

Effective Toon-Style Rendering Control Using Scalar Fields

Alex Harvill
Pixar Technical Memo #07-08
Pixar Animation Studios

An illustration of Gusteau comes to life and introduces a new facet of the film *Ratatouille*. To do this, a technique was needed to convert an animated 3D character into a moving 2D illustration. Existing renderman shaders based on normal and depth maps were difficult to control. Post processing per gprim id information robustly captures object outlines and has been previously extended with texture maps to feature some surface details. However, texture maps introduce artifacts, which are quite objectionable during post processing.

I extended the id based approach using scalar fields to produce well-behaved 2D edge/contour lines from 3D models. Specifically, mesh face ids and per-vertex line weights were rendered and processed to create lines and specify line thickness, respectively. This new method is fast, looks good, and is easy to control.

1 Regions and Lines

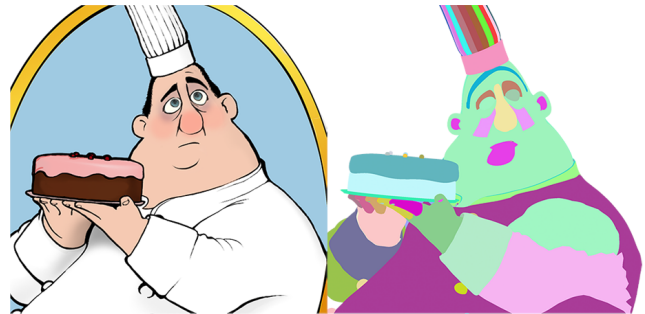
The normal/depth approach relies on controls that operate wholesale on an image. Because it is not possible to change one line without affecting the rest, the problem is recast in terms of surface regions to gain explicit control over line position. Lines in the reference illustration(fig. 1a) were found to closely correspond to mesh edges. Therefore the regions between edges correspond directly to groups of mesh faces. Each mesh region is assigned an id using Maya membership editing tools intended for rigging. When possible, region membership was copied from rigging data, such as the points that make up the index finger.

Next, images are rendered in which each pixel perfectly captures 1 integer face id(fig. 1b). The face varying class of scalar fields must be used so that no interpolation is applied to the ids during rendering. This image is convolved using an edge detection kernel yielding a 1 pixel region outline(fig. 1c). Because the regions are intrinsically bound to mesh vertices, generated outlines exhibit remarkably better frame-to-frame coherence than the normal/depth/texture results. Region membership can easily be amended to fix objectionable line placement.

2 Thickness and Integration

For local control of line thickness a vertex scalar field, or map, is created using Maya's point weighting tools(fig. 1d). Weight values were rendered with vertex interpolation to directly control dilation of region lines in Shake. As weights approach 0 lines taper off to nothing. A weight of 0 removes an extraneous line entirely(fig. 1e). A weight of 1 corresponds to an arbitrary maximum dilation of 10 pixels(fig. 1f). In the reference image, interior lines thicken markedly when crossing onto Gusteau's exterior. Edge detection is performed on the character's alpha channel to form an outline and an alternate weight map is used to set this contour's thickness(fig. 1g).

Because of the limited scope of the project, implementation difficulties were avoided largely by workaround. Although motion-blur and shadows could have been approximated, the director omitted them. Using scalar fields for region ids and line thickness is a solid approach to feature film quality toon rendering.



Illustration(a)

Region Ids(b)



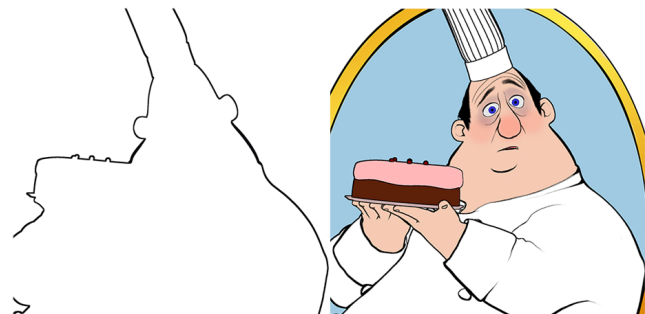
Region Boundaries(c)

Thickness Map(d)



Thickness == 0 Removed(e)

Dilation by Thickness(f)



Contour(g)

Final Composite(h)