

- inverse mapping for a cylinder

. define a cylinder of radius C_r and height C_h as

$$X_c^2 + Y_c^2 = C_r^2, \text{ with } 0 \leq Z_c \leq C_h$$

and an intersection point

- the (u,v) coordinates

. u and v range from 0 to 1

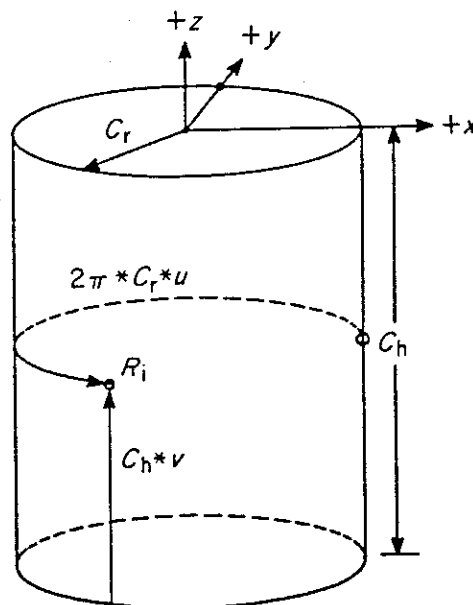
. u starts at the +X axis and moves toward the +Y axis

. v starts at the base of the cylinder and moves toward the top

$$v = Z_i / C_h$$

$$u' = \frac{\arccos(X_i / C_r)}{2 * \pi}$$

if $Y_i < 0$ then set $u = 1 - u'$, else $u = u'$.



- **inverse mapping for a cone**

- **define a cone of height C_h with radius C_{r0} at $Z = 0$ and C_{rh} at $Z = C_h$ as**

$$\sqrt{(X_c^2 + Y_c^2)} = C_{r0} + (C_{rh} - C_{r0}) * Z_c / C_h,$$

with $0 \leq Z_c \leq C_h$

and an intersection point

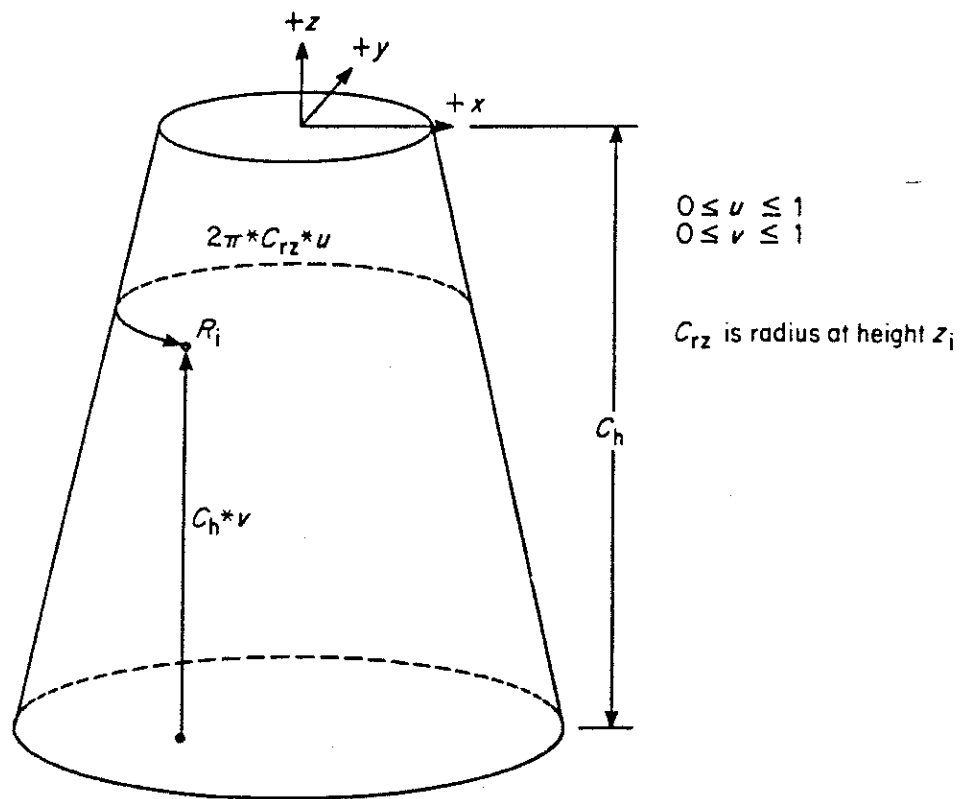
- **the (u,v) coordinates**

- **u and v range from 0 to 1**
- **u starts at the +X axis and moves toward the +Y axis**
- **v starts at the base of the cone and moves toward the top**

$$v = Z_i / C_h$$

$$u' = \frac{\arccos (X_i / (C_{r0} + (C_{rh} - C_{r0}) * Z_i / C_h))}{2 * \pi}$$

if $Y_i < 0$ then set $u = 1 - u'$, else $u = u'$.



- **surface physics**

- **photons, frequency and wavelength**

- the techniques of ray tracing are based mostly on the particle model of light (as opposed to the wave model)
- we can think of a photon traveling through space and "vibrating" up and down at some frequency
- the energy of a photon is its frequency times Planck's constant
- the energy (and the frequency) of a photon determines the color you see

- **light at surfaces - four classes**

- specular reflection
- diffuse reflection
- specular transmission
- diffuse transmission

- **color and spectra**

- a spectrum can be associated with each ray to represent the quantity of photons of each frequency
- a better approach assigns a single wavelength to each ray
 - this doesn't handle refraction very well, because change in direction due to refraction depends on wavelength

- **surface physics, cont.**
 - **reflection**
 - **the reaction of a surface to light arriving at a particular angle is shown in a reflectance spectrum**
 - **the spectrum shows the percentage of incoming light that the surface reflects at each wavelength**
 - **consider a photon striking the surface of some object**
 - **the atoms of the surface are always vibrating at a variety of frequencies**
 - **when the photon strikes the surface, it can transfer some or all of its energy**
 - **if the photon has exactly enough energy to promote the atom to its next stable energy state, the photon is absorbed and the atom moves to a higher energy level for a while (requiring tens of nanoseconds)**
 - **when it sheds the energy, a new photon is generated which has just about the same frequency as the incoming photon, looking like reflection (also requiring tens of nanoseconds)**
 - **lacking sufficient energy for the transfer, some of the photon's energy is absorbed momentarily, but is radiated away in the form of heat (taking just a little longer)**

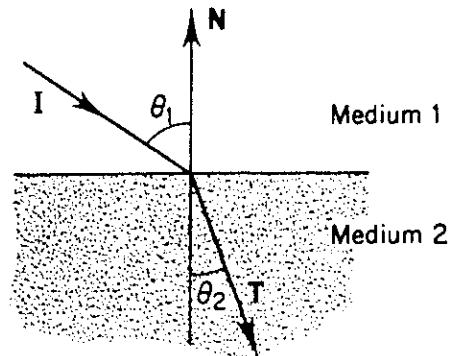
- reflection, cont.
 - . example: when light strikes a blue couch
 - nonblue photons are absorbed and converted to heat
 - blue photons are absorbed and reradiated, or, more simply, are reflected
 - . illustration: black absorbs everything, converting it to heat; white does not

- . **surface physics, cont.**
 - **color and the eye**
 - . **the world has few monochromatic (single frequency) light sources**
 - . **combinations of frequencies produce the colors we "see"**
 - **an infinite number of spectra give rise to a single perceived color**
 - **equivalent colors are called metamers**
 - **the spectra of two pixels may appear unrelated in frequency-vs-amplitude plots, but both spectra may produce perceptually equivalent colors**
 - **perfect diffuse reflection**
 - . **light actually interacts with the surface**
 - . **over the course of time, photons go in all possible directions with equal intensity**
 - . **the only geometry to be considered is how much of the surface is visible to the light source**
 - **the amount of light reaching the surface is proportional to the cosine of the angle between the incident ray and the normal**

surface physics, cont.

- perfect specular transmission

• Snell's law



$$\frac{\sin \theta_1}{\sin \theta_2} = \eta_{21} = \frac{\eta_2}{\eta_1}, \quad \mathbf{T} = \alpha \mathbf{I} + \beta \mathbf{N}$$

where η_1 is the index of refraction of medium 1 with respect to vacuum
 η_2 is the index of refraction of medium 2 with respect to vacuum
 η_{21} is the index of refraction of medium 2 with respect to medium 1.

• the index of refraction depends on the medium; the angle of refraction depends on the wavelength of the incoming light

- that's why a prism separates incoming light into a spectrum

- total internal reflection

• total internal reflection (TIR) occurs when light tries to pass from a dense medium to a less-dense medium at too shallow an angle

- this is the basic mechanism behind fiber optics

• in graphics, when TIR occurs, the equations for perfect specular reflection are used

- . **surface physics, cont.**
 - **perfect diffuse transmission**
 - . **a perfectly diffuse transparent medium would scatter light evenly in all directions**
 - . **of primary concern is the cosine of the angle between the incident ray and the normal, as with diffuse reflection**
 - **practical reflection and transmission - geometry and color**
 - . **surfaces typically have some roughness, so we don't get perfect specular reflection**
 - . **color shifting occurs, for example, when white light strikes a shiny copper kettle, producing a highlight that is more copper colored than white**
 - **caused by multiple bounces off the microfacets of the surface**

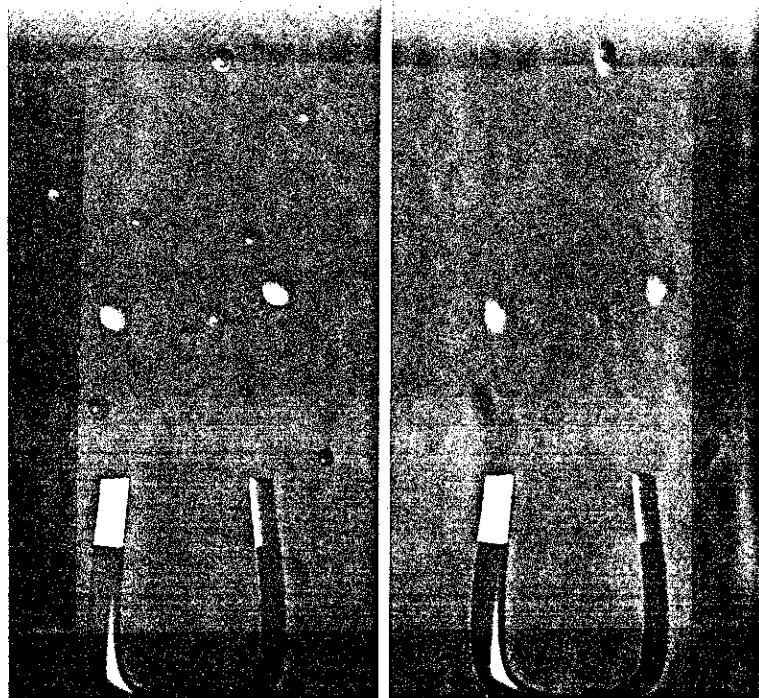
- . **lens and aperture camera model**
 - **the model produces synthetic images which have a depth of field and can be focused on an arbitrary plane**
 - **optical characteristics of a lens can be incorporated**
 - **generation of sample points**
 - . **point samples of light intensity are generated by the Whitted ray-tracing hidden-surface algorithm**
 - . **these samples permit two types of processing**
 - **direct conversion into a raster image using a pinhole camera model**
 - **conversion by a focus processor into a raster image that is focused and has a depth of field**
 - **computing an image using the pinhole camera model**
 - . **sample the 3D scene at the four corners of each pixel**
 - . **extend the ray to intersect the nearest surface element**
 - . **process as in the Whitted algorithm**
 - . **compute the intensity of the pixel by averaging its four point samples**
 - . **perform antialiasing by recursive subdivision of pixels into 2x2 regions and by repeating the sampling and averaging process**

- . lens and aperture camera model, cont.
 - for focused images, save sampled values in a file from nodes of the shading tree
 - . this file is the input for the focus processor
 - . a sample i consists of
 - x''_i and y''_i , the coordinates of the sampled point in the image plane coordinate system
 - z'_i , the accumulated distance along the camera's optical axis of the currently intersected 3D surface point
 - r_i, g_i, b_i , the RGB values contributed by the currently intersected 3D surface point
 - id_i , an identification number

- . lens and aperture camera model, cont.
 - . image defocusing is achieved by introducing a circle of confusion as a projection of a point onto the image plane
 - diffraction effects are studied by determining the light intensity distribution within the circle of confusion for unfocused points
 - the focus processor uses lens and aperture parameters to compute values of impulse-response functions at equally spaced intervals z' between the nearest and farthest points in the image
 - . values of this function are calculated at the z' coordinate for all the pixels within the square area circumscribing the circle of confusion
 - the processor distributes the intensity of each input point to the output pixels according to stored values nearest to the sample z' depth

- **motion blur**
 - **the ray-tracing step**
 - **the ray-tracing hidden-surface process generates intensity sample points of an instantaneous image**
 - **points which are in motion are identified**
 - **the image path of the projected motion is given**
 - **needed parameters**
 - **the list of samples of an image frame**
 - **a list of motion paths of individual objects**
 - **time of the frame exposure**
 - **exposure duration**

- motion blur, cont.
 - . motion-blur processor
 - motion blur is generated by convolving moving points with optical system-transfer functions
 - . functions are derived from the path and velocity of the objects and the exposure time of the virtual camera
 - . the processor separates sample points of a moving object with the same path $r'(t)$ into a raster image i_1
 - . motion blur is computed as a raster image i_2 from t_{frame} (time), T_{frame} (duration) and the path $r'(t)$
 - . the images i_1 and i_2 are convolved into a blurred image $i_1 * i_2$



- **tracing distributed rays**
 - **for effects such as depth of field, motion blur, fuzzy shadows, translucency and fuzzy reflections**
 - **supersampling is assumed**
 - **rays are distributed in time so that rays at different spatial locations are traced at different instants of time**
 - **mirror reflections are determined by tracing rays from the surface in the direction of the mirror**
 - **blurred reflections are calculated by distributing the secondary rays about the mirror direction**
 - **translucency is obtained by distributing the secondary rays about the main direction of the transmitted light**
 - **objects seen through translucent objects are not distinct**
 - **penumbrae are calculated by distributing the secondary rays**
 - **depth of field is calculated by starting with the traditional ray from the center of the lens through point p on the focal plane**
 - **a point q on the surface of the lens is selected and the ray from q to p is traced**
 - **D (lens diameter) is calculated using the focal length F and the aperture number n**
$$D = F/n$$
 - **the motion blur problem also is solved by distributing the sample points in time**

- **optimization techniques for raytracing**
 - **two categories of approaches**
 - **methods for speeding up the intersection process**
 - **methods for reducing the number of ray-object intersection calculations**
 - **software solutions**
 - **several algorithms for efficient calculation of intersections with specific objects**
 - **enclosure of complex objects within simpler objects (bounding volumes)**
 - **a ray which does not intersect the bounding volume does not intersect the enclosed, complex object**
 - **hardware solutions**
 - **subdivision of the image into subimages for processing by a parallel pipelined multimicrocomputer system**
 - **two categories of techniques for reducing the number of ray-object intersection calculations**
 - **algorithms which extend the concept of a ray (cone tracing and beam tracing) and which use the spatial coherence of a polygonal environment or the coherence of rays**
 - **algorithms based on space subdivision**

- **adaptive tree-depth control**
 - . **underlying observation: the percentage of an image comprised of reflective and transparent surfaces generally is low**
 - . **control the depth of the intersection tree used for ray tracing**
 - **a child never contributes more than its parent, which is the aggregate of the children**
 - **when the maximum potential contribution of a node falls below a threshold, terminate the tree**

- **bounding volumes**
 - . the bounding volume must be very simple (a sphere, a rectangular parallelepiped, an ellipsoid)
 - . several objects may be clustered within a bounding volume
 - . clusters may be clustered for more efficient processing
 - . **bounding spheres**
 - intersection may be tested by finding the minimal distance between the ray and the center of the sphere
 - . intersection exists if the distance is less than or equal to the radius
 - . only a few floating point operations are required for spheres centered at the origin
 - . **bounding boxes**
 - typically fit better than bounding spheres
 - translations and rotations are used to make the ray coincident with the z-axis
 - . the 3D bounding box test becomes a 2D sign test
$$x_{\min}x_{\max} < 0 \text{ and } y_{\min}y_{\max} < 0$$
 - rays intersect the transformed box iff the x and y boundaries of the box contain the ray

- . **optimization techniques for raytracing, cont.**
 - **selection of bounding volumes**
 - . **two criteria to satisfy**
 - **tight fit**
 - . **void regions may be estimated by considering the orthographic projections (in the direction of the ray) of the enclosed object and the bounding object**
 - **efficient intersection calculation**
 - . **selection attempts to minimize the product of intersection calculation complexity and the volume of the void regions**
 - . **(at least three) sets of parallel planes can be made to fit convex hulls arbitrarily tightly**

- . **optimization techniques for raytracing, cont.**
 - **use of coherence in ray-tracing**
 - . **image coherence**
 - **adjacent pixels are likely to be the same; the image could be generated with 8x8 macropixels**
 - . **object coherence**
 - **relationships between objects in the environment are considered**
 - . **only objects close to the ray are considered**
 - **ray coherence**
 - . **in many scenes, groups of rays follow virtually the same path from the eye and hence have virtually the same object intersection tree**
 - . **the current ray is checked against the object intersected by the previous ray**
 - **a check of all other objects in the scene is necessary only if the current ray does not intersect the object**
 - . **a "cylinder of security" about the ray is used to detect a closer object**

- . **optimization techniques for raytracing, cont.**
 - **cone tracing**
 - . **the scene is area sampled using a cone rather than a thin ray**
 - . **the radius of the cone at the distance of the virtual screen is one pixel**
 - . **small details do not "fall between the cracks"**
 - . **two complex calculations are required**
 - **the intersection between the cone and the object to be ray-traced**
 - **determination of the fraction of the cone that is blocked by the object (for antialiasing)**

- . optimization techniques for raytracing, cont.
 - beam-tracing
 - . instead of tracing rays, beams approximated by pyramidal cones are traced through a scene consisting of polygonal objects
 - . a recursive beam tracer is used
 - the viewing pyramid is the initial beam
 - beam-surface intersections are computed using 2D polygonal set operations
 - . polygons in a depth-sorted list are searched
 - . the first visible polygon is found by intersecting the beam with the first polygon in the list
 - . the beam tracer builds a beam tree which is similar to a classic ray tree
 - . secondary beams have polygonal cross section
 - for each beam-polygon intersection, new beams are created for reflected and transmitted swaths of light
 - recursion terminates when
 - . maximum tree depth is reached
 - . the intensity contribution of the branch falls below the contribution threshold
 - . the size of the polygon falls below some threshold

- . **optimization techniques for raytracing, cont.**
 - **caching**
 - . **the path of the last ray set is used to predict the path of the current ray set**
 - . **the ray tree is a kind of cache**
 - **a cache hit occurs if the next ray intersects the same object**

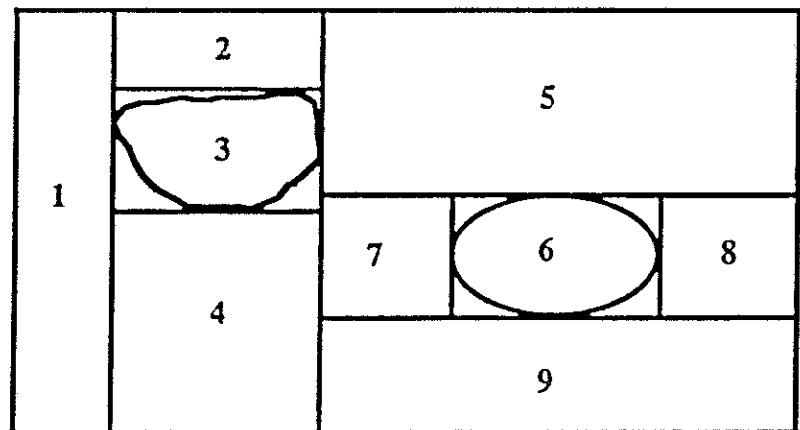
- . **space division for ray-tracing**
 - **space is decomposed into a set of disjoint volumes containing a list of surfaces contained within each volume**
 - **four approaches**
 - . **space decomposition with an octree**
 - . **use of a binary space partitioning tree (BSP tree)**
 - . **digital differential analysis**
 - . **use of CSG models**

- **space decomposition with an octree**
 - . **when a ray is started, the "compartment" within which it originated is determined**
 - . **the ray is followed and compared against the objects it hits in this compartment; two cases:**
 - **one or more object are intersected**
 - . **the color of the closest is returned**
 - . **the tracing of the ray terminates**
 - **no object is intersected**
 - . **the process is repeated with the projection of the ray into the next compartment**
 - . **octree subdivision, as usual, is based on the complexity of the scene**
 - . **an object is intersected with each of the six planes bounding a voxel**
 - **if any point of intersection is within a side of the voxel, the object is kept**

- preprocessing with a binary space partitioning tree (BSP tree)
 - . space tracing recursively subdivides space with respect to three planes parallel to the coordinate axes
 - nonterminal nodes contain the identification of a partition plane which divides space into two half spaces
 - terminal nodes may be empty nodes or box nodes
 - . a box node describes a cubical space region and contains a list of primitive objects intersecting the box

- **digital differential analysis**
 - **the next voxel in the path is determined using a data structure which is a 3D extension of the raster grid**
 - **pixels become voxels (orthogonal cuboidal cells)**
 - **the volume along the path is found using a 3D incremental line drawing algorithm (a 3DDDA)**
 - **when applied along a ray direction, all three indices of a cell are identified**
 - **the algorithm is an efficient tool for identifying rectangular pixels pierced on a raster grid**

- use of CSG models
 - space is arbitrarily subdivided, based on scene configuration
 - subdivision is simplified by replacing each primitive by its bounding box
 - two steps in spatial subdivision
 - in 2D, the infinite plane containing the virtual screen is chosen as the specific plane for a subdivision
 - each bounding volume is projected onto the plane, according to the position of the virtual camera to obtain a rectangle which is decomposed into its four line segments
 - these line segments are used to partition the screen



- . **two steps in spatial subdivision, cont.**
 - . **a segment is selected from among the set of available segments**
 - **the line passing through this segment determines the binary partition of the plane**
 - **the remaining segments are distributed in the two regions created**
 - **the process is applied recursively until all segments have been processed**
 - . **each part of the plane may be empty or may contain primitives**
 - **in 3D, each nonempty region then is subdivided along the third dimension**
 - **information on connectivity is stored in each region so as to pass from one region to another, following a ray**
 - **a primitive may be distributed between several regions**
 - . **the same ray-primitive intersection may be computed several times**
 - . **to avoid redundant calculations, a record is kept of previous ray-primitive intersections**

- . **a partitioning environment to accelerate the shadow testing operation**
 - **shadow testing in ray-tracing is the most computationally expensive operation because each object must be tested to see if it occludes each light source for every ray-intersection point**
 - **the light buffer method reduces shadow testing time**
 - . **a procedure partitions the environment with respect to the position of each light source**
 - . **another procedure tests if a given point is in the shadow by using this partitioned definition of the environment**

- **a partitioning environment to accelerate the shadow testing operation, cont.**
 - **a light buffer is a cube-shaped frame surrounding a light source**
 - **each face is associated with one of the orthogonal axes within the environment and has a gridwork of cells (squares)**
 - **each cell is a small "window" through which the light can "look;" it contains a list of object surfaces which can be seen from the light**
 - **object surfaces "seen" within grid squares can be determined by casting all polygons onto each face of the light buffer cube**
 - **a possibly visible polygon is determined by the Cohen-Sutherland algorithm**
 - **polygons facing away from the light are culled**
 - **remaining polygons are projected onto the light buffer face**

- . **a partitioning environment to accelerate the shadow testing operation, cont.**
 - **during shadow testing, the light buffers are accessed for each view ray-intersection point**
 - . **a list of all surfaces that might occlude the light ray is retrieved**
 - . **each polygon in the list is tested until an occlusion is found or until the depth of the potentially occluding polygon is greater than the intersection point**