

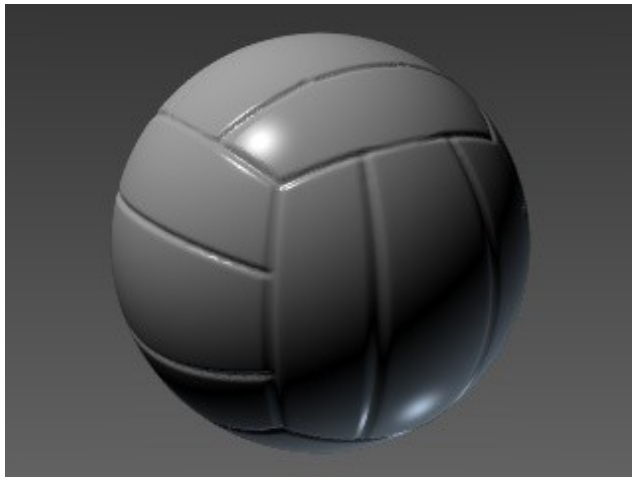


## Introduction

### What Is Displacement Mapping?

Displacement Mapping is a modeling technique that uses a grayscale image to modify the corresponding vertices of a mesh. A black pixel results in no displacement, a white pixel results in full displacement, and gray pixels are in between. The result is an actual bumpy or elevated surface.

Figure 1



Blender supports displacement mapping of planar meshes only. This feature is great for making terrains, rocks, mountains, and even bathroom tiles. However, if you want to displace the surface of a sphere, cylinder, or torus, planar mapping will not be sufficient. To get around this, I have created several python scripts that enable Blender to support spherical, cylindrical, and toroidal displacement mapping. Note: These scripts require Blender version 1.8 through 2.04 and can be [downloaded here](#).

By the end of this tutorial, you will be able to create the Volley Ball as shown in Figure 1.



Introduction — 1 2 3 4 5 6 7 8 9 →

## Step 1:

### The Python Path

The first thing you should do after installing Python for your platform is to set the PYTHONPATH environmental variable properly, if it has not already been configured for you.

For Windows type this on the command line and add it to autoexec.bat:

```
set PYTHONPATH=c:\Python\DLLs;c:\Python\Lib;.
```

For Linux type this on the command line and add it to .bashrc:

```
export PYTHONPATH=/usr/lib/python1.51/:.
```

The paths are separated by semicolons for the Windows platform and colons for Linux. I add a period at the end so that the current directory is searched as well. This is absolutely necessary if a python script uses modules from a script that is located in the current directory. Please note that the example paths listed above may be different than those on your computer.



Introduction — 1 2 3 4 5 6 7 8 9 →

## Step 2:

### Loading And Running Python Scripts

If you have only one main window, split it in two, move the mouse over the one on the right to make it active, and change it to a text window by pressing s|F11. Now press a|o, and select a python script with the middle mouse button. This simply loads it into the text window. When you are ready to execute the script, simply press a|p (making sure the text window is active).



Introduction — 1 2 3 4 5 6 7 8 9 →

### Step 3:

#### The Input Mesh

There are three strict requirements for the input mesh:

1. it must be on the xy-plane
2. it's center must be located at the origin
3. it must be 2 units long by 2 units wide

The easiest way to satisfy these requirements is to make sure you are in Top View (Pad7), move the cursor to the origin (s|c), and add a default 32x32 grid object. Subdivide it 3 times (or as many times as necessary to get the desired level of detail).

Ideally, the number of vertices in the mesh should be equal to the number of pixels in the image. If the image resolution is 100x100, the mesh should have at least 10,000 vertices. Keep in mind that the maximum number of vertices for a single mesh is 65,000, so using an image with a resolution of 1024x1024 will not yield better results than one with a resolution of 256x256.



Introduction — 1 2 3 4 5 6 7 8 9 →

## Step 4:

### The Grayscale Image

You can create your own custom image using a paint program, download one from the web, or use a special purpose utility. Just make sure it's a grayscale image that is in a format that Blender can read.

An excellent utility for creating the input image is gforge v1.3a, by John Beale. It has the advantage of creating very realistic terrains for use with planar mapping. It also creates a seamless image, which is important if you plan to make a tiling mesh with linked duplicates.



Introduction — 1 2 3 4 5 6 7 8 9 →

## Step 5:

### The Material & Texture Settings

Press F5 and add a new material, then press F6 and add a new texture. Click on the 'Image' button and load the grayscale image that will be used as a displacement map. We usually don't want this image showing up in the color channel, so go to the 'Material' buttons again and make sure that the 'Col' button is not pressed in. In editmode and with all vertices selected, press F9 and click on the 'Noise' button.

Technical Note: Each click moves the vertices (that correspond to a white pixel) exactly 0.05 "Blender units" in the direction of the positive z-axis. You may need to click on the 'Noise' button several times to get more "height."



Introduction — 1 2 3 4 5 6 7 8 9 →

## Step 6:

### Displacing the Mesh

So now we have a planar-displaced mesh and we can easily turn it into a spherical, cylindrical, or toroidal shape. Just make sure the planar mesh is selected, press tab to leave editmode, and then run one of these scripts to get the desired shape:

`displace_cyl.py`, `displace_sph.py`, or `displace_tor.py`.

Note that the scripts overwrite the original mesh! You may want to work on a copy in case you need to make changes or if you want to try one of the other scripts.



## Step 7:

### Spherical Example

So now that the basics are out of the way, let's go through some real examples. Here we will use spherical mapping to create a volley ball. Create your input mesh as described in Step 3. You should already have your material and texture added (Step 5). All you need now is to load the input image of an unwrapped volley ball as shown in Figure 2. Finally, run the `displace_sph.py` script.

You will notice that the grooves on the surface of the ball are too deep even though we clicked on the 'Noise' button only once. Fortunately, this can be fixed very easily. Select all vertices in edit mode, press the 'To Sphere' button, and enter a percentage value of 60%. Leave editmode and press the 'Set Smooth' button. Now the grooves looks more like those found on a regulation size volley ball. The results should look similar to the image in Figure 3.

Figure 2

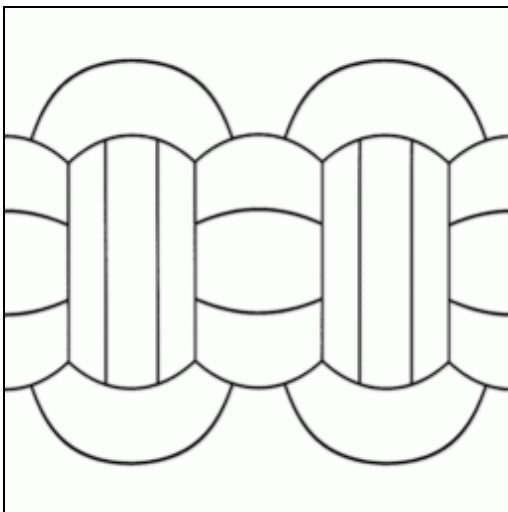
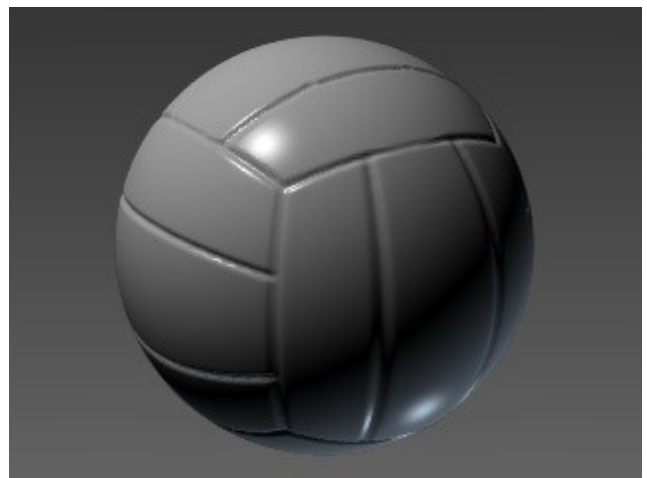


Figure 3







Introduction — 1 2 3 4 5 6 7 8 9 →

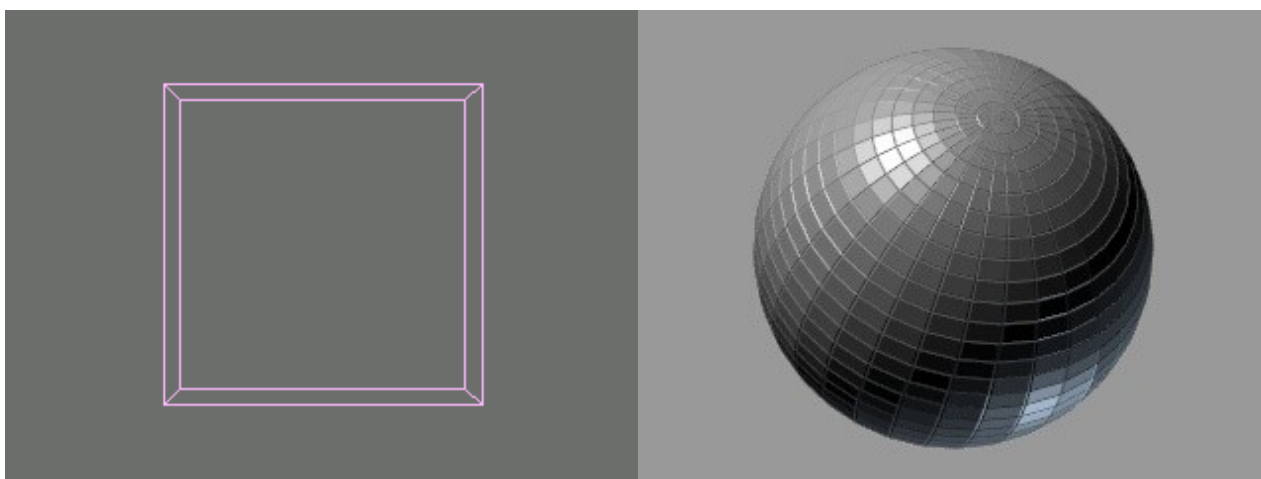
## Step 8:

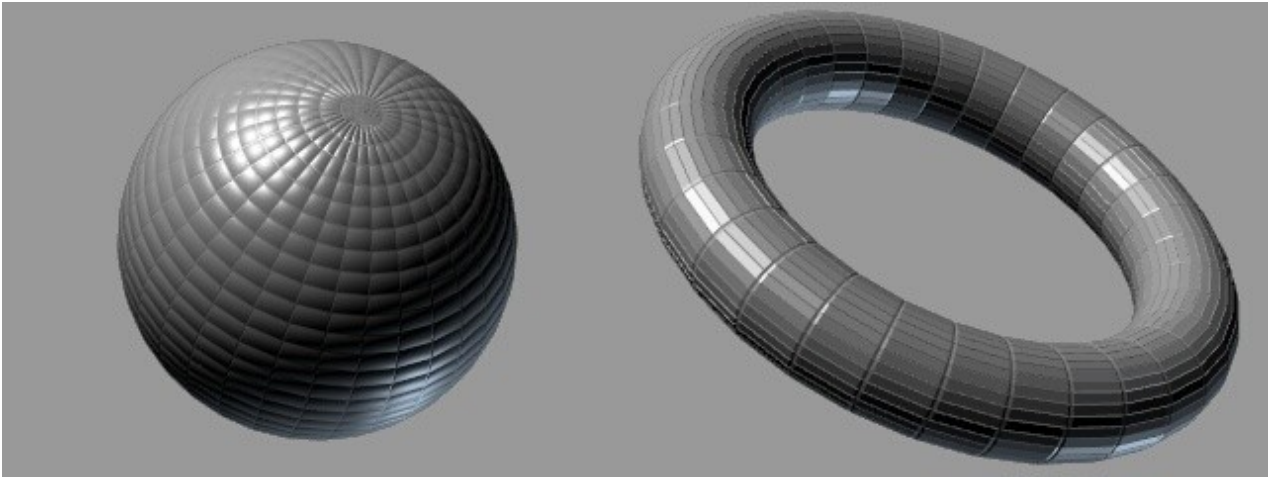
### Other Examples

The initiated reader will soon realize that these scripts really don't need an image in order to be useful. If you run them on a flat mesh, you will just end up with a smooth sphere, cylinder, or torus. This is a rather boring result. However, it proves an important point: if you create your own special mesh, like the one shown at the top left corner, some interesting objects can be created.

Note that the mesh shown is just a very small single "tile" and that you must make several copies along the x and y axes. A quick and easy way to do this is to parent this single tile to a subdivided plane, then simply press the 'Dupliverts' button to instantly make the copies. Now select your subdivided plane and press s|c|a to convert the virtual copies into real objects. Select all of the tiles and join (c|j) them together.

An important step that I often overlook is to make sure that the center of the mesh is located at the origin. This can be done by pressing s|c and then 'Centre Cursor.' Also make sure you press c|a to apply the size and rotation of this object to the individual vertices. This will allow the newly created mesh to satisfy the requirements in Step 3 and must be done prior to running the scripts.







Introduction — 1 2 3 4 5 6 7 8 9 →

## Step 9:

### Conclusion

In conclusion, displacement mapping is a good modeling technique that can be used to create a variety of different objects. Keep in mind that you are not limited to simple geometric shapes. For example, you could use a paint program to draw a face and then apply cylindrical mapping to a highly subdivided mesh, which will probably need to have the maximum number of vertices (65,000) in order to show enough detail. Depending on how good your painting skills are, you could end up with a very nice model of a human head.

There are other good uses for the technique and all you have to do is use your imagination. With a little creativity, you will be able to create almost anything.

If you create any interesting renderings with these scripts, please let me know. Also, feel free to contact me if you have any comments or questions.