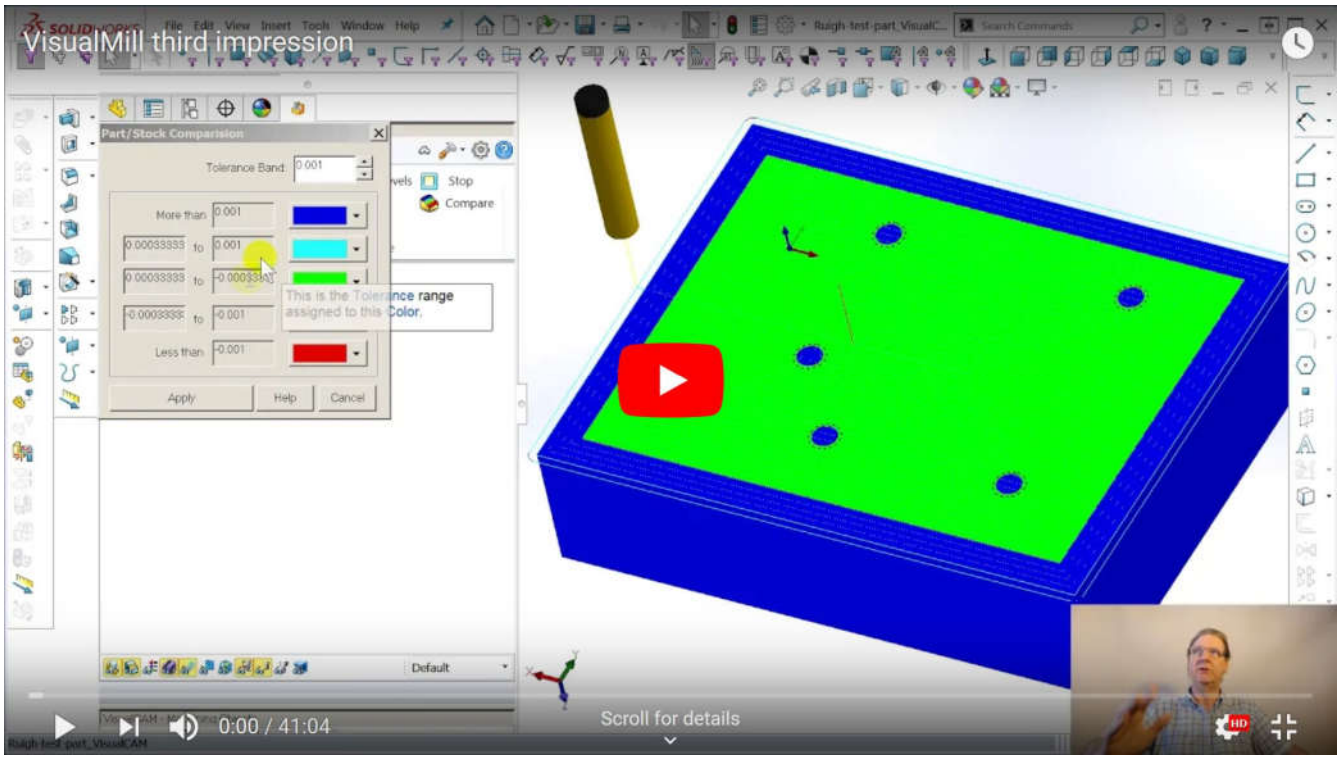
RAKO STUDIOS

VisualMill third impression (youtu.be/Exjk9pTkUR8)

VisualMill is a CAM software that runs standalone and inside SolidWorks. This is the third impression of the program.



SolidWorks part file here.
VisualMill is a mid-range CAM (computer-aided manufacturing) software package. It is priced from $1500 for a basic package to $5,000 for 3D capability, and beyond for mold-making. I have gotten pretty good at using the program, it still has too many quirks on my old Win7 laptop to use. I did look at the "FreeMill" package, and it is a real disservice to the actual program, it is way too simplistic, and more a teaser advertisement than a free CAM program. I had an installation failure with one of the Visual C++ libraries because I had a newer version installed already.

I assume this is why the program has been unstable. My previous installation of Visual C++ might have been done by my TurboCAD 15 installation. It does disturb me that VisualMill needs two different C++ libraries, I assume some of the program uses the older one, and some of the program is written with the newer one, so they have to install both.

Like anything to do with Microsoft, Visual C++ has plenty of bug and memory leaks and other problems. It entices managers who believe the lie that it is faster to program, but it often makes un-maintainable code.

In this episode, I deck off the bottom surface, and drill and tap the 5 holes. Unlike SolidWorks CAM by CAMWorks, VisualMill does not recognize that Solidworks has a tapped 1/4"-20 hole and create three operations, center drill, drill, and tap. Instead, you just do the three operations individually.

I almost prefer this, especially since SolidWorks CAM never was able to automatically do the 5 holes in one set of operations. At first it tried to do two holes from the top, and three from the bottom. When I limited the feature recognition to the bottom, it did them as two separate holes and then a group of three. I'll take manual input any day over fighting some programmer's algorithm.

My biggest concern was that SolidWorks had the holes as 0.38" deep, to get the depths to match, I had to set the VisualMill holes to 0.44" deep. Perhaps one set of programmers has decided to measure from the tip of the drill, and the others are measuring from the start of the flutes.

While a bit tedious, I did get the bottom side faced and the five holes drilled and tapped. I was astonished there are not any existing tools or drill or taps in the program, I assume that is because this is a demo. Otherwise, it will be moderately painful to enter each and every tool you own into the program.

After getting the bottom done, I returned to the top side, which was undone since I didn't want the program to crash like it did in practice runs when I did the whole top side first. That went well, and I actually finished the part. I did see where I had the shallow pocket selected as well as the angled flat in one operations, so that was a dumb mistake.

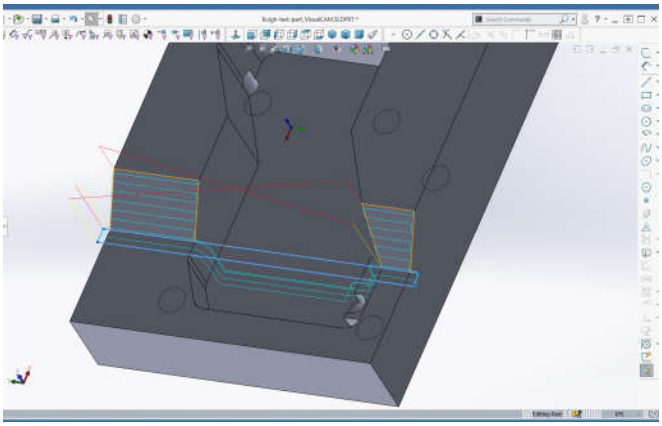
Since doing the "second impression" video, I learned more work-arounds to do the top side operations. To get the pocket corners more accurate, go to the "Advanced Cutting Parameters" tab in the "Parameters" dialog, and tighten it from 0.01' to 0.0001". That will force the program to put little arcs in the tool-path corners instead of a straight-line segment.

Like in SolidWorks CAM, I had to make a sketch in the part so I could make the shallow pocket properly. The sketch bridged the breakout drill holes coming up from the bottom of the part.

I also had to make a sketch to do the flat, though I suspect there is a simpler way, I could not figure it out. Using the 3D operation "Clear Flats worked, but it was not able to machine down in several levels, so I used a sketch and a pocket operation.

The third kludge sketch was to insure the ball mill went all the way down to the flat as it machines the two angled slanted surfaces. I also changed from "Horizontal Finishing" to "Parallel Finishing," so I could extend the tool paths and clean up the angled edge of one slant.

I learned the program is "model aware" when I made a single sketch to do the angles:



The toolpaths will drop down inside the pocket.