Specularity and Shadow Interpolation via Robust Polynomial Texture Maps

Mark S. Drew[1]
mark@cs.sfu.ca

Nasim Hajari[1]
nha16@cs.sfu.ca

Yacov Hel-Or[2]
toky@idc.ac.il

Tom Malzbender[3]
tom.malzbender@hpl.hp.com

[1] School of Computing Science
Simon Fraser University
Vancouver, B.C., Canada

[2] Department of Computer Science
The Interdisciplinary Center
Herzliya, Israel

[3] Media and Mobile Systems Lab
Hewlett-Packard Laboratories
Palo Alto, CA

Abstract

Polynomial Texture Maps (PTM)[SIGGRAPH 2001] form an alternative method for apprehending surface colour and albedo that extends a simple model of image formation from the Lambertian variant of Photometric Stereo (PST) to more general reflectances. Here we consider solving such a model in a robust version, not to date attempted for PTM. But the main upshot of utilizing robust regression is in the identification of both shadows and specularities automatically, without the need for any thresholds, in a tripartite set of weights for pixels that are labelled as matte, shadow, or specularity. Original images are captured using a hemispherical set of lights, and pixel values across the lighting directions are then labelled as inliers, or outliers of two types. A per-pixel robust regression on luminance is carried out using Least Median of Squares, and automatically-identified outlier pixels are labelled as shadows if they are darker than matte and correspondingly, specular outliers are too bright. Inlier identification generates correct values for chromaticity and for surface albedo and thus matte luminance and colour. Then a robust version of PST, using only PTM inliers, improves estimates of normal vectors and albedo recovered. With specular pixel values over the lights in hand we model specularity using a radial basis function (RBF) regression, and non-specular pixel departures from matte using a second RBF set. Then for a new lighting direction, we can readily interpolate both specular content as well as shadows.

1 Introduction

A strength of Polynomial Texture Maps (PTM) [10], in comparison to a simple Lambertian Photometric Stereo (PST) [20], is that a polynomial regression from lighting directions to observed image values can better model real radiance, and thus apprehend intricate dependencies due to self-shadowing and interreflections. In a typical setup, a camera is mounted at the apex of a hemisphere of lights, and each light is then fired in turn, thus generating a sequence of images. Usually, some 40 to 50 images are used, with the larger the number of images the better.

© 2009. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

Thus PTM is a pixel-based method for modelling dependency of luminance (or RGB in a different embodiment) on lighting with the objective being *relightable* images. At each pixel, a 6-vector of coefficients is calculated using nonlinear least squares over the 2-vector of lighting direction x,y-projected components. Subsequently, re-rendering can take place, e.g. by relighting images using a new lighting direction, by calculating surface normals and thence generating artificial specular highlights, by re-mapping colour, by increasing directional sensitivity to lighting direction in order to enhance contrast, by light source extrapolation, or by artificially varying focus [10].

Here, we are interested in using PTM as a vehicle to carry out *interpolation* of specularities and shadows. To the best of our knowledge robust methods have not to date been applied to PTM, and we use these to be able to accomplish interpolation. As well as using robust regression, this paper moreover shows how outliers can be classified as belonging to two types: either specular highlights, or self- or cast-shadows. Knowledge of inlier pixel values means that recovered surface albedo and chromaticity is robust, in the sense of ignoring outlier contributions and thus more accurately mapping surface reflectance and colour.

Here we alter the polynomial used, in PTM, so as to automatically generate regression coefficients that are exactly correct in the case when inliers are indeed Lambertian. In that case, surface normal and albedo falls out of the regression. In the case of non-Lambertian surfaces this is still almost true; but knowledge of inlier labelling allows us to apply ordinary PST just on inlier pixels, yielding what we found to be superior estimates of surface normal and albedo.

Finally, knowledge of outlier labels means that we can independently model specularity and shadow. Then for a new lighting direction, we can generate pixel values interpolating known values of both; here we use a Radial Basis Function (RBF) interpolation model. In this paper we carry out robust regression on the luminance values, not on R,G,B separately. Since we generate the specularity ς in an interpolated image, separately from the remaining contribution σ, we can then produce a full-colour interpolant image using the luminance times matte chromaticity for the non-specular contribution, plus specular-luminance times the chromaticity for the specular colour.

Results on re-creating the *input* images are shown to have excellent agreement with the originals, over a variety of input sequences. For shadow and specularity interpolation — generating images for non-observed lighting directions — the method is shown to indeed generate sensible results. The main contributions in this paper are (1) application of robust regression to PTM; (2) separate modelling and thus better capturing of shadows and specularities; and (3) specular and shadow interpolation.

In §2, we discuss related work, and in §3 outline PTM, PTM as modified here, and specular and shadow outliers. Section 4 shows how vector chromaticity and scalar albedo, as well as normal-vector estimation, falls out of robust regression, and how these are improved via PST. Interpolation of highlights and of non-matte, non-highlight contributions are set out in §5 in an RBF framework, and §6 gives concluding remarks.

2 Related Work

PTM is formulated as a generalization of the simplest variant of PST [20], wherein no calibration object is used but instead a basic Lambertian model is assumed. PTM moves a linear regression involving lighting directions into a nonlinear, polynomial model of lighting dependence. In this paper, in fact we go back to a regression including a linear part corresponding to all three components of light direction, since if the surface is indeed Lambertian plus outliers due to highlighting and shadows, then a robust version of nonlinear regression will still pick up the correct, linear, Lambertian dependence exactly. Then this paper goes on to model specular and non-specular outlier contributions.

Early efforts at specularity detection in PST [4, 6, 16] used four lights and excluded values yielding significantly differing albedos; and [15] used a similar, 5-light approach. In another 4-source method [21], ambient illumination is explicitly included, as is surface integrability, and robust regression is employed in an iterative fashion to eliminate shadows. However, all these methods depend on eliminating a small number of outlier values per pixel, whereas the present paper can operate correctly even in the presence of up to half the number of values per pixel, minus one, being outlier values, due to the high breakdown point of the Least Median of Squares (LMS) robust regression used [14].

In [17] it was shown that the minimum number of lights for PST for the complete visible surface of any convex object is six. Unlike [15], they argued that simply discarding highest or lowest intensity pixels may lead to information loss so they discard only pixels with doubtful intensities. Since extra lights are better, [19] uses a spherical device to capture 156 images. Since the data is then highly overdetermined, they simply discard the lowest 50% and the highest 5% to aid PST. Here, we avoid thresholds or arbitrarily discarding pixels by relying on LMS outlier detection, which automatically generates outliers [13]. In [11], an iterative scheme is devised for finding albedo and normal, which essentially uses the median of each plus smoothness. In this paper, non-outliers are used with no need for iteration to immediately generate both albedo and normal.

Recently, probabilistic models have been proposed to deal with outliers. The method proposed by Chandarker *et al.* [5] poses shadow detection as an iterative Markov Random Field (MRF) formulation, assigning each pixel a shadow label and assuming neighbouring pixels likely have similar shadow labels.

In [18], 169 images of an object are captured and a probabilistic imaging model is introduced, initialized by considering the 50% brightest intensities as inliers. Whereas the choice of prior directly influences their method, here we do not need any prior information. In a method using only three sources, [7] proposes a simple MRF formulation to identify shadow regions. Since one constraint has been lost due to shadowing, they showed that integrability over the two remaining constraints can still lead to surface geometry reconstruction in the shadow region. In this paper, we can indeed find surface normals, but no surface reconstruction is used.

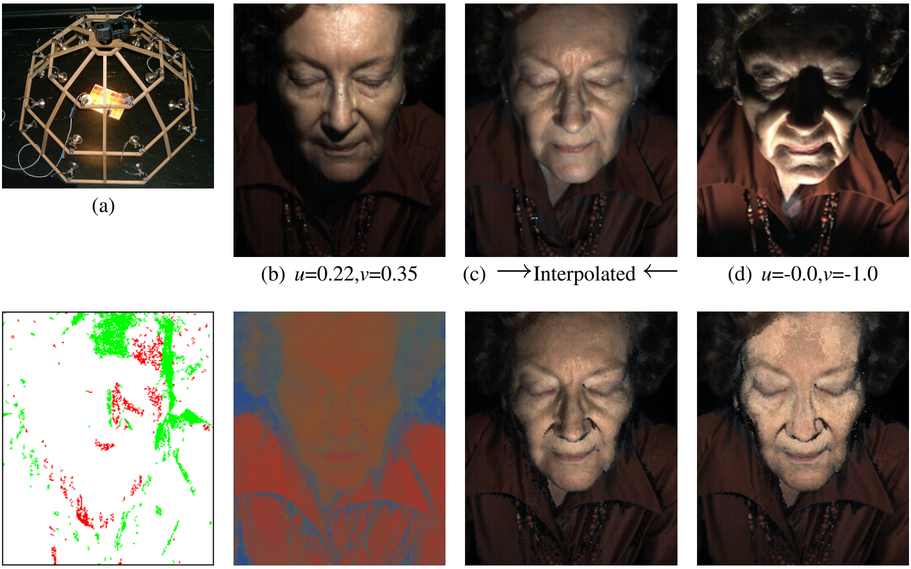
In a generalization of [4], [1, 2] propose a recursive algorithm which eliminates intensities affected by shadows or highlights, based on a least squares error scheme. However, that method can only cope with one highlight plus multiple shadows at a pixel, whereas our algorithm can cope with multiple highlights and shadows.

Here, we extend (an altered version of) PTM [10] to explicitly deal with highlights and shadows. PTM assumes that least squares will effectively be adequate in a polynomial regression, for modelling a smooth dependence of images with a fixed viewpoint on lighting direction. Rather than assuming a basically matte surface, with a polynomial capturing aspects of non-Lambertian reflectance, at each pixel we explicitly model specularities along with non-specular values. Then apprehending these values in an RBF framework, we can go on to interpolate between captured images and arrive at a much more realistic interpolant that correctly displays both highlights and shadows.

3 Image Formation and Polynomial Modelling

3.1 PTM Model

Suppose we have acquired n images of a scene, taken from $I = 1..n$ different lighting directions $\boldsymbol{a}^I$. Fig. 1(a) shows what such a hemispherical lighting frame looks like.



(a)

(b) u=0.22,v=0.35

(c) ⟶Interpolated ⟵

(d) u=-0.0,v=-1.0

(e) Weights for (b): red: shade; green: sheen

(f)

(g)

(h)

Figure 1: (a): Hemispherical dome with multiple, identical lights. (Image due to Ali Alsam of the National Gallery, London.) (b,d): Two inputs. (c): Interpolant for light between (b) and (d). (e): Weights. (f): Recovered chromaticity for (c). (g): Matte from PTM $L = \boldsymbol{p}(\boldsymbol{a})\boldsymbol{c}$. (h): Intrinsic, eq.(6).

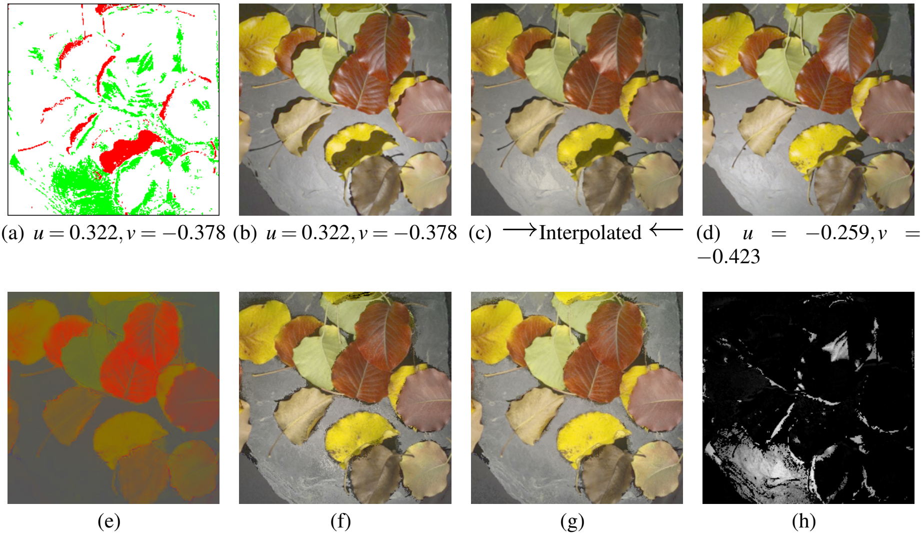
Let each RGB image acquired be denoted $\boldsymbol{\rho}^I$. Suppose we make use of the luminance images instead, $L^I = \sum_{k=1}^{3} \rho_k$. This reduction in dimensionality reduces the computational burden of robust regression and, since we mean to separate out the specularity from the matte component, we can re-insert colour later separately for matte colour and specular colour.

Then a PTM model consists of a nonlinear regression from lighting to luminance via a vector of polynomial terms $\boldsymbol{p}$, with $\boldsymbol{p}$ a function of lighting direction $\boldsymbol{a}$, as follows:

$$\begin{bmatrix} \boldsymbol{p}(\boldsymbol{a}^1) \\ \boldsymbol{p}(\boldsymbol{a}^2) \\ \ldots \\ \boldsymbol{p}(\boldsymbol{a}^n) \end{bmatrix} \boldsymbol{c} = \begin{bmatrix} L^1 \\ L^2 \\ \ldots \\ L^n \end{bmatrix} , \text{ or } \quad \boldsymbol{P}\,\boldsymbol{c} = \boldsymbol{L} \qquad (1)$$

where $\boldsymbol{c}$ is a vector of regression coefficients. Each pixel has its own $\boldsymbol{c}$, and the $\boldsymbol{L}$ vector is the collection of all luminances at that pixel over the n images, for polynomials $\boldsymbol{P}$ for the known lighting directions.

E.g., in the original PTM [10], $\boldsymbol{p}$ is a 6-component set $\boldsymbol{p}_0$ of polynomial terms in $\{u, v\}$,



(a) $u = 0.322, v = -0.378$ (b) $u = 0.322, v = -0.378$ (c) $\longrightarrow$Interpolated $\longleftarrow$ (d) $u = -0.259, v = -0.423$

(e) (f) (g) (h)

Figure 2: (a): Weights. (b,d): Two of inputs. (c): Interpolant. (e): $\boldsymbol{\chi}$ recovered. (f): $\boldsymbol{\rho} = \hat{L}\boldsymbol{\chi}$, $\hat{L} = \boldsymbol{p}(\boldsymbol{a})\boldsymbol{c}$. (g): Intrinsic, eq.(6). (h): Interpolated sheen ς.

the $x-$ and $y-$ projections of (known) lighting vector $\boldsymbol{a}$, with $\boldsymbol{a} = \{u, v, w\}$:

$$\boldsymbol{p}_0(\boldsymbol{a}) = (u^2, v^2, uv, u, v, 1) \tag{2}$$

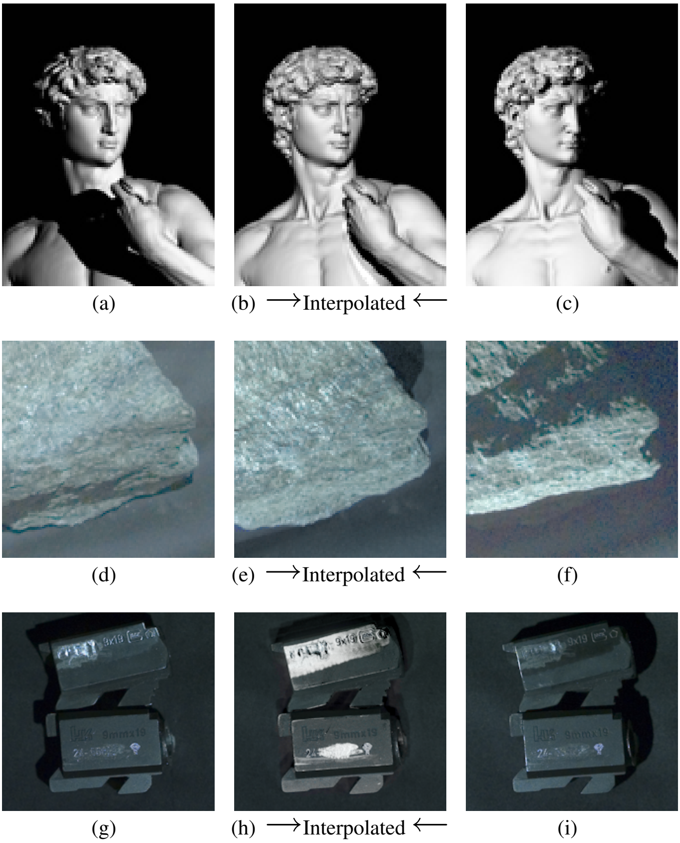
and coefficient vector $\boldsymbol{c}$ is a 6-vector. Thus eq. (1) means that at each pixel we have an $n \times 6$ matrix $\boldsymbol{P}$ of polynomial terms in the lighting directions, times an unknown 6-vector $\boldsymbol{c}$, equaling an observed n-vector $\boldsymbol{L}$ of luminance values at the current pixel.

Here, we start off by firstly replacing polynomial $\boldsymbol{p}_0$ with the following:

$$\boldsymbol{p}(\boldsymbol{a}) = (u, v, w, u^2, uv, 1), \quad \text{where } w = \sqrt{1 - u^2 - v^2} \tag{3}$$

The rationale is as follows: Suppose we indeed happen to have a Lambertian surface; since we mean to use a robust regression to solve for $\boldsymbol{c}$, then regardless of specularities or shadows (assuming at least half plus one of the pixel values at an image location are non-outliers), the regression will just generate the correct surface normal vector, multiplied by a surface albedo times a lighting strength factor. The regression will place zeros in $\boldsymbol{c}$ for the higher order terms. Nevertheless, it is useful to keep a polynomial description, to suit surfaces which are not Lambertian. Note that if we were to use instead $\boldsymbol{p} = \{u, v, w\}$ the method would simply reduce to PST. Here we regress making no assumption about a Lambertian character of the surface, and can reconstruct pixel values without making any such assumption. The reason for 6-D in (3) is simply to retain the same low dimensionality, rather than the 10-D possible for quadratic combinations of $\{u, v, w, 1\}$. (Also, 7-D, with a v^2 term in (3), performed the same in terms of accuracy.)

Within the PTM model then (but modified here as in eq. (3)), for any new lighting direction $\boldsymbol{a}'$ we can utilize the values of $\boldsymbol{c}$ derived from measured images to generate a new image via $L = \max[\boldsymbol{p}(\boldsymbol{a}')\boldsymbol{c}, 0]$.



(a) (b) →Interpolated← (c)

(d) (e) →Interpolated← (f)

(g) (h) →Interpolated← (i)

Figure 3: Left, right: two of inputs. Center: Interpolant for mean of left and right lighting directions.

3.2 Outlier Identification and Mapping to Highlight and Shadow

For a robust regression to replace (1), here we utilize the LMS [13], which generates outliers "on a silver platter", without any intervention. So in this luminance-based variant, regression is

$$\boldsymbol{c} \;=\; LMS\,(\boldsymbol{P}\,,\boldsymbol{L}\,) \tag{4}$$

LMS proceeds by randomly selecting sets of 6 values from $1..n$ and inverting (1) for non-singular sets; the number of sets to try is guaranteed to be much lower than ${}_nC_6$ while still retaining the high breakdown point of LMS [14].

The output of LMS is the set of regression coefficients $\boldsymbol{c}$, plus a set of n weights (labels) $\boldsymbol{w}$ identifying inliers ($w = 1$) and outliers ($w = 0$), thereby excluding some lights at this pixel. But we know here that outliers are generated because (1) L values are too high to suit the model (1) — we take these to be specular contributions; or (2) luminance is too low (or the model generates negative values), for a particular light — these are likely shadow locations. Thus we arrive at a tripartite set of weights $\{w^0, w^+, w^-\}$ at each pixel, with w^0 set for lights generating inlier values, w^+ for specularities, and w^- for shadows. Fig. 1(e) shows weights corresponding to input image Fig. 1(b), w^0 as white, w^+ as green, and w^- as red.

Now we can go on to generate a matte version of the input set of images, or indeed an interpolated matte image, using the regression result $\boldsymbol{c}$, as in Fig. 1(g). We are assured of doing better than standard PTM since we have excluded distracting specularities and shadows from consideration whilst generating coefficients $\boldsymbol{c}$.

4 Colour, Albedo, and Normal

At this point, we already have an advantage of applying a robust method to the problem at hand, *viz.* a more reliable calculation of coefficient $\boldsymbol{c}$. But in fact we also have produced a better grasp of colour, as well. Let us factor each RGB triple $\boldsymbol{\rho}$ into luminance $L = R+G+B$ times chromaticity $\boldsymbol{\chi}$. Luminance will be composed of a scalar albedo α times lighting strength times shading factor s; since we have no way of disentangling lighting intensity from surface reflectance, we shall simply lump both scalars into α. Thus,

$$\boldsymbol{\rho} \ = s\alpha\boldsymbol{\chi}\,, \quad \boldsymbol{\chi} \equiv \{R,G,B\}/(R+G+B) \tag{5}$$

An intrinsic image [3] (for this lighting strength), i.e. surface independent of shading, would then be

$$\boldsymbol{\rho}_{\,intrinsic} \ = \ \alpha\boldsymbol{\chi} \tag{6}$$

I.e., this is what the surface would look like under this light, with shading removed.

Since we have weights w^0 identifying pixels indeed corresponding to the PTM model, the robust regression delivers a reliable estimate of chromaticity. For suppose that at this pixel a set of all chromaticities $\boldsymbol{\chi}$ is given by $\chi_k(i) = \rho_k(i)/L(i)$, $k = 1..3$, for the ith light, $i = 1..n$ (with, say $n = 50$ lights), then we can identify a good estimate for the chromaticity independent of light direction via the median, over each channel $k = 1..3$, of $w^0 = 1$ pixels:

$$\boldsymbol{\chi} \ = \ median\left(\boldsymbol{\rho}\,(w^0)/L(w^0)\right) \tag{7}$$

Fig. 1(f) shows how this recovered chromaticity appears: it is intrinsic colour, without magnitude.

The regression involving (3) automatically generates an estimate of surface normal $\boldsymbol{n}$ and albedo α, by considering the contribution to the first three terms of $\boldsymbol{p}\,(\boldsymbol{a}\,)$:

$$\tilde{\boldsymbol{n}} \ = \ \{c_1, c_2, c_3\}; \quad \alpha = \|\tilde{\boldsymbol{n}}\,\|, \ \ \boldsymbol{n} = \tilde{\boldsymbol{n}}\,/\alpha \tag{8}$$

Fig. 1(h) shows what this Lambertian-based matte intrinsic image looks like: $\boldsymbol{\rho} \ = \alpha\boldsymbol{\chi}$; while Fig. 1(g) shows what the polynomial PTM-based image, $\boldsymbol{\rho} \ = L\boldsymbol{\chi}\,, L = \boldsymbol{p}\,(\boldsymbol{a}\,)\boldsymbol{c}$, looks like: this is what PTM relighting generates, with robust regression (4) slopes $\boldsymbol{c}$ here (i.e., 6-vector slopes as in regression (1)). Since image (g) includes shading, its appearance is more realistic, of course. But note that using robust slopes can generate black output, from slopes that are near zero, since dark values may form a majority at some pixels.

Fig. 4 shows results over increasing percent Gaussian noise, for a synthetic sphere RGB image, with pixel values generated for 50 lighting directions for a Lambertian surface, plus Phong illumination [12] with roughness 1/20, plus noise. Here the solid lines show the results for error in normal vector direction (blue) and albedo (red), from eq. (8). Unsurprisingly, if there is indeed no noise, and the base reflectance is Lambertian, a robust regression ignores the polynomial terms in the regression model and returns regression coefficients proportional to the surface normal since the shading model is normal dotted into light direction, and specularities and shadows do not distract the regression.

Now, suppose we use the robust PTM regression weights w^0, in a robust version of PST wherein we take

$$\tilde{\boldsymbol{n}} \ = \ \left(\boldsymbol{A}\,(w^0)\right)^+ \boldsymbol{L}\,(w^0); \quad \alpha = \|\tilde{\boldsymbol{n}}\,\|, \ \ \boldsymbol{n} = \tilde{\boldsymbol{n}}\,/\alpha \tag{9}$$

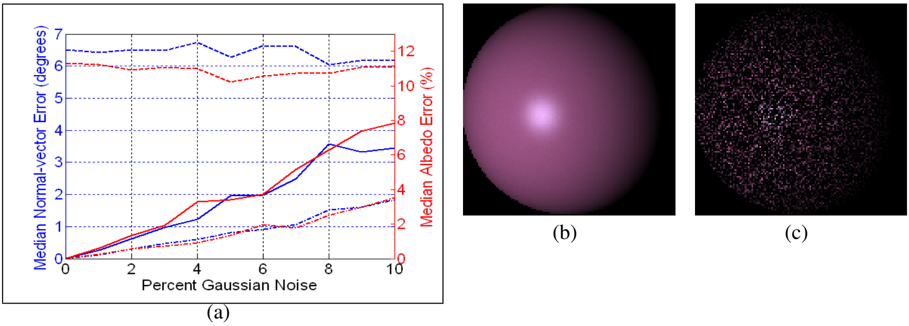


Figure 4: (a): Solid lines: Noise sensitivity for robust PTM estimate of surface normal and albedo. Dot-dashed lines: robust PST – using weights w^0 generated by robust PTM. Dashed lines: standard PST. (b): Synthetic sphere with 10% noise. (c): Noise contribution (positive values shown).

where $\boldsymbol{A}$ is the set of $n \times 3$ lighting directions, and above we use the Moore-Penrose pseudoinverse. Then for estimates of α and $\boldsymbol{n}$ we in fact find lower-error estimates in Fig. 4, shown as dot-dashed lines. This is because the extra polynomial terms in PTM will tend to over-fit the correct underlying noise-free values, whereas (9) assumes a Lambertian model, correctly in this synthetic case.

In comparison, standard PST, based on straightforward least squares including all pixel values, has poorer estimates – dashed lines in Fig. 4 – because the effect of noise is swamped by the main problem, inclusion of shadow and specular values.

For real images below we therefore apply the robust PST (9), with matte weights generated by PTM, as our estimator of albedo and normal. Fig. 1(h) displays an intrinsic image, (6), using such values.

5 Specularities and Shadows

5.1 Modelling Specular and Shadow Pixels

The model (1) will not account completely for the luminance L, but only a basic matte reflectance. Luminance will include highlight and shadow contributions that drive L higher respectively lower than the matte luminance, which is given by $\hat{L} = \boldsymbol{p}\,(\boldsymbol{a}\,)\boldsymbol{c}$. Since we have in hand labels for specular w^+ and shadow w^- pixels (over lights $i = 1..n$ at each x, y location), we can *model* these extra contributions, with a view to being able to interpolate them later, for new, unmeasured, lighting conditions.

Therefore at each pixel we first consider the extra value, on top of matte luminance $\hat{L}$, due to specularity. Let us call this highlight-driven value the "sheen", ς , defined as

$$\varsigma\,(w^+) \;=\; L(w^+) - \hat{L}(w^+), \quad \varsigma\,(\neg w^+) = 0\,, \quad \text{where } (\neg w^+) = \{w^+ = 0\} \tag{10}$$

Then we can model the dependence of specularity on lighting direction using a set of Radial Basis Function (RBF) coefficients. I.e., supposing that the sheen is given in terms of 3-vector light direction $\boldsymbol{a}$ as (cf. [9])

$$\varsigma\,(\boldsymbol{a}\,) \;=\; \alpha + \boldsymbol{\beta}^{\,T}\boldsymbol{a} + \sum_{i=1}^{n} \gamma_i \phi(\|\boldsymbol{a} - \boldsymbol{a}_{\,i}\|), \quad \sum_{i=1}^{n} \gamma_i = 0, \quad \boldsymbol{A}^{\,T}\boldsymbol{\gamma} = \boldsymbol{0} \tag{11}$$

with nodes $\boldsymbol{a}_i$ and Gaussian radial base functions ϕ, we solve for values {scalar α, 3-vector $\boldsymbol{\beta}$, n-vector $\boldsymbol{\gamma}$}. For the Gaussian width we take the approximate average distance between interpolation nodes [9]. Interpolation nodes are mapped exactly, but the interpolated function might model the underlying dependency well or not, depending on the amount of smoothing applied. Here we use no smoothing, but this may be re-considered in future.

Call the remaining non-matte contribution the "shade", denoted σ:

$$\sigma(\neg w^+) \;=\; \hat{L}(\neg w^+) - L(\neg w^+), \qquad \sigma(w^+) = 0 \tag{12}$$

Again, we model this contribution using RBF, separately. The reason for including all non-sheen pixels in the shade σ, and not just w^-, is that then the two RBF interpolations, plus matte, combine to exactly equal the input luminance images.

Thus we have a model that describes luminance L as

$$L(i) \;=\; \hat{L}(i) + \varsigma(i) - \sigma(i) \tag{13}$$

over the $i = 1..n$ lights, at a pixel. Again, note that the input luminance images themselves are essentially exactly regenerated, using this model.

5.2 Interpolating Specularities and Shadows

For any new light $\boldsymbol{a}$, we use the pre-calculated RBF coefficients to generate a sum of Gaussians, for sheen ς and shade σ separately, and $\hat{L} = \boldsymbol{p}(\boldsymbol{a})\boldsymbol{c}$ as the base contribution. Note that $\hat{L}$ may include some measure of shadow or specular contribution, since in fact we have used a polynomial approximation, not simply a Lambertian one. E.g., Fig. 1(g) shows that base $\hat{L}$ value, for an interpolated light, does include some specular content. Note that for locations where a majority of the pixel values are very dark, LMS regression returns slopes that are almost zero and hence the matte contribution is very dark. Least-squares based PTM might return brighter matte values for these cases.

5.3 Adding Back Colour

Since we have utilized only the luminance, it would appear that we must repeat the regressions used in order to generate full-colour, RGB output. However, we can obviate such a requirement by instead recognizing that, since we have broken out the sheen ς separately, we can in fact add back colour for the sheen separately. Thus if we have an estimate of interpolated basic luminance $\hat{L}$, and also the shade contribution σ to luminance, which we assume has the same colour, we can generate RGB colour for these contributions to luminance by simply multiplying luminance by the chromaticity 3-vector $\boldsymbol{\chi}$.

For the sheen contribution, so far we have the scalar luminance ς. We can also obtain the specular chromaticity $\boldsymbol{\chi}_{spec}$ by identifying it with the chromaticity of the maximum-luminance input value colour over all pixels — here using the neutral-interface model of dielectrics, which states that specular colour equals light colour [8]. Then we generate an x,y-dependent difference of chromaticity, $\Delta\boldsymbol{\chi}$, equal to the chromaticity for a pixel from (7) subtracted from $\boldsymbol{\chi}_{spec}$. At pixels with specular contribution, we expect the matte chromaticity to be replaced by that for a highlight, so we add the sheen ς times $\Delta\boldsymbol{\chi}$ to the interpolated colour. As in Fig. 1(c), this works very well. Further results are shown in Figs. 2 (note the shadow interpolation in Fig. 2(c)) and in Fig. 3. For the colour input images themselves, PSNR values between the input and the thusly regenerated RGB images range from 27.54 to 50.43 with median value 35.61, justifying the suitability of this approach to colour. However, note that where robust regression generates near-zero slopes, the interpolated colour is basically that of the sheen, and is therefore greyish – these artefacts are one subject of future work.

6 Conclusion

We have presented a method to interpolate both specularities and shadows, within the PTM framework, by applying robust methods to separately model highlights, matte, and remaining luminance information, and shown how to both recover chromaticity and also combine colour information with luminance for an accurate RGB rendition under new lighting.

The nonparametric regression we use for modelling the sheen and the departure from matte is the RBF framework, but this may not be the best or the most efficient approach, and in future we will consider both the smoothing of RBF as well as other methods. For RBF, the Gaussian base functions may not necessarily be the best choice, and several other common functions could be used, such as a thin-plate model.

Overall, results for interpolating specularities and shadows are seen to be quite promising. In situations where shadows are completely black, as in Fig. 3(b), the method can produce streaks resulting from hard shadow regions. Nevertheless even for cases where pixels are saturated, as in Fig. 3(d,f), the sheen model simply sees these as extra, bright information and successfully models them in interpolants.

In future, we intend to apply the method developed to artworks, with a view to determining their 3D structure and surface properties so that they can be measured before and after they are moved, e.g. by lending to other institutions. Appearance changes for famous artworks under re-lighting are also of interest.

References

[1] V. Argyriou and M. Petrou. Recursive photometric stereo when multiple shadows and highlights are present. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2008.

[2] V. Argyriou, S. Barsky, and M. Petrou. Generalisation of photometric stereo technique to q-illuminants. In *Proceedings of 19 th British Machine Vision Conference*, 2008.

[3] H.G. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics from images. In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*, pages 3–26. Academic Press, 1978.

[4] S. Barsky and M. Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 25(10):1239–1252, 2003.

[5] M. Chandraker, S. Agarwal, and D. Kriegman. Shadowcuts: Photometric stereo with shadows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[6] E.N. Coleman and R.C. Jain. Obtaining 3-dimensional shape of textured and specular surface using four-source photometry. *Comp. Graphics and Image Proc.*, 18:308–328, 1982.

[7] C. Hernández, G. Vogiatzis, and R. Cipolla. Shadows in three-source photometric stereo. In *Proceedings of the 10th European Conference on Computer Vision*, pages 290–303, 2008.

[8] H.-C. Lee, E. J. Breneman, and C. P. Schulte. Modeling light reflection for computer color vision. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 12:402–408, 1990.

[9] G.R. Liu. *Mesh Free Methods: Moving Beyond the Finite Element Method.* CRC Press, 2002.

[10] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *In Computer Graphics, SIGGRAPH 2001 Proceedings*, pages 519–528, 2001.

[11] D. Miyazaki, K. Hara, and K. Ikeuchi. Photometric stereo beyond glass: Active separation of transparent layer and five-light photometric stereo with m-estimator using laplace distribution for a virtual museum. In *International Workshop on Photometric Analysis for Computer Vision*, pages 325–329, 2007.

[12] B.-T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6): 311–317, 1975.

[13] P. J. Rousseeuw. Least median of squares regression. *J. Amer. Stat. Assoc.*, 798:871–880, 1984.

[14] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* Wiley, 1987.

[15] H. Rushmeier, G. Taubin, and A. Gueziec. Applying shape from lighting variation to bump map capturing. In *Proceedings of Eurographics Rendering Workshop 1997*, pages 35–44, 1997.

[16] F. Solomon and K. Ikeuchi. Extracting the shape and roughness of specular lobe objects using four light photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(4): 449–454, 1996. ISSN 0162-8828.

[17] J. Sun, M. Smith, L. Smith, S. Midha, and J. Bamber. Object surface recovery using a multi-light photometric stereo technique for non-lambertian surfaces subject to shadows and specularities. *Image Vision Comput.*, 25(7):1050–1057, 2007.

[18] F. Verbiest and L. VanGool. Photometric stereo with coherent outlier handling and confidence estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[19] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Trans. Graph.*, 24(3):756–764, 2005.

[20] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19:139–144, 1980.

[21] A. Yuille and D. Snow. Shape and albedo from multiple images using integrability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 158, 1997.