



99.7% Isn't Good Enough

The holy grail of image (and now point cloud) processing is automatic feature extraction (AFE). Practitioners would like to bring up a point cloud data set and push a single button that says “Extract Building Roof Structures” and, in the time it takes the display to refresh, have a resultant vector data set that is correct. We, in the software industry, have been diligently trying to satisfy this desire for at least the past 50 years. I argue that there is not a single area where we have achieved complete success.

Least we simply pick on the image processing (and I mean 3D “images” as well as those from LIDAR and image correlation), simply consider automatic voice recognition. I recall (from the 1980's) Dragon Systems' DragonDictate, a voice recognition system for DOS that matured over the years (and still exists today). When this first appeared on the market, persons giving it a try initially found the concept remarkable. Do a bit of training and then simply dictate those memos (this was pre-email). It was hailed in the computer (and even popular) press as a complete revolution in how we would interact with computers and as a breakthrough in the sticky problem of Artificial Intelligence.

However, Dragon fell as quickly as it rose. The thousands of copies purchased by enthusiastic early adopters and weary two finger typists quickly became shelf ware. What went wrong?

The heart of the issue, I am convinced, has to do with the efficiency of the

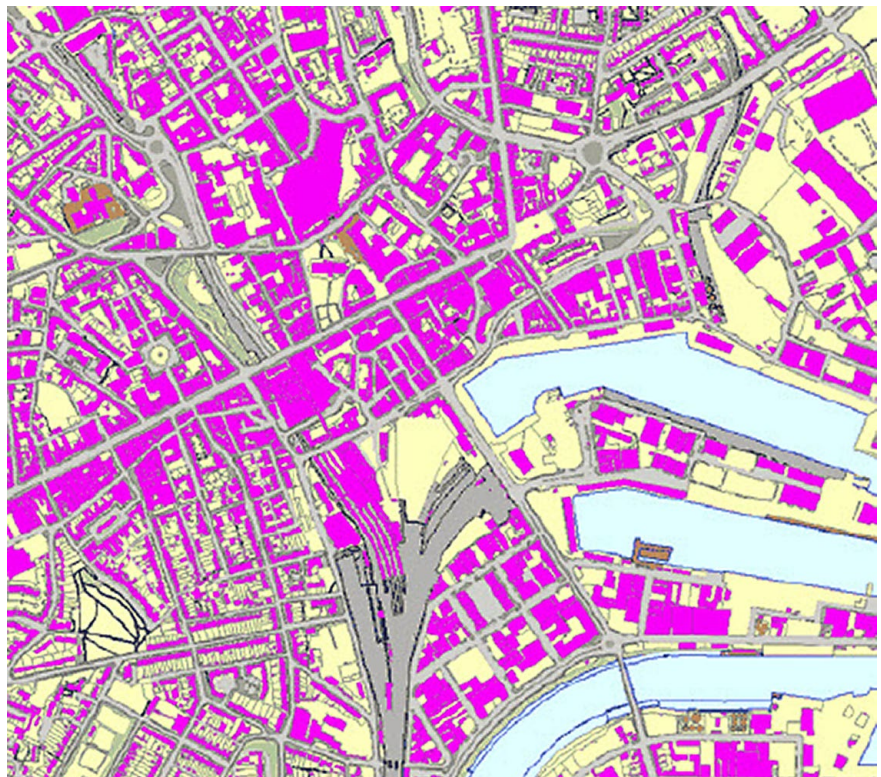


Figure 1: Finding a needle in a haystack

quality check (QC) process. When I dictated a 250 word paragraph to Dragon, it missed perhaps 7 words (on a good day after a lot of training!). This is a success rate of 97.2%. This really sounds great. 97.2% of the dictated words will be correct. The reality of the situation was much different than this respectable success rate would suggest.

How do I clean up this paragraph? All I can do is scan the entire paragraph, looking for incorrect words. It turns out that our brains are against us in this sort of operation. We are incredibly efficient

data interpolators. We have all seen the example of attempting to correctly interpret sentences missing letters from words and, indeed, entire words. We do remarkably well with this challenge.

Unfortunately, our brains merrily interpolate and correct as we scan through data (such as sentences). Thus, when performing this type of QC, we cannot read at our normal reading pace; we must study each word to ensure we see what we think we see. For nearly everyone, this QC process was much more painful (and error prone) than

simply typing the entire memo to start with! Be honest—how many of you use Siri for much more than a party novelty (“Siri, I need to hide a body”).

I have seen this same phenomenon in AFE algorithms. In the 1990s a number of companies were developing algorithms to automatically extract linear features in photogrammetry workflows. As you may recall, this is when base maps for automotive navigation systems and map direction systems were being developed so there was a big economic driver. The commercial systems suffered the same problem as Dragon. Searching through all of the features generated by AFE, looking for errors, was more expensive than simply outsourcing the data collection to cheap, offshore labor. Consider the daunting task of a thorough inspecting of the footprints in the small area of **Figure 1**.

Our company is heavily involved in developing AFE algorithms, particularly in our **LP360** point cloud tools. I am constantly concerned with how we can avoid these algorithms suffering the fate of Dragon—that is, these AFE algorithms becoming shelf ware.

The best solution (and that desired by customers) is to make them function 100% correctly all the time! Unfortunately, I am not aware of any interesting problem where this is possible (note that algorithms such as move all of the green points to class X are not AFE; these are simply catalogers).

A second approach is to tag each identified feature with the probability that the feature was correctly identified. Imagine **low** much easier it **wood** be to proof a dragoon paragraph if known good were green, suspect orange and **wong** red! We simply add a Queued and Interactive Edit process to our workflow and step through the errors.

While this is a great idea in theory, it does not usually work in practice. The first problem with this approach is that any AFE system will, with some frequency, make errors of commission for which the “goodness of fit” criteria are quite high. This leads us right back to the problem of needing to QC every single feature since we do not, with 100% assurance, know the errors of commission. You’ll note the example of this in our color annotated paragraph above with the incorrect word “dragoon.”

The second major problem with this approach is a bit more subtle. If I can devise an algorithm that accurately predicts the accuracy of the AFE, then why not just automatically fix the problem? This is how most algorithms work, of course (consider robust statistics). The problems where I do not predict the outcome then fall right back into the category of undetected commissions. The class of problems where an error can be detected but not corrected are fairly small. The most prevalent that comes to mind is Automatic Point Matching (APM) of multiple overlapping images in aerotriangulation. Here the math clearly indicates when one or more incorrect conjugate points have been selected, but not how to select the correct point.

A third approach is much more pragmatic. This is the approach of a “person in the loop.” For example, in the building extraction problem, the algorithm would proceed as:

1. Automatically vectorize the building
2. Center the display and put the system in Edit mode
3. The person in the loop either presses a “Next” button (all is OK) or starts editing.

This class of algorithm is, of course, semiautomatic AFE and has been the most accepted to date. We have worked with a number of production companies over the years who have compared the efficiency of “person in the loop” semiautomatic AFE to fully automatic AFE/interactive QC. In most cases, the person in the loop approach wins. This is very unfortunate since it relies on continuous human intervention.

I am deeply concerned with these classes of problems. Today, in our **LP360** automatic building footprint extraction, we have taken the full AFE approach. This results in the inspection issue previously discussed. We are trying to develop a novel approach to improving this process. Of course, at this point it is a retrofit. We have one AFE algorithm that simply does not commit errors—a mathematically reduced model of a Triangulated Irregular Network constrained by error parameters (a so-called Model Key Points algorithm). But this sort of algorithm is a rare luxury. We are currently adding rail and wire extraction algorithms to the software. The subject of valid data assurance has been the principle topic during our design meetings. How can we avoid joining the plethora of AFE algorithms that are simply not trusted?

This entire subject is of critical importance to our industry. As data volumes continue to explode and client expectations on the time lines for feature extraction continue to shout “faster, faster!”, we are going to have to consider new approaches. Stay tuned! ■

Lewis Graham is the President and CTO of GeoCue Corporation. GeoCue is North America’s largest supplier of LIDAR production and workflow tools and consulting services for airborne and mobile laser scanning.