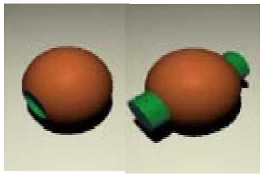
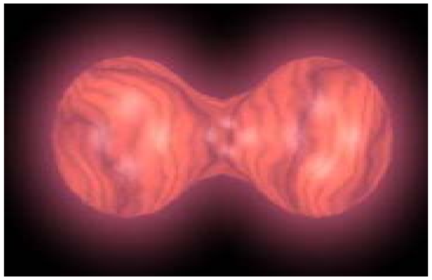
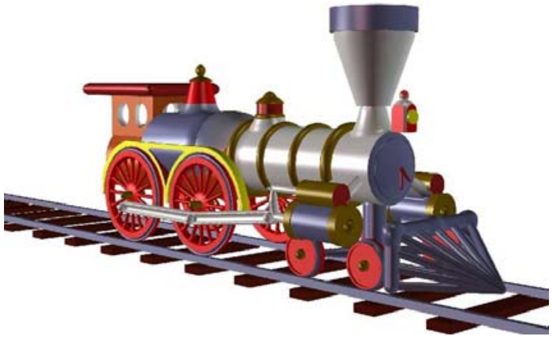



CSC 305
Ray Tracing II

Brian Wyvill



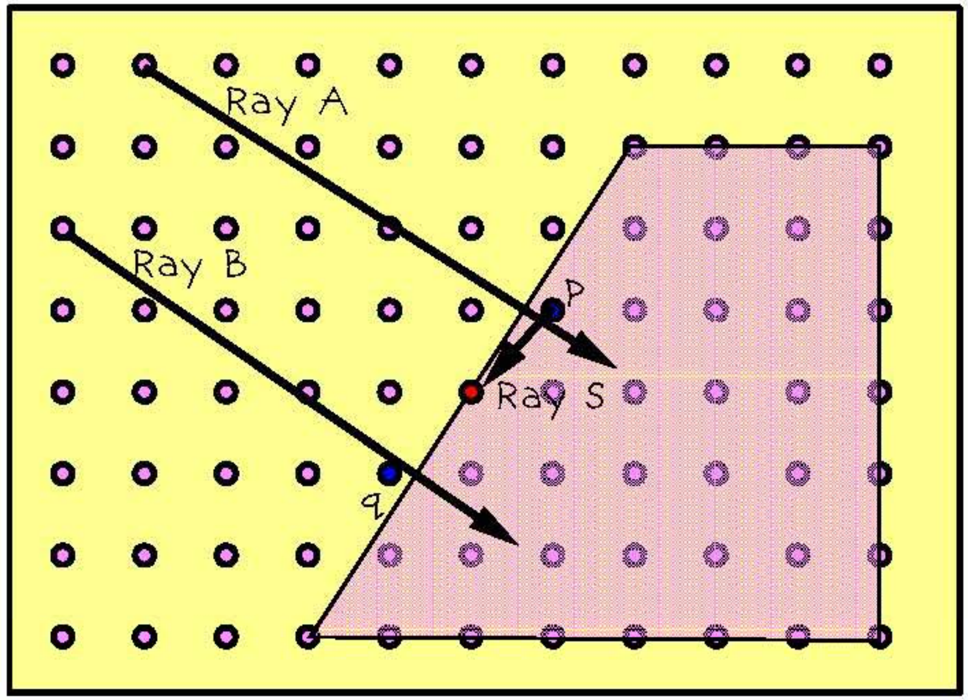

The University of Victoria Graphics Group



Rounding Errors

Numbers represented on a computer can be thought of a sa grid of fixed point fractions. Positions can only be represented by the nearest grid point.

In the example the ray A intersects at p which is inside the surface, whereas the ray B intersects at q on the otherside of the surface. This can lead to problems such as a the surface being in shadow from a source that should be visible.



Possible solutions are to move the point of intersection outside the surface to generate the reflected ray but inside for a refracted ray, by an amount large compared to the rounding error. This deals with many cases but the safe approach is to keep track of which surfaces the ray has crossed.

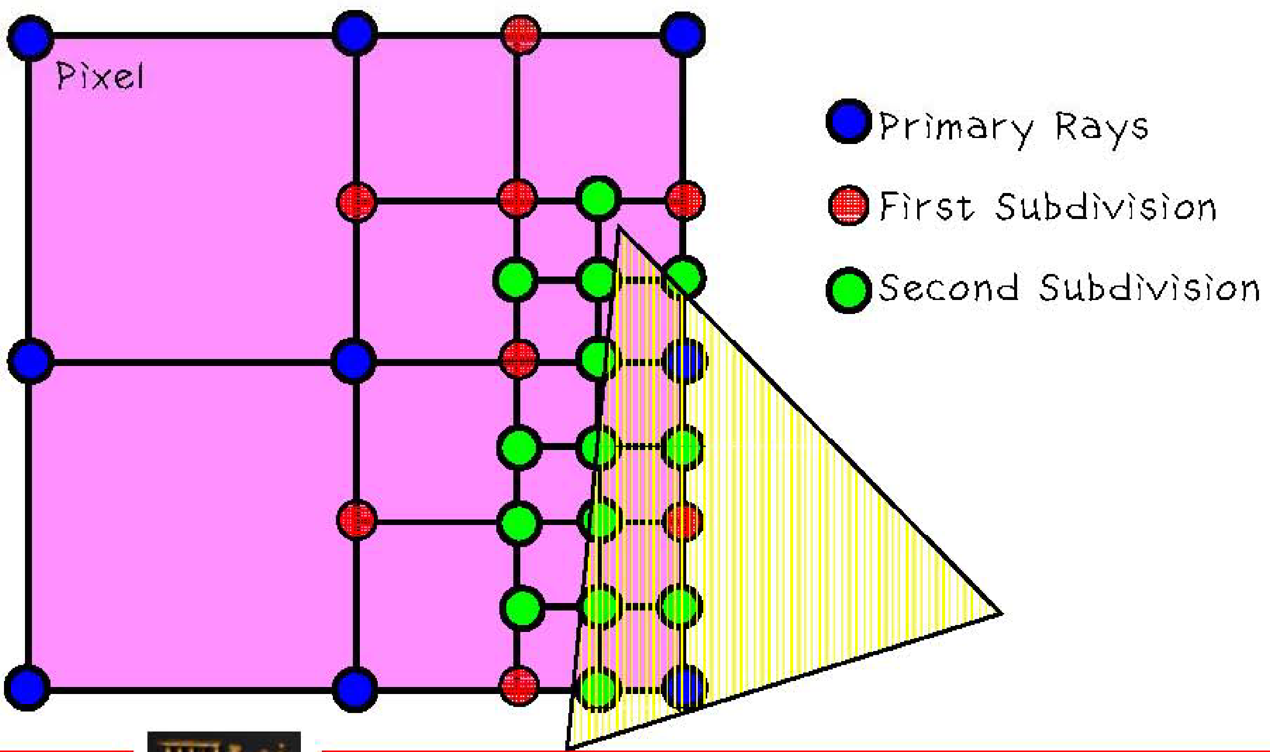
Anti-Aliasing

One Ray/Pixel produces aliasing artifacts due to undersampling.

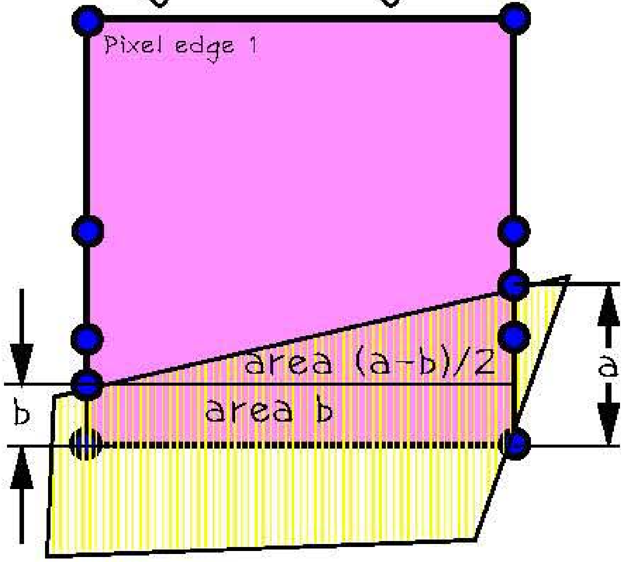
Supersampling

Cast more than one ray per pixel average the result. ~25 rays/pixel are required.

Adaptive supersampling - Binary Subdivision



Finding An Edge



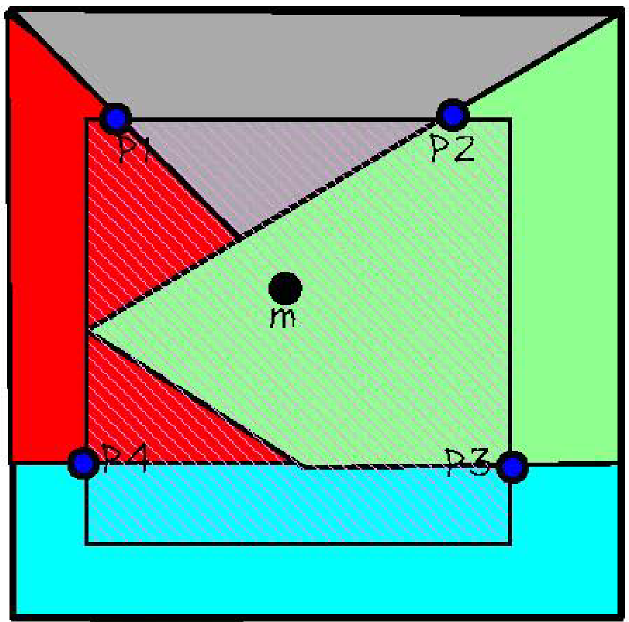
Binary search just the edge of the pixel assuming that edge of polygon cuts the pixel. Just 3 extra rays locates the edge to within 1/8 of the pixel side. Estimate the two areas as follows:

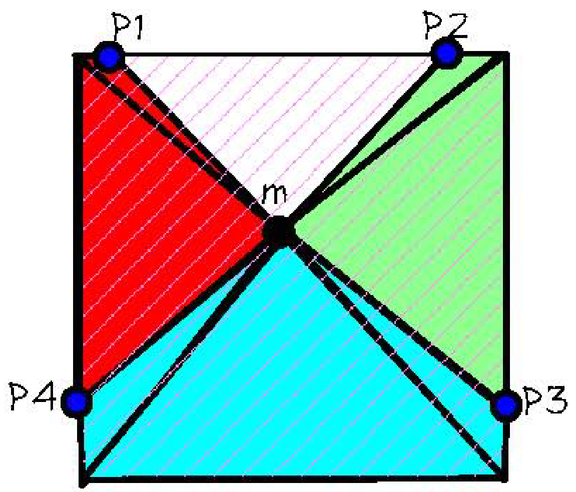
Area covered by polygon in figure: (a+b)/2

More complicated situations require an estimate assuming polygons meet at the centroid m.

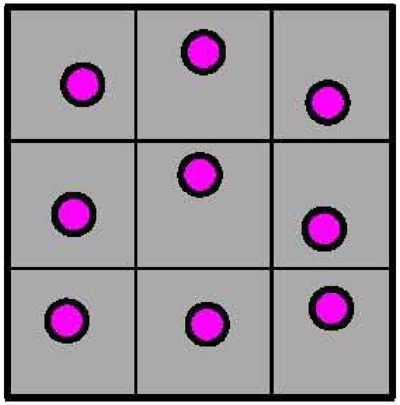
$$m_x = 1/n \sum_{i=1}^{n} P_{ix} \qquad m_y = 1/n \sum_{i=1}^{n} P_{iy}$$

Area calculated as sum of triangles.





Jittered (Stochastic) Sampling



Distributed Ray Tracing

Irregularly spaced samples replace aliasing by noise.

Can also sample colours (rgb are just three samples of the colour space)

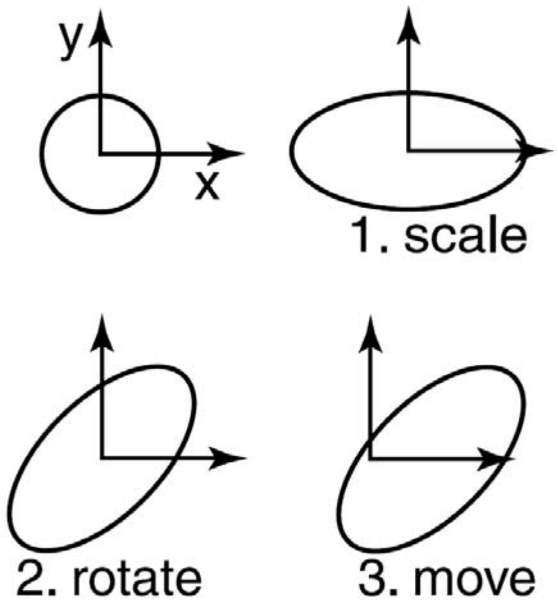
Time - Sample between frames for motion blur.

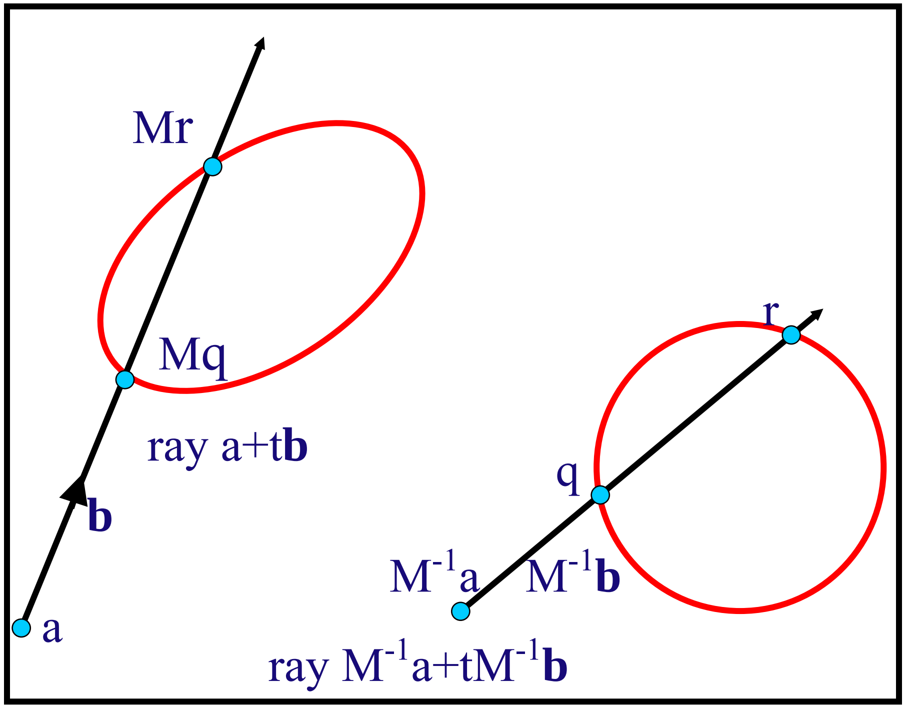
Depth of field - distribute rays over the camera lens.

Blurred Reflections - (and refractions) sample over transmissive and reflection functions for rough objects.

Soft shadows - distribute rays over light source.

Instancing





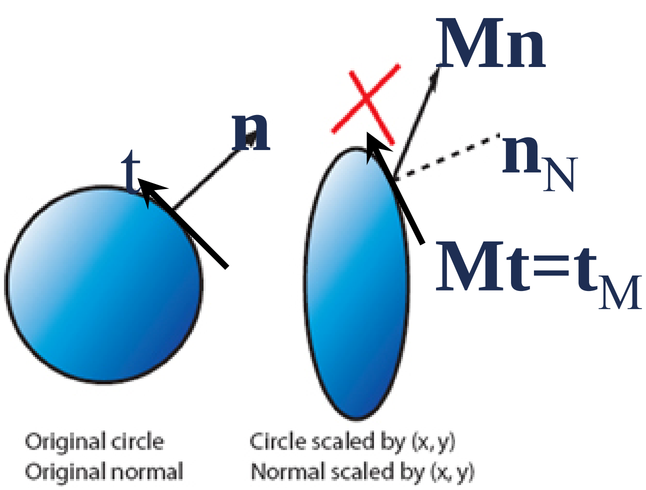
In this case the ray vector is not restricted to unit.
Note normal does not transform like this.

Instancing – surface normal vectors

(see Shirley section 6.2.2)

Points transformed by matrix M. **n** - Normal **t** – tangent.
If **t** tangent so is M**t** **but** M**n** is not necessarily normal anymore.
We derive a transform matrix N **s.t.** N**n** remains normal.

Note that normal and tangent are perpendicular so:



$$\mathbf{n} \bullet \mathbf{t} = 0$$

$$\mathbf{n}^T \mathbf{t} = 0 \quad \text{-----(1) from above}$$

$$\mathbf{t}_M = \mathbf{Mt}$$

$$\mathbf{n}_N = \mathbf{Nn}$$

where **N** is the correct matrix for placing **n**

$$\mathbf{n}_N^T \mathbf{t}_M = 0 \quad \text{-----(2)}$$

Instancing – surface normal vectors

$$\mathbf{n}^T\mathbf{t} = 0 \quad \text{-----(1)}$$

To find **N** we require:

From (1):

$$\mathbf{n}^T\mathbf{t} = \mathbf{n}^T\mathbf{I}\mathbf{t} = \mathbf{n}^T\mathbf{M}^{-1}\mathbf{M}\mathbf{t} = 0$$

$$\mathbf{t}_M = \mathbf{M}\mathbf{t}$$

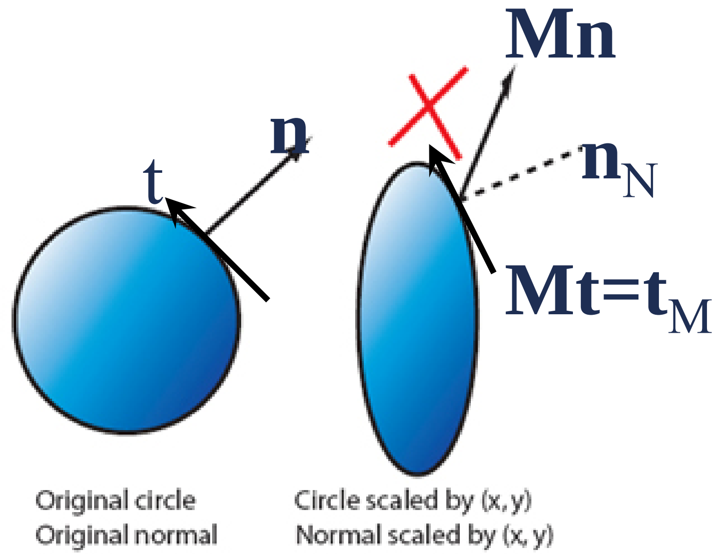
$$\mathbf{n}^T\mathbf{t} = (\mathbf{n}^T\mathbf{M}^{-1})\mathbf{t}_M = 0$$

$$\mathbf{n}_N{}^T\mathbf{t}_M = 0$$

$$\mathbf{n}_N{}^T = (\mathbf{n}^T\mathbf{M}^{-1})$$

transpose

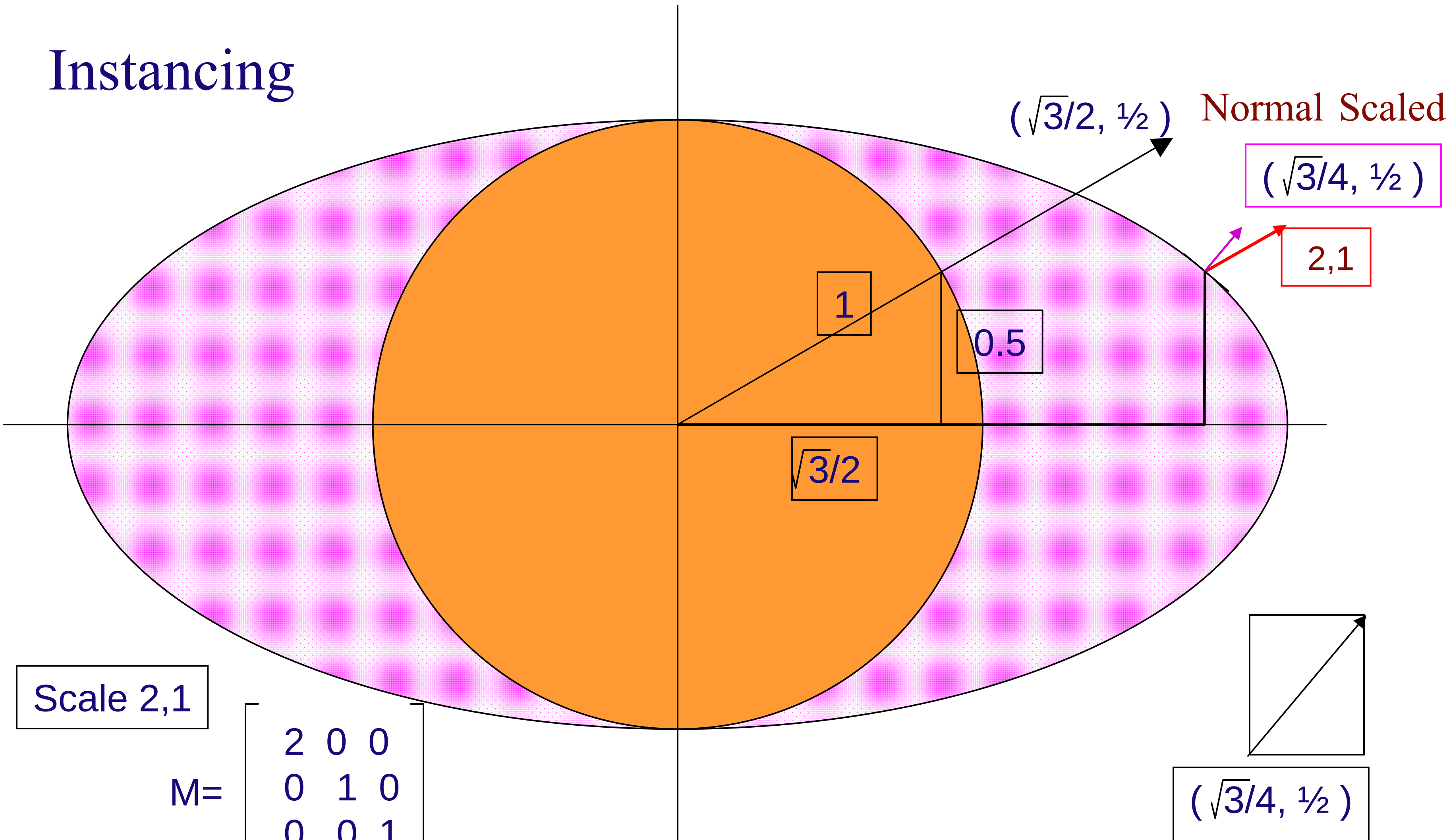
$$\mathbf{n}_N = (\mathbf{n}^T\mathbf{M}^{-1})^T = (\mathbf{M}^{-1})^T\mathbf{n}$$



Matrix to transform normal vector is:

$$\mathbf{N} = (\mathbf{M}^{-1})^T$$

Instancing

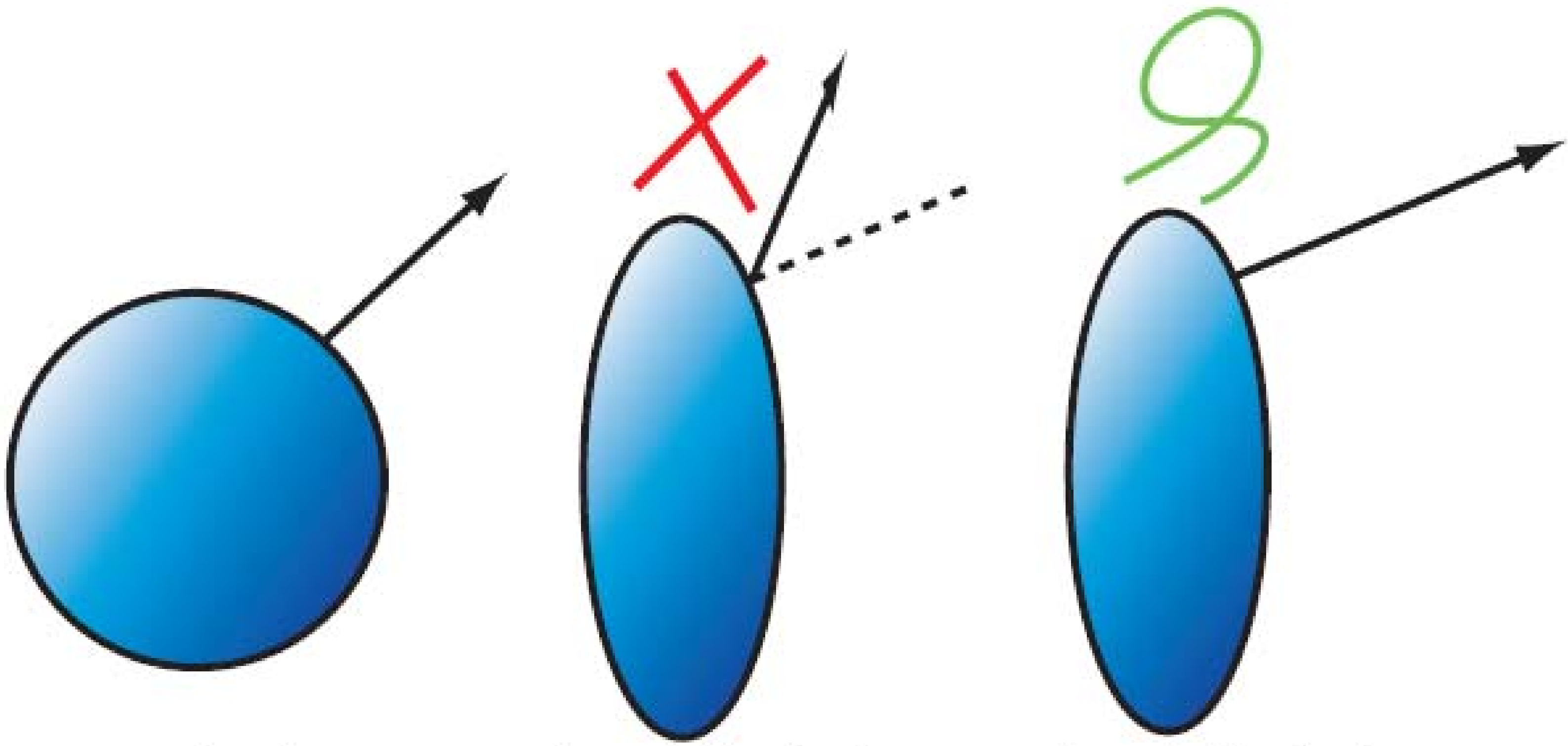


Scale 2,1

$$M = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(M^{-1})^T \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{3}/2 \\ \frac{1}{2} \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{3/4} \\ \frac{1}{2} \\ 1 \end{bmatrix}$$

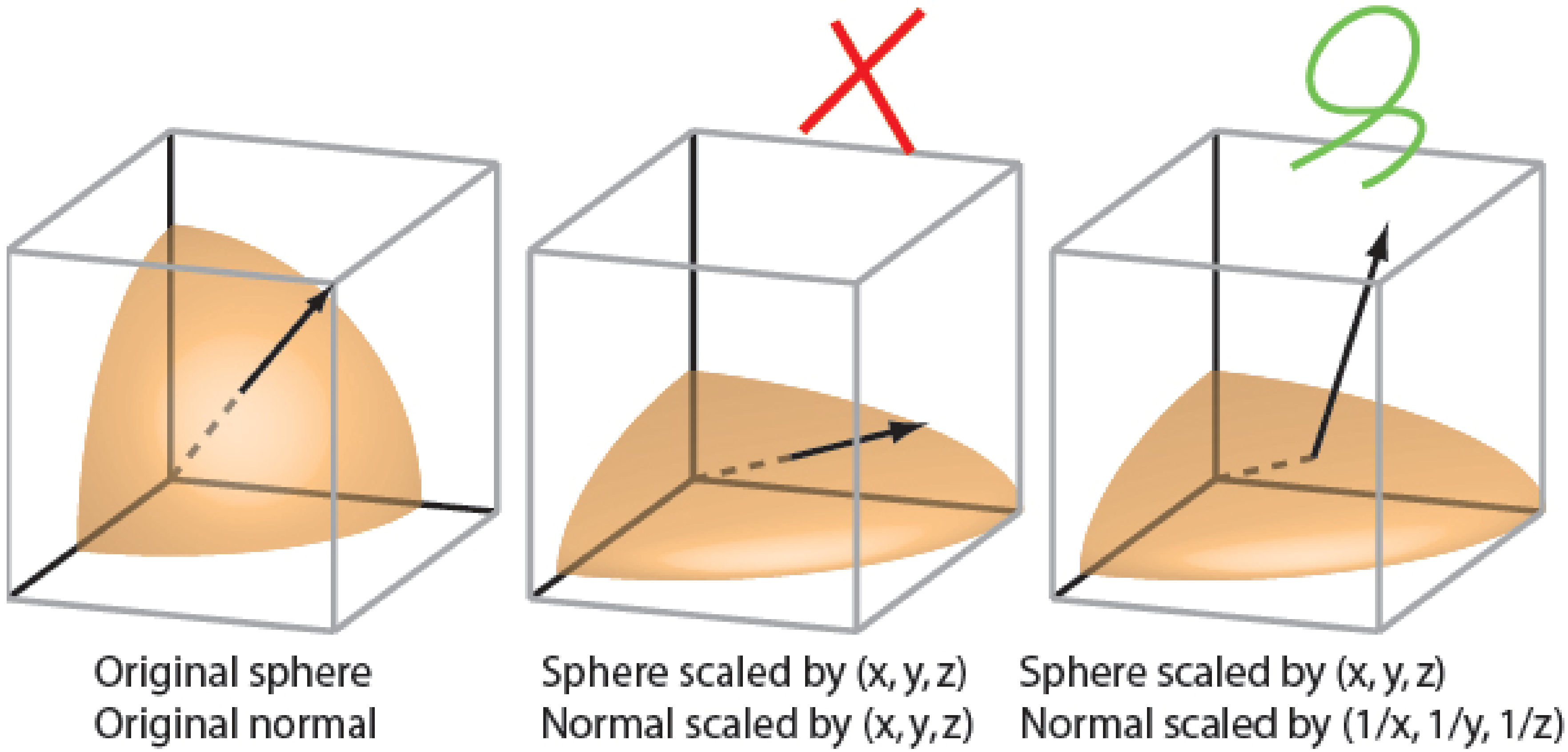
2D example
3x3 matrix



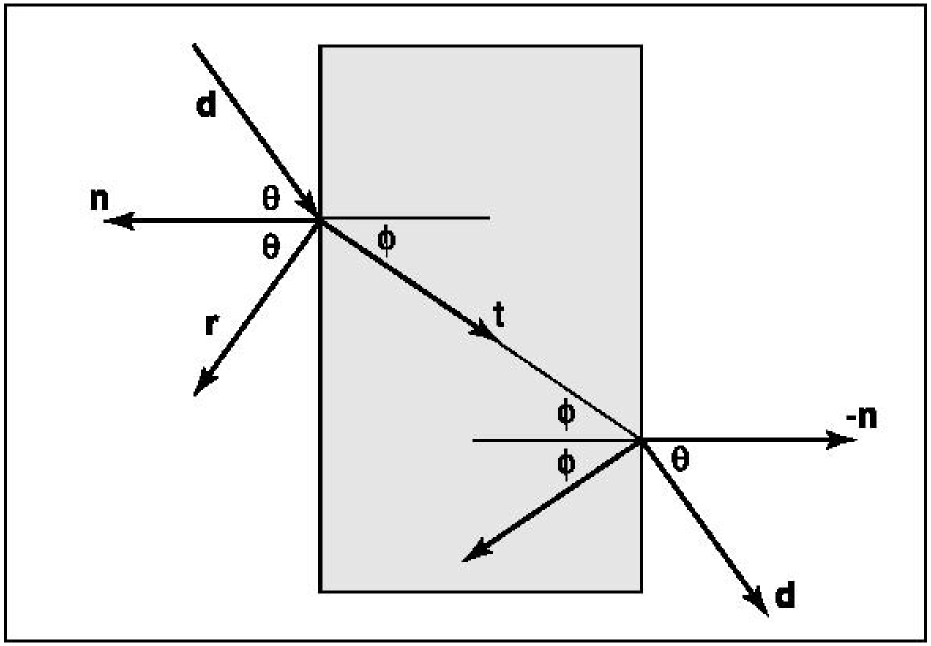
Original circle
Original normal

Circle scaled by (x, y)
Normal scaled by (x, y)

Circle scaled by (x, y)
Normal scaled by (1/x, 1/y)


Original sphere
Original normal
Sphere scaled by (x, y, z)
Normal scaled by (x, y, z)
Sphere scaled by (x, y, z)
Normal scaled by (1/x, 1/y, 1/z)

Refraction



Snell's Law

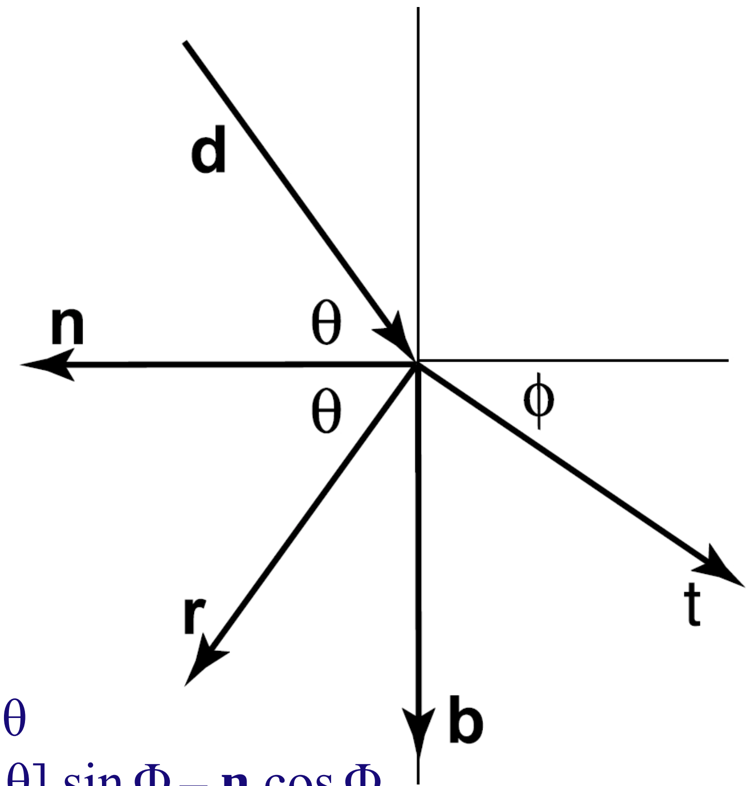
$\mu \sin\theta = \mu_t \sin\Phi$

Replace sin with cos (so we can program using dot product)

Using: $\sin^2 + \cos^2 = 1$

$\cos^2 \Phi = 1 - \mu^2 (1 - \cos^2 \theta) / \mu_t^2$

Refraction



Make the vectors 3D

n and **b** form ortho-normal 2D basis

Transmitted vector **t** in terms of the basis:

$\mathbf{t} = \mathbf{b} \sin \Phi - \mathbf{n} \cos \Phi$

Incident ray **d** is in same plane (basis)

$\mathbf{d} = \mathbf{b} \sin \theta - \mathbf{n} \cos \theta$

Solve for **b**: $\mathbf{b} = (d + \mathbf{n} \cos \theta) / \sin \theta$

And this solve for t: $\mathbf{t} = [(d + \mathbf{n} \cos \theta) / \sin \theta] \sin \Phi - \mathbf{n} \cos \Phi$

But $\mu \sin\theta = \mu_t \sin\Phi$

Yielding:

$$\mathbf{t} = (\mu/\mu_t)(\mathbf{d} - \mathbf{n}(\mathbf{d.n})) - \mathbf{n}\sqrt{\frac{1 - \mathbf{n}^2(1-(\mathbf{d.n})^2)}{\mu_t^2}}$$

(check this before using)