

SURFACE CLASSIFICATION VIA BRDF PARAMETERS, USING POPULATION MONTE CARLO FOR MRF PARAMETER ESTIMATION

Markus Louw Fred Nicolls
 Department of Electrical Engineering
 University of Cape Town
 South Africa
 email: markus.louw@gmail.com

ABSTRACT

A new MRF parameter estimation method is proposed, based on the Population Monte Carlo algorithm. This method is used to estimate MRF parameters which correspond to the Bidirectional Reflectance Distribution Function of a material surface given its reconstructed surface geometry, camera matrices and lighting data. The posterior distributions on the MRF/BRDF parameters are then used as features for material classification.

KEY WORDS

BRDF, MRF, PMC, stereo reconstruction.

1 Introduction

This paper proposes a new method for doing MRF parameter estimation on a lattice of random variables which encode the surface geometry of a stereo reconstructed surface. This method for parameter estimation was inspired by [16], where a MCMC approach is used to approximate the posterior distribution on a MRF parameter set, after which the parameters are used to classify the image by texture. We use the MRF formulation for shape from shading developed in [11], [10], which describes a set of MRF potential functions which allow a surface to be reconstructed given probabilistic range data on surface points and known camera matrices and lighting direction vectors. The method can work for any reflectance model, and convergence results using the Ward BRDF model using synthetic data are presented. We proceed to describe the Ward model, then the MRF formulation for shape from shading, then the Population Monte Carlo method [3], [7], [6]. Classification results are presented using both the Ward model and the anisotropic Phong model of [1].

2 The Isotropic Gaussian Ward reflectance model

This section describes the isotropic Ward BRDF model, as expounded in [17]. If the angles of incident light are (θ_i, ϕ_i) and reflected light are (θ_r, ϕ_r) , then isotropic Ward

model is given by

$$\rho_{bd,iso}(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{\rho_d}{\pi} + \rho_s \frac{\exp(-\tan^2 \delta / \alpha^2)}{4\pi\alpha^2 \sqrt{\cos \theta_i \cos \theta_r}}, \quad (1)$$

where ρ_d is the diffuse reflectance, ρ_s is the specular reflectance, α is the standard deviation of the surface slope, and δ is the angle between the half vector \mathbf{h} and the surface normal $\hat{\mathbf{n}}$. The model is clearly symmetrical, and the normalizing term ensures the correct energy balance. The isotropic model may be extended to the anisotropic one if it is assumed that the surface has different (uncorrelated) roughnesses in perpendicular directions along the surface. These roughnesses are denoted by α_u and α_v . The Ward model function may be calculated using

$$\rho_{bd}(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{\rho_d}{\pi} + \rho_s \frac{1}{4\pi\alpha_u\alpha_v\sqrt{\cos \theta_i \cos \theta_r}} \exp\left(-2 \frac{\frac{\mathbf{h}\cdot\mathbf{u}}{\alpha_u} + \frac{\mathbf{h}\cdot\mathbf{v}}{\alpha_v}}{1 + \mathbf{h}\cdot\hat{\mathbf{n}}}\right), \quad (2)$$

where

$$\mathbf{h}\cdot\mathbf{u} = \frac{\sin \theta_r \cos \theta_r + \sin \theta_i \cos \theta_i}{\|\vec{\mathbf{h}}\|}, \quad (3)$$

$$\mathbf{h}\cdot\mathbf{v} = \frac{\sin \theta_r \sin \theta_r + \sin \theta_i \sin \theta_i}{\|\vec{\mathbf{h}}\|} \quad (4)$$

and

$$\mathbf{h}\cdot\hat{\mathbf{n}} = \frac{\cos \theta_r + \cos \theta_i}{\|\vec{\mathbf{h}}\|}, \quad (5)$$

with

$$\|\vec{\mathbf{h}}\| = [2 + 2 \sin \theta_r \sin \theta_i (\cos \phi_r \cos \phi_i + \sin \phi_r \sin \phi_i) + 2 \cos \theta_r \cos \theta_i]^{\frac{1}{2}}. \quad (6)$$

The following substitutions are used for vector calculations:

$$\vec{\mathbf{h}} = \mathbf{k}_1 + \mathbf{k}_2, \quad \mathbf{h} = \frac{\vec{\mathbf{h}}}{\|\vec{\mathbf{h}}\|}, \quad (7)$$

$$\cos(\theta_r) = \mathbf{k}_1 \cdot \hat{\mathbf{n}}, \quad \cos(\theta_i) = \mathbf{k}_2 \cdot \hat{\mathbf{n}}. \quad (8)$$

In the above equations \mathbf{k}_2 is the reflected ray direction (away from the surface), \mathbf{k}_1 is the incident ray direction

(away from the surface), \mathbf{u} is a unit vector in the surface plane, and \mathbf{v} is a unit vector in the surface plane, perpendicular to \mathbf{u} . It is noted in [17] that there is some spectral dependence between ρ_d and ρ_s , and that the normalization factor $\frac{1}{4\pi\alpha^2}$ loses some accuracy when $\alpha > 0.2$.

3 Markov random field framework for BRDF parameter estimation

This section outlines the Markov random field (MRF) framework which is used to estimate reflectance parameters of stereo reconstructed surfaces. It may seem unnatural to estimate parametric probabilistic BRDF models using MRF parameter estimation, but it can be justified by noting that since (uninterpolated) a disparity map always consists of a set of discrete random variables which have uncertainties associated with them, any stereo based 3D reconstruction of a surface is in fact the realization of a discrete 2-D Markov Random Field, where the labels on the image points correspond to the projected depths of those points into the scene. This idea was stimulated by the previous research of [11] and [10], where the shape from shading problem are cast into a MRF framework. This led us to hypothesize that the same reflectance parameters of the surface can be estimated using the same MRF framework, by treating a probabilistic surface reconstruction as a posterior distribution on a MRF. We now describe this MRF formulation.

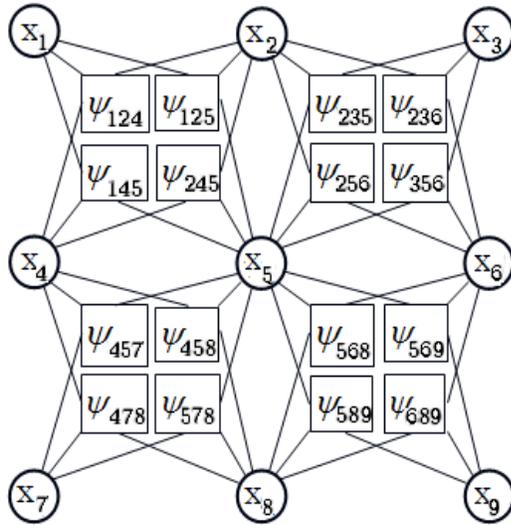


Figure 1. This diagram illustrates the energy terms (square nodes) on the triplets of random variables representing the labels on corner vertex nodes (round nodes labeled x_i). If a loopy belief propagation approach is used to calculate the posterior distribution (as in [11]), the square nodes would represent factor nodes connected to the variable nodes (round nodes).

The plane generated by the triplet of corner vertex

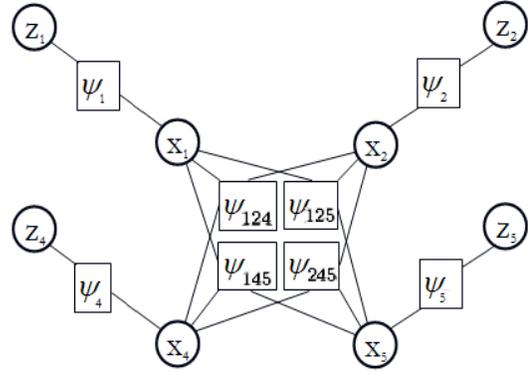


Figure 2. This diagram illustrates the energy terms (square nodes) on the triplets of random variables representing the labels on corner vertex nodes (round nodes labeled x_i) as well as the corresponding dependence of the corner vertex node labels x_i on the range data for each corner vertex node contained in z_i . The round nodes labeled z_i are visible nodes giving range or depth data.

nodes for each clique forms an angle with the incident light, giving a reflectance value for the pixel, or for the image region which corresponds to that triangle on the surface. The expected image intensity value, as perceived by the camera depends on the reflectance function used. This is shown in Fig. 3, where the assumption is made that the corner vertex node labels correspond to discrete distances measured perpendicularly from the image plane.

Next a Markov random field (MRF) is defined on this set (lattice) of corner vertex nodes \mathbf{X} , given the image data \mathbf{Y} and explicit range data \mathbf{Z} (which may come from a source such as a laser scanner or sparse surface reconstruction). This MRF is used to derive a probability for the depth or range of the surface at the location of the surface corresponding to a particular corner vertex node:

$$p(\mathbf{X}|\mathbf{Y}, \mathbf{Z}, \theta) \propto \prod_{\substack{i,j,k \\ i < j < k}} \exp(-\psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta)) \prod_i \exp(-\psi_i(x_i, z_i)) \quad (9)$$

The energy of a particular set of corner vertex nodes (i, j, k) in a clique taking on a particular set of labels (x_i, x_j, x_k) is taken to be

$$\psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta) = |y_{ijk} - \text{BRDF}(i, j, k, x_i, x_j, x_k, \vec{L}, \mathbf{P}, \theta)| / \sigma_b \quad (10)$$

where the BRDF parameters are contained in θ , and y_{ijk} is the pixel intensity (on a gray scale from 0 to 1) of the image region contained by the three vertex nodes (usually one pixel), and x_i, x_j, x_k are the labels of corner vertex nodes i, j, k . \mathbf{P} contains the two camera projection matrices of the two cameras in the stereo rig, and σ_b is a parameter

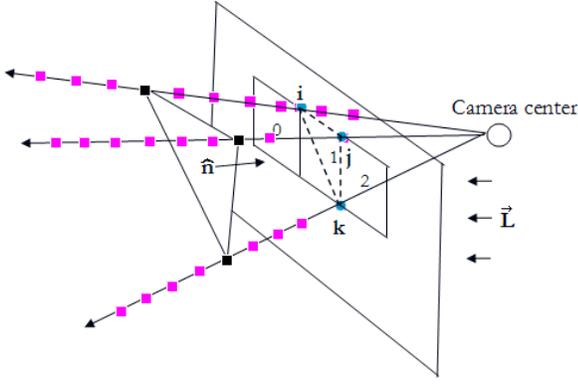


Figure 3. A triangular plane is generated by the three points on the surface, each corresponding to a label value for the site of one of the corner vertex nodes surrounding the pixel. The normal to the plane $\hat{\mathbf{n}}$ and the incident light source direction \vec{L} are indicated. The square region interior to four corner vertex nodes is a pixel

which affects the sharpness of the energy function. The potential term $\psi_i(x_i, z_i)$ encodes probabilistic range data on the label x_i at node i . An example of a workable potential function for an affine camera is

$$\psi_i(x_i, z_i) = |z_i - u_i(x_i)|/\sigma_r, \quad (11)$$

where z_i is the specified depth or range of the surface at point i , and $u_i(x_i)$ is the depth or range at surface point i given the label of the corner vertex node x_i at that point on the surface. As before, σ_r is a parameter which affects the sharpness of the function. Whether the point is given a value because it lies on a known boundary or because we have range data about the point, it is treated the same way. The potential energy term $\psi_i(x_i, z_i)$ in Eqn. 9 can be used to incorporate such a constraint. In addition, the specification of boundary conditions may resolve some of the ambiguities, since there is generally a number of surfaces that generate a particular intensity map under particular lighting conditions [5]. The MRF formulation allows such boundary conditions and range data to take on the form of either hard or soft constraints.

BRDF(\cdot) is the reflectance model, which returns the expected intensity at the location described by the indices i, j, k , given the light source, the surface and the camera information. The simplifying assumption that the camera is affine gives the following equations for the partial derivatives in the height (with respect to change in position in the horizontal and vertical directions on the image):

$$p = \partial u / \partial x \quad \text{and} \quad q = \partial u / \partial y, \quad (12)$$

where u represents the height of the surface, x and y are orthogonal directions on the image plane. Assuming square pixels and an overall scale of one unit per pixel, ∂x and ∂y are set to 1, and the calculated elevation difference on

opposite sides of a pixel is ∂u . With these assumptions, the surface normal is calculated as

$$\hat{\mathbf{n}} = (-p, -q, 1) / \sqrt{(p^2 + q^2 + 1)}. \quad (13)$$

If the BRDF under consideration is the simple Lambertian reflectance model, then

$$\text{BRDF}(i, j, k, x_i, x_j, x_k, \vec{L}, \mathbf{P}, \theta) = |\hat{\mathbf{n}} \cdot \vec{L}|, \quad (14)$$

where the vector θ contains the BRDF parameter set (it is empty in this case, since the Lambertian reflectance model does not have any associated parameters). If the BRDF model follows the Ward model, the parameters would be $\theta = (\rho_s, \rho_d, \alpha)$.

For a projective camera, the calculation of the local surface normal becomes a task of extracting the 3-D locations of the triangulated points and calculating the normal using the vector cross-product. Thus, if \mathbf{Z}_i , \mathbf{Z}_j , and \mathbf{Z}_k are the 3-D locations in world coordinates of surface/image points i, j, k , then

$$\hat{\mathbf{n}} = (\mathbf{Z}_j - \mathbf{Z}_i) \times (\mathbf{Z}_k - \mathbf{Z}_i) / \|(\mathbf{Z}_j - \mathbf{Z}_i) \times (\mathbf{Z}_k - \mathbf{Z}_i)\|, \quad (15)$$

where “ \times ” indicates the vector cross-product operation (it is also required that the normal points in the direction of the camera). “BRDF” is a function which returns the intensity value given local surface geometry information (the 3-D locations of the surface points corresponding to the three corner vertex nodes of a particular clique). The camera projection matrices \mathbf{P}_1 and \mathbf{P}_2 for the stereo pair are contained in \mathbf{P} .

4 MRF parameter estimation using the Pseudolikelihood approximation

The task of parameter estimation, or of the evaluation of the probability of a given labelling with a given set of parameters, requires the evaluation of a function of the form

$$p(f|\theta) = Z(\theta)^{-1} \exp(-U(f, \theta)) = \frac{\exp(-U(f, \theta))}{\sum_{s \in F} \exp(-U(s, \theta))}, \quad (16)$$

where f is the realization of the site labellings, and F is the configuration space of all possible site labellings. The problem with evaluating this probability is that $Z(\theta)$, known as the partition function, is combinatorially difficult to calculate. Much effort has gone into developing methods for approximating the partition function, as described in the literature review.

One method for MRF parameter estimation, and for bypassing the need to evaluate the partition function when evaluating probabilities on lattice site labellings, is to use the pseudolikelihood estimate, proposed in [2]. The conditional probability per site on a graph is estimated using only its immediate (Markov) neighbours. This is now expressed

using a potential function $U^A(\cdot)$. Note that the pseudo-likelihood equals the true likelihood when all the labels are independent. For a single site i (and neighbourhood) labelling,

$$p(f_i|f_{\mathcal{N}_i}, \theta) = \frac{\exp(-U^A(f_i, f_{\mathcal{N}_i}, \theta))}{\sum_{f_s \in \mathcal{L}} \exp(-U^A(f_s, f_{\mathcal{N}_s}, \theta))}, \quad (17)$$

where f_i is the label on node i , \mathcal{N}_i is the neighbourhood system on node i , \mathcal{L} is the set of possible labels, θ is the set of MRF parameters, and S is the set of discrete sites on the lattice. The pseudolikelihood for all the labels on the lattice is

$$PL(f|\theta) = \log \prod_{i \in S} p(f_i|f_{\mathcal{N}_i}, \theta). \quad (18)$$

This pseudolikelihood estimate is necessary to approximate

$$p(\theta|f) = p(f|\theta) \frac{p(\theta)}{p(f)}. \quad (19)$$

If the pseudolikelihood estimate of $p(f|\theta)$ is available, and if we assume uniform priors $p(\theta)$ and $p(f)$, then we can use

$$p(\theta|f) = \exp(PL(f|\theta)), \quad (20)$$

which immediately allows us to use Monte Carlo type techniques to explore the distribution $p(\theta|f)$, and Maximum Likelihood Estimation (MLE) and gradient descent methods to find its MAP value [9].

To calculate the pseudolikelihood of a BRDF label set given the parameters θ , we define the potential function $U^A(\cdot)$ for node i to be

$$U_i^A(x_i, x_{\mathcal{N}_i}, \theta) = \sum_{\substack{j \in \mathcal{N}_i, k \in \mathcal{N}_i \\ k > j}} \psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta), \quad (21)$$

where $j, k \in \mathcal{N}_i$ indicates that j and k are indices of nodes neighbouring node i , and θ contains the BRDF parameters ($\theta = (\rho_d, \rho_s, \alpha)$, in the case of the Ward model). The function implicitly uses \vec{L} (the lighting direction) and \mathbf{P} which contains the camera projection matrices. As potential functions with the clique connectivity indicated in Fig. 2 are being used, the potential term U^A involves a summation of four local clique potentials per corner vertex node.

The pseudolikelihood estimate of the parameters θ , given a particular realization of the MRF, is expressed as

$$\theta_{PL}^* = \arg \max_{\theta} PL(f|\theta). \quad (22)$$

5 The theory of Population Monte Carlo

According to [3], early MCMC literature attempted to dissociate itself from the literature on importance sampling, even though both of these had in common the notion of

sampling from a proposal distribution other than the prior distribution while arriving at the correct posterior distribution by using normalization and reweighting. It was only much later that MCMC and importance sampling were combined (e.g. [12], [8]).

In our implementation, there is little difference between partitioned particle filtering (over a single time step) and PMC. In [3], it is shown that the importance functions per sample (as each sample may have its own importance function) may depend in any way on the previous importance functions and sample distributions. This is because the sample set is immediately reweighted to represent a draw from the target distribution, at every iteration. Theoretically, PPF applies the same importance function g^n (with resampling) in the sequence to all of the particles in partition n . Suppose we want to construct a target distribution $\varrho(\cdot)$ using a MCMC sampler, where we can specify the probability of a set of samples in terms of the product of the probability of each sample individually:

$$\varrho^{\otimes n}(\theta_1, \dots, \theta_n) = \prod_{i=1}^n \varrho(\theta_i). \quad (23)$$

Here the distribution is on the space χ^n , whereas $\varrho(\theta_i)$ is on the space χ . PMC allows one to avoid the problem of calculating the convergence of an MCMC sampler to the correct stationary distribution by using importance sampling to correct at each time step for the bias introduced by the proposal distribution.

To use this, each sample θ_i^t in the sample set $\theta^t = (\theta_1^t, \dots, \theta_n^t)$, (each sample has been drawn from g_i^t) is reweighted by

$$\pi_i^t = \frac{\varrho(\theta_i^t)}{g_i^t(\theta_i^t)}, \quad i = 1, \dots, n, \quad (24)$$

where g_i^t is the proposal distribution for the simulation of θ_i^t . This definition implies that estimators which take the form

$$\mathfrak{J}^t = \frac{1}{n} \sum_{i=1}^n \pi_i^t h(\theta_i^t) \quad (25)$$

are unbiased estimators of $E_{\varrho}[h(\theta)]$ at every iteration t , for every integrable function $h(\cdot)$. Assuming that the variances

$$\text{var}(\pi_i^t h(\theta_i^t)) \quad (26)$$

exist for all $1 \leq i \leq N$, i.e. that the proposals for g_i^t have heavier tails than ϱ , then the variance decomposition rule for \mathfrak{J} , namely

$$\text{var}(\mathfrak{J}^t) = \frac{1}{N^2} \sum_{i=1}^N \text{var}(\pi_i^t h(\theta_i^t)), \quad (27)$$

implies that the importance-weighted terms are always uncorrelated.

Since distributions ϱ are often unscaled and unnormalized, one can instead use

$$\pi_i^t \propto \frac{\varrho(\theta_i^t)}{g_i^t(\theta_i^t)}, \quad i = 1, \dots, N. \quad (28)$$

The weights, although now normalized, have caused the distributions to lose their unbiasedness and variance decomposition properties, although they still approximate the true distributions.

As noted in [13], instead of updating the weights at each iteration, it improves the representation of target distributions to resample N values y_i^t (with replacement) from the sample set $(\theta_1^t, \dots, \theta_N^t)$ at each time step t , according to the sample weights π_i^t . This ameliorates the degeneracy problem, where irrelevant samples are maintained and do not help in representing the distribution compactly. The new sample set (y_1^t, \dots, y_N^t) resulting from this resampling operation is similar to an i.i.d. sample taken from the distribution $\varrho^{\otimes n}(\theta_1, \dots, \theta_N)$.

The essential feature of the PMC sampler is that at iteration t , N values are simulated from a proposal distribution which depends itself on the $N \times (t - 1)$ previous samples. There is almost no constraint on the dependencies of the new importance distributions on the old ones or on the previous samples. In [14], it is noted that in the absence of repeated resampling operations at each iteration of the PMC sampler, the algorithm is equivalent to Metropolis-Hastings sampling in the N dimensional space χ^N (N is the number of samples in the sample set), which converges to the target distribution $\varrho^{\otimes n}$, (i.e. to the same desired target distribution). It is also equivalent to N parallel Metropolis-Hastings samplers which accept or reject each sample in the N dimensional sample set, (i.e. a parallel MCMC sampler), which converges to the target distribution $\varrho^{\otimes n}$.

The generic PMC method can be described (taken from [4]) as follows:

- $n=0$: Initialize sample locations and probability weights:
 1. Generate $(\theta_i^0)_{1 \leq i \leq N} \sim g^0$
 2. Compute $(\pi_i^0)_{1 \leq i \leq N} = (p(\theta_i^0)/g^0(\theta_i^0))_{1 \leq i \leq N}$.
 3. Resample sample set $(\theta_i^0, \pi_i^0)_{1 \leq i \leq N}$, choosing sample i with probability π_i^0 .
- For $n > 0$
 1. Conditionally on previous θ_i^j and $\theta_{i,\text{new}}^j$ ($j < n$), generate independently $(\theta_i^n)_{1 \leq i \leq N} \sim g_i^n$
 2. Compute $(\pi_i^n)_{1 \leq i \leq N} = (p(\theta_i^n)/g_i^n(\theta_i^n))_{1 \leq i \leq N}$.
 3. Resample sample set $(\theta_i^n, \pi_i^n)_{1 \leq i \leq N}$, choosing sample i with probability π_i^n .

6 The softening of a probability density function

It is beneficial to use a sharpening/softening function on the posterior distribution of a random variable if it is represented by a set of samples, especially when applying a set of different cost functions to the particles at different stages. When the resampling of these particles is done, and if one particle has a very high probability, that particle will be sampled repeatedly, resulting in degeneracy. This is especially likely when using probabilities based on multiplying likelihoods over thousands of pixels/nodes. We would rather maintain bad (low probability) samples for a few more resampling iterations, in case they prove to be good samples under some other energy function (a clique potential over other corner vertex node triplets in the image data in our case), or if there are other local maxima close to some of the local minima. To do this we use importance sampling for the same effect as simulated annealing, within the PMC paradigm. The method used to soften our distributions is

$$\pi_i^{\text{new}} \propto (\pi_i^{\text{old}})^l, \quad (29)$$

i.e. for each sample, take the weight of the i^{th} sample and take it to the l^{th} power. After all the weights have been adjusted in this way, the weights $\bar{\pi}^{\text{new}}$ are normalized. Note that with this softening algorithm, values of $l < 1$ soften the distribution, while $l > 1$ sharpen it. The operation of this algorithm on a pdf $p(\theta)$ can be written as

$$p^{\text{new}}(\theta) = s(p(\theta)). \quad (30)$$

7 Overview of process for probabilistic dense stereo reconstruction and MRF parameter estimation

This section outlines the process we used to create a dense probabilistic surface reconstruction, and for doing MRF parameter estimation to estimate BRDF parameters. The process can be divided into the following steps:

1. Capture stereo images **A** of calibration object
2. Capture stereo images **B** of scene under structured lighting (checkerboard pattern)
3. Capture stereo images **I** of scene illuminated by white light
4. Calculate and calibrate projection matrices of cameras **P** in stereo pair using images **A**
5. Rectify images **B** giving images **I^R** using calibrated projection matrices **P** (and point correspondence data for images **A**)
6. Do dense stereo correspondence on rectified images **I^R**, calculating disparity field **X** and probabilities (beliefs) on this field **b**
7. Use probabilistic dense stereo correspondence data **X** and **b**, along with image intensity information (images

I), camera projection matrix data (calculated in 4), and light source information to estimate MRF parameters, using PMC, dynamically weighted MCMC, or multiple-seed Levenberg- Marquardt algorithm.

To do step 6, we chose to use the accelerated Loopy belief propagation dense stereo correspondence algorithm due to Tappen [15], which propagates disparity information quickly across the lattice. Our MRF estimation of BRDF parameters allows us to use the final posterior belief on each corner vertex node label, after the image and pairwise clique interactions have been taken into account, and the information has spread across the whole disparity map. One way of including this in the pseudolikelihood framework, is to convert the belief on the label of any transformed corner vertex node back into a potential function. For a single potential energy term, $p(x) = k_1 \exp(-U^D(x)/T)$. Therefore we convert each probability vector $b(x_i)$ on the disparity label of each corner vertex node into a potential term for that node:

$$U_i^D(x_i, \mathbf{I}^R) = \psi_i^D(x_i) = -T \log(b(x_{r(i)}^{R1})/k_1), \quad (31)$$

where T is the temperature and k_1 is a normalization constant. This formulation permits the incorporation of probabilistic information gained from LBP type dense stereo correspondence algorithms about the likelihood of any label x_i^{R1} in the rectified image \mathbf{I}^{R1} , into a potential energy term, within the paradigm of Gibbs random field type energy interactions.

Note that the LBP algorithm for dense stereo correspondence is calculated using the pair of rectified images \mathbf{I}^R with the set of corner vertex nodes on the lattice of sites in \mathbf{I}^{R1} , and not on the set of corner vertex nodes for the first unrectified image \mathbf{I}^1 . A pseudolikelihood which incorporates this dense stereo-based probability through the potential energy as in Eqn. 31 is

$$\begin{aligned} PL(\mathbf{X}|\theta) &= \log \prod_{i \in S} p(x_i | x_{N_i}, \theta) \\ &= \log \prod_{i \in S} \frac{\exp(-U_i^A(x_i, x_{N_i}, \theta) - U_i^D(x_i, \mathbf{I}^R))}{\sum_{s \in \mathcal{L}} \exp(-U_i^A(s, x_{N_i}, \theta) - U_i^D(s, \mathbf{I}^R))}. \end{aligned} \quad (32)$$

For notational convenience we also write the function

$$\begin{aligned} ePL(\mathbf{X}|\theta) &= \exp(PL(\mathbf{X}|\theta)) \\ &= \prod_{i \in S} \frac{\exp(-U_i^A(x_i, x_{N_i}, \theta) - U_i^D(x_i, \mathbf{I}^R))}{\sum_{s \in \mathcal{L}} \exp(-U_i^A(s, x_{N_i}, \theta) - U_i^D(s, \mathbf{I}^R))}. \end{aligned} \quad (33)$$

8 Pseudolikelihoods

Now that the third pseudolikelihood function PL has been defined, the importance function of Eqn. 34 can be reformulated:

$$G(\theta|\mathbf{X}) = s(ePL(\mathbf{X}|\theta)), \quad (34)$$

remembering that we are using $p(\theta|\mathbf{X}) = \exp(PL(\mathbf{X}|\theta))$, since there are uniform priors on \mathbf{X} and θ , and where $s(\cdot)$ is the softening function, as described previously (Eqn. 30). This is the importance function used for the PMC based MRF parameter estimation. As with the previous pseudo-likelihood functions, one can construct importance functions G^n based on iteration over a certain part the lattice, using

$$ePL^n(\mathbf{X}|\theta) = \prod_{i \in S^n} p(x_i | x_{N_i}, \theta), \quad (35)$$

where S^n is the n^{th} subset of the sites on the lattice, for example $S^n = \{n, n+k, n+2k, \dots\}$, where k is the number of subsets into which the surface is partitioned. This function can be softened, giving

$$G^n(\theta|\mathbf{X}) = s(ePL^n(\mathbf{X}|\theta)). \quad (36)$$

The calculation of correctly weighted samples at every time step can be avoided by using an algorithm that calculates the correctly weighted sample set whenever it is required, but maintains a set of weighted samples which always represents the uniform distribution, retaining the samples which are in positions considered to be important. This is the PMC algorithm used in this thesis:

8.1 Population Monte Carlo algorithm

- $n=0$: Initialize sample locations and probability weights:
1. Generate samples $(a_i^0)_{1 \leq i \leq N} \sim G^0$
 2. Compute weights $(c_i^0)_{1 \leq i \leq N} = (1/G^0(a_i^0|\mathbf{X}))_{1 \leq i \leq N}$
 3. Normalize the weights c_i^0
 4. If the correct posterior distribution $ePL(\mathbf{X}|\theta)$ is needed:
 5. Compute $(\pi_i^0)_{1 \leq i \leq N} = (ePL(\mathbf{X}|a_i^0)/G^0(a_i^0|\mathbf{X}))_{1 \leq i \leq N}$
 6. $\{\vec{\theta}^0, \vec{\pi}^0\} = \text{resampleParticles}(\{\vec{a}^0, \vec{\pi}^0\})$
 7. end.
- For $n > 0$
1. Generate samples $(a_i^n)_{1 \leq i \leq N} \sim G^n(\{\vec{a}^{n-1}, \vec{c}^{n-1}\}|\mathbf{X})$
 2. Compute weights $(c_i^n)_{1 \leq i \leq N} = (1/G^n(a_i^n|\mathbf{X}))_{1 \leq i \leq N}$
 3. Normalize the weights c_i^n
 4. If the correct posterior distribution $ePL(\mathbf{X}|\theta)$ is needed:
 5. Compute $(\pi_i^n)_{1 \leq i \leq N} = (ePL(\mathbf{X}|a_i^n)/G^n(a_i^n|\mathbf{X}))_{1 \leq i \leq N}$
 6. $\{\vec{\theta}^n, \vec{\pi}^n\} = \text{resampleParticles}(\{\vec{a}^n, \vec{\pi}^n\})$
 7. end.

It is necessary to specify how the sampling is done at step 1 in each case. When $n = 0$, at step 1, which reads “ $(a_i^0)_{1 \leq i \leq N} \sim G^0$ ”, the samples are generated in the following way:

1. Generate an evenly spaced set of samples $(u_i^0)_{(1 \leq i \leq N)}$ over the parameter space to represent the uniform distribution in \mathbb{R}^3 , and set the weights $c_i^0 = G^0(u_i^0|\mathbf{X})$, $(1 \leq i \leq N)$

2. $\{\vec{a}^0, \vec{c}^0\} = \text{resampleParticles}(\{\vec{u}^0, \vec{c}^0\}, \sigma_0)$

The second case step 1 ($n > 0$), which reads ‘‘Generate samples $(a_i^n)_{1 \leq i \leq N} \sim G^n(\{\vec{a}^{n-1}, \vec{c}^{n-1}\}|\mathbf{X})$ ’’, is performed in the following way:

1. Reweight each of the samples a_i^{n-1} , for $1 \leq i \leq N$, according to

$$w_i = c_i^{n-1} G^n(a_i^{n-1}|\mathbf{X}) \quad (37)$$

2. $\{\vec{a}^n, \vec{c}^n\} = \text{resampleParticles}(\{\vec{a}^{n-1}, \vec{w}\}, \sigma_n)$.

As before, the additive Gaussian noise at the resampling stage is parameterized by σ_n , which is the standard deviation of the noise, decreasing at each iteration of the PMC algorithm.

Note that each sample θ_i^n contains the information for the BRDF parameters. \mathbf{P} (both camera projection matrices), and \vec{L} (the light source direction), are used implicitly in the pseudolikelihood calculations. \mathbf{P} has been calculated through calibration algorithms using a calibration checkerboard, and \vec{L} is estimated from the scene geometry (the projector is positioned midway between the two cameras).

In our experiments, the standard deviation of the additive Gaussian noise for the resampling operation ($\text{resampleParticles}(\cdot, \sigma_n)$) is initialized with $\sigma_0 = 0.07$, and decreases linearly over the PMC iterations until finally $\sigma_T = 0$.

9 BRDF parameter estimation on synthetic data

9.1 Convergence results for the PMC method

The posterior distribution of Eqn. 33 for a range of synthetic surfaces was explored as the target distribution $\varrho(\cdot)$ using the PMC method. The methodology for gathering synthetic data for the performance analysis of the PMC algorithm follows:

1. For $n=1, \dots, N$
2. Generate random surface and virtual range disparity field \mathbf{X} consisting of the possible labels for the range of disparity values for each corner vertex node in the image, and \mathbf{S} , the labelling of \mathbf{X} which corresponds to the true surface
3. Generate random BRDF parameter vector \vec{b}^n , and render intensity map \mathbf{I}^1 of true surface (\mathbf{S}) given light source \vec{L} , camera data \mathbf{P} , and reflectance model and parameters \vec{b}^n ,
4. Run PMC algorithm with 64,125,1000 samples for 20 iterations, using Eqn. 33 as the target distribution, storing each sample set $(\vec{\theta}, \vec{\pi})^{n,i}$ at each iteration i
5. end(1).
6. The MAP sample is found by choosing the sample θ with the highest probability weight π at any iteration i . The distance of the MAP sample to the true synthetic BRDF parameter set is stored for every iteration. The

averages of these distances over 100 runs are recorded in Tables 1 to 4, per PMC iteration, in the columns labeled ‘‘MAP dist’’.

7. The variance of the sample set weights $\vec{\pi}^{n,i}$ is calculated for each iteration, and averaged over the number of trial runs ($N=100$). These are also recorded in Tables 1 and 2, per PMC iteration, in the columns labelled ‘‘var’’.

The variance of the weights in a weighted sample set is an indication of the number of effective samples, and is used to show that the samples are not being wasted in regions of the target distribution of low probability. If the samples were thus distributed, the sample variance would be higher. A sample variance which decreases over successive PMC iterations thus indicates that the PMC algorithm is finding the local maxima in the distribution (if such local maxima exist).

It is seen in each of these tables that the average Euclidean distance between the true BRDF parameters used to generate each surface image and the MAP sample of the sample set (indicated in columns by ‘‘MAP dist’’, which indicates the Euclidean distance between the true sample and the MAP sample) decreases as the PMC algorithm proceeds through each iteration. The sample variances (indicated in columns by ‘‘var’’) also decrease as the PMC algorithm proceeds through each iteration. Note that the elements of Tables 1 to 4 are average values over 100 runs. These results are presented in table form since there are too many variables to easily distinguish graphically plotted data.

| numLabels | 10 | 10 | 10 | 10 | 10 | 10 |
|------------|----------|----------------------|----------|----------------------|----------|----------------------|
| numSamples | 1000 | 1000 | 125 | 125 | 64 | 64 |
| PMC | MAP dist | var | MAP dist | var | MAP dist | var |
| iters | | ($\times 10^{-6}$) | | ($\times 10^{-6}$) | | ($\times 10^{-6}$) |
| 1 | 0.3439 | 18.2 | 0.4054 | 38.2 | 0.489 | 671 |
| 3 | 0.1872 | 0.887 | 0.355 | 31.79 | 0.4643 | 104.1 |
| 5 | 0.1829 | 0.4873 | 0.3318 | 17.94 | 0.4664 | 31.86 |
| 7 | 0.1595 | 0.4411 | 0.3122 | 92.38 | 0.4647 | 49.42 |
| 9 | 0.1531 | 0.3693 | 0.306 | 90.41 | 0.435 | 60.79 |
| 11 | 0.1578 | 0.3439 | 0.2875 | 39.84 | 0.4389 | 20.02 |
| 13 | 0.1446 | 0.3042 | 0.2752 | 41.95 | 0.4262 | 20.47 |
| 15 | 0.1432 | 0.2662 | 0.2731 | 48.82 | 0.4332 | 22.6 |
| 17 | 0.1367 | 0.2577 | 0.2765 | 30.33 | 0.4325 | 10.96 |
| 20 | 0.1324 | 0.1509 | 0.2831 | 21.13 | 0.4231 | 2.829 |

Table 1. Convergence results of the PMC algorithm when using the isotropic Ward reflectance model. Average statistics are shown over the 20 iterations of the PMC algorithm. The image size is 100x100 pixels. The table is populated with average values over 100 runs.

10 Classification using stereo images of real surfaces

To test the PMC evolved sample sets as features for surface classification, reconstructions of 48 material surface instances were captured, and divided into 12 classes, shown in Figs 4, 5, 6. For the classification of surfaces through the comparison of their BRDFs (represented by sample sets),

| numLabels | 20 | 20 | 20 | 20 | 20 | 20 |
|------------|----------|----------------------|----------|----------------------|----------|----------------------|
| numSamples | 1000 | 1000 | 125 | 125 | 64 | 64 |
| PMC | MAP dist | var | MAP dist | var | MAP dist | var |
| iters | | ($\times 10^{-6}$) | | ($\times 10^{-6}$) | | ($\times 10^{-6}$) |
| 1 | 0.2863 | 3.609 | 0.4547 | 343.1 | 0.5139 | 507.7 |
| 3 | 0.1356 | 0.2929 | 0.3955 | 61.44 | 0.4851 | 91.51 |
| 5 | 0.1266 | 0.148 | 0.362 | 54.08 | 0.4574 | 47.1 |
| 7 | 0.1142 | 0.09438 | 0.3217 | 39.96 | 0.4413 | 39.58 |
| 9 | 0.1221 | 0.07107 | 0.3243 | 36.71 | 0.4413 | 33.72 |
| 11 | 0.1109 | 0.05896 | 0.3187 | 34.53 | 0.4268 | 14.03 |
| 13 | 0.1109 | 0.04885 | 0.3066 | 27.7 | 0.4211 | 26.8 |
| 15 | 0.1026 | 0.03878 | 0.3111 | 22.42 | 0.4192 | 25.01 |
| 17 | 0.1025 | 0.06599 | 0.3042 | 30.07 | 0.4007 | 15.97 |
| 20 | 0.1043 | 0.01832 | 0.2979 | 16.03 | 0.4029 | 1.434 |

Table 2. Convergence results of the PMC algorithm when using the isotropic Ward reflectance model. Average statistics are shown over the 20 iterations of the PMC algorithm. The image size is 100x100 pixels. The table is populated with average values over 100 runs.

| numLabels | 30 | 30 | 30 | 30 | 30 | 30 |
|------------|----------|----------------------|----------|----------------------|----------|----------------------|
| numSamples | 1000 | 1000 | 125 | 125 | 64 | 64 |
| PMC | MAP dist | var | MAP dist | var | MAP dist | var |
| iters | | ($\times 10^{-6}$) | | ($\times 10^{-6}$) | | ($\times 10^{-6}$) |
| 1 | 0.3525 | 13.2 | 0.4325 | 17.7 | 0.5345 | 1685 |
| 3 | 0.2024 | 0.5346 | 0.4254 | 24.9 | 0.5205 | 49.54 |
| 5 | 0.166 | 0.4922 | 0.4129 | 9.79 | 0.5344 | 62.12 |
| 7 | 0.1585 | 0.3615 | 0.3994 | 5.312 | 0.5183 | 44.23 |
| 9 | 0.1363 | 0.4002 | 0.3966 | 8.043 | 0.49 | 47.86 |
| 11 | 0.1383 | 0.3318 | 0.3995 | 6.799 | 0.4934 | 30.02 |
| 13 | 0.1424 | 0.3007 | 0.3982 | 4.16 | 0.4899 | 41.57 |
| 15 | 0.1354 | 0.2186 | 0.3962 | 5.621 | 0.4808 | 27.12 |
| 17 | 0.1178 | 0.1805 | 0.3984 | 4.038 | 0.4811 | 40.7 |
| 20 | 0.1188 | 0.1157 | 0.3984 | 0.0005852 | 0.4815 | 3.233 |

Table 3. Convergence results of the PMC algorithm when using the isotropic Ward reflectance model. Average statistics are shown over the 20 iterations of the PMC algorithm. The image size is 100x100 pixels. The table is populated with average values over 100 runs.

it is necessary to define distance measures between probability density functions. In particular, it is necessary to define measures between weighted sets of samples, since the posterior distributions on the BRDFs are represented by weighted sample sets, which are calculated using the PMC algorithm.

10.1 Comparing sample sets

The analytic expressions for calculating some of these probabilistic distances, if the samples representing the pdfs are assumed to be drawn from normal distributions, are shown in Table 5. Although the sample sets representing the posterior distributions for the parametric BRDF models that are calculated for the material and froth surfaces are not necessarily normally distributed, they are close enough to normal to support the use of these analytic expressions for probabilistic distance calculation. Thus the means and variances of the evolved weighted sample sets, calculated using the PMC method, are estimated and substituted into the equations in Fig. 5.

In Tables 6 to 9, classification results are given using the ‘‘Miss’’, ‘‘WNN’’ and ‘‘NN1’’ error classification statistics. The ‘‘Miss’’ statistic is a count of the number of times in the similarity matrix (iterating once over the entries for each material) that one of the extra-class materials has a

| numLabels | 40 | 40 | 40 | 40 | 40 | 40 |
|------------|----------|----------------------|----------|----------------------|----------|----------------------|
| numSamples | 1000 | 1000 | 125 | 125 | 64 | 64 |
| PMC | MAP dist | var | MAP dist | var | MAP dist | var |
| iters | | ($\times 10^{-6}$) | | ($\times 10^{-6}$) | | ($\times 10^{-6}$) |
| 1 | 0.3622 | 4.176 | 0.4881 | 30.34 | 0.4715 | 3690 |
| 3 | 0.1926 | 1.15 | 0.4788 | 18.7 | 0.4628 | 91.95 |
| 5 | 0.1605 | 0.7327 | 0.473 | 12.08 | 0.4274 | 62.59 |
| 7 | 0.1353 | 0.5818 | 0.4613 | 3.844 | 0.4241 | 69.01 |
| 9 | 0.138 | 0.463 | 0.4576 | 7.981 | 0.4171 | 61.87 |
| 11 | 0.1302 | 0.4178 | 0.4618 | 6.183 | 0.4063 | 32.29 |
| 13 | 0.1281 | 0.3804 | 0.4667 | 9.817 | 0.393 | 34.92 |
| 15 | 0.1327 | 0.3284 | 0.4662 | 8.779 | 0.4064 | 37.01 |
| 17 | 0.1157 | 0.2744 | 0.4694 | 5.663 | 0.3964 | 32.38 |
| 20 | 0.1335 | 0.2167 | 0.467 | 0.001259 | 0.3942 | 7.014 |

Table 4. Convergence results of the PMC algorithm when using the isotropic Ward reflectance model. Average statistics are shown over the 20 iterations of the PMC algorithm. The image size is 100x100 pixels. The table is populated with average values over 100 runs.

| | |
|-----|---|
| C | $J_C(p_1, p_2) = \frac{1}{2} \alpha_1 \alpha_2 (\mu_1 - \mu_2)^T [\alpha_1 \Sigma_1 + \alpha_2 \Sigma_2]^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \log \frac{ \alpha_1 \Sigma_1 + \alpha_2 \Sigma_2 }{ \Sigma_1 ^{\alpha_1} \Sigma_2 ^{\alpha_2}}$ |
| B | $J_B(p_1, p_2) = \frac{1}{8} (\mu_1 - \mu_2)^T [\frac{1}{2} (\Sigma_1 + \Sigma_2)]^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \log \frac{\frac{1}{2} \Sigma_1 + \Sigma_2 }{ \Sigma_1 ^{1/2} \Sigma_2 ^{1/2}}$ |
| KL | $J_R(p_1 p_2) = \frac{1}{2} (\mu_1 - \mu_2)^T \Sigma_2^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \log \frac{ \Sigma_2 }{ \Sigma_1 } + \frac{1}{2} \text{tr}[\Sigma_1 \Sigma_2^{-1} - I_d]$ |
| SKL | $J_D(p_1, p_2) = \frac{1}{2} (\mu_1 - \mu_2)^T (\Sigma_1^{-1} + \Sigma_2^{-1}) (\mu_1 - \mu_2) + \frac{1}{2} \text{tr}[\Sigma_1^{-1} \Sigma_2 + \Sigma_2^{-1} \Sigma_1 - 2I_d]$ |
| PF | $J_P(p_1, p_2) = [(2\pi)^d 2\Sigma_1]^{-1/2} + [(2\pi)^d 2\Sigma_2]^{-1/2} - 2[(2\pi)^d] \Sigma_1 + \Sigma_2 ^{-1/2} \cdot \exp\{-\frac{1}{2} (\mu_1 - \mu_2)^T (\Sigma_1 + \Sigma_2)^{-1} (\mu_1 - \mu_2)\}$ |
| M | $J_M(p_1, p_2) = (\mu_1 - \mu_2)^T \Sigma (\mu_1 - \mu_2)$ |

Table 5. Some analytical expressions for probabilistic distances between Gaussian probability density functions, where $0 \leq \alpha_1, \alpha_2 \leq 1$ and $\alpha_1 + \alpha_2 = 1$. $|\Sigma|$ indicates the determinant of Σ . I_d is the identity matrix.

similarity coefficient with the current material that is higher than the lowest similarity coefficient among the intra-class members of the current material. The ‘‘WNN’’ statistic, which is an abbreviation for ‘‘Worst Nearest Neighbour’’, indicates the number of cases out of the 48 surfaces where there is some extra-class material having a higher similarity (lower probabilistic distance) to the material than the distance from the current material to any one of its intra-class materials. The ‘‘NN1’’ statistic is similar to this: it indicates the number of materials for which the best match (nearest neighbour) is an extra-class match and is better than all intra-class matches. The ‘‘NN1’’ statistic was used for classifiability testing in [18].

11 Conclusion

The PMC algorithm for MRF parameter estimation has been shown to work, and has reasonable convergence behaviour to correct BRDF parameters when tested in a BRDF parameter estimation setting, in the case of synthetic data. The posterior distributions on BRDF parameters were

| | C | B | KL | SKL | PF |
|------|-----|-----|-----|-----|-----|
| Miss | 157 | 155 | 134 | 119 | 198 |
| WNN | 39 | 38 | 37 | 37 | 47 |
| NN1 | 22 | 22 | 19 | 15 | 26 |

Table 6. Classification error statistics for the 48 surfaces, using the anisotropic Phong reflectance model with PMC developed particle sets. The statistics are shown for the Chernoff distance (labelled C, with α_1 arbitrarily set to 0.7), the Bhattacharyya distance (B), the Kullback-Leibler Divergence (KL), the Symmetric KL Divergence (SKL), and Patrick-Fisher distance (PF).

| | M | MAP | EMD | LM512 |
|------|-----|-----|-----|-------|
| Miss | 160 | 174 | 161 | 412 |
| WNN | 38 | 39 | 37 | 48 |
| NN1 | 24 | 25 | 23 | 38 |

Table 7. Classification error statistics for the 48 surfaces, using the anisotropic Phong reflectance model with PMC developed particle sets. The statistics are shown for the Mahalanobis distance (M), Euclidean distance between the MAP sample in each sample set (MAP), the Earth Mover’s Distance (EMD) and classification based on the MAP sample obtained after the final iteration of the PMC method. Also included in the last column, are results using the distances between the best results of multiple-seed LM iterations with 512 seeds, evenly spaced on the parameter space.

shown to be reasonably good features when used as features in a classification framework.

Acknowledgements

The authors are grateful for the financial support given by the National Research Foundation of South Africa, and Anglo Platinum Ltd and Rio Tinto via the Centre for Minerals Research at the University of Cape Town.

References

- [1] Michael Ashikhmin and Peter Shirley. An Anisotropic Phong Light Reflection Model. *Technical Report*, 2000.
- [2] J. Besag. Statistical Analysis of Non-Lattice Data. *The Statistician*, 24(3):179–195, 1975.
- [3] O. Cappé, A. Guillin, J. Marin, and C. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.
- [4] R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. Minimum variance importance sampling via Population Monte Carlo. *CMAP, Ecole Polytechnique, Palaiseau*, 2005.

| | C | B | KL | SKL | PF |
|------|-----|-----|----|-----|-----|
| Miss | 117 | 116 | 98 | 80 | 165 |
| WNN | 42 | 41 | 38 | 38 | 47 |
| NN1 | 23 | 24 | 18 | 15 | 26 |

Table 8. Classification error statistics for the 48 surfaces, using the anisotropic Ward reflectance model with PMC developed particle sets. The statistics are shown for the Chernoff distance (labelled C, with α_1 arbitrarily set to 0.7), the Bhattacharyya distance (B), the Kullback-Leibler Divergence (KL), the Symmetric KL Divergence (SKL), and Patrick-Fisher distance (PF).

| | M | MAP | EMD | LM512 |
|------|-----|-----|-----|-------|
| Miss | 121 | 157 | 135 | 241 |
| WNN | 41 | 42 | 39 | 42 |
| NN1 | 26 | 27 | 25 | 33 |

Table 9. Classification error statistics for the 48 surfaces, using the anisotropic Ward reflectance model with PMC developed particle sets. The statistics are shown for the Mahalanobis distance (M), Euclidean distance between the MAP sample in each sample set (MAP), the Earth Mover’s Distance (EMD) and classification based on the MAP sample obtained after the final iteration of the PMC method. Also included in the last column, are results using the distances between the best results of multiple-seed LM iterations with 512 seeds, evenly spaced on the parameter space.

- [5] Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical Methods for Shape from Shading: A survey with Benchmarks. *CVIU*, 2004.
- [6] Yukito Iba. Population Monte Carlo algorithms. *Transactions of the Japanese Society for Artificial Intelligence*, 16(2):279–286, 2000.
- [7] K. B. Laskey and J. Myers. Population Markov Chain Monte Carlo. *Machine Learning*, 50:175–196(22), 2003.
- [8] J. S. Liu, F. Liang, and W. H. Wong. A theory for dynamic weighting in Monte Carlo computation. *Journal of the American Statistical Association*, 96:561–573, 2001.
- [9] Jun Liu, Lei Wang, and Stan Z. Li. MRMR Texture Classification and MCMC Parameter Estimation. *Vision Interface, Canada*, 1999.
- [10] M. Louw and F. Nicolls. A Spatially Multiresolution, MRF optimization based approach to the Shape from Shading problem. In *Proc. 7th IASTED Int. Conf. Visualization, Imaging, and Image Processing*, 2007.
- [11] M. Louw, F. Nicolls, and D. Bradshaw. A Loopy Belief Propagation approach to the Shape from Shading problem. In *Proc. 2nd Int. Conf. on Computer Vision Theory and Applications*, 2007.



Figure 4. Surfaces 1 to 4 of twelve used to test the BRDF based classification power of the PMC developed sample sets. Each material was reconstructed and its BRDF calculated four times using our PMC based parameter estimation method.



Figure 5. Surfaces 5 to 8 of the twelve surfaces, continued from Fig. 4

- [12] S. N. MacEachern and M. Peruggia. Importance link function estimation for Markov Chain Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 9:99–121, 2000.
- [13] C. P. Robert and G. Casella. *Multiple Imputation for Nonresponse in Surveys*. John Wiley, New York, 1987.
- [14] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 1999.
- [15] M. F. Tappen and W. T. Freeman. Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters. *International Conference on Computer Vision*, 2003.
- [16] Lei Wang, Jun Liu, and Stan Z. Li. MRF Parameter Estimation by MCMC Method. *Pattern Recognition*, 33:1919–1925, 2000.
- [17] Gregory J. Ward. Measuring and Modeling Anisotropic Reflection. *SIGGRAPH*, 1992.
- [18] Shaohua Kevin Zhou and Rama Chellappa. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel Hilbert space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.



Figure 6. Surfaces 9 to 12 of the twelve surfaces, continued from Fig. 5