

DENSE PHOTOMETRIC STEREO: A MARKOV RANDOM FIELD APPROACH

Tai-Pang Wu Kam-Lun Tang Chi-Keung Tang Tien-Tsin Wong

Abstract

We address the problem of robust normal reconstruction by *dense photometric stereo*, in the presence of complex geometry, shadows, highlight, transparencies, variable attenuation in light intensities, and inaccurate estimation in light directions. The input is a dense set of noisy photometric images, conveniently captured by using a very simple set-up consisting of a digital video camera, a reflective mirror sphere, and a handheld spotlight. We formulate the dense photometric stereo problem as a Markov network, and investigate two important inference algorithms for Markov Random Fields (MRFs) – graph cuts and belief propagation – to optimize for the most likely setting for each node in the network.

In the *graph cut* algorithm, the MRF formulation is translated into one of energy minimization. A discontinuity-preserving metric is introduced as the compatibility function, which allows α -expansion to perform efficiently the maximum a posteriori (MAP) estimation. Using the identical dense input and the same MRF formulation, our *tensor belief propagation* algorithm recovers faithful normal directions, preserves underlying discontinuities, improves the normal estimation from one of discrete to continuous, and drastically reduces the storage requirement and running time. Both algorithms produce comparable and very faithful normals for complex scenes. Although the discontinuity-preserving metric in graph cuts permits efficient inference of optimal discrete labels with a theoretical guarantee, our estimation algorithm using tensor belief propagation converges to comparable results but runs faster because very compact messages are passed and combined. We present very encouraging results on normal reconstruction. A simple algorithm is proposed to reconstruct a surface from a normal map recovered by our method.

With the reconstructed surface, an inverse process, known as relighting in computer graphics, is proposed to synthesize novel images of the given scene under user-specified light source and direction. The synthesis is made to run in real time by exploiting the state-of-the-art graphics processing unit (GPU). Our method offers many unique advantages over previous relighting methods, and can handle a wide range of novel light sources and directions.

Keywords

Photometric stereo, Markov Random Fields, belief propagation, graph cuts, normal and surface reconstruction, robust inference, real-time relighting.

I. INTRODUCTION

Since Woodham [44] proposed *photometric stereo* there has been extensive theoretical and experimental research on the problem. While approaches in photometric stereo using two views with known albedos [44], three views [15], four views [7], [35], [3], more views [22], complex reflectance models [28], [37], [18], [35], lookup tables [44], [45], reference objects [16], [13],

[10], and novel object representation [4] have been reported, photometric stereo is still considered to be a difficult problem in the presence of shadows and specular highlights, and for objects with complex material and geometry.

Inspired by [24] where robust stereo reconstruction was achieved by using a *dense* set of images, and by [36] in which a *Markov* network was used to formulate the problem of geometric stereo reconstruction, in this paper, we propose to address the problem of dense photometric stereo by employing the Markov Random Field (MRF) approach to reconstruct dense surface normals from a dense set of photometric images, which can be conveniently captured using a very simple set-up consisting of a handheld spotlight, a reflective mirror sphere and a digital video (DV) camera. Our approach not only infers the piecewise smooth normal field, but also preserves the underlying orientation discontinuities and rejects noises caused by highlight and shadows. As we shall see, the availability of dense data effectively copes with non-Lambertian observations inherent in the dense set. Using the dense data, the initial normal at a pixel is obtained, which is used as the local evidence in a MRF network for solving the problem. A simple surface reconstruction algorithm is proposed to generate an acceptable surface from our recovered normal maps. We shall investigate two important MRF inference algorithms:

Graph cuts (GC) In the first method, we translate the MRF model for dense photometric stereo into an energy function. Estimating the MRF-MAP solution is equivalent to minimizing the corresponding energy function. The MAP estimation can be efficiently performed by the graph cut algorithm [20], where the data term is encoded using the local evidence identical to that used in our belief propagation algorithm. We show that the smoothness term can be encoded into a discontinuity-preserving metric, thus making the more efficient α -expansion [6] rapidly converge to an optimal solution w.r.t. the discrete label space with a *theoretical* guarantee, instead of the slower swap move [6] in a pairwise MRF. Similar to [19], the smoothness constraint is enforced while geometric discontinuities are preserved. In contrast to [19], however, while the energy function we minimize is still regular, our noisy photometric data are treated asymmetrically by resampling the dense and scattered data into an unbiased set.

Tensor belief propagation (TBP) Our second method uses a MRF network where hidden nodes receive initial messages derived using local evidences. These nodes communicate among each other by belief propagation to infer smooth structures, preserve discontinuities and reject noises. In this paper, we propose a new and very fast tensor-based message passing scheme for producing an approximate MAP solution of the Markov network. Although it is an algorithm that estimates the solution, it produces comparable results to GC. Besides, it allows continuous

estimation of normal directions, runs very fast and requires significantly less memory compared to traditional message passing used in belief propagation.

The preliminary versions of this paper have appeared in [38] and [46] where the two inference algorithms were developed independently and were based on different MRF models. In this paper, we evaluate and compare the robustness and efficiency of the two inference algorithms based on the same MRF formulation and using the same input. For high precision normal reconstruction, the graph cut algorithm converges with a theoretical guarantee to an optimal solution in a few iterations. We have improved the graph cut algorithm in this paper, making the system runs much faster than the algorithm presented in [46]. The metric proof has also been revised due to the use of a robust metric in encoding the smoothness term. On the other hand, because the traditional belief propagation is intractable due to the prohibitive size of a message encoded in the conventional way, we propose tensor message passing to approximate the MAP solution, by transforming the estimation from one of discrete to continuous. While results comparable to those produced by graph cuts are obtained, both running time and storage requirement are significantly reduced. Comparing with [38], [46], this paper presents a complete coverage of the two methods. More quantitative evaluation are performed using real as well as synthetic data. Finally, we propose a novel and *real-time* method on relighting based on our photometric stereo reconstruction.

The organization of this paper is as follows: Section II reviews the related work. Section III describes the image capturing system for collecting our dense data. Section IV details the initial normal estimation and the MRF approach for dense photometric stereo. The two inference algorithms are then described in detail. Section V describes the energy minimization by graph cuts. Section VI describes our tensor belief propagation. We present our algorithm on surface reconstruction from normals in section VII. Based on the same MRF formulation and identical dense input, the two normal reconstruction methods are evaluated and compared in section VIII. We present results of normal and surface reconstruction on real and noisy data in section IX. Finally, in section X, using the reconstructed surface, we propose an inverse process to synthesize novel images for the input scene under user-specified lighting directions. By making use of recent hardware technology, the process is made to run in real time. The process is alternatively and better known as real-time relighting in computer graphics. Our method provides many unique advantages in comparison with previous relevant relighting method.

II. RELATED WORK

Woodham [44] first introduced photometric stereo for Lambertian surfaces. In this work, three images are used to solve the reflectance equation for recovering surface gradients p, q and albedo ρ of a Lambertian surface:

$$R(p, q) = \rho \frac{l_x p + l_y q + l_z}{\sqrt{1 + p^2 + q^2}} \quad (1)$$

where $p = \frac{\partial z}{\partial x}$, $q = \frac{\partial z}{\partial y}$ are the unknown surface gradients, $[l_x \ l_y \ l_z]^T$ is the known unit light direction. Later, Belhumeur and Kriegman [5] showed that the set of images of a convex Lambertian object forms a convex polyhedron cone whose dimension is equal to the number of distinct normals, and that this cone can be constructed from three properly chosen images. Many approaches have been proposed to address the photometric stereo problem:

Four images Coleman and Jain [7] used four photometric images to compute four albedo values at each pixel, using the four combinations involving three of the given images. In the presence of specular highlight, the computed albedos will not be identical, which indicates that some measurement must be excluded. In [35], four images were also used. Barsky and Petrou [3] showed that [7] is still problematic if shadows are present, and generalized [7] to handle color images. In these methods, little neighborhood information is considered so they are sensitive to noise caused by incorrect estimation in light directions or violations to the Lambertian model.

Reference objects In [16], a reference object was used to perform photometric stereo, in which isotropic materials were assumed. In this approach, the outgoing radiance functions for all directions are tabulated to obtain an empirical reflectance model. Hertzmann and Seitz [13] used a similar technique to compute surface orientations and reflectance properties. The authors made use of their proposed orientation consistency to establish the correspondence between an unknown object and a known reference object. In many cases, however, a reference object for establishing correspondence is unavailable. A simplified reflectance model will then be used.

Reflectance models By considering diffuse and non-Lambertian surfaces, Tagare and deFiguereiredo [37] developed a theory on m -lobed reflective map to solve the problem. Kay and Caely [18] extended [37] and applied nonlinear regression to a larger number of input images. Solomon and Ikeuchi [35] extended [7] by separating the object into different areas. The Torrance-Sparrow model was then used to compute the surface roughness. Nayar et al [28] used a hybrid reflectance model (Torrance-Sparrow and Beckmann-Spizzichino), and recovered not only the surface gradients but also parameters of the reflectance model. In these approaches, the models used are usually somewhat complex, and a larger number of parameters are estimated.



Fig. 1. Two typical noisy photometric images for *Snail* captured by our simple system. (a) is significantly contaminated by shadows, and (b) is corrupted by highlight. (c) A typical trajectory of the estimated light directions shows that they are scattered and very noisy.

Basri and Jacobs [4] used low-order spherical harmonics to encode Lambertian objects. They assumed isotropic and distant light sources. Lighting may be unknown or arbitrary. Shape recovery is then performed in a low-dimensional space. Goldman et al [10] proposed a photometric stereo method that recovers the shape (normals) and BRDFs using an alternating optimization scheme. Unlike their earlier work [13], a reference object is not needed, which is solved as part of the reconstruction process. The BRDF model used is the Ward model. Since they only used a sparse set of samples, the light calibration should be accurate, and severe highlight and cast shadows must be absent. They built an interactive relighting system whereas we built a real-time relighting system that supports very fast frame rate and more versatile lighting effects (see the supplementary video). Our relighting approach does not require recovery of material properties and any assumption on the reflectance model.

To our knowledge, there is no previous work using belief propagation or energy minimization via graph cuts to address the problem of (dense) photometric stereo. The use of a dense set of photometric stereo data (> 100) has not been extensively explored, possibly due to the difficulty in producing hundreds of accurate light directions, while our approach is robust against inaccurate and scattered estimations in light directions sampled by our simple capturing system. An earlier work [22] investigated two algorithms: the parallel and cascade photometric stereo for surface reconstruction which use a larger number of images. A related work using one image, that is, shape from shading, was reported in [17], where the problem was solved via graph cuts, by combining local estimation based on local intensities and global energy minimization.

Note that exact inference in the Markov network with *loops* is intractable. Algorithms that approximate the solution such as loopy belief propagation or Pearl’s algorithm [32] have been employed. For energy minimization by graph cuts [20], the conditions for an energy function that can be minimized was described and a fast implementation is currently available. The

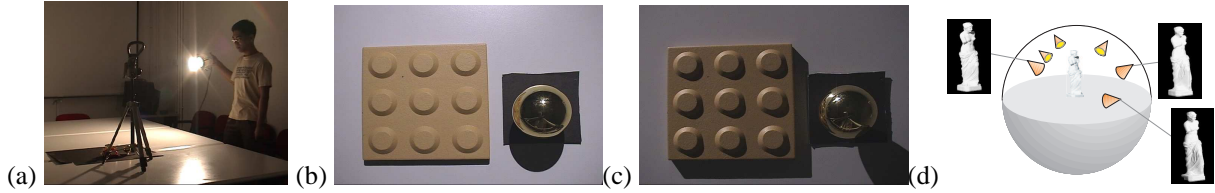


Fig. 2. (a) A typical scenario of data capturing. (b)–(c): Two views of the experimental set-up under different illumination. (d) The captured images correspond to a scattered point set on the light direction sphere.

converged solution given by graph cuts is optimal “in a strong sense” [20], that is, within a known factor of the global optimal solution.

III. DATA CAPTURING

In this section, we first describe our very simple system for efficiently capturing a dense set of photometric images. The light directions and photometric images we capture are very noisy (Fig. 1). Unlike certain approaches in photometric stereo where high-precision capturing systems were built, we propose to resample the dense and noisy observations to infer a uniform set, from which robust normal plane fitting can be performed (section IV) to estimate \tilde{N}_s at each pixel s . The initial normals will be used to encode the matching cost for belief propagation, or encoded into the robust data term in energy minimization using graph cuts.

Our system is inspired by [13] where a reference object of known geometry was used to find out surface normals of the target object. They performed matching on bidirectional reflectance distribution function (BRDF) response based on the orientation-consistency cue, where the specular highlight implicitly gives the surface *normal direction*. The reference object should be similar to the target object in material. On the other hand, our approach explicitly uses the specular highlight to estimate the *light direction*, which is used to obtain the initial surface normal at each pixel. No reference object of similar material is used.

A. Light calibration

Our robust dense photometric stereo requires acceptable estimated light directions but they need not be very accurate. In fact, our proposed light calibration method is very simple. Shown in Fig. 2(a) is our experimental set-up, where two views of the object and a mirror sphere under different illuminations are depicted in Fig. 2(b)–(c).

A video camcorder is used to capture a sequence of images by changing the direction of the light source which is a handheld spotlight. The auto-exposure function of the video camcorder is turned off when the video is captured. In our experiments, we tried to hold the spotlight at a

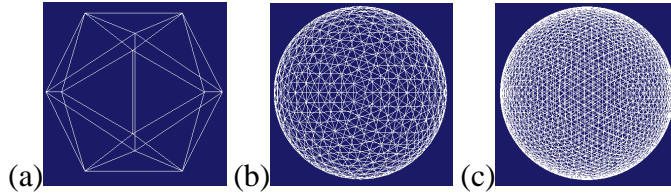


Fig. 3. Icosahedron: (a) shows the original icosahedron with 20 base faces. In (b) and (c), each face of (a) is subdivided into 4 equilateral triangles recursively in total of 4 times and 5 times, respectively

constant distance from the object so as to maintain a constant irradiance impinging on the object. But it is difficult to achieve using a handheld spotlight, and therefore our images suffer various degrees of attenuation in light intensity. To sample as many directions as possible that cover the half space containing the object (Fig. 2(d)), it is inevitable that the shadows of the wires, the camera tripod and the camera itself are cast onto the target object. Therefore, missing directions are not uncommon in a typical set of sampled images. The captured images thus represent a coarse and scattered collection of photometric responses over the light directions sampled on a unit hemisphere (Fig. 2(d)). This mirror sphere approach was not adopted in [10] because sparse samples were used in their photometric stereo method, where light calibration is more critical to the reconstruction accuracy. In our method, we estimate the light direction by locating the mirror reflection, or the brightest point on the mirror sphere. By searching for the maximum intensity, we can readily localize the point of reflection. Since we know the geometry of the sphere and the viewing direction which is assumed to be orthographic, by Snell’s law, the light direction is given by $L = 2N(N \cdot H) - H$ where N is the known surface normal at the brightest pixel (a, b) , $H = [0 \ 0 \ 1]^T$ and L is the estimated light direction. N can be determined given (a, b) , the image of the sphere center (c_x, c_y) , and the image of the sphere radius r . Under orthographic projection, we can measure (c_x, c_y) and r directly on any captured image.

In practice, the light source direction is located on the upper hemisphere containing the object (Fig. 2(d)). So, to minimize the error caused by reflections not due to the light source (e.g. from the table where the object and the sphere are placed), we have to limit the search space of the maximum intensity by considering only the pixels (x, y) satisfying $(x - c_x)^2 + (y - c_y)^2 < r^2 - r^2 \cos(\frac{\pi}{4}) - \epsilon$ where $\epsilon > 0$ is a small constant to offset the small error caused by the measured r, c_x and c_y . Using this condition, all light coming from the half space containing the lower hemisphere of the reflective sphere will be automatically discarded.

B. Uniform resampling

There are two reasons to perform uniform resampling on the captured dense data. First, the data volume and biases will be drastically reduced after resampling. Note that we capture a video sequence at 30 frames/sec, and typically we spend five minutes to capture a data set. The second reason is to partially leverage noise rejection to data resampling. Noise are typically caused by inaccurate estimation of light directions and non-Lambertian observations. As we shall see, our resampling is implemented by image interpolation which helps to smooth out outliers.

The data acquired by the above setup corresponds to a scattered point set on the light direction sphere where undesirable biases are present. To infer a set of light direction samples uniformly distributed on a unit sphere, we use a uniform unit icosahedron and subdivide on each face four times recursively [2] (Fig. 3). Suppose that the object is located at the center of a unit sphere which contains the uniform unit icosahedron after subdivision. Ideally, we want to illuminate the object along the line joining the center and the vertices of the subdivided icosahedron to achieve uniform distribution. In practice, for each light direction L_o at a given vertex of the subdivided icosahedron, we seek a set of light directions L_i that are closest to L_o , and obtain the image I_o at L_o by interpolating the corresponding images I_i at L_i using $I_o(x, y) = \sum_{i \in \mathcal{V}} \frac{L_o \cdot L_i}{\sum_{i \in \mathcal{V}} L_o \cdot L_i} I_i(x, y)$ where \mathcal{V} is a set of indices to the captured light directions that are closest to L_o . Typically, the input data size is reduced to several hundreds after uniform resampling.

IV. INITIAL NORMALS AND THE MRF MODEL FOR DENSE PHOTOMETRIC STEREO

Given a dense set of images captured at a fixed viewpoint with their corresponding distant light directions, our goal is to find the optimal normal vector N_s at each pixel s .

Initial normal estimation: dense vs. sparse We describe how to estimate the initial \tilde{N}_s at each pixel s by making use of the intensity ratios derived from the *dense* and noisy input. As we shall demonstrate, given noisy input, the following method proves to be infeasible for *sparse* input but works for dense and noisy input where the inherent redundancy is invaluable in estimating \tilde{N}_s .

Suppose that the object is Lambertian. Then, the reflectance at each pixel s is described by $\rho_s(\tilde{N}_s \cdot L_s)$, where ρ_s is the surface albedo, \tilde{N}_s is the normal and L_s is the light direction at the pixel s . Note that \tilde{N}_s and ρ_s are the same for all corresponding pixels in the sampled images.

We use the *ratio image* approach to eliminate ρ_s and obtain the initial estimate \tilde{N}_s . Ratio image was proposed in [34] for surface detail transfer. Alternatives for estimating \tilde{N}_s such as the minimization of the residual $\|I_s - \rho_s(N_s \cdot L_s)\|^2$ are also possible. However, because ρ_s itself

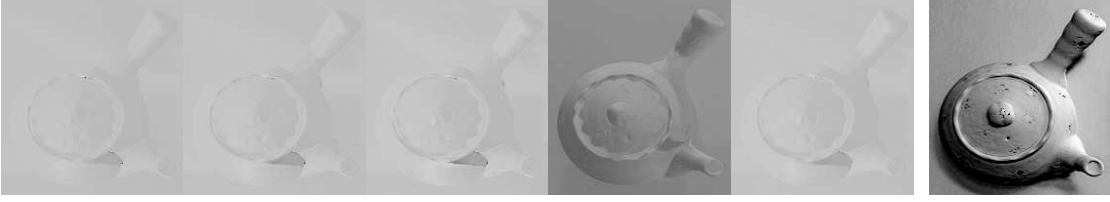


Fig. 4. Initial normal estimation from sparse vs. dense data for *Teapot*. From left to right: Using 5 images only, the 5 normal maps are respectively produced by using each image as the denominator image. The rightmost normal map is produced using a dense set of images. The normal map is displayed as $N \cdot L$ where $L = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})^T$ is the light direction.

is also unknown and the Lambertian model is often violated, the estimation of initial normal \tilde{N}_s would have been more complex and less stable if ρ_s were also considered in the estimation.

Let k be the total number of sampled images. To eliminate ρ , we divide $k - 1$ sampled images by a chosen image we call *denominator image* to obtain $k - 1$ ratio images. Without loss of generality let I_k be the denominator image. Each pixel in a ratio image is therefore expressed by

$$\frac{I_i}{I_k} = \frac{\tilde{N}_s \cdot L_i}{\tilde{N}_s \cdot L_k}. \quad (2)$$

An ideal denominator image is one that is minimally affected by shadows and highlight, which is difficult to obtain. By adopting the simple Lambertian model, we derive the denominator image to roughly eliminate the surface albedo by producing ratio images. The derivation is straightforward and is described in the footnote¹.

By using no less than three ratio images, we produce a local estimation of the normal at each pixel: define $\tilde{N}_s = [n_x \ n_y \ n_z]^T$, $L_i = [l_{i,x} \ l_{i,y} \ l_{i,z}]^T$ and $L_k = [l_{k,x} \ l_{k,y} \ l_{k,z}]^T$. For each pixel s in a ratio image i , rearranging (2) gives the following

$$A_{i,s}n_x + B_{i,s}n_y + C_{i,s}n_z = 0 \quad (3)$$

¹Our denominator image is derived by the following simple method:

1. We stack the sampled images to form a space-time volume $\{(x, y, t)\}$.
2. For each pixel location (x, y) , we sort all space-time pixels (x, y, t) in non-descending intensities along time t . The intensity rank of each pixel is thus known.
3. Since pixels with intensity adversely affected by shadows and specular highlight go to one of the two extremes of the sorted list, for each location (x, y) , if the intensity rank at (x, y, t) is higher than the median and smaller than some upper bound, it is highly probable that pixel (x, y) is free of shadows and highlight.

Thus, given a sampled image I_t , we count the number of pixels whose intensity rank satisfies $rank > R_L$ where $R_L \geq 50$ th percentile. Let $K_{R_L}^t$ be the total number of pixels satisfying this condition, $\bar{r}_{R_L}^t$ be the mean rank among the pixels that satisfy this condition. The denominator image is defined to be the one with 1) maximum K_{R_L} and 2) \bar{r}_{R_L} lower than some threshold R_H . Currently, we respectively set R_L and R_H to be the 70th and 90th percentiles in all our experiments.

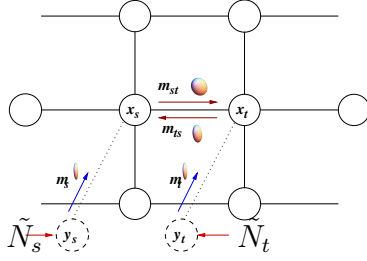


Fig. 5. The graph model of the Markov network. The observation nodes y_s and y_t use the initial normal estimates. In graph cuts, they are encoded into the data term in the energy minimization function. In tensor belief propagation, they are encoded as tensor messages m_s and m_t . A second-order symmetric tensor can be interpreted as a 3D ellipsoid. A stick tensor is an elongated ellipsoid, and hence the shapes of m_s and m_t shown above. Messages are updated and propagated during the iterative procedure, where the shapes of the tensor messages m_{st} and m_{ts} change progressively.

where

$$A_{i,s} = I_i l_{k,x} - I_k l_{i,x}, \quad B_{i,s} = I_i l_{k,y} - I_k l_{i,y}, \quad C_{i,s} = I_i l_{k,z} - I_k l_{i,z}$$

are constants. Given $k - 1 \geq 3$ ratio images, we have $k - 1$ such equations for each pixel. We can solve for $[n_x \ n_y \ n_z]^T$ by singular value decomposition (SVD) which explicitly enforces the unity constraint: $||\tilde{N}_s|| = 1$.

To demonstrate that the ratio image approach does not work for sparse input in the presence of shadows, highlight, and inaccurate estimation in light direction, we randomly pick five images from one of our dataset (*Teapot*) and use each of them in turn as the denominator image to estimate \tilde{N}_s at each pixel. As shown in Fig. 4, all five normal maps produced are unsatisfactory compared with the one produced by our dense input, because no image in the sparse subset is a good denominator image. The dense input provides adequate data redundancy to allow us to choose the best denominator image.

In practice, however, the best denominator image is not perfect because the input can be very noisy. Moreover, \tilde{N}_s estimated at each pixel does not take any advantage of neighborhood information. As we shall show, smoothing technique cannot be done because the underlying discontinuities will also be smoothed out. By using an explicit discontinuity-preserving function, in this paper, we propose to perform MRF refinement to infer the piecewise smooth normal field while preserving discontinuities. In the following sections, the estimated \tilde{N}_s is used to encode the data term for energy minimization using graph cuts (section V) and the local evidence for tensor belief propagation (section VI). Now, let us define the MRF model for dense photometric stereo.

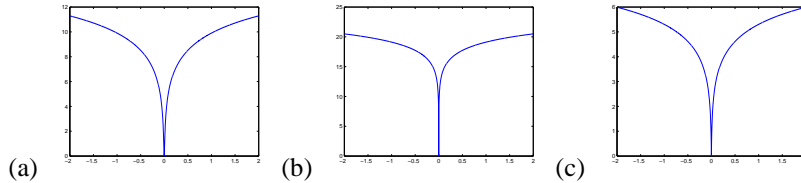


Fig. 6. The robust function for encoding the discontinuity-preserving function: plotting the Lorentzian function $\log(1 + \frac{1}{2}(\frac{x}{\sigma})^2)$ vs. x with (a) $\sigma = 0.005$, (b) $\sigma = 0.0005$, (c) Our modified Lorentzian function $\log(1 + \frac{1}{2}(\frac{|x|}{\sigma^2}))$ with $\sigma = 0.05$. In all cases, the curves are bounded when $x \rightarrow \pm\infty$, which is more robust than the usual norm-squared function (i.e. the unbounded x^2) in terms of encoding the error term.

The MRF model for dense photometric stereo Shown in Fig. 5 is a Markov network which is a graph with two types of nodes X and Y : A set of hidden variables $X = \{x_s\}$ and the set of observed variables $Y = \{y_s\}$. The posterior probability $P(X|Y)$ is defined by:

$$P(X|Y) \propto \prod_s \varphi_s(x_s, y_s) \prod_s \prod_{t \in \mathbf{N}(s)} \varphi_{st}(x_s, x_t) \quad (4)$$

where $\varphi_s(x_s, y_s)$ denotes the local evidence, and $\varphi_{st}(x_s, x_t)$ denotes the compatibility function. $\mathbf{N}(s)$ denotes the first-order neighborhood of node s .

To derive the MRF formulation for dense photometric stereo, we set $X = \mathcal{N}$ where \mathcal{N} is the set of normals visible to the camera (normal configuration) and $Y = \mathcal{I}$ where \mathcal{I} is the dense set of input images. We obtain

$$P(\mathcal{N}|\mathcal{I}) \propto \prod_s \exp\left(-\frac{\phi_s(N_s, \tilde{N}_s)}{2\sigma_1^2}\right) \prod_s \prod_{t \in \mathbf{N}(s)} \exp\left(-\frac{\phi_{st}(N_s, N_t)}{2\sigma_2^2}\right) \quad (5)$$

where N_s is the normal at node s , N_t is the normal at node t where (s, t) are neighboring nodes. The σ 's are used to control the the extent of the corresponding Gaussians. We define

$$\phi_s(N_s, \tilde{N}_s) = \|\tilde{N}_s - N_s\| \quad (6)$$

to measure the conformity N_s to the initial normal estimate \tilde{N}_s at location s .

We use a robust function, the Lorentzian function, to model ϕ_{st} :

$$R_f(x, \sigma) = \log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right) \quad (7)$$

where $x = \|N_s - N_t\|$ is a discontinuity-preserving metric [20]. Fig. 6 shows some plots of the Lorentzian function whose shape can be controlled by adjusting the σ parameter. ϕ_{st} is defined as

$$\phi_{st}(N_s, N_t) = \log\left(1 + \frac{1}{2}\left(\frac{\|N_s - N_t\|}{\sigma}\right)^2\right). \quad (8)$$

which penalizes the assignment of significantly different normal orientations.

V. ENERGY MINIMIZATION USING GRAPH CUTS

The graph cut algorithm is a widely adopted MRF technique in computer vision. Despite the desirable properties and the availability of a fast and simple implementation with a theoretical guarantee [20], there has been no previous work on the use of graph cuts to address the (dense) photometric stereo problem.

In this section, we formulate the problem of dense photometric stereo into one of graph cuts. Let $\mathcal{N} = \{\alpha_1, \alpha_2, \dots, \alpha_D\}$ be the pixelwise normal configuration of the scene, given a set of photometric images $\mathcal{I} = \{I_1, I_2, \dots, I_k\}$ each has a total of D pixels. Recall from (4) that the MRF model for photometric stereo for normal reconstruction is:

$$P(\mathcal{N}|\mathcal{I}) \propto \prod_s \varphi_s(N_s, \tilde{N}_s) \prod_s \prod_{t \in \mathbf{N}(s)} \phi_{st}(N_s, N_t) \quad (9)$$

If we take the logarithm of (9), we obtain

$$\begin{aligned} E(\mathcal{N}) &= \sum_s -\log \varphi_s(N_s, \tilde{N}_s) + \sum_{(s,t)} -\log \phi_{st}(N_s, N_t) \\ &= \sum_s D(N_s, \tilde{N}_s) + \sum_{(s,t)} V(N_s, N_t) \\ &= E_{data}(\mathcal{N}) + E_{smoothness}(\mathcal{N}). \end{aligned} \quad (10)$$

where the functions D and V are energy functions to be minimized by graph cuts. D and V are respectively called the data term and the smoothness term in graph cuts, which relate respectively to the local evidence and compatibility function of the corresponding MRF model.

In the realm of graph cuts, we seek an optimal normal configuration \mathcal{N}^* . Let \mathcal{L} be a set of labels corresponding to the set of all discrete normal orientations. The discrete labels correspond to the vertices on a subdivided icosahedron which guarantee uniform distribution on a sphere [2]. To increase precision, we follow [2] to subdivide each face of an icosahedron recursively in a total of 5 times (Fig. 3), so that $|\mathcal{L}| = 5057$. From our experimental results, it gives seamlessly smooth surface normals on a sphere.

A. Energy function

Our energy function for graph-cut minimization consists of the data and the smoothness terms. **Data term** Because our input consists of images and light directions only, our data term should measure the per-pixel difference between the measured and the estimated ratio images by using (3). However, this will produce a large number of summations in the data term due to plane fitting. As pixel intensity is significantly governed by the pixel's normal, we can instead measure

the difference between the initial normal \tilde{N}_s and the normal N_s at pixel s estimated in the current iteration during the graph-cut minimization (i.e., the current α -expansion [20]). Let \widehat{N}_w be the normal indexed by the label $w \in \mathcal{L}$. We define our data term as the following:

$$E_{data}(\mathcal{N}) = \sum_s D_s(\alpha_s) = \sum_s \|\tilde{N}_s - \widehat{N}_{\alpha_s}\|. \quad (11)$$

Smoothness term On the other hand, the smoothness term should measure the smoothness of the object surface while preserving the underlying discontinuity. To define the discontinuity-preserving smoothness term, we employ the *modified* Lorentzian function as the robust function (c.f. (7)):

$$\widehat{R}_f(x, \sigma) = \log\left(1 + \frac{1}{2}\left(\frac{|x|}{\sigma}\right)^2\right) \quad (12)$$

This function has a similar shape to the original Lorentzian function (Fig. 6). The modified Lorentzian function is necessary to make the energy function *regular* so that it can be graph-representable. The proof is given in the next section. We define our smoothness term as:

$$E_{smoothness}(\mathcal{N}) = \lambda \sum_{t \in \mathbf{N}(s)} V_{s,t}(\alpha_s, \alpha_t) \quad (13)$$

$$= \lambda \sum_{t \in \mathbf{N}(s)} \log\left(1 + \frac{\|\widehat{N}_{\alpha_s} - \widehat{N}_{\alpha_t}\|}{2\sigma^2}\right) \quad (14)$$

where $\lambda = \frac{\sigma_1^2}{\sigma_2^2}$ is a constant resulting from the logarithmic transformation in (10), and \mathbf{N} is the first-order neighborhood of s . The setting of λ depends on the scene and how much discontinuity to be preserved. For *Teapot*, $\lambda = 0.5$ and $\sigma = 0.4$.

B. Graph construction and proof of convergence

To perform multi-labeling minimization, the expansion move algorithm [20] is one suitable choice. Here, we have a quick review on this algorithm:

α -expansion For each iteration, we simply select a normal direction label $\alpha \in \mathcal{L}$, and then find the best configuration within this α -expansion move. If this configuration reduces the user-defined energy, the process is repeated. Otherwise, if there is no α that decreases the energy, we are done.

According to [20], the user-defined energy function has to be regular and thus graph representable so that it can be minimized via graph cuts (in a strong sense). This is also true for $|\mathcal{L}|$ -label configuration if α -expansion is employed. More precisely, for our $|\mathcal{L}|$ -label case, the

energy function has to be regular for each α displacement. In this connection, we will prove that our energy function E is regular in the following:

For any class \mathcal{F}^2 function of the form defined in [20]:

$$E(x_1, \dots, x_\delta) = \sum_i E^i(x_i) + \sum_{i < j} E^{i,j}(x_i, x_j) \quad (15)$$

where $\{x_i | i = 1, \dots, \delta\}$ and $x_i \in \{0, 1\}$ is a set of binary-valued variables. E is regular if and only if

$$E^{i,j}(0, 0) + E^{i,j}(1, 1) \leq E^{i,j}(0, 1) + E^{i,j}(1, 0). \quad (16)$$

From [20], it is known that any function of one variable is regular and hence the data term E_{data} is regular. Therefore, it remains to show that the smoothness term $E_{smoothness}$ satisfies (16) within a move. We prove the following claim on V which makes E regular. This claim also allows for the more efficient α -expansion which runs in $\Theta(|\mathcal{L}|)$ time [20].

Claim: $V_{s,t}$ is a metric.

The proof is as follows. In order that V is a metric, for any label $a_1, a_2, a_3 \in \mathcal{L}$, the following three conditions have to be satisfied:

$$\begin{aligned} V(a_1, a_2) = 0 &\Leftrightarrow a_1 = a_2 \\ V(a_1, a_2) &= V(a_2, a_1) \geq 0 \\ V(a_1, a_2) &\leq V(a_1, a_3) + V(a_3, a_2) \end{aligned}$$

Since the first two conditions are trivially true for our $E_{smoothness}$, we shall focus on the third condition here. Let $K_{ij} = \|\widehat{N}_{a_i} - \widehat{N}_{a_j}\|$. For any adjacent pair of pixels s and t , we write:

$$\begin{aligned} &V_{s,t}(a_1, a_3) + V_{s,t}(a_3, a_2) - V_{s,t}(a_1, a_2) \\ &= \log\left(1 + \frac{K_{13}}{2\sigma^2}\right) + \log\left(1 + \frac{K_{32}}{2\sigma^2}\right) - \log\left(1 + \frac{K_{12}}{2\sigma^2}\right) \\ &= \log\left(\frac{\left(1 + \frac{K_{13}}{2\sigma^2}\right)\left(1 + \frac{K_{32}}{2\sigma^2}\right)}{1 + \frac{K_{12}}{2\sigma^2}}\right) \end{aligned} \quad (17)$$

If the expression inside the logarithm of (17) is greater than or equal to 1, (17) is greater than or equals to zero. It is in fact true:

$$\begin{aligned} &\left(1 + \frac{K_{13}}{2\sigma^2}\right)\left(1 + \frac{K_{32}}{2\sigma^2}\right) - \left(1 + \frac{K_{12}}{2\sigma^2}\right) \\ &= \frac{1}{2\sigma^2}\left(K_{13} + K_{32} - K_{12} + \frac{K_{13}K_{32}}{2\sigma^2}\right) \geq 0 \end{aligned} \quad (18)$$

Note that $\widehat{N}_{a_1} - \widehat{N}_{a_3}$, $\widehat{N}_{a_3} - \widehat{N}_{a_2}$ and $\widehat{N}_{a_1} - \widehat{N}_{a_2}$ are three vectors projected onto the same plane defined by the points \widehat{N}_{a_1} , \widehat{N}_{a_2} and \widehat{N}_{a_3} , which form a triangle on the plane. By the triangle inequality, $K_{13} + K_{32} - K_{12}$ must not be less than zero, and hence the third metric condition holds.

Since $V_{s,t}$ is a metric, $V_{s,t}(\alpha, \alpha) = 0$ and $V_{s,t}(\alpha_s, \alpha_t) \leq V_{s,t}(\alpha_s, \alpha) + V_{s,t}(\alpha, \alpha_t)$, the smoothness term $E_{smoothness}$ is regular [20]. To minimize our energy function in each α displacement, we can construct a graph by using [20], followed by applying the max-flow algorithm [8].

VI. MAXIMUM A POSTERIORI ESTIMATION BY TENSOR BELIEF PROPAGATION

Although the graph-cut minimization described in the previous section for dense photometric stereo has a theoretical guarantee, in which the minimized energy corresponds to the global optimal solution “in a strong sense” [20], as we shall show in the comparison and result sections, the algorithm takes considerable amount of time (in minutes) to run due to the large number of α -expansions necessary for minimizing the energy function.

In this section, we study an alternative MRF inference algorithm to address the dense photometric stereo problem. In belief propagation, messages are propagated and combined in a Markov network. There are two common estimators for belief propagation: MAP and MMSE (minimum mean square error). In discrete labeling, the MAP estimator assigns discrete labels as messages, which are propagated and updated in each iteration. The max-product algorithm is often used in combining the propagated messages. MMSE estimator weighs marginal probabilities and produces an optimal solution at sub-pixel precision. MMSE uses sum-product to compute the marginal probabilities. Comparison with MAP and MMSE on geometric stereo were made in [39].

In photometric stereo, the traditional belief propagation is inefficient if discrete labels are used in encoding a message. Suppose we still subdivide an icosahedron to produce 5057 labels for each message, gigabytes of memory is required for a typical image (256×256). The memory required by sum-product and max-product are similar.

Inspired by tensor voting [26], we propose to apply *tensor belief propagation*, which uses a very compact representation for a message by encoding it into a compact *symmetric tensor* to store the second-order moment collection of the estimated normal directions. Note that the light source used in photometric stereo is located above the object, so the normals inferred should have a consistent orientation toward (or away from) the light source and hence the orientation is known in advance. Second-order moments are used in our message passing to simplify the

inference by making the tensor symmetric. We can simply flip the inferred normal after the estimation if needed.

In fact, tensor belief propagation is a special case of tensor voting where the spatial neighborhood is restricted into the first-order neighbors (given by the image grid structure). Although tensor belief propagation does not have a strong theoretical guarantee similar to graph-cut minimization, for all our experiments, we found that the normal maps produced by tensor belief propagation and graph cuts are comparable, while tensor belief propagation runs much faster (in a few seconds) than graph-cut minimization.

Since 3D normals are inferred, the tensor we use is a 3×3 symmetric matrix. Hence, the storage requirement for each message is reduced drastically to a hundred bytes or less. Using tensor as messages also changes our solution space from one of discrete to continuous.

Given a Markov network where $X = \{x_s\}$ is the set of hidden nodes and $Y = \{y_s\}$ is the set of observed nodes (Fig. 5), let $m_s(x_s)$ be the message received at node x_s from node y_s and $m_{st}(x_s, x_t)$ be the message that node x_s sends to node x_t . Initially, each pixel has an estimate of the normal direction \tilde{N}_s (Section IV). We represent $m_s(x_s)$ by the stick tensor of \tilde{N}_s , i.e. $\tilde{N}_s \tilde{N}_s^T$. The message passing algorithm is described below:

A. Algorithm

1. Initialize all messages $m_{st}(x_s, x_t)$ as a 3×3 identity matrix (i.e. a ball tensor without preferred orientation is used to denote uniform distribution) and $m_s(x_s) = \tilde{N}_s \tilde{N}_s^T$ (i.e. a stick tensor to indicate initial belief in the normal orientation for pixel s).

2. Update messages $m_{st}(x_s, x_t)$ iteratively for $i = 1 : T$ where T is the number of iterations:

2.1 Find the current normal with the highest probability

$$b_s^i(x_s) = m_s(x_s) + \sum_{x_k \in \mathbf{N}(x_s)} m_{ks}^i(x_k, x_s) \quad (19)$$

$$N_s^i = \hat{e}_1[b_s^i(x_s)] \quad (20)$$

where $\hat{e}_1[b_s(x_s)]$ is the unit eigenvector associated with the largest eigenvalue of the tensor $b_s(x_s)$.

2.2 Compute new messages

$$m_{st}^{i+1}(x_s, x_t) = \varphi_{st}(N_s^i, N_t^i) \left(\text{normalize} \left[m_s(x_s) + \sum_{x_k \in \mathbf{N}(x_s) \setminus x_t} m_{ks}^i(x_k, x_s) \right] \right) \quad (21)$$

where the normalization of a tensor scales all eigenvalues so that the largest one equals to 1. Notice that the compatibility function $\varphi_{st}(N_s^i, N_t^i)$ controls the strength of the message passed

to x_t . When the angle between N_s^i and N_t^i is small, $\phi_{st}(N_s, N_t)$ in (8) tends to 0 and hence $\varphi_{st}(N_s^i, N_t^i)$ tends to 1 and vice versa. Therefore, discontinuity between x_s and x_t can be preserved via controlling the strength of the messages passing between them. Furthermore, in the presence of discontinuity, the behavior of the compatibility function $\varphi_{st}(N_s^i, N_t^i)$ can be adjusted by the σ in (8), where $\sigma = 0.5$.

3. Compute beliefs

$$b_s(x_s) = m_s(x_s) + \sum_{x_k \in \mathbf{N}(x_s)} m_{ks}^T(x_k, x_s) \quad (22)$$

$$N_s = \hat{e}_1[b_s(x_s)] \quad (23)$$

In steps 2.1 and 3 we perform eigen-decomposition on b_s to obtain the majority direction, given by \hat{e}_1 , the eigenvector corresponding to the largest eigenvalue. It is similar to tensor voting [26] for inferring the most likely normal in surface reconstruction from a 3D point set. Fig. 5 illustrates the Markov network (graph) with messages passing in a neighborhood. The initial normal estimates \tilde{N}_s, \tilde{N}_t are passed to the hidden nodes, which will be encoded respectively into a stick tensor for representing $m_s(x_s)$ and $m_t(x_t)$ respectively. Messages are updated and passed among x 's accordingly.

The computational and storage complexities of our algorithm are $O(TD)$ and $O(D)$ respectively, where D is the number of pixels and T is the number of iterations. For an image of size 512×512 , it takes roughly 2 seconds only for each iteration on a Pentium-IV 3.2G PC with 512M memory.

It is worth noting that a method based on belief propagation was proposed in [33], which enforces surface integrability for surface reconstruction from normals. A Markov graph model was used where local evidence at each observation node is encoded by initial surface gradient estimated by any photometric stereo or shape-from-shading algorithms. Message passing is implemented by the sum-product algorithm which computes the MAP estimate of the unknown surface gradient at each pixel. Note that both [33] and our method use a graph model. Our tensor belief propagation directly estimates normals and explicitly preserves discontinuities via a robust function, while [33] refines the initial noisy surface gradients by enforcing the integrability (smoothness) constraint and does not use explicit discontinuity-preserving function.

B. Analysis

In our tensor message passing scheme, the tensors interact with each other when a new message is generated. Let us consider the following scenarios, using 2D tensor for illustration

because the 3D case is analogous. After summing up the tensor messages and performing eigen-decomposition, a 2D tensor has the form

$$\begin{bmatrix} \hat{e}_1 & \hat{e}_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \hat{e}_1^T \\ \hat{e}_2^T \end{bmatrix} \quad (24)$$

where λ_1, λ_2 ($\lambda_1 \geq \lambda_2 \geq 0$) are the eigenvalues and \hat{e}_1, \hat{e}_2 are the associated eigenvectors. Graphically, a 2D tensor can be represented as an ellipse with $\lambda_1 \hat{e}_1$ and $\lambda_2 \hat{e}_2$ corresponding to the oriented semimajor and semiminor axes respectively.

Note that a stick tensor is one with $\lambda_2 = 0$, which is used to encode absolute orientation certainty. A ball tensor is characterized by $\lambda_1 = \lambda_2$, and is used to encode absolute orientation uncertainty. Let us consider the following combinations when tensor messages in the extreme cases are added together:

1. *Both messages are stick tensors.* There are two scenarios: (a) When both \hat{e}_1 's in the two tensors are identical, the resulting tensor will have the same eigen-vector but a larger λ_1 , indicating that we have a higher confidence for \hat{e}_1 being the most likely normal direction. (b) Otherwise, the tensor becomes an ellipse, with the resulting \hat{e}_1 after eigen-decomposition still being the most likely direction, and with an uncertainty in direction, encoded as the orthogonal direction \hat{e}_2 with uncertainty λ_2 .

2. *One message is a stick tensor, the other message is a ball tensor.* This case is similar to scenario (b) of case (1).

3. *Both messages are ball tensors.* The resulting tensor is still a ball tensor because the two tensors do not have any preferred direction.

VII. SURFACE RECONSTRUCTION FROM NORMALS

The normals obtained after MRF refinement are used to recover the underlying surface. In this section, we propose a simple height generation algorithm which is empirically shown in the result section to produce an adequate surface given the normals recovered by our method.

Suppose the height at location s is h_s . The normal at s can be rewritten into:

$$N_s = \frac{1}{\sqrt{1 + p_s^2 + q_s^2}} [-p_s, -q_s, 1]^T \quad (25)$$

where $p_s = \frac{\partial h_s}{\partial x} = -\frac{n_x}{n_z}$ and $q_s = \frac{\partial h_s}{\partial y} = -\frac{n_y}{n_z}$. Many traditional approaches for surface reconstruction from normals are based on integration, and the integrability or the zero curl constraint needs to be enforced. Very often, enforcing integrability is translated into minimizing $\|\frac{\partial p}{\partial y} - \frac{\partial q}{\partial x}\|^2$

at each pixel [9]. Assume that all partial derivatives satisfy the integrability constraint, integration [9] can be applied to reconstruct the surface.

However, the normal maps obtained by using our method are not guaranteed (or needed) to be integrable everywhere because fine details and discontinuities are preserved in the map. To reconstruct the surface, one may apply [33] to alter the surface normals when necessary to satisfy the constraint. Another way to reconstruct the surface is to apply the shape from shapelet approach [21]. While a decent surface can be obtained by these methods, the methods are somewhat complicated. Here, we describe a simple method which is an analog of [4] and [10] to reconstruct a surface.

The idea of the method is here: the residual of the reconstructed surface at a pixel location should be minimized when all integration paths are considered. Given a first-order neighbor pair s and t , the residual of the height h_s with respect to h_t is defined by the difference between the estimated h_s and the height integrated starting from t :

$$\begin{cases} (h_s - h_t + p_s)^2, & \text{if } t = t1 \text{ is the left neighbor} \\ (h_s - h_t - p_t)^2, & \text{if } t = t2 \text{ is the right neighbor} \\ (h_s - h_t + q_s)^2, & \text{if } t = t3 \text{ is the up neighbor} \\ (h_s - h_t - q_t)^2, & \text{if } t = t4 \text{ is the bottom neighbor} \end{cases} \quad (26)$$

The total residual E of the reconstructed surface is defined by:

$$E(h) = \sum_s ((h_s - h_{t2} - p_{t2})^2 + (h_s - h_{t4} - q_{t4})^2) \quad (27)$$

Since each individual residual is a convex function, E is also a convex function. Any optimization method for convex optimization such as the gradient decent method can be used to minimize E to obtain h . In our implementation, E is minimized by setting its first derivative with respect to h_s equal to zero. Then h_s is solved iteratively. In each iteration, for each s , we estimate h_s by solving $\partial E(h)/\partial h_s = 0$ until the algorithm converges. All the surfaces in this paper are produced by this simple method, which are comparable to the results generated by [21] used in [38], [46].

VIII. COMPARISON

This section compares tensor belief propagation (TBP) and graph cuts (GC) using synthetic and real data where ground truths are available. Recall that both inference algorithms are based on the identical MRF model.

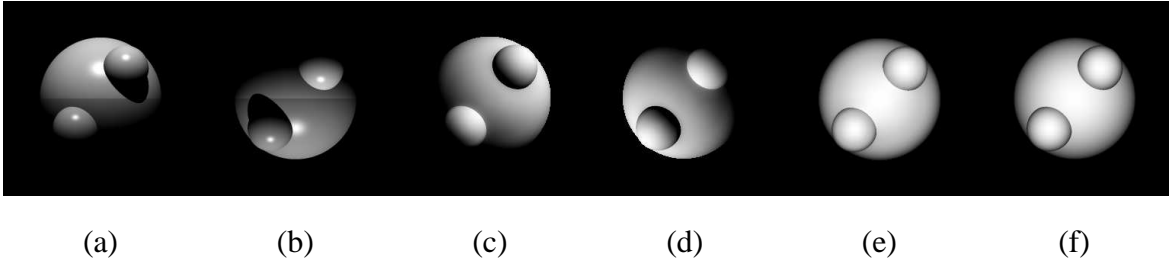


Fig. 7. *Three spheres*. (a)–(b) Two views of the input images. (c)–(e) Normals reconstructed by graph-cut minimization. (f) The ground truth. For ease of visualizing the recovered normals, in (c)–(f), we make the object Lambertian by displaying $N \cdot L$ for each pixel, where N is the recovered normal observed at a pixel, $L = [\frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}}]^T$ for (c), $L = [-\frac{1}{\sqrt{3}} -\frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}}]^T$ for (d) and $L = [0 \ 0 \ 1]^T$ for (e)–(f).

A. TBP vs. GC: synthetic data

We first use the synthetic example *Three spheres* where a total of 305 images are sampled. As shown in Fig. 7, the reflectance captured by the images contain a lot of specular highlight and shadows. The following is the evaluation procedure and the comparison results are summarized in Table I.

1. Obtain the ground truth normal map illuminated at $L = [0 \ 0 \ 1]^T$ (other L will render the $(N \cdot L)$ image too dark at certain pixels. See Fig. 7).
2. For various amount of additive Gaussian noises to the estimated light directions,
 - (a) Run tensor belief propagation to obtain the normal map illuminated at L .
 - (b) Run energy minimization by graph cuts to obtain the normal map illuminated at L .
 - (c) In both cases, note the running time, the number of iterations, and compute the $(N \cdot L)$ image as defined in the caption of Fig. 7.

According to Table I, the results and errors produced by graph cuts and tensor belief propagation are comparable while the running time of the graph cuts method is much longer. Note that both approaches can tolerate significant estimation error in lighting direction (up to a standard deviation (SD) of 15 degrees). In practice, such a large estimation error seldom occurs. Note that the smallest mean error that we can produce is about 4 degrees. This is because some of the surface patches are affected by shadow and specular highlight for most of the time (e.g., the surface patches along the silhouette of the largest sphere, whose normals are nearly perpendicular to the focal plane, are shadowed nearly half of the time). Nevertheless, the estimation accuracy in both algorithms are still very high.

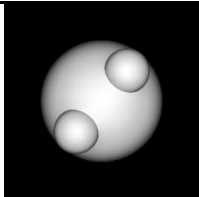
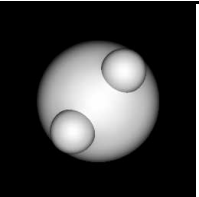
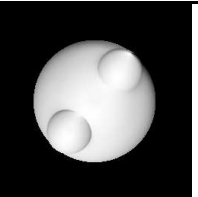
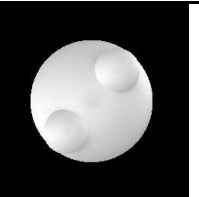
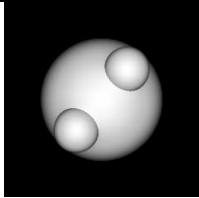
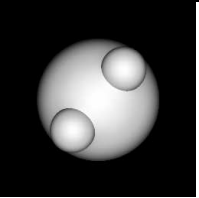
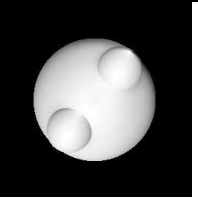
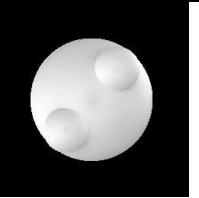
Standard deviation (SD)	0°	15°	30°	45°
				
TBP				
mean error (in deg)	4.0417	4.1008	21.9630	32.187
running time	23.49s	20.89s	33.89s	11.78s
no of iterations	105	89	141	54
				
GC				
mean error(in deg)	4.041	4.0905	22.0260	32.1950
running time	9m56s	9m52s	10m02s	9m58s
no of iterations	2	2	2	2

TABLE I

COMPARISON OF TBP AND GC ON *Three spheres*. THE EFFECT OF PERTURBATION OF LIGHT DIRECTIONS ON THE MEAN ERRORS OF THE RECOVERED NORMAL AND THE MAXIMUM PERTURBATION ANGLES ARE SHOWN.

THE GROUND TRUTH IS SHOWN IN FIG. 7(F). THE EXPERIMENTS WERE RUN ON A CPU SERVER WITH 4 AMD OPTERON(TM) PROCESSOR 844 CPU AT 1.8GHZ AND 16G DDR-333 RAM.

B. The effect of MRF refinement

In the presence of complex geometry, shadows, highlight and other non-Lambertian phenomena, MRF refinement is crucial to produce good normal results. Fig. 8 compares two normal maps and the resulting surfaces for *Teapot*: one is produced by the ratio image approach described in section IV, the other is produced, in addition, using our discontinuity-preserving MRF refinement (GC is used here). Note that the MRF refinement eliminates the errors caused by complex albedos while preserving all subtle geometry including the air hole and the ripple patterns of the teapot. Note that the $N \cdot L$ image depicted here is for display purpose, and existing 2D and 3D anisotropic diffusion or discontinuity-preserving methods cannot be applied to our normal map, where each 2D pixel location refers to a 3D normal.

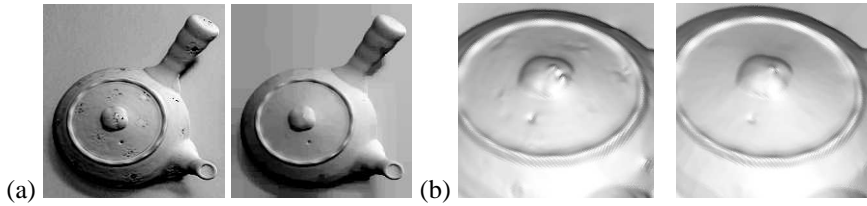


Fig. 8. The effect of MRF refinement for *Teapot*. (a) The noisy $N \cdot L$ image is produced by the least-square solution of the system of equations given by the ratio image approach described in section IV (i.e., without MRF refinement). The other image is produced by our MRF algorithm (GC). (b) Comparison of the generated surfaces from normals, without and with MRF refinement respectively.

Data set	Image size	Number of images	# TBP iterations	TBP running time	# GC iterations	GC running time
<i>Snail</i>	134×240	2074	98	25.02s	2	560s
<i>Cleopatra</i>	159×240	2517	65	82.30s	2	695s
<i>Teapot</i>	188×202	3165	304	175.47s	4	912s
<i>Rope</i>	171×144	2812	166	68.80s	3	614s
<i>Transparency</i>	212×209	3153	192	174.36s	3	820s
<i>Face</i>	223×235	1388	89	72.79s	4	986s

TABLE II

SUMMARY OF RUNNING TIMES. THE EXPERIMENTS WERE RUN ON A CPU SERVER WITH 4 OPTERON(TM) PROCESSORS 844 CPU AT 1.8GHZ AND 16G DDR-333 RAM.

IX. RESULTS

As mentioned in the previous section, both inference algorithms produce comparable results while belief propagation using tensor message passing runs much faster and converges to results comparable to those in GC. In all cases, very faithful normals can be recovered. We also show different views of the surfaces reconstructed using the recovered normals as input to our surface reconstruction algorithm presented in section VII. We have tested very complex objects and scenes containing a lot of highlight and shadows, and even objects with transparency to demonstrate the robustness of our method. For visualization, the normal N recovered at each pixel is displayed using $(N \cdot L)$ where L is the direction of a synthetic light, which allows for easy detection by the human eye if any slight estimation error is present. Table II summarizes the running times. Please also review our supplementary video for our results.

Comparison with ground truth: real data In Fig. 9, an example *Real Sphere* is shown. We chose a spherical object because we can estimate the ground truth normal map of the object by fitting a known sphere. Without considering the distortion resulting by perspective projection of the camera and inaccurate estimations of light directions, the absolute mean angular error

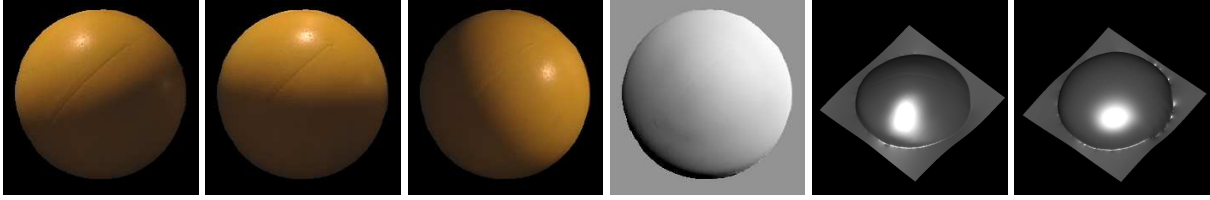


Fig. 9. Results on *Real Sphere*. From left to right: Three typical captured images of *Real Sphere*. The recovered normal N displayed as $N \cdot L$ with $L = [\frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}}]^T$. The reconstructed surface rendered at a novel viewpoint. The reconstructed surface from ground truth normals.

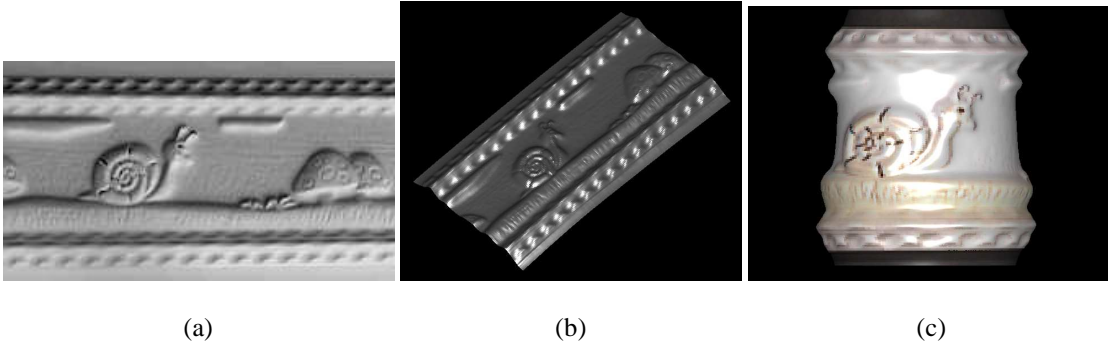


Fig. 10. Results on *Snail*. Three typical images we captured were shown in Fig. 1. (a) The reconstructed normals N are shown as $N \cdot L$ where $L = [\frac{1}{\sqrt{3}} -\frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}}]^T$. (b) The surface reconstructed from the recovered normals. (c) The result of displacement mapping on a cylinder, using the reconstructed surface.

produced in this case is 19.36 degrees. Note that such a large absolute error for real case is due to the presence of non-negligible ambient light and violation of Lambertian assumption. Ideally, because the spherical object is opaque, half of the spherical object in the input images shown in Fig. 9 should be totally invisible. But the strong ambient light makes it visible which offsets the normal estimation. Because of this, the estimated normal tends to point upward resulting in a large mean error. Despite that, our method preserves the overall structure very well. The result looks visually good, and is indeed quantitatively faithful to the original surface if measured on an alternative metric: we define E_r to measure the structural difference between the estimation and the ground truth:

$$E_r = \frac{1}{\mathcal{M}_r} \sum_{a=(s,t)} |\theta_a \psi_{a+1} - \theta_{a+1} \psi_a| \quad (28)$$

where \mathcal{M}_r is the total number of pixel pairs, θ_a is the angle between the neighboring normals at s and t in the estimated normal map and ψ_a is the angle between the neighboring normals at s and t in the ground truth normal map. If (s, t) is a left-and-right neighbor pair, $a + 1 = (t, u)$ where u is the right neighbor of t . The notation is applied similarly to up-and-down neighboring pairs. Thus Eqn. 28 measures the mean of the angular ratio difference in local neighborhood,

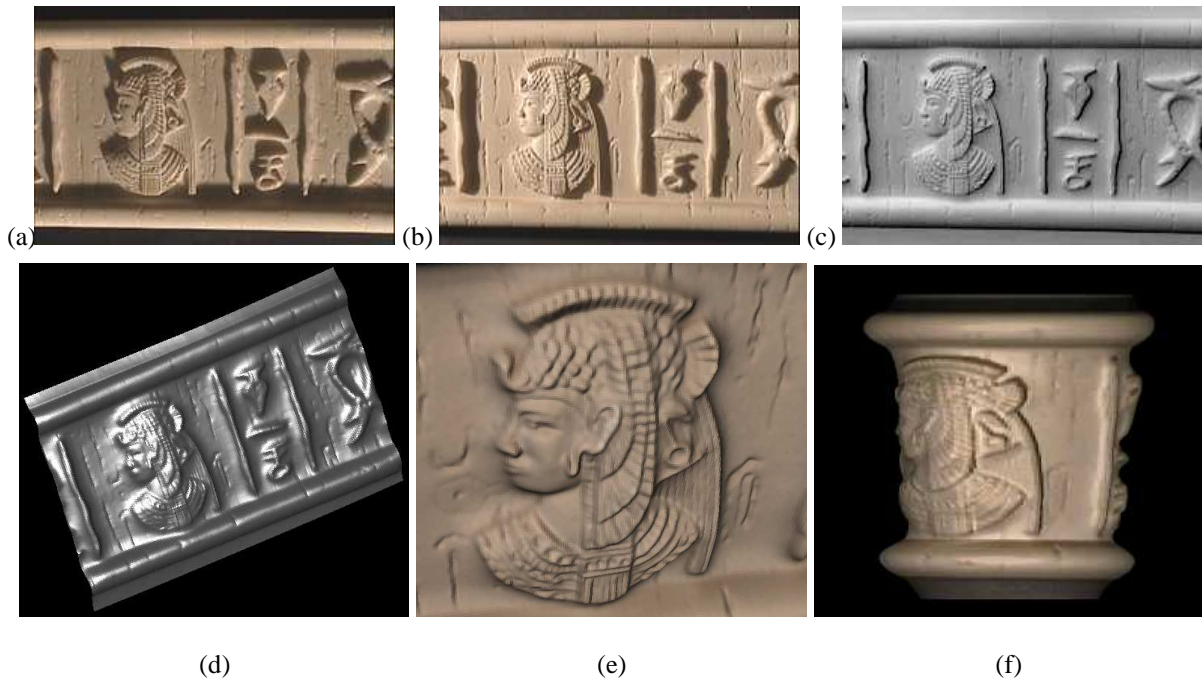


Fig. 11. Results on *Cleopatra*. (a)–(b) Two typical images we captured. (c) The reconstructed normals visualized as $N \cdot L$ where $L = [-\frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}}]^T$. (d) The reconstructed surface visualized at a novel viewpoint. (e) The zoom-in view of the textured surface. (f) The result of displacement mapping on a synthetic cylinder, using the reconstructed surface.

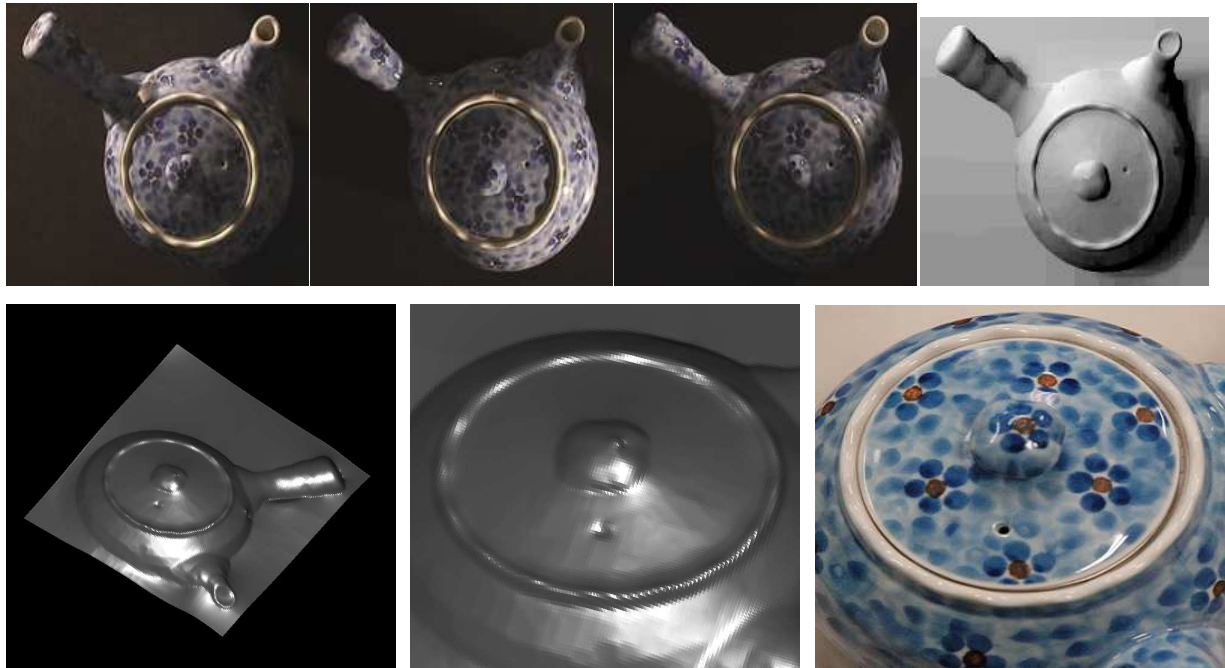


Fig. 12. Top: three typical captured images of *Teapot* where complex geometry, texture and severe shadows are present, and the recovered normals N , each of them is displayed as $N \cdot L$ with $L = [-\frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}}]^T$. Bottom: The reconstructed surface rendered at a novel viewpoint, the zoom-in view of the reconstructed surface, and the zoom-in view of the actual object at the same viewpoint.

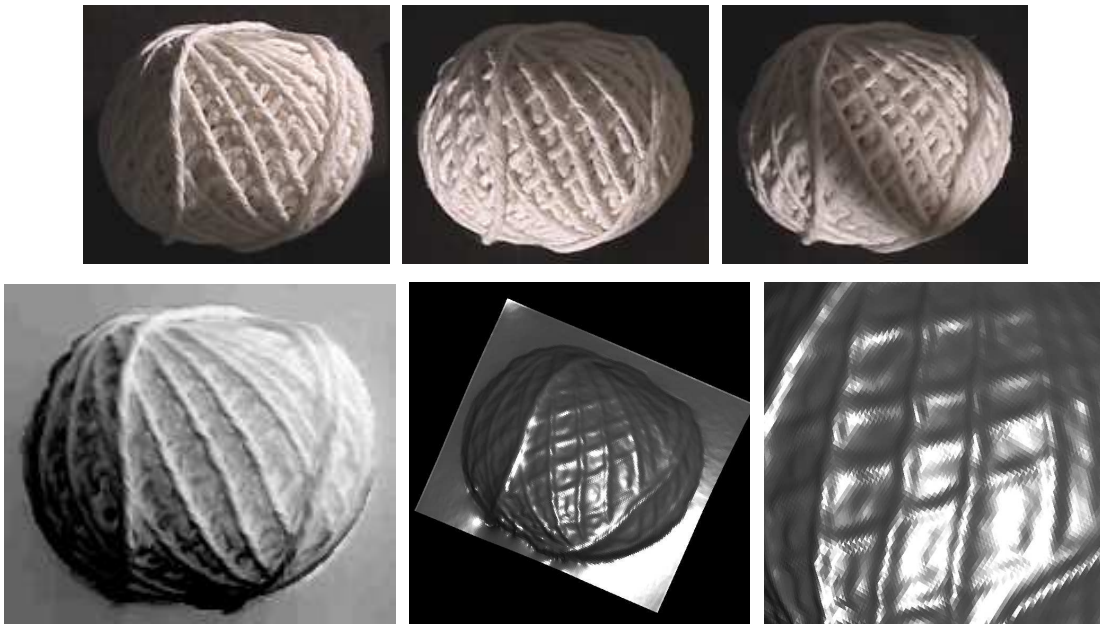


Fig. 13. Top: Three typical captured images of *Rope* where complex geometry, mesostructures, textures and severe shadows are present. Bottom, from left to right: the recovered normals N , each of them is displayed as $N \cdot L$ with $L = [\frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}}]^T$. The reconstructed surface rendered at a novel viewpoint, and the zoom-in view of the reconstructed surface.



Fig. 14. Top: three typical images for *Transparency*, where many assumptions in photometric stereo are violated: shadows, highlight, transparency, spatially-varying albedos, and inter-reflections due to the complex geometry. Bottom, from left to right: The recovered normals N are very reasonable, displayed as $N \cdot L$ where $L = [\frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}} \frac{1}{\sqrt{3}}]^T$, the reconstructed surface rendered at a novel viewpoint, and the zoom-in view of the textured surface.

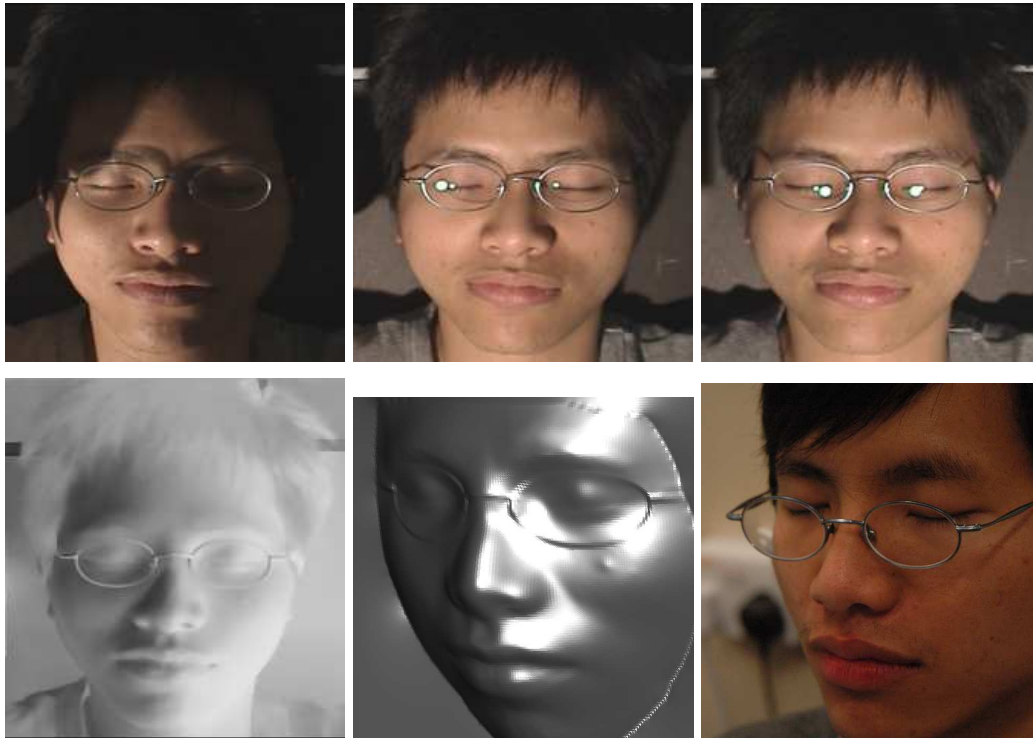


Fig. 15. Top: three typical captured images of *Face* where complex geometry, texture and severe shadows are present, and the recovered normals N , each of them is displayed as $N \cdot L$ with $L = [\frac{1}{\sqrt{3}} \quad -\frac{1}{\sqrt{3}} \quad \frac{1}{\sqrt{3}}]^T$. Bottom: Reconstructed surface rendered at a novel viewpoint, and one view of the actual face captured at similar viewpoint.

and hence the neighborhood structure of the normal map. We found that, although the absolute mean error is large, E_r tends to zero (8.55×10^{-5}), indicating that our method preserves the neighborhood structure very well and the mean error is defined up to a known angular scale factor. By brute-force searching, we found the optimal scale that gives the minimum mean error of 3.87 degrees, instead of the absolute mean error of 19.36 degrees.

Complex patterns with discontinuities In Fig. 10, note the high level of details achieved in our reconstruction, where the cloud, snail and mushroom and other complex patterns are faithfully preserved in the normal reconstruction despite that cast shadows and highlight are ubiquitous. The smooth surface and the underlying surface orientation discontinuities are faithfully restored. We also show the result of displacement texture mapping on a synthetic cylinder by using our reconstructed surface and normals for this object. Fig. 11 shows another result in this category.

Objects with complex geometry and albedos We show the reconstruction results for two complex objects *Teapot* and *Rope* in Figs 12 and 13. The geometry and albedos of the *Teapot* are very complex. Our method can faithfully reconstruct the normal directions and shape of the

teapot including the small air hole on the lid, while rejecting all noises caused by the complex patterns, textures and colors of the teapot. Although the *Rope* has spatially-varying surface mesostructures, the surface and normals are faithfully reconstructed.

Complex objects with transparency Finally, the example in Fig. 14 tests our system to the limit. The toy is contained inside an open paper box, which casts a lot of shadows when the object is illuminated on the three sides of the box. The toy is wrapped inside a transparent plastic container. So when it is illuminated at other directions, a lot of highlight is produced. Surface orientation discontinuities are ubiquitous in the object. It is very tedious to choose the right frames from more than 3000 frames we captured to perform sparse photometric stereo and unbiased statistics is not guaranteed. On the other hand, our simple system which utilizes dense but noisy measurement can effectively deal with these problems. The surface normals we recovered are very reasonable under this complex situation.

Fig. 15 shows another complex example of a *Face* where complex geometry, fine structures (hairs and pimples on the face) and transparency (eye glasses) are present. Observe that the eyes below the glasses are successfully reconstructed. The discontinuity associated with the frames of the glasses is still maintained. The pimple on the subject's face close to his eye glasses has been preserved in the surface reconstruction.

X. THE INVERSE PROBLEM: APPLICATION IN REAL-TIME RELIGHTING

Using the reconstructed surface from dense photometric stereo, we propose an inverse process to synthesize novel images of the same scene under user-specified light source and direction. This inverse process can be made to run in real time by employing current hardware technology, and is better known as real-time relighting in computer graphics.

The ability in controlling illumination offers the user an experience of 3D realism. Recent work [41] allows the captured video (dynamic sense) be composited with an new environment seamlessly. Environment lighting is considered and varying lighting conditions is allowed. However, their method required an expensive and specially designed acquisition system. And, real-time relighting may not be possible. Using our very simple set-up and photometric stereo method, we address the inverse problem of real-time rendering by using compressed data.

Image-based relighting [43], [30] is a method to achieve real-time illumination computation of arbitrarily complex scenes. It shifts the data acquisition (for real scene) or the time-consuming illumination computation (for synthetic scenes) to a preprocessing stage and stores the results in a compact form. During the run-time, illumination effects are achieved by real-time decom-

pression and composition. However, if per-pixel depth information is not available, photorealistic relighting with interesting lighting effects such as illumination due to spotlight, point light source, or slide projection cannot be correctly achieved [43]. On the other hand, the necessary data for relighting basically consists of a dense image set captured under a moving distant light source, which is exactly what our dense photometric stereo need for normal reconstruction. The reconstructed surface is exactly what relighting need for producing versatile lighting effects.

The early work in image-based relighting has much restriction on the novel lighting configuration [11], [30]. The first representation, *the apparent BRDF*, that supports arbitrarily novel lighting configuration is proposed by Wong et al. [43]. The representation is further generalized to *plenoptic illumination function* [42]. Per-pixel spherical harmonics are used as a compact solution for encoding the enormous relighting data in their work. Polynomial function [25], wavelet [29], spherical radial basis function [23] were later proposed by other researchers as the compact solutions. Unlike the rendering goal in computer graphics, researchers in computer vision are more interested in recognition under various lighting conditions. Principal component analysis is frequently used to extract a set of basis images from dense input images for recognition purposes [5], [47], [27], [12]. These methods can be adapted for rendering purposes as demonstrated by the recent work [31], [40], [14].

In this paper, we propose a hybrid, image-and-geometry-based approach which makes use of the dense input (images) and the surface/depth map reconstructed from the recovered normals (geometry) to perform real-time relighting for achieving visually plausible results. Without the recovered depth map from our dense photometric stereo, we can only perform restrictive relighting with distant light sources. This hybrid approach not only provides a unified way to simulate distant and point light sources, but also achieves very fast frame rate by employing the state-of-the-art graphics processing unit (GPU). In order to cope with the limited memory resource on GPU, we adopt the PCA-based representation [14] as a compact solution for rendering.

In the following we start by briefly reviewing the plenoptic illumination function, which is sampled and obtained by processing the raw dense input images captured by the DV for the purpose of real-time relighting. Note that the dense input images are used for *both* the recovery of depth map (using dense photometric stereo) and the radiance data encoding (using PCA-based approach). As the dense photometric stereo has been covered in the previous sections, we shall focus on the PCA-based encoding of radiance data in the following subsections. During the rendering, both the recovered surface and PCA-encoded radiance data are loaded into GPU memory. A unified GPU approach for real-time relighting that supports both distant light source,

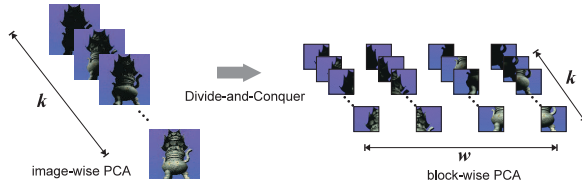


Fig. 16. A divide-and-conquer approach is used to make the computation tractable and facilitate parallelism.

point light source, and even the slide projection, is described. Finally, we present our relighting results using a wide range of synthetic lighting set-up.

A. The Plenoptic Illumination Function

Image-based relighting is grounded on the *plenoptic illumination function*, which is extended from the plenoptic function [1] to include the illumination component [42]:

$$I = P_I(l_x, l_y, l_z, v_x, v_y, v_z, x, y, z, t, \lambda), \quad (29)$$

The function describes the radiance I received along any viewing direction (v_x, v_y, v_z) observed at any viewpoint (x, y, z) in space, at any time t and over any range of wavelength λ . $L = (l_x, l_y, l_z)$ specifies the direction of a distant light source illuminating the scene, and t is the time parameter. This function encodes how the environment looks like when the viewpoint is positioned at (x, y, z) under illumination L . When the viewpoint and time parameters are fixed, the discrete version of the plenoptic illumination function reduces to the dense input for our dense photometric stereo. To relight an image, we apply the following at all pixels on the three color channels respectively:

$$P_I^*(l_x, l_y, l_z)L_r(x, y, z, l_x, l_y, l_z), \quad (30)$$

where $P_I^*(l_x, l_y, l_z)$ is the result of interpolating the dense samples given the desired light vector (l_x, l_y, l_z) (other parameters in P_I^* are dropped for simplicity), L_r is the radiance along (l_x, l_y, l_z) due to the light source, and (x, y, z) is the position where radiance is reflected. This is a local illumination model with three parameters: the direction, the color, and the number of light sources.

B. Unified GPU approach for real-time relighting

The major issue in image-based relighting also present in our hybrid approach is the enormous storage requirement. Note that traditional image compression methods such as JPEG is

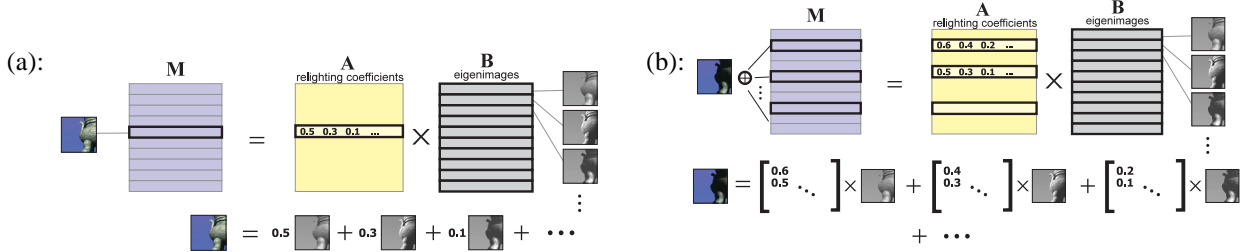


Fig. 17. Our unified approach for (a) distant-source and (b) point-source relighting, which are translated into per-pixel table-lookup and multiplication, highly suitable for hardware implementation. (a) To reconstruct (relight) an image block under directional illumination, each data vector (row) of \mathbf{M} is a linear combination of rows (eigenimages) in \mathbf{B} . (b) Under the illumination of a point light source for which spotlight and slide projector are specific cases, relighting coefficients are sampled from multiple rows because the light direction L at each pixel is different. L is obtained from the depth map inferred from the normals reconstructed using our robust photometric stereo.

not applicable due to their lack of random-accessibility. In order to achieve real-time and complex lighting effects, the relighting engine should be capable of randomly accessing pixel values scattered all over the compressed data. In this section, we first review our illumination-adjustable representation [14] that facilitates the implementation on GPU for encoding a plenoptic illumination function [42] in order to achieve photorealistic relighting at high frame rate. A *unified* GPU computation framework is then proposed for both distant, point (or spotlight) and slide projector sources, which is made possible by our dense photometric stereo technique.

Principal component analysis (PCA) First, we preprocess the input data in order to maximize the correlation among neighboring data. Recall that our input is a set of images taken at the same viewpoint, but illuminated by a distant light source along different directions. Each captured image corresponds to a point on the light direction sphere. After resampling, a total of k images is obtained. We observe that the luminance of the k corresponding pixel values are highly correlated, due to the smooth change in radiance reflected from the same surface element visible at a pixel. Therefore, principal component analysis can be applied to reduce the data dimensionality. First, each 2D image is linearized to an 1D array of pixel values, which we call data vectors. Then, all data vectors are stacked to form a data matrix \mathbf{M} . The size of this matrix is prohibitively large. For example, for a greyscale image of 256×256 sampled under k lighting conditions, the data matrix is of size $k \times 65535$. It is not feasible to compute the principal components from this huge matrix.

A divide-and-conquer approach is therefore adopted to subdivide the images into blocks. Multiple blockwise PCAs are applied on the corresponding blocks. If each image is subdivided into w blocks, we perform w blockwise PCAs (see Fig. 16). With this block-based approach,

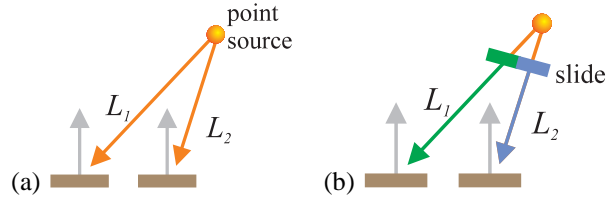


Fig. 18. (a) Point-Source relighting (b) slide projector source. L is different for each pixel under point-source illumination

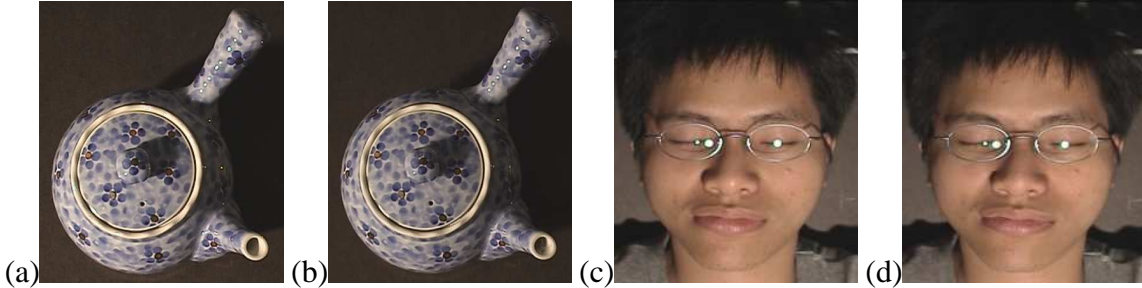


Fig. 19. Relighting results for *Teapot* and *Face* using a synthetic distant light source to simulate the actual illuminant used in data capturing. (a) and (c) are real, (b) and (d) are our synthetic relighting results. They are visually indistinguishable.

the computation becomes tractable and the memory requirement is also reduced. Moreover, the computation can be parallelized easily. The block-wise PCA also helps in capturing high-frequency features, like highlight and shadows, with fewer number of principal components. Interestingly, while shadows and highlight are treated as noises in photometric stereo reconstruction, they are important cue for photorealism during relighting. In our relighting system, we choose a block size of 16×16 .

By applying PCA to the data matrix \mathbf{M} , \mathbf{M} can be well approximated by \mathcal{M} basis images and their corresponding coefficients, where $\mathcal{M} \ll k$. The data volume is drastically reduced by keeping only \mathcal{M} eigenimages and the relighting coefficients. Now, \mathbf{M} can be expressed by the product of two matrices \mathbf{A} and \mathbf{B} , where the dimension of \mathbf{A} is $k \times \mathcal{M}$ and \mathbf{B} is $\mathcal{M} \times q$, where q is the block size.

Distant-source Relighting Every row of \mathbf{M} (image block) is a linear combination of all the rows in \mathbf{B} . The corresponding weights are kept in a row in \mathbf{A} (see Fig. 17(a)). We call the rows in \mathbf{B} the basis images or *eigenimages* and the weights in \mathbf{A} the *relighting coefficients*. The distant-source relighting can be expressed compactly by

$$I(L) = \sum_j c_j B_j \quad (31)$$

where I is the image block relit under distant illumination L , B_j is the j -th eigenimage, and c_j

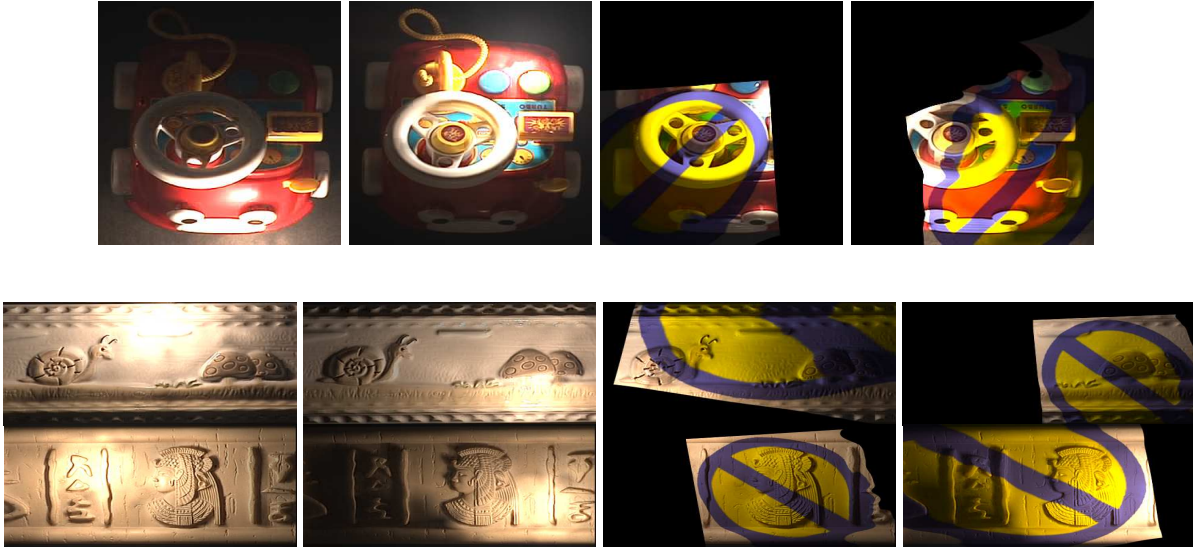


Fig. 20. Relighting results for *Car*, *Snail*, and *Cleopatra* using a synthetic point and slide projection sources.

is the j -th relighting coefficient in one row of \mathbf{A} , which is indexed by L . In case the desired L is not sampled, interpolation using the closest neighbors will be performed to reconstruct the desired I . Given a distant source with direction L , each pixel is relit with the same light vector L . Hence, the relighting coefficients c_j are the same for all pixels in an image block. In other words, distant-source relighting is actually the linear combination of eigenimage blocks B_j with c_j as weights. Such linear combination can be performed in real-time on modern GPU by storing eigenimages and relighting coefficients as textures.

Point-Source Relighting Despite the use of a directional illuminant (simulated by a distant spotlight) during the capturing phase, the captured data can be employed to simulate the illumination due to a point source, spotlight and slide projector source. Unlike the distant-source relighting, the light direction L observed at each pixel is different (as explained in Fig. 18). To obtain $L = S - S_p$ at a surface point S_p , where S is the given position of the point light source. S_p can be derived from the reconstructed surface or the depth map of the scene. Thus, relighting using a point source is now readily achieved, which is otherwise impossible because the surface geometry is either unavailable or difficult to obtain in complex situation using standard techniques.

Now, because L is different for each pixel, the relighting coefficient for each pixel is therefore also different. To relight an image block, relighting coefficients are sampled from *different* rows of \mathbf{A} , again indexed by different L for each pixel in B_j :

$$I(\mathbf{L}) = \sum_j A_j(\mathbf{L})B_j \quad (32)$$

where each $A_j(\mathbf{L})$ is a 2D map of relighting coefficients, whose dimension is the same as that of an eigenimage block B_j . \mathbf{L} is used to indicate the set of light direction vectors at all pixels in I . In other words, the point-source relighting is actually a *pixel-wise* linear combination of two images $A_j(\mathbf{L})$ and B_j . (32) represents the per-pixel table-lookup and multiplication to relight an image under point source illumination, as illustrated in Fig. 17(b). Again such per-pixel operations can be efficiently implemented on modern GPU.

Results Our GPU relighting was implemented and run on a 3.2GHz PC with 512M memory, and GeForce FX5900 graphics board with 256MB video memory. Fig. 19 shows the results for relighting *Teapot* and *Face* using a synthetic distant light source which is made to simulate the actual handheld illuminant used in data capturing to test the correctness of our implementation of the distant-source relighting. Fig. 20 show the results for relighting *Car*, *Snail*, and *Cleopatra* using a synthetic point and a slide projector source respectively, where we can achieve a very high frame rate of 75 for point-source relighting, and 37 for slide-projector-source relighting, which are both suitable for time-critical applications like computer games. Note the relighting results of the slide projector source that uses a circular stop sign, whose projection on the object changes according to the surface geometry of the object, which is derived from the surface reconstructed from our recovered normal map. Please refer to our supplementary video.

XI. CONCLUSION

In this paper we formulate the problem of dense photometric stereo using the MRF framework. Using the identical MRF model, we propose and compare two inference algorithms for estimating the MAP solution: graph cuts and tensor belief propagation. For high-precision message passing in our dense photometric stereo problem, traditional belief propagation is intractable if the set of discrete labels is large, while the graph cut algorithm converges in very few iterations. Tensor message passing for belief propagation is proposed which drastically reduces the running time and storage requirement, and it runs faster than graph cuts with comparable results. Faithful per-pixel normal maps are inferred by both algorithms. Finally, we exploit the inferred normals and the reconstructed surface to perform real-time relighting where distant, point, spotlight and slide projector light sources can be uniformly handled and very fast frame rate can be achieved. Our future work consists of more investigation on the surface reconstruction algorithm and analysis of the efficacy of the available dense information.

Acknowledgments

We thank the Associate Editor and all reviewers for their thorough and constructive suggestions which are instrumental in improving the quality of the final paper. This research was supported by the Research Grant Council of the Hong Kong Special Administrative Region under grant number AoE/E-01/99 and RGC Earmarked Grant reference number 417005.

REFERENCES

- [1] E.H. Adelson and J.R. Bergen. The plenoptic function and the elements of early vision. In M.S. Landy and J.A. Movshon, editors, *Computational Models of Visual Processing*, chapter 1, pages 3–20. MIT Press, 1991.
- [2] D.H. Ballard and C.M. Brown. *Computer vision*. In *Prentice Hall*, 1982.
- [3] S. Barsky and M. Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. *PAMI*, 25(10):1239–1252, October 2003.
- [4] R. Basri and D.W. Jacobs. Photometric stereo with general, unknown lighting. In *CVPR01*, pages II:374–381, 2001.
- [5] P. N. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible lighting conditions? In *CVPR 1996*, pages 270–277, 1996.
- [6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, November 2001.
- [7] E.N. Coleman, Jr. and R. Jain. Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. *CGIP*, 18(4):309–328, April 1982.
- [8] T.H. Cormen, C.E. Lieserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
- [9] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice-Hall, 2003.
- [10] D.B. Goldman, B. Curless, A. Hertzmann, and S.M. Seitz. Shape and spatially-varying brdfs from photometric stereo. In *ICCV05*, October 2005.
- [11] P. Haeberli. Synthetic lighting for photography. online. <http://www.sgi.com/grafica/synth/index.html>, January 1992.
- [12] G. Hager and P. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1996.
- [13] A. Hertzmann and S.M. Seitz. Shape and materials by example: a photometric stereo approach. In *CVPR03*, pages I: 533–540, 2003.
- [14] P.-M. Ho, T.-T. Wong, and C.-S. Leung. Compressing the illumination-adjustable images with principal component analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(3):355–364, March 2005.
- [15] B.K.P. Horn. *Robot Vision*. McGraw-Hill, 1986.
- [16] B.K.P. Horn, R.J. Woodham, and W.M. Silver. Determining shape and reflectance using multiple images. In *MIT AI Memo*, 1978.
- [17] Y. Ju, K. Lee, and S.U. Lee. Shape from shading using graph cuts. In *ICIP03*, pages I: 421–424, 2003.
- [18] G. Kay and T. Caelly. Estimating the parameters of an illumination model using photometric stereo. *GMIP*, 57(5):365–388, 1995.
- [19] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV02*, page III: 82 ff., 2002.
- [20] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, Feb 2004.
- [21] P. Kovesi. Shapelets correlated with surface normals produce surfaces. In *ICCV05*, pages 994–1001, 2005.
- [22] K.M. Lee and C.C.J. Kuo. Shape reconstruction from photometric stereo. In *CVPR92*, pages 479–484, 1992.
- [23] C.-S. Leung, T.-T. Wong, P.-M.Lam, and K.-H. Choy. An RBF-based image compression method for image-based rendering. *IEEE Transactions on Image Processing*. to appear.
- [24] Y. Li, H.Y. Shum, C.K. Tang, and R. Szeliski. Stereo reconstruction from multiperspective panoramas. *PAMI*, 26(1):45–62, January 2004.

- [25] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *Proceedings of ACM SIGGRAPH 2001*, pages 519–528, 2001.
- [26] G. Medioni, M.S. Lee, and C.K. Tang. *A Computational Framework for Feature Extraction and Segmentation*. Elsevier Science, Amsterdam, 2000.
- [27] S. K. Nayar and H. Murase. Dimensionality of illumination in appearance matching. In *IEEE International Conference on Robotics and Automation*, pages 1326–1332, April 1996.
- [28] S.K. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Trans. on Robotics and Automation*, 6(4):418–431, 1990.
- [29] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics*, 22(3):376–381, 2003.
- [30] J. S. Nimeroff, E. Simoncelli, and J. Dorsey. Efficient re-rendering of naturally illuminated environments. In *Fifth Eurographics Workshop on Rendering*, pages 359–373, June 1994.
- [31] K. Nishino, Y. Sato, and K. Ikeuchi. Eigen-texture method: Appearance compression based on 3D model. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 618–624, June 1999.
- [32] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [33] N. Petrovic, I. Cohen, B.J. Frey, R. Koetter, and T.S. Huang. Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models. In *CVPR01*, pages I:743–748, 2001.
- [34] Y. Shan, Z. Liu, and Z. Zhang. Image-based surface detail transfer. In *CVPR01*, volume II, pages 794–799, 2001.
- [35] F. Solomon and K. Ikeuchi. Extracting the shape and roughness of specular lobe objects using four light photometric stereo. *PAMI*, 18(4):449–454, April 1996.
- [36] J. Sun, N.N. Zheng, and H.Y. Shum. Stereo matching using belief propagation. *PAMI*, 25(7):787–800, July 2003.
- [37] H.D. Tagare and R.J.P. deFigueiredo. A theory of photometric stereo for a class of diffuse non-lambertian surfaces. *PAMI*, 13(2):133–152, February 1991.
- [38] K.L. Tang, C.K. Tang, and T.T. Wong. Dense photometric stereo using tensorial belief propagation. In *CVPR2005*, volume 1, pages 132–139, June 2005.
- [39] M.F. Tappen and W.T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *ICCV03*, pages 900–907, 2003.
- [40] M. Alex O. Vasilescu and Demetri Terzopoulos. Tensor textures: multilinear image-based rendering. *ACM Trans. Graph.*, 23(3):336–342, 2004.
- [41] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. E. Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Trans. Graph.*, 24(3):756–764, 2005.
- [42] T.-T. Wong, C.-W. Fu, P.-A. Heng, and C.-S. Leung. The plenoptic illumination function. *IEEE Transactions on Multimedia*, 4(3), September 2002.
- [43] T.-T. Wong, P.-A. Heng, S.-H. Or, and W.-Y. Ng. Image-based rendering with controllable illumination. In *Proceedings of the 8-th Eurographics Workshop on Rendering (Rendering Techniques '97)*, pages 13–22, St. Etienne, France, June 1997.
- [44] R.J. Woodham. Photometric method for determining surface orientation from multiple images. *OptEng*, 19(1):139–144, January 1980.
- [45] R.J. Woodham. Gradient and curvature from the photometric-stereo method, including local confidence estimation. *JOSA-A*, 11(11):3050–3068, November 1994.
- [46] T.P. Wu and C.K. Tang. Dense photometric stereo using a mirror sphere and graph cut. In *CVPR2005*, volume 1, pages 140–147, June 2005.
- [47] Z. Zhang. Modeling geometric structure and illumination variation of a scene from real images. In *ICCV'98*, Bombay, India, January 1998.