SpecVar Maps: Baking Bump Maps into Specular Response

Patrick Conran*
Industrial Light + Magic

SpecVar maps provide a way of baking the effect that bump maps have on the specular response of a surface into a modified specular response that is independent of the bump map.

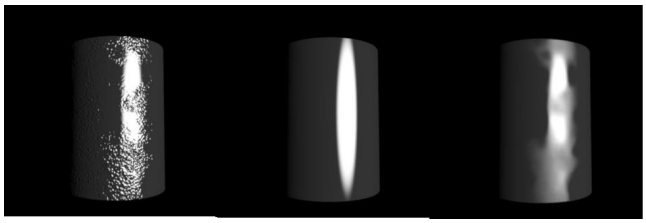

Figure 1: A cylinder with a bump map, with a filtered-out bump map, and with a specVar map

The reason for doing this is to overcome the filtering effect that occurs in prman when objects get small in the screen.

With a colour or a gain type map that modulates, for instance, the diffuse or the specular gain across a surface, texture filtering of the map is usually a fairly accurate representation of the average of the pixels being filtered. However, a bump map is actually modifying the orientation of the surface, and small changes can have a dramatic effect on the shading of the surface, in particular specular reflection. If you filter the bump map, you don't get an average of the specular light hitting the surface within the filtered region. You get the average of the bump map (e.g. typically for a bumpy/rough surface the average is zero), which then causes bumps which would previously have caught specular highlights to now miss them, and it is now this smoothed out surface whose specular response is then averaged. The net result is that if you have a surface roughened by a bump map, as this surface gets smaller in screen the apparent roughness of the surface will get less and less.

This is particularly true for skin. At very closeup views the bump-mapped skin cells and pores cause an apparent broadening and dimming of the average specular level. When this bump map is filtered out with distance, the specular quality of the skin becomes sharper and more pronounced.

SpecVar maps provide a way to maintain the apparent specular quality of the surface regardless of the filtering effects.

1 Technical Approach

The bump map causes each micropolygon in the rendered model to be reoriented relative to the original geometric surface normal. To create the specVar maps we first measure the effect of the bump map on the specular response of every micropolygon. We then average this response over a local region of a size that is much larger than the features of the bumpMap. Finally we find new values for our specular function that best fit our measured average specular response, using a minimization technique, and then bake out the modified specular gain, plus the solved specular parameters.

*e-mail:patc@ilm.com

2 Caching the Specular Response of the Surface

The first stage in creating specVar maps is to measure and store the specular response for each micropolygon due to its reorientation. In effect, we shine a light down the geometric normal, and sample the specular response of the micropolygon over the 90 degrees from geometric normal to tangent. (Note: we don't actually shine a real light. All we are doing is putting vectors of light direction(LN), camera direction (VN) and surface normal (N) into our specular function and storing the result). This table of specular samples (we sampled at 3 degree intervals, so 31 samples per micropolygon) is then stored in a point cloud.

3 Finding the Local Average Specular Response

The tables of specular samples are then averaged over a user-defined radius, to give a table of average specular samples over an area sufficiently larger than the level of detail of the features of the bump map we are trying to capture.

4 Solving for Best-Fit Specular Response

The value of the first sample in each table (light and camera pointing down the surface normal, which corresponds to the maximum specular response) is now stored as the specVar gain. i.e. The value is the amount that the specular gain has been reduced due to the effect of the bump map. The resulting pointcloud of tables of average specular samples is then normalized, and solved for the best fitting values in our specular function.

5 Rendering using the SpecVar Maps

The specVar values are baked into texture maps. In the beauty render, the specVar values for specSharpness and specSize are absolute values that will override your original shader specular parameter settings. The specVar gain value is a multiplier that will attenuate, not replace, your other shader specular gain settings. SpecVar maps should only be used when your surface is small enough in screen that the bump map filtering is having a detrimental effect. The transition from bump map plus original specular values, to no bump map plus specVar parameters can either be a user-controlled switch in the shader, or an empirically determined mix based upon some metric in the shader that determines how large in screenspace it is. The ideal is to fade inversely proportional to the amount that the bump map is being filtered out.