CAP 3027 - Team Mayans: Texture Mapping and Filtering

Introduction

[Slide 1]
Good afternoon, my name is Nipuna, and this is Jennifer, Melanie and Evenson and we are Team Mayans. Our topic for this presentation is on texture mapping & filtering. Texture mapping is a technique that is used to add detail to a surface. We will go over the main types of texture mapping including bump mapping, normal mapping, displacement mapping, horizon mapping, relief mapping and parallax occlusion mapping. Evenson will first take us through bump mapping.

Bump Mapping

[Slide 2]
Bump mapping is a texture mapping technique used to simulate bumps and wrinkles on a surface. The key to bump mapping is that the changes it makes to the surface are an illusion. It does not actually change the surface geometry of the object--it just makes the illusion as though the surface geometry were changed. This is achieved by perturbing (disrupting if you will) the surface normals of the object and then using the perturbed normals when doing light calculations for that object. More specifically, the original normals of the surface are found, then the surface displacement for the actual object which is given by a function F(u,v). The position of a point on the surface is then displaced by an amount equal to the value of F. This gives a new position vector for each point. The new surface normals have to be calculated as well and this is done by taking the cross product of the partial derivatives for that surface.

[Slide 3]
In the way the bump map works is that it is a grayscale image that is limited to 8-bits of color information. Which amounts to 256 different intensities of black, white, and grays. The values in the bump map tell how the details will appear on the texture map. The lighter the values get, the more the texture appears to push out of the surface and by contrast the darker the values get, the more the texture appears to push into the surface.

[Slide 4]
This slide gives an example of the results of when a bump map is applied to a surface and difference in level of detail when compared to a basic texture map. Notice how the edge of the wall is still straight.

Normal Mapping

[Slide 5]
Normal mapping is comparable to bump mapping because it is a way to create more detailed textures on a surface without creating additional geometry. It creates the shading texture from a high-poly model and maps it onto a low-poly model, using surface normals and their orientation in three-dimensional space to determine their R, G, B values from their x, y, z orientation, respectively. This differs from how bump mapping uses a height map, which simply stores the heights of points on the surface. Normal mapping creates the illusion of depth on the surface through the use of lighting and shadows, which is how it simulates the appearance of very detailed wrinkles and indentations on the surface.
There are different types of normal maps-- tangent space, object space, and world space. World-Space maps are the most efficient, Object-Space maps are second-most efficient, and Tangent-Space maps are the least efficient. In this slide, you can see how the maps look different. Tangent-Space maps are blue-purplish because all the surface vectors are "up" even if it is not the case in World-Space, meaning their z-component is 1 and the blue value is maximum. World-Space and Object-Space normal maps look exactly the same, but are applied differently. The choice of normal map depends on the type of model it will be applied to. Tangent-Space normal maps can rotate and deform, making it ideal for character models. Object-Space normal maps can be applied to models that do not deform but move, such as cars. World-Space normal maps should be applied to models that cannot deform, rotate, or move at all and are most computationally efficient, so they would be ideal for houses.

[Slide 6]
In this picture you can see the difference in how a normal map is created compared to how a bump map is created. The normal map uses surface normal vectors while a bump map simply uses height vectors going in one direction.

[Slide 7 - Demo]
In this model, only the normal map is applied. As you can tell, a normal map is moreso used for the purpose of lighting and shading in the model, and not for simulating the model's general structure. One of its drawbacks is that it is not good at displaying prominent, large details on the surface because they become flatter as viewed closer to the horizontal of the surface. A bump map would be preferable for this purpose, while a normal map would be used for more complex details. They would be used to create the illusion of facets, tubes, and other rounded surfaces, such as the rounded corner in this clip. Small specular highlights are displayed on the rounded surfaces as the object is being rotated. Usually, models will have a combination of mapping techniques applied to it, such as bump+normal, or displacement+bump, so that the textures will have both larger bumps and smaller wrinkles and highlights on the surface.

Displacement mapping

[Slide 8]
Displacement mapping functions in a similar way to bump mapping and normal mapping except for one major difference. Unlike the previously described types of mapping, displacement mapping actually physically displaces the mesh to which it is applied. This is a good way to create detail in low resolution meshes however due to the physical displacement, it is more system intensive and costly to render than the other mapping techniques. As the slide shows, the displacement map has the ability to show real depth since the physical surface of the 3D model is altered when rendered. This adds additional geometry according to the height map function to which it is applied.

[Slide 9]
The theory behind displacement mapping outlines the process of displacing the geometric position of points along the surface normal. It does this by decomposing the surface into a macrostructure geometry which is assumed to be a triangle mesh. This tessellated mesh is then displaced using a height-map in the direction of the normal surface vector.

[Slide 9 - Next]
The displacement mapping algorithm, as shown in the slide, takes sample points from the surface and displaces them perpendicularly to the normal. The distance to which they are displaced is based on the height map. The parameters u and v refer to the texture coordinates in the texture space. The p is the macrostructure surface which is assumed to be a triangle mesh. The N refers to the unit length normal vector while the h is the height map function. The height map function is usually stored as a texture.

[Slide 10]
Instead of using a tessellated mesh, displacement mapping can also be applied to the corresponding texel centres and this is called per-pixel displacement mapping. Per-pixel displacement mapping is often more computationally expensive than mesh tessellation, but it is more exact and easily refinable. This method of mapping can be implemented in a GPU's pixel shader while the tessellation method cannot. The sense of depth is given by displacing the mesh texture coordinates that the viewer is directly looking at.

[Slide 11 - Demo]
As can be seen from the demo, unlike the Normal mapping, the displacement mapping actually physically displaces the surface. This can be seen when looking side on at the image. The slowness of the video also shows how much more system intensive this method of mapping is compared to normal and bump mapping.

Horizon Mapping

[Slide 12]
Shadows are helpful in providing cues for understanding surface shape. One of the main shortcomings of bump mapping is that it doesn't define any new geometry meaning that there are no actual bumps to cast shadows. In order to simulate the casting of shadows, a separate shadow map is required for every light source direction. The main idea behind horizon mapping is to precompute limited visibility from each point on a surface. The horizon angles are computed by shooting rays from the horizon towards the surface normal until there is no intersection.

[Slide 13]
The precomputation step includes the following. Firstly a vector valued perturbed normal map is produced using the bump map.

[Slide 13 - Next]
Secondly, 2 horizon maps are created from the 8 directions: North, North-East, East and so on. This is done by encoding 4 different directions into the four color channels: Red, green, blue and alpha.

[Slide 13 - Next]
Finally 2 basis maps are created from the 8 directions. The basis function evaluates to 1 for the given direction and 0 for the neighbouring directions. This creates a radial function where the horizon angles are linearly interpolated between discrete directions. The main limitation of horizon mapping is that it cannot cast bumpy shadows onto other objects, and other objects cast shadows onto receiving surfaces as if it was not bumped. There is also a cost involved in precomputing where the geometry of the object can have a significant impact on performance.

Relief Mapping

[Slide 14]
Displacement mapping provides a good level of detail because it actually changes the surface geometry. In particular, the desired aspects are parallax and occlusion. Parallax is the difference in apparent direction of an object when seen from different viewpoints, while occlusion is the blocking of objects from view. Parallax motion, then, is when objects further away appear to be moving slower relative to objects that are closer.

[Slide 15]

Unlike displacement mapping, these aspects were not accounted for with bump and normal mapping, which is why they were appropriate only for small surface variations.
However, because displacement mapping is expensive, other techniques have been made to simulate parallax and occlusion.

[Slide 16]
Relief mapping is one such method, wherein a texture map with a height map is pre-warped prior to texture mapping onto a polygon. With these two steps--pre-warping and texture mapping, complicated objects can be represented on simple geometry with parallax and occlusion.

[Slide 17]
In the first step of pre-warping, a horizontal and vertical pass are performed on the original texture map, or source image. Part of the necessary perspective changes are done here, as may be seen on the diagram. The warped image is much closer to what should be seen from the camera's viewpoint in comparison to the unwarped original. The diagram to the right shows a breakdown of the warping, where the texels A and B are shifted to the right in the horizontal pass and down in the vertical pass to account for the camera view of the object; the colors are interpolated during rasterization. The complete texture map is then mapped onto the geometry in the conventional way.

[Slide 18]
Here, you can see that complicated geometry may be represented with relief mapping. This object was actually mapped onto a box. In this case, there were 6 texture maps, one for each face of the box. From there, the needed quads were determined, and the textures were pre-warped based on the original textures. The texture mapping of these pre-warped textures completes the appearance of parallax.

[Slide 19]
In this demo, you can see that relief mapping provides good visual results, and accounts for the parallax that comes from change in perspective--unlike normal mapping. However, because relief mapping references texels--or pixels in the texture map--aliasing artifacts are apparent at sharp viewing angles.
This particular method of relief mapping works on simple geometry, but on arbitrary polygonal surfaces, such as a teapot or robot, would require a different sort of computation.

[Slide 20]
For relief mapping on arbitrary polygonal faces, the idea is to first compute the viewing direction as the vector from the viewer to the 3D position of the point on the polygonal surface. You then transform the vector to the tangent space associated with the current fragment.
Next, you use the transformed vector and point A--the texture coordinates of the fragment--to compute B--the texture coordinates where the ray reaches the depth value 1.0.

Then, compute the intersection between the transformed vector and the height-field surface using a binary search starting with A and B
Finally, perform the shading of the fragment using the attributes associated with the texture coordinates of the computed intersection point.

Parallax Occlusion Mapping

[Slide 21]
In a similar vein, Parallax Mapping is a method of creating real-time parallax distortion by shifting the texture coordinate to the one that is actually seen on the represented surface. An improved version of parallax mapping is Parallax Occlusion Mapping, or POM.

The basic idea behind POM and parallax mapping is shifting the texture coordinates of what you see to what you should see. So here, the view direction is the vector from the camera or viewer towards a point on the surface. The polygon itself is a flat surface, so what you see is texel T_a; but the real surface you want to see is at point B, which coincides with texel T_b. The solution to this is to calculate the offset from T_b to T_a and adjust the texel such that when you are looking at point B on the surface, you see texel T_a.
On the next slide, you will see this in more detail.

[Slide 22]

With POM, the displacement information for the surface is encoded in a height map. This diagram is the same as the previous, but with the polygonal surface and the actual surface switched. In this diagram (unlike the previous), the polygonal surface is shown at the top, while the curved purple line below represents the surface given by the height map. It's just a different way of visualizing the same idea.

Here are the steps for parallax occlusion mapping for each vector or pixel--notice how similar it is to relief mapping on arbitrary polygonal faces:
First, you compute the tangent-space viewing direction v_{ts} and the light direction L_{ts} per vertex. These are interpolated and normalized before the next step. The view ray is then ray casted along the parallax offset vector P to find the height profile-ray intersection point. In this step, you may follow the method done in relief mapping, which used a combination of linear and binary search. For this particular diagram, the author used a linear search that sampled the height field at intervals of δ to find t_{off}. This is a key difference between relief and POM mapping, which you will see on the next slide.
Estimating the light visibility deals with the amount of light on the area. It is found by ray casting L_{ts} and sampling the height profile for occlusions. This way, areas with some occlusion would be darker, or in a shadow, as opposed to areas that are not occluded.

[Slide 23]

As you can see here, if you perform Relief Mapping with only linear search, there are very obvious problems with viewing at steep angles.
In comparison, using Parallax Occlusion Mapping's method of linear search produces much better results. This is because relief mapping's linear search searches along the vector from the viewer towards the bottom of the depth map; it requires a binary search for precision. With POM's method, the linear search is performed along the actual curve of the surface.
Both Relief mapping and POM are subject to problems at grazing angles, however, because their searches are dependent upon δ.

[Slide 24 - Demo]

Comparison slides

[Slide 25]
Here, you can see all the maps that we have gone over in this presentation.
The first map, bump mapping, was the first technique developed for adding detail to surfaces. However, it is limited to small surface extrusions; that is, the surface's height variations must be small compared to the overall size of the surface for it to be visually appealing.
Normal mapping is essentially an upgraded version of bump mapping, with a wider range of intensities to dictate height variations on the surface.
Despite this, normal and bump mapping are not very good for surfaces with large bumps or dips. Thus, displacement mapping was developed. It produces the best results by actually displacing the geometric position of vertices at render time; for this very same reason, however, displacement mapping is very expensive and impractical for real-time applications as in video games. As such, other techniques have been developed.
The two mapping techniques that we presented are relief and parallax occlusion. Both serve to simulate parallax and occlusion, which were not accounted for in bump and normal mapping.
In relief mapping, image warping and texture mapping are performed as a two-step process for each render or change in perspective. The cost of relief texture mapping lies in the number of texture maps, while its advantage lies in the simplicity of the geometry.
In parallax occlusion, is a very good technique for real-time rendering as with video games; but like relief mapping, it is a sampling based algorithm where aliasing is affected by the number of samples and, thus, computation time.

Conclusion

Of course, there are many more types of texture mapping. Because each mapping technique is built upon a previous one, most texture maps seek to achieve the same goal but through different or improved algorithms and implementations. There is not one mapping technique may be considered the best; rather, they are used as appropriate depending on the context for the best efficiency and result.