

CS 284 Project:
Escher On Genus Four Object

Kerstin Keller

December 11, 2009

Introduction

The goal of my project was to find a nice looking Escher-like texture to put onto the genus four object.

Mainly, my project consists of two parts: One part consists of finding an appropriate texture, the other part consists of developing a program that allows the user to interactively modify and adjust the texture that is put onto an object. Therefore this report will be divided into two sections, one concerning the texturing and one concerning the program itself.

Texturing the genus four object

I started out from the genus four object that we developed for our homework in class and looked for an Escher like textures to put onto the surface. Escher, a Dutch graphics artist who lived from 1898 - 1972, made symmetry drawings that repeat itself and can cover an arbitrary sized, flat surface. On the homepage one can find various of Escher symmetry drawings, see Fig. 1.

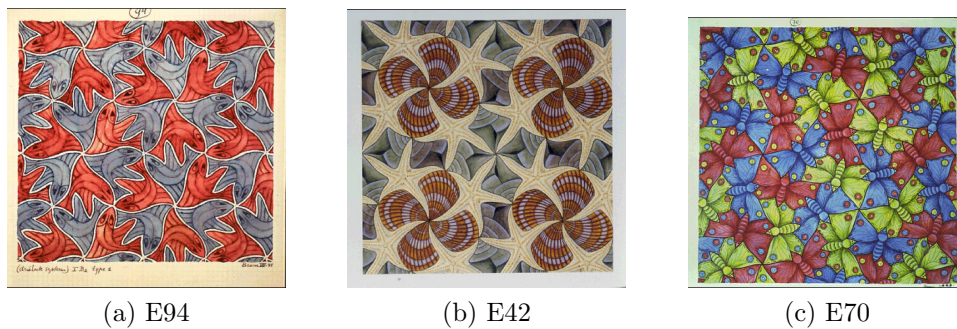


Figure 1: Some Escher patterns that might be used as texture

The pattern from Fig. 1 (a) seems to be very appealing to map onto the surface since it basically only consists out of only two different triangles that can be put together (see Fig. 2). A first constraint for using this particular pattern as a texture is that all vertexes have even valences.

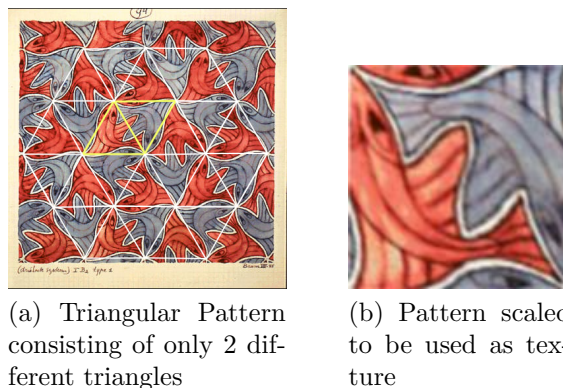


Figure 2: Escher Pattern based on triangles

The result of mapping the texture onto my original genus four object can be seen in Fig. 3. The texture wraps around the object nicely, however the result does not look appealing. Since the original triangles of the original mesh have uneven side lengths, after subdivision, the pattern gets deformed to an extent such that the user can't tell that the pattern resembles fish any more.

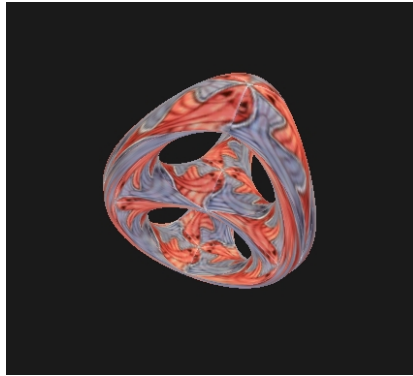


Figure 3: First Escher pattern applied to genus four object

In order to receive a texture that is not distorted after subdivision, the original mesh needs to consist of more or less equilateral triangles. In order to reproduce the object with better shaped triangles, more triangles need to be used (in this case 168 triangles compared to 72 triangles of the original mesh). However, it is hard to preserve the appearance of the subdivided surface of the original when using more triangles. The arms of the subdivided object become thinner and the object becomes less round (see Fig. 4).

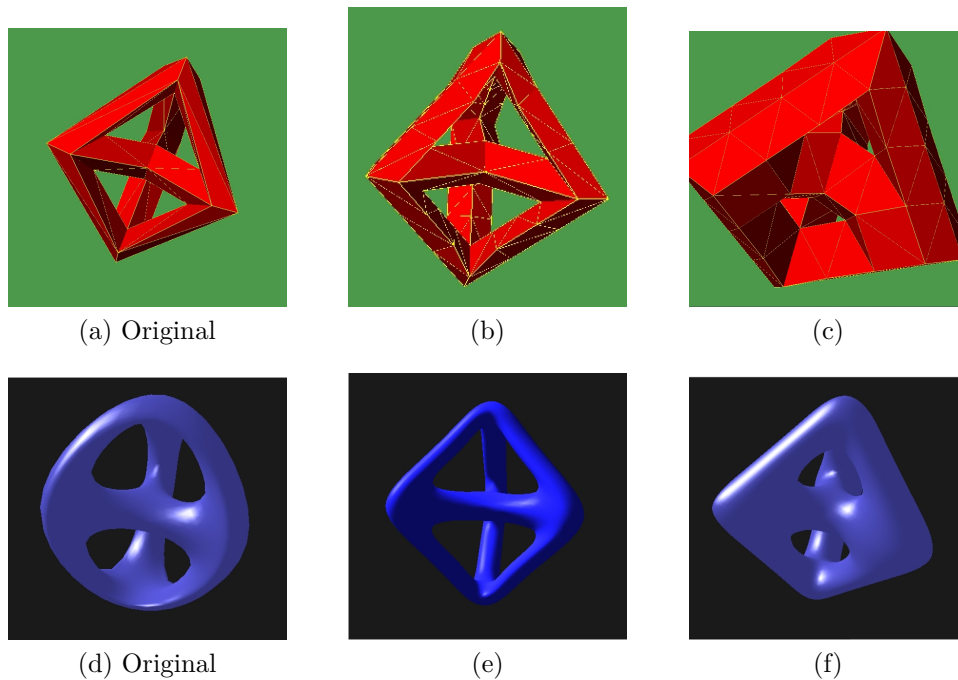


Figure 4: Altered meshes compared to the original one

Putting the fish texture onto the new surface works nicely, it wraps around the genus four object without any problems. Results for this and for a butterfly pattern (which is three colored but has essentially the same texture properties as the fish) can be seen in Fig. 5.

I have also used my program to assign a given texture to a given genus three object, the results can be seen in Fig. 6.

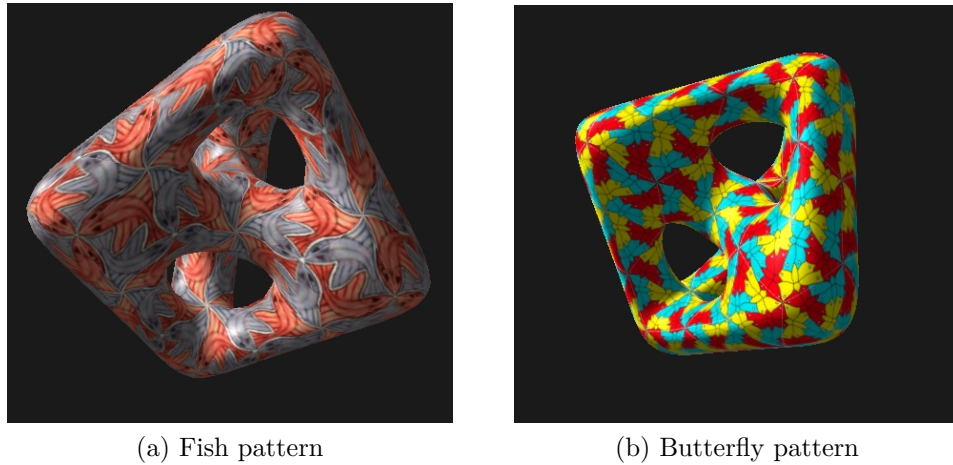


Figure 5: Textured genus four object

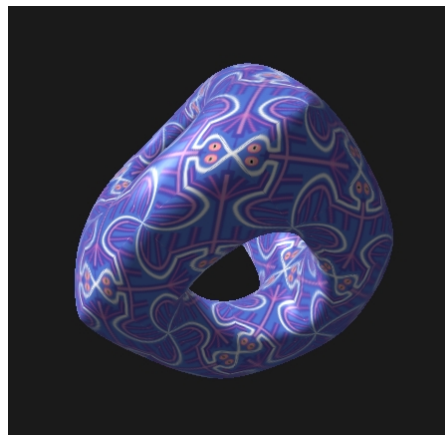


Figure 6: Textured genus three object with bird pattern

The attempt made by Prof. Sequin to reduce the number of triangles posed a problem for texturing. Although the valence of each vertex is even, I was not able to fully cover the objects with the fish or the butterfly texture. Clearly there seem to be another constraints other than the valence of all vertices that determines if an object can be textured with the given texture. In this case, when wrapping the texture along the sides of the object, it would not come back together nicely (see Fig. 7. The butterfly pattern would probably need at least four colors and more than two different tiles, but I have not figured out how many colors and different tiles one would need.

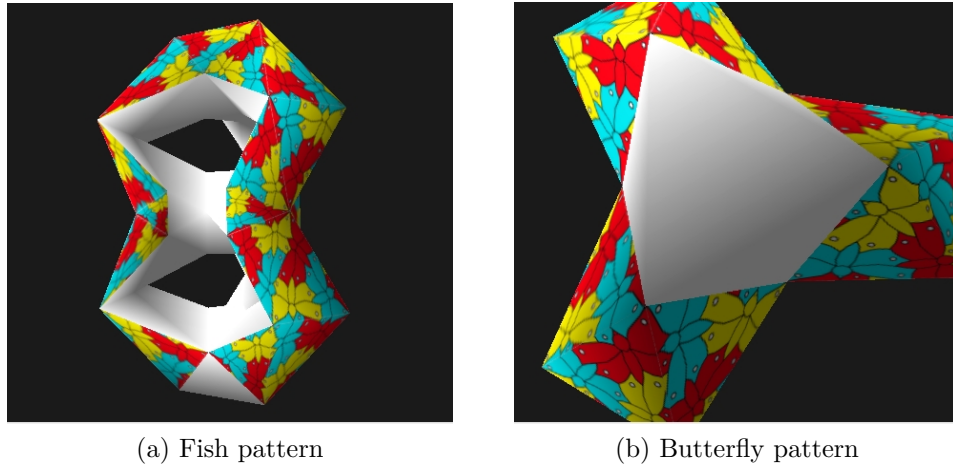


Figure 7: The tiles do not come together nicely at the bottom

The program

The goal of my program is to make assignment of texture to a given surface as easy as possible.

After starting the program, the user needs to specify a *.txt file to load the object. That file contains the total number as well position of all vertices and then the total number and vertex indices for all faces. All in all it is the output of a slide file where the additional information about genus, etc. has been deleted.

```
(vertices 78
(v -0.017321 -0.030000 0.090000)
(v -0.003481 -0.053971 0.060000)
...
(v 0.000000 0.000000 -0.120000)
)
(triangles 168
(t 0 42 17)
(t 0 55 42)
...
(t 21 15 16)
)
```

Figure 8: Example for object input file

The user can then load a texture file into the application. This texture file is a *.txt file that contains information about the textures used as well as the texture coordinates that have been assigned to each triangle. The file begins with

```
(images #of_images
```

followed by a list of relative paths to the files of all textures that will be used on this object. The syntax for this is

```
(f relative_path ).
```

This section is concluded by a right bracket).

The second part of the file assigns the texture coordinates for every face. It starts by

```
(triangles #of_triangles
```

If the number of triangles indicated and the number of triangles of the loaded object do not correspond, the user is asked to reload the object and specify another texture file. Following is a list for every face specifying the information

```
(t #of_texture v1.s v1.t v2.s v2.t v3.s v.t )
```

#of_texture indicates which of the specified textures should be associated with the particular triangle. v1.s and v1.t specify the texture coordinates s and t that are associated with the first vertex of the particular face (also holds for the second and third vertex). The file concludes with a right bracket).

```
(images 1
(f ../Textures/escherPattern1.bmp )
)
(triangles 168
(t 0 0.0 1.0 1.0 0.0 0.0 0.0)
(t 0 0.0 1.0 1.0 1.0 1.0 0.0)
...
(t 0 0.0 1.0 1.0 10 1.0 0.0)
)
```

Figure 9: Example for texture input file

As soon as object and texture are loaded, the user can modify the assigned texture coordinates. The user first needs to specify a triangle by either using the slider on the bottom left or by using the spin box (see Fig 10 (b)). The specified triangle will be highlighted in red. Afterwards, the texture coordinates for the given triangle can be altered by explicitly specifying new coordinates in the text edit fields or performing a "rotate" or "flip" operation. "Rotate" rotates the specified texture clockwise on the highlighted face. "Flip" flips the texture coordinates for vertices one and two of the highlighted face while vertex three keeps its coordinates. This results in mirroring the current image.

The user can also change the texture that is currently associate with the current face by using the spin box next to the "Apply" button.

After all faces have been edited according to the users wish, the texture information can be saved in the same texture format that was read in earlier.

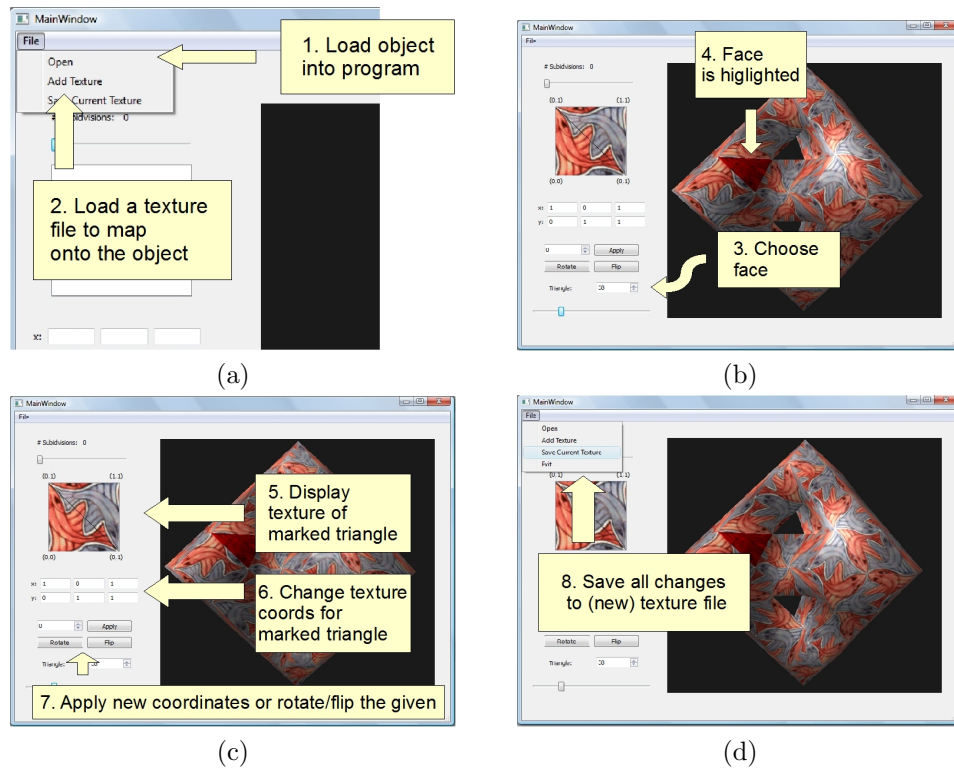


Figure 10: Procedure for applying texture to an object and saving the new tiling.

Limitations of the program

There are a few things that are not possible with this program, however they do not really pose any limitations:

- It is not possible to assign another texture file to an object that already has a texture file assigned to it. Please reload the object and then assign the new texture file.
- It is also not possible to change texture coordinates for objects that have already been subdivided. If the user wants to see a subdivision before he is done assigning all texture coordinates, the file should be saved before subdividing the object and then be reloaded with the object.