CS488/688

Sample Exam Questions

University of Waterloo
School of Computer Science
Computer Graphics Lab

August 31, 2017

This is a set of questions covering the course material. They have been asked on quizzes, midterms, and finals in past courses, and the faculty draw liberally upon them for ideas in making up new questions. Some of these questions are answered in the text, some in lecture, some in the materials on reserve in the EMS Library. Some will require that you think on your own.

Occasionally a question contains an indication of the last time it (or a similar question) appeared on a CS488/688 exam. You should use this as a guide to the "relevance" of the various questions to the course material being taught this term. Because not all topics are covered every term and because the hardware support for the course has ranged from line printers to monochrome HP 2648A graphics terminals, to limited-colour Telidon decoders and DEC Gigi terminals, to Raster Tech frame buffers, to SGI 3000-series IRIS workstations, and now to SGI Indigo workstations, a few questions are no longer entirely up to date. They are, however, updated when they become the basis of questions used on exams, and thus are worth looking at during the term.

1 Animation

1.1 Animation Techniques [Last Used: Fall 2002 Final]

Describe how you would animate the following actions, and what problems you would encounter using your method.

1. A ball bouncing off the ground.
2. A dog running across a road.
3. A couple dancing.
4. A fireworks display.
5. An octopus catching fish with all 8 arms out of a passing school of fish.

1.2 Keyframing [Last Used: Winter 1998 Final]

Suppose you had decided for your final project to extend your assignment 3 puppet into a key frame system.

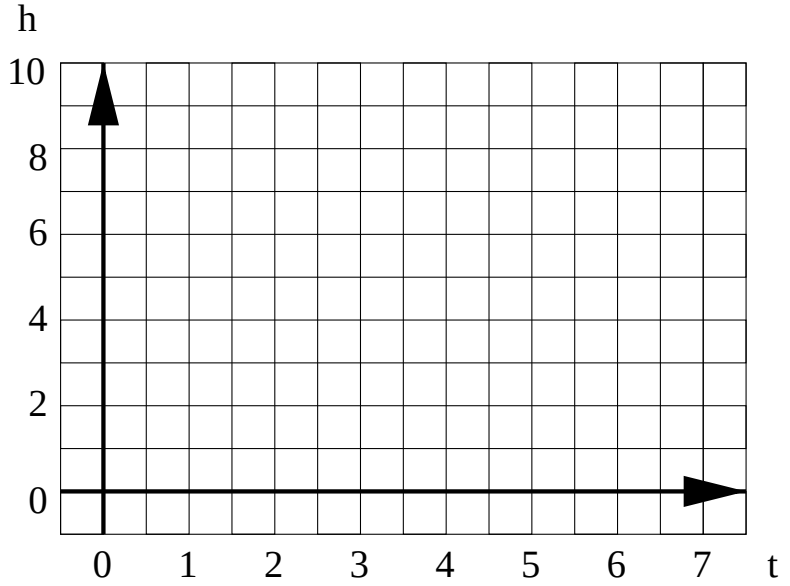
1. The current restrictions on the motion of the puppets shoulders and hips are too restrictive. Describe the additional degrees of freedom you would need to add to get reasonable motion, discuss what changes you would need to make to your data structure to support these degrees of freedom, and suggest a reasonable UI for manipulating the arms and legs given these new degrees of freedom.

2. What additional functionality would you have to add to the program to support a key frame animation system?

1.3 Animation and Splines [Last Used: Fall 2009 Final]

Suppose we wish to animate a bouncing ball. We are given keyframes for the ball's motion. The key frames are at times T={0,1,2,3,4,5,6} with associated height values of H={0,5,8,9,8,5,0}.

a) Plot the graph of height versus time using linear interpolation between key frames.



b) Give an equation for the height h(t) over the interval t=4...5 using linear interpolation.

c) In general, suppose we are given times T={t_0, t_1} and associated heights H={h_0,h_1} what is the equation for h(t) in the interval [t_0,t_1] using linear interpolation?

d) Suppose we wish to achieve a smoother animation by using Catmull-Rom interpolation. What would the parametric derivatives of the interpolated curve be at t=4 and t=5?

e) Give 2D cubic Bezier control points for a Hermite interpolation given two sequential points (t_0,h_0) and (t_1,h_1) with associated derivatives v_0 and v_1.

- P_0 =

$P_1 =$

$- P_2 =$

$- P_3 =$

1.4 Euler Angles [Last Used: Winter 2000 Final]

Euler angles are one method of representing orientations, with the orientation represented as a rotation around each coordinate axis. While common in computer graphics, Euler angles have several problems. To address these problems, people often use quaternions.

For this question, we will use the following representation of Euler angles: With $c_a = \cos(\theta_a)$ and $s_a = \sin(\theta_a)$, the matrix for Euler angles is

$$\begin{aligned} R(\theta_x, \theta_y, \theta_z) &= \begin{bmatrix} c_y c_z & c_y s_z & -s_y & 0 \\ s_x s_y c_z - c_x s_z & s_x s_y s_z + c_x c_z & s_x c_y & 0 \\ c_x s_y c_z + s_x s_z & c_x s_y s_z - s_x c_z & c_x c_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= R_x(\theta_x) R_y(\theta_y) R_z(\theta_z), \end{aligned}$$

where $R_x(\theta_x)$, $R_y(\theta_y)$ and $R_z(\theta_z)$ are the standard rotation matrices.

The following Euler angles and quaternion represent (to 3 digits of precision) the same orientation:

$$\begin{aligned} \theta_x &= 10, \quad \theta_y = 90, \quad \theta_z = 30 \\ q_1 &= 0.696 - 0.123\vec{i} + 0.696\vec{j} + 0.123\vec{k} \end{aligned}$$

1. (3 pts) Give a second set of distinct Euler angles $\beta_x, \beta_y, \beta_z$ that represent the same orientation. Your angles must be in the range [0, 360] degrees.

   $\beta_x =$

   $\beta_y =$

   $\beta_z =$

2. (2 pts) Give the quaternion q_2 corresponding to your Euler angle solution in part (a) of this question.

   $q_2 =$

1.5 Animation Techniques [Last Used: Winter 2006 Final]

You are working on a scene in a movie that will incorporate ray traced CG characters of a Vice President shooting (with buckshot) an apologetic lawyer in slow motion. (Buckshot is a type of "bullet" that consists of many small pellets that spread out as they fly through the air.)

Describe what part of the animation each of the following techniques could be used for, and suggest and justify the use of a fifth technique that would be useful in creating this animation.

1. Keyframing

2. Particle systems

3. Motion blur

4. Inverse kinematics

Fifth technique:
No point bonus question: In Texas, a hunting license costs $100. How much does a permit to shoot a lawyer cost?

1.6 Animation—Quaternions [Last Used: Winter 2007 Final]

1. A quaternion is a value of the form $a + ib + jc + kd$ with real numbers a, b, c, and d, and imaginaries i, j, and k. What property of the coefficients must hold in order for the quaternion to represent an orientation in 3D?

2. Describe one of the problems with Euler angles that are solved by using quaternions to represent orientations.

3. Explain or demonstrate with an example why the entries in a transformation matrix should not be interpolated directly.

4. Now suppose we are animating a square with side length 1, centred at the origin. At time $t = 0$, the square is displayed with this 2D transformation matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then, at time $t = 1$, the square is displayed with this matrix:

$$\begin{bmatrix} \sqrt{2} & -\sqrt{2} & 5 \\ \sqrt{2} & \sqrt{2} & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

Derive a single matrix, whose entries are functions of time t, which animates between the states represented by these matrices. The square's position, orientation and size should all change smoothly and minimally. The matrix should never distort the square into another shape. (Note: you can't just animate the matrix entries using linear interpolation. But you can still write down a matrix parameterized by t that carries out the animation by considering how the square ought to move.)

2 Antialiasing

2.1 Stochastic Sampling [Last Used: Winter 1995 Final]

What are the advantage(s) of stochastic sampling over other antialiasing methods?

2.2 Super- and Stocastic-Sampling [Last Used: Winter 1996]

Super-sampling and stocastic sampling are two methods for anti-aliasing.

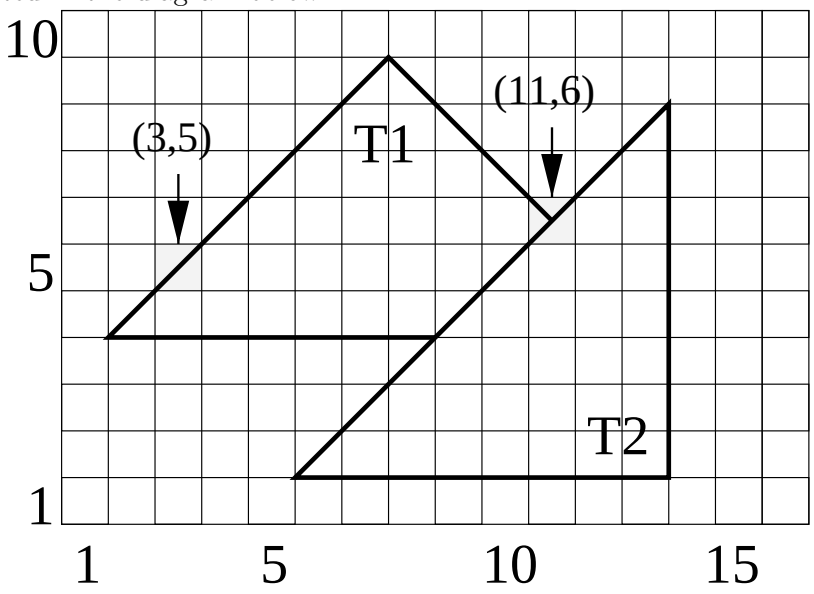
1. Describe both methods.
2. With both techniques, we are still taking a discrete sampling of a continuous signal. Describe why/how each method reduces aliasing artifacts.

2.3 Aliasing and Anti-aliasing [Last Used: Winter 1998 Final]

- Describe 3 aliasing artifacts present in Computer Graphics.
- Describe one method of anti-aliasing.

2.4 Aliasing [Last Used: Winter 2000 Final]

Suppose we wish to rasterize two triangles T1 and T2 given screen space coordinates for their vertices as indicated in the diagram below.



a) If both triangles are black on a white background, a naive approach would be to fill in each pixel that contains any piece of either triangle. What is the common name for the aliasing effect that would result?

b) A *triangle strip*, often used to quickly render surfaces, is a sequence of triangles joined at shared edges. What is inefficient about rasterizing a triangle strip by filling the triangles one-at-a-time using the method in (a)?

c) Suppose T1 has RGB colour (16,16,16), T2 has RGB colour (0,0,16) and the background has RGB colour (0,0,0). If the rasterization was antialiased using the obvious approach, what colour should pixel (3,5) be shaded? What colour should pixel (11,6) be shaded?

d) Analytic antialiasing is difficult in ray tracing. Name an alternative approach and two possible variations.

3 Backface Culling

3.1 Perspective and Culling [Last Used: Spring 1992 Final]

We may remove backfacing triangles from a polyhedral object with triangular faces by transforming each set of triangle vertices P_0, P_1, and P_2 into perspective space to obtain transformed points P'_0, P'_1, and P'_2, then computing a normal to the "prime" triangle, and finally checking the sign of the z component of this normal.

(a) Explain why this computation correctly allows us to identify backfacing triangles.

(b) Show, by the following example, that we cannot take the triangle normal in world space and apply the perspective transformation to it to achieve any valid backfacing test in perspective space.

EXAMPLE

The eye is at position $(0, 0, -d, 1)$ looking in the positive z direction at the origin. The x-y plane is the projection plane. The world-space triangle has vertices $P_0 = (1, 0, 0, 1)$, $P_1 = (0, 1, 0, 1)$, and $P_2 = (0, 0, 1, 1)$. Develop a perspective matrix; transform the vertices by the matrix and divide by the resulting 4-th coordinate to produce the P'_i points. Take a normal to the world-space triangle and subject it to a perspective transformation with the same matrix (no division — 4-th coordinate is zero, as it should be for a vector). Show that this resulting vector is not a normal for the "primed" triangle.

Conventions

All points in 2D have the form $(x, y, 1)$ and in 3D the form $(x, y, z, 1)$.
All vectors in 2D have the form $(x, y, 0)$ and in 3D the form $(x, y, z, 0)$.
All 4×4 transformation matrices (*including perspective*) in 3D act on vectors only with regard to to their upper-left-hand 3×3 submatrix.
That is,

$$\begin{bmatrix} a & e & m & r \\ b & f & n & s \\ c & g & p & t \\ d & h & q & u \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \quad \text{is equivalent to} \quad \begin{bmatrix} a & e & m & 0 \\ b & f & n & 0 \\ c & g & p & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

4 Clipping

4.1 Frustum Clipping [Last Used: ?]

Define the frustum of vision to be centered on the z-axis with the eyepoint at $(0, 0, -d)$ and with the window on the x, y-plane having its center at $(wcx, wcy, 0)$, width $2 * wsx$, and height $2 * wsy$. The far clipping plane is to be parallel to the window and intersect the z-axis at $(0, 0, f)$.

(a) Give the equations (in the window coordinate system) of the six clipping planes. Adjust the signs in your equations (if necessary) so that a positive value denotes "visible" and a negative value denotes "invisible".

(b) What is the perspective matrix appropriate for this formulation of the problem. Explain your answer!

4.2 Homogeneous Clipping [Last Used: ?]

Consider our version of the clipping algorithm in which we clip against the $x = w$, $x = -w$, $y = w$, $y = -w$, $z = w$, and $z = 0$ planes.

(a) What are the column vectors associated with the six planes (be careful to orient the planes correctly so you can answer (b) & (c) below)?

(b) Show that the window edge coordinates WEC(I) of a point S are actually the dot products $S \cdot P(I)$ $(I = 1, 6)$ where $P(I)$ is the vector for the Ith plane.

(c) In light of (a) & (b) above, explain why the Cohen-Sutherland "out" codes are correct.

4.3 Sutherland-Hodgman Clipping [Last Used: Fall 1986 Final]

Note: This is actually *two* problems — do it once with the (x, y) coordinates and once with the (x', y') coordinates. You will get *different* answers.

Assume that a 2-D reentrant polygon clipper clips vertices against the *left, top, right*, and *bottom* edges of the window (in that order) and that all coordinates are represented in clipping space (i.e., after all of the modeling and viewing transformations have been applied to the vertices).

Consider the polygon whose vertices are given by the following coordinates.

label	x	y	x'	y
P_1	-2.0	+2.0	-2.0	+2.0
P_2	+2.0	+2.0	+2.0	+2.0
P_3	+2.0	0.0	+2.0	0.0
P_4	0.0	0.0	0.0	0.0
P_5	-2.0	-2.0	-2.0	+2.0

(a) Draw a diagram showing the 2-D clipping boundaries of the window and the original vertices of the polygon with their labels.

(b) Assume that the clipper works as in the Sutherland and Hodgman paper so that it passes vertices from one stage of the clipping pipeline to the next as soon as `Output` is called during a clipping stage. Remember to include vertices generated by calls to `WClose` as well as those to `Clip` itself. Also assume that all arithmetic calculations are *exact* (i.e., *no* roundoff error occurs).

Show the sequence of vertices (*in order*) generated by the 2-D reentrant polygon clipper as *input* to each stage of the clipper (the procedure `Clip` for each of the edges). Label the new vertices *a, b, c, . . .*, in the order in which they are generated by the clipper. Indicate the approximate positions and the (exact) coordinate values of the new vertices by including them in your diagram for Part (a) on the previous page.

The inputs to the first stage (`Clip[left]`) and the last stage (`Display`) have been provided for you.

Clipping Stage	Input Vertices
`Clip[left]:`	`P1, P2, P3, P4, P5`
`Clip[top]:`	
`Clip[right]:`	
`Clip[bottom]:`	
`Display:`	`c, P4, d, e, f`

4.4 Anticlipping [Last Used: Winter 1986 Midterm]

Discuss the changes necessary to draw only the portion(s) of a line segment that is(are) *not* contained within the clipping region. You may assume that points lying on the border of the clipping region are to be drawn. Show the outcode and window edge coordinate definitions for this case. Be sure to include the conditions for trivial acceptance and rejectance.

4.5 Cross Product Rejection [Last Used: ?]

The magnitude of the cross product between two vectors, v and w, is given by

$$|v \times w| = |v||w| \sin(\theta).$$

The direction of the cross product perpendicular to the plane of v and w and points in the direction of the right hand's thumb when the right hand's fingers are curled in the v,w-plane from v to w. The magnitude of the cross product is just the area of the parallelepiped whose two adjacent sides are the vectors v and w.

Show that, if a clipping window is defined in the x,y-plane, an acceptance/rejection test and window-edge coordinates for points can be derived from the cross product.

4.6 Polygon Splitting [Last Used: Fall 1990]

You are given a sequence of three or more vertices p_0, p_1, p_2, ... defining a convex polygon O in 3-space, and the equation $Ax + By + Cz + D = 0$ of an arbitrarily oriented plane P. Describe an algorithm that splits O into two polygons O_1 and O_2, with O_1 lying on or to one side of P, and O_2 lying on or to the other side of P.

Clearly written and commented pseudo-code or English is desired here, not hacker's C. Be careful to explain and justify how each step of the computation is performed.

4.7 Sutherland-Hodgman Clipping [Last Used: Winter 1994 Final]

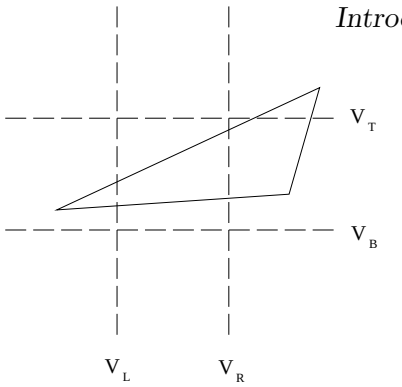
The Sutherland-Hodgman polygon clipping algorithm is described in the text in terms of a single clipping boundary. In practice we usually want to clip against some viewport. This involves passing the polygon through a series of clippers. For example:

```
POLYGON->TopClipper->BottomClipper->LeftClipper->RightClipper->RESULT
```

If we need to clip an edge at a boundary an expensive intersection calculation must be performed.

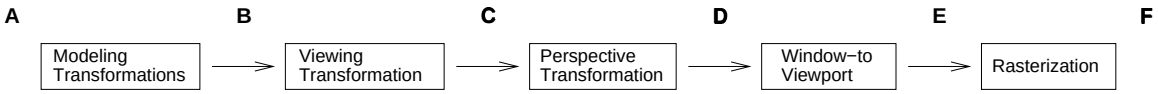
(a) Show that the number of intersection calculations required is dependent on the order of the clippers. HINT: Consider the diagram below.

(b) Prove, or give a counter-example to the statement : "In general there is a BEST ordering of the clippers."



4.8 Pipeline [Last Used: Fall 1996 Midterm]

Excluding clipping, the stages of the three dimensional graphics pipeline are given by the following diagram:



At which point (**A-F**) do we normally perform clipping and why do we clip there rather than at one of the other steps? (Each point is considered to be between two steps of the pipeline except **A** and **F**, which are before and after the entire pipeline, respectively. Point **B**, for example, occurs after modeling transformations and before viewing transformations.)

4.9 Clipped Convex Polygon Vertices [Last Used: Winter 1993 Midterm]

Suppose you clip a convex polygon with n vertices to a square window. What is the maximum number of vertices that the clipped polygon can have?

4.10 Clippped Concave Polygon Vertices [Last Used: ?]

Suppose you clip a concave polygon with n vertices to a square window. What is the maximum number of vertices the clipped polygon might have?

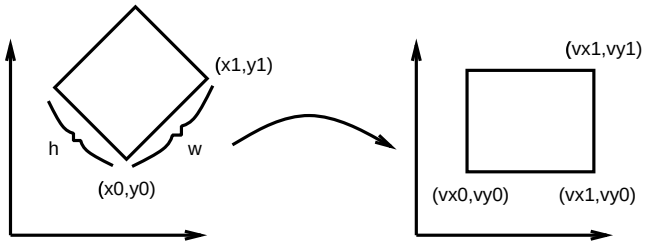
4.11 Degenerate Polygons [Last Used: ?]

In class, we showed that clipping a concave polygon to a rectangular window may result in degeneracies. State a simple test of the clipped polygon to determine if it has degeneracies.

4.12 Diamond-shaped Window [Last Used: Fall 2004 Midterm]

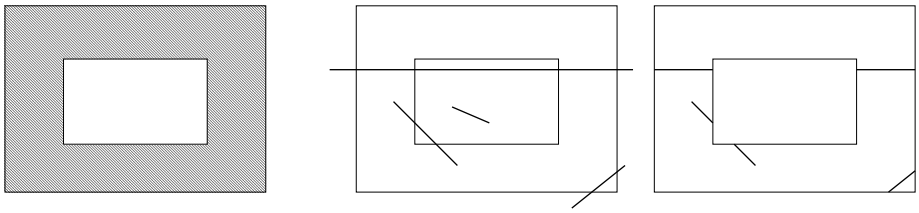
In our window-to-viewport mapping, we assumed that both the window and viewport were rectangular in shape and aligned with the coordinate axes. In the following, assume that the viewport is rectangular and aligned with the coordinate axes. Suppose that the window is rectangular, but not aligned with the coordinate axes. To simplify the problem, assume the edges of the window form a 45 degree angle with the coordinate axes, with one corner at $(x0, y0)$, a second corner at

$(x1, y1)$, and with the window having a width of w and height of h as shown in the figure below. Further, assume we want to map this window to a viewport that is fully defined by the two points $(vx0, vy0)$ and $(vx1, vy1)$. The mapping should map $(x0, y0)$ to $(vx0, vy0)$ and map $(x1, y1)$ to $(vx1, vy0)$. Explain how we would perform the window-to-viewport mapping.



4.13 Clip to Window with Hole [Last Used: Spring 2000 Midterm]

Suppose we wish to clip lines to a rectangular window with a rectangular hole in it:

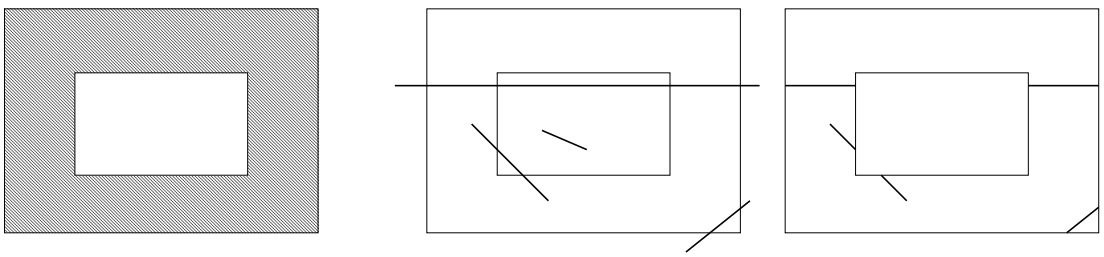


The shaded region is our "window" | A set of lines before clipping | A set of lines after clipping

Call the outer rectangle R with vertices $r_0,\ r_1,\ r_2$, and r_3, and call the inner rectangle H with vertices $h_0,\ h_1,\ h_2$, and h_3. You may assume that h_i is the corner closest to r_i. Describe an algorithm for clipping lines to this object.

4.14 Clipping (Variation of Previous) [Last Used: Spring 1997 Midterm]

Suppose we have a rectangular window with a triangular hole in it:

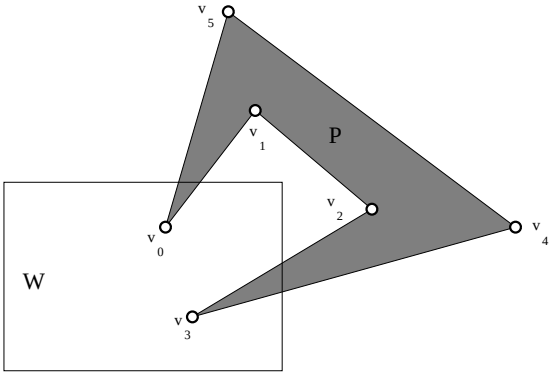


The shaded region is our "window" | A set of lines before clipping | A set of lines after clipping

- Present the formulas of all the intersection computations needed to clip lines to this window.
- Is an outcode scheme possible for this "window"? If not, explain why. If so, propose one.

4.15 Specific Clipping Problem [Last Used: Winter 2006 Midterm]

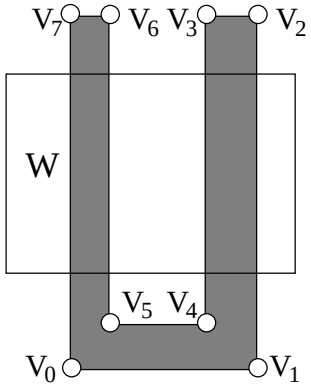
Suppose we clip polygon $P = (v_0, \ldots, v_5)$ to window W in the following situation using the repeated half-plane clipping algorithm:



Label any new vertices resulting from clipping in the figure above as $p_0, \ldots, p_i$. Give an ordered list of the vertices of the clipped polygon. You may chose any vertex as the initial vertex.

4.16 Specific Clipping Problem [Last Used: Winter 1998 Midterm]

Suppose we clip polygon $P = (v_0, \ldots, v_7)$ to window W in the following situation using the repeated half-plane clipping algorithm:



Label any new vertices resulting from clipping in the figure above as $p_0, \ldots, p_i$. Give an ordered list of the vertices of the clipped polygon. You may chose any vertex as the initial vertex.

4.17 Clipping [Last Used: Winter 1996 Final]

1. When designing fast algorithms, although it is important to handle the common cases quickly, it is acceptable to spend more time on the less common cases.

   In lecture, to clip a single line segment to a window, we performed repeated half-space clipping. Describe how we clip a line segment to a single half-space, noting how the "common cases" are handled quickly relative to the "less common cases".

   Note: The question is *not* asking about outcodes.

2. If we clip a polygon to a window, it is inadequate to merely perform repeated half-space clipping on each polygon line segment independently. Give an example illustrating why independent clipping of polygon line segments may fail to correctly clip a *convex* polygon. A picture alone is inadequate; you should also give a short answer describing the picture and why it illustrates the problem, noting both the incorrectly clipped polygon and what the correctly clipped polygon should be.

5 Colour

5.1 Colour Coordinate Systems [Last Used: Spring 1986 Final]

Name three colour coordinate systems and describe their defining axes. For each system, indicate the shape of the region containing the colours they can define.

5.2 Additive and Subtractive Colour [Last Used: ?]

Explain the difference between additive and subtractive colour mixing.

(a) According to the opponent-colour theory of vision, what information is passed on to the brain from the eye and how is it generated from cone excitations?

(b) Explain the relevance of opponent-colour theory to computer graphics.

5.3 Colour Space Mapping [Last Used: Winter 1986 Final]

The colours D, E, F are defined in terms of the R, G, B primaries as:

$$D = 1.0R + 0.5G \quad (1)$$
$$E = 1.0R - 0.5B \quad (2)$$
$$F = 1.0R \quad (3)$$

(a) What are D, E, F values that produce an equivalent stimulus to the eye as $R = 0.0$, $G = 1.0$, and $B = 1.0$?

(b) Write the equations that map from three values in DEF colour space into RGB colour space.

(c) Can the colours D, E, F be considered as primaries? Justify your answer.

5.4 Subtractive Colour [Last Used: Fall 1990]

Explain what is meant by the statement that "subtractive colour is inherently multiplicative."

5.5 Exact Colour [Last Used: Fall 1990]

Suppose that we are designing a modelling program for a museum that will use images generated by our software to evaluate architectural and lighting proposals for the interior of a new building, and that it is particularly important to model accurately the appearance of the paintings that will hang in the building. Explain why representing object colour by RGB triplets would probably not be satisfactory.

6 Geometry

6.1 Affine combination [Last Used: Fall 2009 Final]

Which of the following are valid expressions in affine geometry, where P, Q, R are points, $\vec{v}$ is a vector, and $B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i$.

Expression	Valid?	Expression	Valid?	Expression	Valid?
$(P+Q)/2$		$2P - Q$		$Q - 2P$	
$P - 2Q + R$		$P - 20\vec{v}$		$P - 2Q + \vec{v}$	
$(1-t)P + tQ$		$cos^2(t)P + sin^2(t)R$		$\sum_{i=1}^{n} B_i^n(t)P_i$	
$(P+Q)/R$		$R/(P \cdot Q)$		$\vec{v} \times (P - Q)$	

6.2 Spaces [Last Used: Winter 2007 Midterm]

Here, in alphabetic order, are the four kinds of spaces we constructed to understand the geometry of computer graphics:

1. Affine space
2. Cartesian space
3. Euclidean space
4. Vector space

Write them in order so that each space is an extension of the previous one. Between every two spaces, write down what was added to the first space to get the second one.

Which of these spaces is the simplest one in which we can talk about parallel lines?

Which of these spaces is the simplest one in which we can talk about perpendicular lines?

6.3 Planar Polygon Test [Last Used: ?]

Given a sequence of n points $P_1, \ldots, P_n$ in in three dimensions, describe a boolean function $Planar(P_1, \ldots, P_n)$ that returns *true* if and only if the points lie in a common plane. If you give pseudo-code, comment the code to indicate what it is doing.

6.4 Geometric Meaning [Last Used: Fall 2010 Final]

Suppose P, Q, R are non-colinear points in a Euclidean three space $\mathcal{E}$, and $\vec{v}$ and $\vec{w}$ are vectors defined as $\vec{v} = P - Q$ and $\vec{w} = Q - R$. Let $\mathcal{A}$ be an affine transformation whose domain is $\mathcal{E}$ and whose range is an arbitrary Euclidean space. Let $< \vec{u}, \vec{w} >$ denote the dot product of $\vec{u}$ and $\vec{w}$, let the cross product of these two vectors be denoted as $\vec{u} \times \vec{w}$, and let $|\vec{v}|$ denote the length of the vector $\vec{v}$.

Each expression in the following table may or may not have geometric meaning, and if an expression does have geometric meaning, the expression may or may not be an equality. You should check the box the best describes the equality.

- Check the 'ng' box if the expression potentially has no geometric meaning;

- Check the '=' box if the expression will always have geometric meaning, and if the expressions on either side of the '?=' will always be equal;
- Check the '$\neq$' box if the expression will always have geometric meaning, but the expressions on either side of the '?=' might not be equal.

Expression			=	$\neq$	ng
P	?=	$Q + \vec{v}$			
$\mathcal{A}(P)$	?=	$A(Q) + A(\vec{v})$			
$\mathcal{A}(P + Q)$	?=	$\mathcal{A}(P) + \mathcal{A}(Q)$			
$\mathcal{A}(\vec{v} + \vec{w})$	?=	$\mathcal{A}(\vec{v}) + \mathcal{A}(\vec{w})$			
$\mathcal{A}((1-t)P + tQ)$	?=	$(1-t)\mathcal{A}(P) + t\mathcal{A}(Q)$			
$< \vec{v},\ \vec{v} \times \vec{w} >$	?=	0			
$< \mathcal{A}(\vec{v}),\ \mathcal{A}(\vec{v} \times \vec{w}) >$	?=	0			
$\lvert\vec{v}\rvert$	?=	$\lvert\mathcal{A}(\vec{v})\rvert$			
$R + \mathcal{A}(\vec{v} + \vec{w})$	?=	$R + \mathcal{A}(\vec{v}) + \mathcal{A}(\vec{w})$			
$\mathcal{A}(P) + \mathcal{A}(\vec{v}) + \mathcal{A}(\vec{w})$	?=	$\mathcal{A}(R)$			

6.5 Geometric Ambiguity [Last Used: Spring 1997 Midterm]

We are given a pair of 2 dimensional points, a 2 dimensional vector, and a matrix:

$$P = \begin{bmatrix} a \\ b \\ 1 \end{bmatrix}, \quad Q = \begin{bmatrix} c \\ d \\ 1 \end{bmatrix}, \quad \vec{v} = \begin{bmatrix} r \\ s \\ 0 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 2 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1. Give at least three geometric interpretations of $\mathbf{M}P$. Sketch a picture for each interpretation.
2. For each of your interpretations, in order, what will be the length of $\mathbf{M}\vec{v}$?
3. For which of your three interpretations of $\mathbf{M}P$ will it be meaningless to talk about the distance between Q and $\mathbf{M}P$? For which interpretation will the distance between Q and $\mathbf{M}P$ be the same as the distance between Q and P? For which interpretation will the following expression be meaningful?

$$Q - \mathbf{M}P \equiv \begin{bmatrix} c \\ d \\ 1 \end{bmatrix} - \begin{bmatrix} 2 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ 1 \end{bmatrix}$$

4. What information must we know to fully interpret the matrix product $\mathbf{M}P$ unambiguously?

6.6 Geometric [Last Used: Winter 2007 Final]

The following questions are in the context of an OpenGL rendering application in which back-face culling and depth testing is always enabled. A user has modelled various objects but when they try to use them with this system they run into a number of problems. Try to diagnose each of the problems they encounter.

1. The user has modelled a symmetric object by only modelling the left side and then using a reflection to model the right side. The left side renders correctly but the right side does not. What might be going on?

2. The user has modelled an object, but when they render it they see strange jagged stripy patterns on the surface, and these patterns change with the viewpoint. The problem is much worse when the model is far away. There is no texture on this model, and aliasing problems have been ruled out. What might be going on?

3. The user has modelled an object, but when they render it, parts that should be visible through holes in the surface seem to be missing. What might be going on?

4. The user has modelled an object, but when they render it, the edges between certain polygons occasionally have missing pixels and holes in the surface through which the color of the background is visible. The user insists that all vertices lie on the boundaries of other polygons in the mesh and shared parts of edges are aligned so the mesh should be watertight. What might be going on?

7 Hidden Surfaces

7.1 Visual Complexity [Last Used: ?]

(a) Distinguish between the *complexity* of a scene and its *visual complexity*.

(b) Characterize the running time of Robert's, Warnock's, Watkin's, and the scan line depth buffer algorithms in terms of these two complexities.

(c) Define as many types of "coherence" as you can. For each algorithm above indicated which types of coherence are used in solving the hidden surface problem.

7.2 Visible Surface Algorithms [Last Used: Winter 1998 Midterm]

This question concerns the following algorithms which address the visible surface problem:

1. The binary space partitioning algorithm (BSP)

2. Warnock's algorithm

3. The Painter's algorithm

4. Backfacing polygon removal

5. Z-buffering

(a) Briefly describe the visible surface problem.

(b) Give the number (or numbers) of the relevant algorithms for the following questions (if none of the algorithms have the sated property, write 'none'):

Which algorithms

A. involve area subdivision?

B. require building a tree?

C. often use extra memory hardware?

D. require significant preprocessing time?

E. require knowing polygon orientations?

F. involve comparing pixel depths?

G. are device independent?

H. are view-independent?

I. can be easily implemented in hardware?

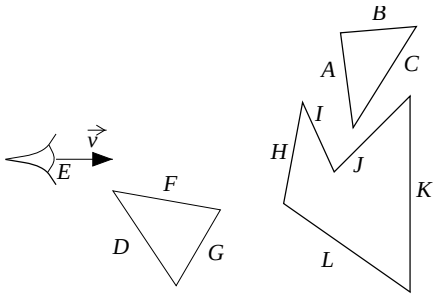
J. which algorithm is clearly insufficient to solve the visible surface problem?

(c) Briefly describe each of the four algorithms.

(d) The Iris provides support for backfacing polygon removal. Give a brief algorithm for determining whether a simple convex polygon's vertices in screen space are oriented clockwise or counterclockwise.

7.3 Backface culling, Painter's algorithm [Last used: Fall 1997 Midterm]

The picture below is an example of the 2D version of hidden surface removal problem. In this problem, we want to render those portions of the lines that are visible to the viewer.



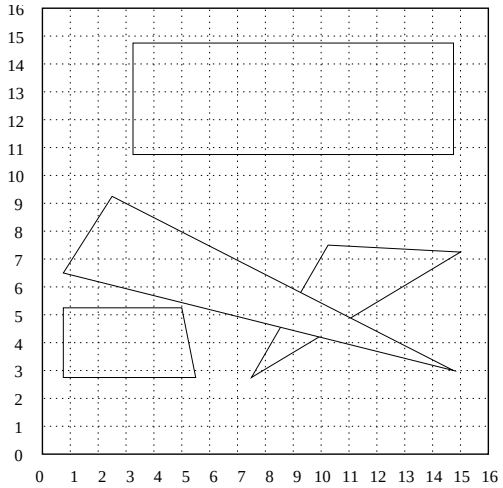
Note: For the purposes of this problem, we are thinking of the eyepoint E as being located at the front of the eye (ie, where the line representing $\vec{v}$ starts).

- Suppose we use backface culling to remove some lines segments before drawing them. Which lines will be drawn after culling? What visual artifacts (due to deficiencies of this hidden line removal algorithm) will we (potential) see?
- Give a simple test (eg, no more than 3 lines of code) to determine whether we should backface cull (remove) a line segment X. Assume that the line segment X has normal $X.n$ and is defined by its endpoint, $X.v1$ and $X.v2$.
- Suppose instead of backface culling that we use the Painter's algorithm. In which order will the line segments be drawn? (Note: This order is not unique; you just have to give any one order that the Painter's algorithm might produce). What visual artifacts (due to deficiencies of this hidden line removal algorithm) will we (potential) see?

7.4 Warnock's Algorithm [Last Used: Fall 1997 Final]

In the picture below, the large square is the region of interest on the screen, the dotted lines show the boundaries between pixels, and the two triangles and the two quadrilateral are polygons that we want to draw on the screen.

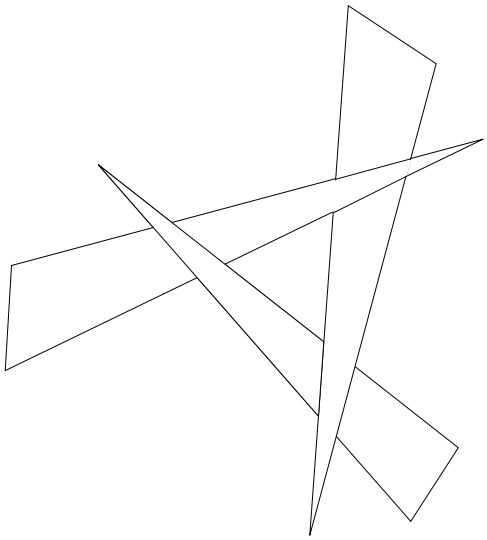
In this figure, indicate the regions into which Warnock's hidden surface removal algorithm will subdivide the screen.



7.5 Painter's Algorithm [Last Used: Fall 2002 Midterm]

Suppose you implement the Painter's Algorithm for hidden surface removal. Assume you have implemented the algorithm for non-overlapping polygons and non-overlapping polygons only (i.e., for when for each pair of polygons at least one of the x, y, or z extents do not overlap).

1. Describe in words what you would have to change/implement in your code to render the following scene:



2. Indicate in the picture what you code from part (a) would do to these polygons, label the polygons, and give an order in which the Painter's Algorithm would render them (the order may not not unique; any correct order is acceptable).

7.6 Z Buffer/Convex Polygons [Last Used: Fall 1990]

(a) Sketch the organization of a z-buffer hidden surface algorithm for convex polygons.

(b) Describe the storage requirements under a reasonable set of assumptions. (State your assumptions!) What is the complexity of this algorithm?

(c) Explain how to reorganize the algorithm so as to completely compute the image on one scan line before proceeding to the next.

(d) Explain how to modify the algorithm so as to completely compute the image at one pixel before proceeding to the next. In what circumstances would this be a desirable thing to do?

(e) Indicate how to modify the algorithm of (c) so as to handle concave polygons.

7.7 Z Buffer vs. Watkins, Ray Tracing [Last Used: Fall 1986 Final]

(a) Define the term depth buffer (z buffer) and explain how a depth buffer can be used to solve the hidden surface problem without explicitly computing polygon intersections.

(b) Does the depth buffer algorithm take advantage of scan line coherence? Explain your answer.

(c) Compare the depth buffer algorithm with Watkin's algorithm for hidden surface elimination. Under what circumstances would one algorithm be preferable over the other? Relative criteria for judgement include time and space requirements, ability to incorporate transparency and anti-aliasing, and ease of programming. Justify your answer.

(d) Keeping in mind what you wrote for part (c), what are the advantages/disadvantages of naive ray tracing as introduced by Turner Whitted in 1979?

(e) Distributed ray tracing was first introduced in 1984 by Cook, Porter & Carpenter from Lucasfilm. How does it differ from the naive ray tracing algorithm? Explain.

(f) Keeping in mind what you wrote for parts (c), (d) and (e), what are the advantages/disadvantages of distributed ray tracing?

7.8 Z Buffer and Translucency [Last Used: Spring 2000 Final]

The standard z-buffer algorithm in the raster subsystem of most workstations goes like this:

```
for each draw-object request {
    for each pixel covered by the object {
        if (the object z is greater than the stored z) {
            replace the color by the object color;
            replace the stored z by the object z;
        }
```

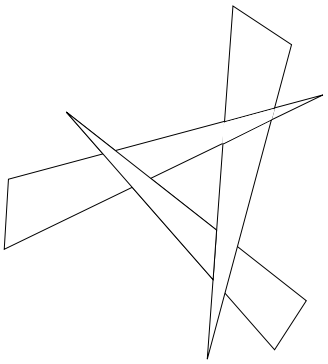
```
}
```

This assumes that objects are opaque. If objects are translucent, then the color of a pixel should be, recursively,

```
factor * front-object-color
+
(1 - factor) * color-resulting-from-all-further-objects
```

where `factor` is a number between 0 and 1 related to the light transmission properties of the front object. Why can't we handle this with the the z-buffer algorithm? Are there any modifications of the z-buffer algorithm that you can think of for handling transparency? If so, describe them briefly. If not, can you say what the insurmountable difficulty with the algorithm is?

An interesting picture to look at while you are thinking is the following:

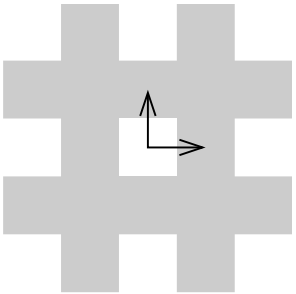


8 Hierarchy

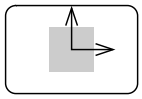
8.1 2D Hierarchical Modeling [Last Used: Winter 2004 Midterm]

In this question, the following notation is used: I is the do-nothing identity transformation, $T(x, y)$ is a 2D translation by vector (x, y), $S(x, y)$ is a non-uniform scale with factors x and y, and $R(\theta)$ is a counter-clockwise rotation about the origin by angle θ (in degrees).

We want to model the following 2D "checker board":

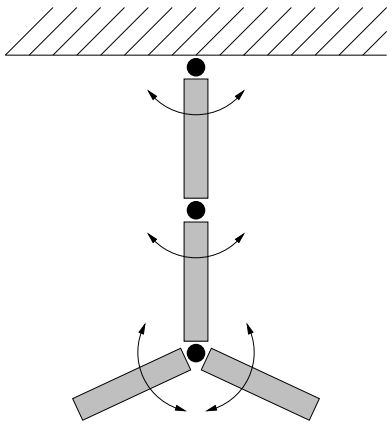


Draw a DAG to model the checker board, with a single root node and a single leaf node containing a unit square centered at the origin as shown below. You should draw your internal nodes as boxes with one of the transformations listed above. The boxes should be connected with arcs; you may use multiple arcs to/from any box. You may use no more than 6 transformations (i.e., you may have no more than 6 internal nodes).



8.2 Hierarchical Swinging Links [Last Used: Fall 1995 Final]

The schematic diagram that follows shows a 4-link jointed structure hanging from the ceiling. Each link has length L, and for simplicity we model each link object using an affine transformation of the line segment $\overline{(0,0,0)(0,0,-L)}$. The black circles indicate pivot (rotation) points (assumed to have no diameter). The rotations are indicated by arrows. We want to simulate this structure as it swings under gravity, but first we have to build it.



Part (a) : Set up a display structure of the sort discussed in lectures (transformation nodes, object nodes, sibling links, child links) that represents this jointed structure.

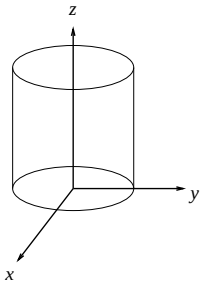
Part (b) : Show how the matrix stack will change as you traverse your data structure.

8.3 Hierarchical Robot Arm [Last Used: Fall 1992 Final]

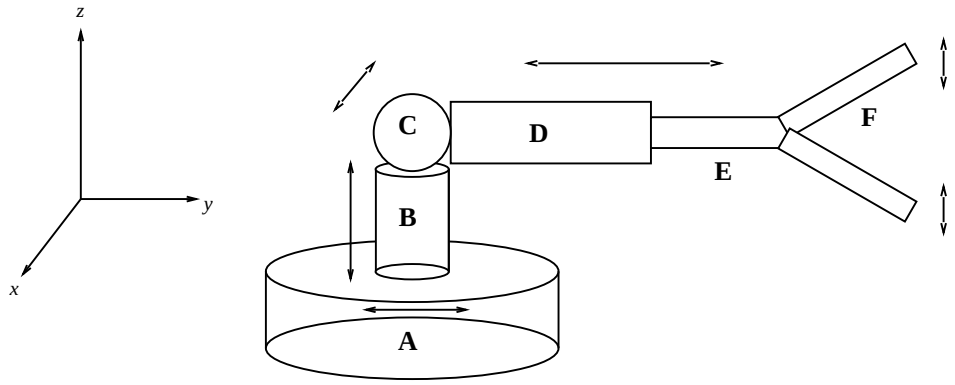
We wish to model the robot arm depicted below. You are to design an appropriate display structure. Show the nodes, their hierarchy, and what purpose each node serves (e.g. *rotation about z-axis* or *base object*).

The *base* (**A**) can rotate about the z axis. The *post* (**B**) can extend and retract along the z axis. It moves in and out of the base like a piston in a cylinder. The *joint* (**C**) can rotate about an axis parallel to the $x-y$ plane (orientation depending on how the base has rotated). The angle that the *upper arm* (**D**) makes with the post is determined by the rotation of this joint. The *lower arm* (**E**) slides in and out along the upper arm, like a piston in a cylinder. The *grippers* (**F**) pivot at their ends of attachment to the upper arm.

All components (**A** through **E** and both **F**'s) are formed by scaling a cylinder with unit diameter and unit height whose base is centered at the origin:



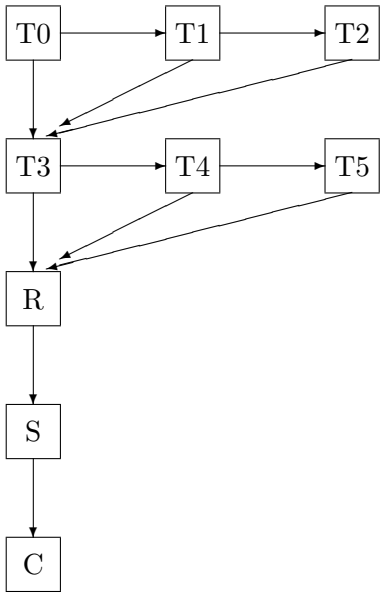
A has height $h_{\mathbf{A}}$ and diameter $d_{\mathbf{A}}$, and similarly for all other components. In building your data structure, do **not** worry about restrictions on rotations or translations, since that would add too many complications for the time you have.



8.4 Cube Array [Last Used: Spring 1992 Final]

Below is a data structure that, if properly walked and interpreted, will result in the display of 3 rows each consisting of 3 copies of the cube represented by the node "`C`". The nodes with a "`T`" label

are translations. "`T0`" is the translation that takes (x, y, z) to $(x, y-4, z)$. "`T1`" is the translation that takes (x, y, z) to (x, y, z). "`T2`" is the translation that takes (x, y, z) to $(x, y+4, z)$. "`T3`" is the translation that takes (x, y, z) to $(x-4, y, z)$. "`T4`" is the translation that takes (x, y, z) to (x, y, z). "`T5`" is the translation that takes (x, y, z) to $(x+4, y, z)$. The "`R`" node is a rotation, and the "`S`" node is a scale.



(a) Show how the successive stages of the matrix stack will look as this data structure is walked in a depth-first manner (the manner presented in class).

(b) Change the data structure so as to omit the scaling on the topmost row of cubes (the one corresponding to the $(x, y+4, z)$ translation).

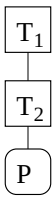
(c) How would you use the IRIS picking mode and namestack to permit any of the 9 cubes on the screen to be picked individually?

(d) How would you modify the data structure to permit any cube, once picked, to be scaled uniformly larger or smaller?

8.5 Hierarchical Ray Tracing [Last Used: Spring 2000 Final]

When ray tracing a hierarchical model, you can transform the ray instead of transforming the primitives. This allows you to specialize your 'ray-intersect-object' code as you only need to intersect the ray with simple primitives.

Assume we have a simple scene consisting of one object (a geometric primitive, P) that has been transformed by a series of two transformations, T_1 and T_2 as illustrated in the following hierarchy tree:



Describe the ray tracing process discussed in the paragraph above on a ray $(p, \vec{v})$ as it is traced in this scene. Assume that the transformed ray strikes the simple primitive at point q and that the primitive has normal $\hat{n}$ at q, where both q and $\hat{n}$ are specified relative to the modeling coordinates of the simple primitive.

Be sure to note what geometric entities get transformed and what transformation they get transformed by, what ray is used in the object intersection calculation, and what geometric surface information is used in the actual shading calculation.

8.6 Ray Tracing and Hierarchical Transformations [Last Used: Winter 2000 Midterm]

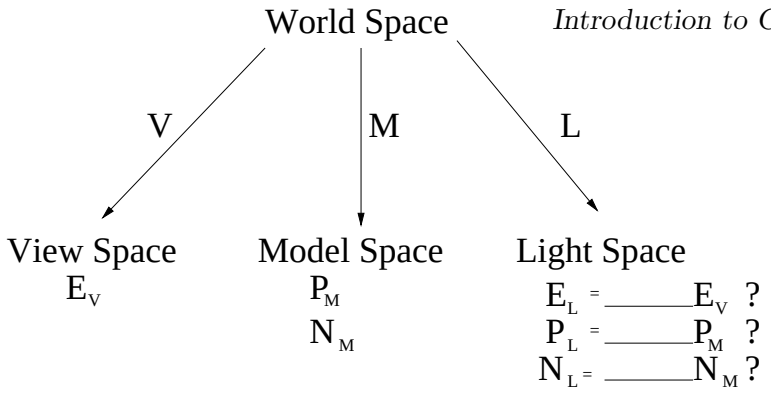
Suppose you are given the following parameters for viewing a scene relative to a left handed world frame:

View direction:	(0, -3, 4)
Up vector:	(0, 1, 0)
Eye point:	(0, 10, -10)
Field of view:	90 degrees

a) Define a left handed orthonormal view frame with respect to the world frame. The z-axis should point in the view direction, the y-axis should be in the same plane as the view direction and the up vector, and the origin should be at the eye point.

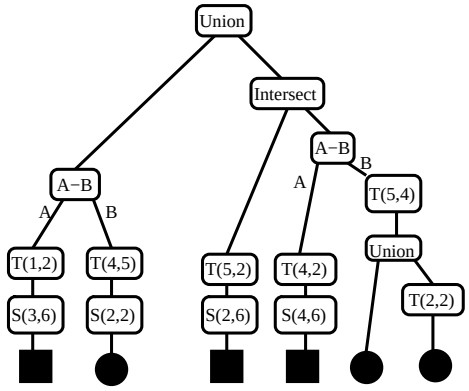
b) Give a change of basis matrix from coordinates relative to the view frame to coordinates relative to the world frame.

c) Suppose you are given the following DAG. Transformation matrices V, M, and L represent change of basis transformations from child space to parent space. We have found a ray-object intersection point and normal in model space and wish to perform a lighting calculation in light space. Indicate the required transformation to convert E_V, P_M and N_M into light space (fill in the blanks below).



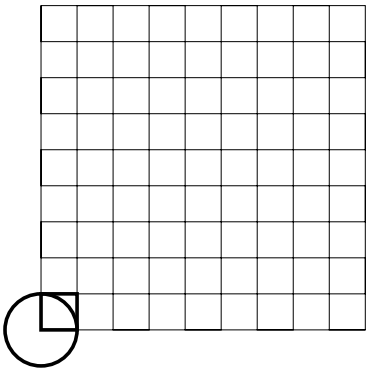
8.7 Hierarchical Transformations [Last Used: Winter 2006 Final]

Suppose you are given the following two-dimensional CSG scene hierarchy



where the circles and squares at the bottom are primitives, the T(x,y) nodes represent translations by those amounts, the S(sx,sy) nodes represent non-uniform scales by those amounts, and the A-B, Union, and Intersect nodes represent those CSG operations.

In the grid below, draw the resulting scene. In this figure, we have drawn the square and circle primitives in their base locations. We have drawn them "hollow" to indicate their locations more clearly; however, you should think of them as "filled in" as shown in CSG tree above, and you should shade in the appropriate areas in your figure below.



9 Input

9.1 Virtual Input Devices [Last Used: Winter 1992 Midterm]

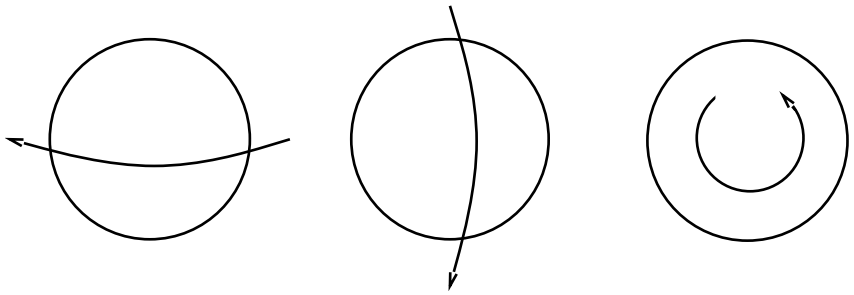
(a) The text names six virtual graphic input primitives:

locator, pick, choice, valuator, string, stroke

Write a brief (one or two sentence) description of each.

(b) With the exception of *pick*, indicate a "natural" hardware device to associate with each.

(c) Abraham Maslow once said, "If the only the tool you have is a hammer, the whole world tends to look like a nail." Suppose that the only physical input device available is a *trackball*, with no buttons, that responds to rotation around three axes.



Describe what would you would do to implement each of the five virtual devices of (b) using only the trackball. The *locator* you implement should be a *3D locator*. (You may assume that there is screen feedback provided; i.e., rolling the ball left/right moves the cursor horizontally, rolling the ball up/down moves the cursor vertically, and twisting the ball makes the cursor blink.)

10 Modeling

10.1 Tetrahedron [Last Used: Spring 2000 Final]

Consider the tetrahedron whose vertices are $A = (1, 1, 1)$, $B = (0, 1, 1)$, $C = (0, 0, 1)$, and $D = (0, 0, 0)$.

(a) What are the normals of the four triangles that bound the tetrahedron?

(b) Assume that the eyepoint is at the point (-1,-1,-1). Which triangles (ABC, ABD, ACD, or BCD) are backfacing?

(c) Assume that a light source is at the point ($-\infty$,0,0), has intensity 10.0, and all faces of the tetrahedron have a diffuse reflectivity coefficient of 1.0. Compute the light due to diffuse reflection from all four faces of the tetrahedron.

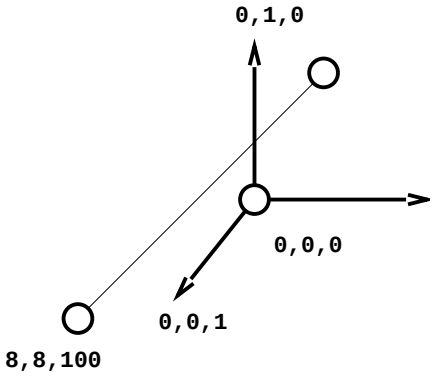
(d) Describe how to calculate vertex normals given the normals for each surface of an object. Calculate the vertex normals for the tetrahedron.

10.2 Cube Projected as Hexagon [Last Used: Winter 1992 MidTerm]

A cube viewed in parallel projection along a body diagonal (corner to corner through the centre of the cube) has a hexagonal cross-section. The objective of this question is to show the sequence of transformations needed to place a cube in space so that a hexagon appears on the display.

Suppose that you have viewing parameters given by:

```
projection reference point: 8, 8, 100
view up direction:          0, 1, 0
view plane normal:          0, 0, 1
view reference point:       0, 0, 0
```



Suppose that the window extends from `-1,-1` to `17,17` on the view plane, and the projection is parallel.

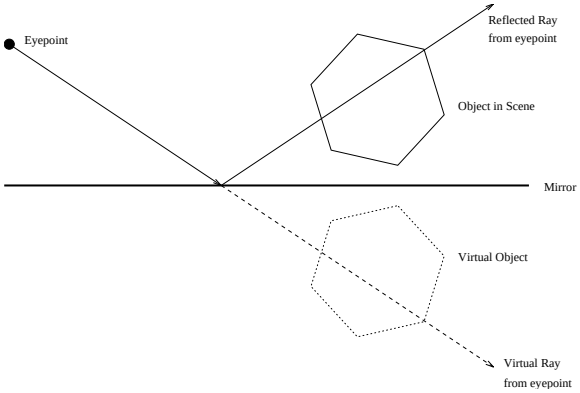
Start with a cube centred at the origin, with unit-length edges and faces parallel to the coordinate planes, and subject it to a sequence of transformations until it appears in the centre of the window with the body diagonal (originally `-0.5,-0.5,0.5` to `0.5,0.5,-0.5`) perpendicular to the view plane. Scale the cube so that the hexagon's side is one unit.

List the sequence of transformation you would use, and sketch the cube and its projection (with dimensions shown) after each transformation in the sequence. You may express the transformations in simple terms; e.g. "rotate clockwise around the x-axis by 60 degrees."

11 Mirrors

11.1 Geometric Mirror Reflections [Last Used: Fall 1994 Midterm]

We wish to create an image of a scene that has a planar mirror in it. Refer to the figure below. Assume that every scene has only one mirror, the mirror is opaque, only one side of the mirror is reflective, and objects in scene can be on either side of the mirror.



(a) If the mirror is an infinite plane, describe how to create an image of any polygonal scene containing the mirror using the transformation $\text{Refl}(n_x, n_y, n_z, d)$ and a polygon-plane clip operation. The vector-matrix product $V\text{Refl}(n_x, n_y, n_z, d)$ reflects point V about the plane given by the equation $n_x p_x + n_y p_y + n_z p_z + d = 0$, where $P = [p_x, p_y, p_z]$ is any point on the plane and the vector $\vec{n} = [n_x, n_y, n_z]$ is the normal to the plane.

(b) The matrix

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

reflects all points about the yz plane, i.e. it negates the x coordinate of any point to which it is applied. Using primitive rotation, translation, and reflection transformations, describe how a composite transformation can be implemented to perform the operation described by $\text{Refl}(n_x, n_y, n_z, d)$.

(c) Assume further that we want to create images of 3D scenes that contain a single planar, convex polygonal mirror (i.e. a mirror bounded by a convex polygon). Describe how this could be implemented using general clip planes for arbitrary viewpoints. You may use a diagram.

12 Other

12.1 Homogeneous Coordinates [Last Used: Fall 1992 Final]

Convention

Projective points in 2D:	$[x, y, z]$
Projective points in 3D:	$[x, y, z, w]$
Projective lines in 2D:	$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$
Projective planes in 3D:	$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$

(A) Describe each item given below as precisely as you can:

(i) $[0, 0, 1]$

(ii) $[0, 0, 0, 1]$

(iii)

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

(iv)

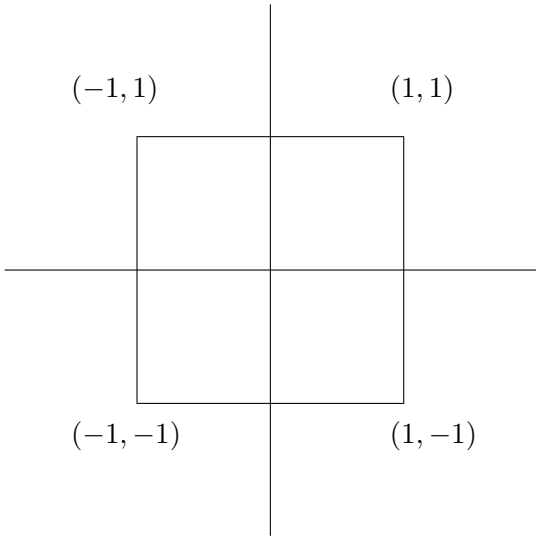
$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

(v)

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

(B) Find the homogenous coordinates of the point P in the plane, where the line through $A = [2, 3, 4]$ and $B = [1, -1, 0]$ meets the line $L = \begin{bmatrix} -3 \\ 2 \\ -1 \end{bmatrix}$

(C) Here is a test pattern in the plane:



Draw its image after being transformed by each of the following (individually, not cumulatively):

(i)

$$\begin{bmatrix} \cos 30 & \sin 30 & 0 \\ -\sin 30 & \cos 30 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

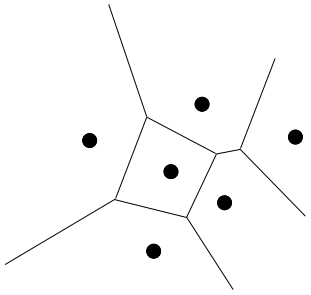
(ii)

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

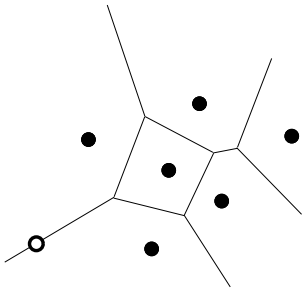
$$\begin{bmatrix} 1 & 0 & 0.2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

12.2 Voronoi Diagrams [Last Used: Winter 1993 Final]

In the figure below, the black circles are the data points and the lines show the Voronoi diagram for these points. Sketch in the corresponding Delauney Triangulation.



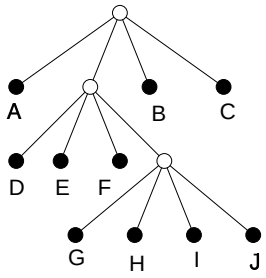
In the figure below, suppose we have add an additional point at the hollow circle. Sketch how the Voronoi Diagram will change. Put hashmarks through any segment or portion of a segment that is no longer in the diagram.



12.3 Quadtrees [Last Used: Winter 1993 Final]

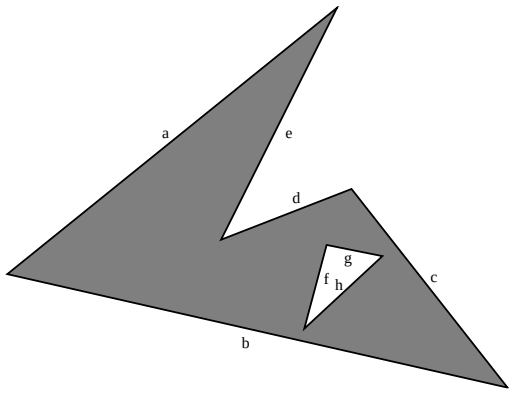
The tree below represents a quadtree subdivision of a square. The left most branch is the upper left quarter, the next branch is the upper right, the third branch is the lower left, and the right most branch is the bottom left of a region. The leaf nodes are labeled A-J.

Subdivide the square below to indicate the subdivision of the quad tree. Label each portion of the square with the appropriate label from the tree.



12.4 BSP Trees [Last Used: Spring 2000 Final]

Below is a concave polygon with a hole. Draw a BSP tree that represents this polygon. The letter appearing near each edge is the label you should use for that edge in your tree. State what 'in'/'out' convention you are using, and label the leaves of your tree appropriately.



12.5 Event Queues, Callbacks [Last Used: Winter 1996 Midterm]

Describe the differences between *Event Queues* and *Callbacks*. Pay particular attention to the structure of the code you write to use each technique, and to the details of how control and events are passed to your program.

12.6 Volume Rendering [Last Used: Winter 2000 Final]

The ray-casting integral for volume rendering may be written

$$I = \int_a^b g(s) e^{-\int_a^s \tau(x)dx} ds$$

where $I(a, b)$ is the final itensity of the pixel, and a and b represent distances along the ray to the nearest and farthest voxels of the volume, respectively. Assuming we collect samples of the volume along the ray from $i = 1$ near the viewer to $i = n$, giving a voxel intensity I_i and opacity α_i at each sample:

a) Write an equation approximating the integral I in terms of the samples I_i and α_i, for $i = 1, \ldots, n$.

b) Assuming sample $i = 1$ is nearest the viewer, why is summing the equation from "back to front" (from $i = n$ to $i = 1$) fast to evaluate?

c) Why could summing the equation from "front to back" (from $i = 1$ to $i = n$) allow an even faster approximation of I than in part (b)?

d) Given a light source L, if we want to compute an illumination equation for each I_i, what vector should we use as a normal in the equations? Precomputing this vector for all voxels can save rendering time; however, what is the main disadvantage of precomputing this vector?

e) If we don't precompute the normal vector described in (e) above, in a simple, unoptimized implementation of volume rendering, how many evaluations of the illumination equation typically contribute to each sample I_i on the ray

- for Gouraud-style lighting?
- for Phong-style lighting?

f) If I_i is computed by interpolating the voxel intensities in the volume, *nearest neighbour* and *trilinear* interpolation are two possible methods. Give an advantage for each method.

g) For rendering purpoess, storing a volume as a large 3D array makes inefficient use of cache memory. Suggest a better storage method and briefly describe why it is more efficient.

12.7 Bounding Boxes [Last Used: Fall 2000 Midterm]

Bounding boxes are one method of increasing the speed of a ray tracer. Typically, the bounding boxes are aligned with the coordinate axes.

1. Discuss the benefits/costs of axis aligned bounding boxes vs bounding boxes that are not axis aligned.

2. Discuss the benefits/costs of using bounding spheres instead of bounding boxes.

13 Perspective

13.1 Perspective Matrix [Last Used: ?]

(a) Give the equations for the clipped 3-D coordinates $(x1', y1', z1')$ and $(x2', y2', z2')$ in terms of the unclipped 3-D homogeneous coordinates $(x1, y1, z1, 1)$ and $(x2, y2, z2, 1)$ plus the two clipping parameters $\alpha 1$ and $\alpha 2$ that are computed by the clipper. You may assume that $\alpha 1 < \alpha 2$.

(b) Prove that the line segment produced by the algorithm actually lies on the original line. [**Hint:** Show that if $\alpha 1 = 0$, the point $(x1', y1', z1')$ is the transformed $(x1, y1, z1, 1)$, and if $\alpha 2 = 1$), the point $(x2', y2', z2')$ is the transformed $(x2, y2, z2, 1)$. Then prove that as $\alpha 1$ is varied from 0 to 1, dy/dx and dz/dx are both constant, and hence we get a straight line.]

13.2 Near Plane not Viewing Plane [Last Used: Spring 2000 Midterm]

The perspective matrix **P** we have discussed in class assumes that the near clipping plane coincides with the viewing (projection) plane. This matrix maps any point in the viewing volume into a point with a fourth coordinate $\neq 1$:

$$\mathbf{P} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}$$

Division by w' completes the projection process. Construct a perspective matrix for the more general case in which the viewing screen and the near plane are parallel, but do not necessarily coincide. Specifically, the eye is to be located at the origin, looking in the $-z$ direction, with the y axis pointing up, and the viewing screen is to be located at $z = -v$, between the near clipping plane at $z = -n$ and far clipping plane at $z = -f$.

13.3 Z Mapping [Last Used: Spring 1994 Midterm]

Consider an SGI style projective transformation that maps z values from $[n, f]$ to $[-1, 1]$ where both n and f are positive, and $n < f$ (for this problem, we are not interested in the mapping of x and y). Answer the following questions in as concise a manner as possible.

1. Where does the perspective projection map the point $(0, 0, n/2)$? Does this cause us any problems?
2. Where does the perspective projection map the point $(0, 0, -2n)$? What problems does this cause and how may this be avoided?
3. What happens as n approaches 0?
4. Why is it advantageous to define a far clipping plane? Where should we place the far clipping plane?

In class, we investigated where the perspective projection mapped various z intervals. For parts (a) and (b), you only need to give the appropriate interval, and do not need to note the precise location after perspective projection.

13.4 Transformation Derivation [Last Used: ?]

A perspective transformation can be performed on a point $P = [x, y, z, 1]$ in three space by multiplying P by matrix

$$\begin{vmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & e & f \end{vmatrix}$$

and then dividing all coordinates by the fourth coordinate.

In class, we found values for mapping x and y onto the window, and z into the range $[0, 1]$. SGI GL, on the other hand, maps x, y, z into the range $[-1, 1]$ (x is actually mapped into slightly different values that don't affect the discussion here).

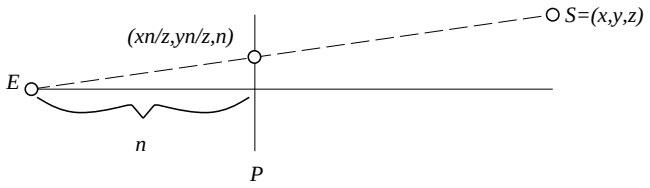
Find values of a, b, c, d, e, f that map x and y into the window, but map z to $[-1, 1]$.

13.5 Line Segments and Z [Last Used: Winter 1998 Final]

If we are perspectively projecting onto a plane P (where P is perpendicular to the viewing direction and a distance n from the eye point) the mapping

$$(x, y, z) \rightarrow (xn/z, yn/z, n)$$

gives the projection of point (x, y, z) onto P.



If we then map the end points of a line segment ℓ onto the plane in this manner, then the line segment connecting these projected points is the projection of ℓ. However, if we project multiple segments, we may want to know which segments occlude other segments. Thus, we need to retain some form of the z information. Because this information is lost by the above mapping, we normally use a mapping similar to

$$(x, y, z) \quad \rightarrow \quad (xn/z, yn/z, (z - n)/z).$$

If we use this mapping, the realtive depths of the endpoints of a line segment are preserved.

1. Why do we choose the mapping given above rather than the simpler mapping

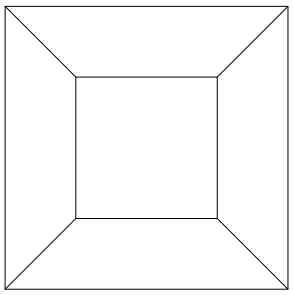
   $$(x, y, z) \quad \rightarrow \quad (xn/z, yn/z, z)$$

   which also preserves the relative depths of the endpoints of a line segment? You do not need to prove your answer.

2. We are only interested in the portions of the line segments that are within the viewing volume. Should we clip the line segments to the viewing volume before or after the perspective projection? Justify your answer.

13.6 View Transformation [Last Used: Fall 2004 Midterm]

Suppose we look at a unit cube from the point $(.5, .5, 2)$ in the direction $(0, 0, -1)$. (The unit cube has corners at $(0, 0, 0)$, $(0, 0, 1), \ldots, (1, 1, 1)$.) From this location, the wireframe line drawing we see will look like two squares whose corners are connected by line segments, where the inner square has sides exactly one half the length of those of the outer square. The result will look something like



Now suppose we perform a uniform scale of 2.0 about the view point (e.g., about the point $(.5, .5, 2)$). Describe how the picture above will change. You may assume there are no clipping planes and that nothing else changes (e.g., we use the same perspective transformation). You may label the drawing above if it will help your description, and you may also sketch a picture below if you find that helpful (but you must still give a verbal description of any changes to the picture).

13.7 Perspective Projection [Last Used: Fall 2002 Midterm]

A point $p = [x\ y\ z\ 1]^t$ is transformed by multiplying by the matrix product PVM, where M represents the modeling transformation, V represents the world-to-view change of basis, and P is the following OpenGL perspective transformation matrix

$$P = \begin{bmatrix} \frac{\cot(\theta/2)}{\text{aspect}} & 0 & 0 & 0 \\ 0 & \cot(\theta/2) & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix},$$

where 'aspect' is the ratio of the width to height of our viewport.

The resulting matrix product yields

$$PVMp = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}$$

1. What must be done after computing $PVMp$ to complete the perspective transformation? Be specific (i.e., show the final result in terms of A, B, C, and D).

2. After completing part (a), what coordinate system is your result in? Describe the coordinate system in words, stating anything you know about the range of the coordinates.

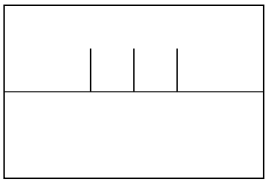
3. Suppose our viewport is 200 pixels high. How must we further adjust the coordinates to map to this viewport? For simplicity, assume the lower left corner of the viewport has device coordinates (0,0) and that x increases to the right and y increases upward.

4. Suppose we transform our model by a transformation represented by a matrix T_m relative to modeling coordinates. Show how to modify PVM to get a new transformation representing the composition of the new transformation with the old transformation.

5. Suppose we transform our model by a transformation represented by a matrix T_w relative to world coordinates. Show how to modify PVM to get the new transformation representing the composition of the new transformation with the old transformation.

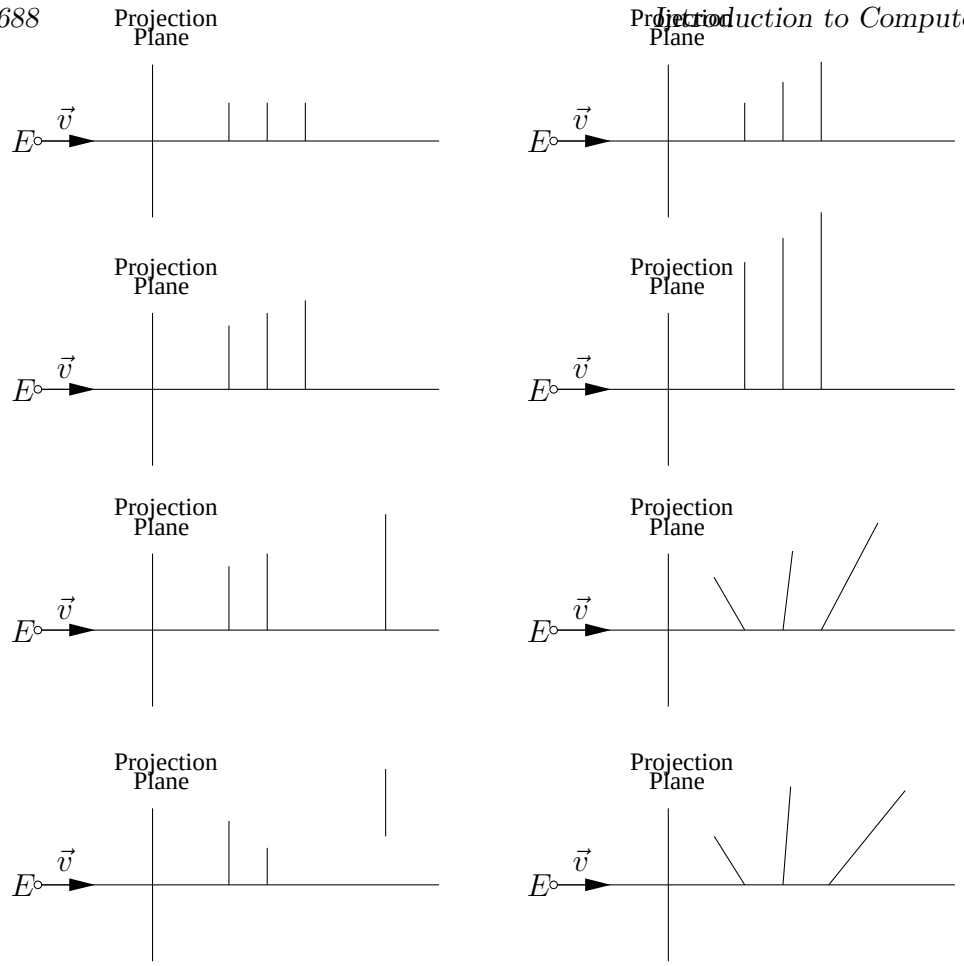
13.8 Perspective transformation of line segments [Last Used: Fall 2009 Final]

Suppose you have three line segments that project onto the screen as in the following diagram:



Now suppose we look at an orthographic projection of these line segments from a side view in the direction indicated by the arrow in the above diagram (with the arrow being parallel to the projection plane). Circle the figures below that could possibly be a view of this orthographic projection of the line segments appearing in the above perspective projection, and cross out those that can not be a view of this orthographic projection of the line segments.

Notes: in each of the figures below, the viewing parameters used in the figure above are indicated by E, the eye point, and $\vec{v}$, the view direction. Also note that the height of the line segment representing the projection plane is equal to the height of the window in the picture above.



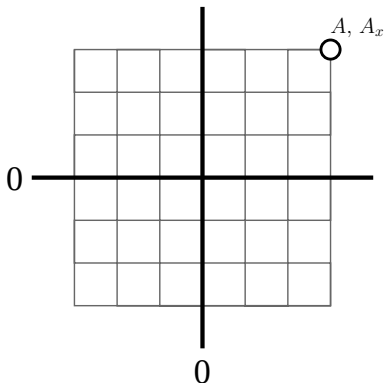
13.9 Projecting a box [Last Used: Winter 2006 Midterm]

The eye point is at the origin, you are looking along the $+z$-axis. You are looking at an axis aligned box whose corners are at

$$(4,4,4),\ (4,4,8),\ (4,-4,4),\ (4,-4,8),\ (-4,4,4),\ (-4,4,8),\ (-4,-4,4),\ (-4,-4,8).$$

On the graph paper below, draw the box in wireframe as it would appear projected on to the $z = 1$ plane. The $x = 0$ and $y = 0$ lines have been drawn on this grid.

The grid isn't marked with units — be sure to assign x, y-values to the point A to precisely indicate the location of the projected box!



13.10 Viewing coordinates and perspective [Last Used: Winter 2007 Final]

You are rendering a wireframe drawing of a cube onto a screen. A camera looks directly at the centre of one of the cube's faces, in a direction perpendicular to that face. The camera is outside the cube. The rendering looks like a square centred inside a larger square, with diagonal lines joining correpsonding corners.

Your goal is to manipulate the camera's parameters in order to achieve desired changes to the cube's projection.

1. What must be done to the camera so that the cube appears to move to the right on the screen, with the two squares still being drawn as squares?

2. What must be done to the camera so that the front and back faces of the cube get larger on the screen, but not with the same scaling factor?

3. What must be done to the camera so that the entire drawing appears to be scaled up uniformly on the screen?

4. What must be done to the camera so that the front face of the cube stays the same size, while the back face gets larger?

13.11 Perspective [Last Used: Winter 2007 Final]

You are standing on the middle of the railroad tracks when you see that a train is approaching in the distance. You hold your thumb out in front of one eye at arm's length, and notice with interest that your thumb exactly covers the width of the train. You know that your arm is one metre long and your thumb is two centimetres wide. Looking down, you judge that the railroad tracks are 1.5 metres apart, and you see that the train's width extends exactly to the width of the tracks.

1. How far away is the train?

2. If the train is travelling at 50km/h, how many seconds do you have to get out of the way?

13.12 Forced perspective [Last Used: Winter 2011 Final

Photo by Shane Huang

A classic "forced perspective" trick in photography is to have two friends Nero and Farley pose for the camera, with Farley standing much farther away than Nero. Nero then holds her hand up in such a way that in the resulting photograph she appears to be holding a tiny statue of Farley in her upturned palm. The photograph on the right gives an example.

Assume that the ground is perfectly level, and that the photographer, Nero and Farley stand in a straight line. The photographer holds the camera at height h_C above the ground. Nero's body (including her hand) is at distance d_N from the photographer (measure d along the ground). Her hand is at height h_N above the ground.

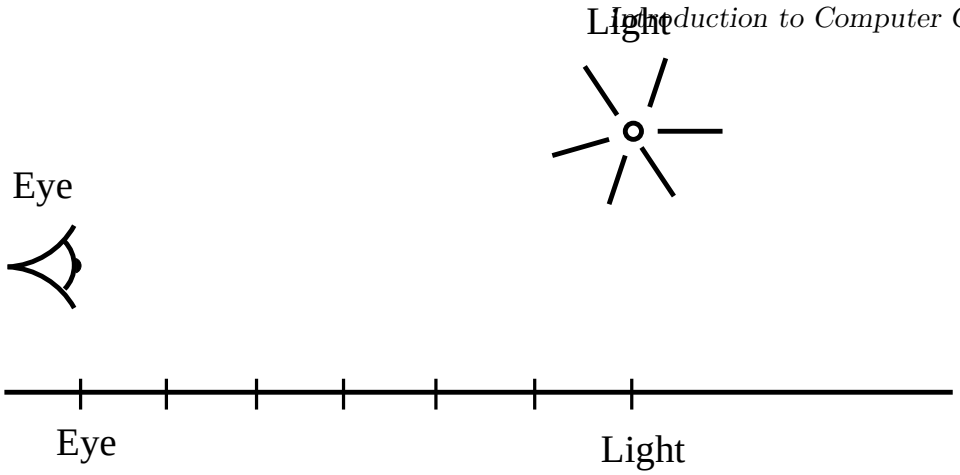
(a) Assuming that you are given h_C, d_N and h_N, find an expression for d_F, the distance required between Farley and the camera in order to ensure that his feet rest precisely in Nero's hand. (You may wish to draw a diagram of this scene as viewed from the side.)

(b) Assume that Nero and Farley are of identical heights. At what height h_N must Nero hold her hand so that in the photograph, Farley appears to be exactly 1/10 as tall as Nero?

14 Shading and Lighting

14.1 Diffuse and Specular Last Used: Fall 2010 Final

In the 2D figure below, we have the eye located on the left, a plane (line) along the bottom, and a point light source towards the right; the light is twice as far from the plane/line as the eye is from the plane/line. The marked and labeled locations on the plane/line indicate the closest point of the eye and the light to the plane/line. For the purposes of this question, the dot at the front of the eye is used as the eye point, and (for the first two parts) the intensity of the light does NOT attenuate with distance.



1. Assume the plane/line only has a diffuse material component. Mark with a 'D' on the figure where the plane/line will appear brightest to the viewer.

2. Assume instead that the plane/line only has a specular material component. Mark with an 'S' on the figure where the plane/line will appear brightest to the viewer.

3. Suppose now the light has r^2 attenuation. How will the location of the brightest point change if the plane/line only has a diffuse material component?

4. Again suppose the light has r^2 attenuation. How will the location of the brightest point change if the plane/line only has a specular material component?

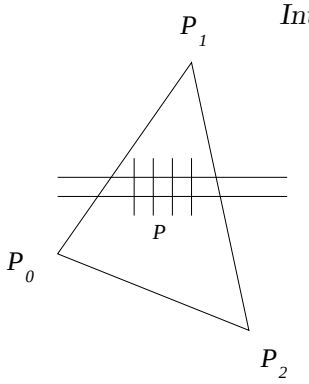
14.2 Phong Lighting and Gouraud Shading [Last Used: Spring 1992 Final]

Answer the following parts, giving details and/or reasons for all answers.

(a) The shading intensity at any given point on a surface is, in general, comprised of three contributions, each of which corresponds to a distinct physical phenomenon. List all three, stating how they are computed in terms of the appropriate vectors (which you should define). In particular, what is the formula for the "reflection vector"?

(b) How is any interior pixel of a triangle, indicated below by P, to be coloured according to the Gouraud method? In which spaces (object, world, perspective, or screen) could one compute the normals? In which spaces could one compute the shades? In which spaces should the normals not be computed? In which spaces should the shades not be computed? What information must the pixel-filling algorithm have in order to assign pixel shades to the edges and interior of the screen projection of a polygon?

(c) Answer all of the questions in part (b) for the Phong method of assigning pixel colours.



(d) Suppose **n** is the normal to an object and that object is subjected to an affine transformation represented by the 4×4 matrix **M**. How should the normal be transformed? Why?

(e) A sphere of unit radius is centred at the origin. In sequence the sphere is: (1) scaled in x by 2 and y by 4; (2) illuminated by a light at infinity in direction $(1, 1, 1, 0)$; (3) viewed by an eye at $(5, 5, 0, 1)$ looking at the point $(\sqrt{2}, 2, 1/2, 1)$, which in on the surface. Give as much detail as you can about how to compute the shade of this point as it appears to the eye.

14.3 Barycentric Coordinates and Gouraud Shading [Last Used: Fall 95 Final]

The *barycentric coordinates* of a point, P, in a triangle are the unique numbers u, v, w such that

$$u \geq 0,\ v \geq 0,\ w \geq 0,\ u + v + w = 1$$

and

$$P = uP_0 + vP_1 + wP_2$$

where P_0, P_0, P_0 are the vertices of the triangle. They generalize the notion of the parametric description of a point, P, in a line segment

$$P = \alpha P_0 + \beta P_1$$

where P_0, P_1 are the endpoints of the segment and α, β are nonnegative numbers summing to 1 (usually we just let $\beta = 1 - \alpha$).

The Gouraud method of shading, for a triangle, takes the colors C_0, C_1, C_2 at the vertices P_0, P_1, P_2 and produces the color of each pixel in the triangle by successive linear interpolations. Specifically, the color of each edge pixel is taken as the linear interpolation of the colors of the vertices defining the edge; for example,

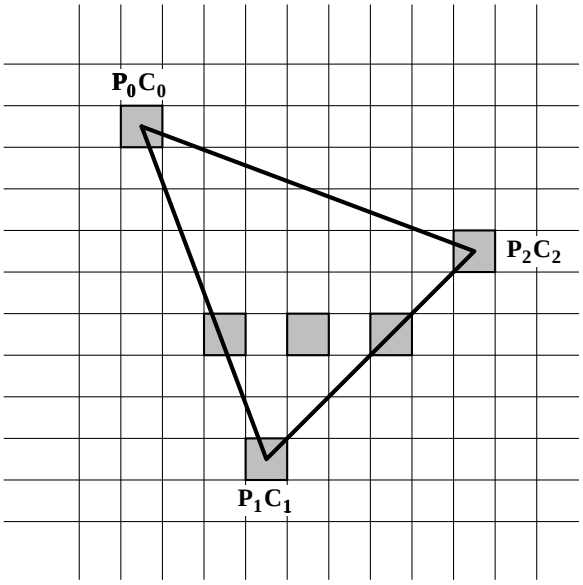
$$C_\alpha = (1 - \alpha)C_0 + \alpha C_1 \text{ for the pixel at } P_\alpha = (1 - \alpha)P_0 + \alpha P_1$$

and

$$C_\beta = (1 - \beta)C_1 + \beta C_2 \text{ for the pixel at } P_\beta = (1 - \beta)P_1 + \beta P_2$$

Finally, the color of an interior pixel is taken as the linear interpolation, along a scan line, of the colors on the edges intercepting the scan line; for example

$$C_\gamma = (1 - \gamma)C_\alpha + \gamma C_\beta \text{ for the pixel at } P_\gamma = (1 - \gamma)P_\alpha + \gamma P_\beta$$



Part (a) : Let u, v, w

$$P_\gamma = uP_0 + vP_1 + wP_2$$

be the barycentric coordinates of P_γ. Express u, v, w in terms of α, β, γ.

Part (b) : Show that the color of the pixel at P_γ can be given as

$$C_\gamma = uC_0 + vC_1 + wC_2$$

Part (c) : Consider the results of Parts (a) and (b) "in the limit." That is, suppose the triangle is large enough that the size of a pixel can be neglected, $u, v, w, \alpha, \beta, \gamma$ can be regarded as continuously varying, and all rotations are possible. Claim: the color at any position on the triangle will be unchanged if the triangle is rotated. Is this claim true or false? Why?

Part (d) : Whether or not you believe the claim in Part (c), it is **not true** that the color at any position of a square will be unchanged if the square is rotated, assuming that the color is interpolated in the same way as on a triangle (linear interpolation along edges followed by linear interpolation into the interior). Give an example to show that the color of the center point of a square could depend on the square's orientation.

14.4 Phong Lighting [Last Used: Fall 1992 Final]

A spot on a shiny plastic surface could to be coloured according to:

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} \left[k_d O_{d\lambda} (N \cdot L) + k_s (R \cdot V)^n \right]$$

(a) Explain what each of the symbols means and what role it plays. (e.g. What does k_a stand for? What does the a stand for? What effect does k_a represent? etc.)

(b) What are reasonable values for the k's, I's, and O's? What are reasonable restrictions on the "sizes" of N, L, R, and V?

(c) What are reasonable values of n? What does increasing or decreasing n do?

(d) What colour will highlights have using this formula?

(e) $(N \cdot L)$ is related to what angle? $(R \cdot V)$ is related to what angle? What formula relates R to N and L?

(f) What formula does the book use for f_{att}? Explain why this formula is a reasonable hack.

(g) The formula we used for the Ray Tracing Assignment differed from (16.14) in at least two modifications. What were the modifications? In what ways did these changes change the behaviour of the lighting model? In particular, what could we do with highlights that formula (16.14) could not?

14.5 Lighting Approximations [Last Used: Winter 1993 Final]

Z-buffer, Ray Tracing, and Radiosity model various lighting effects. In the table below, put a '+' if the method is well suited to modeling the effect, a '0' if the effect is modeled by a coarse approximation, and a '-' if the method can not model the effect (by "Z-buffer", we mean Phong shading of a set of triangles using the standard graphics pipeline).

	Z-buffer	Ray Tracing	Radiosity
Direct Illumination			
Indirect Illumination			
Shadows			
Reflections			
Refraction			
Diffraction			

14.6 Ray Tracing Techniques [Last Used: Fall 2000 Final]

There are a large number of techniques for improving the realism of ray traced images. Briefly describe how each of the techniques below modifies the lighting calculation:

1. Texture Mapping
2. Bump Mapping
3. Solid Textures

Which of these techniques would be most appropriate for modeling each of the following?

(i) A picture hung on a wall.

(ii) A marble statue.

(iii) A golf ball (with no logo).

(iv) A patterned tile floor.

(v) The ripples created by dropping a stone in a pool of water.

(vi) A wooden table top.

14.7 Phong vs. Gouraud Shading [Last Used: Fall 2004 Midterm]

A new machine arrives in the graphics lab. You know that it renders shaded polygons, but the manuals don't specify whether they are flat shaded, Gouraud shaded, or Phong shaded. Give a test (or series of tests) to determine which shading method is used. You may specify the vertex positions and normals of polygons, the surface property of each polygon, the location of the light source(s), and the viewing position and direction.

Be specific about the scene itself, and what you would look for in the resulting image to determine which shading method is used.

14.8 Lighting Model [Last Used: Winter 1996 Final]

In our illumination model, there were three terms: an ambient term, a diffuse term, and a specular term.

- What effect does the ambient term model?
- What assumption(s) do we make for the diffuse term?
- In the specular term, there is a factor of $(\vec{r} \cdot \vec{v})^p$. What effect does varying the p power have?

14.9 Gouraud Shading Invariance [Last Used: Spring 1994 Final]

Suppose we have a triangle and a rectangle whose vertex colours may be arbitrary. Further suppose we have 2D viewing software that shades both of these primitives by linearly interpolating vertex colours along the edges of the primitives, and then for each scanline, linearly interpolating the edge colours for that scanline to shade the interior pixels.

Now suppose that our viewing software allows us the rotate the primitives. Since this is a 2D viewer, the shapes of the primitives won't change significantly as we rotate.

1. Can the shading of the triangle change signficantly as we rotate? If the shading does not change significantly, justify your answer. If the shading does change significantly, give an example, noting the vertex colours and two orientations of the primitive that result in different shading, and give a rough idea of how the shading will appear in each orientation.
2. Answer the same questions for the shading of rectangles.

14.10 Shading [Last used: Fall 1997 Final]

It has been suggested that the ambient term in shading model can be replaced by a light source at the eye point. That is, instead of just adding in ambient illumination, a new light source at the eye is added with no attenuation, and is treated just like the other light sources in the scene.

How is this approach similar to using ambient illumination? How is it different? Consider both raytracing and the polygon rendering engines common in graphics hardware.

14.11 Gouraud and Phong Shading [Last Used: Spring 2000 Midterm]

You are given a rectangle ABCD in screen space. Each rectangle vertex has an RGB colour, to be used for Gouraud shading:

Vertex	*Screen Coordinates*	*RGB Colour*
A	(0,0)	(250,0,0)
B	(10,0)	(250,250,0)
C	(10,5)	(0,250,0)
D	(0,5)	(0,0,0)

a) Express point E=(6,0) as an affine combination of points A and B.

b) Express point F=(6,5) as an affine combination of points C and D.

c) Express point G=(6,4) as an affine combination of points E and F.

d) If Gouraud shading is used at point G, what should its RGB colour be?

e) If Phong shading is used at point G, we need to determine a normal vector at G for the illumination equation. Assuming that the rectangle samples an underlying curved surface having normals n_A at A, n_B at B, n_C at C, and n_D at D, write an expression for the normal at G.

14.12 Lighting [Last Used: Spring 2000 Final]

Given a point for shading on a surface, we usually consider a normal vector $\vec{n}$, a vector to the light source $\vec{l}$, a vector to the viewer $\vec{v}$, and a reflection direction $\vec{r}$ (all vectors are of unit length). One possible version of the illumination equation is

$$I_{out}(\vec{v}) = k_a I_a + \sum_i \left(k_d + k_s \frac{P(\vec{r}, \vec{v})}{\vec{n} \cdot \vec{l}} \right) I_0^i \frac{\vec{l} \cdot \vec{n}}{c_1 + c_2 d_i + c_3 d_i^2}$$

a) Give an equation for the reflection direction vector $\vec{r}$ in terms of vectors $\vec{l}$ and $\vec{n}$.

b) Define the Phong shading term $P(\vec{r}, \vec{v})$ in terms of $\vec{r}$, $\vec{v}$, and a parameter p. For wider highlights on the surfaces is p increased or decreased?

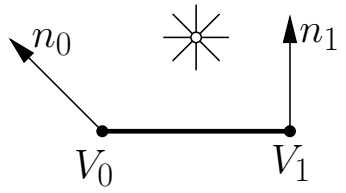
c) What is the term $k_a I_a$? How does increasing I_a affect the rendering of objects in the scene?

d) What is the term $c_1 + c_2 d_i + c_3 d_i^2$? How is d_i calculated? Suggest values for c_1, c_2, c_3 that would give harsh point lighting, as in outer space?

e) This equation doesn't mention R, G, or B, and yet we use it to render coloured objects. How do we adapt the equation to handle RGB?

14.13 Flat, Gouraud, and Phong Gouraud Shading [Last Used: Winter 2006 Midterm]

Suppose we are working in 2D and shading lines. Consider the following line segment (the thick line) with normals at its vertices, and a point light source:



Describe how the line segment will look when shaded with the following three shading models, noting in particular where the brightest point on the line segment V_0V_1 will occur. Assume that the line segment has only diffuse reflection components, that there is no ambient lighting, and that all lighting is gray scale (i.e., no colour).

1. Flat shading

   Description:

   Brightest point:

2. Phong Shading

   Description:

   Brightest point:

3. Gouraud Shading

   Description:

   Brightest point:

14.14 Illumination [Last Used: Winter 2007 Midterm]

A unit sphere sits at the origin. In the following, assume Phong lighting:

$$L_{\text{out}}(\vec{v}) \quad = \quad k_a I_a + \left(k_d \max(0, \vec{n} \cdot \vec{l}) + k_s \max(0, \vec{r}_l \cdot \vec{v})^p \right) I,$$

where I is the illumination power after attenuation used for diffuse and specular reflection and I_a is a separate "ambient" illumination.

1. Assume $k_a = k_d = 1$ and $k_s = 0$. For a point light source located at (x, y, z) outside the sphere, what point on the sphere will be illuminated most brightly?

2. Assume $k_a = 0$, $I_a = 0$, $k_d > 0$, and $k_s > 0$. A directional light points in the direction $(0, 0, -1)$. Sketch the scene by showing the sphere and the light rays that illuminate it. Indicate clearly the curve the divides the illuminated and unilluminated parts of the sphere. You may sketch the scene in 2D, as long as it clearly demonstrates the interaction between the light and the object.

3. Using the same parameters as in the last question, what range of z values will be illuminated by the directional light?

4. Now assume that the lighting and material properties are the same as in (b) and (c), but the directional light is replaced by a point light at $(0, 0, 4)$. Will the same range of z values be illuminated by the light? Explain why or why not, possibly using another drawing.

Bonus A point light sits at (x, y, z). There is no attenuation on the light. The global ambient light is set to intensity $0 \leq I_a < 1$, and the sphere has full ambient reflectance ($k_a = 1$). Brightnesses are clamped at a maximum of 1. What is the area of the sphere's surface that is maximally bright?

14.15 Illumination [Last Used: Winter 2011 Final]

A unit sphere sits at the origin. Assume that we are using the following Phong illumination model:

$$L_{\text{out}}(\vec{v}) = k_a I_a + \left(k_d \max(0, \vec{n} \cdot \vec{l}) + k_s \max(0, \vec{r} \cdot \vec{v})^p \right) I$$

Here, I is the illumination power of a single light source after attenuation, and I_a is a separate ambient illumination level.

(a) Assume $k_a = k_d = 1$, $k_s = 0$, and the light source is a point light source located at (x, y, z) with $x^2 + y^2 + z^2 > 1$. What point on the sphere will be illuminated most brightly?

(b) If more than one of k_d, k_s and k_a are nonzero, then it becomes possible for some primary rays to yield pixels with intensities that are larger than the maximum intensity displayable on the output devi ce. Briefly describe two methods for dealing with output values that are outside the legal range.

(c) Assume $k_a = 0$, $k_d > 0$, and $k_s > 0$. A directional light point s in the direction $(0, 0, -1)$. What range of z values on the sphere will be illuminated by the directional light?

(d) Now assume that the lighting and material properties are as in (c), but the directional light is replaced by a point light a t $(0, 0, 4)$. Will the same range of z values be illuminated by the light? Explain why or why not, possibly using a diagram.

14.16 Shaders, toon shading [Last Used: Fall 2010 Final]

You are to write the fragment shader for an OpenGL toon shader. The vertex shader computes an "albedo" value (i.e., the cosine term computed in Lambertian/diffuse shading) with the following code:

```
varying float albedo ;

void main ( )
{
  vec3 pos_EC = vec3 ( gl_ModelViewMatrix * gl_Vertex );
  vec3 normal_EC = normalize ( gl_NormalMatrix * gl_Normal ) ;
  vec3 light_EC = normalize ( gl_LightSource[0].position.xyz - pos_EC );

  albedo = dot ( normal_EC , light_EC ) ;
  gl_Position = ftransform( ) ;
}
```

Write the code for the fragment shader that uses a bright red material if the albedo term is greater than 0.7 and a dark red material otherwise.

Fragment shader note: the resulting colour should be stored in the variable `gl_FragColor`.

15 Radiosity

15.1 Radiosity vs. Ray Tracing [Last Used: Spring 1996 Final]

1. In radiosity, what does a form factor represent?

2. Both ray tracing and radiosity are global illumination models. Explain why radiosity is a better method than ray tracing for computing global illumination for an architectural walk-through.

15.2 Radiosity Solution Techniques [Last Used: Fall 2000 Final]

In radiosity, briefly describe the full matrix solution and the progressive refinement solution, noting the differences between the two methods. Why might you prefer to use one method instead of the other?

15.3 Hemi-cube [Last Used: Winter 1995 Final]

1. Describe the hemi-cube algorithm for estimating form factors.

2. What advantages are there to using a hemi-cube algorithm? Be sure to consider ways to implement it and effects it can achieve.

15.4 Radiosity [Last Used: Fall 1996 Final]

- Briefly described the aspect of light modeled by radiosity techniques.
- Briefly describe how radiosity models this aspect of light.
- Although computing a radiosity solution is as slow (or slower) than ray tracing a single image, what advantage does radiosity have over raytracing?

15.5 Form Factors and Equations [Last Used Spring 1997 Final]

The *simple energy balance equation* is

$$B(x) = E(x) + \rho_d(x) \int_{y \in \mathcal{S}} B(y) \frac{\cos\theta_{xy} \cos\theta_{yx}}{\pi \parallel x - y \parallel^2} V(x,y) dA_y$$

where $B(x)$ is the total radiosity emanating from any surface point x in a scene $\mathcal{S}$, $E(x)$ gives the portion of the radiosity self-emitted from x, and the integral gives the portion of the radiosity received from all other surface points y and then re-emitted. $V(x,y) = 1$ if the line between surface points x and y is unobstructed by any surface, and $V(x,y) = 0$ otherwise. θ_{yx} is the angle between the surface normal at y and the $\overline{xy}$ line, and θ_{xy} is the angle between the surface normal at x and that same line. This equation assumes that all surfaces in the scene are *ideal diffuse* (*Lambertian*) *reflectors*. $\rho_d(x)$ is the reflection distribution function, which describes the proportion of incoming energy to any point that is re-emitted.

In general, the above equation can only be solved numerically, and the result will only provide an approximation to B. A common way to begin the computation is to express the approximation to the (unknown) function B in terms of *basis functions* N_i:

$$B(x) \approx \tilde{B}(x) \equiv \sum_i B_i N_i(x)$$

and to approximate the (known) function E in the same terms:

$$E(x) \approx \tilde{E}(x) \equiv \sum_i E_i N_i(x)$$

Substituting $\tilde{B}$ and $\tilde{E}$ into the energy balance equation will result in a residual $r(x)$ no matter what values for B_i are computed. The idea, however, is to find values for B_i that make $r(x)$ small in some way. Two approaches are common:

(a) Make $r(x_j) = 0$ for a selection of points $x_j \in \mathcal{S}$;

(b) Make $\int_{x \in \mathcal{S}} r(x) N_j(x) dA_x = 0$ for all basis functions N_j.

Assume all surfaces in the scene are decomposed into disjoint polygons P_j of area A_j and that N_j is 1 on P_j and 0 otherwise. Assume that x_j is a selected point on P_j and that $\rho_d(x) \equiv \rho_d(x_j) \equiv \rho_j$ on P_j.

1. Show that approach (a) leads to linear equations

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j$$

and give an expression for F_{ij}.

2. Show that (b) leads to the same linear equations with a different expression for F_{ij}.

15.6 Radiosity vs. Ray Tracing [Last Used: Fall 2002 Final]

The energy balance equation gives an expression for the amount of energy (radiance) leaving a point on a surface in the scene. Essentially, it is a formula for $L(x, \theta_x, \phi_x, \lambda)$, where x is a point on the surface, θ_x is the incoming direction, ϕ_x is the outgoing direction (both directions are 3-space vectors), and λ is the wavelength of light for which we are computing the energy leaving the surface.

This equation involves an emission term, and a triple integral over all incoming directions and wavelengths of the incoming energy from the scene, with terms to account for the relevant geometry. The equation also accounts for the reflective properties of the surface in question.

In general, not only is the equation impossible to solve analytically, we are unable to measure many of the properties that are required (such the reflective properties of the surface).

For those who are interested, one version of this equation is given below. However, the equation itself is not needed to answer this question.

1. Describe what simpflifying assumptions and approximations are made in ray tracing to allow us to solve a simplified version of this equation (you do not have to give this equation).

   As a result of these assumptions, what visual effects are we unable to model, and what other artifacts are introduced in the final results?

2. Describe what simplifying assumptions and approximations are made in radiosity to allow us to solve a simplified version of this equation (you do not have to give the equation).

   As a result of these assumptions, what visual effects are we unable to model, and what other artifacts are introduced in the final results?

Energy Balance Equation:

$$\begin{aligned} L^o(x, \theta_x^o, \phi_x^o, \lambda^o) &= L^e(x, \theta_x^o, \phi_x^o, \lambda^o) + \\ &\int_0^{\frac{\pi}{2}} \int_0^{2\pi} \int_{\lambda_{min}}^{\lambda_{max}} \rho_{bd}(x, \theta_x^i, \phi_x^i, \lambda^i, \theta_x^o, \phi_x^o, \lambda^o) \\ &\qquad \cos(\theta_x^i) L^i(x, \theta_x^i, \phi_x^i, \lambda^i) d\lambda^i \sin(\theta_x^i) d\phi_x^i d\theta_x^i \end{aligned}$$

- $L^o(x, \theta_x^o, \phi_x^o, \lambda^o)$ is the radiance
  - at wavelength λ^o
  - leaving point x
  - in direction θ_x^o, ϕ_x^o
- $L^e(x, \theta_x^o, \phi_x^o, \lambda^o)$ is the radiance emitted by the surface from the point
- $L^i(x, \theta_x^i, \phi_x^i, \lambda^i)$ is the incident radiance impinging on the point
- $\rho_{bd}(x, \theta_x^i, \phi_x^i, \lambda^i, \theta_x^o, \phi_x^o, \lambda^o)$ is the BRDF at the point
  - describes the surface's interaction with light at the point
- the integration is over the hemisphere above the point

15.7 Radiosity [Last Used: Winter 2000 Final]

Radiosity ultimately requires the solution of a large system of equations of this form:

$$\begin{bmatrix} 1-\rho_1 F_{1,1} & -\rho_1 F_{1,2} & \dots & -\rho_1 F_{1,n} \\ -\rho_2 F_{2,1} & 1-\rho_2 F_{2,2} & \dots & -\rho_1 F_{2,n} \\ \vdots & & \ddots & \vdots \\ -\rho_n F_{n,1} & 1-\rho_n F_{n,2} & \dots & 1-\rho_n F_{n,n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

a) What does n represent?

b) What does B_i represent?

c) What does E_i represent?

d) What does ρ_i represent?

e) What does the form factor $F_{i,j}$ represent, relative to patches i and j?

f) *Progressive refinement* is often used to speed rendering. Which patches are dealt with first in this method?

g) *Adaptive subdivision* is often used to break large surfaces into patches. What feature in a scene is most likely to look better after adaptive subdivision? Why?

15.8 Radiosity — Colour bleeding [Last Used: Fall 2004 Final]

Sketch/describe a scene that results in colour bleeding when rendered with a radiosity renderer. You may use a 2D scene if you like. Be sure to label all objects with necessary surface properties, and be sure to indicate where colour bleeding occurs.

15.9 Radiosity — Progressive

In progressive radiosity, when shooting the radiosity from one patch in the scene, the loop to distribute the radiosity to the other patches in the scene looks like the following:

```
Foreach j
    drad = pj*dBi*Fji*Ai/Aj
    dBj += drad
    Bj += drad
endfor
```

where `j` is the index into the list of patches in the scene.

Why do we add the delta radiosity (`drad`) to two variables (`dBj` and `Bj`) instead of just one variable?

15.10 Ray Tracing or Photon Mapping [Last Used: Winter 2006 Final]

Please answer each of the following questions with one of "radiosity", "photon mapping", "both", or "neither".

Which rendering algorithm....

1. ...can be used to render caustics?
2. ...can model colour bleeding?
3. ...requires scene geometry to be tessellated?
4. ...produces noisy images?
5. ...is an extension to ray tracing?
6. ...requires extra work to handle shadows correctly?
7. ...requires no additional storage beyond the geometry of the scene?

16 Ray Tracing

16.1 Ray Tracing – Photon Mapping Last Used: Fall 2009 Final

In ray tracing, we get shadows by casting rays from a point P at which we are doing a lighting calculation to a light L. To determine if a contribution from L should be included in the illumination at P, we check if the ray hits another object before reaching L.

With shadow maps (typically used for interactive purposes), we compute a "distance image" that contains the distance from a light to the closest object in the scene. When computing the illumination at a point P, we use the shadow map to determine whether to include a contribution from a particular light by checking if the distance in the shadow map is equal to the distance from P to the light.

Thus, shadow rays and shadow maps are being used for essentially the same purpose. Since the distance look-up and distance computation required by using shadow maps is much cheaper then intersecting the shadow ray with the scene, we possibly could speed-up our ray tracer by incorporating shadow maps into our ray tracer to do the shadow computation.

Discuss what would be involved in integrating shadow maps into a ray tracer, noting what costs, problems and benefits there would be. State any suggestions you have for addressing the problems. You should not use the graphics hardware for this question.

16.2 Ray Tracing – General

1. Describe naive ray tracing as introduced by Turner Whitted in 1979. What are its advantages/disadvantages as a rendering/hidden-surface algorithm?
2. Discuss and contrast the use of *space partitioning* and *bounding volumes* as a means of speeding up ray tracing.
3. Distributed ray tracing was first introduced in 1984 by Cook, Porter & Carpenter from Lucasfilm. How does it differ from the naive ray tracing algorithm? What are the advantages/disadvantages of distributed ray tracing?

16.3 Ray Tracing – Ray Intersect Object [Last Used: Fall 2010 Final]

1. Suppose we have a triangle $\triangle P_0P_1P_2$ and a ray from a point E in direction $\vec{v}$. How do you compute the intersection of this ray with this triangle, or determine that they do not intersect? Give a description, formulas and/or code to solve this in detail. If you wish to access coordinates, use P^x, P^y, etc., but you should make no assumptions about the values of any of the coordinates.

2. Suppose that this triangle is one of thousands on a polyhedral object, and that the vertices of all triangles have been consistently ordered in counter-clockwise order (i.e., to the "left" is on the "inside" of the triangle when viewing the triangle from the "outside").

   State generally how you can improve the speed of the intersection test when you know that P is on the outside of the polyhedron, and give a detailed description, formulas and/or code to solve in detail.

3. Suppose that this triangle is one of thousands on a polyhedral objects and you wish to decrease the exucution time by intersecting the ray with a bounding box that you place around the polyhedron.

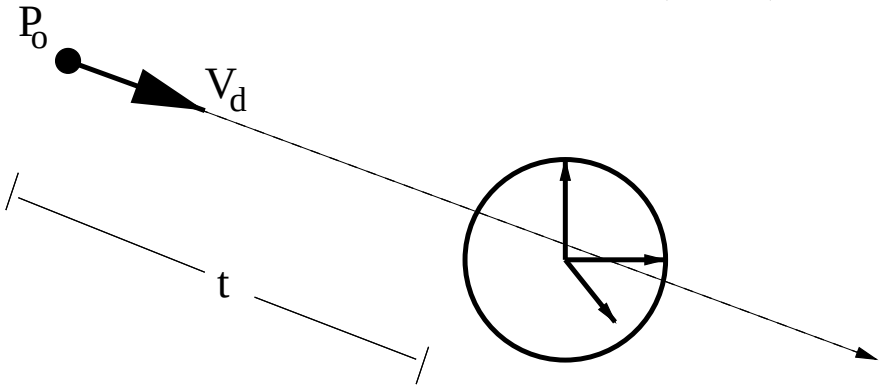
   Give a description, formula and/or code to compute this bounding box. Let $P_0, P_1, ...P_n$ be the vertices of the polyhedron and $T_i = \triangle P_{i_0}P_{i_1}P_{i_2}$ for $i = 0..F$ be the faces, with $0 \leq i_0, i_1, i_2 \leq N$.

16.4 Ray Tracing — Bounding Volumes [Last Used: Fall 2004 Final]

Two common bounding volumes used in ray tracing are bounding spheres and axis aligned bounding boxes. What are the advantages and disadvantages of each of these two types of bounding volumes?

16.5 Ray Tracing [Last Used: Winter 2000 Final]

Give a quadratic equation to find the distance, t, between a point $P_0 = (x_o,y_o,z_o)$ and a unit sphere centered at the origin along the normalized direction vector $V_d = (x_d,y_d,z_d)$.



16.6 Ray tracing a paraboloid [Last Used: Spring 2000 Final]

A paraboloid surface is given by the equation $z = (x-1)^2 + 3y^2$. We want to ray-trace this surface.

a) Give a rough sketch of this surface.

b) Write the paraboloid equation as a parametric equation in u and v, that is, as $[x(u,v), y(u,v), z(u,v)]^T$.

c) Given a ray at the origin $[0,0,0]^T$ and in the direction $[1,0,1]^T$, write a parametric equation for the ray as $[x(t), y(t), z(t)]^T$.

d) Compute the first point of intersection of the ray with the paraboloid. (Hint: this is really a 2D problem.)

e) For ray-tracing, we need a normal at the intersection point above. Compute this normal vector (you may want to make use of the hint above).

16.7 CSG and Ray Tracing [Last Used: Fall 2004 Final]

In Constructive Solid Geometry (CSG), it is difficult to explicitly represent the boundaries of the modeled objects. Briefly explain why this isn't an issue when ray tracing, indicating how the ray-intersect-scene calculation is performed on a CSG model.

16.8 Recursive Ray Tracing [Last Used: Fall 2002 Final]

You just bought a commerical raytracing package, but you didn't buy the source code. You've been told that the termination criteria for tracing rays is either (a) to always trace rays to a recursive depth of 4, or (b) to adaptively terminate based on ray contribution. Describe a scene that can be used to discern which method is used.

Be specific about the scene itself, and what you would look for in the resulting image to determine which termination method is used. Draw a sketch of your scene if possible.

16.9 Rays, rays, rays [Last Used: Fall 2010 Final]

For each of the following types of rays, describe the start position of the ray, where they go, and what they're used for.

1. Primary rays
2. Shadow rays
3. Reflected rays

16.10 Distribution Ray Tracing [Last Used: Winter 2006 Final]

In distribution ray tracing, by casting many rays in a random fashion (i.e., "distributing them") we can simulate many effects. The "random fashion" part is what allows us to achieve our effects. Describe what types of rays get "distributed" and what they "get distributed over" to achieve each of the following effects. We have provided a model answer for "Anti-aliasing" to better indicate what you should give for your answers.

Anti-aliasing — distribute **primary** rays over **pixel area**.

1. Soft shadows — distribute rays over

2. Motion blur — distribute rays over

3. Depth of field — distribute rays over

16.11 Ray tracing lighting [Last Used: Winter 2006 Final]

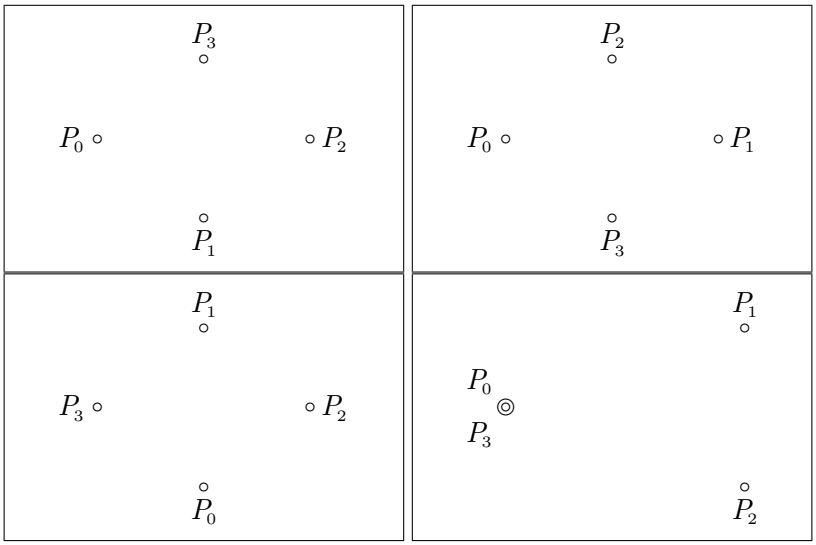
Consider a ray $(P, \vec{v})$ that hits a surface S at a point Q, where the normal to S at Q is $\hat{n}$, and where S has only diffuse reflection material components (d_r, d_g, d_b). Further, assume there is a white point light at position L with RGB intensity values (I, I, I), and a white ambient light with RGB intensity values (A, A, A).

In ray tracing, we want to know the intensity of light that travels back along this ray and eventually reaches the eye. For the situation described above, explain how we compute this intensity, noting any additional rays we cast and any local lighting computations we perform.

17 Splines

17.1 Bézier Curves: Evaluation and Properties [Last Used: Fall 2009 Final]

The following pictures show possible arrangements of control points for a cubic Bézier curve.



(a) In the case of each picture, make as accurate a sketch of the curve as possible.

A cubic Bézier curve has the mathematical form

$$C(t) = \sum_{i=0}^{3} P_i b_i(t)$$

(b) Give explicit formulas for $b_0(t), \ldots, b_3(t)$.

(c) Show that $b_i(t) \geq 0$ for $0 \leq t \leq 1$.

(d) Show that $\sum_{i=0}^{3} b_i(t) = 1$ for all t.

(e) What implications do **(c)** and **(d)** have in locating the points of $C(t)$?

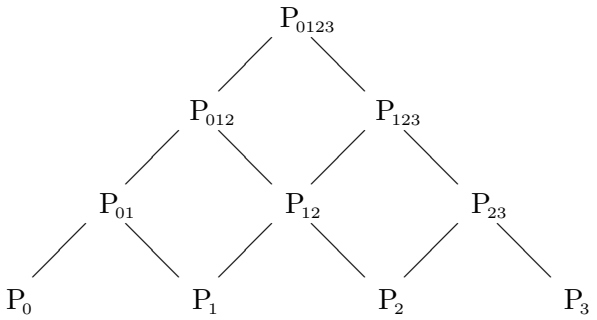
(f) Suppose $\mathbf{R}$ is a rotation matrix. Show that $\mathbf{R}C(t)$ is given by the Bézier curve that has control points $\mathbf{R}P_i$.

(g) Sometimes a Bézier curve is presented in the format

$$C(t) = [t^3, t^2, t^1, 1]\mathbf{B}\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Show how to find the matrix $\mathbf{B}$ from the formulas in part (b) above.

(h) Subdivision of the segment at the parametric value a is given by a process that can be visualised by means of a pyramid of calculations:



Complete this pyramid diagram by putting appropriate expressions on the edges. Make an enlarged copy of picture 1 and draw in and lable the locations where the points $P_0, P_{01}, P_{012}, P_{0123}$ would appear when $a = 1/3$.

17.2 Bilinear Patches [Last Used: Spring 1992 Final]

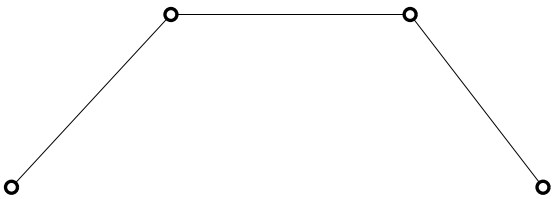
A *bilinear* patch is given by

$$(1-\alpha)(1-\beta)P_{00} + (1-\alpha)\beta P_{01} + \alpha(1-\beta)P_{10} + \alpha\beta P_{11}$$

If the patch is subdivided at $\alpha = 1/2$ and $\beta = 1/2$, the four resulting patches involve control points Q_{00}, Q_{01}, Q_{02}, Q_{10}, Q_{11}, Q_{12}, Q_{20}, Q_{21}, and Q_{22} Give formulas for the Q's in terms of the P's.

17.3 The de Casteljau Algorithm [Last Used: Winter 1996 Final]

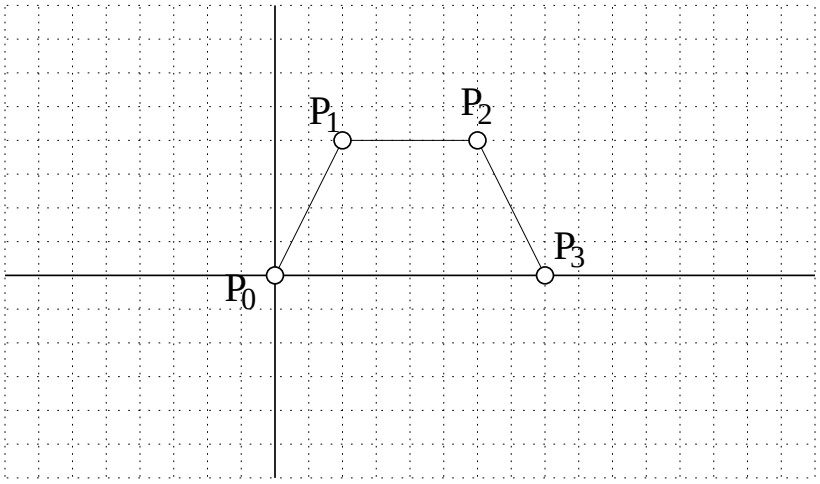
1. Show the de Casteljau construction on the Bézier curve shown below at $t = 0.5$ (assume the curve is parameterized over [0,1]).



2. In the figure above, draw the tangent to the curve at the point $t = 0.5$.

17.4 The de Casteljau Algorithm [Last Used: Fall 1997 Final]

In the figure below are the control points for a cubic Bézier curve P, defined over the interval $[0, 1]$ by the control points P_0, P_1, P_2, and P_3.



Suppose we want to join another cubic Bézier curve segment Q (defined over the interval $[1, 2]$) with control points Q_0, Q_1, Q_2, and Q_3, and we wish P and Q to meet with C^1 continuity at parameter value 1 (i.e., the first derivatives should be equal at $t = 1$).

- In the figure above, draw and label any control points of Q whose placement is forced by the continuity conditions. List any control points whose placement is **not** forced by the C^1 conditions here:

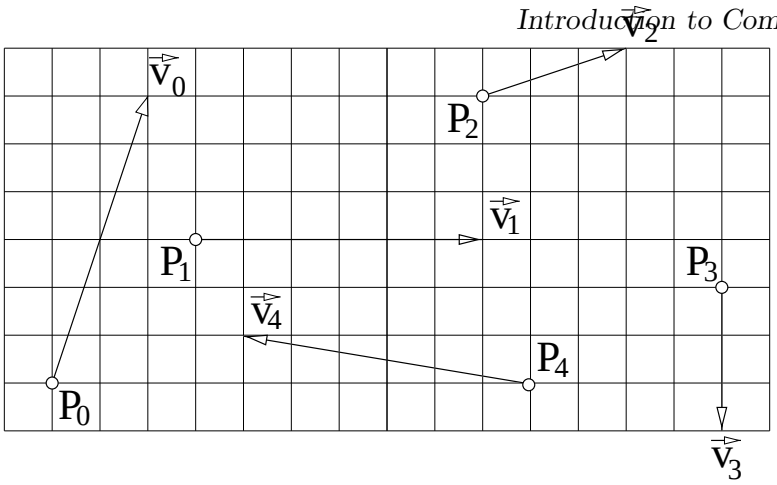
- Draw the de Casteljau evaluation of P at $t = 1/2$ in the figure above. Label $P(1/2)$ on your drawing.

17.5 Cubic Hermite with Bézier Segments [Last Used: Winter 1998 Final]

In cubic Hermite interpolation, we are given a set of n points and n vectors, and we want to find a piecewise cubic C^1 curve that interpolates the points, where the first derivative of the curve at these points should be the specified vectors.

Given the points and vectors below, draw and label the cubic Bézier control points for a piecewise cubic that interpolates the points with derivatives that interpolate the vectors. Your curve P should interpolate the ith data value at parameter value i, i.e., $P(i) = P_i$ and $P'(i) = \vec{v}_i$.

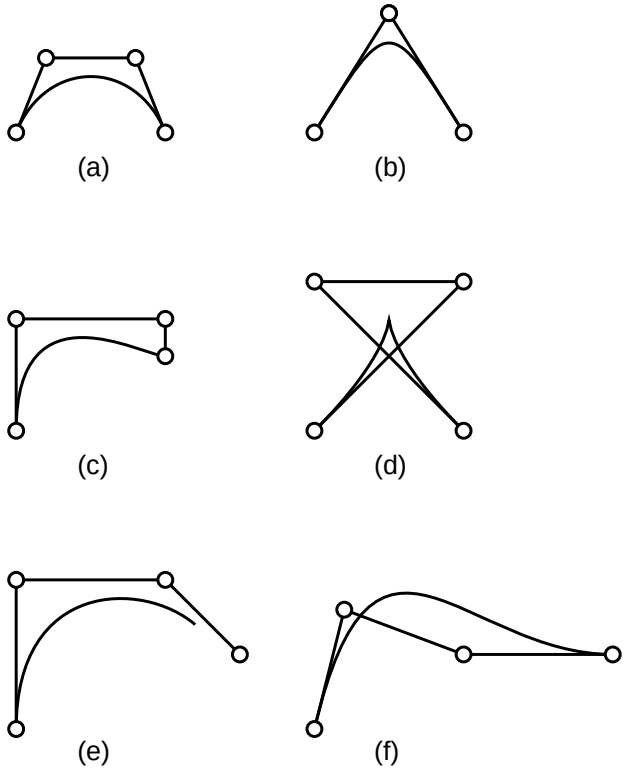
Label the ith control point of the jth Bézier segment as P_i^j.



17.6 Cubic Bézier Curves [Last Used: Fall 2002]

Below are six curves and their "control points/polygon." Two of the control polygons are the Bézier control polygon for the curve drawn with it; the other four are not. Indicate which of the two control polygons are Bézier control polygons for the corresponding curve and which four are not Bézier control polygons for the corresponding curve. Justify your answer for the control polygons that are not Bézier control polygons.

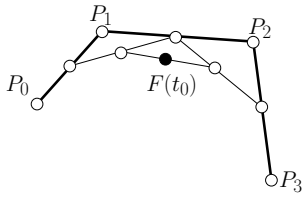
You may assume that none of the control points overlap or are repeated.



17.7 Cubic Bézier Curves [Last Used: Winter 2006 Final]

Suppose we have a cubic Bézier curve F with control points P_0, P_1, P_2, P_3, with F parameterized over the interval $[0, 1]$.

The following is the de Casteljau diagram for evaluating a curve F at $t_0 \in [0, 1]$:



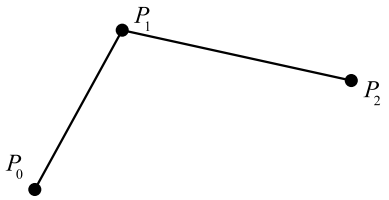
$F(t_0)$ is located at the black point in the diagram above.

1. Show the positions of $F(0)$ and $F(1)$ in the diagram above.
2. Indicate in the diagram the tangent line to F at $F(t_0)$.
3. We wish to join to F a second cubic Bézier curve $G(t)$ parameterized over $[1, 2]$ with control points G_0, G_1, G_2, G_3.

   Give formulas for positioning some or all of G's control points to have F and G meet with C^1 continuity at $t = 1$.

17.8 Quadratic Bézier Curves [Last Used: Winter 2009 Final]

1. The diagram below shows three control points P_0, P_1 and P_2 that together define a quadratic Bézier spline $P(t), t \in [0, 1]$. Use de Casteljau's algorithm to evaluate the spline at $t = \frac{1}{3}$. Sketch the intermediate lines and points that are used in the algorithm and clearly label the point $P(\frac{1}{3})$.

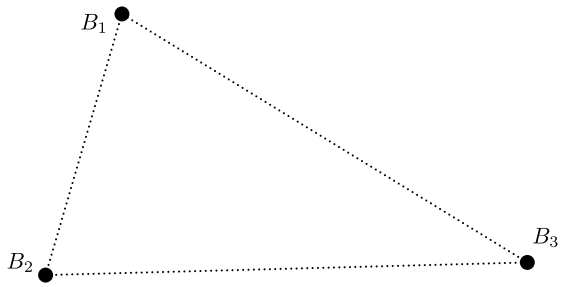


2. Based on the formula for an affine combination of two points, use de Casteljau's algorithm to derive an explicit equation for $P(t)$ in terms of P_0, P_1 and P_2.
3. Suppose that we wish to place three additional control points Q_0, Q_1 and Q_2 defining a second Bézier spline segment, parameterized over $[1, 2]$. Give formulas for whichever of the Qs are necessary to guarantee that the two curve segments will be joined with C^1 continuity.
4. Suppose now that we have three control points P_0, P_1 and P_2 where P_1 lies at the midpoint of the line segment joining P_0 and P_2. Show that as the parameter t grows at a constant rate from 0 to 1, a particle tracing out the curve $P(t)$ will move along the line from P_0 to P_2 with a constant velocity.

(a) Draw and label four points P_0, P_1, P_2 and P_3 such th at the cubic Bézier curve $P(t)$ derived from them is a simple, closed curve. Sketch the resulting curve freehand, as accuratel y as you can. (A curve is closed if it starts and ends at the sam e point. A closed curve is simple if it never touches itself anyw here else.)

(b) Explain why it would be impossible to construct a simple, closed cubic Bézier curve that has C^1 continuity everywhere. Incl ude a small diagram if you want.

(c) The diagram below shows three B-Spline control points B_1, B_2 and B_3, arranged in a closed loop. These three control point s give rise to three cubic Bézier segments that meet each other with C^2 continuity. Draw and label the twelve Bézier control points $\{P_0, P_1, P_2, P_3\}$, $\{Q_0, Q_1, Q_2, Q_3\}$, and $\{R_0, R_1, R_2, R_3\}$. (It doesn't matter which segment gets the Ps, which gets the Qs, and which gets the Rs, as long as they are labelled consistently.)



(d) Recall that a Bézier curve can be constructed from $n+1$ contr ol points $P_0, \ldots, P_n$ using the equation

$$P(t) = \sum_{i=0}^{n} B_i^n(t) P_i \qquad \text{where} \qquad B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i.$$

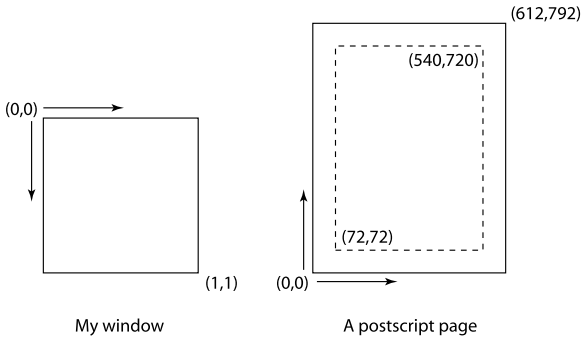
Assume that points P_0, P_1, P_2 and P_3 define a cubic Bézier segment in the plane. Suppose that we wish to draw a second cubic Bézier segment defined by points Q_0, Q_1, Q_2 and Q_3 in such a way that the two segments join with C^2 continuity. Give explicit formulas for the locations of Q_0, Q_1 and Q_2, each in terms of the Ps, that guarantee C^2 continuity. (You can derive the formulas by drawing and examining a diagram of the two segments, or by working from the mathematical definition directl y.)

(e) **Bonus.** Is it possible to position control points $\{P_0, P_1, P_2, P_3\}$ and $\{Q_0, Q_1, Q_2, Q_3\}$ in such a way that the two cubic Bézier segments together form a simple closed curve that is C^2 everywhere? Either show control points that realize this construction, or prove that it is impossible.

18 Transforms

18.1 Viewport Transformation [Last Used: Winter 2004 Midterm]

I have a picture that I've expressed in a normalized 2D coordinate system with top left corner $(0, 0)$ and bottom right corner $(1, 1)$ (y grows downward, as in screen coordinates). I'd like to print the picture as PostScript, centered on an 8.5" $\times$ 11" page with a one-inch margin. One unit in PostScript is equal to $\frac{1}{72}$". The picture should be centered on the page, and scaled uniformly to be as large as possible while still fitting within the margins. The two coordinate systems are shown below.



Write a short C function or give a matrix that takes normalized coordinates and outputs PostScript coordinates.

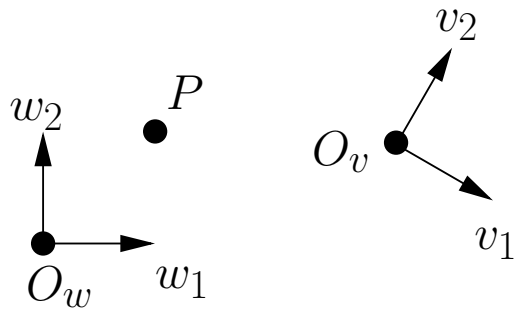
18.2 Matrix Transformations [Last Used: Fall 2004 Midterm]

Let T be the matrix

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

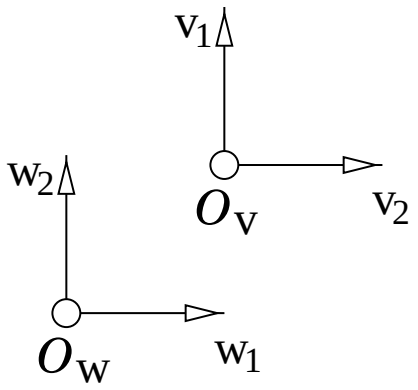
1. Suppose T represents a transformation mapping points specified relative to frame $F_w = \{w_1, w_2, O_w\}$ to points specified relative to frame $F_v = \{v_1, v_2, O_v\}$. In the figure below, draw $T(P)$ where P's coordinates relative to F_w are $P = [1, 1, 1]^T$. Be sure to label $T(P)$ in the figure.

2. Now suppose that T represents a change of coordinates from frame F_w to a frame $F_q = \{q_1, q_2, O_q\}$. In the figure below, draw and label the frame elements of F_q.



18.3 Change of basis, geometric transformation [Last Used: Winter 1998 Midterm]

In the figure below are two coordinate frames, $F_W = \{w_1, w_2, O_W\}$ and $F_V = \{v_1, v_2, O_V\}$ with $v_1 = w_2$, $v_2 = w_1$, and $O_V = O_W + v_1 + v_2$ (note the relationship between the basis vectors). Both coordinate frames are orthonormal.



1. Give the change of basis matrix mapping coordinates relative to F_W to coordinates relative to F_V.

2. Give the transformation matrix for the geometric transformation that maps O_W onto O_V, w_1 to v_1 and w_2 to v_2. Assume that the coordinate representation of both the domain and range are relative to F_W.

18.4 Composition of Transformations [Last used: Fall 1997 Midterm]

In this question, we are working in 2 dimensions. All transformations map from the standard Cartesian frame to the standard Cartesian frame.

Let $R(\theta)$ be the matrix for a rotation about the origin of θ in the counter-clockwise direction.

Let $T(\vec{v})$ be the matrix for a translation by $\vec{v}$.

Let $S(s_x, s_y)$ be the matrix for a non-uniform scale about the origin by an amount s_x in the x direction and s_y in the y direction.

Given a point p, two perpendicular unit vectors $\vec{v}$ and $\vec{w}$, and two scale factors a and b, suppose we want to perform a non-uniform scale about p by a in direction $\vec{v}$ and b in direction $\vec{w}$. Give the appropriate matrix product to achieve this transformation using the above notation for the matrices.

Note: You should *not* give expanded forms of the matrices; instead, your matrix product should be products of $R()$, $T()$, and $S()$ (each operation may be used more than once). Also, these should be treated as matrices and not transformations (which is important for the order). Further assume that points and vectors are represented as column matrices.

18.5 Model, Viewing, and Perspective Transformations [Last Used: Fall 1996 Final]

For assignment 3, you implemented hierarchical transformations that could transform the view frame relative to itself, and the modeling transformation relative to the current model frame.

If PVM is the composition of the perspective transformation, the world-to-viewing transformation, and the model-world transformation respectively (with M embodying both the modeling transformations and the Model-to-World change of basis) give the composition of transformations in each of the following situations. Note: for each case, assume you start from PVM.

1. If we transform the view frame by a transformation embodied in a matrix T represented relative to viewing coordinates.

2. If we transform the view frame by a transformation embodied in a matrix T represented relative to world coordinates.

3. If we transform the model frame by a transformation embodied in a matrix T represented relative to current model coordinates.

4. If we transform the model frame by a transformation embodied in a matrix T represented relative to world coordinates.

5. If we use an alternative perspective transformation P'.

18.6 Transformations [Last Used: Winter 1996 Midterm]

Suppose we have a transformation T, points $P = [1, 1, 1, 1]^t$, $Q = [3, 2, 2, 1]^t$, $R = [5, 3, 3, 1]^t$. and the vector $\vec{v} = P - Q = Q - R$ (all points are specified relative to the standard Cartesean coordinate frame $F = (\vec{\imath}, \vec{\jmath}, \vec{k}, \mathcal{O})$).

Note: Some of the operations described below are invalid. In these cases, state why the operation is invalid.

1. Suppose T is a change of basis transformation that maps from the standard coordinate frame to the coordinate frame $F_2 = (2\vec{\imath}, \vec{\jmath}, \vec{k}, \mathcal{O} + \vec{\jmath})$.

   - Give the matrix representation of T.
   - Give the coordinates of P relative to F_2.
   - What is $|T(P) - P|$?
   - Give the coordinates of $\vec{v}$ relative to F_-.
   - What is the length of $T(\vec{v})$?

2. Suppose T is a scale by a factor of 2 in the $\vec{\imath}$ direction.

   - Give the matrix representation of T (assuming its domain and range frames are both F).
   - Give the coordinates of $T(P)$ relative to F.
   - What is $|T(P) - P|$?
   - Give the coordinates of $T(\vec{v})$ relative to F.
   - What is the length of $T(\vec{v})$?

3. Suppose T is the projective transformation for when we are looking down the positive z axis and projecting through $\mathcal{O}$. Further, suppose that T maps P to $(1, 1, -1, 1)$ and R to $(5/3, 1, 1, 1)$. The matrix representation for this mapping is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & -3 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

where the domain frame is F and the range frame is an orthonormal frame for the projection space after normalization.

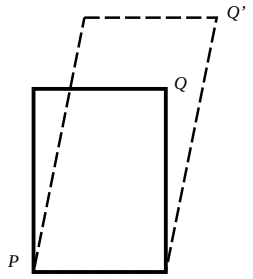
- Give the normalized coordinates of $T(Q)$.
- What is $|T(P) - P|$?
- Give the normalized coordinates of $T(\vec{v})$.
- What is the length of $T(\vec{v})$?
- What are the normalized coordinates of $T(\mathcal{O}-?$

18.7 Rotation About an Arbitrary Axis [Last Used: Spring 1994 Final]

Describe, in general terms, how you would accomplish a rotation about an arbitrary 3D line through a point p in direction $\vec{v}$ (i.e., $l(t) = p + t\vec{v}$). Illustrate with a diagram.

18.8 Stroke Fonts [Last Used: Spring 1997 Midterm]

The Hershey stroke fonts define each character in its own local, rectangular coordinate system. Suppose we wanted to slant or skew every character to the right, simulating the effect of handwriting. This could be accomplished by a 2D transformation in the xy plane that maps a rectangle into a parallelogram as depicted below:



Assuming that the lower left corner of the rectangle is P and the upper right corner of the rectangle is Q, the transformation is completely determined by observing that P maps into P and Q maps into Q'.

(a) Write the equations to accomplish this transformation.

(b) Write the associated homogeneous transformation matrix.

18.9 World/viewing transformations [Last used: Winter 1995 Midterm]

Suppose we have two coordinate frames in a three dimension space. One frame, $VF = (\vec{v}_0, \vec{v}_1, \vec{v}_2, \mathcal{V})$ is the coordinate frame specifying the current view. The other frame, $WF = (\vec{w}_0, \vec{w}_1, \vec{w}_2, \mathcal{W})$, represents the world coordinate system. All vectors and points are specified in world coordinates as follows:

$$\begin{aligned}
\vec{v}_0 &= [vx_0\ vy_0\ vz_0\ 0] \\
\vec{v}_1 &= [vx_1\ vy_1\ vz_1\ 0] \\
\vec{v}_2 &= [vx_2\ vy_2\ vz_2\ 0] \\
\mathcal{V} &= [vx_3\ vy_3\ vz_3\ 1] \\
\vec{w}_0 &= [1\ 0\ 0\ 0] \\
\vec{w}_1 &= [0\ 1\ 0\ 0] \\
\vec{w}_2 &= [0\ 0\ 1\ 0] \\
\mathcal{W} &= [0\ 0\ 0\ 1]
\end{aligned}$$

We have a perspective transformation matrix P for projecting our view to NDC coordinates with the desired field of view and aspect ratios.

We have another routine `CoC(`$\mathcal{W}$`, `$\mathcal{V}$`)` that returns a matrix representing a change of coordinate from frame $\mathcal{W}$ to frame $\mathcal{V}$.

Assume that the perspective divide is performed by the hardware.

In each of the following, give the matrix needed to draw what's specified. You should give the matrix as a product of several matrices, some of which may be returned from the two routines listed above. You should explicitly write any other matrix you need. For example, the matrix needed to draw the scene as viewed from the current viewing position is

$$\texttt{CoC}(WF, VF) \circ P$$

You may construct new coordinate frames and intermediate points and vectors if you need them, but you should give formulae for these constructions.

1. Draw the scene after translating the view frame by 2 units in direction $\vec{v}_0$.
2. Draw the scene after translating the view frame by 3 units in direction $\vec{w}_1$.
3. Draw the scene from the current view after translating the rest of the scene by 4 units in direction $\vec{v}_2$.
4. Draw the scene from the current view after translating the rest of the scene by 5 units in direction $\vec{w}_1$.
5. Draw the scene from the current view after scaling the scene by 6 units in the $\vec{w}_0$ direction and 7 units in the $\vec{w}_1$ direction.

18.10 Parallel Lines, Affine Maps [Last Used: Winter 1994 Final]

Do parallel lines map to parallel lines under affine transformations? If so, then prove it. If not, give a counter example.

18.11 Orthographic Projections [Last Used: Spring 1994 Midterm]

The transformation T mapping $[x, y, z, w]$ to $[x, y, w]$ (where w may be either 0 or 1) is an example of an orthograhic projection. Prove that T is an affine transformation.

18.12 Coordinate Frames [Last Used: Winter 1993 Midterm]

Let $\mathcal{F}_\infty$ be a Cartesian frame in a two space. Let $\mathcal{F}_\in$ be a frame given by $\{[\frac{1}{2}, 0, 0], [0, 1, 0], [1, 0, 1]\}$ relative to $\mathcal{F}_\infty$. Let P, Q, and $\vec{v} = Q - P$ have the following coordinates relative to $\mathcal{F}_\infty$:

$$\begin{aligned} P &= [1, 1, 1] \\ Q &= [2, 1, 1] \\ \vec{v} &= [1, 0, 0]. \end{aligned}$$

Let C be the *coordinate transformation* from $\mathcal{F}_\infty$ to $\mathcal{F}_\in$ with matrix representation M:

$$M = \begin{vmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{vmatrix}.$$

Let T be an *affine transformation* whose matrix representation is M when mapping points with coordinates relative to $\mathcal{F}_\infty$ to points with coordinates relative to $\mathcal{F}_\infty$

In the following, assume that P, Q, and $\vec{v}$ are represented relative to $\mathcal{F}_\infty$.

1. Give the coordinates of $T(P)$, $T(Q)$, and $T(\vec{v})$ relative to $\mathcal{F}_\infty$.
2. Give the coordinates of $T(P)$, $T(Q)$, and $T(\vec{v})$ relative to $\mathcal{F}_\in$.
3. Give the coordinates of $C(P)$, $C(Q)$, and $C(\vec{v})$ relative to $\mathcal{F}_\in$.
4. What are the lengths of $\vec{v}$, $T(\vec{v})$, and $C(\vec{v})$?

18.13 Inverse Transformations [Last Used: ?]

In general, given a transformation T represented by a matrix M, we must invert M to find the matrix representation of T. Suppose, however, M is represented as the composition of transformations:

$$M = M_1 \circ M_2 \circ \ldots \circ M_n,$$

where each M_i is a simple translation, rotation about the origin, or scalings about the origin. Find an expression for M^{-1} that can be computed without explicitly inverting M.

18.14 Perpendicular Lines, Affine Maps [Last Used: Winter 1993 Final]

Do perpendicular lines map to perpendicular lines under affine transformations? If so, then prove it. If not, give a counter example.

18.15 Projective Transformations [Last Used: Winter 2000 Midterm]

For the following question, assume that we are viewing the scene by looking down the negative z-axis of a right handed VCS.

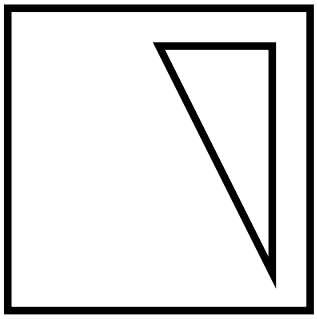
a) If we project an arbitrary point (x, y, z, 1) onto the near plane at z=-n, what are the coordinates of the projected point?

b) Give a matrix that performs the transformation from part (a), assuming the usual normalization (division by the last coordinate) will subsequently be applied.

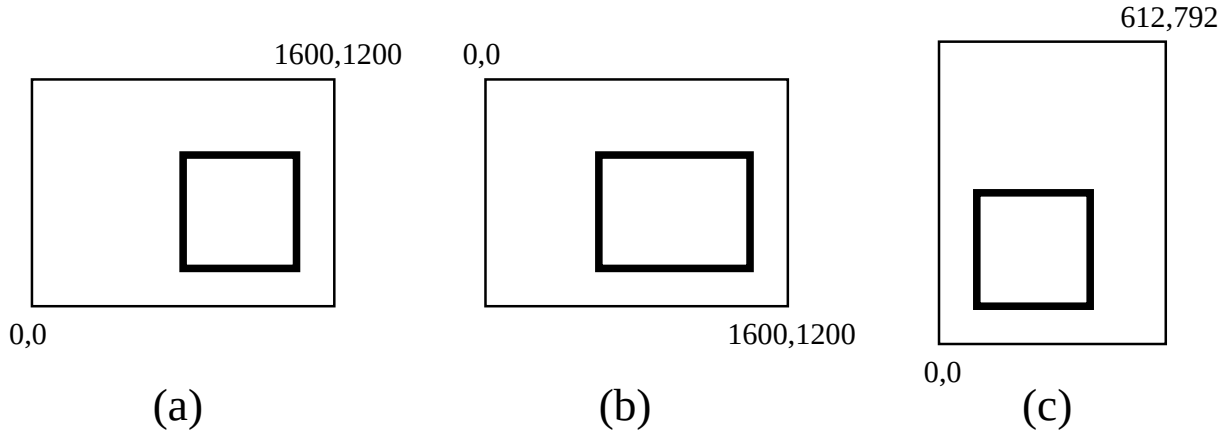
c) Suppose we want to map the z values so that the near plane maps to z=a and the far plane maps to z=b (instead of all z values mapping to -n) while mapping x and y to the same values as in (a) and (b). Give a matrix to perform this transformation.

18.16 Window to Viewport Transformation [Last Used: Winter 2009 Final]

After projection, suppose we have a triangle appearing in our window as shown in the following diagram:



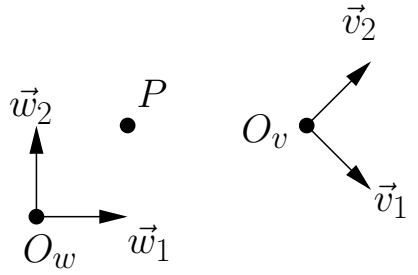
Sketch how this triangle will map to each of the following viewports on our output device (the large rectangle is the output device, with two of its corners labeled with pixel coordinates; the small rectangle is the viewport on our output device).



18.17 Change of Coordinates [Last Used: Winter 2006 Midterm]

Suppose we have two coordinate frames $F_w = \{\vec{w}_1, \vec{w}_2, O_w\}$ and $F_v = \{\vec{v}_1, \vec{v}_2, O_v\}$ as shown below, with

$$\vec{v}_1 = \frac{\sqrt{2}}{2}\vec{w}_1 - \frac{\sqrt{2}}{2}\vec{w}_2, \quad \vec{v}_2 = \frac{\sqrt{2}}{2}\vec{w}_1 + \frac{\sqrt{2}}{2}\vec{w}_2, \quad O_v = O_w + 3\vec{w}_1 + \vec{w}_2$$



1. What are the coordinates of P relative to F_w?
2. What are the coordinates of P relative to F_v?
3. Give the change of coordinates matrix M_{wv} mapping from F_w to F_v.
4. Give the change of coordinates matrix M_{vw} mapping from F_v to F_w.
5. What is $M_{wv}M_{vw}$?

18.18 Transformations [Last Used: Winter 2007 Midterm]

Suppose you are given a triangle in 2D with vertices at positions $(0, 0)$, $(1, 0)$ and $(0, 1)$. In the following, assume the positions of these vertices are expressed with homogeneous column vectors as $\mathbf{a} = [0, 0, 1]^T$, $\mathbf{b} = [1, 0, 1]^T$, and $\mathbf{c} = [0, 1, 1]^T$, using an orthonormal basis in which the first unit-length basis vector points to the right and the second points up.

1. Give a transformation matrix that translates this triangle by 5 units to the left.
2. Give a transformation matrix that rotates this triangle counterclockwise by 90 degrees.
3. Give a transformation matrix that translates this triangle by 5 units to the left and also rotates by 90 degrees counter-clockwise about its own origin (that is, about point $\mathbf{a}$ in its original coordinate system).
4. Suppose under some affine transformation T the triangle is mapped to $\mathbf{a}' = [4, 2, 1]$, $\mathbf{b}' = [2, 4, 1]$ and $\mathbf{c}' = [4, 4, 1]$. Find the matrix that represents this transformation.

18.19 Transformations [Last Used: Winter 2007 Final]

In the following, assume Δ is a 2D triangle with vertices at $A = [1, 1]^T$, $B = [3, 2]^T$, and $C = [2, 3]^T$. Suppose $P = \alpha A + \beta B + \gamma C$ with $\alpha = 1/6$, $\beta = 1/3$, and $\gamma = 1/2$.

1. Is $P = \alpha A + \beta B + \gamma C$ a valid affine combination? Justify your answer.

2. Is the point P in the interior of the triangle? Justify your answer.

3. Suppose points A, B, and C are transformed by an affine transformation T. Give an expression with numerical values for the weights that expresses $T(P)$ as an affine combination of $T(A)$, $T(B)$, and $T(C)$.

4. Compute the coordinates of P in the same frame as A, B, and C.

5. Compute the coordinates of P relative to the frame formed by the vector $B - A$ and $C - A$ with origin A.

19 Short Answer

1. Suppose you are using Euler angle rotations to manipulate the orientation of an object by rotating around the coordinate axes. Suddenly, you encounter Gimbal lock. Describe the effect of Gimbal lock and state how you would "break the lock".

2. Photon mapping traces rays from a light source into the scene in order to achieve forward ray tracing affects such as caustics. One forward effect of light is the separation of white light into a rainbow of colours by shining the white light through a prism due to different wavelengths refracting different amounts. What difficulties would you encounter in trying to reproduce this effect with photon mapping?

3. What is the mathematical difficulty associated with using piecewise cubic splines for animating the path of a moving object at a constant speed?

4. Give one advantage and one disadvantage of using motion capture to create animation.

5. Name and describe two different styles of non-photorealistic rendering.

6. Name and describe two objectionable visual artifacts that are caused by aliasing.

7. Name and describe two advanced visual phenomena, other than antialiasing, that can be computed using Monte Carlo techniques (that is, by averaging random samples).

8. The Blinn-Phong lighting model uses the half-vector $\vec{h} = (\vec{v} + \vec{l})/|\vec{v} + \vec{l}|$ and the expression $(\vec{n} \cdot \vec{h})^k$ in its specular term. Describe the motivation behind the use of the half-vector in this lighting model.

9. Give at least one situation where a bounding box hierarchy would would *decrease* the performance of a raytracer.

10. Name one visual effect that can be rendered with a photon mapping algorithm but not a raytracer, and describe why a raytracer cannot render this effect.

11. Physically-based lighting models should not emit more energy than they absorb. However, some common lighting models, such as the Phong lighting model, do not satisfy this property. Why can this be a problem for global illumination algorithms?

12. What does it mean for a polygon to be convex?

13. Is the intersection of two convex polygons necessarily convex? Why or why not?

14. Explain what homogeneous coordinates are and why we use them.

15. Explain what metamers are and say briefly what their existence implies about our perception of colour.

16. Here are four steps in a typical 3D graphics pipeline:

    (a) Apply a perspective projection

    (b) Clip against the near plane

    (c) Clip against the other frustum planes

    (d) Divide by the homogeneous coordinate

    Give an order for these four steps that produces correct results.

17. Explain why you don't need to preserve depth information during projection if you're using the painter's algorithm.

18. In key-frame animation, points on objects follow paths over time that can be displayed as curve plots to the user. What physical property does the first derivative of a position versus time plot represent? What does the second derivative represent?

19. What are "key frames" and "in between frames" in computer assisted key frame animation?

20. Give one reason that Quaternions are preferred over Euler Angles for key framing orientations.

21. Give one reason that animating human walking motion is hard.

22. How do we identify silhouette edges in a polyhedral model?

23. Show how we intersect a parametric ray $r(t) = p + t\vec{v}$ with an implicit surface $f(P) = 0$.

24. In ray tracing, bump mapping can be used to model ripples in water. Describe an artifact that might be visible when using bump mapping in this manner.

25. In radiosity, *form factors* are computed between every possible pair of patches. What does a form factor represent?

26. A real-time graphics program can use a single frame buffer for rendering polygons, clearing the buffer, and repeating. Why do we usually use two buffers instead?

27. Any point in triangle ABC may be written as a *barycentric* combination of the vertices. Write a general expression for point P as a barycentric combination of A, B, and C, including any constraints on the coefficients. Where is the point if one coefficient is negative?

28. Two cubic Bezier curves, with control points P_0, P_1, P_2, P_3 and Q_0, Q_1, Q_2, Q_3 respectively, are joined together at $P_3 = Q_0$. What must be true for them to meet with C^1 continuity?

29. If the curves in part (e) above meet instead with G^1 continuity, what must be true?

30. Given a tensor product Bézier patch

$$B(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} P_{i,j} B_i^3(u) B_j^3(v)$$

with u, v defined over the unit square, give the formula for the normal to the patch at $u = v = 0$. (Do not worry about the sign of the normal.)

31. The Painter's algorithm for hidden surface removal is not an *on-line* algorithm, but z-buffering is. What do we mean by an on-line algorithm for hidden surface removal?

32. For each of the following rendering algorithms, indicate whether it is view-dependent or view-independent.

    Ray tracing:

    Radiosity:

    Photon mapping:

33. To render caustics with photon mapping, light rays are bounced off one type of surface and intensities stored at a second type of surface. What material properties do those two types of surfaces have?

    Bounced surface material:

    Stored surface material:

34. In Warnock's algorithm for hidden surface removal, what two situations constitute a *simple polygon list in viewport* problem?

35. Of these material properties—ambient, diffuse, and specular—which is modelled worst by radiosity? Also, what characteristic of shadows is often an indication that a radiosity algorithm was used?

36. Give an example of a set of basis colours for an additive colour scheme and a device for which such a scheme would be appropriate.

37. Give an example of a set of basis colours for a subtractive colour scheme and a device for which such a scheme would be appropriate.

38. List two virtual device with two degrees of freedom that could be used to input both a rotation axis and an angle.

39. Briefly describe the difference between an affine transformation and a linear transformation.

40. Give the general form for an affine combination expressing a vector V in terms of a set of points P_i. Indicate any necessary constraints.

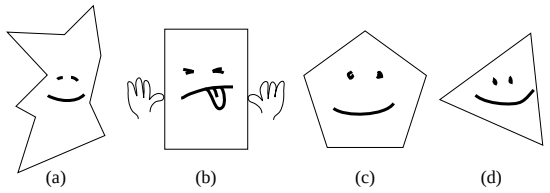
41. Give the general form for an affine combination expressing a point Q in terms of a set of points P_i. Indicate any necessary constraints.

42. Suppose T is a transformation acting on points like Q in the part above. If T is an arbitrary transformation, it may map Q to any point. If, however, T is an affine transformation, what constraint must it satisfy in terms of the points P_i?

43. Name two general guidelines for creating 3D models that help ensure proper hidden surface removal and rendering of the objects.

44. Briefly describe, with any appropriate equations, the algorithm for removing (or "culling") backfacing polygons. Assume that the normal points out from the visible side of the polygon.

45. Which of the following polygons is making fun of you? Circle one.



46. A ray $(P, \vec{v})$ hits a surface S at a point Q, where the normal to S at Q is $\hat{n}$. Draw a picture and give the formula for computing the reflected ray $\vec{r}$ emanating from Q.

47. There is one special situation where backfacing polygon removal, as in the part above, is a reasonable hidden surface removal technique. What types of 3D models must we use in this case?

48. Describe a situation in which backface culling is insufficient for hidden surface removal, and a situation in which you might not wish to use backface culling.

49. What is the difference between *polling* an input device, and *sampling* an input device?

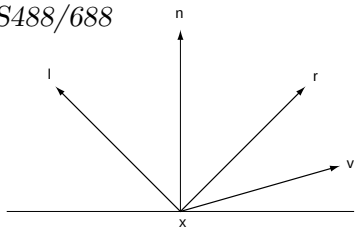
50. Is the multiplication of two 3D translation matrices commutative? Two scaling matrices? Two rotation matrices?

51. If traversing a DAG structure applies a matrix M to a point p in a leaf node, what matrix should be applied to a tangent vector? To a normal vector?

52. Does the basis represented by $v_x = [1, 0, 0, 0]^T, v_y = [1/\sqrt{2}, 0, -1/\sqrt{2}, 0]^T, v_z = [0, -1, 0]^T, O_v = [0, 0, 0, 1]$ represent a right-handed or a left-handed coordinate system?

53. Describe in words or by an equation a *cross-ratio*, the quantity preserved by a projective transformation.

54. For lighting calculations, a point x on a surface S is the origin of a normal vector n, a unit vector l to the light source, and a unit vector v to the viewpoint. For specular reflections and Phong shading, a unit vector r is reflected away from x in the same plane as l and n, and making the same angle with n as l does but on the reflected side. Give an equation for unit vector r in terms of the vectors above.

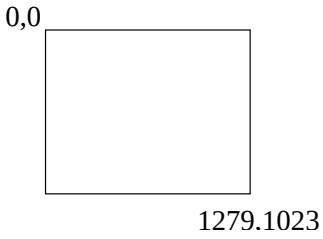


55. What effect does the Phong lighting term ($k_s(\vec{r} \cdot \vec{v})^p$) attempt to model? What effect does varying p have?

56. What is a matrix stack and how is it used in computer graphics?

57. Suppose we have a 1280x1024 raster display, with a single window/viewport that covers the entire display, with pixel 0,0 in the upper left hand corner as illustrated below.

    In the picture, give a reasonable sketch indicating how normalized device coordinates would map on to this display. Further, give the NDC coordinates of at least two points in the picture.



58. The Lambertian lighting term approximates the amount of light coming in from a light in direction $\hat{\ell}$, reflecting off a matte surface at a point with normal $\hat{n}$, and reflecting in direction $\hat{v}$. The Lambertian lighting term is

$$I_o = k_d I_i(\hat{\ell} \cdot \hat{n}),$$

    where k_d is an RGB triple describing the material properties of the surface, and I_i is an RGB triple of intensities of the light coming into the surface, and I_o is the intensity of light leaving in direction $\hat{v}$.

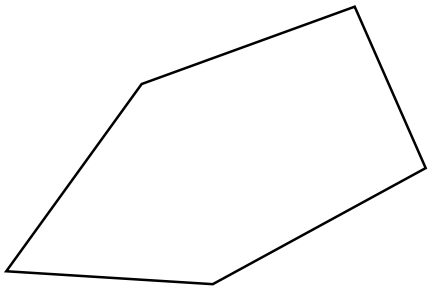
    Why doesn't the outgoing direction $\hat{v}$ of the light enter into this equation?

59. Why do you have to clip to the near plane before doing perspective projection (or at least, before doing the normalization step)?

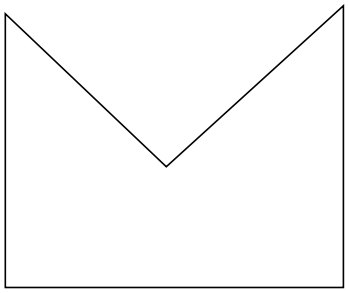
60. Draw a picture of a set of simple polygons that the Painter's algorithm can not render without splitting the polygons.

61. In class, we studied the Liang-Barsky clipping algorithm for lines (i.e., the one for lines, not the one for polygons). Typically this algorithm is used to clip lines to an axis aligned, rectangular window.

Can the Liang-Barsky algorithm be used to clip lines to the following, non-rectangular window? If so, note any additional difficulties in implementation (as compared to implementing it for axis aligned, rectangular windows), and if not, justify your answer.



62. Can the Liang-Barsky algorithm be used to clip lines to the following, non-rectangular window? If so, note any additional difficulties in implementation (as compared to implementing it for axis aligned, rectangular windows), and if not, justify your answer.



63. What is the difference between Gouraud shading and Phong shading?

64. Demonstrate, using algebra or a picture, that order matters when composing transformations.

65. Suppose we have the standard, orthonormal World Frame $F_w = \{\hat{w}_1, \hat{w}_2, \hat{w}_3, O_w\}$ and a Viewing Frame $F_v = \{\hat{v}_1, \hat{v}_2, \hat{v}_3, O_v\}$ where

$$\begin{aligned} v_1 &= w_2 \\ v_2 &= -w_3 \\ v_3 &= w_1 \\ O_v &= O_w + w_1 \end{aligned}$$

Give the change of basis matrix to take coordinates relative to F_v and returns coordinates relative to F_w.

66. In volume rendering, you can either render from "front to back" or from "back to front." What are the advantages each method?

67. A perspective transformation transform everything that's behind the viewer (in the unprojected space) to be in front of the viewer. Potentially, this could result in objects that are behind the viewer erroneously appearing on the screen. How do avoid this problem?

68. After multipling a 3D point whose matrix is $[x, y, z, 1]^t$ by the perspective transformation matrix, we get the matrix $[A, B, C, D]^t$. What remains to complete the perspective transformation? Be specific (i.e., show the final result in terms of A, B, C, and D).

69. When performing a lighting calculation, we attenuate the intensity of the light based on the distance from the light to the point on a surface at which we are performing the lighting calculation. However, we do not attenuate the intensity to account for the distance from the viewer to the point on the surface.

    Why do we attenuate the intensity in the former case but not in the latter case?

70. Shadow maps are one way of achieving shadows on an OpenGL renderer. Describe the shadow map method and what hardware support/features are needed to use it.

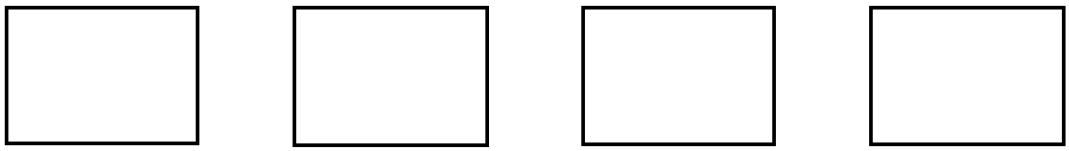
71. Name (or describe or otherwise illustrate) the three real-time shadowing algorithms described in class. Circle the one that could most easily be adapted for use in a ray tracer.

72. Give one specific example of how Pixar animators made use of the traditional Disney animation principles to create more engaging characters in the animation *Luxo Jr.*.

73. Explain in one or two sentences why the CMYK colour space is referred to as a "subtractive" colour space.

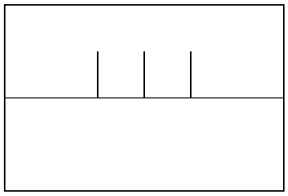
74. The Sutherland-Hodgman clipping algorithm can be used to clip an arbitrary polygon $\mathcal{P}$ to a convex window $\mathcal{W}$. Below are four copies of the same rectangular window $\mathcal{W}$. Draw a quadrilateral (four-sided polygon) P for each window so

    that the resulting clipped polygons have 3, 4, 5, and 6 sides.

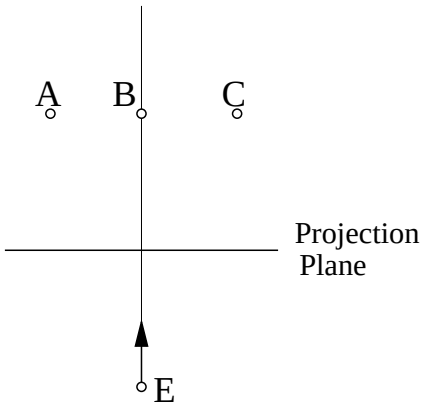


3 SIDES 4 SIDES 5 SIDES 6 SIDES

75. Suppose you have three line segments that are parallel to the projection plane and are aligned in 3D with the up axis of the screen, and whose perspective projections onto the screen are shown in the following diagram:



Now suppose we look at an orthographic projection of these line segments from an overhead view, and that they appear as in the diagram below:



We have labeled the line segments A, B, and C. Which of the following relationships of the heights of these unprojected lines segments is true (where A_h, B_h, C_h are the heights of segments A,B,C respectively) :

(a) $A_h = C_h = B_h$

(b) $A_h = C_h > B_h$

(c) $A_h = C_h < B_h$

(d) None of the above

20 Definitions

Briefly define the following terms as they apply to computer graphics and the material taught in CS 488/688. Where appropriate, make reference to other terms in this list that are related to, similar to, or the opposite of the term in question.

1. affine combination
2. affine space
3. aliasing
4. alpha blending
5. ambient reflection
6. anti-aliasing
7. B-spline
8. backfacing polygons
9. barrel distortion
10. barycentric coordinates
11. baud rate
12. beam settling delay
13. beam penetration crt
14. Bézier
15. bitblt
16. Bresenham
17. BRM
18. bump mapping
19. button
20. callback routine
21. calligraphic display
22. cap polygon
23. cathode
24. chromaticity diagram
25. CIE
26. collector
27. collimator
28. color map
29. compensation table
30. concave object
31. convex combination
32. convex object
33. crossbar switch
34. cross product
35. DAC
36. DDA
37. decay
38. deflection coil
39. delta gun
40. diffuse reflection
41. distributed ray tracing
42. dithering
43. DMA
44. doping
45. dot product
46. dragging
47. DVST
48. ECL

49. electromagnetic
50. electrostatic
51. error diffusion
52. field (even/odd)
53. flicker
54. flood gun
55. florescence
56. form factor
57. fragment shader
58. frame
59. frame buffer
60. fusion point
61. function box
62. gamma correction
63. GKS
64. Gouraud shading
65. gravity field
66. GSPC
67. hemi-cube
68. Hertz
69. HLS
70. homogeneous coordinates
71. horizontal retrace
72. HSV
73. hysteresis
74. image space algorithm
75. inking
76. ink jet
77. in-line gun
78. interlace
79. interpolating spline
80. jaggies
81. joystick
82. just noticeable difference
83. kerning
84. keyframe animiation
85. keypad
86. kinematics, forward and inverse
87. left-handed coordinate system
88. ligature
89. light handle
90. light valve
91. lightpen
92. linear space
93. liquid crystal light valve
94. local control
95. locator
96. look-up table (LUT)
97. Mach bands
98. master object instancing
99. menu
100. modelling transformation
101. motion capture
102. mouse
103. normalized device coordinates
104. NTSC

105. Nyquist
106. Nyquist frequency
107. off-axis correction
108. on-axis correction
109. opponent colors
110. ordered dithering
111. orthographic projection
112. outcodes
113. PAL
114. PDI protocol
115. partical systems
116. persistence of phosphor
117. persistence of vision
118. perspective space
119. PHIGS
120. Phong shading
121. phosphor
122. phosphorescence
123. physical input device
124. physical output device
125. pick
126. pick id
127. pincushion distortion
128. pixel
129. plasma panel
130. predictive lightpen tracking
131. pseudo-color
132. rack
133. radiosity
134. random dithering
135. raster display
136. RasterOP
137. ray tracing
138. read/modify/write cycle
139. retrace
140. RGB
141. right-handed coordinate system
142. rubber banding
143. run length encoding
144. scanline coherence
145. Schlieren optics
146. Schmidt optics
147. SECAM
148. segment
149. segmented display list
150. seven plus/minus two
151. shader
152. shadow mask
153. shadow mapping
154. shadow volume (or shadow polygon)
155. Sketchpad
156. Solid Texture
157. specular reflection
158. SRL output
159. staircasing
160. storage tube

161. storage grid
162. stroke
163. structured display list
164. subjective spot size
165. surface of rotation
166. tablet
167. texture mapping
168. tracker
169. tracking ball
170. transformed display list
171. turntable
172. UIMS
173. valuator
174. vertical retrace
175. vertex shader
176. video controller
177. viewing frustum
178. viewing transformation
179. viewport
180. viewporting transformation
181. virtual input device
182. virtual output device
183. volume matrix
184. wheel of reincarnation
185. widget
186. window
187. window edge coordinates (WEC)
188. window coordinates
189. windowing transformation
190. world coordinates
191. write mask
192. write through
193. z-axis modulation
194. z-buffer
195. z-depth
196. zoom/pan/scroll