Term Project Report
Normal Map Estimation using Model Geometry and Texture
Team: Avengers

Jang Wonjong, Lee Dahun, Jacob Morton
20140337, 20130221, 20172327
Computer Science and Engineering, POSTECH

Figure 1. Example of a face model with **Left:** Geometry only, **Right:** Geometry + Normal Map

1. Introduction

1.1. Motivation

There are several easy to use tools to generate 3D models and textures, like 3D Studio Max, Maya, blender, and Adobe Photoshop. It is not too difficult to program or load these into OpenGL as well. The problem of using 3D geometry and Texture alone is that it can produce flat looking renderings, especially if the geometry is less detailed than the texture. This is typically the case, as texturing can hide simple or flawed geometry. There is more we can do with texture mapping than just wrapping the geometry with an RGB image. We can also add bump maps, normal maps, displacement maps, and even subsurface scattering to make the rendering more realistic. But these additional mappings are difficult to produce and are often hand-made or produced in multi-step procedures, i.e. baking. Using detailed normal maps in addition to texture can improve the perceptual quality of 3D reconstructed scenes and objects while being illuminated by a light source. We wish for there to be a simple automated procedure that can take an RGB picture with a geometric prior and produce a high-quality normal map. Such a tool would be useful in entertainment and 3D Reconstruction.

1.2. Project Description

Inspired by Shape-From-Shading, we propose a tool for estimating a normal map from a given texture. We extract interpolated normals from a geometric prior. We use off-the-shelf intrinsic image decomposition to estimate the albedo texture and remove shading from a single photograph. Given the albedo and interpolated normals, we use spherical harmonics to alternatively estimate the environment illumination first and then compute a per-pixel normal map second. We can then apply this normal map to the model in addition with texture during the texturing process. We demonstrate this by using a fragment shader in OpenGL.

This project requires knowledge in Intrinsic Image Decomposition, Spherical Harmonics (SH), least-squares optimization, and illumination and normal mapping in OpenGL using fragment shaders.

2. Development Environment

Operating System:	Windows 10
IDE:	MS Visual Studios 2017
Libraries:	OpenGL, FreeGlut, GLEW, GLM, chumpy

3. Overview

3.1. Intrinsic Image Decomposition

We also need a good approximation of the albedo texture $\rho(x, y)$, but it is difficult to acquire as we need an image without shading. We will pre-process the RGB image to remove shading $s(x, y)$ using widely available open-source intrinsic image decomposition[4]. intrinsic image decomposition is used to decompose an image into reflectance (albedo) and shading images, and is typically formulated as

$$I(u, v) = \rho(u, v)s(u, v), \tag{1}$$

Where $I(u, v)$ is the input raw image with shading from a lit environment.

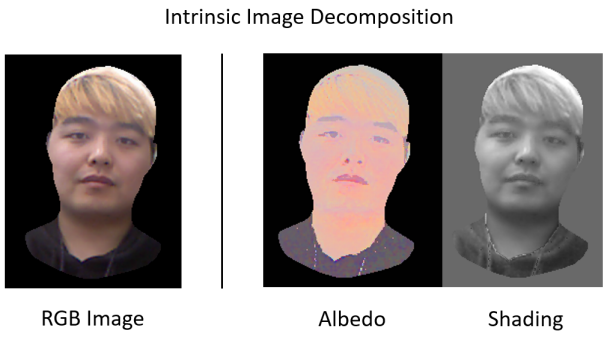

Figure 2. Decompose image I into albedo ρ and shading s

3.2. Spherical Harmonics

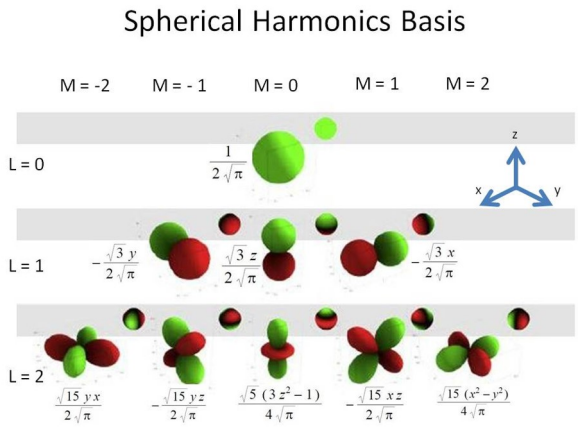

Figure 3. L the band (order), M is the degree (basis)

Spherical harmonics are a set of harmonic functions defined on the surface of a sphere. Many surfaces are mostly diffuse, so we assume Lambertion diffuse surfaces. We will be using the Real and Cartesian form of 2^{nd}-order spherical harmonics. The first band represents ambient energy, the remaining bands represent more and more complex diffuse reflectance. The Lambertian factor a_l acts as a blur kernel reducing the energy at higher orders.

Spherical harmonics is a common method for illumination and geometric refinement. The spherical harmonics reflectance model is widely adopted in many areas of computer graphics, from rendering to 3D reconstruction. We can estimate diffuse illumination parameters of an image scene, given some rough geometric assumptions. Illumination is assumed to be multiple light sources at an infinite distance. We can also measure energy at each pixel on an image to determine the pixel's normal vector when illumination is roughly known. Spherical harmonics is commonly used in single image Shape-from-Shading methods due to these properties. The first 3 bands of spherical harmonics are most typically used, which represent reflectance sufficiently in most cases. We will use the geometric prior to extract the initial normal map N_{ref}. Given N_{ref} and $\rho(x, y)$, we will solve the least-squares problem to estimate lighting parameters L_l^m,

$$I = \rho \sum_{l,m} a_l L_m^l Y_m^l(N_{ref}), \tag{2}$$

where $Y(n_x, n_y, n_z)$ is the spherical harmonics basis function. After lighting has been estimated, we will solve for the normal map by refining N_{ref} to produce N, the recovered lighting parameters L_l^m, and ρ,

$$I = \rho \sum_{l,m} a_l L_m^l Y_m^l(N). \tag{3}$$

We will render the resulting normal map N in OpenGL using a fragment shader. We represent a per-pixel formulation in uv-coordinates,

$$I(u, v) = \rho(u, v) \sum_{l,m} a_l L_m^l Y_m^l(n(u, v)). \tag{4}$$

The set of real spherical harmonics basis functions in Cartesian space Y_m^l for the first 3 bands are defined [5] as

$$\begin{aligned}
Y_0^0 &= c_0 \\
Y_1^{-1} &= c_1 n_x \\
Y_1^0 &= c_1 n_y \\
Y_1^1 &= c_1 n_z \\
Y_2^{-2} &= c_2 n_x n_y \\
Y_2^{-1} &= c_2 n_x n_z \\
Y_2^0 &= c_2 n_y n_z \\
Y_2^1 &= \frac{c2}{2}(n_x^2 - n_y^2) \\
Y_2^2 &= \frac{c2}{2\sqrt{3}}(3n_z^2 - 1),
\end{aligned} \tag{5}$$

where coefficients c_l are denoted as,

$$c_0 = \frac{1}{\sqrt{4\pi}}$$
$$c_1 = \frac{\sqrt{3}}{\sqrt{4\pi}} \quad (6)$$
$$c_2 = \frac{3\sqrt{5}}{\sqrt{12\pi}}.$$

The Lambertian diffuse reflectance factors a_l for the first three bands are defined as,

$$a_0 = \pi$$
$$a_1 = \frac{2\pi}{\sqrt{3}} \quad (7)$$
$$a_2 = \frac{2\pi}{\sqrt{8}}.$$

The Illumination parameters L_l^m are coefficients which represent multiple unknown light sources. In 2^{nd}-order Spherical harmonics, we will need 9 (greyscale) or 27 (RGB) coefficients. We will use the first 3 bands and greyscale similarly as [5].

4. Pipeline

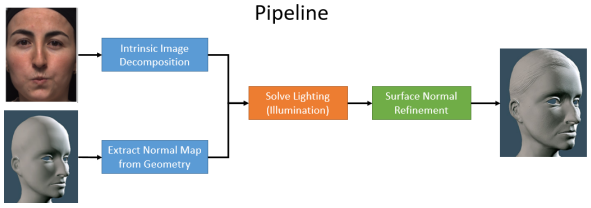

Figure 4. Pipeline

Our pipeline begins with a pre-aquired geometry assumption and an image I. We first use intrinsic image decomposition to obtain the albedo ρ. We use the interpolated normals of the assumed geometry as the prior normal map N_{ref}. Next we give the image I, albedo ρ, and the prior normal map N_{ref} to our SH illumination and normal refinement solver, which outputs the refined normal map N. For face models we fit a parametric model (PCA) with facial landmarks[6] and project the model into image space to rasterize the interpolated normals and map uv-coordinates on the corresponding image. We first solve for the unknown illumination parameters L_l^m by minimizing the energy

$$\min_L I(u,v) - \rho(u,v) \sum_{l,m} a_l L_{lm} Y_{lm}(n_{ref}(u,v)). \quad (8)$$

With illumination known, we next refine the prior normal map by minimizing

$$\min_n I(u,v) - \rho(u,v) \sum_{l,m} a_l L_{lm} Y_{lm}(n(u,v)). \quad (9)$$

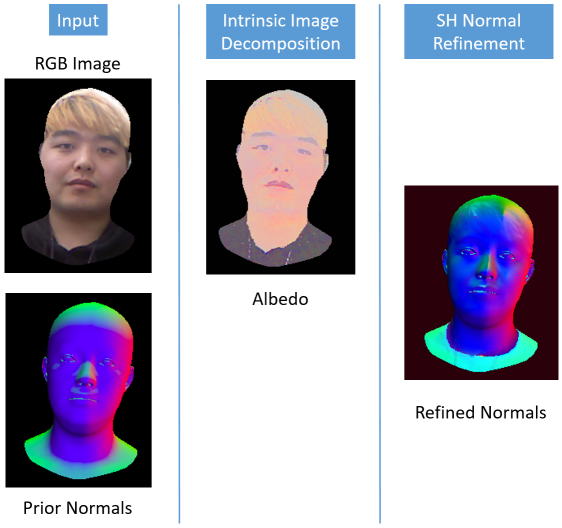

Figure 5. Refinement Pipeline: Inputs are raw RGB image I and normals extracted from target geometry N_{ref}. First, intrinsic image decomposition of raw RGB image to get albedo ρ. Then solve for lighting using prior normal map N_{ref} and albedo ρ. Finally, refine Normals N using albedo ρ and solved lighting parameters L_l^m.

5. Visualization

6. Results

6.1. Normal Refinement

Illumination is solve Normal Refinement Result

6.2. Rendered Results

The improved reflectance is clearly visible in the illuminated untextured (uniform color) face models. The rendered untextured models using interpolated normals clearly lack fine surface details. Rendered untextured models using the refined normal map have far more reflected detail added, including nose shape, improved eye shape, more details in all areas of the face and neck, with the addition fo hair and clothing not represented by the model. There is some noise, this is caused by some loss in converting 32-bit floating point normals to unsigned char. We hope to fix this conversion issue, and we are currently seeing improvements in this area. Also regularization can also help prevent some noise, to smooth sudden changes in surface normals.

7. Team Member Roles

See Table 1.

Name	Roles
WonJong	Intrinsic Image Decomposition
Dahun	Rendering: Model, Texture, and Normal Map
Jake	Spherical Harmonics Normal Map Refinement

Table 1. Team Member Roles

References

[1] R. Green. *Spherical Harmonic Lighting: The Gritty Details*. GDC 2003.

[2] Rich Forster. *Spherical Harmonics for Beginners*. https://dickyjim.wordpress.com/2013/09/04/spherical-harmonics-for-beginners/

[3] Igor Goldvekht. *Shape-From-Shading*. https://github.com/IgorGee/Shapes-From-Shading

[4] Sean Bell, Kavita Bala, Noah Snavely *Intrinsic Images in the Wild* ACM Transactions on Graphics (SIGGRAPH 2014) https://github.com/seanbell/intrinsic

[5] Ira Kemelmacher-Shlizerman, Member, IEEE, and Ronen Basri, Senior Member, IEEE *3D Face Reconstruction from a Single Image Using a Single Reference Face Shape* IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 33, NO. 2, FEBRUARY 2011.

[6] Volker Blanz, Thomas Vetter *A Morphable Model For The Synthesis Of 3D Faces* SIGGRAPH 1999