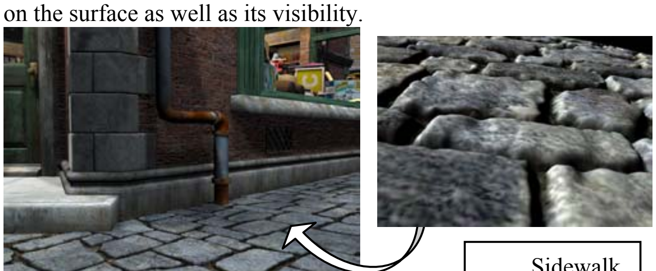
Practical Dynamic Parallax Occlusion Mapping

Natalya Tatarchuk*
ATI Research, Inc

Introduction

We present an improved parallax occlusion mapping algorithm for dynamic real-time lighting of surfaces including soft shadows, a directable LOD system and increased precision of the critical height field-ray intersection computation. We simulate the effects of motion parallax with perspective-correct depth at any viewing angle. Per-pixel ray intersections of a height field with view and light directions are performed to determine correct displacement on the surface as well as its visibility.



Sidewalk close-up view

Parallax Occlusion Mapping on a cobblestone sidewalk

The algorithm can be applied to arbitrary polygonal surfaces for dynamic lighting with self-occlusion and interpenetration effects. We generate smooth soft shadows during lighting computation. Shader complexity is automatically controlled with our explicit level-of-detail system. The technique dynamically controls the sampling rate for per-pixel ray-tracing based on the surface information and viewing orientation. This method performs well for animated objects and supports dynamic rendering of height fields for a variety of interesting displacement effects. The algorithm is scalable for a range of consumer grade GPU products. It exhibits a low memory footprint and is well designed for existing art pipelines for games and effects rendering.

Parallax Occlusion Mapping

We encode surface displacement information in a tangent-space normal map with a scalar height map. Inherent local planarity of the tangent space for any point on an arbitrary surface provides an intuitive mapping for surface detail information. We perform all calculations for height field intersection and visibility determination in tangent space, and compute illumination in the same domain. Rendering is performed using the programmable pixel pipeline of the GPU.

The effect of motion parallax [1] for a surface can be computed by applying a height map and offsetting each pixel in the height map using the geometric normal and the eye vector. For true perspective simulation, one would need to displace every pixel in the height map using the eye vector and the geometric normal. The core idea of the presented algorithm is to trace the pixel being currently rendered in reverse in the height map to determine which texel in the height map would yield the rendered pixel location if in fact we would have been using the actual displaced geometry.

The input polygonal mesh acts as the "reference" plane from which the displaced surface is extruded. Unlike traditional displacement mapping, the displaced surface is extruded downward from the reference polygons.

*e-mail: natasha@ati.com

During parallax occlusion mapping on a per-vertex basis:

- We compute the viewing direction V, the light direction L in world space and transform them into tangent space using the per-vertex tangent basis vectors (resulting in V_{ts} and L_{ts})
- If desired, we compute the parallax offset vector P to determine maximum visual offset in texture-space

On a per-pixel basis:

- We ray-cast the view ray V_{ts} along the interpolated parallax offset vector P to compute ray – height field profile intersection point and sample the height field profile along the parallax offset vector to determine the correct displaced point on the extruded surface This yields the texture coordinate offset necessary to arrive at the desired point. We add this amount to the original sample coordinates to yield texture coordinates t_{off}
- Starting at t_{off} we cast the light ray L_{ts} to perform visibility computation, sampling the height profile along the way to determine occlusion. This results in a visibility coefficient v for computing shadowing and self-occlusion
- We shade the given pixel using its attributes, the normal from the normal map at t_{off} , the light vector L_{ts}, and v

Height Profile Intersection Computation

We present a higher precision result for the intersection of the extruded surface profile with the viewing direction. This is accomplished by dynamically scaling the sampling rate based on the surface orientation and viewing direction relationship. We approximate the height field profile as a piecewise linear curve and intersect rays with each linear segment for a more robust result.

Soft Shadows Computation

During lighting computation, we convolve the light ray – height field intersection samples with a heuristic kernel approximating the surface thickness. This generates smoothly blurred shadows without any edge filtering artifacts.

Level-of-Detail Control System

In order to incorporate this fill-intensive technique in practical scenarios, we designed an explicit level-of-detail system for automatically controlling shader complexity. We compute the current mip map level directly in the pixel shader and use this information to transition between different levels of detail from the full effect to simple bump mapping. We gradually reduce the sampling rate during the level transition layer. There is no associated visual quality degradation as we move into lower level of detail and the transition appears quite gradual. This LOD system provides a significant rendering optimization and smooth transitions between the full parallax occlusion mapping and a simplified representation without visual artifacts such as ghosting or popping. Since all calculations are performed per pixel, the method robustly handles extreme close-ups of the object surface thus providing an additional level of detail management.

References

[1] KANEKO, T., TAKAHEI, T., INAMI, M., KAWAKAMI, N., YANAGIDA, Y., MAEDA, T., TACHI, S., "Detailed Shape Representation with Parallax Mapping", *ICAT* , 2001