

Watercolor Rendering



Description

Watercolor rendering is one of the many types of Non-Photorealistic Rendering (NPR). This aims to render images that appear to have been painted, versus other types of rendering that aim for realism. While watercolor paintings in real life are too complex to fully simulate, there are a few details that can be approximated to give a close replica. These include ragged edges between different adjacent colors in the scene, as well as the paper texture showing through the paint.

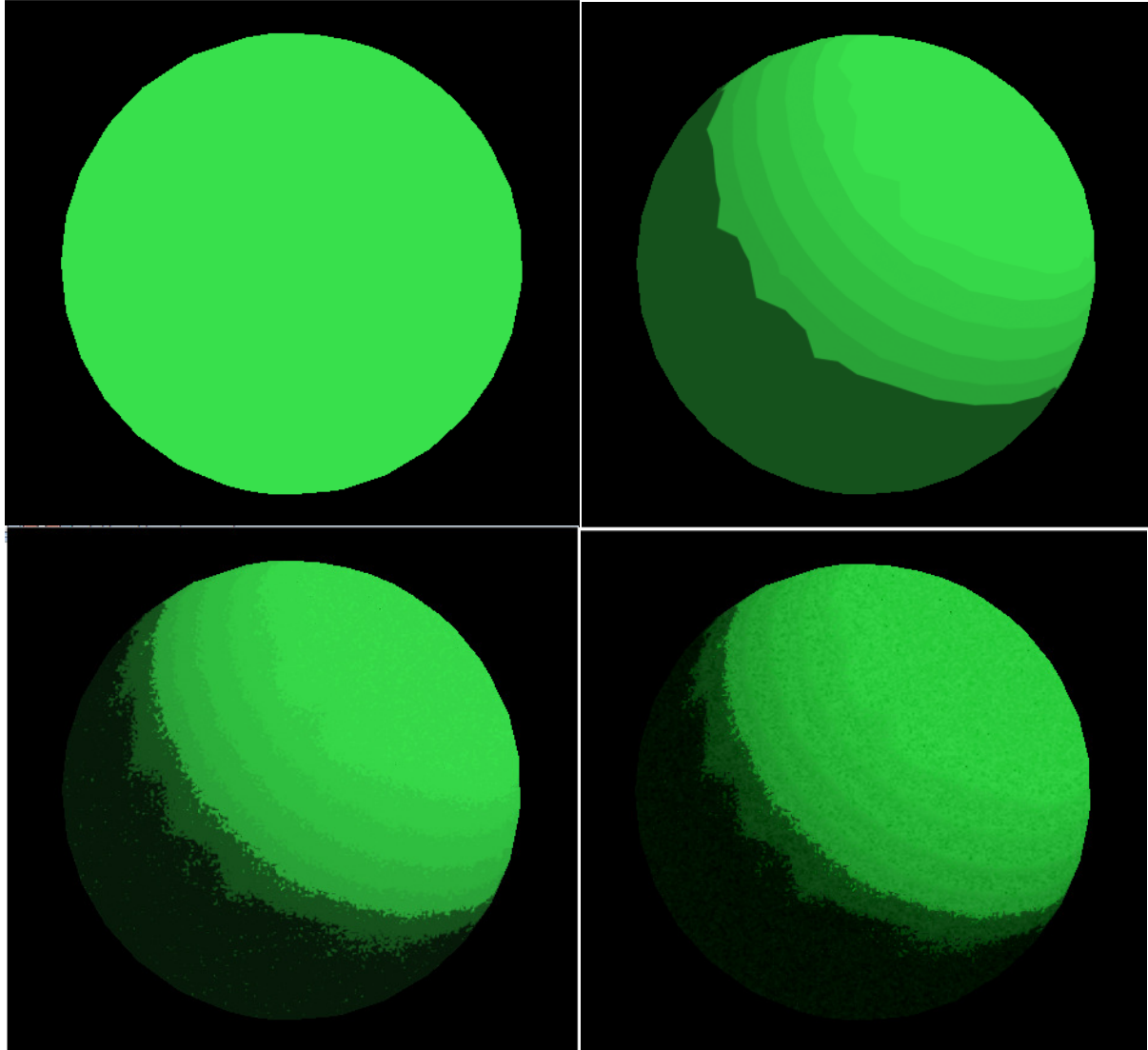
Development

This algorithm closely follows toon shading, but adds in a few details to give a paint-like effect. First, the vertex shader receives light and material information from the OpenGL application, and calculates color as normal using ambient, diffuse, and specular light. This is calculated in the range $[0, 1]$. Next, the RGB values of the light are averaged, returning a single float $[0, 1]$ that will be passed to the pixel shader.

The inputs to the fragment shader are a 1D colorband texture, a 1D granularity texture, a 2D paper texture, a color or scene texture (color will be used in this example), and the float calculated from the vertex shader. The fragment shader first calculates the texture coordinate to use on the paper texture, based on the screenspace of the current pixel. A fraction of the color returned from the paper

texture is added to the float. The new float indexes the colorband texture, and the color returned is multiplied with the input color. This gives us the uneven borders that is seen between adjacent colors. Next, the paper texture itself is subtracted from our current color, and edge detection is performed to enhance the graininess of the paper.

Progression



Top left: The input color.

Top right: Input color with lighting and a colorband texture creates traditional toon shading. Gouroud shading is used here, hence the uneven borders.

Bottom left: Adding a paper texture to the colorband texture coordinate creates paint flecks along the colorband borders.

Bottom right: Subtracting the paper texture as a whole, along with slight Sobel edge detection creates the final image.

References

1. E. Lei, C. Chang. "Real-Time Rendering of Watercolor Effects for Virtual Environments." National Tsing Hua University. 2004. <http://www.cs.nthu.edu.tw/~chunfa/pcm2004.pdf>
2. R. Rost. "OpenGL Shading Language." Addison-Wesley. 2006.