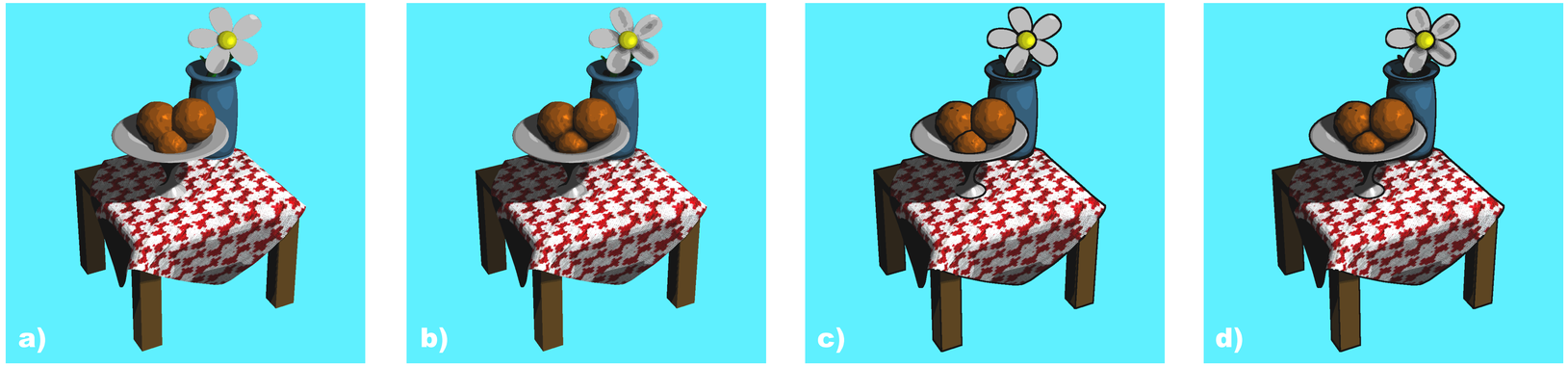
Screen Space Ambient Occlusion for Non-Photorealistic Rendering

Brett Lajzer
Rensselaer Polytechnic Institute

Daniel Nottingham
Rensselaer Polytechnic Institute

Barbara Cutler
Rensselaer Polytechnic Institute



A scene rendered with different effects turned on. In a) the scene is rendered with only cartoon shading. SSAO is added to the image in b), and outlining is added to added to a) to produce c). In d), the scene has been rendered with both effects. The SSAO and outlining together serve to bring out silhouettes and internal details that are lost in a).

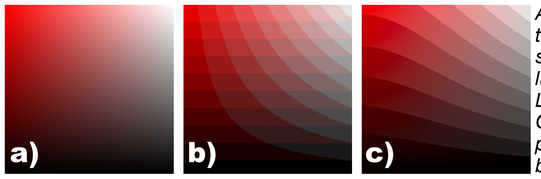
Abstract

Methods for non-photorealistic rendering of 3D scenes have become more popular in recent years and are leveraged in computer animation and games to artistically enhance or minimize the detail and textures within a scene. We present a combination of screen-space ambient occlusion, which is a simple estimate of global lighting, with traditional non-photorealistic rendering methods. We also present a simple method for giving cartoon-shaded qualities to arbitrary textures. The results of the method are illustrated on a variety of models, demonstrating its appropriateness for games and other interactive rendering applications. Because there is no precomputation, the method allows for fully dynamic scenes.

Previous Work

Screen-space ambient occlusion (SSAO) is an approximation of ambient occlusion, an implementation of which was developed by CryTek for their game Crysis to facilitate real-time rendering. This version computes ambient occlusion for each pixel visible on the screen by generating random points within a fixed distance in the hemisphere around that pixel and determining occlusion for each point by comparing its depth to a depth map of the scene. A pixel is occluded if its depth is greater than the depth of the nearest visible object at that point, unless the difference in depth is greater than the sample radius. This method results in a noisy, high-frequency image, which must be blurred for a smooth result.

The field of non-photorealistic rendering encompasses a wide variety of visual appearances. We focus on cartoon rendering. This includes outlining in black the silhouettes and other important edges on objects based on depth and normal discontinuities. Our outlining method captures silhouette edges, sharp ridges, and sharp valleys. Cartoon shading is used to threshold lighting intensity to several discrete values.



*A gradient in a) is thresholded in RGB color space in b), and in the luminosity component of L*a*b* color space in c). Color information is preserved in c), rather than being lost as in b).*

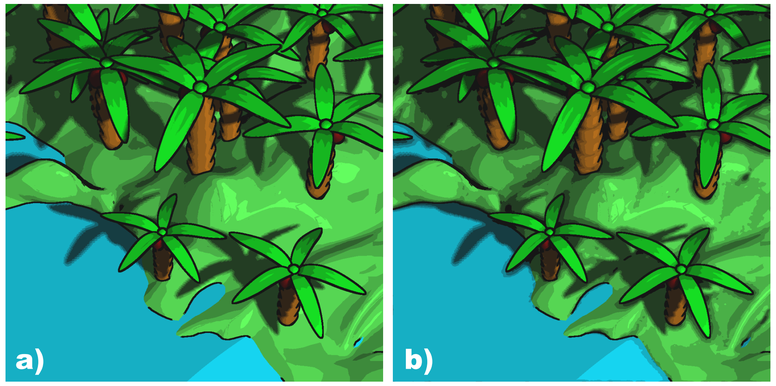
Implementation

We implemented our technique in OpenGL and GLSL. Most of the work is done in the fragment shader stage of the pipeline through several passes, rendering intermediate results to textures.

Render Stages:

1. Render normals and depth buffer to textures
2. Render light-space depth buffer to texture (for shadowing)
3. Use depth buffer from (1) to render SSAO to texture
4. Blur SSAO texture from (3)
5. Render lit geometry, combining toon shading, SSAO, and shadows
6. Render outlines using normals and depth from (1)

When rendering the shaded geometry, we multiply the SSAO values rather than add them, and threshold textures to a few discrete luminosities. Texture thresholding is done on the L* component in L*a*b* color space rather than on the RGB values of the texture. This prevents the loss of color information that occurs when thresholding in RGB color space.



A forest scene rendered with toon shading, outlining, and shadows. The scene has been rendered without SSAO in a), and with it turned on in b).

(Right) A Buddha statue demonstrating SSAO. a) shows the raw SSAO output, which is blurred to produce b). c) and d) show a toon-shaded rendering of the statue (no outlining), with SSAO enabled in d) and disabled in c).

Results

As can be seen from the images in this paper, the application of Screen-Space Ambient Occlusion to cartoon shading has remarkable results. One can see that even without the thick outlines, SSAO can provide visual cues that help separate foreground and background. Adding outlines gives a distinct visual style that further enhances the details of the scene. This technique brings out details that are missed by both the Phong shading and the outlines, and it provides visual cues in the form of soft shadows/halos around objects. Our implementation renders models with 215,000 triangles at 69 frames per second at 1024x1024 resolution on a nVidia GeForce 9800 GT, and scales well with the number of pixels on screen.