

Standard Tessellation Language

## STL File Format (3D Printing) – Simply Explained

Written by *Dibya Chakravarty*  
Edited by *Francisco de la Tejera*

### Editor's note:

No file – no print. We all know that.

All 3D printers need a CAD file to start from. There are several forums and discussions about 3D printing, but rarely about the source.

It is complicated to understand 3D printing if its origin is not clear. Knowing the concept, principle, advantages and limitations of an STL encoded file helps anyone involved into 3D printing one way or another; from engineers and designers to operators and manufacturers.

3D print is all about layers. We can say that 3D printing (or Additive Manufacturing) is the progression of two dimensions stacked or added one after other to create three dimensional objects. Therefore, a layer as the first step is virtually 2D information.

During my almost 10 years involved in 3D printing and 20+ years in CAD, manufacturing and engineering, I have helped people to get immerse into the fascinating world of the art and technique of stacking thin layers of material, that we call 3D printing. Coded tessellated triangles will be simply explained.

## 1. What is an STL File?

In a nutshell, an STL file stores information about 3D models. This format describes only the surface geometry of a three-dimensional object without any representation of color, texture or other common model attributes.

These files are usually generated by a computer-aided design (CAD) program, as an end product of the 3D modeling process. “.STL” is the file extension of the STL file format.

The STL file format is the most commonly used file format for 3D printing. When used in conjunction with a 3D slicer, it allows a computer to communicate with 3D printer hardware.

Since its humble beginnings, the STL file format has been adopted and supported by many other CAD software packages, and today is widely used for rapid prototyping, 3D printing, and computer-aided manufacturing. Hobbyists and professionals use it alike.

## 2. What does the file extension .STL stand for?

The true meaning of the file extension .STL has been lost to the mists of time.

It's widely believed to be an abbreviation of the word **ST**ereo**L**ithography, though sometimes it is also referred to as "Standard Triangle Language" or "Standard Tessellation Language".

## 3. How does the STL file format store a 3D model?

The main purpose of the STL file format is to encode the surface geometry of a 3D object. It encodes this information using a simple concept called "tessellation".

### 3.1 Tessellation

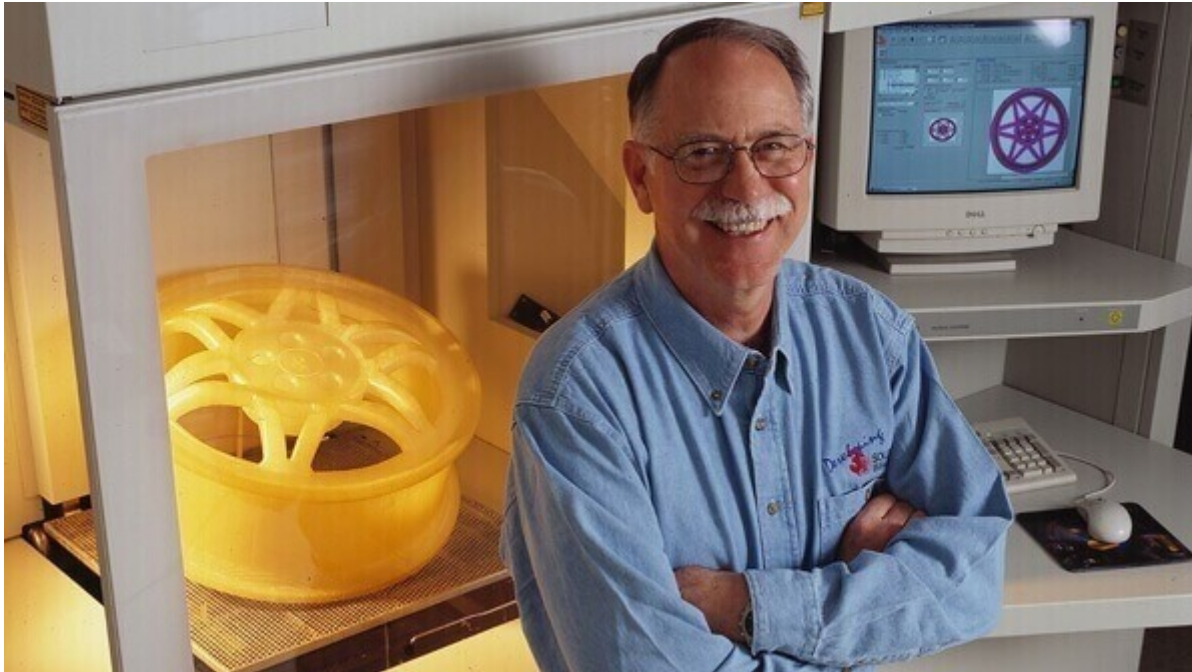
Tessellation is the process of tiling a surface with one or more geometric shapes such that there are no overlaps or gaps. If you have ever seen a tiled floor or wall, that is a good real life example of tessellation.



The tiled wall and floor are simple real life examples of tessellation

### 3.2 The invention of the STL file format: exploiting tessellation to encode surface geometry

Back in 1987, Chuck Hull had just invented the first stereolithographic 3D printer, and **The Albert Consulting Group for 3D Systems** were trying to figure out a way to transfer information about 3D CAD models to the 3D printer. They realized that they could use tessellations of the 3D model's surface to encode this information!

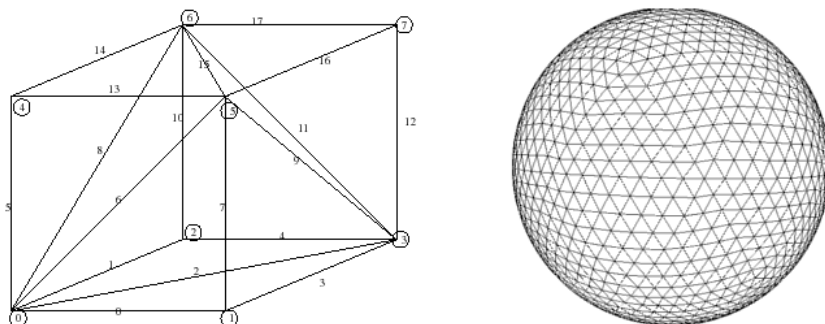


Chuck Hull - Inventor of the first 3D printer and Founder/Former CEO of 3D Systems

The basic idea was to tessellate the 2 dimensional outer surfaces of 3D models using tiny triangles (also called “facets”) and store information about the facets in a file.

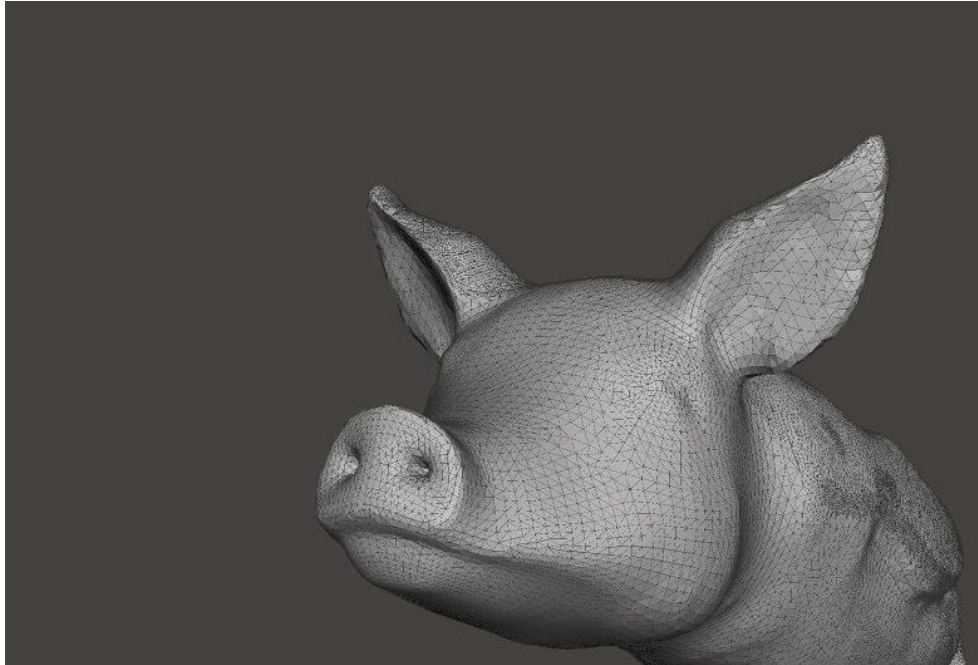
Let's look at a few examples to understand how this works. For example, if you have a simple 3D cube, this can be covered by 12 triangles, as shown in the image below. As you can see, there are two triangles per face. Since the cube has six faces, it adds up to 12 triangles.

If you have a 3D model of a sphere, then it can be covered by many small triangles, also shown in the same image.



Tessellations of a cube and a sphere

Here is another example of a very complicated 3D shape which has been tessellated with triangles.



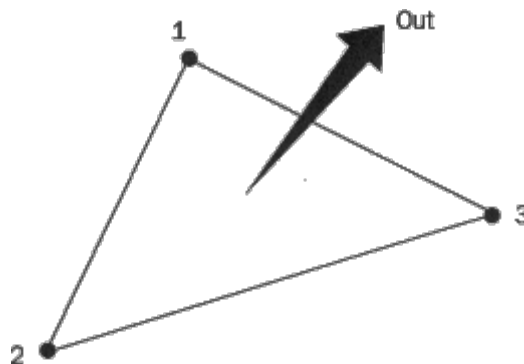
Tessellation of a 3D pig (source : i.materialize)

The Albert Consulting Group for 3D Systems realized that if they could store the information about these tiny triangles in a file, then this file could completely describe the surface of an arbitrary 3D model. This formed the basic idea behind the STL file format!

#### 4. How does an STL file store information about facets?

The STL file format provides two different ways of storing information about the triangular facets that tile the object surface. These are called the *ASCII encoding* and the *binary encoding*. In both formats, the following information of each triangle is stored:

1. The coordinates of the vertices.
2. The components of the unit normal vector to the triangle. The normal vector should point outwards with respect to the 3D model.



An STL file stores the co-ordinates of the vertices and the components of the unit normal vector to the facets

## 5. Special rules for the STL format

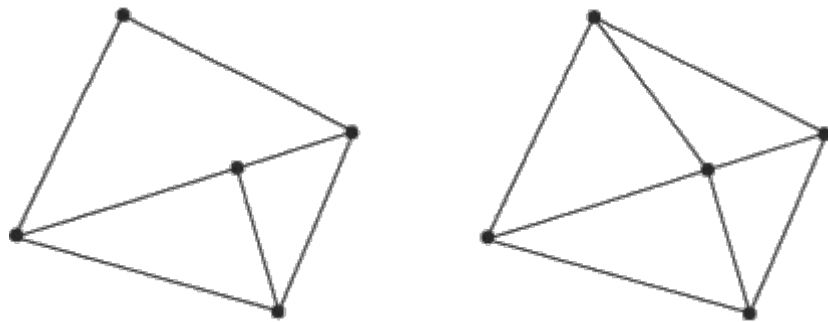
The STL specification has some special rules for tessellation and for storing information.

### 5.1 The vertex rule

*The vertex rule states that each triangle must share two vertices with its neighboring triangles.*

This rule is to be respected when tessellating the surface of the 3D object.

Here's an example of a valid and invalid tessellation, according to this rule. The figure on the left violates this rule and is an invalid tessellation, while the figure on the right is conformant and a valid tessellation.

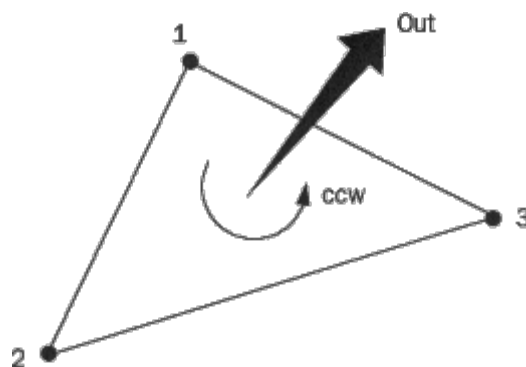


Vertex rule for STL files: The figure on the left is an invalid tessellation, while the figure on the right is acceptable.

### 5.2 The orientation rule

*The orientation rule says that the orientation of the facet (i.e. which way is “in” the 3D object and which way is “out”) must be specified in two ways.*

First, the direction of the normal should point outwards. Second, the vertices are listed in counterclockwise order when looking at the object from the outside (*right-hand rule*).



The orientation of each facet is specified in two ways: by the direction of the normal vector and by the ordering of the vertices



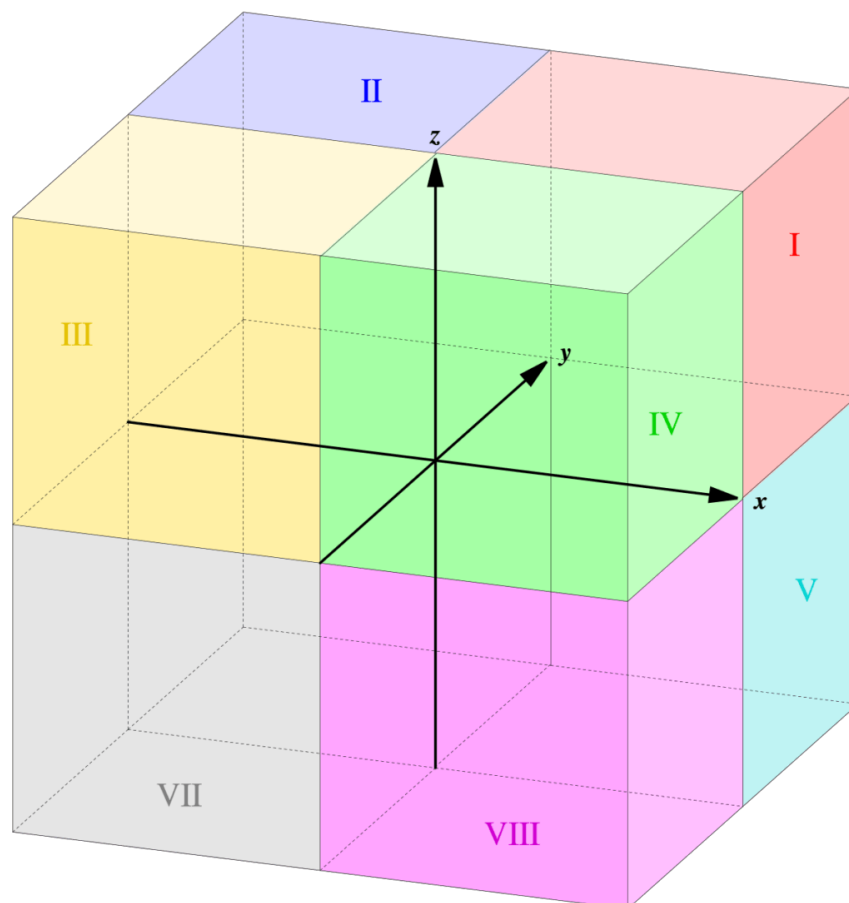
This redundancy exists for a reason. It helps ensure consistency of the data and spot corrupt data. A software can, for example, calculate the orientation from the normal and subsequently from the vertices and verify whether they match. If it doesn't, then it can declare the STL file to be corrupt!

### 5.3 The all positive octant rule

*“The all positive octant rule says that the coordinates of the triangle vertices must all be positive.”*

This implies that the 3D object lives in the all-positive octant of the 3D Cartesian coordinate system (and hence the name).

The rationale behind this rule is to save space. If the 3D object was allowed to live anywhere in the coordinate space, we would have to deal with negative co-ordinates. To store negative co-ordinates, one needs to use *signed floating point numbers*. Signed floating point numbers require one additional bit to store the sign (+/-). By ensuring that all coordinates are positive, this rule makes sure that we are able to use *unsigned numbers* for the coordinates and save a bit for every coordinate value we store.



Octant I (red) is the all positive octant

## 5.4 The triangle sorting rule

*The triangle sorting rule recommends that the triangles appear in ascending z-value order.*

This helps Slicers slice the 3D models faster. However, this rule is not strictly enforced.

## 6. How is an STL file 3D printed?

For 3D printing, the STL file has to be opened in a dedicated slicer. What's a slicer? It's a piece of 3D printing software that converts digital 3D models into printing instructions for your 3D printer to create an object.

The slicer chops up your STL file into hundreds (sometimes thousands) of flat horizontal layers based on the settings you choose and calculates how much material your printer will need to extrude and how long it will take to do it.

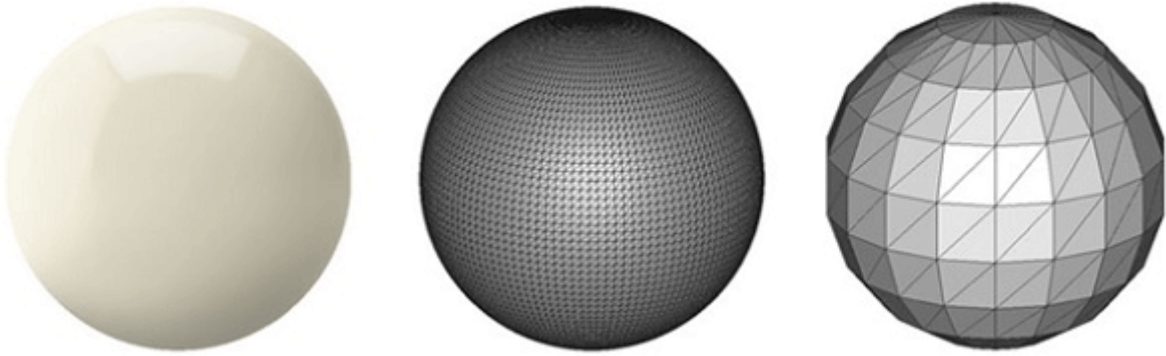
All of this information is then bundled up into a GCode file, the native language of your 3D printer. Slicer settings do have an impact the quality of your print so it's important to have the right software and settings to get you the best quality print possible.

Once the GCode has been uploaded to your 3D printer, the next stage is for those separate two-dimensional layers to be reassembled as a three-dimensional object on your print-bed. This is done by depositing a succession of thin layers of plastics, metals, or composite materials, and building up the model one layer at a time.

## 7. Optimizing an STL file for best 3D printing performance

The STL file format approximates the surface of a CAD model with triangles. The approximation is never perfect, and the facets introduce coarseness to the model.

The 3D printer will print the object with the same coarseness as specified by the STL file. Of course, by making the triangles smaller and smaller, the approximation can be made better and better, resulting in good quality prints. However, as you decrease the size of the triangle, the number of triangles needed to cover the surface also increases. This leads to gigantic STL file which 3D printers cannot handle. It's also a pain to share or upload huge files like that.



The perfect spherical surface on the left is approximated by tessellations. The figure on the right uses big triangles, resulting in a coarse model. The figure on the center uses smaller triangles and achieves a smoother approximation (source: i.materialize)

It is therefore very important to find the right balance between file size and print quality. It does not make sense to reduce the size of the triangles ad infinitum because at some point your eye is not going to be able to distinguish between the print qualities.

Most CAD software offer a couple of settings when exporting STL files. These settings control the size of the facets, and hence print quality and file size. Let's dig into the most important settings and find out their optimum values.

## 9. Are there any alternatives to the STL File Format?



The STL file format is not the only format used in 3D printing. There are over 30 file formats for 3D printing. Most important is the OBJ file format, which can store color and texture profiles. Another option is Polygon file format (PLY), which was originally used for storing 3D scanned objects.



## 10. Advantages and disadvantages of using STL file format over other file formats

Since there are many 3D printing file formats, the obvious question is: which one should you use for your prints? The answer, as it turns out, depends a lot on your use case.

### 10.1 When *not* to use an STL file

As we saw earlier, the STL file format cannot store additional information such as color, material etc. of the facets or triangles. It only stores information about the vertices and the normal vector. This means that if you want to use multiple colors or multiple materials for your prints, then the STL file format is not the right choice. The OBJ format is a popular format enjoying good support which has a way to specify color, material etc. Therefore, this is the right choice for this task.

### 10.2 When to use an STL file

On the other hand, if you want to print with a single color or material, which is most often the case, then STL is better than OBJ since it is simpler, leading to smaller file sizes and faster processing.

### 10.3 Other advantages of the STL file format

**Universal:** Another big advantage of the STL file format is that it is universal and supported by nearly all 3D printers. This cannot be said for the OBJ format, even though it enjoys reasonable adoption and support as well. The VRML, AMF and 3MF formats are not widely supported at this point of time.

**Mature ecosystem:** Most 3D printable models you can find on the internet are in the STL file format. The existence of this ecosystem, combined with STL-based software investments made by 3D printer manufacturers, has given rise to a large user-base that's heavily invested in the format. This means there's plenty of third party software dealing with STL files, which is not the case with the other file formats.

### 10.4 Some disadvantages of the STL file format

There are some glaring disadvantages to using STL as well. As the fidelity of printing processes embraces micron-scale resolution, the number of triangles required to describe smooth curved surfaces can result in massive file sizes. It's also impossible to include metadata (such as authorship and copyright information) in an STL file.

### 10.5 Verdict

If your 3D printing needs are simple, then perhaps there is no reason to move away from the STL file format. However, for more advanced prints using multiple material and color, it is perhaps advisable to try the OBJ or other available formats.

## 11. Conclusion

You made it to here! Now you can praise yourself as an STL expert.

STL files are so far the most widely adapted language for 3D printers.

Only time will tell for how long STL format will continue to be used. As 3D printing capabilities continue to increase, the information available (and manageable) on STL formats may need to be updated to future 3D printing super-features.

License: This text is licensed under a Creative Commons Attribution 4.0 International License.

<https://all3dp.com/what-is-stl-file-format-extension-3d-printing/>

<https://creativecommons.org/licenses/by/4.0/>