Nicolas Galoppo von Borries

COMP236 Spring 2004 Project Proposal:

View-Dependent Displacement Mapping

I will be implementing (part of) *"View-dependent displacement mapping" (VDM)*, presented at SIGGRAPH'03 by Wang et al. In order to efficiently render detailed surface mesostructure, VDM models surface displacements along the viewing direction. Unlike traditional displacement mapping, VDM allows for efficient rendering of *self-shadows*, *occlusions* and *silhouettes* without increasing the complexity of the underlying surface mesh, as it is performed in screen-space. VDM is based on per-pixel processing, and with hardware acceleration it can render mesostructure with rich visual appearance in real time.

The algorithm takes as input a textured height field sample and converts this to a VDM representation, which is a function $d_{VDM}(x,y,\theta,\varphi,c)$ encoded in a high dimensional texture. The distances are obtained by a ray-casting preprocess. During rendering, the parameters are obtained on a per-vertex basis and interpolated. Then, silhouettes, real texture coordinates, shadows and shading are computed for each pixel.

The principal techniques that were covered in COMP236 and that are employed in the algorithm are: *perspective-correct texture mapping, bump mapping, meshes, ray casting algorithms* and *illumination models* for shading.

Minimally, I would like to have a working version of the algorithm for 128x128 texture patches, without doing the compression that is discussed in the paper. The basic steps will include meshing the height field, determining d_{VDM} by casting rays for 32x8 viewing directions and no curvature, implementing the rendering part in a vertex and a fragment program. If time permits, I will account for curvature. Performing PCA for data compression is also a possible extension. If I really don't know what to do after that, I can always look into the issue of interreflections.