Oregon State University

Paper Analysis Project

Hyuntaek Oh

ohhyun@oregonstate.edu

Due: Mar. 17, 2025

1. **What is the general theme of the paper you read? What does the title mean? What are they trying to do? Why are they trying to do it? (I.e., what problem are they trying to solve?)**

According to paper “Practical Parallax Occlusion Mapping for Highly Detailed Surface Rendering”, Natalya Tatarchuk explained the approach of how to make the objects in realistic games be as rich and detailed as possible and how to make the game truly immersive.

The title of it means that Parallax Occlusion Mapping (POM) can be a solution to rendering detailed, realistic surface in a practical, efficient way, compared to existing approaches.

However, there are three problems to achieve realism in games: rendering complex surface topology is needed to overcome an age-old rendering balancing act, vertex transformation and the amount of memory usage is costly and rendering accuracy issues that preserve depth at all angles, dynamic lighting, and self-occlusion.

The author provides a solution called “Parallax Occlusion Mapping (POM)” to these problems. By utilizing this method, five benefits are expected: Per-pixel ray tracing of a height field in tangent space, correctly handling complicated viewing phenomena and surface details such as motion parallax and complex geometric surfaces, soft self-shadowing, flexible lighting model, and adaptive Level of Detail (LOD) system to maximize quality and performance. These benefits can be seen in Figure 1 and 2.

With POM, the program can render extreme high details for various surfaces such as brick buildings, wood-block letters for the shop sign, and cobblestone sidewalk. In addition, it is possible to incorporate multiple lighting models. For instance, diffusion lighting, simulated wet materials, integrated view-dependent reflections, and shadow mapping are the possible effects of using POM.

The reason why the author is trying to address the realistic rendering is that she would like to offer realistic experience in games, and the current problems she mentioned can be handled by computer graphics technologies. The realistic experience can also provide fantastic entertainment and make players dive deeply into the game plays.

2. **You *can* copy and paste images or graphs from the paper into your paper, but**

not as a way to have to write less to hit the 5-page requirement.

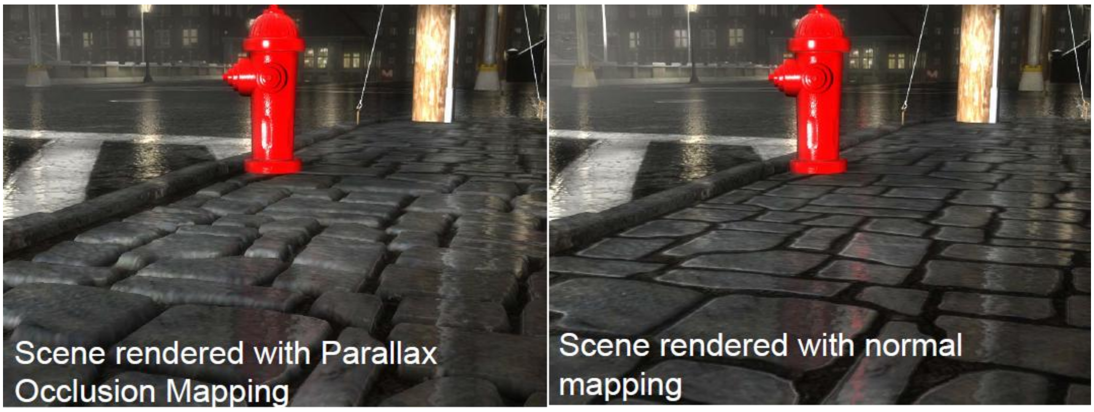

Figure 1. Comparison Parallax Occlusion Mapping with Normal Mapping

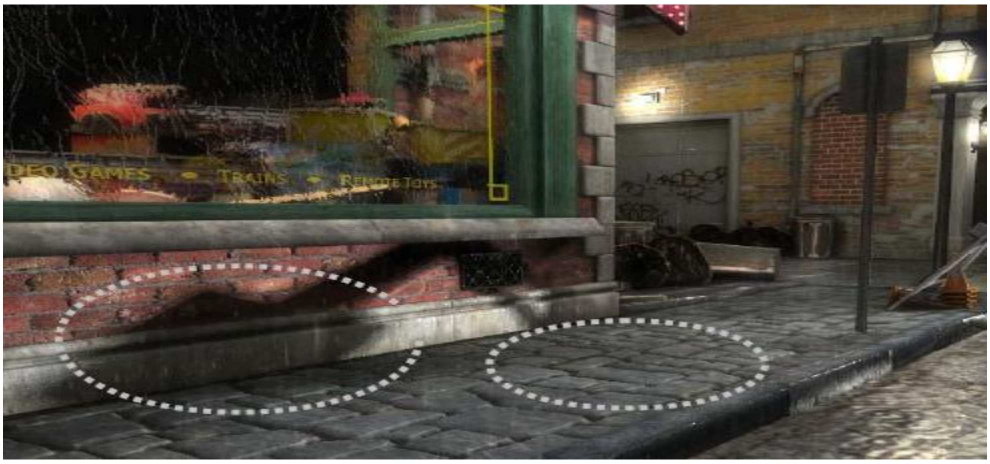

Figure 2. The lighting effect of Parallax Occlusion Mapping

3. **Who are the authors? Where are they from? If you can tell, what positions do they hold? Can you find out something about their backgrounds?**

Natalya Tatarchuk is a graphics engineer and a rendering enthusiast who is a research engineer, a lead engineer on ToyShop, 3D Application Research Group, and ATI Research, Inc [1]. The author was the Graphics Lead and an Engineering Architect at Bungie before joining Unity. She has been encouraging sharing in the game graphics community for several decades, largely by organizing a popular series of courses such as Advances in Real-time Rendering, Rendering Engine Architecture, and Open Problem in Real-Time. She holds an M.S. in Computer Science from Harvard University with a degree in Computer Graphics and B.A. degrees in Mathematics and Computer Science from Boston University [2].

4. What did the authors do?

The writer explained the details of Parallax Occlusion Mapping (POM) to solve the difficulty to make objects realistic. The contributions of POM are four points: Increase precision of height field (ray intersections), dynamic real-time lighting of surfaces with soft shadow, directable Level of Detail (LOD) control system with smooth transitions between levels, and motion parallax simulation with perspective by correcting depth.

Since all computations are done in tangent space, displacement can be applied to arbitrary surfaces. By introducing parallel polygonal surface and using input texture coordinate, displaced point on surface can be found from the view of ray. This can be implemented with per-vertex and per-pixel. In the case of per-vertex, it computes the eye and light direction in tangent space, meaning that it can also compute the parallax offset vector as an optimization, which is interpolated by the rasterizer. Per-pixel is related to the ray that casts the view ray along the parallax offset vector as well as height profile intersection for occlusion computation to determine the visibility coefficient. This method is derived from the idea of using a distance map to arrive at the extrude surface in the paper "Per-Pixel Displacement Mapping with Distance Functions."

For surface-ray intersection, linear search is used to take advantage of fast performance. However, simple linear search can produce visible aliasing artifacts that make object lack smoothing, and thus parallax occlusion mapping rendered with just linear search needs the high precision height field intersection computation.

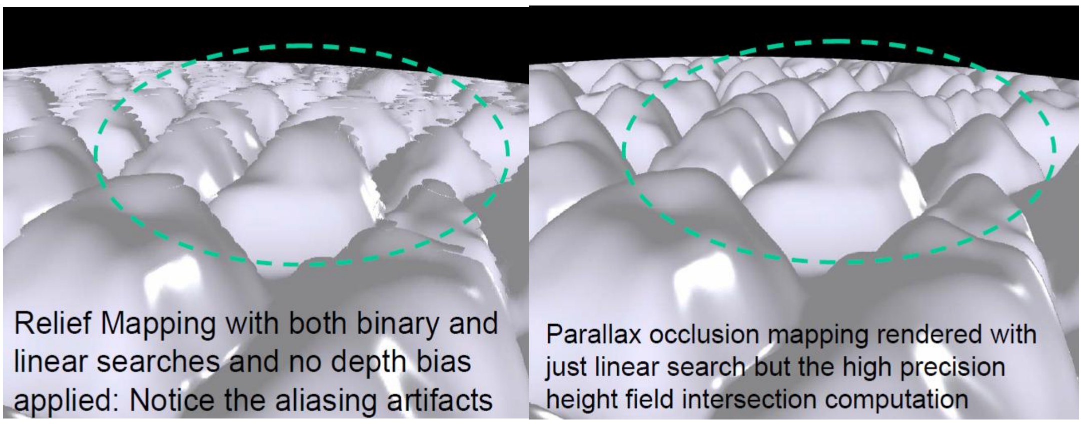

Figure 3. The difference of using high precision height field intersection computation

Sampling-based algorithms tend to show aliasing, so there is a need to dynamically adjust the sampling rate for ray tracing as a linear function of angle between the geometric normal and the view direction ray.

Self-occlusion shadows are implemented by using the extruded surface point that derived from view ray, producing a point on polygonal surface. There are two types of shadows: hard shadows and soft shadows. From hard shadows computation, simply determining whether the current feature is occluded yields hard shadows. On the other hand, soft shadows can be computed by filtering the visibility samples during the occlusion computation and not computing shadows for objects not facing the light source.

Surface illuminating uses the computed texture coordinate offset to sample desired maps (albedo, normal, and details). These parameters and visibility information can be applied to any lighting model as desired such as Phong and Computing reflection/refraction. What's more, for many illumination effects, simply diffuse lighting with base texture is great choice because diffuse only suffices for many effects. Glossy specular can be utilized to reduce specularity in the valleys.

Adaptive Level-of-Detail (LOD) system computes the current mip map level. For the furthest LOD levels, it renders using normal mapping with threshold level. LOD increases the sampling rate as a function of the current mip map level as the surface approaches the viewer. In transition region between the threshold LOD level, it blends between the normal mapping and the full parallax occlusion mapping.

The cost of POM is much cheaper than actual geometry. 8 to 50 samples are tested, using 1100 polygons with parallax occlusion mapping. The memory usage is 79K vertex buffer, 6K index buffer, 13Mb texture (3Dc) (2048 x 2048 maps), and thus the total amount of the memory usage is less than 14Mb. The cost of actual geometry, on the other hand, covers 1.5 million polygons with normal mapping, and the footprint is 31Mb vertex buffer, 14Mb index buffer. The total cost of the geometry is 45Mb. The frame rate is also considered to compare the performance of POM with the geometry. POM yields about 245 frame per second on ATI Radeon hardware whereas the geometry produces 32 frame per second on the same hardware.

Due to these benefits and superior performance, POM is a great method to deal with making objects and expressions in games authentic.

5. What conclusions did the paper draw?

Parallax Occlusion Mapping (POM) is a powerful method for rendering complex surface details in real-time, providing higher precision height field with ray intersection computation, self-shadowing, and Level of Detail (LOD) rendering technique for textured scenes. POM also produces excellent lighting results and has modest texture memory usage, compared to normal mapping. Moreover, it efficiently uses existing pixel pipelines for highly interactive rendering and supports dynamic rendering of height fields and animated objects. From these benefits, players can feel and enjoy their games in a more realistic way.

6. **What insights did you get from the paper that you didn't already know?**

For implementing authentic objects, the idea of using polygonal surface is ingenious for me. As explained in section 4, only displacement lighting would be able to show some unrealistic, unintended lines that look like bump layers, making objects unmatched to real-world texture. However, parallax occlusion mapping using polygonal surface with high precision approximation takes advantage of the points from view-ray and extruded surface in tangent space, resulting in smoothing and generating natural shape in real-world objects. This mathematical approach provides the effect of making 3-D objects in a computer appear as if they exist in the real world.

7. **Did you see any flaws or short-sightedness in the paper's methods or conclusions? (It's OK if you didn't.)**

Parallax Occlusion Mapping (POM) has a tradeoff between speed and quality. Less samples means more possibility for missed features and incorrect intersections, resulting in stair stepping artifacts at oblique angles. Additionally, silhouettes are not computed correctly. Art can be authored to hide this artifact or use vertex curvature data and texkill in the pixel shader to clip pixels at the silhouettes in an alternative way at the expense of memory and extra computation.

Correct depth output is an indispensable part of parallax occlusion mapping. Simply using parallax occlusion mapping will yield incorrect object intersection. It may display object gaps or cur-throughs. Depth will be computed for the reference surface. This can be solved if each pixel's Z value updates when computing the displacement. It compensates for simulated extruded surface and uses the height field value and the reference plane Z value to compute correct depth. The performance will be affected by

Z, which is output from the pixel shader.

When handling curved surfaces with POM, there is a need to approach them carefully since the computation is in tangent space. If vertex curvature can be encoded vertex data, current algorithm needs to be extended to use that data to improve height-field intersection using the curvature. This reduces aliasing and potential misses at steep grazing angles.

The current approach to providing realistic effects is based on the "pushing down" fashion, not up. The view and ray direction are not always from upside to downside, meaning "pushing down", sometimes it needs to be from downside to upside in many different types of game playing. In this case, parallax occlusion mapping may not work well even though the view and ray direction is based on the tangent space.

8. If you were these researchers, what would you do next in this line of research?

If I were these researchers, I would consider how to make more realistic shadows after applying parallax occlusion mapping using polygon surface with high precision. Since the shadows after POM are not handled so much, it would be the next stage of consideration for realistic materialization. In the paper, the contents address soft and hard shadows, but it does not look enough to represent real-world examples. In my opinion, using shadow map for calculating shadows depending on the degree of lights can be a solution to shadowing issues.

Reference

[1] N. Tatarchuk, "Practical Parallax Occlusion Mapping For Highly Detailed Surface Rendering Practical Parallax Occlusion Mapping Practical Parallax Occlusion Mapping For Highly Detailed Surface For Highly Detailed Surface Rendering Rendering." Accessed: Mar. 10, 2025. [Online]. Available: https://advances.realtimerendering.com/s2006/Tatarchuk-POM.pdf

[2] https://cesium.com/guests/natalya-tatarchuk/