Robust Estimation of Surface Properties and Interpolation of Shadow/Specularity Components

Mark S. Drew[a,1,*], Yacov Hel-Or[b], Tom Malzbender[c], Nasim Hajari[a]

[a] *School of Computing Science, Simon Fraser University, Vancouver, Canada V5A 1S6*
[b] *Department of Computer Science, The Interdisciplinary Center, Herzliya, Israel*
[c] *Mobile and Immersive Experience Lab, Hewlett-Packard Laboratories, Palo Alto, CA*

Abstract

The Polynomial Texture Map framework (PTM) extends the simple model of image formation from the Lambertian variant of Photometric Stereo (PST) to more general reflectances and to more complex-shaped surfaces. It forms an alternative method for apprehending object colour, albedo, and surface normals. Here we consider solving such a model in a robust version, not to date attempted for PTM, with the upshot that both shadows and specularities are identified automatically without the need for any thresholds. Instead of the linear model used in few-source PST for Lambertian surfaces, PTM adopts a higher degree polynomial model. PTM has two aims: interpolation of images for new lighting directions, and recovery of surface properties. Here we show that a robust approach is a good deal more accurate in recovering surface properties. For new-lighting interpolation, we demonstrate that a simple radial basis function interpolation can accurately interpolate specularities as well as attached and cast shadows even with a medium-sized image set, with no need for reflectance sharing across pixels or extremely large numbers of interpolation coefficients.

Keywords: Polynomial texture mapping; photometric stereo; robust estimation, shadows; specular; radial basis functions; continuous interpolation.

1. Introduction

The Reflectance Transformation Imaging (RTI) approach [1] captures the broad range of reflectance characteristics for any surface. One implementation is PTM (Polynomial Texture Maps) [1], using a fixed standard digital camera at constant exposure and varying lighting. A strength of PTM, in comparison to a simple Lambertian Photometric Stereo (PST) [2], is that the PTM applies a higher order polynomial regression from lighting direction parameters to model image values; thus it can better model real radiance, and to some degree apprehend intricate dependencies due to self-shadowing and interreflections. In a typical setup, a camera is mounted at the apex of a hemisphere or other collection of lights (see Fig. 2(a)), and each light is then fired in turn, thus generating a sequence of images. Usually, some 40 to 80 images are captured, with the larger the number of images the better.

While the number of source images seems large compared to traditional PST [2], this amount of data is in fact necessary to capture the intricacies in, e.g., highlights, self shadowing, and cast shadows. It is for this reason that many museum professionals are interested in this technique, and the original PTM method is

[*] Corresponding author.

Email addresses: mark@cs.sfu.ca (Mark S. Drew), toky@idc.ac.il (Yacov Hel-Or), tom.malzbender@hp.com (Tom Malzbender), nha16@cs.sfu.ca (Nasim Hajari)

URL: http://www.cs.sfu.ca/∼mark (Mark S. Drew)

[1] Mailing address: School of Computing Science, Simon Fraser University, 8888 University Drive, Vancouver, B.C. CANADA V5A1S6, Tel: 1-778-782-4277, Fax: 1-778-782-3045

already in use at major museums in the U.S. including the Smithsonian, the Museum of Modern Art, and the Fine Arts Museums of San Francisco, and is planned for the Metropolitan and the Louvre [3]. There is also a good deal of interest in applying the method in-situ in situations when artwork cannot be moved, such as imaging paleolithic rock art [4]. In this application, the highlight on a reflecting sphere can provide direction information. Of course, one could instead apply matrix factorization strategy to *recover* lighting directions as well as surface normals simultaneously (e.g. [5]) but, typically, due to non Lambertian phenomena matrix factorization methods can produce inaccurate lighting directions.

With the objective being *re-lightable* images, PTM models the dependency of a pixel's luminance (or RGB in a different embodiment) on lighting directions using a 6-term polynomial model while fitting its coefficients using least squares (LS) minimization. The polynomial variables are the light-direction parameters and thus, re-rendering can take place by relighting images using a new lighting direction. Using the PTM, surface normals can be extracted and used for other effects, in non-photorealistic rendering, such as generating artificial specular highlights, increasing directional sensitivity to lighting direction in order to enhance contrast, or extrapolating directions of light source [1]. However the basic method rests on a matte model and linear regression, and thus fails to model higher-degree phenomena such interreflections, specularities and shadows. Increasing the degree of the PTM polynomial model may help to model these effects but at the expense of deteriorating the interpolated results at non-sampled light directions due to overfitting phenomena.

In this paper, we set out a robust version of PTM to generate increased accuracy of recovery of surface normals, albedo, and colour. As a consequence we also automatically identify specularities and shadow pixels and, as a result, we can interpolate these. We show that a simple RBF interpolation can deliver excellent specularity and shadow interpolants for such medium-sized image input sets, without the need for sharing spatial information across pixels. We also provide a hierarchy of methods, from simple to complex, trading off accuracy of recovery of surface properties against speed.

To the best of our knowledge our approach to modelling non-matte components has not been applied to PTM before, and the results are striking. E.g., considering an interpolation of an object partially covered in gold leaf for a novel lighting direction (Fig. 1). [2] In the top row, for in-sample lighting directions with images shown in (a,c), the reconstructed output using our proposed method exactly matches (to machine accuracy) the measured input images, and the interpolant is sensible. In contrast, in the bottom row, using the original LS method we note that clearly a matte model cannot describe the reflectance properties of such a surface. Not only is this the case for interpolated light directions, but even in reproducing the original source imagery.

For finding the matte part of the photometric model, we make use of the Least Median of Squares (LMS) method [6], although other robust approaches could likely be used as well with comparable results; e.g. one could use robust M-estimation or an iterated re-weighted LS. Robust methods have not to date been applied to PTM. Using LMS robust regression for the matte part is advantageous because LMS provides an automatic identification of outliers, with no programming intervention required. Outliers are excluded from the input data used to estimate the matte component and, moreover, we can also classify outliers as belonging to one of two types — either specular highlights, or shadows — by asking whether the observed luminance at a pixel in a particular source image is greater respectively smaller than the predicted matte luminance.

Knowledge of inlier pixel values means that recovered surface albedo, chromaticity, and surface normals are robust, in the sense of ignoring outlier contributions and thus more accurately mapping surface

[2] Elba image data courtesy of Dr. Roberto Scopigno, Visual Computing Laboratory, ISTI-CNR, Pisa, Italy: object is "Allegory of the meeting between Corsica and Elba", gilded wood (100 × 70 cm), 1904, Magazzini della Collezione Foresi, in Portoferraio, Isola d'Elba, Tuscany.

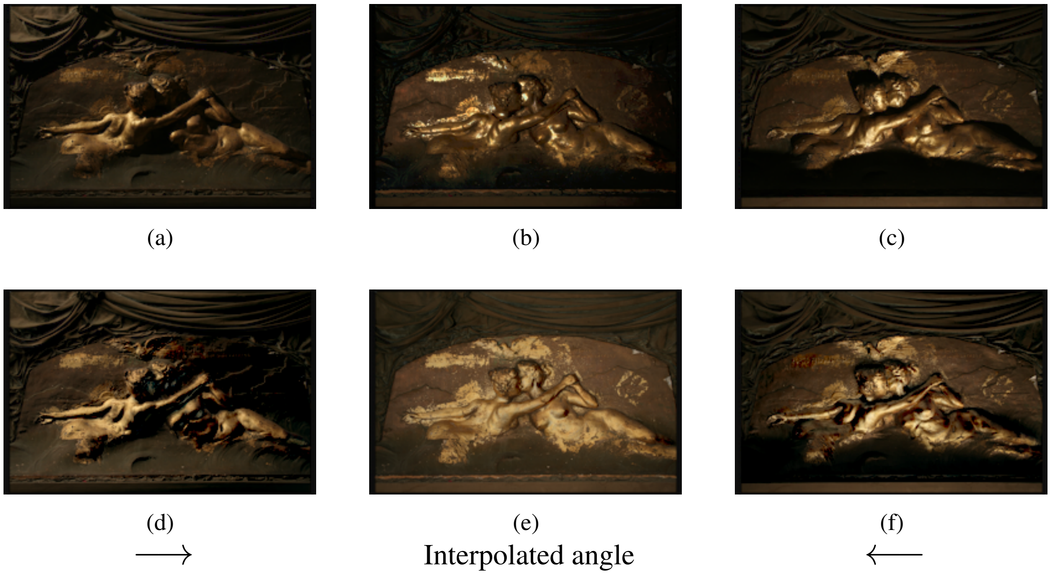


Figure 1: Image “Elba”: Reconstruction of input images (Left, Right) and interpolation (Center) for new, non-sampled light. (a,b,c): Proposed method — input images are (to machine accuracy) identical to the reconstructed image (a,c) for in-sample images, and the out-of-sample interpolation (b) is reasonable; (d,e,f): Original matte-based PTM method — while image representations appear useful and interesting, they are not close enough to actual physical appearance of object.

reflectance, colour, and shape. We also change the matte model that was used in the original PTM method. Because we go over to a robust regression from a least-squares one, here we can alter the polynomial used in PTM so as to automatically generate a subset of three regression coefficients that are *exactly* correct in the case when inliers are indeed Lambertian. In that case, surface normal and albedo directly fall out of the matte regression. In the ordinary case of surfaces that are not in fact Lambertian, this subset of matte regression coefficients will not equal the scaled normal; instead, though, knowledge of inlier labelling allows us to apply ordinary PST just on inlier pixels, yielding what we show to be superior estimates of surface normal and albedo. Finally, by generating a tripartite set of labels — inlier, outlier-shadow, and outlier-specular — we can independently identify and model specularity and shadow. Thus for a new lighting direction, we can generate pixel values interpolating robustly-identified values of both, specular as well as shadow components, using RBF interpolation. We found that for the matte regression it was sufficient to carry out robust regression on the luminance values, not on R,G,B separately, in order to find outlier pixel values. We then go on to use RBF to generate the full-colour specularity, denoted ζ , in an interpolated image, either separately from or together with the remaining full-colour non-matte contribution, denoted σ , to produce a full-colour interpolant image.

Aside from re-creating the *sampled* images in complete agreement with the originals, as opposed to in the original PTM, we also demonstrate using a leave-one-out analysis that the proposed method greatly outperforms the original PTM in generating images for non-measured, out-of-sample lighting directions. The price we pay for being able to model specularities and shadows is that we need to carry regression coefficients for each pixel numbering in fact more data than in the original image set. The benefit, however, is in being able to correctly physically model non-smooth visual information.

In §2, we discuss related work, and in §3 we outline the PTM model as well as the PTM as modified here with 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 shadows is set out in §5 in an RBF framework. Section 6 shows how colour may then be incorporated. Section 7 displays and discusses a set of results, and §8 gives concluding remarks.

2. Related Work

PTM is formulated as a generalization of the simplest variant of PST [2], wherein no calibration object is used but instead a basic Lambertian model is extended to a polynomial model. PTM moves a linear regression involving lighting directions into a nonlinear, polynomial model whose variables are two of the components of the normalized lighting direction. 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 specularities and shadows, then a robust version of nonlinear regression will still pick up the correct, linear, Lambertian dependence in an accurate manner Then this paper goes on to model specular and non-specular outlier contributions.

Early efforts at specularity detection in PST [7, 8, 9] used four lights and excluded values yielding significantly differing albedos. Rushmeier et al. [10] used a similar, 5-light approach. They proposed a PST system where the highest and lowest values in five components are discarded to avoid highlights and shadows, using the three middle intensities to reconstruct surface normals.

Miyazaki et al. [11] proposed a five light-source PST which is suitable for images taken from a virtual museum, where objects are kept behind glass. Their aim is to detect specularities and exclude them from normal and albedo reconstruction. The main idea behind this algorithm is as in [12] with the addition of a smoothness constraint on the albedo. Their method is robust to outliers under the condition that only a small number of images is supplied.

In another 4-source method [13], ambient illumination is explicitly included, as is surface integrability, and 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 if half the number of values per pixel, minus one, are outlier values, due to the high breakdown point of the LMS robust regression used [6].

In [14] it was shown that the minimum number of lights for PST for the complete visible surface of any *convex* object is six. Unlike [10], they argued that simply discarding highest or lowest intensity pixels may lead to information loss so they discard only pixels with doubtful intensities. As extra lights are better, [15] uses a spherical device to capture 156 images. Since the data is then highly over-determined, they simply discard the lowest 50% and the highest 10% to aid PST. Here, we avoid thresholds or arbitrarily discarding pixels by relying on LMS outlier detection, which automatically generates outliers [16]. In [11], an iterative scheme is devised for finding albedo and normal, which essentially uses the median of each plus smoothness. In this paper, inliers are used with no need for iteration to immediately generate both albedo and normal.

Julia et al. [17] defined two thresholds to simply discard shadow and highlight pixels. The lower threshold, which corresponds to shadows, depends on the intensity values in each set of images, while they set the upper threshold, related to highlights, to the maximum grey value, so the second threshold actually applies only to the quite special circumstance of pixel saturation.

Recently, probabilistic models have been proposed to deal with outliers. The method proposed by Chandarker et al. [18] uses a fast graph-cut method for performing Lambertian PST in the presence of shadows. They pose 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 [19], 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, [20] 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 required or used.

In a generalization of [9], Refs. [21] and [22] extend [9] into a multiple light source technique. They 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.

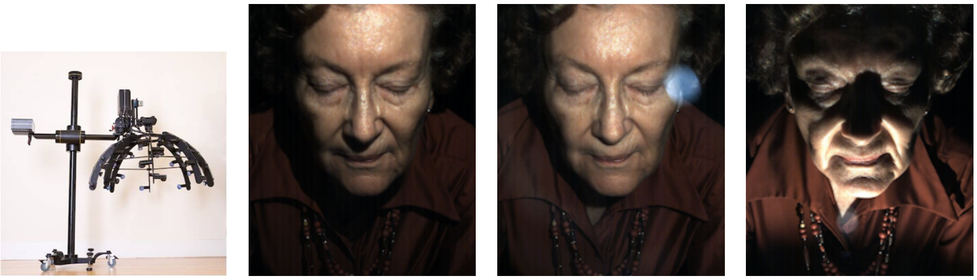
These works are concerned with dealing with specularities and shadows so as to better model surface properties: surface normal, albedo, and colour. As well, many approaches to a purely appearance-based modelling have been developed for specular interpolation. In Reflectance Sharing [23, 24] a known surface geometry is assumed, and it is argued that in order to capture high frequency information such as specular highlights it is necessary to share across neighbouring pixels. These authors use RBF for interpolation of BRDFs (Bidirectional reflectance distribution functions), over some thousands of centers for modelling spatially-varying BRDFs, using a subset of all pixels as the sample points. In contrast, here we focus on recovery of a robust estimate of surface normals (along with colour and albedo), and demonstrate that one can use a number of RBF centers equal to the number of input images, separately for each pixel, for image input sets of modest size (some 40 or 50 images). Here we model departures from a basic matte model using RBFs, but instead of interpolating spatially we interpolate over luminance values at a pixel. The argument in [23, 24] is that image sets may have no evidence of specular highlights, at a particular pixel, and in fact these authors aim at reducing the input image acquisition down to a sparse set. Here, we take a middle ground and use a medium-sized set. As well, we explicitly assume that some evidence of specular content exists at *many* pixels, even if high frequencies are not observed. The reasoning behind this assumption is that most materials possess some "sheen" quality, i.e., distributed specular content. Once RBF coefficients are obtained, rendering with lighting from any direction is simple. We demonstrate using real datasets that indeed convincing appearance modelling can be accomplished, including highlights and shadows, without neglecting recovery of surface properties including normals.

For recovery of surface normals, some works have made use of polarized light ([25] and references therein). In [26], the authors view pixel values under lighting with varying directions as measurements of the bidirectional reflectance distribution function (BRDF), for a 2D slice of the BRDF for fixed view. They compute the local tangent and binormal directions and thus normal by using symmetry of the BRDF in an orthographic projection of a hemispherical function the data parameterized with respect to the half-angle vector.

Appearance-based rendering approaches (e.g., [27, 28, 29]) are indeed concerned with appearance and BRDF modelling, and not surface recovery. Here we strike a balance by proposing a hierarchy of methods that progressively trade speed for accuracy of recovered surface properties. Asking for appearance modelling only is the simplest and fastest algorithm, while the best, robust recovery of surface normals is the slowest.

PTM has been used in numerous applications. Padfield et al. [30] employed PTM to monitor surface changes of paintings. They examined several paintings and showed that PTM can record surface features such as craquelure, planar distortion, wood grain, canvas wave, etc. They applied PTM to paintings before and after physical changes, enabling them to monitor surface changes. Hammer et al. [31] used PTM in paleontology to visualize fossils. Since many fossils cannot be fully imaged using conventional photographic approaches, PTM was used to provide a better view of the data. Other applications include criminal forensics, cultural heritage preservation, painting brushstroke visibility, and art conservation [32].

In the initial version of this work [33] we compared the method proposed to the reference point of the original work [1]. Here we retain this comparison but also go on to compare against a more shadow-error resistant method [15]. We show that this method performs relatively well, but not as well as our new method. To test the noise resistance of all the methods, in [33] we made use of synthetic data as ground truth to quantify accuracy of recovered surface properties; here we add a more stringent test, using measured range-finder data to generate synthesized, but inherently noisy, ground truth. Whereas in [33] we indicated in general how the algorithm proceeds, here we explicitly introduce and specify each step, justifying the approach for each algorithmic stage. We also investigate the tradeoff involved in separately modelling



(a) (b) u=0.22,v=0.35 (c) u=0.18,v=0.08 (d) u=-0.0,v=-1.0

Figure 2: (a): Hemispherical dome with multiple, identical lights [Image due to Phillip Klausmeyer of the Worcester Art Museum].; (b,c,d): Three of inputs (from set of 50), with lighting direction denoted (u, v, w).

specular and shadow interpolation, versus more simply modelling the entire departure from an underlying (polynomial) matte model. Considering accuracy of recovered surface properties and also of appearance of interpolated images, we set out a hierarchy, from complex to simpler, of methods that can be used depending upon the level of accuracy required for obtaining surface normals and colour. We replaced the method in [33] for obtaining the colour of the specularities with a much more reliable method based on a separation of the modelling procedure for each colour channel. This eliminates colour artifacts seen previously in the output. In experiments, we also devise an exhaustive test consisting of generating interpolated views for out-of-sample light-direction angles consisting of a large set of averages over random pairs of in-sample values measured, for synthetic data. This shows the benefit of going over to a robust method. As well, for real measured in-sample images here we carry out a leave-one-out analysis whereby an interpolation basis is generated in turn leaving out one input image and then regenerating the missing image, over a large image set. Finally, we test the algorithm for a set of new data that is very challenging to the method, consisting largely of gold, or gold-leaf applied to art objects. Both the complexity of these surface shapes, as well as the metallic surface reflectance, make these images sets very demanding on the method, with the original method failing on these sets. Instead, here we obtain excellent results.

3. Image Formation and Polynomial Modelling

3.1. Polynomial Texture Mapping

In the following we show how we extend (an altered version of) PTM [1] to explicitly deal with highlights and shadows. PTM assumes that least squares fitting will effectively be adequate in a polynomial regression, for a smooth dependence of images on lighting direction. Along with adopting a polynomial for capturing aspects of matte reflectance, at each pixel location, we also explicitly model change with light direction for specularities and shadows. 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.

3.2. PTM Model

Suppose we have acquired n images of a scene, taken with a fixed-position camera but with lighting from $i = 1..n$ different lighting directions $\boldsymbol{a}^i = (u^i, \, , v^i \, w^i)^T$. Fig. 2(a) shows what a typical hemispherical lighting frame looks like (but note that there is no requirement that a hemisphere is used for lighting positions), with Figs. 2(b,c,d) showing a subset of measured images.

Let each RGB image acquired be denoted $\boldsymbol{\rho}^i$. Suppose we make use of the luminance images instead, $L^i = \sum_{k=1}^{3} \rho_k^i$. This reduction in dimensionality reduces the computational burden of robust regression and we can re-insert colour later.

A matte PTM model consists of a polynomial regression from lighting to luminance via a vector of polynomial terms $\boldsymbol{p}$, with $\boldsymbol{p} = \boldsymbol{p}(\boldsymbol{a})$ a low-dimensional vector function of vector lighting direction $\boldsymbol{a}$. PTM uses a regression that is pixel-wise, so if we have n lights, then we are interested in regressing from a set of n vectors $\boldsymbol{p}(\boldsymbol{a}^i)$ onto the set of n luminance values L^i observed at that pixel over the n images taken. For example, suppose $n = 50$ (i.e., there are 50 images, taken under 50 different lights) and suppose $\boldsymbol{p}(\boldsymbol{a}^i)$, $i = 1..50$, is a 6-vector function of the three components of normalized lighting direction $\boldsymbol{a}^i$. In the original PTM [1], the polynomial function used was

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

So with a separate regression at each pixel (x, y), we seek a 6-vector of regression coefficients $\boldsymbol{c}(x, y)$ via least squares as follows:

$$\begin{bmatrix} \boldsymbol{p}(\boldsymbol{a}^1) \\ \boldsymbol{p}(\boldsymbol{a}^2) \\ \ldots \\ \boldsymbol{p}(\boldsymbol{a}^n) \end{bmatrix} \boldsymbol{c}(x,y) \;=\; \begin{bmatrix} L^1(x,y) \\ L^2(x,y) \\ \ldots \\ L^n(x,y) \end{bmatrix} \qquad \text{, or} \quad \underset{50\times 6}{\boldsymbol{P}} \quad \underset{6\times 1}{\boldsymbol{c}(x,y)} \;=\; \underset{50\times 1}{\boldsymbol{L}(x,y)} \tag{2}$$

Each pixel has its own $\boldsymbol{c}(x, y)$, and the $\boldsymbol{L}(x, y)$ vector is the collection of all observed luminances at that pixel over the n images. All lights are assumed to have the same intensity. [3]

In the original PTM [1], $\boldsymbol{p}(\boldsymbol{a})$ is the 6-component set of polynomial terms in (u, v) [Eq. (1)], and so coefficient vector $\boldsymbol{c}(x, y)$ is a 6-vector.

Here, we start off by firstly replacing the polynomial $\boldsymbol{p}$ in (1) 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 an underlying Lambertian surface plus specularities and shadows; 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. One is then not trying to model a linear function by using a nonlinear model, and the regression will simply place zeros in $\boldsymbol{c}$ for the higher order terms. Nevertheless, it is useful to keep a polynomial description, to suit surfaces with matte component which is not Lambertian, and also because even for a purely Lambertian surface more than three components are useful in modelling the discarding of negative shading components (cf. [35]). 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, we found that 7-D, with a v^2 term in (3) that symmetrizes the use of u and v, performed the same in terms of accuracy.)

Within the modified PTM model then (as in eq. (3)) at each pixel located at (x, y) we can utilize the values of coefficients $\boldsymbol{c}(x, y)$ derived from measured images to generate a new matte image illuminated from a novel lighting direction $\boldsymbol{a}'$ as $L(x, y) = \max[\boldsymbol{p}(\boldsymbol{a}')\, \boldsymbol{c}(x, y), 0]$, where negative values are nullified as they represent self-shadowing values.

[3] Multi-view acquisition hardware, and the use of spherical harmonics (SH) instead of polynomials, are discussed in [34]. However note that SH are not appropriate for modeling specular and shadow phenomena since their support is too wide and one may need a great many basis functions to approximate local and sharp changes. Wavelets could be used, but then there are many options for basis functions and here we model non-matte contributions using RBF.

3.3. Outlier Identification and Mapping to Highlight and Shadow

To make the regression process in eq. (2) robust to outliers we utilize the LMS [16], which generates outliers "on a silver platter", without any intervention. So here, regression is

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

LMS proceeds by randomly selecting test sets of six values of L from $1..n$ and inverting eq. (2) for non-singular sets; the number of sets to try is guaranteed to be much lower than ${}_nC_6$ while still retaining a high breakdown point [6]. Outliers are first identified as values of L outside a band around a predicted linear regression on the independent variables (the components of $\boldsymbol{p}$). The band's width is chosen to be proportional to the least median of squared residuals over all tests, and then a Reweighted Least Squares regression is carried out using only the accepted points. (Note the lens flare artefact in Fig. 2(c): this type of non-specular and non-shadow artefact will be handled as well by the robust regression.)

Fig. 3 shows the effect of robust regression for a single pixel (from the noisy-sphere image in Fig. 5 below) over 50 lights, on synthetic data for which we know ground truth. Here we wish to recover the matte contribution to luminance by using regression: correct matte values are shown as black crosses whereas the total contribution of matte, specularity, and shadow is shown as black dots (the data is sorted by luminance value). While a least squares regression (blue circles) is led astray by the specularity and shadow content, the robust regression (red squares) almost always finds close to the correct value.

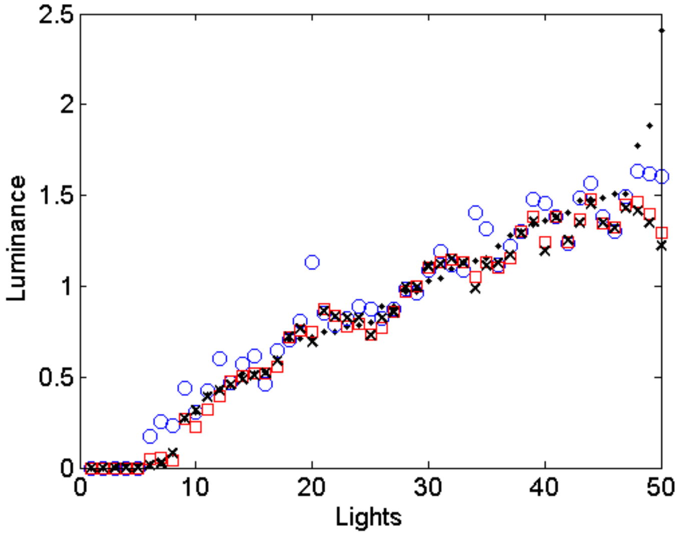


Figure 3: Robust regression versus non-robust regression for matte plus specular plus shadow data in noisy sphere of Fig. 5. (The pixel used here is mid-way on the vertical axis and 1/4 of the way across the horizontal axis.) Black dots are Matte+Shadow+Specular luminance; blue circles are non-robust matte LS regression; black crosses are correct matte luminance; and red squares are robust LMS fit. The RMS error to the correct matte values for non-robust fit is 15.3% whereas for robust fit is 3.4%.

The output of LMS is the set of regression coefficients $\boldsymbol{c}$, plus a set of n weights (labels) $\boldsymbol{\omega}$ identifying inliers ($\omega \;=\; 1$) and outliers ($\omega \;=\; 0$), thereby excluding some lights at this pixel. But we know here that outliers are generated because (a) L values are too high to suit the model eq. (2) — we take these to be specular contributions; or (b) L values are 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 binary weights $\{\omega^{\,0}, \omega^{\,+}, \omega^{\,-}\}$ at each pixel, with $\omega^{\,0}$ set for lights generating inlier values, $\omega^{\,+}$ for specularities, and $\omega^{\,-}$

for shadows. Fig. 4(b) shows weights corresponding to input image 4(a), $\omega^{\,0}$ as white, $\omega^{\,+}$ as green, and $\omega^{\,-}$ as red.

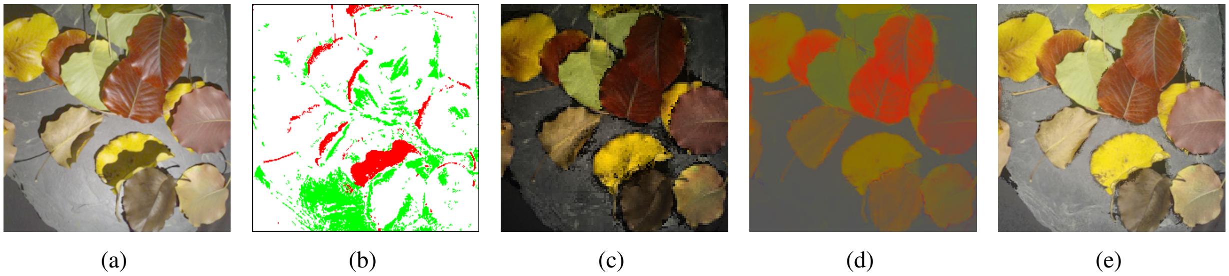


Figure 4: (a): One input image, light direction $(u = 0.322, v = -0.378)$. (b): Weights: green are outliers with luminance having positive residual, red are outliers with luminance having negative residual. (c): Matte output from robust regression for this lighting direction (once colour is added back). (d): Chromaticity, eq. (5). (e): Instrinsic image, as defined in eq. (6).

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. 4(c). We are assured of doing better than the standard PTM since we have excluded distracting specularities and shadows from consideration whilst generating coefficients $\boldsymbol{c}$.

4. Colour, Albedo, and Normal

4.1. Chromaticity

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 L will be composed of a scalar albedo α times lighting strength I times shading factor s (e.g., the shading factor for a Lambertian surface would be surface normal dotted into normalized lighting direction). Since we have no way of disentangling scalar lighting intensity from scalar albedo, we shall simply lump both scalars I and α into α. Thus,

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

An *intrinsic image* [36] (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. Chromaticity and intrinsic image are shown in Fig. 4(d,e).

Since we have weights $\omega^{\,0}$ identifying pixels indeed corresponding to the polynomial matte 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 $\chi^i_k = \rho^i_k / L^i$, $k = 1..3$, 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, for each colour channel $k = 1..3$, over $\omega^{\,0} = 1$ inlier pixels:

$$\boldsymbol{\chi} \;=\; \underset{i \in \omega^{\,0}}{\text{median}} \left(\boldsymbol{\rho}^i / L^i\right) \tag{7}$$

That is, we form chromaticity (colour without magnitude) by dividing input image RGB values at the current pixel by input luminance, and obtain a best estimate as the median of these vectors over inlier values.

Fig. 4(d) shows how this recovered chromaticity appears: indeed it is colour without magnitude and, to large measure, without shading as well. And Fig. 4(e) shows the Lambertian-based matte intrinsic image $\boldsymbol{\rho}_{intrinsic} = \alpha\boldsymbol{\chi}$ from eq. (6) with albedo α as generated below. Since image (c), the matte output $\boldsymbol{\rho} = L\boldsymbol{\chi}$ with $L = \boldsymbol{p}(\boldsymbol{a})\boldsymbol{c}$, includes shading its appearance is more realistic, of course. But note in image (c) that using robust slopes can generate black output since dark values may form a majority at some pixels, giving slopes that are near zero.

4.2. Surface Normal and Albedo from Matte Regression

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

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

Let us denote this surface-normal algorithm as Method:COEFF. We compare the accuracy of this estimate of albedo and surface normal with LS, with LMS, and with a third alternative method, described below in the next section.

4.3. Comparison to a Quantile-Based LS

We can evaluate how well the robust regression of matte quantities does, compared to the original LS approach, in a simple noise sensitivity test. However, to also include a comparison with a better method than a simple LS, consider a less-simple but still very fast method suggested in [15]. As explained in §2, at each pixel Ref. [15] simply drops the bottom 50% and top 10% of luminance values. However, rather than performing a straightforward LS on the set of values that are left, these authors use a "tent" (triangular) function to weight lighting directions in the middle of the resulting range highest when carrying out PST. We denote this method as Method:QUANTILE, and denote the original PTM method as Method:LS — but using our improved polynomial model as given in eq. (3). The robust method based on LMS in eq. (4) we denote Method:LMS.

For both Method:QUANTILE and our proposed Method:LMS we can generate a better estimate of surface normals than by simply using PST: Since we have classified the pixel values as inliers and outliers, we can write a *robust version of PST* by using only inlier pixels (i.e. $\omega^{\,0} = 1$). Suppose we collect all lighting directions $\boldsymbol{a}^i$, $i = 1..n$ into an $n \times 3$ matrix $\boldsymbol{A}$. Now suppose that the number of inlier pixels is $m < n$, i.e., there are m pixel measurements (at the current pixel location (x, y)), amongst the n measured values, that are deemed to be inliers. The remaining $n-m$ values are regarded as outliers. Let $\boldsymbol{A}(\omega^{\,0})$ be the set of rows of matrix $\boldsymbol{A}$ corresponding to these "good" lighting directions (at this pixel): submatrix $\boldsymbol{A}(\omega^{\,0})$ still has 3 columns but only m rows. Likewise, let $\boldsymbol{L}(\omega^{\,0})$ be the subset of luminance values corresponding to inliers, so that $\boldsymbol{L}(\omega^{\,0})$ is an m-vector.

Then we can generate a robust estimate of normalized surface normal $\widehat{\boldsymbol{n}}$ and albedo α via

$$\tilde{\boldsymbol{n}} \;=\; \Big(\boldsymbol{A}(\omega^{\,0})\Big)^{+} \boldsymbol{L}(\omega^{\,0}); \quad \alpha = \|\tilde{\boldsymbol{n}}\|, \quad \widehat{\boldsymbol{n}} = \tilde{\boldsymbol{n}}/\alpha \tag{9}$$

where $\boldsymbol{A}^{+}$ denotes the Moore-Penrose pseudoinverse. For Method:QUANTILE we instead use matrices weighted by the triangular functions.

Comparing the four methods, Figs. 5(a,b) shows results over increasing percent Gaussian noise, for a synthetic sphere RGB image, with pixel values generated for 50 non-uniform lighting directions for a Lambertian surface plus Phong illumination [37] with roughness 1/20, plus noise. Here the lines show the results for the median error over the sphere for normal vector direction, in Fig. 5(a), and for albedo, in Fig. 5(b). Results for our proposed method are shown as the solid line in blue, and for the LS-based PTM method as the dotted line in red. For an intermediate approach, the dashed magenta line shows median results for Method:QUANTILE; and we also show the result for Method:COEFF from §4.2 in dot-dashed green. As an further example, we shaded the depth map for a Mozart bust, again for 50 lighting directions, [4] showing both attached and raycaster cast shadows as well as specularities: error results are similar.

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 the surface normal dotted into light direction, and specularities and shadows do not distract the robust regression. Therefore both Method:COEFF and Method:LMS start at zero error. As noise increases, Method:LMS still does very well. Method:COEFF is much worse so will be dropped. In comparison, standard PST, based on straightforward least squares including all pixel values, has the poorest estimates because the effect of noise is swamped by the main problem, inclusion of shadow and specular values. The intermediate approach, Method:QUANTILE – dashed magenta lines – does reasonably well, but not as well as our Method:LMS. Obtaining better estimates of surface normals has practical implications, in that as well as PTMs providing interpolation of appearance to novel lighting, the techniques of reflectance transformation [1] have lead to the most important breakthroughs. These methods transform the reflectance properties to new material types, by either introducing specular components (specular enhancement) or increasing the second derivative of the reflectance function (diffuse gain). Both of these techniques require estimates of the surface normal.

For real images below we therefore apply robust PST, eq. (9), with matte weights generated by LMS, as our estimator of albedo and normal.

5. Specularities and Shadows

5.1. Modelling Specular and Shadow Pixels

The model (2) for the luminance L for a basic matte reflectance will not account completely for the luminance, since L will include highlight and shadow contributions that drive L higher respectively lower than the matte luminance. Since we have in hand labels for specular ω^+ and shadow ω^- 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.

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

$$\zeta\,(i) \;=\; \begin{cases} L(i) \;-\; \hat{L}(i) & \text{if } \omega^+(i) = 1, \\ 0 & \text{if } \omega^+(i) = 0 \end{cases} \tag{10}$$

for each pixel location. That is, at a pixel $\boldsymbol{x}$ we define $\boldsymbol{\zeta}$ as non-zero only for specular-outlier values.

Then we can model the dependence of specularity on lighting direction using a set of RBF coefficients.

We call the remaining non-matte contribution the "shade", denoted $\boldsymbol{\sigma}$:

$$\sigma\,(i) \;=\; \begin{cases} \hat{L}(i) \;-\; L(i) & \text{if } \omega^+(i) = 0, \\ 0 & \text{if } \omega^+(i) = 1 \end{cases} \tag{11}$$

Again, we model this contribution using RBF, separately. The reason for including all non-sheen pixels in the shade $\boldsymbol{\sigma}$, and not just ω^-, is that then the two RBF interpolations, plus matte, combine to exactly equal the input luminance images and for in-sample lighting directions we always have input images reproduced, as in Fig. 1 above.

[4] The laser range data for the bust of Mozart is due to Fridtjof Stein of the USC Institute for Robotics and Intelligent Systems.

Thus, we have a model that describes luminance L at $\boldsymbol{x} = (x, y)$ as

$$L^i(\boldsymbol{x}) \;=\; \hat{L}^i(\boldsymbol{x}) + \zeta^{\,i}(\boldsymbol{x}) - \sigma^{\,i}(\boldsymbol{x}) \tag{12}$$

over the $i = 1..n$ lights, at a pixel. Again, note that the input luminance images themselves are exactly regenerated, using this model. From the n images imaged under n different lighting directions, we now interpolate the specular and shadow components between those lights, generating values for novel, unmeasured light angles.

5.2. RBF modelling

Let us consider the sheen interpolation first. We may wish to model specularity separately from shadows, or indeed we can instead combine both sheen and shade into a single quantity. Here we carry out a per-pixel interpolation. We might first consider using a linear interpolation scheme but, while appealing, such a simple approach is bound to fail: e.g. for cast shadows if one image has a cast-shadow to the left, and a second image has one to the right, then their linear interpolant will have both, rather than a shadow in an interpolated direction. So instead we use per-pixel RBF modelling.

For our set of n images, each having N pixels, let the sheen at one pixel for one light be $\zeta(\boldsymbol{a}^i)$. We wish to model the sheen in terms of a set of radial basis functions ϕ storing functions of distances from a particular lighting-direction $\boldsymbol{a}$ to the n known lighting-direction vectors. That is, we adopt an interpolation model

$$\zeta(\boldsymbol{a}) \;=\; \sum_{i=1}^{n} \psi_i \, \phi(\|\boldsymbol{a} - \boldsymbol{a}^i\|)$$

with RBF coefficients ψ_i, and functions ϕ that could be Gaussians, multiquadrics, logarithmic RBFs, thin-plate splines, etc. [5]

In some cases a first-degree polynomial in $\boldsymbol{a}$ is added to account for linear and constant portions of an interpolation and ensure positive definiteness of the solution [39]. Therefore, our model is

$$\zeta(\boldsymbol{a}) \;=\; \alpha + \beta_1 u + \beta_2 v + \beta_3 w + \sum_{i=1}^{n} \psi_i \, \phi(\|\boldsymbol{a} - \boldsymbol{a}^i\|) \tag{13}$$

where a lighting direction is $\boldsymbol{a} = [u, v, w]$.

At a pixel, we have n sheen values corresponding to n lights $\boldsymbol{a}^i$ and additional 4 coefficients, $\alpha, \beta_1, \beta_2 \beta_3$, so a system of n equations (13) is under-determined. Therefore one scalar and one vector constraint are usually introduced into the system [38]:

$$\sum_{i=1}^{n} \psi_i = 0, \quad \sum_{i=1}^{n} \psi_i a_k^i = 0, \quad k = 1, 2, 3 \tag{14}$$

where we denote $[u^i \; v^i \; w^i] = [a_1^i \; a_2^i \; a_3^i]$.

[5] Here we make use of the special case of Gaussian RBF basis functions, and use distance functions $\phi(r) = \exp(-0.5\, r^2/\kappa^2)$ with constant κ. If the entire set of n lighting-vectors $\boldsymbol{a}^i$ is the $n \times 3$ matrix $\boldsymbol{A}$, the dispersion (κ for Gaussians) can be taken to be the approximate average distance between interpolation nodes $\boldsymbol{a}_k$ consisting of the columns of $\boldsymbol{A}$ [38]:

$$\kappa = \left\{ \frac{1}{n} \Pi_{k=1}^{3} \left[\max_{i=1..n} (a_k^i) - \min_{i=1..n} (a_k^i) \right] \right\}^{1/3}$$

I.e., κ is $(1/n)^{1/3}$ times the geometric mean of the bounding-box widths enclosing the lights.

So extending the RBF model by including the 4-component "polynomial" term in eq.(13) we have now a $(n+4)$-dimensional basis of RBF functions

$$\boldsymbol{\phi}' \;=\; \left[\left\{\phi(\|\boldsymbol{a}-\boldsymbol{a}^i\|),\; i=1..n\right\},\; 1,\; u,v,w\right]^T \tag{15}$$

Accordingly, the set of n RBF coefficients is now extended into $n+4$ coefficients $\boldsymbol{\psi}'$. Consider the collection of n sheen values ζ^i at the current pixel for the $i = 1..n$ lights. In order to solve for the coefficients $\boldsymbol{\psi}'$ we substitute the estimated sheen values in eqs. (13) and (14), as well as the additional constraints. We can define an $n \times n$ matrix $\boldsymbol{\Phi}$ of functions of distances between all n lights: $\Phi(i,j) = \phi(||\boldsymbol{a}^i - \boldsymbol{a}^j||)$. Additionally, we define an $n \times 4$ matrix $\boldsymbol{Q}$as follows:

$$\boldsymbol{Q} \;=\; [\underset{n\times 1}{\mathbf{1}_n},\;\; \underset{n\times 3}{\boldsymbol{A}}]$$

where $\mathbf{1}_n$ is an $(n \times 1)$-vector of 1's and where $\boldsymbol{A}$ consists of the set of 3-vectors $\boldsymbol{a}^1, \boldsymbol{a}^2, ..., \boldsymbol{a}^n$ as rows.

Then we define a larger matrix $\boldsymbol{\Phi}'$ by

$$\boldsymbol{\Phi}' \;=\; \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{Q} \\ \boldsymbol{Q}^T & \mathbf{0}_4 \end{bmatrix} \tag{16}$$

where $\mathbf{0}_4$ is a 4×4 array of zeros.

Extending the sheen n-vector $\boldsymbol{\zeta}$ into an $(n+4)$-vector by appending four zeros,

$$\boldsymbol{\zeta}' \;=\; \{[\zeta^i,\, i=1..n,],\; 0,0,0,0\}$$

we can then solve the RBF model for RBF coefficients $\boldsymbol{\psi}'$, via

$$\underset{(n+4)\times 1}{\boldsymbol{\zeta}'} \;=\; \underset{(n+4)\times(n+4)}{\boldsymbol{\Phi}'} \;\; \underset{(n+4)\times 1}{\boldsymbol{\psi}'} \tag{17}$$

Here we solve this equation for an approximation $\hat{\boldsymbol{\psi}}'$ of the RBF coefficients $\boldsymbol{\psi}'$ using the (pseudo-) inverse of $\boldsymbol{\Phi}'$. Then the approximation $\hat{\boldsymbol{\zeta}}'$ of sheen (extended by four components), over all the lights is given by back-substituting $\hat{\boldsymbol{\psi}}'$ into eq. (17) and the sheen n-vector itself is the first n components of the result; i.e.,

$$\begin{aligned} \hat{\boldsymbol{\psi}}' \;&=\; (\boldsymbol{\Phi}')^+\,\boldsymbol{\zeta}'\,, \\ \hat{\boldsymbol{\zeta}}' \;&=\; \boldsymbol{\Phi}'\;\hat{\boldsymbol{\psi}}'\,, \\ \hat{\boldsymbol{\zeta}} \;&=\; \hat{\boldsymbol{\zeta}}'_{i=1..n} \end{aligned} \tag{18}$$

Since there are $n+4$ equations and $n+4$ unknowns, interpolation nodes are mapped exactly, but the interpolated function might model the underlying dependency well or not, and this might also depend on the amount of smoothing applied (but above we use no smoothing).

As opposed to in the Reflectance Sharing approach [23, 24], here we do not combine information across pixels, but model each pixel separately. Implicit in such an approach is that we expect there to be some specular and some shadow content at any pixel. For the intermediate-sized image sets used here (as opposed to a sparse set aimed at in Reflectance Sharing) the results displayed below justify the suitability of this assumption. The method set out here is considerably simpler than many methods advanced for strictly appearance modelling (e.g., [27, 28, 29]) while at the same time also recovering surface properties such as surface normal.

Above we have dealt with an RBF expansion for the sheen values. The shade values can be treated similarly separately. Alternatively we can apply RBF modelling onto the entire departure η from matte:

$$\eta \;=\; L - \hat{L} \tag{19}$$

Also, for simplicity here we concentrate on luminance L, but below in §6 we separately model the non-matte contribution for each R,G,B colour channel.

In summary, since we expect the matte component to be comparatively smooth with respect to light direction, compared to specular highlights and shadows, we have adopted the (modified, and also made robust) original PTM for the matte component and reserve the RBF formulation for the other components. For the reconstructed luminance, then, we have the decomposition from eq. (12),

$$\hat{L}^i \;=\; p(\boldsymbol{a}^i)\boldsymbol{c} + \hat{\zeta}^i - \hat{\sigma}^i \tag{20}$$

and have perfect reconstruction of the original images, because nodes are perfectly regenerated.

So far we have considered a single pixel. Grouping all N pixels together, the above considerations generate the $N \times (n+4)$ sheen part, which we denote by $\boldsymbol{\mathcal{Z}'}$, and the corresponding shade part is denoted $\boldsymbol{\mathcal{S}'}$. For $\boldsymbol{\mathcal{Z}'}$, using the extended matrix $\boldsymbol{\Phi}'$ derived above we can solve for all RBF coefficients $\boldsymbol{\Psi}'$ at once by inverting

$$\underset{(n+4)\times N}{\boldsymbol{\mathcal{Z}'}} \quad = \quad \underset{(n+4)\times(n+4)}{\boldsymbol{\Phi}'} \quad \underset{(n+4)\times N}{\boldsymbol{\Psi}'} \tag{21}$$

where $\boldsymbol{\mathcal{Z}'}$ is sheen vectors each extended by four zeros.

This gives all interpolation coefficients $\boldsymbol{\Psi}'$ for all pixels and all lights. Now we can use this $\boldsymbol{\Psi}'$ determined by (21) to interpolate for new lighting directions using their associated distance vectors.

5.3. New Light

For interpolation, given RBF coefficients $\boldsymbol{\Psi}'$ determined above, an interpolated $(N \times 1)$ sheen image $\hat{\mathcal{Z}}$ is found by first generating an extended Gaussian distance $(n+4)$-vector $\boldsymbol{\phi}'$ for the new light $\boldsymbol{a}$ via eq. (15). Then we derive an estimate of sheen over the whole image via

$$\underset{1\times N}{\hat{\mathcal{Z}}} \quad = \quad \underset{1\times(n+4)}{\boldsymbol{\phi}'(\boldsymbol{a})} \quad \underset{(n+4)\times N}{\boldsymbol{\Psi}'} \tag{22}$$

and similarly for shade $\hat{\mathcal{S}}$. (We reserve calligraphic letters for recovered single images, as opposed to the input set of images.)

An interpolated $(N \times 1)$ luminance image is thus given by

$$\hat{\mathcal{L}} \;=\; \boldsymbol{C}\,\boldsymbol{p}(\boldsymbol{a}) + \hat{\mathcal{Z}} - \hat{\mathcal{S}} \tag{23}$$

where $\boldsymbol{C}$ is the $N \times 6$ collection of matte PTM slope coefficients over all the pixels. We can verify the correctness of eq. (23) by simply regenerating $\mathcal{L}$ for any of the input lights, and within machine accuracy the input luminance image is perfectly reconstructed.

For a new, unmeasured light the interpolation summarized in eq. (23) is expected to deliver a reasonable luminance image, finding continuous and accurate interpolations for non-sampled parameters. Below in §6 we verify the accuracy of this interpolation model by carrying out experiments on synthetic data with ground truth and also in §7 using a leave-one-out experiment to reconstruct real data. We also introduce a small amount of Tikhonov regularization into the pseudoinverse in eq. (18), to guard against instability. If we use a small Tikhonov parameter ($\lambda = mean(diag(\boldsymbol{\Phi}'))/(5 \times 10^4)$, here) then we will guard against overshooting between in-sample nodes while only reducing the mean PSNR for in-sample image reconstruction down from machine-infinity to the completely acceptable level of 45dB (effectively negligible difference). I.e., in place of $(\boldsymbol{\Phi}')^+$ in eq. (18) we instead use $[(\boldsymbol{\Phi}'^T\boldsymbol{\Phi}' + \lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}'^T]$.

5.4. Modelling Combined Non-Matte Component

Suppose that rather than using RBF modelling to apprehend each non-matte component separately, as in eq. (23), we instead model all departure from matte luminance using the total difference η. If the reconstructed η image is denoted $\hat{\mathcal{H}}$, then our model is simply

$$\hat{\mathcal{L}} \;=\; \boldsymbol{C}\, p(\boldsymbol{a}) \,+\, \hat{\mathcal{H}} \tag{24}$$

Since RBF modelling returns the nodes themselves essentially exactly, reconstructing all input luminance images will be accurate. Therefore for interpolation we can in fact use η instead of sheen and shade separately, and produce virtually identical results. Hence in the interests of reducing complexity we therefore move to using a unified departure η from luminance. Nevertheless, for some applications we may indeed wish to carry both sheen and shade separately.

In the next section, we will show how to include colour in the interpolant.

6. Interpolating Colour

6.1. Recovered Matte Image

In [33], we used a simple approach to recover *colour* information for the interpolated PTM imagery. Since we recovered chromaticity $\boldsymbol{\chi}$ robustly via eq. (7), we can produce a recovered colour matte image by multiplying the recovered luminance $\hat{\mathcal{L}}_{matte}$ by chromaticity:

$$\hat{\mathcal{L}}_{matte} \;=\; \boldsymbol{C}\; p(\boldsymbol{a}); \qquad \hat{\mathcal{R}}_{matte} \;=\; \hat{\mathcal{L}}_{matte}\, \boldsymbol{\chi} \tag{25}$$

where $\hat{\mathcal{R}}_{matte}$ is our recovered matte colour image. Fig. 6 shows how this quantity appears for one input dataset.

Carrying on with eq. (9) for robustly recovered albedo and surface normal, and eq. (6) for the intrinsic image, we recover images as displayed in Fig. 7. Note that since Figure 6(d) includes shading, its appearance is somewhat more realistic, of course, than is Fig. 7(c). From Fig. 6(d) and from Fig. 7 it is clear that the high specular content in this set of images means that settling for a matte-only reconstruction will simply not be good enough. Instead, we have to interpolate either the shade and sheen separately or the entire non-matte contribution η.

But the problem remains as to how to properly interpolate colour of highlights. In [33], we solved this problem by modelling shade and sheen in luminance only, with colour given by the chromaticity; then colour change associated with highlights was added back by identifying the chromaticity of highlights as that for the maximum value of R, G, and B over all input images. However, that solution led to some colour artefacts in the output images, which looked grey in that the highlight colour was achromatic.

Here we take a simpler and much more satisfactory approach: we simply model the non-matte component η separately in each colour channel. Fig. 7(d) shows one image from the input set, and Fig. 7(e) shows an image interpolant for a nearby new lighting direction: clearly, both specularities and shadows are properly interpolated. [6] Fig. 7(f) shows another example of interpolation, clearly showing interpolation of cast shadows as well as of specularities.

6.2. Modelling Colour

Now all RBF quantities become functions of the colour channel as well. Firstly, our approximated colour matte image is given by eq. (25) for each of the input images, giving an $N \times 3 \times n$ matte colour

[6] A video showing interpolation over new lighting directions is online at http://www.cs.sfu.ca/∼mark/ftp/Ivc2012/barb.wmv .

tensor $\boldsymbol{R}_{matte}$. Then we define an $(N \times 3 \times n)$ set of non-matte colour values $\boldsymbol{H}$ (the "excursion") for our input set of colour images, via

$$\boldsymbol{H} \;=\; \boldsymbol{R} \;-\; \boldsymbol{R}_{matte} \tag{26}$$

where $\boldsymbol{R}$ is the $(N \times 3 \times n)$ set of input images. (We use boldface upper-case letters for quantities associated with the entire set of n input images.) We extend $\boldsymbol{H}$ into an extended version $\boldsymbol{H}'$ by appending a set of $(n \times 3 \times 4)$ zeros.

Then for each colour channel we recover an approximation $\hat{\mathcal{H}}$ of the non-matte image as we did in eqs. (21) and (22) to develop the sheen $\hat{\mathcal{Z}}$. We find that using a 3-channel approach, the grey artefacts that appeared in [33] are no longer present.

The complete algorithm is thus as follows:

Phase 1: Offline

Over the input set of n images, we first consider the colour:

$$\begin{array}{lccccc}
\text{approx. luminance:} & \underset{N\times n}{\boldsymbol{L}_{matte}} & = & \underset{N\times 6}{\boldsymbol{C}} & \underset{6\times n}{(\boldsymbol{p}(\boldsymbol{A}))^T} & \\
\text{matte colour:} & \underset{N\times 3}{\boldsymbol{R}^i_{matte}} & = & \underset{N\times N}{diag\left(\boldsymbol{L}^i_{matte}\right)} & \underset{N\times 3}{\boldsymbol{\chi}}\,, & i = 1..n, \\
\text{excursion:} & \underset{N\times 3\times n}{\boldsymbol{H}} & = & \underset{N\times 3\times n}{\boldsymbol{R}} & - & \underset{N\times 3\times n}{\boldsymbol{R}_{matte}} \\
\text{excursion}' : & \underset{N\times 3\times (n+4)}{\boldsymbol{H}'} & = & \underset{N\times 3\times n}{[\boldsymbol{H}\;,} & \underset{N\times 4}{\boldsymbol{0}]} &
\end{array} \tag{27}$$

Here, $\boldsymbol{p}(\boldsymbol{A})$ once again means the set of $n \times 6$ polynomial functions of the $n \times 3$ lighting-direction collection $\boldsymbol{A}$. Matte regression coefficient matrix $\boldsymbol{C}$ is $N \times 6$ for N pixels. And chromaticity $\boldsymbol{\chi}$ is $N \times 3$.
Then we calculate and store the RBF coefficients $\boldsymbol{\Psi}'$ over all the input lights:

$$\underset{N\times 3\times (n+4)}{\boldsymbol{\Psi}'} \;=\; \underset{N\times 3\times (n+4)}{\boldsymbol{H}'} \;\; \underset{(n+4)\times (n+4)}{(\boldsymbol{\Phi}')^+} \tag{28}$$

Phase 2: Online

For a new light $\boldsymbol{a}$, firstly eq.(15) gives the extended Gaussian distance vector $\boldsymbol{\phi}'$ from the new light to the measured lights. The recovered matte luminance pixel values for the new light are given by eq. (25) and here this reads

$$\underset{N\times 1}{\hat{\mathcal{L}}_{matte}} \;=\; \underset{N\times 6}{\boldsymbol{C}} \;\; \underset{6\times 1}{\boldsymbol{p}(\boldsymbol{a})} \tag{29}$$

and then recovered matte pixel colours are

$$\underset{N\times 3}{\hat{\mathcal{R}}_{matte}} \;=\; \underset{N\times N}{diag\left(\hat{\mathcal{L}}_{matte}\right)} \;\; \underset{N\times 3}{\boldsymbol{\chi}} \tag{30}$$

For the non-matte component, in colour, the recovered RGB values for the recovered excursion-from-matte image $\mathcal{H}$ is given by eq. (13), expressed here for excursion instead of sheen as follows:

$$\begin{array}{lcccc}
\text{excursion:} & \underset{N\times 3}{\hat{\mathcal{H}}_k} & = & \underset{N\times (n+4)}{\boldsymbol{\Psi}'_k} & \underset{(n+4)\times 1}{\boldsymbol{\phi}'}\,, \quad k = 1..3, \\
\text{interpolated colour:} & \underset{N\times 3}{\hat{\mathcal{R}}} & = & \underset{N\times 3}{\hat{\mathcal{R}}_{matte}} \;+ & \underset{N\times 3}{\hat{\mathcal{H}}}
\end{array} \tag{31}$$

7. Experiments

7.1. Reconstructing Synthetic Noisy Data: Put-One-In Analysis

Using the noisy synthetic data in Fig. 5, we can very simply test the efficacy of the weighted-LS method (Method:QUANTILE) compared to the robust approach in Method:LMS. For the 50 images of a sphere with 10% Gaussian noise, suppose we randomly choose any pair of image lighting directions and interpolate an output image re-lit from the mean of the two lighting directions. We can then compare the results with a perfect synthetic sphere (with shadows and specularities appropriately constructed) for that same lighting direction. We call this test a "put-one-in" analysis.

Such a test is simply meant to guide us in understanding the properties of the methods available. Table I shows results for three methods as follows: Method:QUANTILE is the relatively simple weighted least-squares approach set out in §4.3; Method:LMS+RBF is the proposed robust method, using LMS regression for the matte component followed by RBF interpolation for shadows and specularities; and Method:QUANTILE+RBF is the Method:QUANTILE again for the matte component, but followed by RBF interpolation for the excursion from matte. Here we run over 100 pairs out of the possible ${}_{50}C_2$ pairs of directions. Note that here we are concerned only with interpolated image appearance, not recovered surface properties.

Method	Mean PSNR	Median PSNR
QUANTILE	26.68	27.05
LMS+RBF	29.90	29.81
QUANTILE+RBF	29.90	29.81

Table 1: Mean and median PSNR (dB) for reconstructing synthetic image data, including both shadows and specularities.

Unsurprisingly, the least-squares approach Method:QUANTILE, albeit one that uses a heuristic to exclude shadows and highlights, performs worst in interpolating out-of-sample data. [7] However, the present method, Method:LMS+RBF, does quite well, in fact regenerating image data at about the same PSNR as for the noisy data used compared to perfect synthetic data. But notice that, if we did indeed use the simpler QUANTILE method for detecting shadow- and specular-weighted pixels, we would nonetheless arrive at just the same median PSNR for output images, provided we utilized RBF interpolation for the non-matte part of an interpolation image — i.e., provided we use Method:QUANTILE+RBF. This makes sense in that the RBF method is capable of excellent interpolation whatever the non-matte excursion η happens to be. Therefore this seem to argue for using Method:QUANTILE+RBF.

However, the problem arises if we also wish to recover surface normals and albedo (and chromaticity): we might indeed regenerate images that are good interpolants, but we must also ask how the methods do in recovering surface properties. From Fig. 5 we note that comparing the correct surface normals and albedos with those recovered from noisy synthetic data, Method:QUANTILE is worse than the robust method, Method:LMS. So in reconstruction, if appearance is the main concern, then it is fine to use the simple approach, Method:QUANTILE. However it makes sense to use a robust approach when we indeed are motivated to generate correct surface properties, such as in the case of museum objects.

7.2. Leave-One-Out Analysis, Real Data

Since we generally have no ground truth data for interpolants of image sets of real objects, we can test the different methods by leaving out in turn each of the input set images used in forming the interpolation

[7] Of course the PSNR in Table I does not fully capture visual saliency of the lack of non-matte information in Method:QUANTILE since the many matte pixels will indeed approximately agree with the correct synthesized image.

basis, and then regenerating the deleted image by interpolation.

For the particular image set of 50 images of which some are shown in Fig. 2, we compare values of PSNR for the reconstructed left-out image, plotted against the $x-$ and $y-$ components of the lighting direction for that left-out image, in Fig. 8(a). Here we show both the 50 data points (blue circles) as well as an error-interpolation surface. In Fig. 8(b) we also show the PSNR for Method:QUANTILE *without* any use of RBF interpolation: not surprisingly, its accuracy is much poorer and consequently its PSNR is a great deal lower. Fig. 8(c,d) show the original and interpolated rendition, for the case with error closest to the mean, for method LMS+RBF.

Table 2 gives the performance statistics for these methods. For completeness we also give the performance of a Method:0+RBF, meaning that *no* regression is carried out for the matte part, but instead only RBF interpolation is applied. We note that accuracy performance is almost the same as for Method:LMS+RBF or Method:QUANTILE+RBF, in terms of performance in regenerating the *appearance* of an interpolated image including specularities and shadows. However, such a method would return no information about the surface properties.

For an interactive, real-time viewer, though, we may not be interested in obtaining surface properties, but only interested in generating the correct appearance of interpolated results. Therefore Method:0+RBF is included in Table 2.

Method	Mean PSNR	Median PSNR	Min PSNR	Max PSNR
LMS+RBF	29.91	29.99	15.47	37.30
QUANTILE+RBF	29.95	30.00	16.43	37.36
QUANTILE	23.36	23.03	13.86	31.59
0+RBF	29.92	30.02	15.43	37.33

Table 2: PSNR statistics (dB) for reconstructing real image data including both shadows and specularities, in a leave-one-out analysis.

From the above results we can conclude that we are left with an accuracy/speed tradeoff: LMS is best for recovering surface properties but is slowest; but for any method RBF generates the best appearance for interpolants.

7.3. Further Interpolation Results

Here we visually inspect results for the interpolation algorithm Method:LMS+RBF proposed, for more datasets. Of course, another aspect of the interpolation presented is the recovery of a matte image. Fig. 9(a) [8] shows how a purely Least Squares approach to PTM (but using the modified polynomial model (3)) would produce a matte image, for lighting direction averaged from two of the 3-D input light directions. We see that the specular content in the input set of images has unduly influenced the output matte interpolant. This is not surprising since a least-squares based method is based on a sum, which is greatly influenced by absolute brightness (theoretically it has an infinite breakdown point for even one single outlier). In comparison, the weighted-LS Method:QUANTILE produces an interpolant as in Fig. 9(b) – clearly, this has gone some way to repair the inclusion of specular information but still does not appear acceptably matte. In contrast, Method:LMS generates a matte approximation $\hat{\mathcal{R}}_{matte}$ as in Fig. 9(c), a decidedly preferable result that ignores the strong highlights in the original captured dataset. The images whose lighting directions are being interpolated are shown in Fig. 9(d,e). [9] The Method:LMS+RBF interpolation result is shown in (f) for

[8] Tavern Scene painting by School of Teniers, data courteously supplied by Philip Klausmeyer of the Worcester Art Museum.

[9] The strong specularity derives from the original paint surface plus a spray-varnish applied after an adhesive (fish glue) was introduced into the cracks of the painting surface to repair damage.

an interpolation angle given by the mean of the two 3-D lighting vectors in (d) and (e), normalized. Clearly Method:LMS+RBF produces physically plausible results and note that to date no previous PTM method has been able to produce results reproducing physical highlights in this way. Image (g) shows the closest in-sample input image, which is quite close in appearance to the interpolant produced in (f). On the other hand, the interpolant is not simply the same as the closest input image, which here has lighting direction 7.0° away from from the interpolated light direction. (Note that here only 16 input lights were available for this painting. For such a small dataset one expects accurate reconstructions only when there is sufficient input-data support near enough to the interpolated light direction. In general in PTM, datasets with 40 to 70 input images are common.)

Fig. 10 shows further results: In Fig. 10, column (a) shows the input image for one lighting direction and column (b) shows that for another direction; column (c) shows the interpolated result for that non-measured angle; and column (d) shows the recovered matte image for the interpolated light direction; column (e) displays the recovered normals. Examination of these results shows them to be sensible, at least visually. Angles from the z-axis for interpolant directions in Figure 10 have median 33.95° and maximum 49.27° for these particular tests.

The three rows of images in Fig. 10-Continued are especially interesting in that the objects being imaged [10] consist largely of highly specular gold or gold leaf. Notwithstanding this difficulty the method provides a convincing and compelling interpolation.

8. Conclusion

In this paper we have introduced a robust version of PTM, based on a polynomial modified from the original method. Using the inliers found from the luminance regression, we identify both specular and shadow pixels and then model their contribution, separately or together, using an RBF interpolation. The main success of the work is in being able to interpolate data that includes both shadows and specularities for out-of-sample lighting directions, continuously and accurately. Not only do we arrive at accurate interpolants but, importantly, the images reconstituted for actually measured directions are always essentially correct.

One disadvantage of using a robust regression method is that the number of lights used must inherently be larger than the minimum number had we used least squares. For if we have 6-D regression coefficients then using LS we could use a minimum of 6 lights; [11] but for LMS regression we need more than twice as many observations as variables, so a minimum of 13 lights is required. However, the advantage is that the regression is not swayed by specular and shadow outliers. In practice, we found that interpolation could proceed even with a relatively small number of lights if necessary — as in Fig. 9 for a basically flat but highly specular surface that uses only 16 lights.

We found that the robust method returns more accurate surface normals, surface colours, and albedos in the face of additive noise than other methods tested. As well, to test accuracy of appearance interpolation we carried out two different error analyses. In the first, we used noisy synthetic data and generated an interpolant in a new, non-measured direction equal to the mean between two sampled directions, repeating the test for 100 randomly chosen direction pairs. We compared the interpolant against a noise-free synthesized image generated with correctly modelled shadows and specular highlights. The result was that using our shadows and specular interpolation, we achieved about the same accuracy as simply the difference between the noisy and non-noise data, thus showing that the method performed as accurately as possible. In a second analysis,

[10] Data was courteously supplied by Mark Mudge of Cultural Heritage Imaging. Licensing is via Creative Commons license http://creativecommons.org/licenses/by-nc-nd/3.0/us/

[11] However, the use of only 6 lights is not practical for other reasons, namely that in practice the quality of the PTM normals recovered would not be good enough to produce useful reflectance transformation results.

for real data we left out one measured image and regenerated that image, repeating over all the images in the dataset. We found that using the RBF approach substantially increased accuracy.

We note from the plot of error in the Leave-One-Out analysis in §7.2 that in general the error increases as the light-direction interpolation direction approaches the perimeter of sampled directions – there is less information to interpolate. Also, insisting on exact reproduction of in-sample inputs will lead to overfitting the data to some degree. A small amount of smoothing will maintain accuracy of in-sample images while reducing overshooting between nodes. As well, clearly the quality of interpolants will depend on the quality of input data sets. Here, we have used JPEG-compressed inputs only, stored using lossy compression.

Overall we found that there is a tradeoff between model complexity versus accuracy of surface properties recovery. Even a simplest, agnostic, model of matte properties can return accurate rendering of image *appearance* for unmeasured interpolants. A model with moderate complexity can also do well in interpolation; but only a robust model can return both correct imagery and correct surface properties.

Visually, we found that for further image sets tested the interpolated results make sense and, for the first time, incorporate specularities and shadows into PTM results in a way that shows how the actual surface would indeed appear under a new light. The proposed approach thus runs counter to the previous objective of seeing matte versions of objects under new lights and, instead of avoiding specular highlights, here we include them. In the original PTM work [1], specularities were subsequently regenerated using a standard, but unphysical, graphics model of highlights based on the recovered normals.

So instead, what we aim at here is seeing specularities (and shadows) as the real surface would generate them – not synthesizing them via graphics. The robust approach does deliver a more accurate set of surface normals, so in fact one could still generate non-photorealistic results as in the original work, including reflectance transformation results. Also, more accurate normals make it more likely that one could accurately regenerate the 3D surface itself, e.g. the geometry of a wrinkled parchment document. In future work, we intend to examine how the method lends itself to this objective, with a view to better comparing surfaces before and after damage or restoration. Future work also would involve addressing the fairly large number of regression coefficients carried at each pixel, more than the amount of input data itself, with a subspace reduction approach being possibly useful for reducing this storage burden.

In summary, the method presented represents a novel realistic re-lighting scheme as well as a promising new approach for more accurately accessing colour and surface properties of imaged objects.

References

[1] T. Malzbender, D. Gelb, H. Wolters, Polynomial texture maps, in: Computer Graphics, SIGGRAPH 2001 Proceedings, 2001, pp. 519–528.

[2] R. J. Woodham, Photometric method for determining surface orientation from multiple images, Optical Engineering 19 (1980) 139–144.

[3] M. Mudge, Cultural Heritage Imaging, personal communication (2010).

[4] M. Mudge, T. Malzbender, C. Schroer, M. Lum, New reflection transformation imaging methods for rock art and multiple-viewpoint display, in: 7th Int. Symp. on Virtual Reality, Archaelogy and Cultural Heritage, 2006.

[5] K. Sunkavalli, T. Zickler, H. Pfister, Visibility subspaces: Uncalibrated photometric stereo with shadows, in: ECCV2010, 2010, pp. II: 251–264.

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

[7] E. Coleman, R. Jain, Obtaining 3-dimensional shape of textured and specular surface using four-source photometry, Comp. Graphics and Image Proc. 18 (1982) 308–328.

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

[9] S. Barsky, 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) (2003) 1239–1252.

[10] H. Rushmeier, G. Taubin, A. Gueziec, Applying shape from lighting variation to bump map capturing, in: Proceedings of Eurographics Rendering Workshop 1997, 1997, pp. 35–44.

[11] D. Miyazaki, K. Hara, 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, 2007, pp. 325–329.

[12] E. Coleman, R. Jain, Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry, in: L. Wolff, S. Shafer, G. Healey (Eds.), Physics-Based Vision: Principles And Practice. Shape Recovery, Jones and Bartlett Publishers, Inc., 1992, pp. 180–199.

[13] A. Yuille, D. Snow, Shape and albedo from multiple images using integrability, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, p. 158.

[14] J. Sun, M. Smith, L. Smith, S. Midha, 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) (2007) 1050–1057.

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

[16] P. J. Rousseeuw, Least median of squares regression, J. Amer. Stat. Assoc. 798 (1984) 871–880.

[17] C. Julia, A. Sappa, F. Lumbreras, J. Serrat, A. Lopez, Photometric stereo through an adapted alternation approach, in: Proceedings of IEEE Conference on Image Processing, 2008, pp. 1500–1503.

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

[19] F. Verbiest, L. VanGool, Photometric stereo with coherent outlier handling and confidence estimation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[20] C. Hernández, G. Vogiatzis, R. Cipolla, Shadows in three-source photometric stereo, in: Proceedings of the 10th European Conference on Computer Vision, 2008, pp. 290–303.

[21] V. Argyriou, S. Barsky, M. Petrou, Generalisation of photometric stereo technique to q-illuminants, in: Proceedings of 19 th British Machine Vision Conference, 2008.

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

[23] T. Zickler, S. Enrique, R. Ramamoorthi, P. Belhumeur, Reflectance sharing: image-based rendering from a sparse set of images, in: Eurographics Symposium on Rendering, 2005, pp. 253–265.

[24] T. Zickler, R. Ramamoorthi, S. Enrique, P. Belhumeur, Reflectance sharing: Predicting appearance from a sparse set of images of a known shape, IEEE Trans. on Patt. Anal. and Mach. Intell. 28 (2006) 1287–1302.

[25] W.-C. Ma, T. Hawkins, P. Peers, C. Felix Chabert, M. Weiss, P. Debevec, Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination, in: Eurographics Workshop on Rendering, 2007, pp. 183–194.

[26] M. Holroyd, J. Lawrence, G. Humphreys, T. Zickler, A photometric approach for estimating normals and tangents, in: ACM SIGGRAPH Asia, 2008, pp. 133:1–133:9.

[27] V. Masselus, P. Peers, P. Dutré, Y. D. Willems, Smooth reconstruction and compact representation of reflectance functions for image-based relighting, in: Eurographics Symposium on Rendering, 2004, pp. 287–298.

[28] J. Lawrence, A. Ben-Artzi, C. DeCoro, W. Matusik, H. Pfister, R. Ramamoorthi, S. Rusinkiewicz, Inverse shade trees for non-parametric material representation and editing, ACM Trans. on Graphics 25 (3) (2006) 735–745.

[29] M. Fuchs, H. P. A. Lensch, V. Blanz, H.-P. Seidel, Superresolution reflectance fields: Synthesizing images for intermediate light directions, Comput. Graph. Forum 26 (3) (2007) 447–456.

[30] J. Padfield, D. Saunders, T. Malzbender, Polynomial texture mapping: A new tool for examining the surface of paintings, in: ICOM Committee for Conservation, Vol. 1, 2005, pp. 504–510.

[31] Y. Hammer, S. Bengtson, T. Malzbender, D. Gelb, Imaging fossils using reflectance transformation and interactive manipulation of virtual light sources, in: Manipulation of Virtual Light Sources, Palaeontologia Electronica, 2002.

[32] M. Mudge, J. Davis, R. Scopigno, M. Doerr, A. Chalmers, O. Wang, P. Gunawardane, T. Malzbender, Image-based empirical information acquisition, scientific reliability, and long-term digital preservation for the natural sciences and cultural heritage, in: Eurographics Tutorials, 2008.

[33] M. Drew, N. Hajari, Y. Hel-Or, T. Malzbender, Specularity and shadow interpolation via robust polynomial texture maps, in: BMVC 2009, British Machine Vision Conf., 2009.

[34] J. Happa, M. Mudge, K. Debattista, A. Artusi, A. Gonçalves, , A. Chalmers, Illuminating the past – State of the art, in: VAST'09: The 10th Int. Symp. on Virtual Reality, Archaeology and Cultural Heritage, 2009.

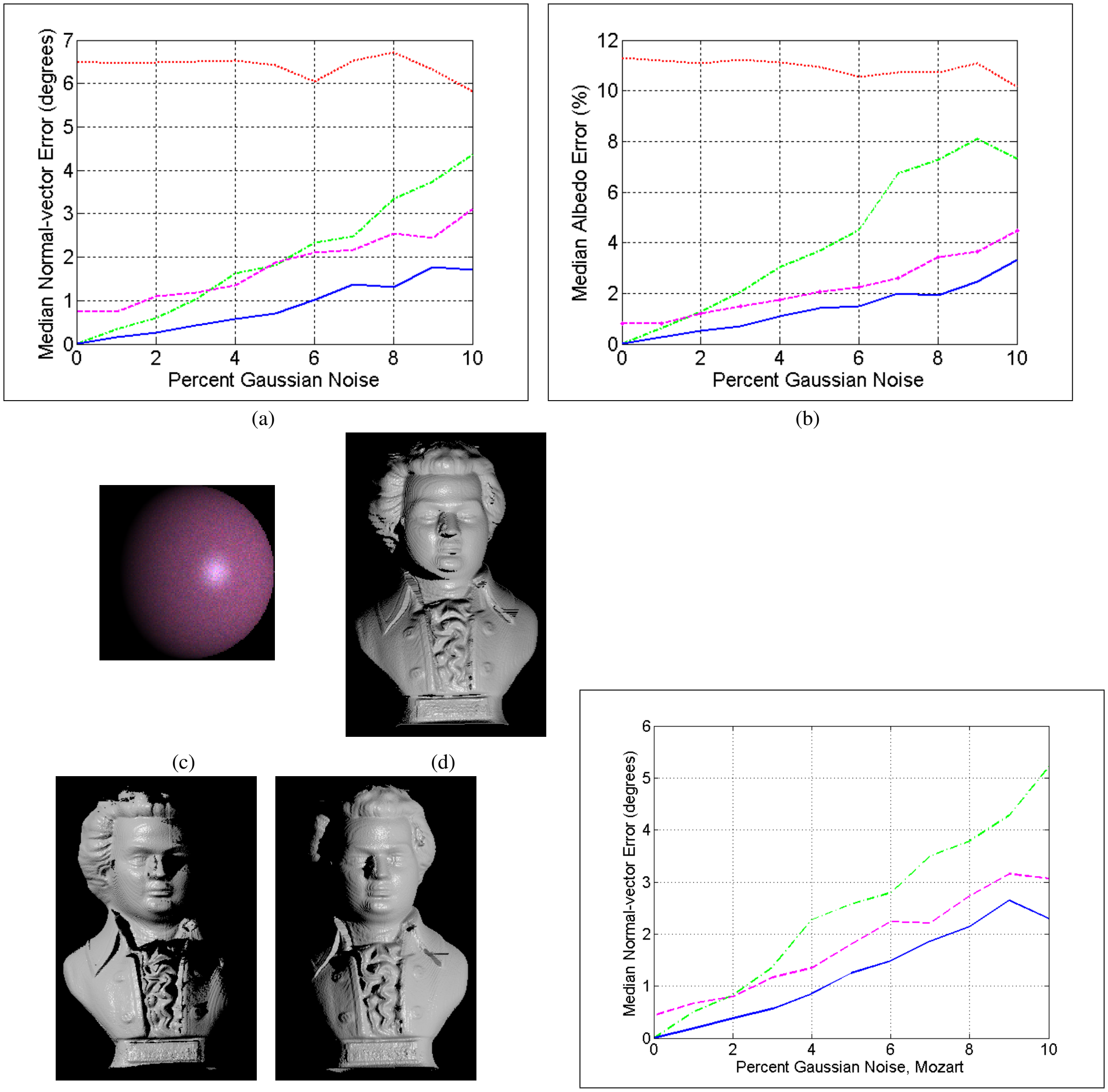
[35] R. Basri, D. Jacobs, Lambertian reflectance and linear subspaces, IEEE Trans. Patt. Anal. and Mach. Intell. 25 (2003) 218–233.

[36] H. Barrow, J. Tenenbaum, Recovering intrinsic scene characteristics from images, in: A. Hanson, E. Riseman (Eds.), Computer Vision Systems, Academic Press, 1978, pp. 3–26.

[37] B. Phong, Illumination for computer generated pictures, Commun. ACM 18 (6) (1975) 311–317.

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

[39] B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, K. R. Subramanian, Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions, in: SMI'01: Proceedings of the International Conference on Shape Modeling & Applications, 2001, p. 89.



(a) (b)

(c) (d)

(e) (f) (g)

Figure 5: Noise sensitivity for PST estimate of surface normal and albedo for synthetic images. (a): Median angular error of recovered surface normal for synthetic sphere images. (b): Median percent albedo error. Solid blue line: robust PST – using weights w^0 generated by robust PTM (Method:LMS). Dashed magenta line: robust PST using weights from Method:QUANTILE. Dot-dashed green line: Method:COEFF. Dotted red line: standard PST. (c): Synthetic sphere with 10% Gaussian noise. (d-f): Shaded (noisy) laser range finder depth map, with attached and raycasting cast shadows. (Three out of set of 50 images are shown.) (g): Gaussian-noise sensitivity for median angular error for Mozart image set (standard PST results are high and are similar to those in (a)).

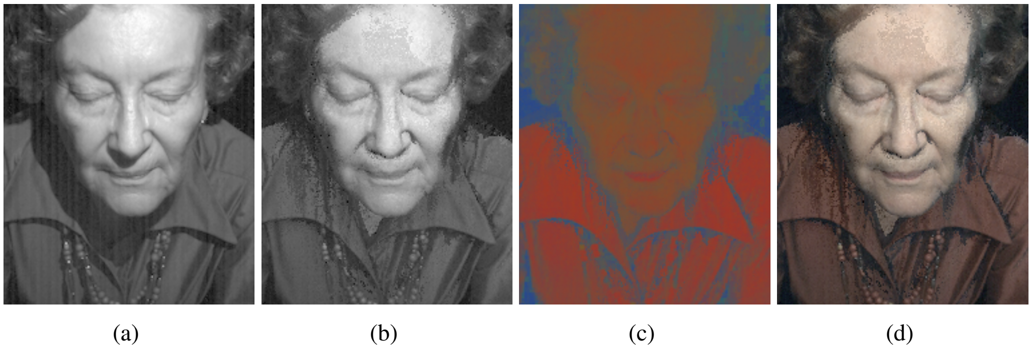


Figure 6: (a): Actual luminance. (b): Recovered luminance for matte component. (c): Recovered chromaticity. (d): Approximated colour version for matte contribution.

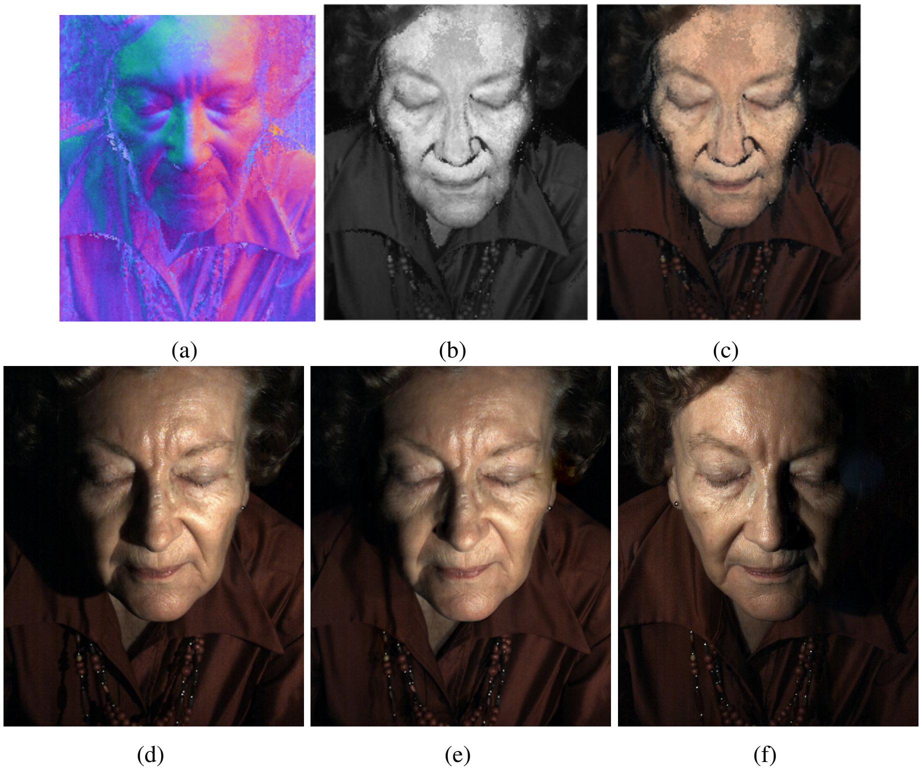
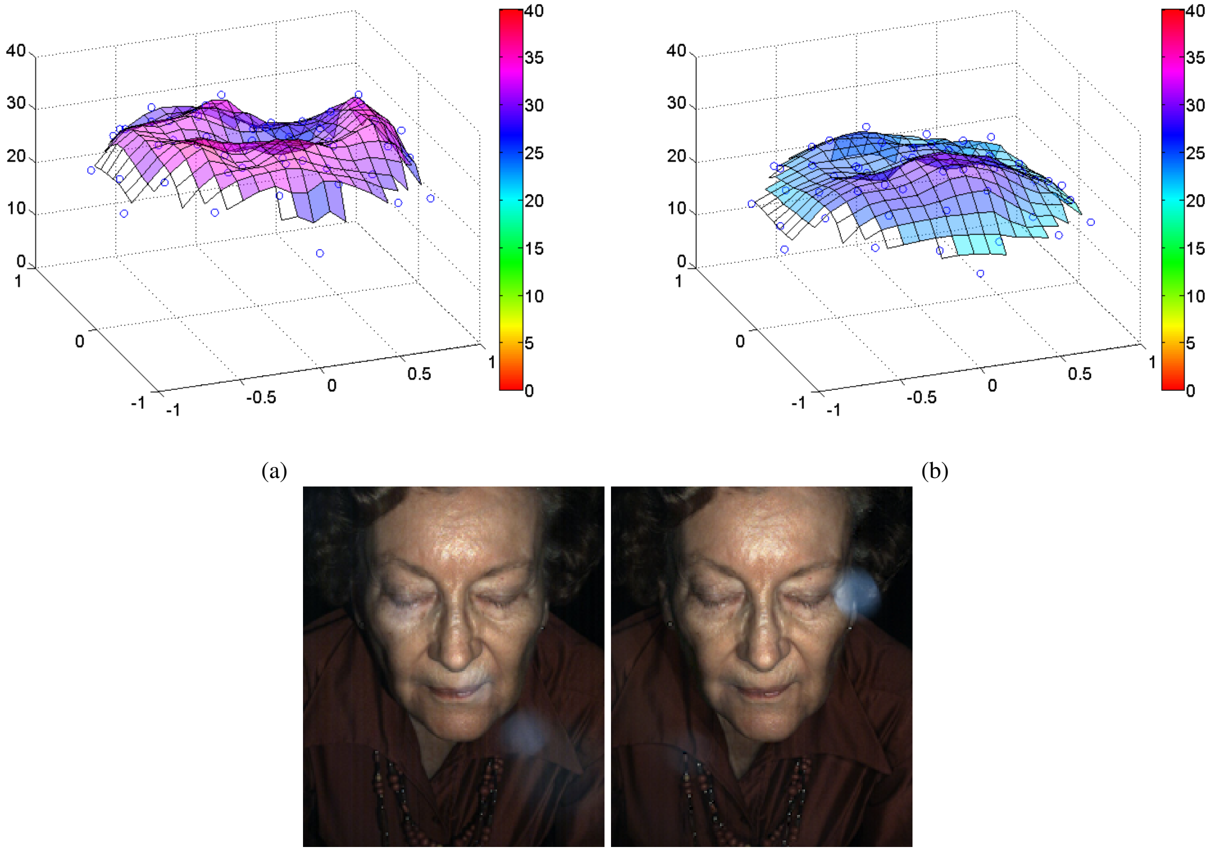


Figure 7: (a): Recovered surface normal components, in pseudocolour. (b): Recovered surface albedo. (c): Intrinsic image. (d): To compare to an in-sample image, an input set image $[(u, v) = (0.61, -0.29)]$ is displayed that has the closest light-direction to one of the interpolants, displayed in (e). (e): Interpolant for direction $(u, v) = (0.49, -0.41)$ that is close to the in-sample image displayed in (d), with angular difference 10.01° from in-sample image. (f): Another interpolant, for relighting direction (u, v)=(-0.67,-0.05), showing interpolation of cast shadows as well as of specularities.



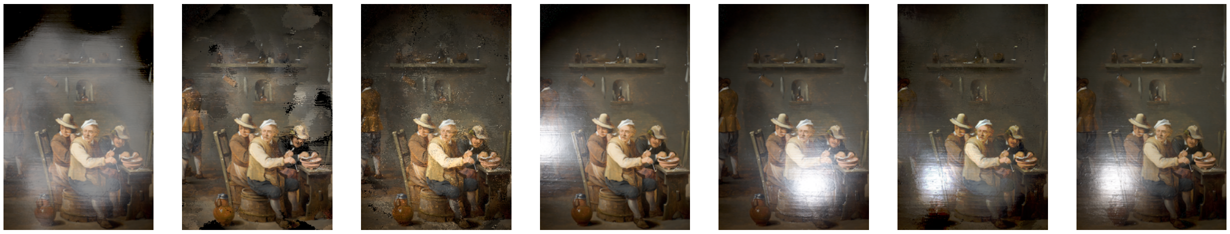
(a)

(b)

(c)

(d)

Figure 8: (a): PSNR values for Method:LMS+RBF in a leave-one-out analysis (for the dataset partially shown in Fig. 2). PSNR is plotted versus the $x-$ and $y-$components of the lighting direction of the left-out image. (b): PSNR values for Method:QUANTILE, a weighted-LS algorithm. (c): Original, left out, for case closest to median error, [(u, v)=(0.13,-0.15)]. (d): Interpolant for left-out image, PSNR=30.07.



(a) (b) (c) (d) ⟶ (e) ⟵ (f) (g)

Figure 9: (a): Matte result from LS fit; (b): matte result from weighted-LS fit in Method:QUANTILE; (c): matte result from robust fit. (d,e): Input images for lights $(u, v) = (-0.39, -0.15)$ and $(u, v) = (0.16, -0.38)$: lights in (d,e) are 34.8° apart; (f): Method:LMS+RBF robust interpolation onto lighting direction $(u, v) = (-0.12, -0.28)$, which is 17.4° from either light in (d,e). Note that of course there are indeed 16 input images used in generating the interpolant, not just the two images (d,e) shown above. We display (d,e) to show that the interpolated light direction is sensibly interpolated in (f). For comparison, (g) shows the closest image in the input set, with light $(u, v) = (-0.17, -0.38)$, which is 7.0° away from the interpolated light direction. This verifies that the interpolation is generating reasonable images, whilst not simply choosing the nearest-neighbour light.

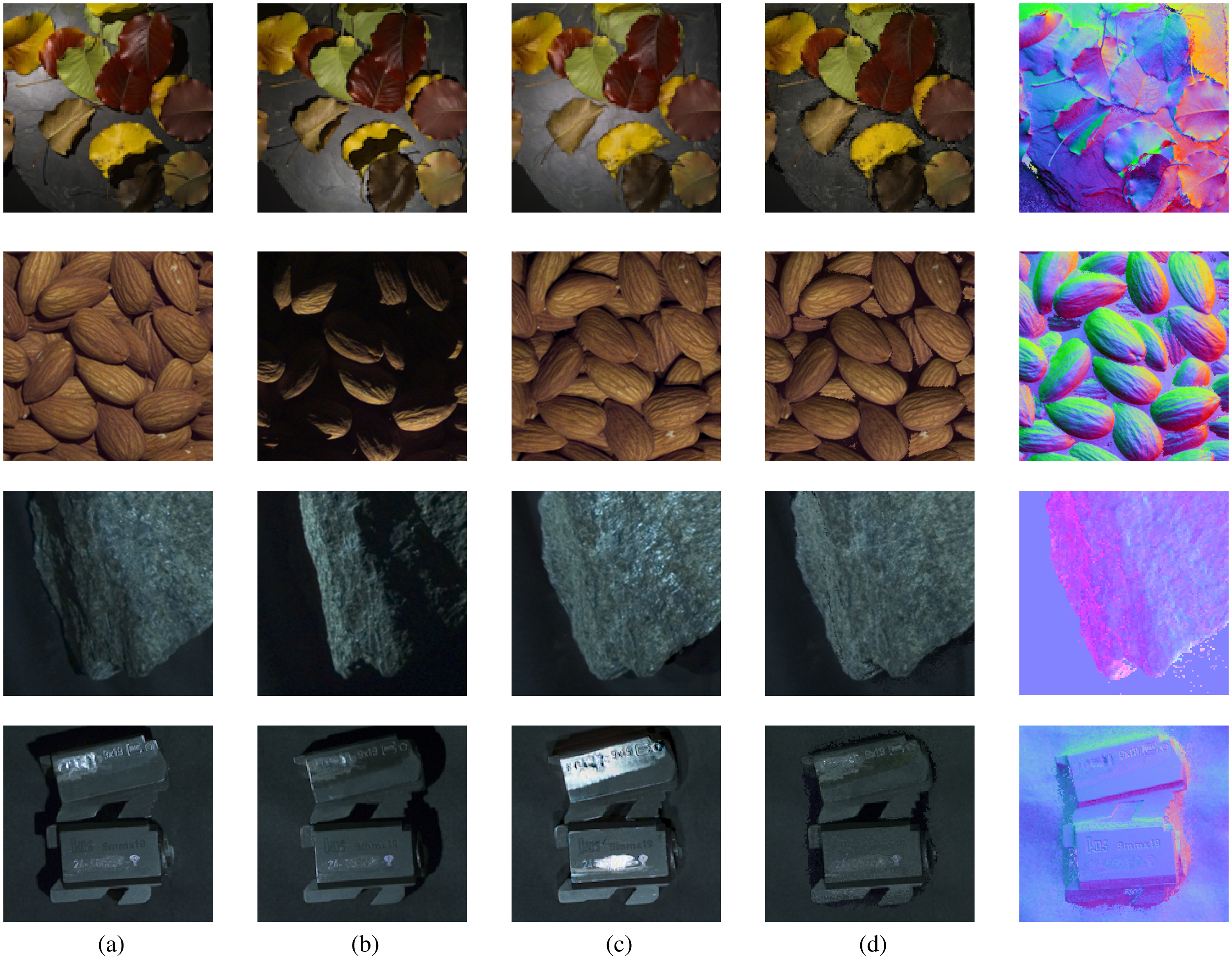


Figure 10: (a,b): Input images with lighting directions to be interpolated; (c): interpolants; (d): robust matte image for interpolated angle; (e): normals. Interpolating directions for (c) are (u, v)=0.03,-0.04), (-0.17,0.53), (0.19,-0.57), (-0.17,0.53).

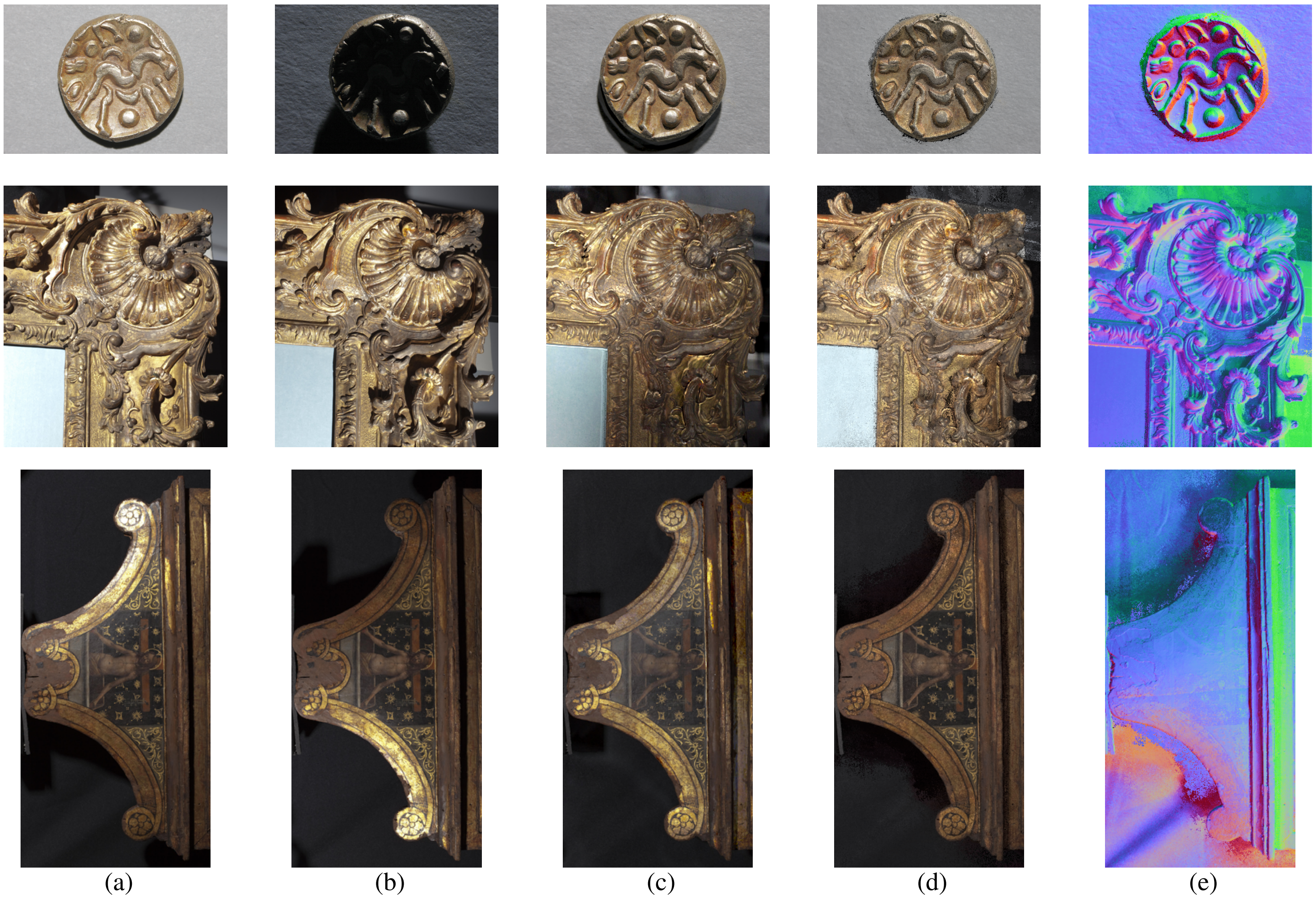


Figure 10: - Continued: (a,b): Input images with lighting directions to be interpolated; (c): interpolants; (d): robust matte image for interpolated angle; (e): normals. Interpolating directions for (c) are (u, v)=(-0.42,0.0), (-0.05,-0.13), (0.07,-0.41).