Beteckning:________________

HÖGSKOLAN
I GÄVLE

Department of Mathematics, Natural and Computer Science

Different Mapping Techniques for Realistic Surfaces

Tina Öhrn
June 2008

Thesis, 10 points, C level
Computer Science

Creative Computer Graphics
Supervisor/Examiner: Sharon A Lazenby
Co-examiner: Torsten Jonsson

Different Mapping Techniques for Realistic Surfaces

by

Tina Öhrn

Department of Mathematics, Natural and Computer Science

University of Gävle
S-801 76 Gävle, Sweden

E-post:
tina.ohrn@helsingenet.com

Abstract

The different mapping techniques that are used increases the details on surfaces without increasing the number of polygons. Image Based Sculpting tools in the program Modo and Z-Brush is used to create folds and wrinkles from photographs of actual fabrics instead of trying to create these shapes by modeling them. This method makes it easier to achieve photorealistic renderings and produce as realistic fabric dynamics as possible when they are applied on objects.
Keywords: normal mapping, bump mapping, displacement mapping, image based sculpting, Modo, Maya, Z-brush.

Table of Contents

1 Introduction 4
1.1 Problem definition 4
1.2 Aim 4
1.4 Expected results 5
1.5 Method 5
1.6 Earlier research 5

2 Theoretical framework 6
2.1 Bump mapping techniques 6
2.2.1 Bump mapping 6
2.1.2 Normal mapping 7
2.1.3 Realtime games – Parallax mapping 9
2.2 Displacement Mapping 9
2.2.1 Height Field Displacement 9
2.2.2 Vector Displacement 10
2.3 Ambient occlusion 11
2.4 Self-shadowing bump maps 11
2.5 OpenGL 12

3 Discussion of Creative Process 12
3.1 Modeling 12
3.2 Creating maps 14
3.2.1 Modo 14
3.2.2 Z-Brush 21
3.3 Rendering 22

4 Conclusion 25
References 27

1 Introduction

Today it is common to use computer generated pictures and animations by using three dimensional (3D) techniques among other areas to visualise products, architectural, and interior design. In many fields, they have discovered the potential and possibility to create photorealistic pictures in 3D compared to using a real photographer. It is a constantly growing and developing market.

When the desirable goal is to obtain realistic shapes and surfaces on soft objects similar to fabrics, pillows, and quilts there are different techniques available, such as normal, bump and displacement mapping. These procedures makes it easier to achieve a realistic appearance rather then creating a high polygon object with every little detailed modeled. These techniques are used to calculate and change a flat or low polygon surface in order to make it appear detailed and dynamic. Depending on which method is used, it leads to different working processes, visual results, and effects on rendering time.

1.1 Problem definition

The topic of creating photorealistic pictures in 3D is enormous. All the development in this area, the creation, and research of the available tools has been created to achieve the desirable results. Therefore, one may think that all the research has been completed. But every specific goal demands different solutions and techniques to achieve the best procedure and the best possible photorealism. Plus the advance in software and updates causes additional problems.

1.2 Aim

My purpose is therefore to focus on one type of dynamic object and apply materials to the surface that will resemble the fabric on the object. These types of dynamic surfaces are difficult to accomplish in the matter of making it appear photorealistic, particularly in making the folds and wrinkles in fabrics. I will also search for available techniques and try to find the best procedure to make the work process effective.

1.4 Expected results

Instead of modeling dynamic shapes that resembles folds and wrinkles on fabric I hope to find other methods to simplify the work process and increase the quality and realism.

1.5 Method

I am planning on modeling a sofa and then use images taken from photographs as reference material. Different mapping techniques will be applied to the sofa to simulate the reality and then images will be rendered out as photorealistic as possible. Various types of materials will be used to figure out a method to combine them to create a realistic and dynamic appearance.

I am going to use different available software to generate the different maps and set up a scene with lights and finally rendering out the photorealistic images.

Aside from Maya, I will use programs such as Modo, Z-brush, and Crazy Bump. During my internship, I have a contact with a company called Norrblick that have extensive experience in 3D in general. They also have the programs available that I will be using during this research.

1.6 Earlier research

Plenty of research has been made on how the different mapping techniques perform which shows the underlying mathematical calculations and visualizations of the differences in rendering results.

My focus area, which has been to create realistic fabric with folds and wrinkles, is not something that I have found specific reports about in my research. However, I found a report from 2007 made by Maur´ıcio Bammann Gehling, Christian Hofsetz, Soraia Raupp Musse [2]. It was a report on a novel 3D Painting System that allows interactive painting on normal maps directly onto the 3D model. They describe that when creating a normal map by using a high and low polygon object it may sometimes lose important aspects of the model due to the subtle details and/or right angles in the complex.

By using their 3D Painting System these lost aspects are eliminated. The technique works by adding a virtual grid on the object and the normal map is assigned to this grid. The technique is quicker compared to traditional modeling. This technique is still under development and the interface and tools are not fully completed.

2 Theoretical framework

2.1 Bump mapping techniques

Bump mapping is a computer graphics technique to enhance the appearance by giving a low polygon object more details without using more polygons. The effects are achieved by applying a generated two dimensional (2D) map on the surface and use the color information at each pixel to protrude the original objects surface information. The illumination calculations changes accordingly to the map and therefore obtains a more detailed surface while the underlying geometry stays the same.

2.2.1 Bump mapping

Bump mapping is an extension of the Phong Shading technique and was presented by James F. Blinn in the year 1978. It works in the same way as Phong according to adding a map and estimating the brightness of the pixels. The difference is that Phong shading calculates the reflection of light to give the surface a specular appearance and Bump mapping change the surface shape.

In Figure 1, a bump map is a greyscaled (single-channel) texture map where the brightness of each pixel represents how much it sticks out from the surface. It perturbs the existing normal by moving the vectors magnitude and therefore the technique is described as a height field method which is shown in Figure 2. [7] [8] [9]

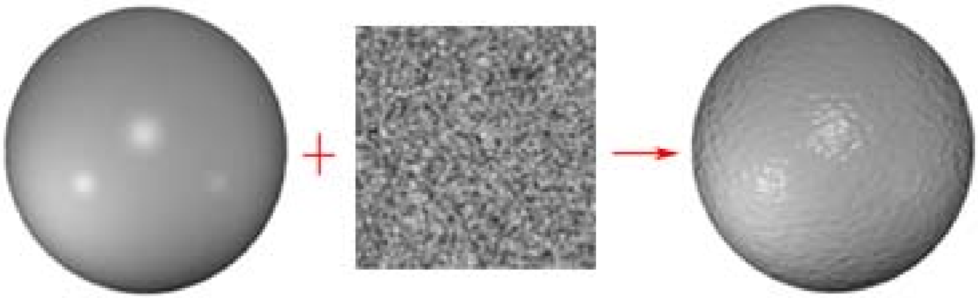

Figure 1: A sphere with Bump map and rendered picture.

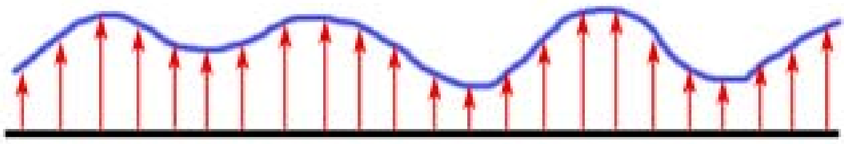

Figure 2: Visualization on the normals changing height from surface.

2.1.2 Normal mapping

Normal mapping, also referred to as "Dot3 bump mapping" is an advancement of the Bump mapping technique due to the fact it replaces the normal entirely compared to Bump mapping that protrudes the existing normal in one direction.

A normal is a vector perpendicular to a polygon. In Figure 3, the camera measure the angle between the normal and the light vector (the line and direction from the light source to the starting point of the normal) to determine how dark or light the face should render.

The technique called "normal map" is a generated map by using one low polygon object and one high polygon refined object. It calculates and stores the direction of high-resolution surface normals relative to the low-resolution surface normals into a map as in Figures 4 and 5. This increases the number of existing normals to the same amount as the high polygon object.

The map uses a 24 bits RGB image (R8G8B8) where every pixel represent a normal and the colors blue, green and red represent x, y and z coordinates of each axel. This provides the possibility to move the normals in any number of different directions that are not dependent on the current normals. The result by moving the normals is that it alters the way the camera perceives it and the actual surface stays the same.

Normal mapping is usually found in two varieties: object-space and tangent-space normal mapping. They differ in coordinate systems in which the normals are measured and stored. [3] [4] [5]

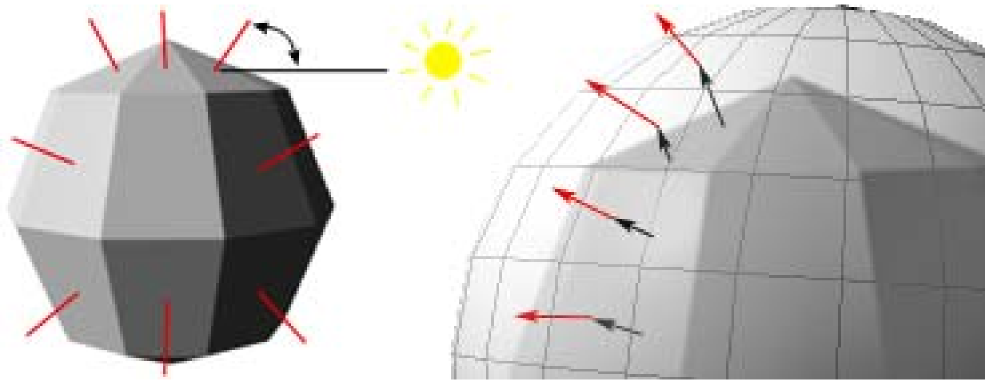

Figure 3: Direction of the normals from the faces. The angle between normal and light vector (first picture). Casting rays from the low polygon object onto high polygon object to create a normal map.

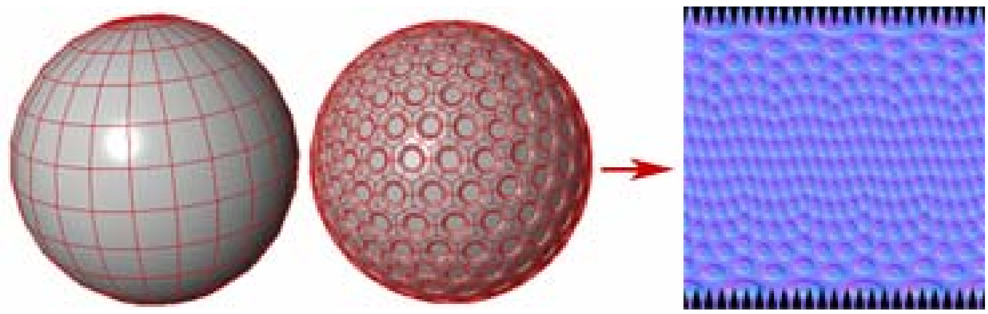

Figure 4: One low polygon object and the refined high polygon object and the generated map.

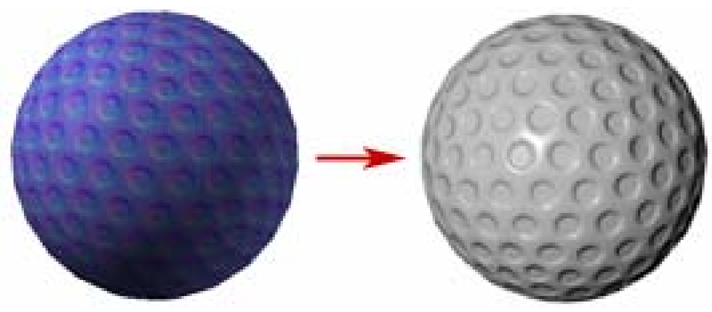

Figure 5: The normal map applied on the low polygon object and the resulting rendered picture.

2.1.3 Realtime games – Parallax mapping

Normal mapping is commonly used in realtime games and Parallax mapping is another type of mapping used in realtime. It is heavier than normal mapping but increases the surface appearance. Therefore, normal mapping is used more frequently in realtime games.

“Bump mapping is a texture-based technique, which means that it never changes the fact that the polygons are flat. Displacement mapping remedies this problem by modifying the actual geometry which is described in the Displacement Mapping section. "Parallax mapping" is an advanced bump mapping technique which is a method towards solving the same problem, but without the expense of displacement mapping. It will provide the proper parallax effect when the camera pans across a surface, but it will not modify the geometry similar to displacement mapping. Therefore, a surface will still appear flat when viewed under an angle, and an object's silhouette will not be affected by it.” [6]

“Parallax mapping, also known as "offset mapping" or "virtual displacement mapping" requires a height map as input. The parallax effect is simulated by offsetting each fragment's texture coordinates a certain distance towards the eye, with the distance being dependent on the value that was read from the height map at that location.” [6]

2.2 Displacement Mapping

2.2.1 Height Field Displacement

Displacement mapping was introduced by Robert L. Cook in the year 1984. [1] This technique works the same as the bump mapping technique according to the use of a height map. The difference is that bump mapping protrudes the normals on the surface and the underlying surface stays the same while displacement mapping alters the geometry shown in Figure 6. The pixels in the map are treated as polygons and the result is the same as if it was real geometry. Therefore, it can occlude other objects and cast shadows. This creates the enhancement, compared to bump mapping, when the surface is parallel with the viewer and provides the improved and correct silhouette. Due to the fact that it treats the map as a real object which has not been implemented in the hardware gives the disadvantage of needing more processing power and longer rendering time. [11]

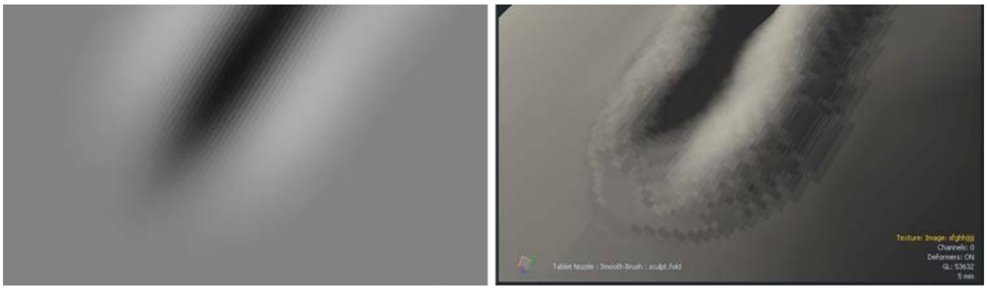

Figure 6: The Height map. The map applied on geometry and visualized with low resolution.

2.2.2 Vector Displacement

Vector displacement works as the Height displacement because it alters the geometry. But this technique also offers the possibility to change direction by using a 32 bit/pixel floating point color image. A grayscale image shown in Figures 7 and 8, as in displacement maps, can only store information of the offset strength and therefore cannot indicate a direction. [12] [13]

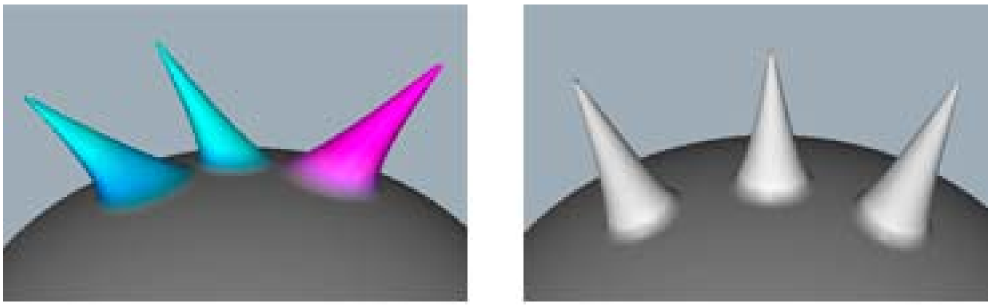

Figure 7: Vector displacement where the direction changes. Height displacement moving up in a straight direction.

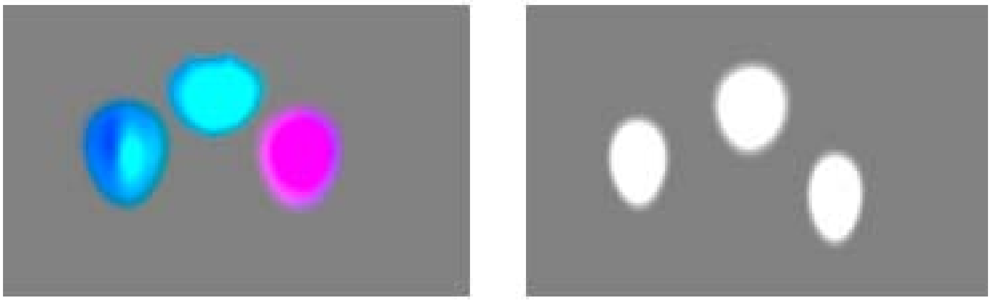

Figure 8: Vector displacement map applied on geometry. Height displacement map applied on geometry.

2.3 Ambient occlusion

Ambient occlusion, also referred as “dirt map” is a method to increase the level of realism in rendered pictures by using this global illumination effect which provides mesh intersections, cracks, corners and approximation of darkness according to surfaces that are close together. It can simply be explained as how much ambient light a surface point would likely receive. This is purely a rendering trick due to the fact it does not exist in real life. But it presents a better perception and depth of the rendered objects.

Ambient occlusion can be calculated in different approaches where casting rays in every direction from the surface is one of them. This technique tends to provide flickering areas when rendering animations.

The technique called Point Based Ambient occlusion is therefore more commonly used in animations due to the fact it is noise free and will not produce flickering areas. This method approximates the occlusion by treating polygon meshes as a set of elements that can shadow each other. [14] [15] [16]

2.4 Self-shadowing bump maps

Bump mapping affects lighting by protruding the surface normal, but the protruded normal will not affect shadowing in any manner. Therefore, one "bump" will not cast a shadow onto the next. However, techniques do exist that allow bump maps to self-shadow. One of them is "horizon mapping."

A Horizon map is a texture that encodes the elevation of the visible horizon in all directions from a particular sample point. If another point is determined to be below this horizon, the point is invisible to the current sample point. Likewise, if a light source is below the horizon, it does not illuminate the current sample point.

The main drawback, however, is that the Horizon maps tends to take up too much texture memory to be really practical. They are also fairly expensive to compute. [6]

2.5 OpenGL

OpenGL (Open Graphics Library) is a standard specification defining a cross-language cross-platform API for writing applications that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex three-dimensional scenes from simple primitives. OpenGL was developed by Silicon Graphics Inc (SGI) in 1992 and is widely used in CAD, virtual reality, scientific visualization, information visualization, and flight simulation. It is also used in video games where it competes with Direct3D on Microsoft Windows platforms. [17]

3 Discussion of Creative Process

3.1 Modeling

I have choosen to model the basic meshes of the sofa, in Maya, which is shown in Figure 9. The reason for choosing Maya is my own knowledge in the software. It allows me to be able to create the objects faster. I made the ten (seven small pillows, two big pillows to sit on and the main body of the sofa) different objects separate from each other instead of combined to be able to apply different maps on the objects.

The objects were made with a low polygon count and the UV mapping made with Maya’s different UV tools, and applying the textures was also created in Maya. UV mapping is a method of texture mapping a 3D model which arranges the 2D image on the surface using different attributes.

I assigned all UV maps a range of 1x1 to prepare them for all other mapping techniques which requires that resolution shown in Figure10.

Objects have been duplicated with the “duplicated special” tool in Maya which causes shifted normals. Then, they are imported into the 3D software Modo, which I will do later, and therefore they become invisible. This can be solved in Maya before exporting it to .obj file format by combining the object with a temporarily created primitive object, such as a cube, and then delete the primitive objects faces. Also remember to delete the history and freeze the transformations which should be performed on all objects.

After the basic meshes were created they were exported into .obj file formats to import into the 3D software Modo.

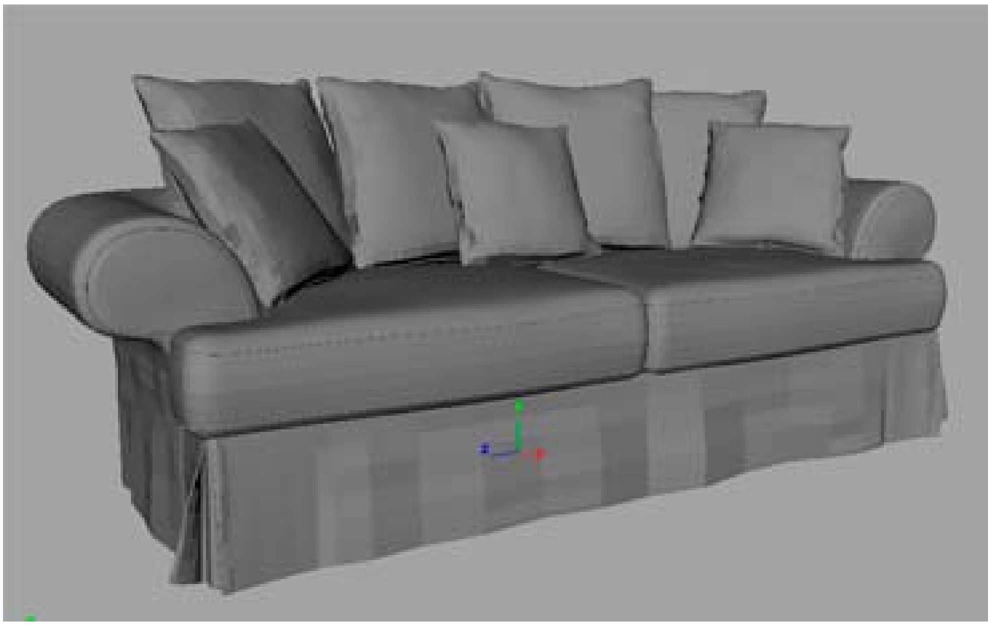

Figure 9: Snapshot of the modeled lowpolygon sofa in Maya.

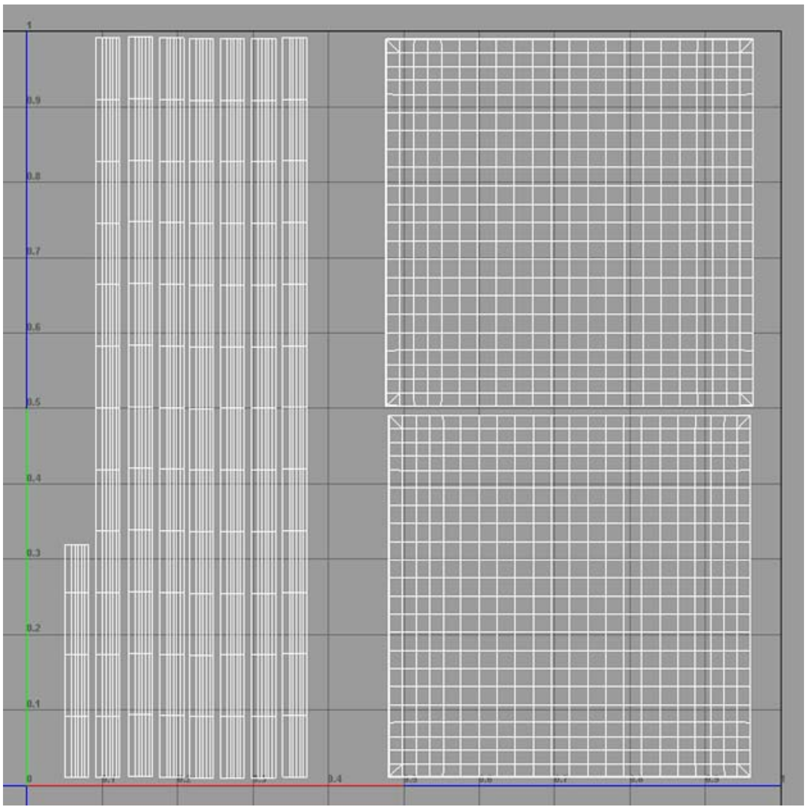

Figure 10: UV-mapping of one pillow.

3.2 Creating maps

3.2.1 Modo

I choose to explore the 3D software Modo which has available tools that enables Image based sculpting. This is, compared to Mesh based sculpting where you work on the mesh vertices directly, a method where the mapping techniques are used which are described earlier to store the detail of the sculpting. Also, there is a possibility to use stored maps as a brush tool and paint them on the object.

These maps can be visualized in realtime by the use of Advanced OpenGL mode which offers a preview of the added details while the sculpting is taking place. Therefore, it provides the same appearance as Mesh based sculpting.

The imported .obj files created an issue to solve. Objects that have several textures applied obtain separate layers in Modo, however they becomes locked together. If the UV horizontal and vertical wrap are changed in one layer, all the other layers obtain the same properties. It is solved, in Modo, by deleting the layers with textures and then creating new layers and importing the same textures again.

To cause the low polygon objects with a softer surface and edges, they are transformed into subdivision surfaces.

Subdivision surfaces are a method to represent a smooth surface by adding new vertices and faces based on the position of nearby original vertices. This technique can broadly be classified into two categories depending on how the division of the polygon is calculated. They are called interpolating and approximating where the first leaves the original position of the vertices and the second can move them if it is necessary to produce a smoother surface. It offers the choice to choose a level which determines how many times each polygon of the original surface will divide. [18]

A subdivision level of two was used on all objects to get a satisfying smooth surface, which indicates that each polygon is divided two times. One pillow which has a polygon count of 1672 before converting to subdivision will obtain a count of 26752 afterwards. A square polygon is represented by pre-tessellated triangles, which implies that the polygon count of 1672 is actually 3344 triangles. The subdivision object will acquire the tessellation when the object is rendered. The polygon count of 26752 will acquire a tessellated count of 53504 as in Figure 11.

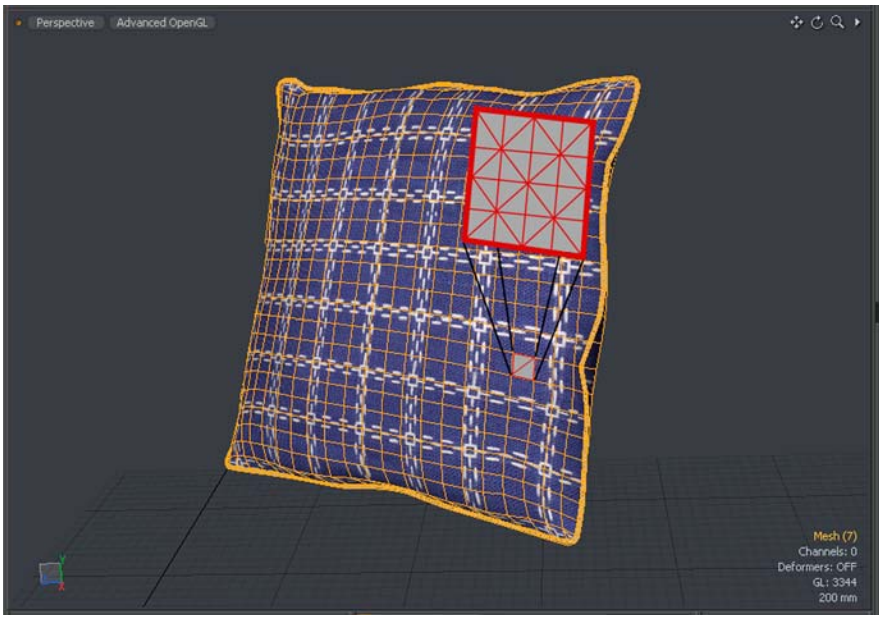

Figure 11: Visualization of one tesselated polygon with subdivision level of two.

The reason for using the subdivision technique is also due to the fact that two (normal mapping and bump mapping) of the mapping techniques, does not move or add the basic geometry vertices and therefore creates angular silhouettes when viewing close up. In Figure 4, the circles silhouetted on the rendered picture are still angular. This technique also presents the possibility to easily step up or down the resolution. It also provides less impact on memory compared to increasing the basic polygon count due to the fact that vertices that are created inside the subdivision surface do not carry any vertex map data or other metadata that is needed by real vertices. [10]

A bump map was created simply by converting the texture that resembles the fabric on the sofa, into grayscale, which was created in Photoshop. Then it was applied on all objects. This method was used as the first step in all the following tests.

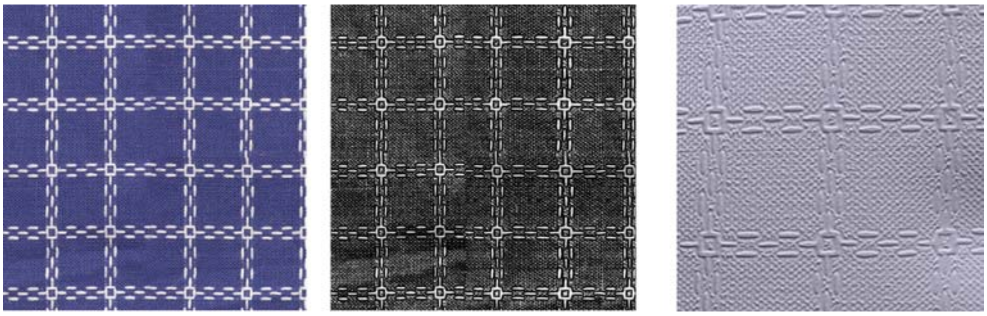

Figure 12: Texture that resembles the fabric, the grayscaled bumpmap and the rendered example.

I tested the basic method of creating a normal map which is one low and one high polygon which was explained earlier. The procedure will be showed in the final render on one of the pillows in Figure 12.

The low polygon object was copied. Then, the polygons were subdivided three times to obtain a high polygon object. The suitable amount of polygons will be able to refine and add folds by changing the basic geometry vertices. The objects must be in the same position to be able to create the map where the position of the object will be left at the point where it was created. By hiding the low polygon object refinements can be made on the high polygon object by using Modo’s available Sculpting Tools.

After the refinements, the object is scaled down slightly so that the normals from the refined object could be projected on to the low polygon surface.

After adding a blank normal map texture onto the low polygon object and making visible the original object, the final map can be created by using the property “Bake from object,” which is show in Figure 13.

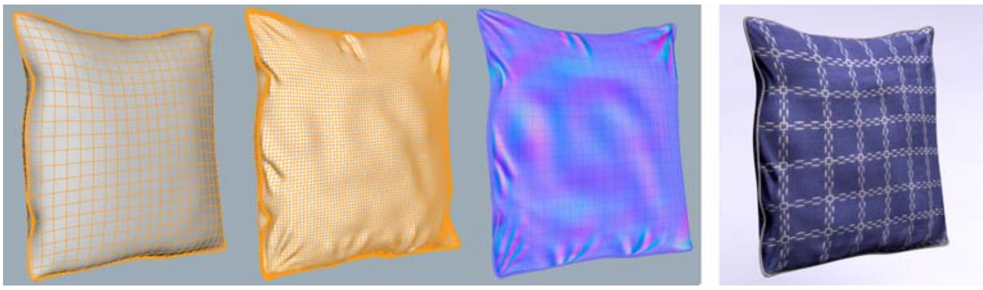

Figure 13: Low and high polygon objects and the created normal map applied. And the last picture is the rendered result.

In this next example, Image Based Sculpting was explored further. A blank normal map texture was added onto the low polygon object. Then three normal map textures were created in different ways to use as “Image Ink” brushes in Modos “Paint Tools.” Below are the three procedures to create the maps:

1) The software’s Crazy Bump was used which converts pictures into different types of maps (normal map – displacement map – ambient occlusion – specular map). The picture of fabric with folds was converted and import into the created normal map of Modo’s Paint Tool.

2) The Crazy Bump was used again to convert another picture of fabric with folds. This time the created displacement map was imported into Modo and assigned it to a created flat polygon face. Then, the assigned displacement map automatically alters the flat geometry to the folds in the picture. The displacement, however, is small and the face almost looks flat as shown in Figure 14.

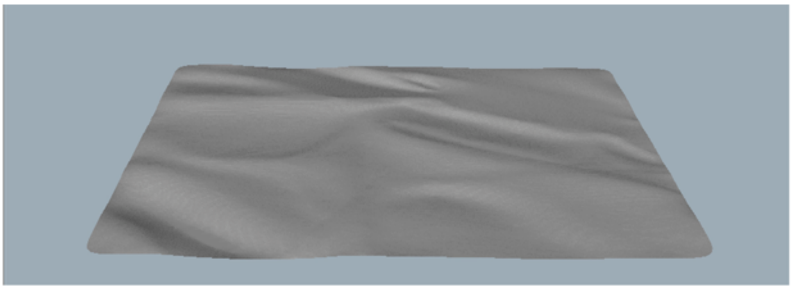

Figure 14: The flat face with displacement map.

This was procedure was created because the imported displacement map is a 16 bit .tif image which cannot handle negative values. To support and provide the possibility to change how high or low to sculpt the height map Modo has a simple method for solving this. On the displacement map properties, there are settings for Low and High values that can be decreased or increased until you find the correct value. And by changing the properties in the “displacement distance” in the Base Material node, it can determine how long the distance should be from the lowest point to the highest as in Figure 15.

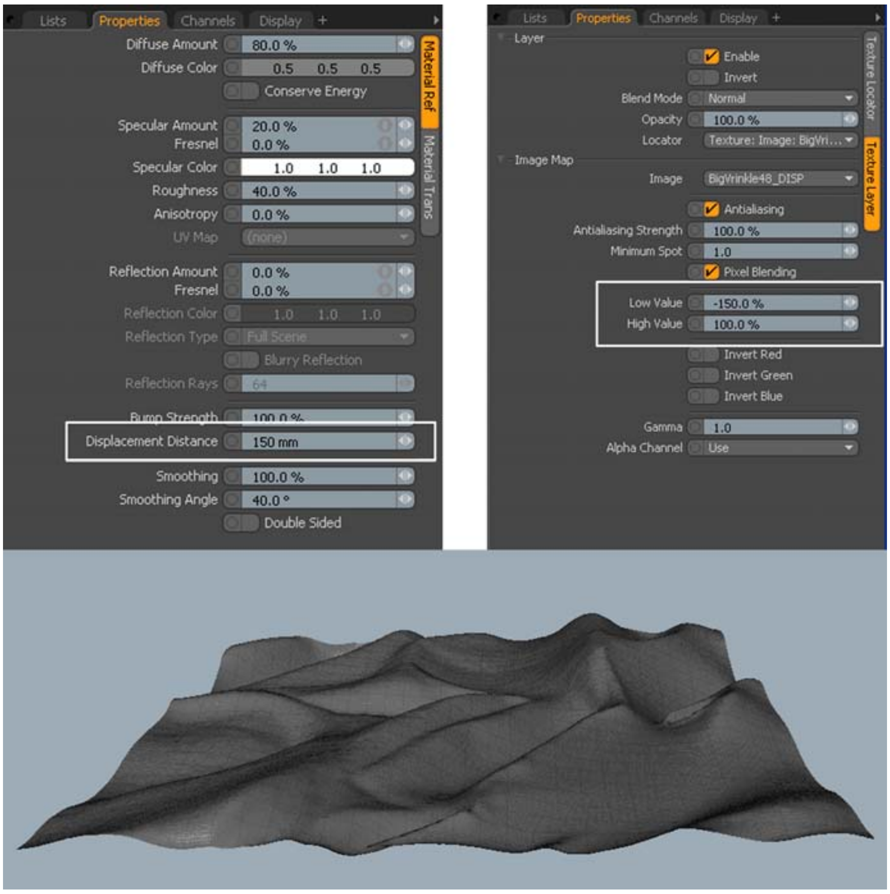

Figure 15:The properties to change in Modo marked with white squares. A snapshot of the flat face with changed values on the assigned displacement map.

After finding the proper values, a blank texture was added and converted it to normal map. By using the property Bake, it creates the normal map from the flat face with the assigned displacement map. Then, it is ready to import into Paint Tools and use as an Image Ink.

3) One effective technique is to use a section of an already existing normal map. By opening one created normal map (the map created with a high and low polygon object) in Photoshop and selecting the part to use and then save a new image, the image can be imported into Paint Tools and used as an Image Ink.

The pictures to use in Paint Tools are now made and imported into Image Ink and can be used as brushes shown in Figure 16. By marking the earlier created blank normal map that was applied on the low polygon object, it is ready to start painting on it using Image Ink Tool. The blank normal map automatically updates with the applied details while working. The Image Ink pictures can also be scaled up or down to vary the size of the details when you paint it onto the object. Once all of the details have been added, it is complete for rendering which is shown in Figure 17.

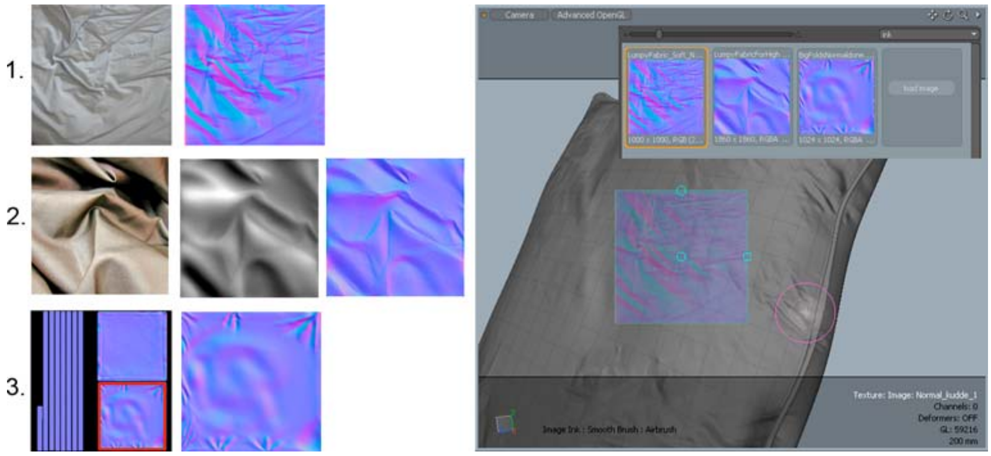

Figure 16: Three different ways to create normal map pictures to use in Paint Tools, and the display in Modo while working.

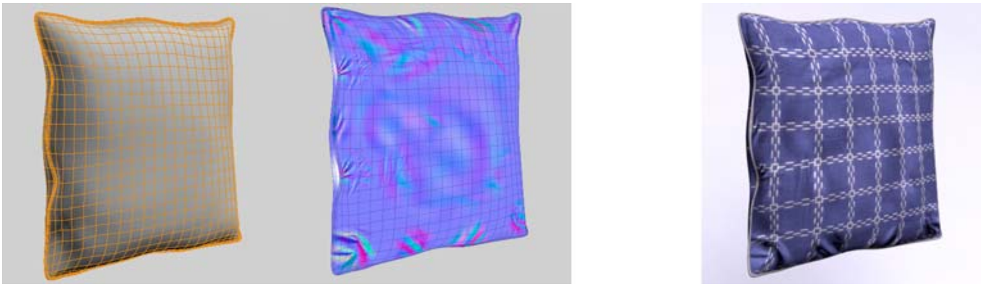

Figure 17: The low polygon object, the painted normal map applied and the rendered result.

In this third example, Displacement mapping was explored as well as the Paint Tools. A blank Displacement map was added to start with on the low polygon object.

Then, three Displacement map textures were created in different techniques to use as “Image Ink” brushes in Modo’s “Paint Tools.” Below are the three procedures to create the maps:

1. The software Crazy Bump was used again. The picture of a fabric with folds was converted and imported the created Displacement map into Modo’s Paint Tool.

2. Then, a picture was painted to resemble a fold by using the “gradient tool” in Photoshop. By saving it as a grayscale 16 bit .tif image it automatically becomes a Displacement map and can be imported into Paint Tool.

3. In the third example, a polygon geometry object was created to resemble a fold. By using the available tool "Object To Brush," the object was converted into a Height Displacement map. Then, the image can be imported into Paint Tools.

The finished pictures for use in Paint Tools are now made and imported into Image Ink and can be used as brushes. By marking the earlier created blank Displacement map that was applied on the low polygon object, painting can now start on it using the Image Ink Tool. The blank Displacement map automatically updates with the applied details while working. The Image Ink pictures can also be scaled up or down to vary the size of the details when painting it onto the object which is shown in Figures 18. When all of the details have been added, it is ready to be rendered as shown in Figure 19.

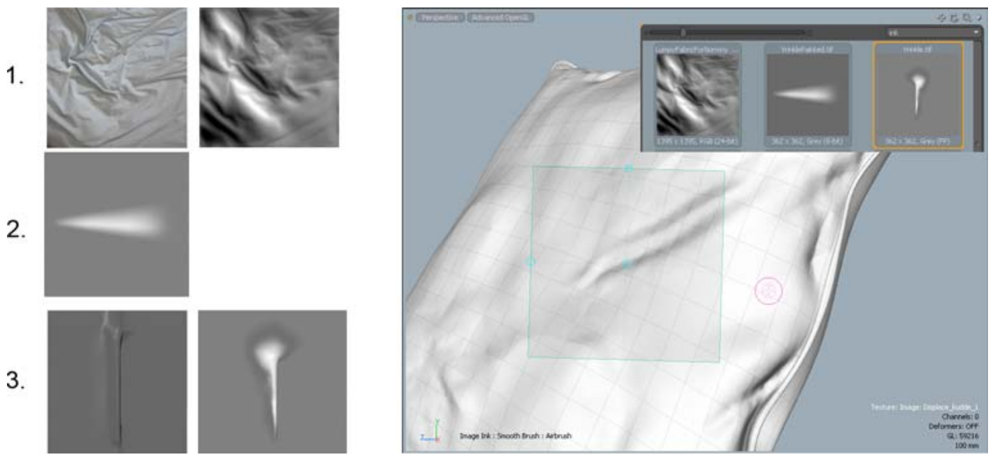

Figure 18: Three different ways to create Displacement map pictures to use in Paint Tools, and the display in Modo while working.

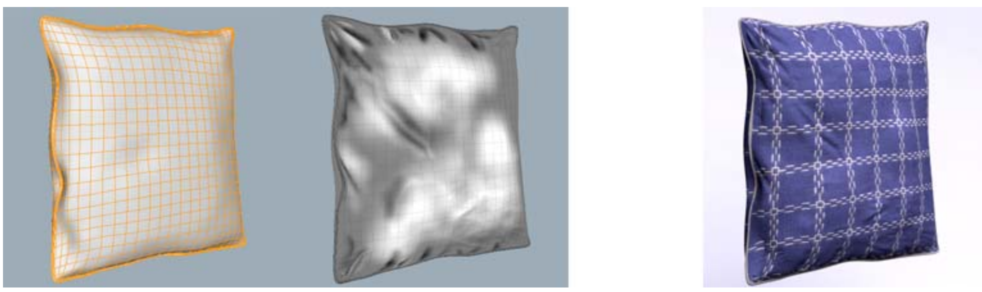

Figure 19: The low polygon object, the painted Displacement map applied, and the rendered result.

3.2.2 Z-Brush

To explore additional software, I selected Z-Brush, which also has the Image Based Sculpting tool. Z-Brush is a digital sculpting tool that can process a large amount of polygons because it works with pixols instead of pixels and therefore is a combination of 2D and 3D techniques, and informally it is called 2.5D. Z-Brush has the ability to export normal maps and displacement maps adjusted for other software.

An .obj file format of a pillow was imported into Z-Brush. Then by using the available tools, the large distinguishing features were formed on the surface. To resemble the smaller wrinkles in the fabric, a couple of pictures were created by using Gradient tool in Photoshop and by editing photos of fabrics with folds. The pictures are grayscaled so that the 100 % black areas in the picture are handled as alpha shown in Figure 20.

The pictures were imported into Z-Brush to use as brushes. These are handled the same way as in Modo, they can be scaled up or down to vary the size of the details when it is painted onto the object.

Then, normal maps are generated by using the tool ZMapper. The generated normal map was then imported into Modo and applied to the edited pillow shown in Figure 21.

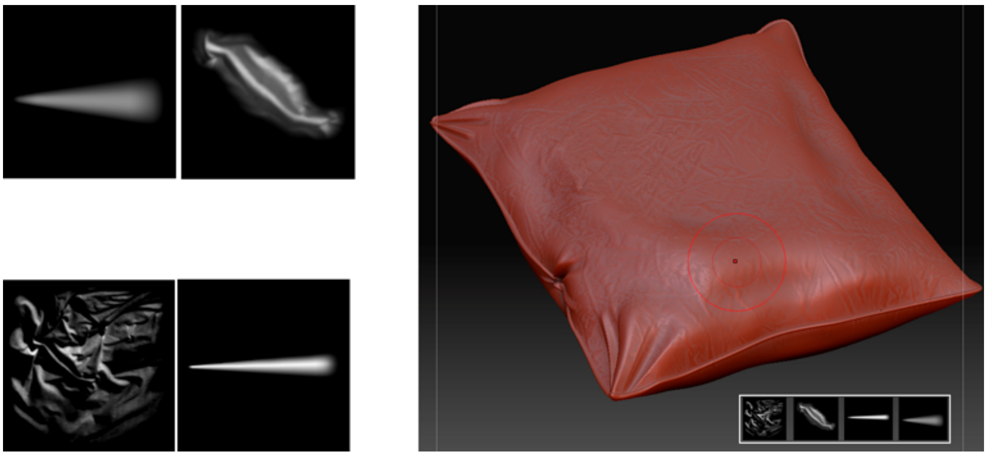

Figure 20: Pictures created in photoshop and the display in Z-brush.

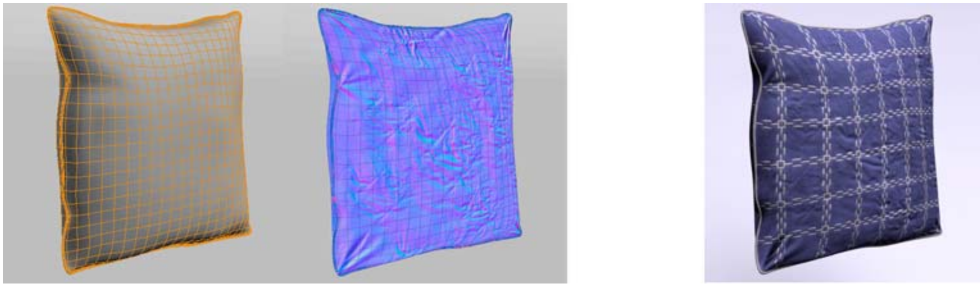

Figure 21: The low polygon object, the generated Normal map applied from Z-brush, and the rendered result.

3.3 Rendering

The images below are representing renderings created from Modo. These final images are rendered with HDRI imaging. The mapping techniques show realistic wrinkles and folds on the pillows and sofa shown in Figure 22a and 22b.

Figure 22a: Rendered sofa.

Figure 22b: Rendered close up on sofa.

In Figure 23 number 1 column, the normal map is shown from baking a normal map from a high polygon object and the resulting Ambient Occlusion is shown underneath.

Number 2 column, of the same Figure 23, is made by Image Base Sculpting on the object and resulting Ambient Occlusion is also shown underneath.

And in Figure 23 number 3 column, there is a displacement map on the object and Ambient Occlusion is also shown below.

All the maps are treated the same as if the wrinkles and folds were modeled as a high polygon object. Therefore, these techniques are excellent due to the fact that we often use the Ambient Occlusion to increase the realism in rendered images.

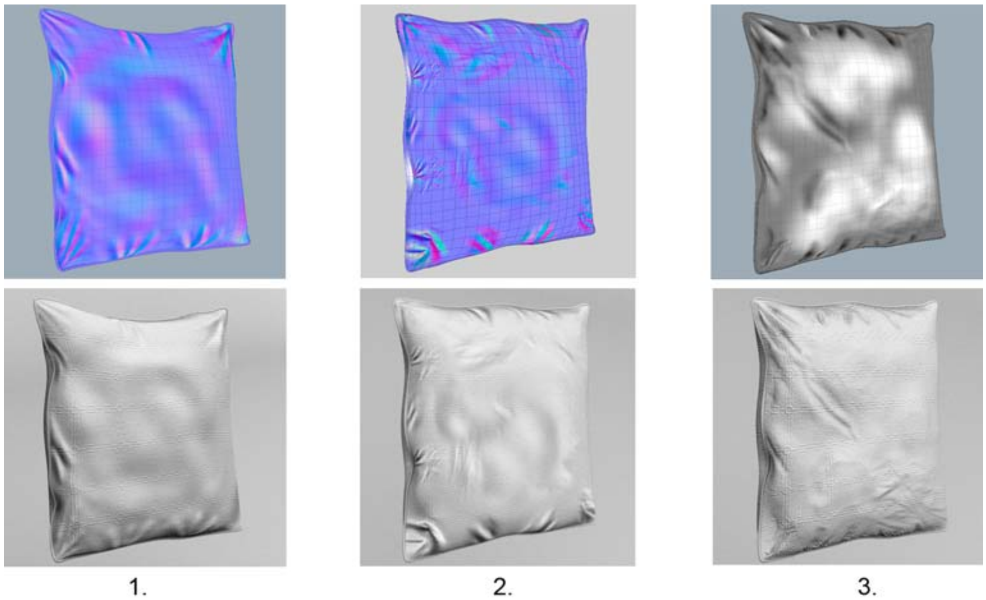

Figure 23: Rendered Ambient occlusion pass with different maps.

4 Conclusion

The different mapping techniques used is primarily techniques that increase the details on surfaces without increasing the number of polygons. It has not been my primary objective to model with as few polygons as possible, instead I have used these techniques in order to find a method to produce as realistic fabric dynamics as possible.

I also perceive projects where a large number of sofas or similar objects could be created. The amount of modeling with polygons would become hugely time consuming, as well as difficult to become realistic. Therefore, there is a possibility that the procedure would become more efficient through creating a huge library with completed pictures in order to use in Image Based Sculpting.

I have used the Image Based Sculpting tools in the program Modo and Z-Brush. Therefore, I was able to create folds and wrinkles from photographs of actual fabrics instead of trying to create these shapes by modeling them. This method makes it easier to achieve realism in working with fabrics when they are applied on objects.

Displacement mapping is my primary choice when it comes to creating larger shapes because these shapes often throw shadows. However, I believe that horizon mapping could also be used as an alternative.

For the surface structure, I have used Bump mapping. The Bump mapping technology is excellent at handling more delicate surface structures and does not need the type of manipulation that the normal map can create. The simple procedure of converting a color picture to grayscale picture provides the desirable structure because the same picture is used and the same UV mapping is used and thereby it matches the shapes perfectly.

Worth mentioning are the different procedures with normal maps and the use of Image Based Sculpting. In Modo, the picture is ”brushed” directly onto the normal map texture and therefore the normals are not calculated based on the underlying geometry. Therefore, it is a ”fake method” of creating it. But, it works! In Z-Brush, the normal map is created based on the finished modeled object.

This research project was limited by learning both Z-Brush and Modo, as well as learning the techniques to create the normal and displacement maps. When it comes to which program to use when working with Image Based Sculpting, Modo or Z-Brush, I consider that a question of which software the person is most comfortable with using.

From my experience, Z-Brush is just for creating models and different maps and does not have a pipeline to the finish project, where as Modo has the complete process of modeling, and creating the maps to rendering the images. If more time was allocated, it would have been beneficial to research more on the Horizon map.

References

[1] Cook, R.L. “Shade trees”. In: SIGGRAPH’84: Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, pp. 223–231. ACM, New York (1984) (DOI 10.1145/800031.808602)

[2] Mauricio Bammann Gehlin, Christian Hofsetz, Soraia Raupp Musse. “Normalpaint: an interactive tool for painting normal maps.” (DOI 10.1007/s800371-007-0132-9) Online.Internet. July 3, 2007.

[3] Lanier, Lee. *Maya Professional Tips and Techniques*. Sybex: Autodesk maya Press, Wiley Publishing, Inc. 2007. pp 184-191.

[4] http://en.wikipedia.org/wiki/Normal_mapping. [online] 3 5, 2008.

[5] http://highend3d.com/maya/tutorials/texturing/229-1.html. [online] 3 5 2008.

[6] http://www.delphi3d.net/articles/viewarticle.php?article=bumpmapping.htm. [online] 5 5 2008.

[7] J.F. Blinn, “Simulation of Wrinkled Surfaces”, In Proceedings SIGGRAPH’78, pp. 286-292, 1978

[8] http://freespace.virgin.net/hugo.elias/graphics/x_polybm.htm. [online] 5 5, 2008.

[9] http://en.wikipedia.org/wiki/Phong_shading. [online] 5 5, 2008.

[10] Software: Modo version 301, User Guide pp. 149.

[11] Doggett, Michael and Hirche, Johannes. “Adaptive view dependent tessellation of displacement maps”, Page 59-67. ACM (DOI 10.1145/346876.348220)

[12] http://en.wikipedia.org/wiki/Displacement_(vector). [online] 7 5, 2008.

[13] Software: Modo version 301, User Guide pp. 159.

[14] http://wiki.blender.org/index.php/Manual/Ambient_Occlusion.[online] 7 5, 2008.

[15] http://en.wikipedia.org/wiki/Ambient_occlusion . [online] 7 5, 2008.

[16] http://http.download.nvidia.com/developer/GPU_Gems_2/GPU_Gems2_ch14.pdf. [online] 8 5, 2008

[17] http://en.wikipedia.org/wiki/OpenGL. [online] 8 5, 2008.

[18] http://en.wikipedia.org/wiki/Subdivision_surface. [online] 19 5, 2008.