CSE 252A: SUV Color Space, Filtering

Lecturer: David Kriegman

Scribed by Owen Jow on October 25, 2018

1 Uncalibrated Photometric Stereo

In calibrated photometric stereo (known lighting directions), we can exactly estimate the $n \times 3$ matrix $\mathbf{B}$ of surface normals scaled by albedo. In uncalibrated photometric stereo (unknown lighting), we can only recover $\mathbf{B}$ up to a *Generalized Bas-Relief* transformation that determines how squished the surface is, and how slanted the surface is.

2 SUV Color Space

The SUV color space allows us to apply Lambertian photometric stereo to non-Lambertian surfaces. Namely, we can pull out specularity and get the underlying diffuse color. To do this, we use a property called the dichromatic (two-color) reflection model which holds for a number of different materials. The basic idea behind this model is that specular reflected light is the color of the light source,[1] whereas diffuse reflected light is a combination of the color of the light source and the diffuse material color.

So we can decompose our colors into diffuse and specular colors. Writing this as an equation, the brightness for channel k at a particular location is

$$I_k = (D_k f_d + S_k f_s(\theta))\ \mathbf{n} \cdot \mathbf{l}$$

where D_k is the diffuse color, S_k is the specular color, f_d is the diffuse BRDF, f_s is the specular BRDF, and θ is the angular geometry parameters.

This can be rewritten once for each of the three R, G, B color channels:

$$\begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix} = (f_d\ \mathbf{n} \cdot \mathbf{l}) \begin{bmatrix} D_r \\ D_g \\ D_b \end{bmatrix} + (f_s(\theta)\ \mathbf{n} \cdot \mathbf{l}) \begin{bmatrix} S_r \\ S_g \\ S_b \end{bmatrix}$$

$$\mathbf{I} = (f_d\ \mathbf{n} \cdot \mathbf{l})\ \mathbf{D} + (f_s(\theta)\ \mathbf{n} \cdot \mathbf{l})\ \mathbf{S}$$

where $\mathbf{D}$ is the material color and $\mathbf{S}$ is the light source color. The image color lies in the span of these colors. Note that $\mathbf{D}$ varies over the image, while $\mathbf{S}$ does not.

We want to convert RGB to a new color space which separates out the diffuse and specular colors. To do this, we define a transformation of color, i.e. a mapping from a color in RGB space to a color in SUV space. This mapping is a simple rotation of color space:

$$[\mathbf{R}]\mathbf{I}_{RGB} = \mathbf{I}_{SUV}$$

[1] Consider a mirror.

for rotation matrix

$$[\mathbf{R}] = \begin{bmatrix} — & \mathbf{S} & — \\ — & \mathbf{U} & — \\ — & \mathbf{V} & — \end{bmatrix}$$

The first row is the normalized specular color (the color of the light) $\mathbf{S}$. The other rows $\mathbf{U}$ and $\mathbf{V}$ are normalized vectors which are orthogonal to $\mathbf{S}$ and orthogonal to each other (so spanning a plane orthogonal to $\mathbf{S}$). The form of this matrix shows that we are projecting points into the $\mathbf{S}$, $\mathbf{U}$, $\mathbf{V}$ coordinate frame.

(The rotation matrix should take $\mathbf{S}$ in the RGB frame and make it $(1, 0, 0)$.)

In this new color space, it is easy to separate out the diffuse color of the image. If we apply the matrix $[\mathbf{R}]$ to our $\mathbf{I}_{RGB}$ from before, we see

$$\begin{bmatrix} — & \mathbf{S} & — \\ — & \mathbf{U} & — \\ — & \mathbf{V} & — \end{bmatrix} [(f_d \ \mathbf{n} \cdot \mathbf{l}) \ \mathbf{D} + (f_s(\theta) \ \mathbf{n} \cdot \mathbf{l}) \ \mathbf{S}] = (f_d \ \mathbf{n} \cdot \mathbf{l}) \begin{bmatrix} 0 \\ \mathbf{U} \cdot \mathbf{D} \\ \mathbf{V} \cdot \mathbf{D} \end{bmatrix} + (\text{something}) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

So we have one channel with specularities, and two channels without specularities. After we drop the $\mathbf{S}$ channel, which contains both specular and diffuse components, we are left with only diffuse components (as I_u and I_v). And we can apply photometric stereo just to these channels – perhaps a combination of these channels like $\sqrt{\mathbf{U}^2 + \mathbf{V}^2}$, although there are multiple ways to formulate it.

In summary, the SUV color space is a rotation of the RGB color space, i.e. we can define a rotation matrix $[\mathbf{R}]$ *which takes* $[r, g, b]$ *vectors to* $[s, u, v]$ *vectors, i.e. the axes used to represent* r, g, b *and now they represent* s, u, v*, i.e. the rotation matrix turns* $\mathbf{S}$ *(the specular color) into the* $(1, 0, 0)$*-axis.*

3 Filtering

- In image filtering, we set each pixel according to an arbitrary function of its neighborhood.
- *Shift-invariant*: if you shift the input, the output is shifted the same way.
- As templates, filters look like the effects they're designed to find. This is because the application of a filter boils down to dot products between each neighborhood and the filter.

3.1 Denoising

We can apply filters in an attempt to recover the underlying value of each pixel based on its neighborhood. For example, taking a pure average of noisy values reduces the effect of the noise. One way to denoise is via an isotropic 2D Gaussian filter (s.t. weights come from the Gaussian formula), which is just weighted averaging (like all of these linear filters) with more weight toward the center.

The Gaussian filter, along with e.g. the box filter, is *separable*. This means that $f * I = f_H * f_V * I$, i.e. you can split f into two filters[2] and do row/column filtering separately. This might be more efficient: **assuming "same" zero padding**, it averages out to *six multiplies plus four additions* for what was previously one application of a 3×3 filter (*nine multiplies and eight additions*). In this example, you are now centering both a row filter and a column filter over each pixel.

[2]For example, isotropic Gaussian equals horizontal Gaussian convolved with vertical Gaussian.