Transforming Objects

Ray : \( R(t) = s + ct \)
Objects : Sphere, box, cone etc.

We assume the objects to be normalized so that the ray-object intersection test is easier.
e.g. Sphere is \( x^2 + y^2 + z^2 = 1 \)

Now while instantiating the objects in our hierarchical scene we can apply various transformations to this normalized primitives to get variety in our scene.
Transforming Objects

Let us say that the sphere is transformed under an Affine transformation \( T : M, d \)

\[ q = q' M + d \]
Transforming Objects

Thus, \( s' + c' t = (s + ct) M^{-1} \)

Solving we get

\( s' = (s - d) M^{-1} \)

\( c' = c M^{-1} \)
Transforming Objects

For rendering we also require the normal at the point of intersection of the transformed primitive. Suppose the normalized primitive had the normal $n$ at its point of intersection with the ray. And $p_a$ and $p_b$ be two points arbitrarily close on the normalized object. Then

$$(p_a - p_b) \cdot n = 0$$

$$(p_a - p_b) \, n^T = 0$$

After applying the transformation $T: M \cdot d$ to $p_a$ and $p_b$

$$(p_a - p_b) \, M \, (n')^T = 0$$

Where $n'$ is the normal of the transformed object.
Transforming Objects

Now
\[
(p_a - p_b) M (n')^T = 0
\]

Let N be the transformation which gets applied to the normal of the normalized primitive. Then
\[
(p_a - p_b) M (nN)^T = 0
\]
\[
(p_a - p_b) MN^T n^T = 0
\]

This holds true when \(N^T = M^{-1}\) i.e. \(N = (M^{-1})^T\)

Transformed objects lets us add flexibility to the basic primitives.
Constructive Solid Geometry (CSG)

Compound objects using Boolean Operations

$S_1 \cap S_2$
Ray Tracing

Constructive Solid Geometry (CSG)

Ray inside test

\( S_1 \cap S_2 \)
Ray Tracing

Texture Mapping
Texture Mapping

How do we model the surface details?

Explicit detailed geometry modeling
Expensive and may be unnecessary

Geometry and texture mapping

Shape
Details
Texture Mapping

Mapping Function

Object: \((x(u,v), y(u,v), z(u,v))\)
Texture: \((s, t)\)
Texture Mapping

Mapping Function

Object : \((x(u,v), y(u,v), z(u,v))\)
Texture : \((s, t)\)

\[ u = f(s, t) \quad s = h(u, v) \]
\[ v = g(s, t) \quad t = i(u, v) \]

linear mapping
\[ u = As + B \]
\[ v = Ct + D \]
Ray Tracing

Texture Mapping

Mapping Function

Forward Mapping
Ray Tracing

Texture Mapping

Another approach

Texture

Inverse Mapping

Screen

Pixel
Texture Mapping

Examples

Simple patterns for skin, bricks, etc. May need to repeat texture (tiling)
Texture Mapping

Examples

Requires establishing correspondence between texture and surface points.