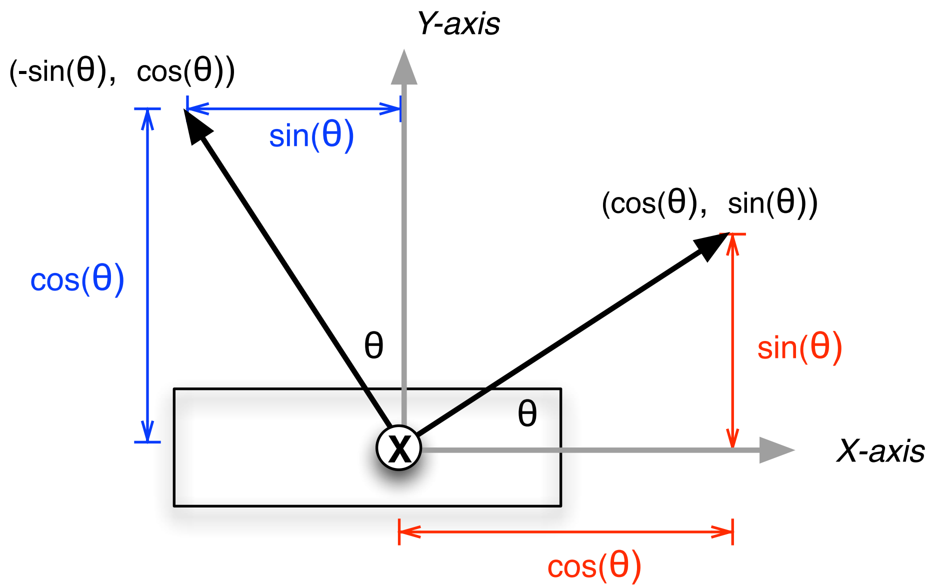
glRotate(angle, Rx, Ry, Rz)



R00	R10	R20	0
R01	R11	R21	0
R02	R12	R22	0
0	0	0	1

Rotation

glRotate(θ, 0, 0, 1)



Rotation

Z-axis *roll*

cos(θ)	-sin(θ)	0	0
sin(θ)	cos(θ)	0	0
0	0	1	0
0	0	0	1

Y-axis *Yaw*

cos(θ)	0	-sin(θ)	0
0	1	0	0
-sin(θ)	0	cos(θ)	0
0	0	0	1

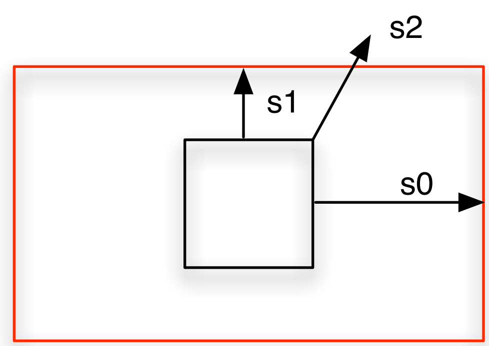
X-axis *Pitch*

1	0	0	0
0	cos(θ)	-sin(θ)	0
0	sin(θ)	cos(θ)	0
0	0	0	1

Compound rotation of 3 angles:

glRotate(*Pitch*, 1, 0, 0)
glRotate(*Yaw*, 0, 1, 0)
glRotate(*Roll*, 0, 0, 1)

glScale(Sx, Sy, Sz)



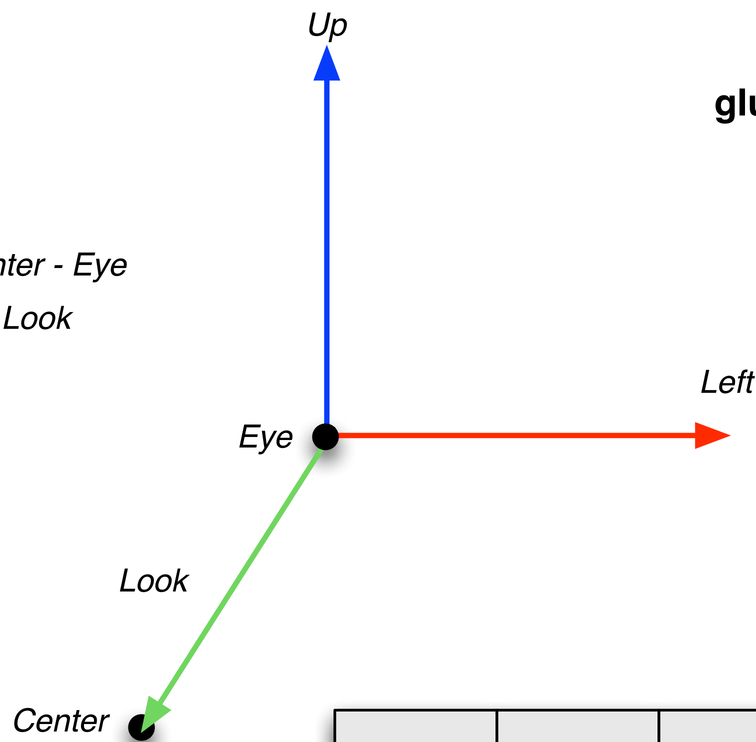
Sx	0	0	0
0	Sy	0	0
0	0	Sz	0
0	0	0	1

Scale

gluLookAt(eye, center, up)

Look = Center - Eye

Left = Up X Look



Left(x)	Left(y)	Left(z)	0
Up(x)	Up(y)	Up(z)	0
Look(x)	Look(y)	Look(z)	0
0	0	0	1

Viewing Axes

•

m00	m10	m20	T0
m01	m11	m21	T1
m02	m12	m22	T2
0	0	0	1

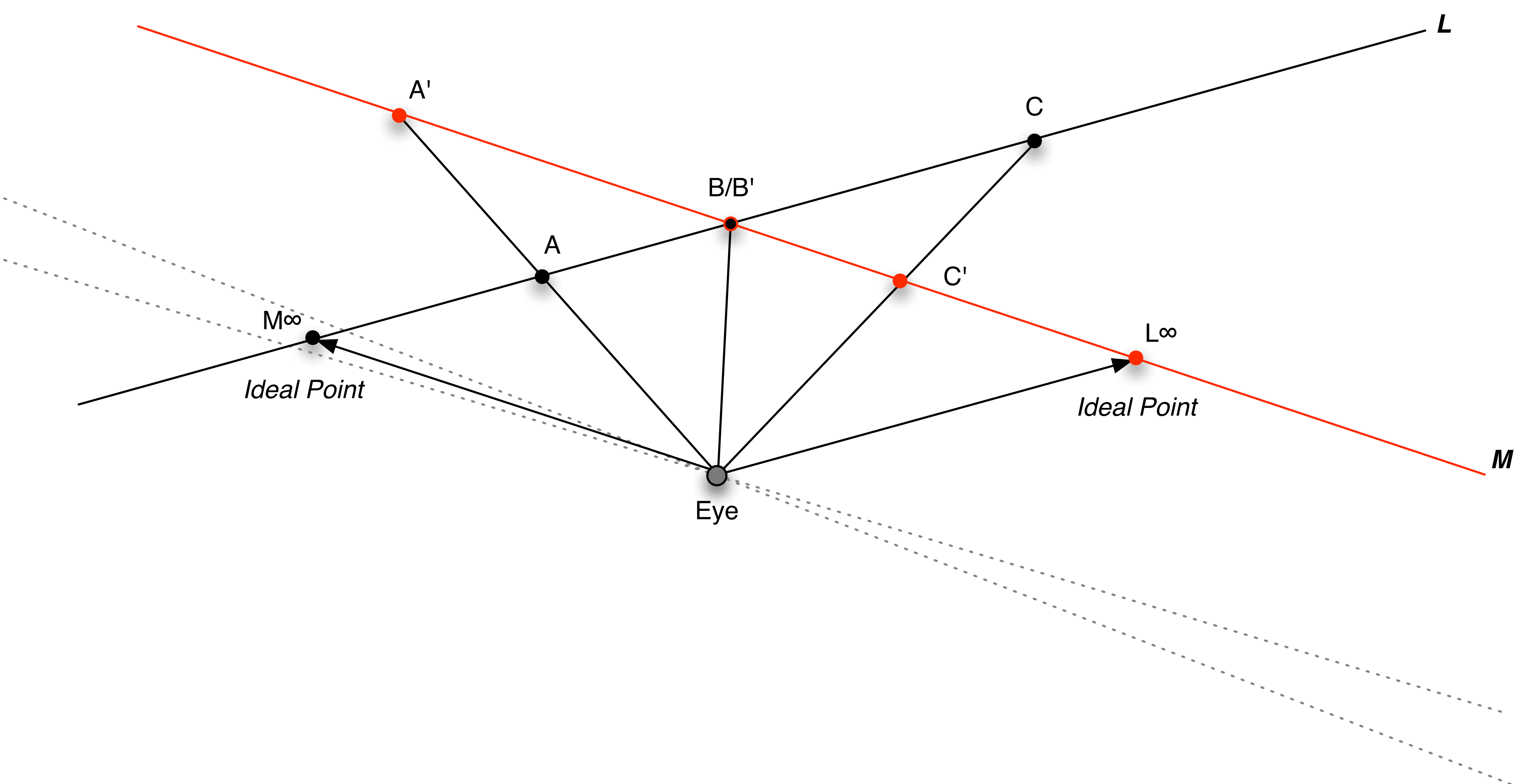
Current MV

Note: viewing transforms can be thought of as opposing model transforms, thus the transposed axes in the View Matrix

1D Projective Mapping

How to map line ***L*** to line ***M***?

Desirable Properties:
bijectivity

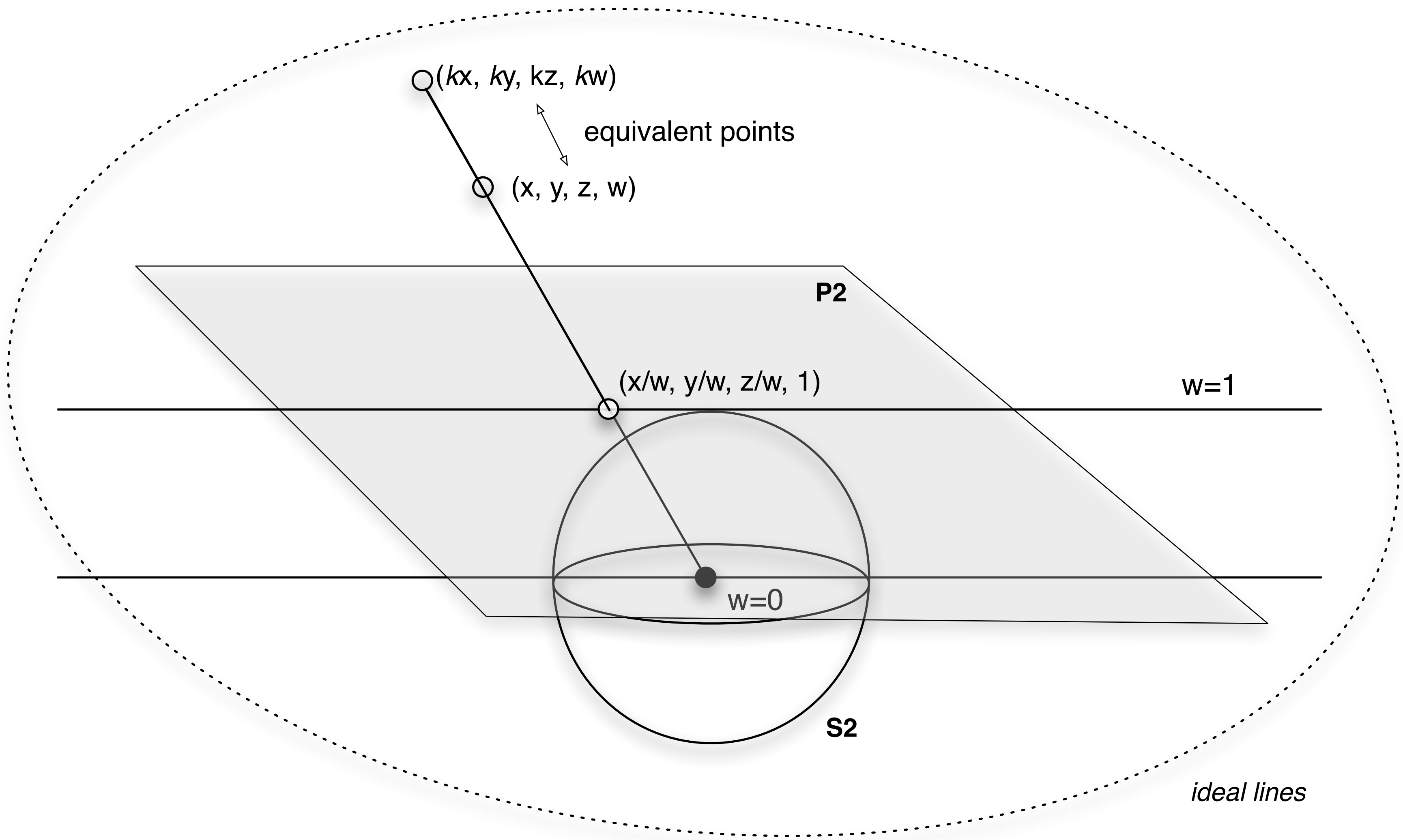


Projective Plane (P2):

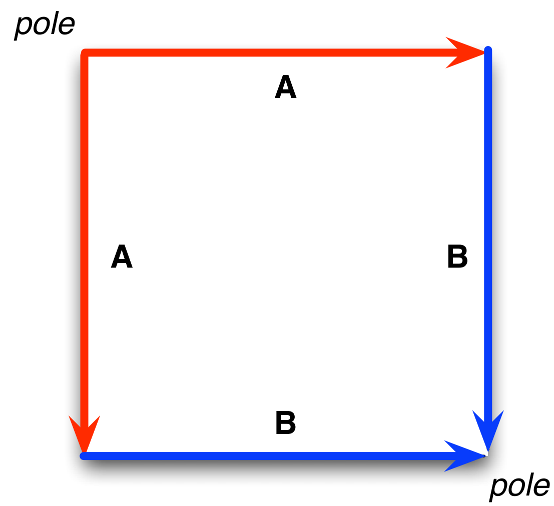
Set of all equivalence classes of ordered triples of non-zero vectors in E3 where equivalence is the mutual proportionality of two vectors

Set of all lines passing through the origin of E3

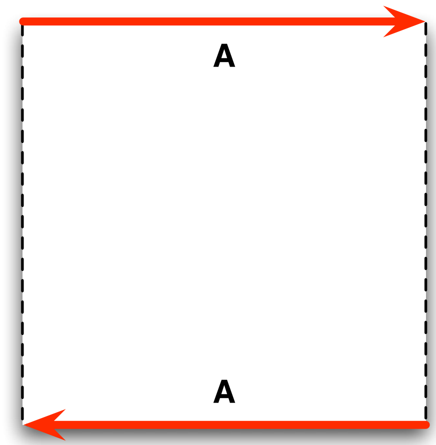
E3 = 3D Euclidean Space
P2 = Projective Plane
S2 = Unit sphere in E3



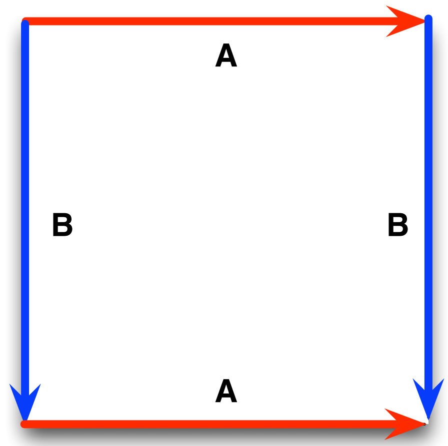
Fundamental Polygons



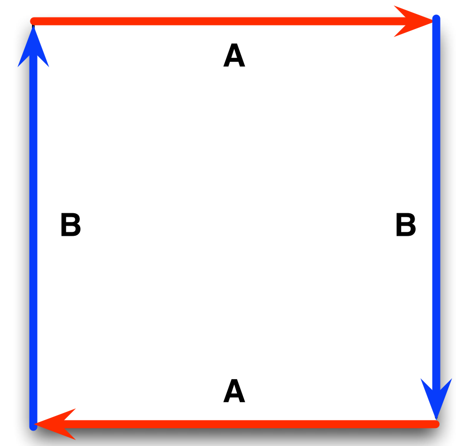
Sphere



Möbius Strip



Torus



Projective Plane

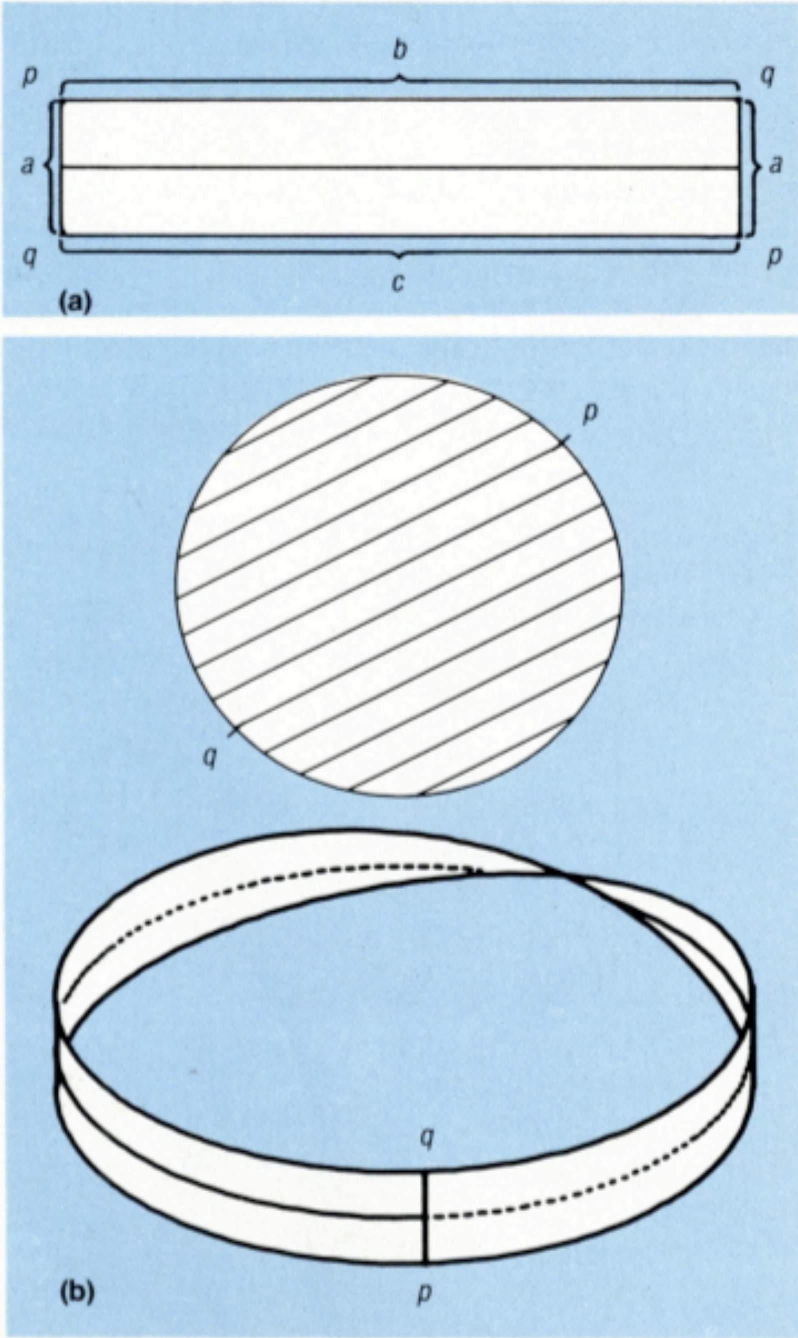


Figure 5. Schematic illustration of Moebius band (a); joining points p and q on the band and the disk (b) produces a projective plane.

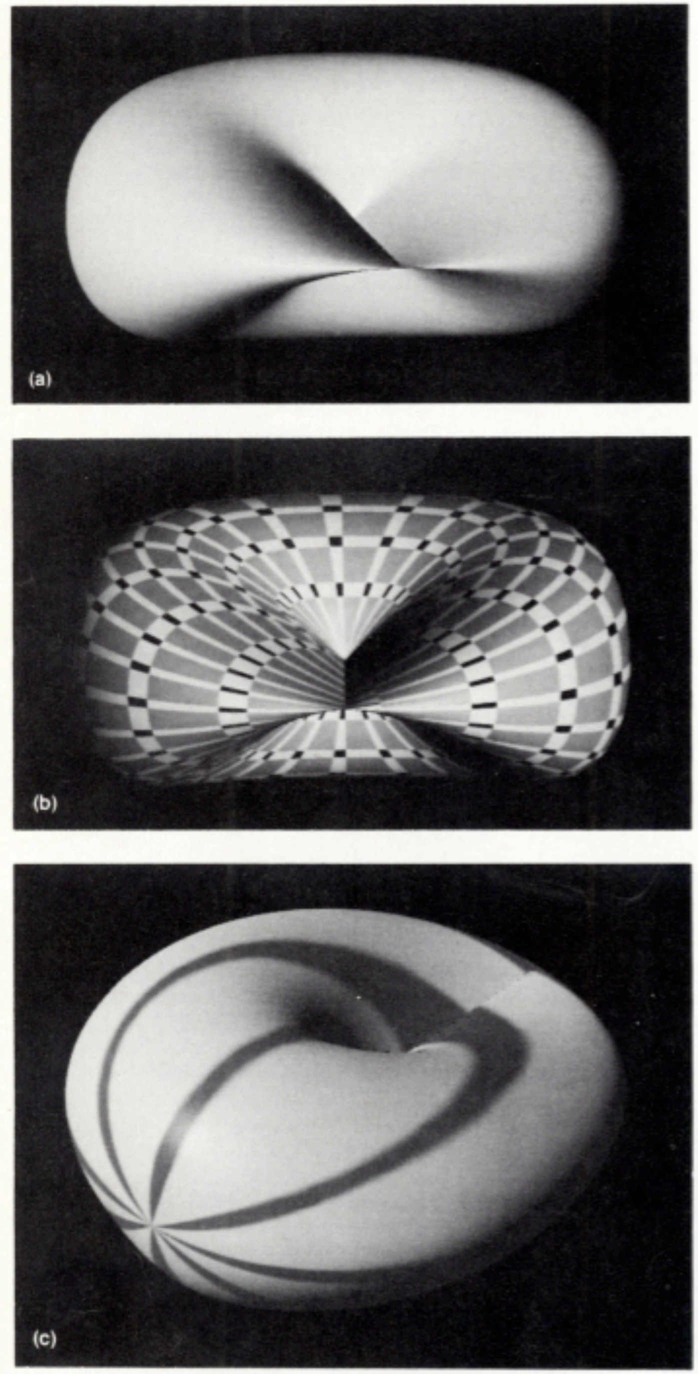


Figure 8. Computer-generated half-tone images of a four-dimensional surface equivalent to the projective plane P^3: (a) the simple projective plane, (b) the plane with textured surface, and (c) the plane with radii emanating from the origin (courtesy of J. Blinn, University of Utah).