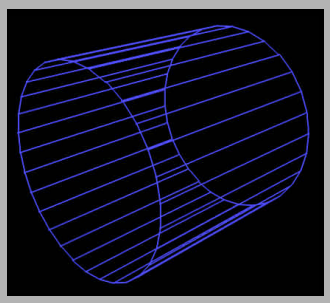
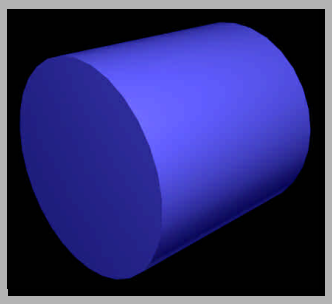
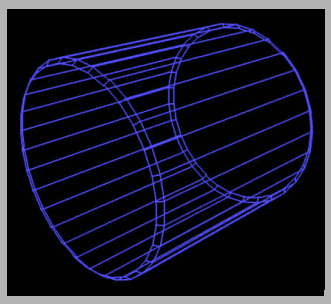
Figure 1: Basic Column

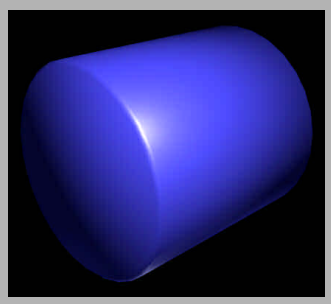




Perfect,but perfectly impossible.

Figure 2: Beveled Column





An improvement over the basic column,but rendering problems emerge.

The Bevel

An introduction to a fundamental tool of modelling realism.

by Dave Hill
www.aarrgghh.com
dave@aarrgghh.com

Most real-world edges consist not of the abrupt meeting of two or more surfaces but of a small often unnoticed plane or area called a bevel that exists between the surfaces — unnoticed except for the subtle lighting effects it imparts to the objects on which it appears. Bevels occur in the real world because of imperfections in human manufacturing processes and continual exposure to natural erosion processes and ordinary usage. Most people probably could not explain the significance of bevels to realistic 3d imagery, but all will notice the differences when shown the same object with and without bevels. As a modeller, you'll learn to add some kind of bevel to all the edges of your objects, especially on objects that you can afford the extra polygons.

Figure 1 shows a simple column — a ring of rectangular polygons (the shaft) capped by two multi-sided circular polygons (the endcaps). Most people wouldn't have any objections to such a figure and it renders smoothly and without errors. But the edges, while matematically perfect, are unrealistically sharp. That is the edge of a knife, not the corner of a column.

We can give the edge some realism by inserting an extra ring of polygons between the shaft and the endcaps. Figure 2 shows our new beveled column. The extra polygons were added by extruding the endcaps a very short distance and downscaling the endcaps the same distance (about five percent the diameter of the endcap) in order to produce an angled edge.

At first glance the new column is strikingly more realistic than the simple column we started with. The bevel, by introducing a new angle to the shape, catches ordinarily subtle specular highlights that the old column lacks. (I say ordinarily subtle because our life-long familiarity with these particular details has made them invisible; in the context of this simple demonstration, these details now have the effect of a sledgehammer.)

But upon closer inspection, we can nonetheless pick out a number of rendering errors that keep our column from being quite ready for prime-time. Notice that the endcap has acquired a sweeping shadow which gives it the appearance of being domed when it is actually flat. The lighting fades along the diameter of the encap, and abruptly so, as it does along the length of the shaft. The specular highlight on the bevel spills onto the shaft. Lastly, the shadow and highlight created by the underlighting along the bottom of the shaft is interrupted for no apparent reason. Compare the lighting to the origimal beveless column and it just "feels" wrong, even when we're not certain about the true source of the lighting.

The rendering errors occur because of the way most rendering packages or modules attempt to distribute lighting effects across polygons. Most renderers

attempt to ramp the effects across each individual polygon, or set of polygons that are coplanar (lying within the same plane), which results in a gradient unless the polygon or polygons are facing either directly toward or directly away from the light source, in which case would result in a single flat shade. The effects become of course more complex as you introduce additional lights to the scene, but the rules for lighting each polygon remain the same.

In most cases, this technique is acceptable because most polygons in any given region of a well-formed mesh are roughly similar in size. However, where polygons of radically differing proportions meet at an angle, lighting errors emerge. This is more often a problem with low-resolution meshes, where an economy of polygon count is a priority.

What we need to do is help the renderer properly illuminate the bevel by introducing a narrow "buffer zone" around it. By introducing a ring of short polygons on one side of the bevel, coplanar to the endcap, and another ring on the other side, coplanar to the shaft, we can restrict the renderer's attempts to ramp the lighting to the area contained within the buffer zone, where the angles of the adjacent polygons change.

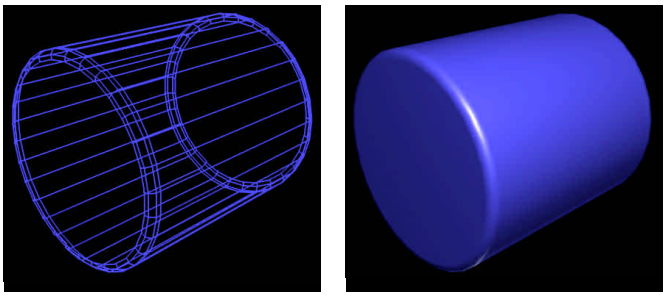
Figure 3 shows the effects of adding a ring of coplanar polygons to the shaft of the column. Note that the rendering is identical to that of our original column in Figure 1. This is because, as far as the renderer is concerned, it makes no difference whether the length of the shaft consists of one, two or ten polygons (or even whether these polygons are equallly or unequallly proportioned), as long as they remain coplanar.

Figure 4 shows the effects of subsequently adding the bevel. Our column is developing nicely, but the endcap, with its sweeping shadow, still needs its remedy.

Finally, Figure 5 shows the effects of adding a ring of polygons coplanar to the endcaps. Welcome to the real world!

Postscript for Poser users: Because Poser's renderer was specifically optimized to render the organic forms of humans and animals, it will attempt to smooth the Dickens out of almost any edge, making it possible to get your bevel without actually creating one! All that's necessary is to create the coplanar polygons that would be placed on either side of the bevel. While most renderers would render a sharp edge as shown in Figure 1, Poser will use the extra polygons to calculate a smooth edge. In effect, the coplanar polygons have become the bevel. So those of you with an eye for thrift may be able to save yourselves a few polygons by dispensing with the bevel itself.

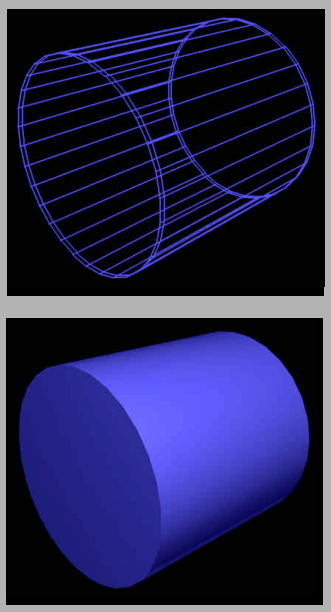
Figure 5: Final Buffer



The finished,properly rendered beveled column.

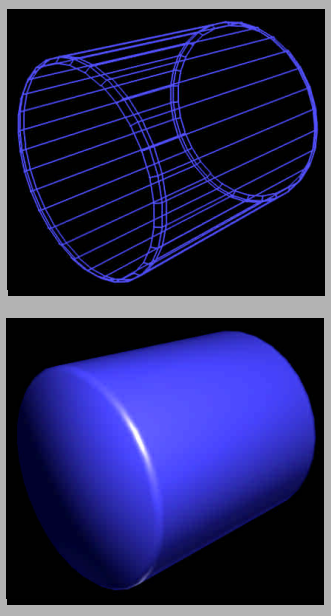
– Here endeth the lesson –

Figure 3: Buffer Facets



The added facets will keep the lengthwise polygons rendered properly.

Figure 4: Added Bevel



We've fixed it lengthwise, but the cap still looks domed.