ModelBehavior

By Kenneth A. Crawford, PE

Ken Crawford is a Registered Professional Engineer in Pennsylvania and a U.S. Army Corps of Engineers Quality Construction Manager (QCM). Ken's experience includes 25 years in civil engineering, surveying, and construction, with significant background in GIS applications. He is a principal in the firm Harken-Reidar, Inc.

The How-To Guide to Successful Surface Modeling

I'd like you to take a moment to think about what it is we do in this (not so very) new digital world as it relates to our traditional work products, specifically making maps. I realize many of you may perform boundary surveys, write legal descriptions, and engage in the noble practice of property ownership and survey retracement, but for those of us also involved in the practice of measuring and documenting the physical world as it exists, this series of articles will address some techniques in preparing a *digital surface model* and sharing it with our clients, colleagues, and other project stakeholders. It is no longer enough to simply deliver a paper map showing contours and spot elevations. Today's practice requires leveraging the work of all those involved along the way to gain greater productivity throughout the design process and to retain as much fidelity of the original basis of work as possible. And the modern surveyor has arguably the most vital role in this process: preparing an accurate model of the existing site conditions as required for the work at hand. If the starting stages are invalid, the rest of the project will be fundamentally flawed and often won't show up until construction. By then, mistakes mean delays, money, and damage to reputations.

I'll be showing you the steps involved in creating an accurate surface model and we'll discover techniques to verify the work of others prior to incorporating it into our own work as it moves forward in the process. We'll learn some basic guidelines that we can apply both in the field and in the office to fine-tune the process and guarantee that we have an accurate model that we can stand behind, even in its digital form.

And no more "hiding behind the paper!" Very often I hear from contractors how they can't get the digital design files from the engineers and surveyors who prepared it, only the final paper drawings. Reason number one cited: "We can't give you the files for liability reasons." What this says to me is that the preparer is not confident of the design and is afraid to let anyone else work from the same model she used in preparing the paper drawing. So you'll be learning how to analyze your own work and be able to stand behind it with confidence. The day is coming when all deliverables will be requested in digital form as full three-dimensional models, as soon as the lawyers work out the details on digitally certifying our work. We'll leave that battle for another day.

So let's begin.

Rule #1: If it is an edge on the ground, then we need to make it an edge in our modeled data set

The Basics

As with any process, there are a certain set of basic concepts that we must be aware of that govern surface modeling. Assuming that we will be preparing a model through the use of *triangulation*, probably the most basic concept to grasp is that every point in our selected data that we wish to include in our model–those whose elevations will indicate a point on the surface being modeled–will form the apex of a triangular shape. The triangle is the most basic geometric figure used to form surfaces because it requires a minimum of three points to form a *plane* in space. It is by combining the resulting triangles that represent individual "planes" or facets of any given surface, edge-to-edge, that we arrive at a collective representation of the entire site.

The resulting network of triangles will by necessity be irregular in shape, hence the term "Triangulated Irregular Network," or TIN. This is the basic file type for triangulated surfaces, and you will hear this term used almost interchangeably for the general concept of a modeled surface. Some software applications name the resulting surface files in this form with TIN as an extension, but a TIN file from, say Land Development Desktop, will not be compatible with a TIN file from Carlson Survey. They simply share a naming convention.

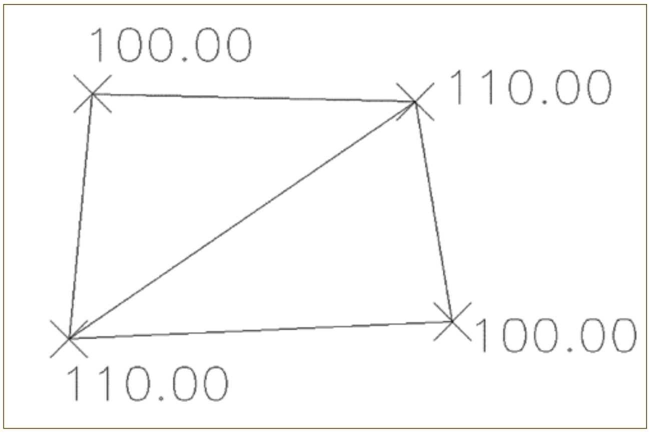

Figure 1 Points as apexes of triangles

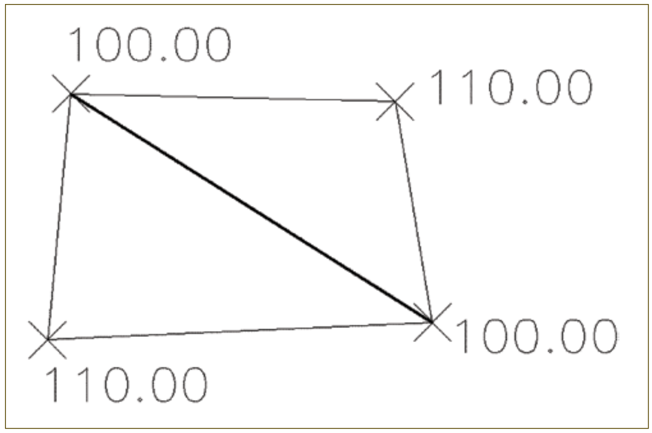

Figure 2 Breakline added to force triangulation in opposite direction

There are other ways, however, and we'll be exploring those as well.

Every triangle edge that forms is understood to be a line of linear change in elevation between the two end points, and therefore if two points are not lying along a straight-line change in grade, they must be separated by a breakline or by sufficient numbers of additional points to indicate the lower or higher points in between.

The next basic rule to remember is that every point can "see" every other point around it (and possibly form a triangle edge) unless and until there is something in between the point and another point to act as a barrier. This is the basic notion of a *breakline*. I presume that we all are fairly familiar with these two concepts at this stage in the game, but just to clarify and to help you gain a better understanding of what exactly "breaklines" are and how and when to apply them in and around your point data, let's examine their functionality a bit.

In **Figure 1**, once the triangle lines have been formed, they act as barriers to a point's "visibility." For example, the point in the northwest corner with elevation of 100 can no longer "see" the point in the southeast corner since there is a barrier in the way (the line formed between the southwest corner and the northeast corner). This is, in effect, a breakline. So triangle edges–and in fact any linear feature that interrupts the "visibility" of one point in relation to another–is a breakline. As in the example above, these breaklines can be the result of the computer forming the triangle edges, or they can be created before the triangulation process by the user.

Proper data structuring of the above sequence of points would dictate that if we intended the feature to be modeled to be that of a ridgeline running from the southwest to the northeast, we would need to create a line along the ridge top (southwest to northeast) as in Figure 1. If on the other hand we had intended the feature to be modeled as a valley running from the northwest to the southeast, we should have created a breakline before processing connecting the two points having elevation of 100 (northwest to southeast). This would have prevented the triangles from forming as shown in Figure 1 and would result in that of **Figure 2.** This prevents the 110 elevation from "seeing" the opposite 110 elevation, preserving the valley floor between the two points with elevation 100.

If we fail to indicate our intent for the modeling (*i.e.*, omit creating any breaklines), then the triangles are free to form in either fashion, following the rules set by the software's algorithm for triangle formation. This is often simply the order of creation of the points being processed.

Think back to what we learned in school about breaklines. The books all told us that these represented lines along a "break" in grade, and in fact they are. A valley floor, a ridge line, tops and toes of slopes, etc. are all examples of this. But this concept always sort of confused me, and it wasn't until I started creating surface models in digital form that I finally understood them. Breaklines are really nothing more than "edges" in the surface that we are modeling. And the primary rule of thumb is this: *if it is an edge on the ground, then we need to make it an edge in our modeled data sets.*

It is not enough to simply start collecting a large number of points, and in fact, we often don't even need to collect as many points as we typically do. The only points we actually need to collect are those along features that change direction, either horizontally or vertically. And if we connect those points that lie along a particular feature such as the edge of pavement, top of curb, bottom of slope, etc. with linework (*i.e.*, a "breakline"), then the model will be created exactly as we intend. We may need to add additional points along those straight features for more detail in the modeling, but this is more easily accomplished in the computer than by occupying and observing the points in the field. This is a process known as *densification*, and we will explore it a little later in the series.

As an illustration of the necessity of adding breaklines to our observed data, let's consider a very basic yet informative example of a curb along a street edge. As shown in **Figure 3**, we may only actually observe points A and B along the edge of pavement, points C and D along the face of curb, points E and F along the top of curb, and finally points G and H along the back of curb, and not designate any of the edges as being breaklines.

In this condition, triangles are free to form as indicated in **Figure 4.** Clearly, the only valid triangle in this case would be triangle ABC, and while lines EG, GF, FC, and EC are valid edges, the closing segments of the triangles formed (lines GC, FB, and EA) are obviously invalid. This is because point A is allowed to "see" point E, point C can "see" point G, and so on, unless we designate breaklines preventing points from one side of the imposed breakline from "seeing" the points on the other side of this barrier.

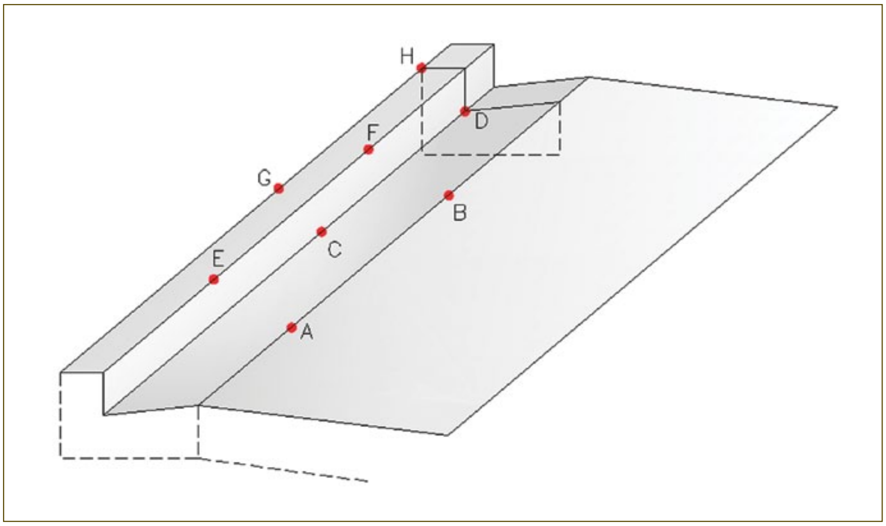

Figure 3 Points observed along curb structure

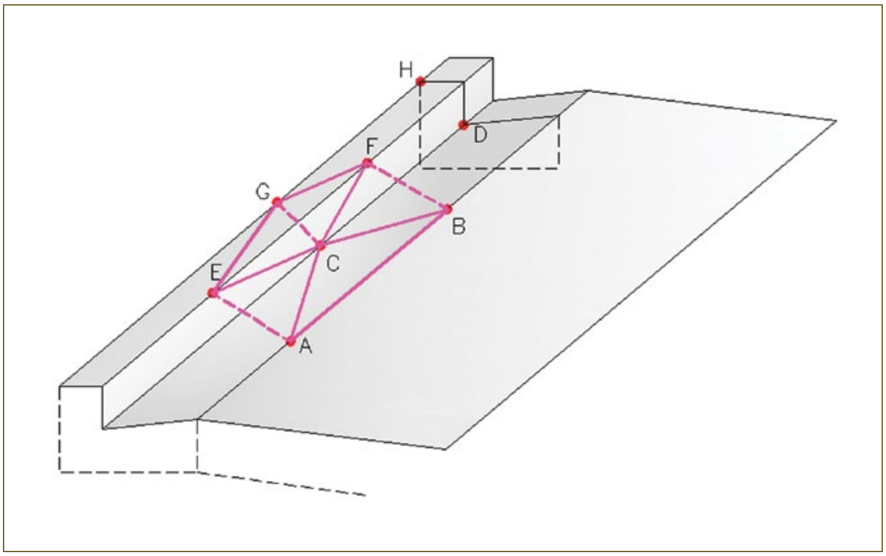

Figure 4 Invalid triangles

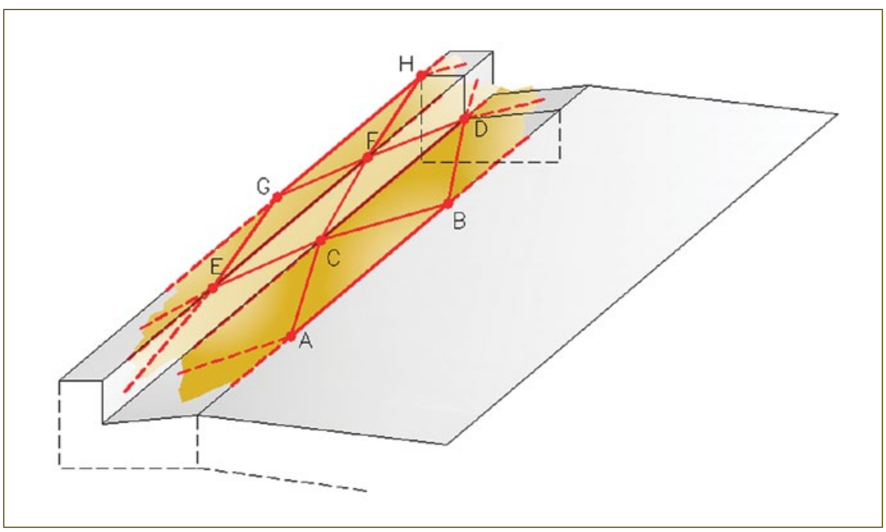

Figure 5 Valid triangles

While if we designate the lines through points A & B, points C & D, points E & F, and points G & H as breaklines (*i.e.*, actually draw them into our data set and include them when we select the data to triangulate), then these serve as barriers and prevent the invalid triangle edges from forming since the points on one side of the barrier can't "see" the points on the other. This will force the triangulation to form only valid triangles which actually lie in the planes of the feature being modeled, which is what we want! See **Figure 5**.

Where to From Here?

Now that we have established a few ground rules, let's take a fairly thorough journey into the actual process in some future articles. We'll learn how to limit the extent of our surfaces (boundaries), how to omit certain areas of doubt (exclusions, such as bottoms of ponds, buildings, tree cover, etc), how to make edits to our surface (although we'll also learn a better perspective on this!), how to combine multiple surfaces into one, and possibly the most important, how to verify what we are doing using good QA/QC techniques and our most trustworthy tool for this analysis . . . our own eyes.

We'll analyze surface data that we "inherit" from others to see if we need to repair anything or if it is in fact usable. We'll learn how to deliver our own surfaces to the next stakeholders downstream so that they can leverage our work to further the project's goals. And we'll learn one of my biggest rules: "the MODEL informs the CONTOURS; the CONTOURS do not inform the MODEL!" More on that later.

Once we're familiar with the process, we'll examine how best to model things we see on the ground by deciding what needs to be observed while we're in the field, what can be modified back in the office, and how to streamline this aspect of our work. You often hear the term "field-to-finish" to describe the act of simply drawing our maps when we get back to the computer, but it can be much more than that . . . it can establish our breaklines for us based on our field coding and keep us from having to do this part by hand once we do get back to the office. A HUGE time-saver, and as you all know, especially now . . . time is very definitely money, and the more we can save, the more we can have left over at the end of the project!