Tuesday, April 3. 2012

Use netfabb to Manually Repair STL Holes, Edges and More

Tweet

In a previous post we showed how to use a free tool, netfabb Studio Basic ('netfabb') to automatically repair STL files in preparation for 3d printing. In this post we use netfabb to manually repair some common defects found in STL files including holes, boundary edges and invalid orientation. This is by no means an exhaustive example of netfabb's manual repair capabilities. Rather it is meant to introduce common problems encountered in STL files and how netfabb's manual tools can be used to repair them. For more in-depth information check out netfabb's wiki.

For this example we'll use a file which Objet distributes as an example of a bad STL. Appropriately enough it's called 'BadSTL.stl'. We'll repair the file using the following steps:
Load the file and check netfabb's preliminary analysis
Perform a Standard Analysis and note results
Examine the file in repair mode
Perform manual repairs.
Apply repairs and export the repaired file as a new STL

1)Load the file and check netfabb's preliminary alerts
After loading the file into netfabb we're greeted with the now familiar red exclamation point informing us that there are potential problems with the file. Lack of a calculated volume is another indicator of potential problems. Further, netfabb makes extensive use of color coding and trouble spots are colored maroon according to the color scheme in Settings>General>Colors (note that our part is 'selected').

2) Perform a Standard Analysis and note results
Like our previous example, we'll start by selecting the part and pressing the Standard Analysis button on the upper right. netfabb's colors will change again and we see a gray rendering of the part with possible problem areas again highlighted.

A new pane will appear on the lower right of the screen with quite a bit of new information. For the time being we'll only be concerned with the green/red Yes/No near the bottom. Two of the most common STL mesh problems are surfaces which are not closed and/or orientable. netfabb indicates with a 'Yes' if it detects these problems. See green and you've probably got an STL file which is technically printable; red and you might have problems. In our case we see that our surface is NOT closed but it is orientable:

Let's briefly explore the meaning of these two defects. If an STL mesh is not closed (or 'manifold', or 'watertight') the 3D printer software will have difficulty defining where material should be placed. This could occur if, for example, in the design process a single edge is actually made up of two non-coincident edges or there are holes in the surface. The difference may not be visible on the screen but it can confuse the printer software. With a red 'No', netfabb is telling us that we have some sort(s) of openings in our mesh.
If an STL mesh is not orientable it means that there does not appear to be a consistent inside and out (as defined by the normals of each triangle). Again, if parts of the model are 'inside-out', 3D printer software won't be able to discern where to put material. In this case, "Yes" indicates that netfabb is able to make some sense of inside and out (although there may still be 'fixable' problems).
If we study some of the other information provided we'll discover that netfabb has found 3 holes, 9 boundary edges and 51 flipped triangles. Flipped triangles indicate areas in which inside and out seem to be reversed. Our surface is orientable so we won't worry too much about these for the moment. Holes occur where there is an opening exposing the interior of the mesh. Boundary edges occur when we have a 'naked' edge exposed to the world. Since mesh surfaces

have no thickness these edges have no meaning for physical printing. Printer software is often robust enough to ignore such edges but results can be unpredictable.
Perhaps our 3 holes and nine boundary edges are related; if our three holes are triangular each would create three boundary edges. If we examine the on-screen rendering and rotate it around a bit, we definitely see some triangular holes. netfabb colors the interior and exterior of each triangle differently and what we might have initially mistook for flipped triangles we actually discover is the inside of the opposite surface; i.e. we're looking through a 'hole'. Since we clearly have holes - and they might be triangular - we'll concentrate on hole fixing first
3) Examine the file in Repair mode
Now that we have an idea what some of the problems with the STL might be we're ready to begin repairs. We'll press the red cross repair button and note that we're now presented with a new rendering; a mesh with a yet another color scheme (as defined in Settings>Part Repair>Colors).
At this point, like in our previous example, we could try to use netfabb's automatic repair scripts. If we did, we would in fact find that netfabb would repair this file automatically with no difficulty. However, to explore netfabb's repair capabilities (and a little bit about typical STL repairs) we'll continue manually.
4) Perform manual repairs
First we'll address the unclosed nature of the surface. Some purple flipped triangles and yellow boundary edges (or whatever colors you've set) are clear.Again, if we hold the right button and rotate the part we'll notice that some of what were apparently flipped triangles are actually the inside of the opposite surface as seen through holes in the mesh.
To fix these holes we'll need to add triangles. We find the green triangle with a "+" on it on the top toolbar and select it.

Now we can select any two border edges (in yellow) and netfabb will create a new, surface triangle using those two edges and the most logical third edge. If the hole is triangular it will be filled normally - if not we may need to add more than one triangle. We do this for all of the holes we can find. Note that we've adjusted some of the color settings from defaults to make defects easier to see. (The Pro version of the software offers automated selection of defects - a real timesaver on larger files with thousands of defects). Now we update the status:

Not only have our 3 holes gone away but our 9 boundary edges have disappeared as well. As we suspected, the boundary edges were a result of the holes in the mesh - 3 per hole.
However, the status box (and the colors on the rendering) suggest that we still have 51 flipped triangles. We'll address those now by selecting problem triangles and re-flipping them. In the free version of netfabb it's necessary to hunt down flipped triangles manually. Triangles can be selected on-by-one but with large regions it worth trying the surface or maybe even solid tools for selection. In practice the surface selection tool (green pentagon) seems to be most useful.

After selecting triangle(s) we change their orientation by pressing the flip triangle button - just to the right of the selection buttons:

After de-selecting these triangles they will be the same color as their neighbors (in our case, blue).
Updating our status we see that there are still triangles with invalid orientations. In this case we need to hunt for particularly small and/or hidden triangles (again, a good reason to go with the pro version if you're going to be doing this often). We find a couple more very small triangles which can be selected and flipped in turn.

Again updating our repair status we find zero border edges, invalid orientations and holes.
5) Apply Repairs and export the repaired file as a new STL
As in our previous post we still need to apply repairs to the original file and export a new STL. We press Apply Repair (see above) and discard the previous part. Under Part we now select Export as STL (repairing the file further as needed) to create a new 'repaired' version of our original STL. This new version is clean and printable and we're done.
3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 11:43

Monday, March 19. 2012

Automatically Repair STL Files in 2 Minutes with netfabb

Tweet

If you're reading this post you probably know that to print in 3D you need a 'good' or 'clean' STL file exported from your 3D CAD program; models seemingly perfect on-screen may be filled with defects which make 3D printing difficult if not impossible. You might have even heard of the apparent black magic that is STL repair. Chances are though that you have no inkling as to what that really means or how you could do it yourself.
In this post we're going to go through a simple example showing how to use a great free tool, netfabb Studio Basic ("netfabb") to automatically 'repair' STL files for 3D printing. Although the program allows for manual and semi-automatic repairs (we'll cover some of these in a later post) automatic fixing seems to work for about 90% of current STL files. Best of all it's fast and you don't need to understand much about STL files or their defects; you just need to be able to find your way around the sometimes unintuitive user interface.

If you don't have netfabb download it from here. There are versions for Linux, Mac and Windows. For this example, we're using the just released Windows v 4.9.0 and a Porsche STL from here. The process we'll follow is roughly as follows.

Load your file and check netfabb's preliminary analysis
Perform a Standard Analysis
Perform Repairs
Apply repairs to your file
Export the repaired file as a new STL

1) Load your STL file and check netfabb's preliminary analysis
Load your file by choosing 'Open' under the 'Project" menu in netfabb. When opening an STL file, netfabb performs a preliminary analysis to determine if there are issues that could cause problems during 3D printing. The most common issues include holes, 'naked' edges, and triangles with invalid orientations (i.e they are inside-out).
If problem(s) are found, a large, red "!" will be displayed in the lower right hand corner and (usually) a part volume will not be calculated. In the case of our porsche file netfabb has found problems and alerted us:
If you do not see the red attention warning, congratulations, your file is very likely ready for 3D printing and nothing further is needed.
2) Perform a Standard Analysis
Now that we know that netfabb has found problem(s) we'll perform a more thorough analysis. Find the menu icon in the upper right which looks like a circle with a magnified section. Under this select the option for 'Standard Analysis'. netfabb will work for a bit and a new 'layer' will appear with the part now rendered in gray and defects in different colors (as defined under Settings>Colors>Repairs).

Nothing has changed with your original file. The information panel on the right will now include a summary of the type and number of problems. Pay attention to the number of problems but especially take a look a little further down to see if the surface is closed and/or some of the mesh seems to be pointing inside-out.
In our case, a red 'No' tells us that our surface is not closed. 3D printers don't like unclosed (non-manifold, non-watertight) surfaces and will often not print or will print additional artifacts when they are encountered. They are usually caused by small holes or edges that don't meet and may not even be visible in an on-screen rendering. A green 'Yes' indicates that our file is 'orientable' and therefore seems to be free of another common defect: inside-out triangles.
Now that we have loaded and analyzed the part we are ready to attempt to automatically repair the file.
3) Perform Repairs
To repair our file, find the red cross menu item in the upper right next to the analysis button and press it:

You'll notice that yet another 'layer' is created underneath the 'Part Analysis'. The triangular mesh is now shown on the model and new options and information are available in the lower pane. Press 'Update' to see a count of each type of error. Next, select 'Automatic Repair' and then choose 'Default'.
netfabb will now go through a series of repair algorithms to attempt to make the STL file printable. (You can see what these steps are by clicking on the Repair Scripts tab in the lower right of the information pane.) This may take a little while especially if your part has a large number of triangles. For example, on a dual core 2.4 GHz computer this 7.4MB, 151k triangle file took just over 6 seconds to be run all of the default repair scripts. A not-so-obvious status bar in the lower right hand corner will show progress.When the process is complete you can again press the 'update' button under the status tab. You should see zero border edges, invalid orientations and holes. If your part is a single object (vs. an assembly) it will likely indicate the preferred 1 shell. While not optimum multiple shells will not usually cause printing problems. You should also visually verify that your model still looks the same as your original. In some cases, netfabb automatic repairs may create solids where in fact a hole was intended. This is rare but you should still check visually.

At this point you could continue with manual repairs if needed/desired but we'll stop here since it looks like the default automatic repairs by netfabb have been sufficient. In a subsequent post we'll look at some common manual repairs in netfabb.
4) Apply Repairs to your file
We're not quite done as we still need to apply the repairs to the originally loaded file by pressing 'Apply Repairs' in the lower right hand corner. This removes the analysis and repair layers and fixes the original rendering. You should see the original green rendering - but without the red attention warning. You should also see a volume calculation.

(If you still see an attention warning and/or lack of volume calculation it means that netfabb was unable to completely repair your file. The file may still be printable or it may require further manual repairs beyond the scope of this post.)
5) Export the repaired file as a new STL
At this point it is important to understand that you have made no changes to your original file. If you choose "Save" you will create a new netfabb 'Project'. Since we started this process to create a clean STL file we now need to create a new (repaired) file.
Under the Part menu select "Export as" STL. This will bring up a new window. First, check that the file name and location are what you would like. netfabb will automatically create a filename composed of the original with '(repaired)' appended so you don't have to worry about inadvertently overwriting your original file. However, the location will be not necessarily be in the same folder but rather the last folder saved into.
When you press 'Save' a new dialog will appear with another analysis of the file to be created and possible errors. It seems that this may occur because the netfabb file format contains more information than, for example, the STL format; some of the repairs it has made might therefore not be carried over. Again though, automatic repairs can be made. Press the 'Repair' button if you see a large red 'x'.

If successful, instead of a large red 'x' you will see a green check mark. In this case, 9 seconds and a single iteration were sufficient to repair the 422 manifold edges. In other cases it might take multiple iterations and you may still have a non-zero number of errors. The most recent release of netfabb Studio Basic seems to have improved this functionality. Just keep trying until you either have a green check mark or a minimum non-changing number of errors. (if your totals go up you can cancel the export and select again to start fresh).Once your file is 'repaired' you can press 'Export'.
Congratulations, you now have a printable STL file. Want to learn more? See our post on using netfabb to manually repair STL files.
3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 10:20

Friday, January 20. 2012

Of Geckos and Fat Fingers: Why We Don't Have 3D Printing Replicators

Tweet

In Star Trek: The Next Generation Captain Picard is often seen ordering "Tea, Earl Grey. Hot" from the ship's computer. Flawlessly delivering his order every time, this replicator is everything we could hope for in 3D printing: on-demand, customized, localized, rapid manufacturing. Unfortunately, the high profile rollouts of consumer level 3D printers at this month's Consumer Electronic Show only serve as a reminder of how far we are from true in-home replicator reality. Crudely fabricating small parts by extruding hot plastic through a nozzle is a long way from replicating an iPhone or Earl Grey. So what's the holdup?

The Perfect 3D Replicator
In a 1959 presentation to the American Physical Society titled There's Plenty of Room at the Bottom Richard Feynman famously postulated on the future of fabrication in asking:
... whether, ultimately – in the great future – we can arrange the atoms the way we want; the very atoms, all the way down! What would happen if we could arrange the atoms one by one the way we want them ...
This presentation is credited with being the inspiration for today's nanotechnology, launched by Eric Drexler in his groundbreaking 1986 book Engines of Creation. With the ability to manipulate individual atoms using what he calls a universal assembler, Drexler expounds on many future possibilities, including a new industrial revolution as we change the way we 'make' things at the most fundamental level.
But how to create a 'universal assembler'? Our natural tendency is to start with what we already know. If, for example, we can build cars and buildings by picking up materials and placing them into a predetermined position why not the same thing for atoms? Why not build a nano-scale robotic arm which could pickup one atom at a time and place that atom in a pre-programmed position? And heck, while we're at it, why don't we use our arm to precisely position carbon atoms so that everything we make is out of incredibly strong diamond? With such an arm we would be well on our way to atomically precise manufacturing and transforming the science fiction of a Star Trek replicator-style 3D printing into reality.
The problem is, this universal assembler has proved extremely difficult to build. In fact some think it's impossible. Why?
Fat Fingers
Imagine a bin of beach balls that we wish to move to another bin. Also imagine that our hands are encased in ... beach balls. Of course our dexterity would suffer and our ability to perform the task would be severely compromised since our tools - our hands - would be essentially the same size as our materials. This is what we are faced with at the atomic level. Atom sizes do vary but the size of those with potential as structural materials tend to cluster around an angstrom or two in diameter. Building a universal assembler will be like typing on a small keyboard with 'fat fingers'.
While working with 'fat fingers' might be mechanistically difficult, one can imagine different scenarios where it may be possible. Unfortunately that is not the end of our difficulties.
Gecko Force
For years scientists were puzzled by the geckos' ability to climb on just about any surface. They ruled out any secreted glues or simple water adhesion before finally concluding in 2002 that the millions of tiny 'hairs' (setae) on the surface of a geckos' feet adhered to other surfaces through a weak atomic interaction known as van der Waals forces.
Although stable atoms are electrically neutral their electrons can vary in position causing minute electric field fluctuations. These fluctuations polarize the atom and, like the opposite poles of a magnet, other atoms can be attracted. This van der Waals force is very small in magnitude especially compared to common covalent, ionic and metallic bonds. It's also very dependent on the geometry of the interaction. This allows the gecko to adhere or peel off by simply changing the angle of its foot. Still, we normally don't come across this at a macroscopic scale; put a book on a table and it doesn't stick.
At an atomic scale it's a different story. This force is very real and of strength to throw a wrench into our atom handling. Quite simply, our tools will have a tendency to stick to our materials. Once we 'pick up' an atom, we'll have difficulty putting it down. Imagine trying to type with fingers that are both fat and covered with glue.

We will be faced with still more problems such as atom delivery and assembler programming and speed (self-replicating nanobots anyone?). But having to work with fat and sticky fingers are perhaps the most fundamental problems of all. (Richard Smalley - the originator of the fat, sticky finger idea - and Eric Drexler famously debated the issue in 2003.)

So Where Does That Leave Us?

The holy grail of 3D printing would be a universal assembler. With it we could precisely place whatever atoms we would like, in whatever fashion we would like, whenever we would like in order to exactly replicate anything we have a design for. Unfortunately, the problems of fat and sticky fingers mean that simply scaling down our macroscopic machines to an atomic level will likely be difficult if not impossible.

However, 3D replication already does take place at the atomic level, most significantly in biological systems. Ribosomes, for example, use a sophisticated interplay of mechanical and chemical properties to create macroscopic structures out of individual atoms and molecules. We've also shown that we can, under specialized conditions, nudge atoms around, make small 'machines' and a patent was even issued in 2010 for a simple diamond mechanosynthesis tool. But while the idea that we can fabricate objects with atomic precision is by no means dead, all of these examples are for specialized end products under specialized conditions. Eventually we may truly be able to fabricate replicator-style, but it will likely involve customized methods of manufacture for each end product. And it likely won't happen anytime soon. The simplest most elegant solution, a universal assembler may forever elude us simply because of fat and sticky fingers.

For more on the fascinating debate between Eric Drexler and Richard Smalley see here and here. Drexler has backed away from the universal assembler but still insists on the feasibility of atomically precise manufacturing.

3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 16:18

Monday, January 2. 2012

1859 - The Year 3D Printing Hit the Mainstream

Tweet

3D printing and additive fabrication made it onto a number of 'top lists' for 2011 and 2012. Often it was something about how in 2011 3D printing had become much cheaper and more accessible or how, in 2012, 3D printing will finally be 'going mainstream'. Based on all of the hype one might think that 3D printing is new on the scene. The reality is much different.
In 1859, a year before Abraham Lincoln was elected president, a Frenchman in Paris, François Willème began creating photosculptures of real people using techniques which, while perhaps primitive by today's standards, are really not all that different than those techniques currently used in 3D printing.
It's 1859. How Do You Build a 3D Scanner?
The first challenge in 3D printing an object is collecting the data necessary to represent the object. Today we create or collect that information either through designing in 3D on a computer (3D CAD) and/or using some form of laser or whitelight scanners to map a 'point cloud' of the surface of an existing object.
In 1859 Willème, faced with the same challenge, went to some of the hottest new technology of the day - photography - to capture the essence of his subjects, their profiles. Willème would arrange his subject on a circular platform, surrounded by 24 cameras, one every 15 degrees. He would then simultaneously photograph their silhouette from each camera. This photographic set of profiles contained the data for a complete representation of his subject in 3 dimensions, albeit with relatively coarse resolution. (Interestingly, the artist Auguste Rodin apparently used a similar technique by examining his subject from numerous angles to create a mental 'profils comparés').
Although Willème's technique was analog and chemical rather than digital and magnetic it is remarkably similar to today's 3D scanning techniques which collect and record electromagnetic radiation reflected off the surface of a subject in order to store data for later three dimensional representation and/or fabrication.
3D Printing - 1859 Style
In additive fabrication (we use the term interchangeably with '3D printing') an object is created by laying down successive layers of material. Whether it's a layer of sintered metal, melted plastic or some other material, 3D printing is a layer technology. 3D representations are virtually sliced and machines are used to fabricate each slice.
Willème had now collected layer data for his subjects in the form of 24 different photographs of their profile. To 3D print his subject he first needed to make accessible each layer's information by projecting each image onto a screen. Next, he needed to translate each image into the movements required to fabricate each layer. This was accomplished using an already 250 year old technology, a precursor of numerical control (NC), the pantograph attached to a cutter (here's a fun java virtual pantograph). Now he was able to trace each profile with one end of the pantograph while the other end cut a sheet of wood with the exact same movement. (Might we call his tracing a precursor to g-code?) The pantograph allowed the cuts to be smaller, larger or the same size as the original projection. The layers of wood were then assembled to create the photosculpture. Finally, if desired, an 'artist' could finish the work making it truly a work of art.
Again, we see that while seemingly primitive, Willème's techniques are very similar to today's 3D printing: 1)take a 3D representation of an object and slice it into layers; 2)use that slice information to fabricate material layers which are combined to make the finished object.
Combinatorial Creativity and 'Killer Apps'
So what's the big deal? First, as the hype surrounding 3D printing continues to grow, it's important to remember that, as a concept - and even as a practical technology - it's not really that new. And until a molecular assembler or similar is developed, we're perhaps being a bit over-exuberant when we talk about, for example, new paradigms for manufacturing. 3D printing is darn cool but it's not yet changing the world.
More important however is recognition of the combinatorial creativity demonstrated by François Willème's creative mashup of new and existing technologies. Maria Popova describes combinatorial creativity as:
...the idea that creativity is combinatorial, that nothing is entirely original, that everything builds on what came before, and that we create by taking existing pieces of inspiration, knowledge, skill and insight that we gather over the course of our lives and recombining them into incredible new creations.
This concept is important for 3D printing since, so far, it doesn't have it's 'killer app'; no uses for 3D printing have been

created which would make all (or most) of us users. But most anyone paying attention to 3D printing agrees that there WILL be a killer app for 3D printing. The convergence of different technologies just seems too ripe for it NOT to happen. Will it come from the marginal improvements of those already in the industry or might it come from someone outside? Perhaps an artist or maybe an engineer in an entirely unrelated field? When we understand what François Willème did in 1859 the outsider idea seems even more probable. Someone, with just the right combination of knowledge and experience to truly take 3D printing into the mainstream, is likely already out there. They might not even be aware of 3D printing, let alone use it in any practical sense. We can try to help them along by hyping 3D printing, by developing cheaper hardware and materials or by introducing children to 3D printing. In the end though, our killer app will likely arise out of a set of circumstances that in retrospect seems random but in reality represents a continuation of the combinatorial creativity that began more than 150 years ago in the studio of François Willème.
For more on the history of 3D printing and additive fabrication check here, here and here.

3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 18:13

Wednesday, December 14. 2011

Why are 3DAddFab's Prices So Low?

Tweet

"You want how much for that little hunk of plastic?!" When it comes to 3D printing we think the answer is usually 'too much'. Sure, if you've been in the additive fabrication industry for 10 or more years you've seen some pretty significant price drops. But if you've just discovered 3D printing, and the wow factor still makes you giddy, you're likely to be taken aback by how expensive it can be. Some of the price, at least for a non-OEM 3D printing service provider, is unavoidable. (See here and here for our earlier look at how equipment and materials lead to higher prices.) The rest of the price however is directly related to the business practices and policies of 3d printing service providers.

At 3DAddFab you'll find a perfect storm for low prices. Ok, so Fast Eddie's Used Cars might put it that way but let me explain. First, we want 3D printing prices to be lower. We think all of the cool things that seem possible with additive fabrication will only ever occur if more and more people are involved, spawning creativity and innovation. It's not a coincidence that the explosion in creative uses for the internet has coincided with an explosion in users since the mid 1990s.

Additive fabrication needs the same thing. What's the 'killer app' that will lead us all to have 3D printers in our homes? For desktop computers it was word processing and for the internet it has been web browsing. We don't pretend to know what it might be for 3D printing. One things seems clear though: it's not out there yet. How might we shorten the timeframe and increase the likelihood of development of an additive fabrication 'killer app'? Get more people involved. There are any number of ways to do that but the one area where we, as 3DAddFab, can have a direct impact is through our pricing. We feel that the lower the price for high quality printing, the more people that will be attracted to 3D printing. So in our gut, we want lower prices.

The second part of the perfect storm is that we're a startup business. We're bare bones and other than offering instant quotes and really fast service on a high quality product, we don't have too many bells and whistles. In addition, you likely haven't heard of us and we might lack credibility. But we want your business. So in the spirit of Adam Smith and free markets we price aggressively. We do whatever we can to get you to look in the window and then we try to help you mitigate the risk of trying us out by offering substantially lower prices than the more established Goliaths out there.

So check us out. No user registration required for an instant quote at our On-Demand prices. Or save even bigger by buying in bulk.3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 14:46

Tuesday, November 29. 2011

Visualizing 3D Printing and Additive Fabrication on Twitter

Tweet

For those of us more right-brained than left, pictures so often seem better than words for getting points across. How then can we make sense of something that is primarily words, like Twitter?
Summarizing keywords in a 'word cloud' and indicating their importance with font size, weight and maybe color is one way to do it. Word clouds have become especially popular in the last ten years as we've moved on to Web 2.0 and a number of tools exist for creating them. One particularly cool tool is Infomous by Icosystem. Anyone can set up a free account and create embeddable word clouds using their patented text visualization algorithm. After choosing your keywords you then select from a variety of sources such as Twitter, Facebook, blogs etc.. Finally, you can copy code for your new word cloud and embed on your own page.
But a picture is worth a thousand words so check check out the current Twitter conversations for '3d printing' and 'additive fabrication':

3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 14:00

Wednesday, November 9. 2011

AMF - The 3D Printing Format to Replace STL?

Tweet

In a previous post we examined the STL files used for moving from 3D CAD designs to 3D printer hardware. Despite the STL format's shortcomings it has remained the standard of the 3D printing industry for nearly 25 years. However, continued innovation, especially in 3D printing hardware, have forced a variety of stakeholders to adapt and extend the STL format to address their particular needs. Today, one can work with a standard STL file or any of a number of custom and/or proprietary STLs. Recognizing this as inefficient and unsustainable, in 2009 the ASTM Committee F42 on Additive Manufacturing Technologies formed a task force to specify 'STL 2.0'.

The PDF of 3D Printing?

The committee's goal was to create a standard additive manufacturing file format that would be more compact than STL and would include additional features such as colors, units and materials; sort of a Portable Document Format (pdf) for additive manufacturing. In this way, as much information as possible describing an object could be included in the 3D CAD design phase before being exported in a standard format for the fabrication phase. Additive manufacturing OEMs (or others) would then develop software to read this format and use whatever information would be applicable for their particular hardware.

In June, 2011 the first revision of this new format, dubbed AMF (Additive Manufacturing Format or Additive Manufacturing File) was approved. AMF format was designed to:

be non-proprietary

backwards compatible with STL

able to handle complex objects and microstructures within objects

support 'constellations' of objects (e.g. a full build tray)

be forward compatible to allow future incorporation of new features

AMF Format is XML-Based

To allow for ease of use and forward compatibility, the AMF format is text-based XML. XML is highly compressible and easy to read, write and process. An AMF file describes first the object (currently with the tag) and then the materials and other properties relating to it's manufacture. Significantly, AMF provides no information as to how to fabricate an object. For example, although additive fabrication requires 'slicing' files before generating machine paths, the inclusion of slice information was rejected as too machine-specific. Just like the 2D pdf format, the idea of the AMF format is to provide as much information as possible about and object and let each printer fabricate to the best of its abilities.

AMF Uses Triangular Meshes to Describe Surfaces

An AMF is made up of and volumes . Like .PLY files, all vertices are defined (later to be referenced according to write order) and then for each , triangles () are defined as a set of three numbered numbered vertices. This is a bit more efficient than STL files since vertices are only defined once. An excerpt of our original 10 centimeter diameter sphere,

translated to AMF format using an open source AMF editor:
00000001 00000002 00000003 00000004 sp00000005 00000006 00000007 00000008
00000009 -3.06161e-01500000010 000000011 -49.999900000012 00000013
00000014 00000015 00000016 -0.049378700000017 -1.4014400000018
-49.980200000019 00000020 ...00173292 00173293 00173294 Default00173295
00173296 2464200173297 2464000173298 2441700173299 00173300 00173301
2464000173302 2463800173303 2441500173304 ...00420820 00420821
2464000420822 2464200420823 2475400420824 00420825 00420826 00420827 00420828
With additional tags and definitions, an AMF file might actually have more lines than an equivalent STL file, but it will likely be smaller and more easily parsed. For example, although the AMF file above has 20% more lines than the equivalent STL file it is 27% smaller in size, even as human readable text.

AMF Meshes Not Limited to Straightedged, Planar Triangles

Like STLs an AMF can simply be composed of straight edged, planar triangles. But an AMF mesh doesn't have to stop there. AMF triangles can more accurately describe a 3 dimensional surface by being curved edge and/or non-planar. By adding a unit normal tag () to vertices, non-planar triangles can be described. And by adding an an tag specifying the tangent direction vector at the end of an edge, curved edge triangles can be described.

Finally, and perhaps most interestingly, an AMF mesh requires a certain depth of recursion when it is read. It is expected that a reading program will subdivide each triangle into a specified number of planar subtriangles to be used for viewing or CAM algorithms. For example, a four-fold recursion would result in each triangle being subdivided into 256 subtriangles. The encoding software would assume a four-fold recursion and would then determine the minimum number of triangles required to specify the target geometry to a chosen tolerance or precision. This results in a SIGNIFICANTLY fewer number of required triangles to define an object (although read/process times are increased slightly). For example, a 10 cm sphere defined to a precision of 10 microns requires less than 400 non-planar triangles to define. A similar STL file requires 20-50 thousand triangles. In our ealier post we were only able to reduce the number of STL triangles by reducing the precision to 1mm. Looked at another way, an STL file with a similar number of defined triangle facets is 2-3 orders of magnitude lower resolution than an AMF file describing the same surface but with non-planar, curved edge triangles.

Comparison of STL & AMF - 10cm Sphere (Solid Edge)

STL Format
(Binary)

AMF Format
(Curved Triangles)

PRECISION

10 micron

10 micron

NO. OF MESH TRIANGLES

49,500

320

FILE SIZE

2400k

10k

Alternatives to Possible in Future
In choosing how to describe the surface of an object several options were considered: the STL-like triangular mesh described above; using 'voxels' to define a 3D bitmap; or functional representations where the function evaluates to zero at the object surface. Ultimately, to ensure a quicker transition and simple backward compatibility, the triangular mesh () was settled upon although this choice was not unanimous (e.g. see here). However, it is anticipated that 3D bitmap () and functional () representations will be incorporated into the standard in the future.
AMF Can Describe Materials, Compositions, Colors, Textures and More
In addition to definition AMF goes beyond standard STL files in being able to define available materials (). Materials are defined with particular properties, given an ID and subsequently associated with AMF volumes by ID. Compositions () colors () textures (etc. can be specified for various materials. Microstructures, colored or textured objects can now be defined in CAD programs and then have that information available in a standard AMF file. 3D printers would then use whatever information the need to match their capabilities. Can't create compositions or colors? That information would simply be ignored.
AMF Can Describe Arrangements of Different Objects
AMF also allows for the creation of groups of objects in constellations (. Instances () of defined objects () are grouped under the tag and specify the displacement and rotation of each instance of an object. In this way, entire build envelopes can be specified.
AMF Can Include Metadata
Finally, AMF allows for the incorporation of metadata () at whatever level is feasible. A number of 'types' have been reserved including Name, Author, Company, Description, Volume, Tolerance, etc. and they can be placed under everything from the top-level tag all the way down to the tag.
The Future of AMF
The AMF format is an exciting development in an area that has seen little in the last 20 years. It will now be possible to more accurately define objects using a simpler more efficient file structure.
Unfortunately, though a first revision has been approved there is little use for it at this time; CAD programs are incapable of exporting in AMF format and 3D printer OEM programs do not allow for AMF import. The ASTM F42.04 committee predicted that "Big CAD" will first allow for export of basic geometry to AMF files and 3D printer OEMs will allow for imports of AMF files with information that supports their unique hardware capabilities such as color or material specifications. The committee hopes that in parallel smaller or new CAD companies will introduce programs to actually create objects with different microstructures, colors, grade materials etc. which the AMF format - an a number of actual printers - already support. Finally Big CAD would/should follow.
Will it happen? More than likely. The committee has done a good job of creating a robust, extensible format that addresses both the increased 3D CAD and 3D printing capabilities which now exist. All that are needed are conversion and creation tools. To that end, Hod Lipson and Johnathan Hiller have developed an STL-AMF converter/viewer, a second generation STL-AMF converter/viewer (although we found it a bit buggy) and are working on open source C++ libraries for basic AMF functionality including read/write/view etc.
In the meantime, for most of us, we must simply sit tight and wait for new or existing software to incorporate the AMF format. For those with programming skills this might represent an opportunity. With open source C++ libraries, it may be possible for independent programmers to begin developing the solutions which the industry needs but which, it is expected, Big CAD and OEMs will be hesitant to implement.
3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 16:09

Sunday, November 6. 2011

What is an STL file and is it obsolete?

Tweet

For nearly 25 years the STL file has served as the bridge between 3D CAD designs and 3D printer hardware. To slice object designs and calculate machine paths, 3D printer CAM programs have been designed to accept precise design data as formatted in STL files. However, STLs are likely reaching the end of their useful lifespan as today's printer hardware has capabilities not easily incorporated into their limited format.

Developed in 1987 for 3D Systems the STL format was designed as a standard format to allow data movement between CAD programs and stereolithography machines. 'STL' stands for Surface Tesselation Language (or, depending on who you talk to, perhaps 'STereoLithography file' or 'Standard Transform Language file').
A tessellation is a gap-less, repeating pattern of non-overlapping figures across a surface. Any shape can be used; STL format uses triangles. This triangular mesh is most often derived from the surface - and only the surface - of a 3D CAD designed object. Meshes can also be derived from 'point clouds' generated through laser scanning.
In an STL file, each and every triangle 'facet' is defined in terms of the coordinates of its three vertices. In addition a normal vector is included so that the inside and outside of each facet are clear. The number of triangles is primarily a function of the size of the surface and the resolution of the tessellation. The higher the resolution, the greater the number of triangles and the closer the approximation to a true curved surface. STL files can be human-readable ASCII or binary encoded. Since each and every triangle is defined, ASCII STL files can become somewhat large and binary files are usually preferred.
STL Files Inefficient for High Resolution 3D Printers
Although STL files are capable of holding triangular meshes at whatever resolution is desired, they quickly become large and unwieldy. Take for example the STL for a simple, smooth surfaced, 10 cm sphere, generated in SolidEdge with 10 micron conversion tolerance, or precision (i.e. any point on any STL triangle can be no further than 10 microns from the 'true' surface of the original 3D CAD design). Why 10 cm and why 10 micron? 10 cubic cm build envelopes are common and 10 micron accuracy is possible with today's higher resolution 3D printers.

Triangular mesh for a 10cm sphere with 10 micron precision
At this resolution the triangular mesh for this simple sphere is composed of nearly 50,000 triangles. The STL file is almost 350,000 lines long and is more than 13MB in size:
000001 solid ascii000002 facet normal 4.206248e-002 -5.926095e-004 9.991148e-001000003 outer loop000004 vertex 1.402310e+000 0.000000e+000 4.998023e+001000005 vertex 1.401753e+000 -3.950587e-002 4.998023e+001000006 vertex 2.802404e+000 -7.898065e-002 4.992124e+001000007 endloop000008 endfacet...346537 facet normal 1.402451e-002 -1.975884e-004 9.999016e-001346538 outer loop346539 vertex 1.401753e+000 -3.950587e-002 4.998023e+001346540 vertex 1.402310e+000 0.000000e+000 4.998023e+001346541 vertex 9.184833e-015 0.000000e+000 4.999990e+001346542 endloop346543 endfacet346544 endsolid
Saving the STL with binary encoding does result in file size savings (2.4MB) but at the cost of human readability and increases in write and read times.
Reducing STL Precision Not an Option
As we see above, even the simplest objects can result in unwieldy STL files when generated at resolutions required by today's 3D printers. Triangle counts in the millions and file sizes greater than 100MB are not uncommon.
To reduce STL file complexity one can reduce the resolution but the results quickly become less than optimum. For example the mesh for the same sphere generated with 1mm (1000 micron) precision is two orders of magnitude less in both number of triangles (506) and file size (0.13MB). However, as can be seen below, the resulting facets are larger than the features possible even with today's lowest resolution 3D printers.

Triangular mesh for a 10cm sphere with 1mm precision
3D Printers Have Capabilities Which Cannot Be Specified in STL Files
In 1987 stereolithography machines could 3D print in single mono-chromatic materials at relatively low resolutions; STL files were more than up to the task. The number of triangles to describe a surface to the resolution it could be printed was manageable and other information was not needed.
However, printer manufacturers - existing and new - continued to innovate and machines which could print not only at

higher resolutions, but with different materials, colors, textures etc. were introduced. STL files contain only vertex and facet definitions. Not even units are a part of a standard STL file, let alone material or color definitions. As a result a number of 'extended' STL formats have been developed by OEMs and software companies to include the additional information which today's printers can work with. (See here and here for non-commercial examples.) Presently one is faced with an ineffective standard STL format and a mishmash of incompatible semi-custom STL formats.
STL has to go - or does it?
For nearly 25 tears the STL format has been the 3D printing workhorse. 3D CAD programs can easily export to it and 3D printer OEM software has been developed to read it before slicing and creating machine paths. But as machine resolutions have increased so has the size of STL files required to print high resolution objects – hundreds of megabytes describing millions of triangles are not uncommon. Further, machines with capabilities not possible to be described in the standard STL format have been developed. To describe object characteristics such as materials or colors, the STL format has been modified, most often on the OEM or support software side. So instead of adding these design features during the design phase (in 3D CAD programs) one is forced to add them later during the print phase. One would think the STLs would be shown the door.
However, although STLs are inefficient and incapable of describing a number of 3D printing features they are still the de facto industry standard. The triangular meshes they describe are particularly amenable to common slicing algorithms and 20+ years of use means that their shortcomings are well known and dealt with. Still, it's difficult to get passed the feeling that 'we've always done it that way' is one of the primary reasons the STL format is still around. But whether it's simple inertia or the fact that at least on a basic level it gets the job done, for the time being, the STL format will continue to reign as the king of 3D printing.
But what if it was possible to get 10 micron precision with a file similar to a 1000 micron precision STL file - and include material, color and metatdata? Next, STL 2.0 - AMF Format to Replace STLs?

3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 11:49

Tuesday, October 25. 2011

3 Reasons Why 3D Printing Is Still Expensive-Part 3-Service Bureaus

Tweet

In our last two posts we explored some of the reasons why the price to 3D print objects is still high - at least compared to traditional manufacturing technologies. In particular, we examined how the high price of low output equipment combined with materials that are expensive by both necessity and design, leads to additive manufacturing prices that are usually higher than traditional manufacturing.

For the final piece of the high price puzzle it is necessary to understand how 3D printing is typically provided: through 'service bureaus'. (Service bureaus are companies which provide business services for a fee. SAAS - software as a service - companies are good examples.)

Since 3D printing equipment has always been expensive, and the applications limited primarily to prototyping, only the largest companies could justify the investment in their own 3D printers. Still with a steady demand for 3D printing from smaller companies requiring prototyping services, new and existing companies evolved to offer 3D printing to a multitude of customers as 'service bureaus'. Over time, service bureaus have become a significant source for 3D printed parts. With service bureaus come two other factors which typically increase the price of 3D printed objects: overhead and pricing policies.Traditional Service Bureaus Have Significant Overhead3D printing is an automated process so the direct labor component for any particular part should be small. The indirect expenses that a company must spend in order to operate is usually lumped into overhead. This might include support staff, facilities, etc. The larger the company, the greater the overhead. These costs must be allocated to the various products which the company sells. (How that is allocated is a large part of cost accounting.) When you order a 3D printed part from some companies, you're in effect paying for the company picnic.Further, creating a print-ready STL file was not always as straightforward as it is now. Many companies needed dedicated personnel to take client-supplied 3D CAD files and produce printable files. This was - and apparently still is - built into the overhead of many 3D printing services.

In recent years however, producing print-ready STL files has become much easier. If good 3D CAD design practices are followed most design programs will export relatively 'clean' STL files. Even if these files are not fully print-ready they can often be 'fixed' by any of a number of free automated stl fixing services.What this means is that it is now possible for service bureaus to offer 3D printing services with substantially less overhead; it's simply easier to support the additive fabrication process now than it was, say, 5 or 10 years ago. Smaller service bureaus following a low overhead model can now compete with larger bureaus. And unless they have a significantly greater number of customers (thus benefiting from economies of scale) larger bureaus with extensive services will likely have larger overhead, and subsequently higher 3D printing prices.

Cost-Based Pricing Is Usually Lower Than Market-BasedIn setting prices companies generally have two options: pricing according to what the market will bear or pricing according to its costs plus a standard profit margin. Although most companies will use a hybrid of market-based and cost-based pricing, one or the other will typically take precedent. This could be due to factors such as the company's understanding of the market or plans for the future. In one company, maximizing per customer profit to eventually invest in other ventures might be the goal while in another it might be maximizing the number of customers (with lower cost-based prices).For example, 'traditional' 3d printing service bureaus often see rapid prototyping as the 3d printing market. This is a B2B market with relatively cost-insensitive customers - especially since even with large profit margins 3D is competitive with conventional prototyping. These companies will therefore lean towards the market pricing model; they will price as high as their customer base will bear.On the other hand, companies which view the 3d print market as perhaps broader may lean towards a cost-plus pricing. In this view, 3d printing can and should have many applications. As much as possible it should compete with traditional manufacturing. Some customers might be particularly cost-sensitive and prices should therefore be as low as possible. This typically results in lower, cost-based pricing.

*

The final price of a 3D printed part is affected by service bureau overhead and pricing choices. Larger service bureaus may have higher overhead than smaller service bureaus. Further, service bureaus which view the 3d print market as fairly established and cost-insensitive will likely follow a higher priced market-based pricing scheme; those that view the market as broad and expanding will likely follow a lower priced cost plus scheme.For 3D print customers the best advice is to choose your service bureau carefully by comparing technology, delivery and most especially, prices. 3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 14:21

Wednesday, October 12. 2011

3 Reasons Why 3D Printing Is Still Expensive - Part 2/3 - Materials

Tweet

In our previous post we explored why 3D printing prices - despite having dropped in recent years - are still high. In particular, we examined one major factor, the initial capital investment required to acquire 3D printing equipment and how that gets built into the price of each part to come out of that equipment.
In this post, we'll examine the second major factor keeping 3D printing prices high, the cost of the raw materials.

3D Print Materials Need Special Preparation
It is now possible to print in a large variety of materials from plastics to metals and even glass. Although some of these materials begin life as commodity items they usually require additional preparation before they can be used in particular machines. ABS must be formed into precise spools for fused deposition machines, metals and glass must be in a uniformly sized powder for sintering or other technologies. Other materials, such as Objet's photopolymers must be precisely formulated as specialty chemicals.
In other words, due to technological constraints (and perhaps by design - more about that below) the materials used in additive fabrication are usually value-added products before they even enter the machine. This additional cost ends up in the price of the 3D printed part.
3D Print Materials Are Expensive Because the Manufacturers Want It That Way
Adding to the already more expensive materials, is what is apparently the chosen business model of the major 3D printer manufacturers: emulation of the 2D printing business model and in particular, 2D inkjet printing.
In this business model, also known as 'freebie marketing' a manufacturer sells both hardware and the materials (consumables) that can be used in that same hardware. Typically the hardware is sold at a low profit margin and the consumables are sold at a higher profit margin. Think of how inexpensive an inkjet printer seems and how expensive the little ink cartridges are. Over the life of the hardware, the manufacturer hopes to sell you as much or more in consumables as you originally paid for the hardware itself. Since the profit margin is higher on the consumables, they plan on making more profit there also. This is the business model that most major 3D printer manufacturers seem to be following (with some understandable exceptions). Their printers - whether by design or otherwise - can only use materials which they provide. Especially on the lower end of the spectrum, these printers have been coming down in price. Not surprisingly, the industry has NOT seen an equivalent decrease in the price of the materials to be used in those printers.
Should one begrudge the manufacturers for running their business in this way? Of course not. Still though, other models are possible. One has to wonder how different the automobile industry would have been had Henry Ford sold cars that would only run on (expensive) Ford fuel. If Ford and others had been able to make a tidy profit simply by selling fuel, would we have seen the same innovation in the hardware, the cars themselves? Might we still be driving Model T's? Could 3D printer manufacturers be going down this road?
Finally it should be noted that this only applies to major manufacturers. On the lower end and on periphery there are plenty of efforts aimed at producing affordable machines which use materials as close to commodity prices as possible. So far machines using commodity materials cannot rival those using proprietary or exclusive materials but the gap is growing ever narrower. Further, despite the threat of lost warranties, some are jailbreaking their 3D printers with 'hacked' materials, i.e. non-manufacturer supplied or approved, cheaper materials. And at least one major manufacturer is attempting to change the entire materials paradigm with 'digital materials'.
Unfortunately though, at this time, materials to be used in commercial grade 3D printers tend to command a significant price premium as compared to commodity materials used in more traditional manufacturing. This is due to both additional preparation costs and manufacturer business models. This premium ends up in the price of the part and is one of the major factors why 3D printing is still expensive.
*
Next, I'll look at the final piece of the puzzle as to why 3D printing prices are still high: service bureau overhead and

pricing policies.
3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 14:30

Friday, October 7. 2011

3 Reasons Why 3D Printing Is Still Expensive - Part 1 - Equipment

Tweet

Let's be honest, 3D printing is pretty darn cool. The possibilities seem endless; one day we'll all sit at home, download designs, and additively fabricate everything we need using our home 3d printer. Well that's the theory, at least.
The reality is that 3D printing, while still cool, has been around for 25 years, is still somewhat limited in it's capabilities and is usually more expensive than traditional manufacturing technologies. But we've been hearing how affordable 3D printing has become, right? Yes we have. If you compare the current prices to those even ten years ago, 3D printing is cheaper. Compared to conventional methods of rapid prototyping (the most common use for 3D printers) 3D printing is often cheaper. If a product is fully or semi-customized, 3D printing might be cheaper. However, for true paradigm-busting applications, such as on-demand manufacturing of standard parts (not everything need or should be customized) 3D printing is most often more expensive than conventional manufacturing. A significant run of 3D printed plastic parts just won't compete on price with injection molding. Ditto laser-sintered stainless steel versus traditional machining. If you want 1 or maybe 50 pieces 3D printing might be your best option, even for standard parts. Beyond a certain number however, more traditional methods will almost always be lower cost. Why is this? The finished product is the same, the materials seem the same; why should there be a significant price difference?

In short, three major factors work to raise 3D printing prices. Some of these are unavoidable, some are by choice but all are serving to limit the development of novel applications for 3D printing. In this post I want to examine one such factor: the price of the equipment.
*
3D Printers Are Expensive
A primary driver of 3D printing prices is the initial cost of the equipment. For commercial-grade equipment this capital investment starts in the tens of thousands of dollars and can go as high as hundreds of thousands of dollars - or more - for a single machine. These are precision engineered and manufactured machines that are produced in limited quantities. Unlike inkjet printers, microwave ovens and flat screen televisions, they haven't been around long enough or sold in sufficient quantities to have been designed to be affordable; they are designed to work, sometimes regardless of cost. One look inside will convince you of this. Whereas a peek inside an inkjet printer will reveal primarily plastic parts, that of a 3D printer will reveal significant aluminum and steel. Although less expensive and homebrew versions of 3D printers do exist their tolerances and accuracy, while having improved significantly, cannot match commercial-grade machines.
Equipment is Paid for Over the Life of the Machine
To justify such an investment businesses need to at least recoup that investment over the life of the machine. For a simple example, suppose a company spends $100k on a machine that it thinks will have a lifetime of 5 years. That machine better generate at least $20k per year in revenue (this is equivalent to straight-line depreciation). If that company uses the machine to fabricate one item per week, for 52 weeks per year, each item will need to have incorporated into the cost nearly $400 just to account for initial capital investment.Even if the machine is run once a day, seven days a week, each item will still have more than $50 in equipment costs. Compare this, for example, with a similarly priced commercial grade CNC router that can cut into parts, tens or even hundreds of sheets of 4' by 8' plywood per day. Per part equipment cost might only be pennies.
Equipment Cost Contribution to 3D Printed Part Prices (Equipment Price: $100k, Lifetime: 5 Years) 3D Printed Parts Per Week $/Part   1 $385   10 $38.50   50 $7.70   100 $3.85
Part Production Capabilities Are Limited
Further exacerbating this is the fact that machine production capability is limited. Build envelopes are relatively small and build times are relatively long. One cubic foot build envelopes are the common so that $400, or $50, or whatever the per part contribution is, ends up being priced into something you can very likely hold with one hand. And unlike conventional machinery with similar build envelopes (e.g. commercial mills) build times can take hours, severely limiting the actual number of parts a machine can produce, even when running 24/7.
Prices Will Drop When More Applications Exist
3D printing equipment prices will likely come down only when the volume of printers sold can justify the development of less expensive methods and materials of manufacture. There seems to be little incentive for existing major manufacturers to truly push down the hardware price point to increase sales, if more applications (i.e. use of consumables) do not exist. Although bottom up efforts through open source and inexpensive printer options may help to

economize hardware design and manufacturing, there will need to be broader market applications for 3D printing before equipment prices come down significantly.
*
3D printing is an exciting technology with great potential. However, the high cost compared to traditional manufacturing limits the development of new applications. Some of the factors to blame for high cost - such as the high cost of manufacturing precision machinery in limited quantities - are perhaps unavoidable. Others, maybe not.
Next: 3 Reasons Why 3D Printing Is Still Expensive - Part 2 -Materials

3D Additive Fabrication, Inc. (3dAddFab) is a start up company located in Colorado, USA. 3dAddFab provides high quality 3D printing that is easy to price and order, at a lower cost than existing fabricators.

Posted by 3DAddFab at 12:35