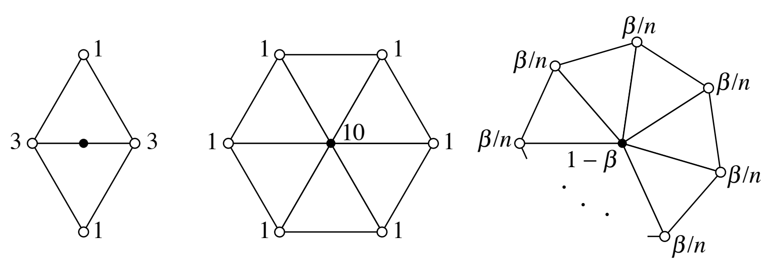
CS 417 Homework 5

out: Thursday, March 27, 2003
due: Friday, April 4, 2003

Problem 1: Subdivision

A subdivision surface is defined using a sequence of refinements, or subdivision steps, applied to a polyhedron called the *base mesh*. Each of these steps takes a *coarse* mesh and introduces new vertices that subdivide the faces and edges of that mesh to produce a *fine* mesh.

In lecture we discussed how to compute the vertex positions for the fine mesh in terms of masks that apply directly to the coarse mesh. For instance, Loop's subdivision rules for triangle meshes position new vertices according to these masks:



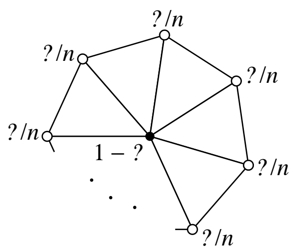
where

$$\beta = \frac{5}{8} - \frac{(3 + 2\cos(2\pi/n))^2}{64}.$$

This approach requires two masks, one for the odd vertices (the ones that are added in the middle of an edge in the coarse mesh) and one for the even vertices (the ones that already existed in the coarse mesh). In this drawing, note that the middle mask is a special case of the right-hand mask; it is included just to concretely illustrate the common case.

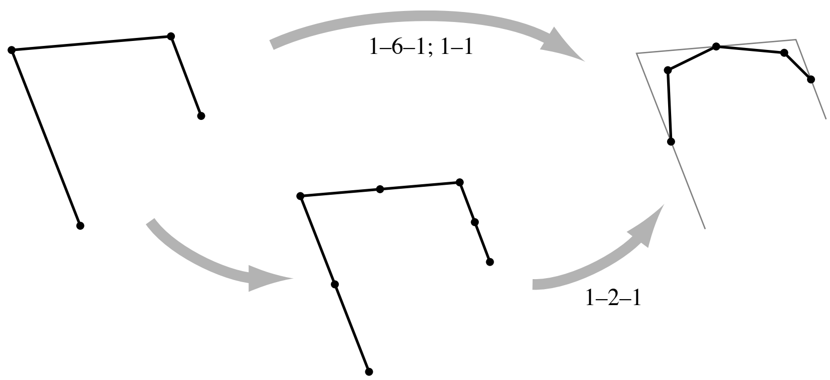
Another way to implement the subdivision process is in two steps: subdivision with the new vertices positioned at the midpoints of the coarse edges (this does not change the geometry) followed by a smoothing step in which all the vertices are repositioned according to a mask that applies to the fine mesh. This amounts to subdividing with these masks:

and then smoothing using a single mask:



The smoothing process illustrated by this mask computes a new position for every vertex in the fine mesh as a linear combination of the positions of neighboring vertices.[1]

To provide a 1D example: the masks that implement the B-spline as a subdivision curve are 1–6–1 for even vertices and 1–1 for odd vertices when the positions are computed directly from the coarse mesh (in this case "mesh" means polygon). The alternate formulation is midpoint subdivision followed by smoothing with the mask 1–2–1:



You should work out how the final smoothed positions depend on the original coarse positions in this case and convince yourself that the result is the same either way.

1. Give a smoothing mask that, in conjunction with midpoint subdivision, implements Loop's subdivision rules for regular areas of the mesh (that is, you can assume that all relevant vertices will have valence 6).

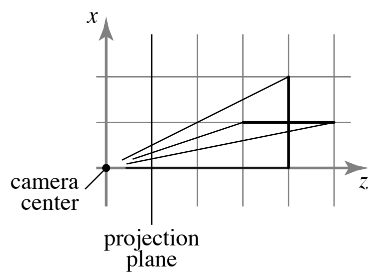
2. Give a smoothing mask that, in conjunction with midpoint subdivision, implements Loop's subdivision rules for any mesh. The weights will depend on the valence.

---

[1]You should think of the smoothing process as writing to a second copy of the vertex positions—only the unsmoothed vertex positions are used to compute the smoothed positions.

Problem 2: The z buffer

Consider a scene made up of two intersecting squares being viewed by a perspective camera. The camera is at the origin pointing along the $+z$ axis with $+y$ pointing up. The perspective projection plane is $z = 1$. The extents of the two rectangles are $(0, -1, 4)$–$(2, 1, 4)$ and $(1, -1, 3)$–$(1, 1, 5)$. Each of the rectangles is drawn using two triangles. Here is an overhead view of this setup:



and the squares both extend from -1 to 1 in y.

1. Draw the image that results from rendering this scene with a z buffer. Include two drawings, one under the assumption that the z value at each pixel is correct and the other under the assumption that the z value at each pixel is generated by interpolating linearly across the image between the (correct) z values of the vertices. If it matters which way the constant-x square is split into triangles, draw both cases. (It does not matter which way the square that is parallel to the projection plane is split.) It is a good idea to use graph paper for this.

Now replace the constant-x square with a diamond inscribed in the square, with its vertices at the midpoints of the edges of the square. (That is, the new shape is a square at a 45° angle, with its vertices at $(1, 0, 3)$, $(1, 0, 5)$, $(1, -1, 4)$, and $(1, 1, 4)$).

2. Repeat part 1 for this new configuration.

Your drawings should be accurate enough to ensure that each vertex is occluded or unoccluded as it should be, and so that it's clear which square shows up at every point in the image. Annotate the drawings briefly to indicate the differences between the cases being compared.

Problem 3: Lighting

Consider a unit sphere that is viewed orthographically from the +z direction (with +y up on the camera) and illuminated by directional lighting with unit intensity. The sphere fills up 3/4 of the image width. For each of the following material parameters, sketch a plot of the intensity along the center row of the image (the one that goes through the center of the sphere) for illumination coming from the directions $(1, 0, 1)$ and $(1, 0, -1)$ (on two separate plots). Note any discontinuities in the value or derivative of this function.

1. $k_D = 1.0, k_S = 0, n = 1$.
2. $k_D = 0.0, k_S = 1, n = 1$.
3. $k_D = 0.0, k_S = 1, n = 2$.
4. $k_D = 0.0, k_S = 1, n = 100$.

Your plots should show the values at maxima and place the maxima in approximately the right position.

Problem 4: Texture mapping

1. Consider a parametric surface defined by the following mapping from the unit square $[0,1] \times [0,1]$ to 3D space:
$$(s,t) \mapsto \begin{bmatrix} (r_2 + r_1 \cos 2\pi s) \cos 2\pi t \\ (r_1 \sin 2\pi s) \\ -(r_2 + r_1 \cos 2\pi s) \sin 2\pi t \end{bmatrix}.$$
What is the shape of the surface defined by this embedding?

2. In order to texture map this surface we define texture coordinates using the projection methods described in lecture. For each of the following mappings, draw a copy of the unit square and mark the places where the mapping from the surface to the texture image has a zero derivative (to be precise, that is when the 2-by-2 derivative matrix is singlular):
   (a) Projection onto the x–y plane.
   (b) Projection onto the x–z plane.
   (c) Projection from the y axis onto a cylinder.
   (d) Projection from the origin onto a sphere.
   (e) Projection from the origin onto a cube.

3. Consider a camera with 90° by 90° field of view positioned at the origin with its projection plane at $z = -1$ and its view direction parallel to the z axis. An infinite plane is positioned at $y = -1$ and texture mapped with texture coordinates $(u, v) = (x, -z)$ (think of the checkerboard examples shown in lecture). The texture coordinates wrap around to tile the plane with copies of a texture image that is defined on the unit square ($[0,1] \times [0,1]$). Write down the 2-by-2 derivative (or Jacobian) matrix of the mapping from texture space (u, v) to image space (x, y).

4. Write down the derivative matrix of the mapping from image space to texture space, as a function of u and v.

5. Assume the camera's image is 200x200 pixels and the texture map is 100x100 pixels. According to the approximation provided by part 3, and using a unit-area circle to define the region belonging to a pixel:
   (a) Along the v axis, at what values of v is the area of the elliptical footprint of a pixel equal to 2, 3, and 4 times the area of a texel? (*Hint:* the answer does not depend on the shape of the pixel's region, only its area.)
   (b) Along the v axis, at what values of v is the long axis of the pixel footprint equal to 2, 3, and 4 times the width of a texel?

6. As we move off the v axis, do the (a) area and (b) long axis of the pixel footprint get larger, smaller, or stay the same?