Specularity and Shadow Interpolation via Robust Polynomial Texture Maps

Mark S. Drew[1], Nasim Hajari[1],
Yacov Hel-Or[2]
& Tom Malzbender[3]

[1]School of Computer Science,
Simon Fraser University, Canada

[2]Department of Computer Science,
The Interdisciplinary Centre, Israel

[3]Media and Mobile System Lab,
Hewlett-Packard Laboratories, CA

BMVC 2009

Overview

Introduction

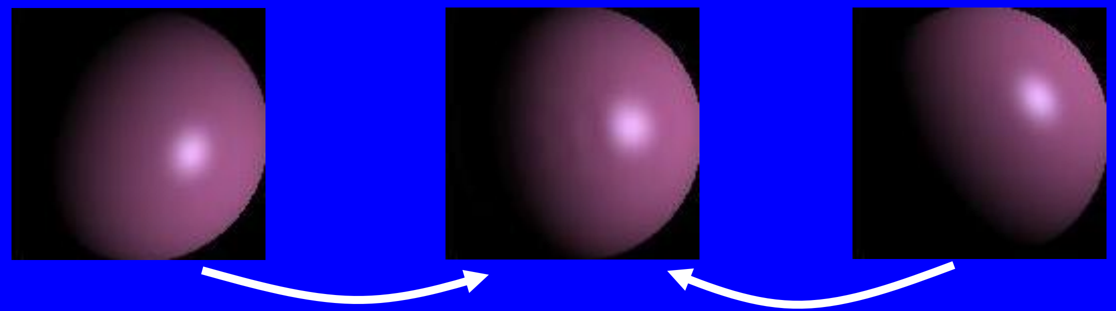
PTM Model

Outlier Identification

- Colour, Albedo and Normal

Specularity and Shadow

The Aim: Shadow and Specularity Interpolation



Known lighting

Interpolation of two lights

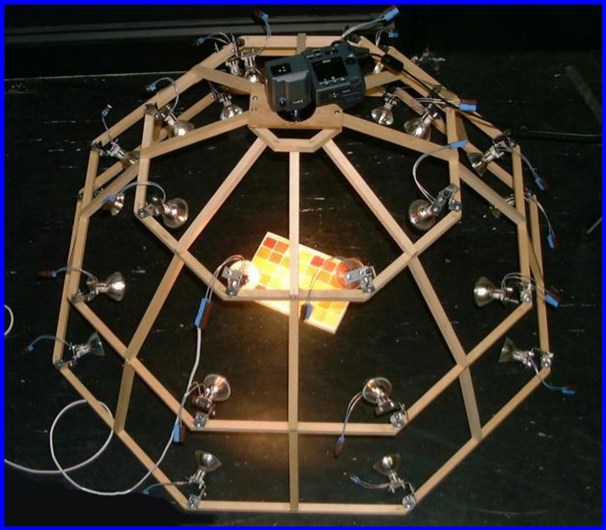
Known lighting

We would like to interpolate shadow and specularity to see what the image would look like under a new, non-measured lighting direction.

Methodology

1. Solving the PTM model in a robust version which leads to identification of outliers and inliers.
2. Generating surface normal and surface albedo using inliers.
3. Modelling specularity and shadow using RBF regression over outliers in hand.

What is PTM?



n images of a scene from *n* different lighting directions.

PTM (Polynomial Texture Mapping) is a pixel based method for modelling dependency of luminance L on lighting.

L = R+G+B

PTM Model

- PTM is a generalization of Photometric Stereo (PST).
- PTM performs a non-linear polynomial regression.
- Polynomial Regression can better model intricate dependencies due to self shadowing and interreflections.
- The aim is finding vector ***c***, Regression Coefficients, at each pixel position.

$$\begin{bmatrix} p_0(a^1) \\ p_0(a^2) \\ \dots \\ p_0(a^n) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_6 \end{bmatrix} = \begin{bmatrix} L^1 \\ L^2 \\ \dots \\ L^n \end{bmatrix}$$

$$p_0(a) = (u^2, v^2, uv, u, v, 1)$$
$$a = \{u, v, w\}$$

Modified PTM Model

We would like to use robust regression to find the coefficients

Modified PTM we define as follows:

$$p(a) = (u, v, w, u^2, uv, 1) \qquad w = \sqrt{1 - u^2 - v^2}$$

Note:

Suppose we happen to have a Lambertian surface; then get normal $\boldsymbol{n}$ and albedo α exactly:

$$n' \equiv \{c_1, c_2, c_3\}$$

$$\alpha = \| n' \|, \; n = n' / \alpha$$

⇒If at a pixel the collection of images are Lambertian + shadow + spec., **<u>using robust</u>** approx'ly still get correct regression coefficients.

Why Robust PTM?

- Robust Regression helps in identification of outliers and inliers, <u>automatically</u>.
  - Outliers are shadow and specularity.
  - Each pixel labelled as matte, shadow or specularity.
- Knowing inlier pixel values helps in recovering more accurate surface normal and albedo.

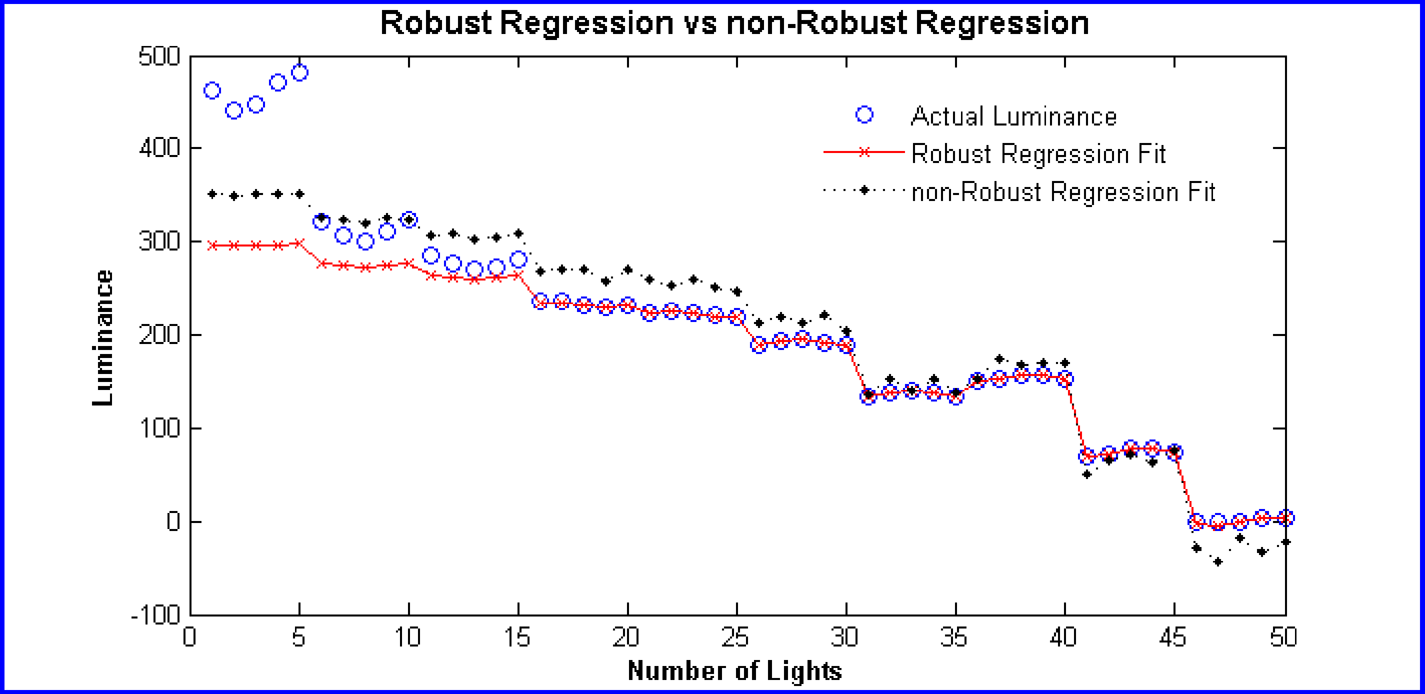
Robust Regression

- LMS (Least Median of Squares) finds an estimation for coefficients by minimizing the median of squared residual.

$$c = LMS(p, L)$$

- Breakdown point is 50%.
- Output: Set of regression coefficients and tripartite set of *n* weights $\{w^0, w^+, w^-\}$ at each pixel position: inlier, specular-outlier, shadow-outlier.
- Exclude outliers in calculating coefficients.

Robust vs. Non-robust Fit for Matte Component



PTM Re-lightening
Generating Matte Component

- Using Modified PTM.
- Calculate regression coefficient vector for each pixel.
- For any new lighting direction ***a'***:

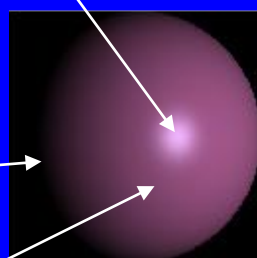
$$L' = \max [p(\mathbf{a'})\mathbf{c},0]$$

Non-Robust PTM and Robust-PTM on a Synthetic Sphere

Specularity

Shadow

Matte



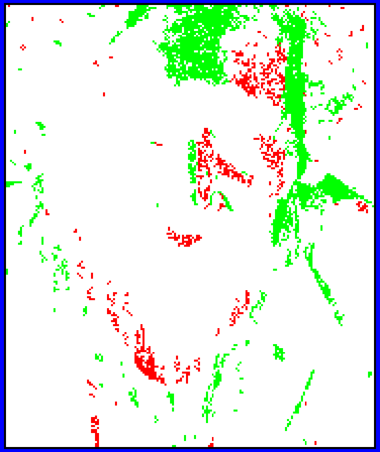
Synthetic sphere with Phong illumination

Regenerated *matte* sphere using non-robust PTM

Regenerated *matte* sphere using <u>robust</u> PTM

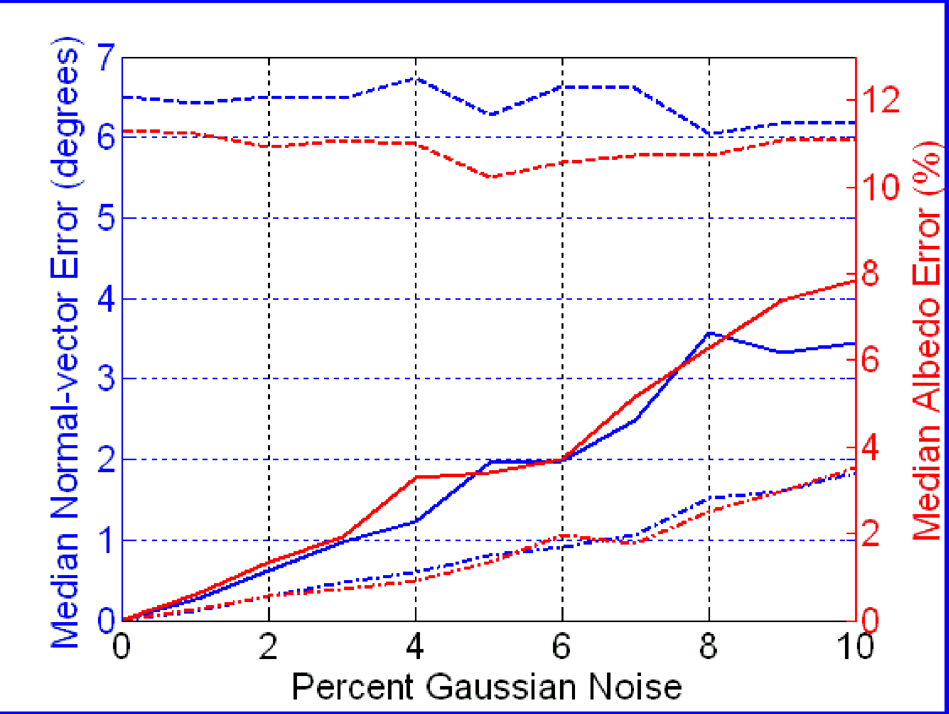
Shadow and Specularity Identification

- Specular highlights: an outlier with positive residual
- Self or cast shadow: an outlier with negative residual



Red: Shadow
Green: Specularity

Surface Normal and Surface Albedo



PST

$$n' = A^{+}L$$

Robust PST

$$n' = A(\mathit{Inliers})^{+}L(\mathit{Inliers})$$

$$\alpha = \|n'\| \quad n = n'/\alpha$$

PTM Coefficients

$$n' = \{c_1, c_2, c_3\}$$

Chromaticity

- We know inliers (matte values) at each pixel position.
- The chromaticity is RGB triple divided by Luminance
- A good estimate for chromaticity is:

$$\chi = median(\{R, G, B\}(Inliers) / L(Inliers))$$

Shade and Sheen

- The robust PTM model only accounts for a basic matte reflectance.
- We know the lights that lead to specularity and shadow at each pixel location.

$$\varsigma(w^{+}) = L(w^{+}) - L'(w^{+}), \quad \varsigma(\neg w^{+}) = 0$$ Sheen Contribution

$$\sigma(\neg w^{+}) = L'(\neg w^{+}) - L(\neg w^{+}), \quad \sigma(w^{+}) = 0$$ Shade Contribution

Modeling Specularities and Shadows

- To model the dependency of specularity and shadow on lighting direction, we use two sets of RBF.

$$\varsigma(\vec{a}) = \alpha + \vec{\beta}^{T} \vec{a} + \sum_{i=1}^{n} \gamma_i \varphi(\| \vec{a} - \vec{a}_i \|), \quad \sum_{i=1}^{n} \gamma_i = 0, \quad A^{T} \vec{\gamma} = 0$$

Polynomial term

RBF Coefficients

Gaussian RBF

Interpolation

- Using pre-calculated RBF coefficients to generate shade and sheen.
- Using PTM model to generate matte contribution.
- The model that describes luminance at each pixel is then:

$$\hat{L}(i) = L'(i) + \varsigma(i) - \sigma(i)$$

Adding Back Colour

- Colour is Luminance times Chromaticity.
- The estimated chromaticity is not accurate in sheen area.
- Thus we assume specular chromaticity is the chromaticity of the maximum luminance over all pixels.
- Then colour is:

$$Color = \chi \times \hat{L} + (\chi_{spec} - \chi) \times \varsigma$$

Reconstruction of input image

The PSNR for reconstructed input image ranges from 27.54 to 50.43 with median of 35.61

Original image

Reconstructed image

Interpolation Results

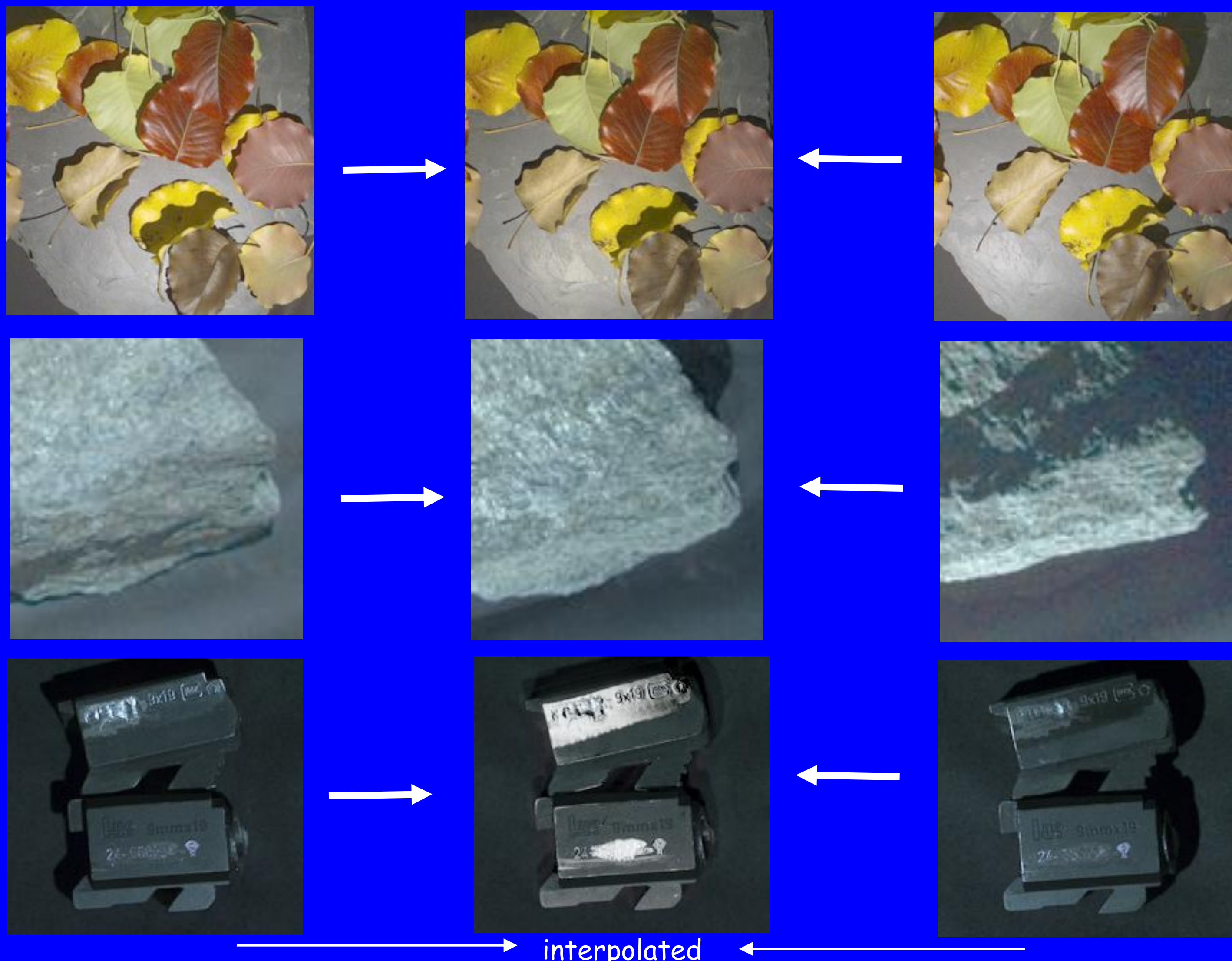
U=0.22, V=0.35

Interpolated *angle*

U=0.0, V=-1.0

interpolated

Interpolation Results



Summary

We have presented a method to interpolate specularity and shadows within the PTM framework

The method uses robust regression to separate matte, highlight and shadow contribution

We used PTM to model matte and RBF to model specularity and shadow

We also showed how to recover chromaticity and combine colour information with luminance to get an accurate RGB rendering under new lighting

Future Work

RBF framework may not be the best or most efficient approach for modelling shade and sheen ...

… Also the Gaussian base function may not necessarily be the best choice

In future, we intend to apply the methods to artworks, with a view to determining their 3D structures and surface properties.

Acknowledgements

The authors would like to thank Natural Sciences and

Engineering Research Council of Canada and Hewlett-Packard Incorporated