Lars Hesselgren explores “blobby architecture” in MicroStation V8

By Lars Hesselgren

At four a.m. on a wintry day in 1999, I found myself nodding off while driving on the Pennsylvania Turnpike in a snowstorm. I had just crossed the Atlantic from London, driving from New York’s JFK airport to Bentley headquarters. In the morning, I had a meeting scheduled with several members of the Bentley development staff to review recent accomplishments in solid modeling for architecture and to talk about what should come next.

This meeting was mainly a brainstorming session—what did I as a user want? Solid modeling was a fait accompli and something I had championed for since using Intergraph’s EMS ten years earlier.

Looking at other software, what was inching up the agenda were surfaces. At the time, MicroStation had a pretty decent B-spline surface library, largely underused by the user community. Somehow, the assumption was that you did surfaces in 3D Studio MAX or Catia or Form•Z or Maya, or even Rhino. Too many assumed MicroStation wasn’t in the same league.

At a Bentley user conference the year before, I had met Paul Richens, director of the Martin Centre for Architectural and Urban Studies at Cambridge University and an expert in computer graphics. We talked about Form•Z. He pointed out that the great trick of most surface modeling packages was that they were polygon-based. In MicroStation, a different philosophy meant that, while the programmers were perfectly aware of polygons, surfaces were mathematically precise and were polygonalized at render time before being thrown to the rendering engine.

When you consider MicroStation’s use for construction and architectural design, you realize something strange is going on. We often deal with large geometries, defining objects hundreds of meters tall (KPF International’s design for the Shanghai World Financial Center is 450 meters tall). At that scale, curving individual components, such as a window or sheet of glass, is madness. What we do as architects is to stroke the curves to straight elements and surfaces to planar patches.
There is a parallel in such shortcutting with how polygon-based software works. But it’s all subtle. For instance, rendering systems (every one of them, as far as I know) depend on the simple trick of ‘averaging normals’ to create smooth-looking surfaces. There was a time—10 years ago—when not every software product could smooth polygonal surfaces. It was quite entertaining to spot the errors. It ranged from bad ModelView or Studio MAX renderings to most 3D games to AutoCAD’s display list system (generating many ‘regens’).

So the notion I tried to sell to Bentley was to think ‘dual-mode’—a surface or a curve should be displayable either as smooth curve/surface or as a stroked element. There was a lot more to it and we batted it back and forth.

Then I went to KPF’s New York office and was told that Form•Z was “just so superior” to MicroStation. I got their best user to show me why. Between us, we could only find one significant feature that really was better in Form•Z, unfold. Because the system is polygon-based, half the work is already done.

Remember, this was 1999.

My good friend Hugh Whitehead, who was just about to join Foster and Partners, had showed me some cool unfolding in the late 1980’s, and he had developed a MicroStation application in the meanwhile (with Charlie Tonkiss, current IT Director of Whitby Bird & Partners). So there were a lot of precedents about. I tackled Bentley’s Earlin Lutz with this, and he wrote a neat prototype, currently called Facet. It is available in the current MicroStation V8 build, but only as a key-in. To load the application, key-in mdl I facet. Then to get the dialog, key-in facet dialog.

Today, there is another button under 3D utility, ‘Construct facet,’ which strokes any surface or solid to facets.

**A bunch of triangles**

There is yet another strand to this whole saga. DTM’s, Digital Terrain Models, have been the mainstay of civil engineering modeling packages for decades. I call it a relatively trivial format, essentially just a pile of triangles. A whole design mystique and a host of applications have grown up around DTM, including Bentley’s GEOPAK Site. A modeling product like Form•Z handles DTM’s trivially—after all, they are just what everything is made of in Form•Z! Ditto for 3D Studio MAX, which only gives you triangles—it seems to have no other surface geometry!

There has been a nice Java application floating about for generating DTM’s from contours inside MicroStation/J. I mentioned this to Lutz not long ago, and he asked me if I had tried the key-in facet triangulate contours. I did—what a treat! It is extremely fast and completely inside MicroStation V8; you just have to know the secret key-in. Since now you do, try it. Also, please note that you will have to select the contours before the key-in (and, of course, they must be at their true height in space).
Making blobby architecture

So now we have the history, what can you use this stuff for? As somebody said on the new Bentley newsgroup not long ago, the new architectural style we are all getting into is ‘blobby architecture.’ Let’s be more polite and call it the double-curved surface architecture syndrome (DoCuSAS). DoCuSAS is the kind of movement that promotes the attitude, ‘if you can do it, you must do it.’ It’s also liberating to get away from the perpendicular and almost as sexy as being a car designer. So we’ll do it.

Let’s take a few recent examples I have worked on and see how some of these tools come in handy.

Let’s first look at the simplest possible application, unfolding a sphere. The sphere has a 10m radius, but we’ll want to have relatively few faces, so I have set maximum number edges to 8. Under options, there are two outputs, the new Mesh Element and shapes. Mesh elements don’t unfold, so we’ll use shapes. See Figure 1.

Mesh elements are quite interesting, they are a new type of element where the vertices and edges are stored once only, even when shared between faces, so they are very compact and fast. If you want individual shapes with a mesh, just drop solids to surfaces.

The facet dialog is loaded by default, but there is no button (yet), so you have to know the key-in facet dialog. If you do, you get the dialog and choose tab ‘Manipulate.’ Select all the shapes, press the unfold button and give a datapoint in a top view (the unfold always happens in the x,y plane). There is no self-avoidance built in to the unfold yet (I’m told it is in the works), and the way it unfolds is interesting. But if you take the orange snake (Figure 2), plot it and fold it, you’ll get a sphere.

Because you can use Smart Select to select rows, you can control the unfolded pattern much better if you wish, see Figure 3. There is, of course, an infinite number of ways you can unfold the set of polygons.

Let’s get back to rendering briefly. If we press the raytrace button we get a throwback—the faceted sphere on the right in Figure 4 looks like a really bad rendering from 20 years ago. But MicroStation has a ‘Smooth Facet’ button. If we smooth the facets to a high angle tolerance (180), you will find the facets are indeed smoothed (Figure 5). But the real giveaway that we are still dealing with a faceted object is the silhouette—it is still faceted.

Real architecture

Now let’s deal with a more complex surface. This one is from a competition we did recently; it’s a skyscraper somewhere in Europe. The atrium enclosure was made of intersecting surfaces, which needed to be unfolded for the model makers, as shown in Figure 6. Reading from the bottom right is the original solid, then the faceted solid, then the actual surfaces I wanted, then the individual surfaces. Lying on the ground are the unfolded surfaces. We also, of
course, used the CAD model for a rendered image (Figure 7).

In another job, also in Europe, we recently proposed a toroidal roof over an office building. Remember that any arc rotated around the axis perpendicular to the plane of the arc is a toroid. It can be self-intersecting, and the sphere is an extreme example of a self-intersecting toroid. (Figure 8). Of course all toroids can be faceted with square facets easily.

The roof we designed is almost spherical—the two radii differ by only a few meters. Even though we could have tiled with squares (strictly speaking, planar quadrilaterals), we choose at this stage to make it out of triangles. Building the model was pretty trivial—all you needed was a stroked arc to the required number. It’s worth remembering that the ‘Drop B-spline Curve’ tool in MicroStation drops any curve, not just B-splines, to really useful things, like equal chord lengths—a real lifesaver when setting out buildings. After that, there is a handy Facet command under the Coons tab of the Facet dialog. To repeat the tiling, simply use the polar array command (once you have rotated yourself looking straight down the rotation axis).

After that, the real joy is chopping the surface into shapes using Fence from Element (the edge of the dome follows the slab edges, below which have curved corners). Ordinary Fence Clip keeps the shapes intact, and it is easy to just create an edge beam—simply Copy Parallel of trimming shape and another Clip Fence.

The down stand beams are equally trivial; select the triangle shapes and extrude them to surfaces, leaving the profiles behind. The profiles are glass, the down stands ordinary white. Figure 10 shows the interior view.

It took about an hour or so to build the roof. But it looks like magic to those who don’t know the MicroStation tools. We also rendered all the elevations from the model as in Figure 11, but we actually overlaid them with vector hidden lines just to emphasize the edge definitions.

There are a great number of additional tools in facet dialog, which I leave to you to explore on your own. Some of them are based on parabolic and hyperbolic geometries of a very advanced kind, and I look forward to your buildings designed with them.

Undoubtedly we are in the era of smoothly curved buildings—blobby architecture, if you will. Now we finally have the tools to generate really interesting shapes. I know I’m not alone when I say, thank you Bentley!

Lars Hesselgren is an associate and Director of Information Technology at Kohn Pederson Fox International in London. Images from this article, as well as other radiosity studies, are available at MSM Online, and at the TMC UK Web site, http://www.bentleyuser.org/. You may write to Lars with questions or comments at lhesselgren@kpf.co.uk.

Bentley MMS Online, 2002 Április
http://www.msmonline.com/
© 2002 by Bentley Systems, Incorporated. All rights reserved.