Computer Graphics
Exercise 1 - Surface representation

Handout date: 30.09.2011

Submission deadline: 14.10.2011, 23:00h

This is an individual exercise. You may submit your solution by either:

- Emailing a PDF (scanned or typeset) of your solution to introcg@inf.ethz.ch with the subject and the filename as cg-ex1-firstname_familyname, or
- Handing in a paper copy at the beginning of Friday's exercise session.

Implicit and parametric surface representations

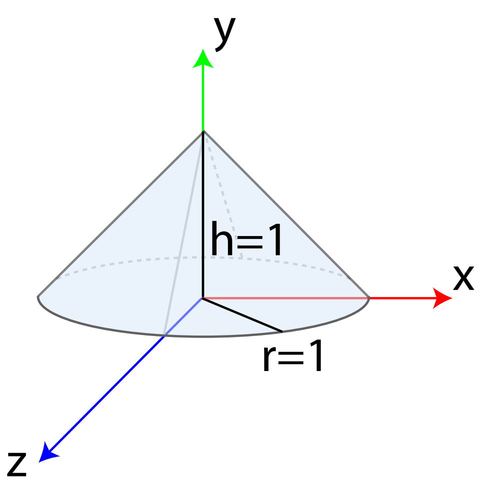

Figure 1: A cone with its *apex* (tip) at point $(0, 0, 1)$ and its *base* (circular bottom) at the unit disk on the xz-plane centered at the origin.

(a) Explain the difference between an *implicit* and *parametric* representation of a surface.

(b) Give a *parametric* representation of the points on the surface of the cone in Fig. 1.

(c) Give an *implicit* representation of the points on the surface of the cone in Fig. 1.

Modeling with implicit representations

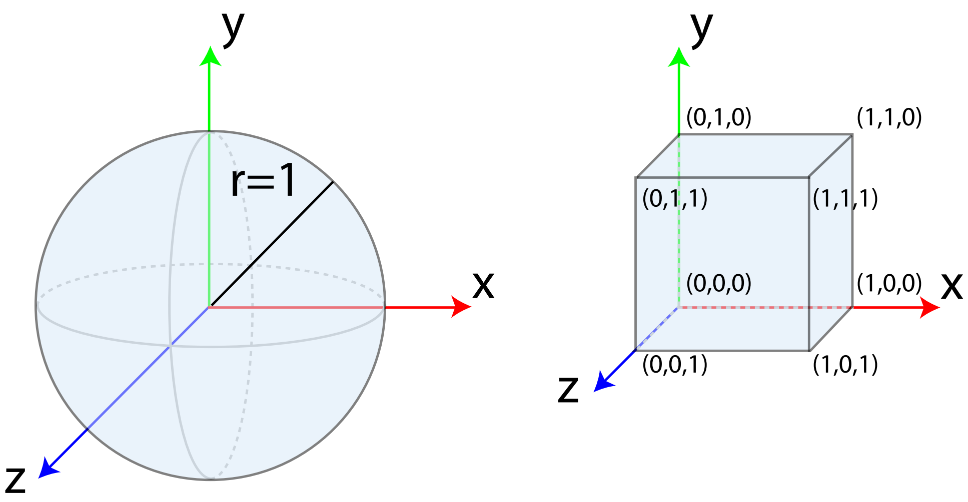

Figure 2: Left: the unit sphere, a sphere with radius 1 centered at the origin. Right: the unit cube, a cube with all sides of length 1 and corners at: $(0,0,0)$, $(0,0,1)$, $(0,1,0)$, $(1,0,0)$, $(0,1,1)$, $(1,1,0)$, $(1,0,1)$, and $(1,1,1)$.

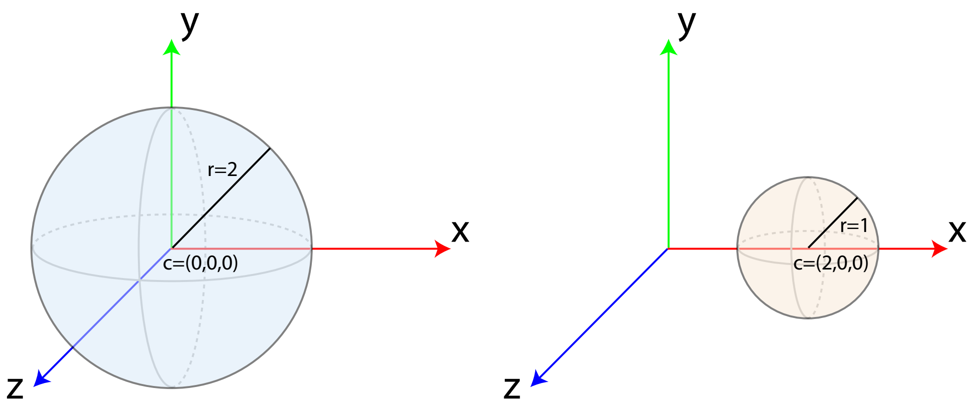

Figure 3: Left: a sphere with radius 2 centered at the origin. Right: a sphere with radius 1 centered at $(2,0,0)$.

(a) Give an *implicit* representation of the points on the surface of the union of the points *inside* the unit sphere and those *inside* the unit cube (see Fig. 2).

(b) Give an *implicit* representation of the points on the surface of the union of the points *on* the unit sphere and those *on* the unit cube.

(c) Given the implicit function representation of a sphere with radius r centered at $\mathbf{c} = (c_x, c_y, c_z)$ as $r^2 = (x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2$, what happens if you *sum* the implicit functions representing the two spheres in Fig. 3?

(f) What happens if you instead sum blob representations of these spheres?

Mesh data structures

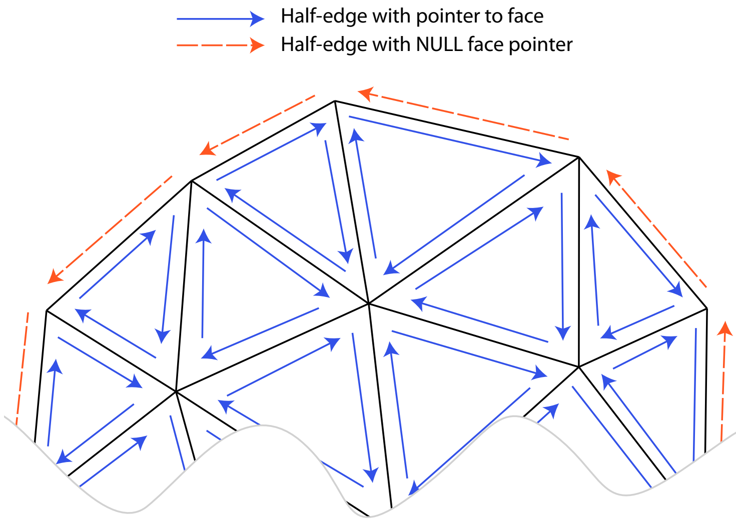

Figure 4: Normal half-edges are shown in blue. These half-edges have valid pointers to their corresponding face. Boundary half-edges are shown in red. These half-edges have NULL face pointers.

(a) Given a vertex list and a face index list representation of a mesh, give an $O(\#\text{faces})$ algorithm for determining the neighboring faces of a given face, f.

(b) Using a *half-edge data structure* representation of a mesh, give an $O(\max(\text{number face size}))$ algorithm for enumerating the neighboring faces of a given face, f.

(c) Given a vertex list and a face index list representation of a manifold mesh, give a $O((\#\text{faces})^2)$ algorithm for walking along the boundaries of the mesh. Where walking along the boundary of the mesh means enumerating the vertices along *each* boundary of the mesh in order.

(d) Using a *half-edge data structure* representation of a manifold mesh, give a $O(\#\text{edges})$ algorithm for walking along the boundaries of the mesh. You may assume that boundary half-edges have a twin half-edge whose face pointer is NULL (see Fig. 4).

(e) For a manifold surface mesh with fixed topology, prove that $O(\#\text{faces}) = O(\#\text{edges})$.

(f) Prove the Euler formula for meshes with sphere topology.

(g) Prove the Euler formula for meshes with disk topology.