

# Terrain Rendering using Displacement Mapping

by Per Rasmussen and Jacob Nielsen

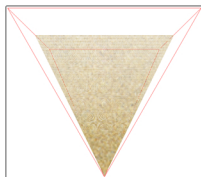
## Motivation

In many modern computer games it is often desirable to have vast outdoor scenarios. To achieve this at interactive frame rates many techniques have been employed. We will here present a novel technique that utilizes displacement mapping in the vertex shader to render a huge terrain at a constant high frame rate.

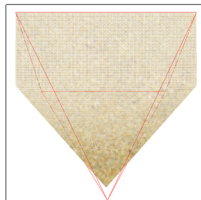
The technique is based on the principle of having a static grid mesh which always covers the view frustum.

## Objectives

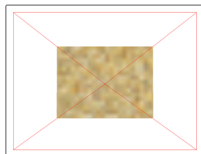
- Creating a grid mesh that covers the view frustum and nothing else.
- Form the grid mesh as the projection of the view frustum onto the ground plane.



The grid mesh seen from above when the camera is aiming horizontally



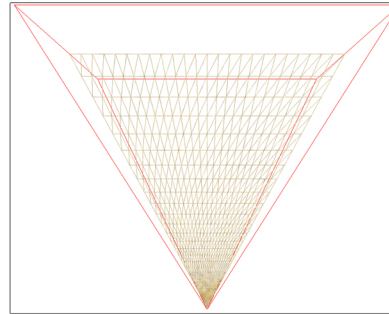
Top-down view of the grid mesh when the camera is aiming approximately 30° down.



Top-down view of the grid mesh when the camera is aiming straight down. To optimize, the projection is limited to the path of the view frustum that is above ground level.

## Level of detail

- Rectangular grid mesh placed in the view frustum gives a higher density of vertices near the camera



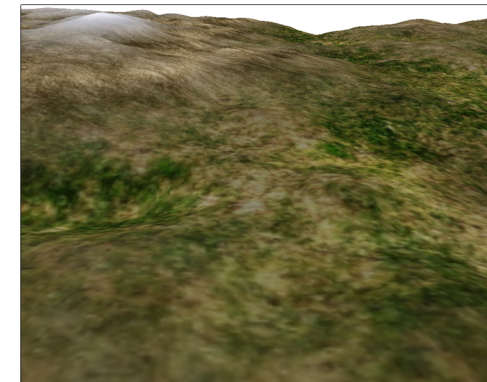
A low resolution version of the grid mesh seen from above. Notice how the polygons gets bigger further from the camera (in the top of the image).

## Benefits

- Low bandwidth usage. As the static grid mesh can reside completely in the graphics memory it is only matrices and constants that needs to be sent to the graphics card.
- Constant frame rate. As it is always the same number of triangles that are rendered to the screen, this technique results in a very constant frame rate.
- Infinite terrain! Because the height map is represented as a texture in the vertex shader the texture sampler can easily be set to wrap (repeat) or mirror to make the terrain features continue infinitely.
- Scalability. This technique is simple to scale by simply change the number of rows and columns in the grid mesh.

## Drawbacks

- Requires a graphics card that supports texture sampling in the vertex shader (1).
- Can cause problems with very jerky height maps. When the distance between the vertices far away from the camera becomes large, aliasing can occur due to subsampling



Terrain rendered with the described technique.



Terrain rendered with wireframe using the described technique

1) ATI only supports texture sampling in the vertex shader this from DirectX 10 generation graphics cards and above. NVIDIA, however, have supported this since the GeForce 6 series. The Xbox 360 support it too.