A Population Monte Carlo approach to estimating parametric Bidirectional Reflectance Distribution Functions through Markov Random Field parameter estimation

Markus Louw
Thesis presented for the degree of
Doctor of Philosophy in Engineering in Electrical Engineering
at the Department of Electrical Engineering
University of Cape Town

Cape Town, January 2009

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Doctor of Philosophy of Engineering in Electrical Engineering at the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author . . . .

Cape Town
December 3, 2009

Abstract

In this thesis, we propose a method for estimating the parameters of a parametric bidirectional reflectance distribution function (BRDF) for an object surface.

The method uses a novel Markov Random Field (MRF) formulation on triplets of corner vertex nodes to model the probability of sets of reflectance parameters for arbitrary reflectance models, given probabilistic surface geometry, camera, illumination, and reflectance image information. In this way, the BRDF parameter estimation problem is cast as a MRF parameter estimation problem.

We also present a novel method for estimating the MRF parameters, which uses Population Monte Carlo (PMC) sampling to yield a posterior distribution over the parameters of the BRDF.

This PMC based method for estimating the posterior distribution on MRF parameters is compared, using synthetic data, to other parameter estimation methods based on Markov Chain Monte Carlo (MCMC) and Levenberg-Marquardt nonlinear minimization, where it is found to have better results for convergence to the known correct synthetic data parameter sets than the MCMC based methods, and similar convergence results to the LM method.

The posterior distributions on the parametric BRDFs for real surfaces, which are represented as evolved sample sets calculated using a Population Monte Carlo algorithm, can be used as features in other high-level vision material or surface classification methods.

A variety of probabilistic distances between these features, including the Kullback-Leibler divergence, the Bhattacharyya distance and the Patrick-Fisher distance is used to test the classifiability of the materials, using the PMC evolved sample sets as features. In our experiments on real data, which comprises 48 material surfaces belonging to 12 classes of material, classification errors are counted by comparing the 1-nearest-neighbour classification results to the known (manually specified) material classes. Other classification error statistics such as WNN (worst nearest neighbour) are also calculated.

The symmetric Kullback-Leibler divergence, used as a distance measure between the PMC

developed sample sets, is the distance measure which gives the best classification results on the real data, when using the 1-nearest neighbour classification method.

It is also found that the sets of samples representing the posterior distributions over the MRF parameter spaces are better features for material surface classification than the optimal MRF parameters returned by multiple-seed Levenberg-Marquardt minimization algorithms, which are configured to find the same MRF parameters. The classifiability of the materials is also better when using the entire evolved sample sets (calculated by PMC) as classification features than it is when using only the maximum a-posteriori sample from the PMC evolved sample sets as the feature for each material.

It is therefore possible to calculate usable parametric BRDF features for surface classification, using our method.

Acknowledgements

I am very grateful to the following people, without whom this project would not have been possible:

- Dr. Fred Nicolls

- Prof. Dee Bradshaw

- Prof. Gerhard de Jager

I would also like to thank Gordon Forbes and Sam Morar for assisting with the data gathering and discussions about froth imaging and feature extraction. I am also grateful to my family members, and the international happy campers at 38 Long St.

List of Figures

List of Tables

Contents

Glossary of Symbols and Abbreviations

In this thesis, the following abbreviations are used:

- AGMMP-Average Gaussian Mixture Model Probability

- BRDF-Bidirectional Reflectance Distribution Function

- EMD-Earth Mover's Distance

- LM-Levenberg Marquardt

- PMC-Population Monte Carlo

- MRF-Markov Random Field

- MCMC-Markov Chain Monte Carlo

- NN1-First nearest neighbour

- KL-Kullback-Leibler

- MAP-Maximum-a-posteriori

- PF-Particle Filtering

- PPF-Partitioned Particle Filtering

- SFS-Shape from Shading

- SVM-Support Vector Machine

- WNN-Worst Nearest Neighbour

Chapter 1

Introduction

This chapter describes the machine vision problem which the parametric BRDF estimation algorithm we have designed is intended to solve. It also describes the series of machine vision methods we have used to solve the parametric BRDF estimation problem, in this thesis.

1.1   Control of froth flotation cells

In the South African mining industry, after the mineral ore extraction and comminution but before the smelting process, there is the flotation process. The mineral ore is ground to dust (comminuted), mixed into a solution, and air is bubbled through the liquid. The minerals which are the desired product for smelting attach themselves to the bubbles, and are extracted from the flotation cell froth surface, where the minerals have been concentrated.

The flotation process is an active area of research, since it promises to improve the profitability of the mining process significantly. At present, human flotation cell operators are responsible for optimizing the froth surface yield, but since they make relatively simple decisions based only on the appearance of the froth at any point in time, attempts are being made to partially automate/facilitate this process using machine vision techniques. To this end, research is being done to extract significant features from video footage of the froth so that ultimately the control loop may be closed automatically, allowing automatic operation and control of the flotation cells without human intervention. Previous research in extracting features for froth classification includes [34] and [35].

1.2   Problem and hypothesis formulation, goals and methodology

Our primary hypothesis is that it is possible to extract a meaningful probability density function over the parameters of any parametric bidirectional reflectance distribution function (BRDF), given a probabilistic dense disparity field or elevation map of the object surface, a reflectance map (image) of the surface, and light source and camera information, by formulating it as a Markov Random Field (MRF) parameter estimation problem. Our secondary hypothesis is that such posterior distributions on parametric BRDFs can be used as features for material surface classification. We make certain simplifying assumptions regarding light sources, estimating them from the physical geometry of the problem rather than calculating them algorithmically.

To calculate a parametric BRDF for a surface, it is necessary to create a three dimensional reconstruction for the material surface in question, using available machine vision techniques. The techniques implemented in this thesis that are used to achieve a reasonable quality of dense stereo reconstruction for the froth surfaces, include the fields and applications of structured lighting, sparse point correspondence estimation, camera calibration, and dense stereo reconstruction through Markov random field energy minimization (using Belief Propagation algorithms).

The purpose of obtaining the reflectance characteristics of the material surfaces is to use them as features to classify the surfaces, grouping them into materials which have similar BRDFs.

However, noise is introduced in the estimation of the camera, lighting parameters, scene geometry, and by inaccuracies in the surface reconstruction. Additionally, depending on the BRDF parameterisation used, certain parameter sets generate very similar-looking surfaces under the same illumination configurations. An iterative nonlinear minimizer such as the Levenberg-Marquardt algorithm may thus easily be trapped in a local maximum rather than the true (global) maximum of such a posterior parametric BRDF distribution. By exploring the space of possible BRDF parameters using an iterative resampling scheme, we benefit from exploring the surface characteristic with respect to the goodness of fit of each of the tested BRDF parameter samples, while simultaneously selecting the MAP sample as the best guess for the reflectance parameter set.

A reflectance feature can therefore be used as summary data for possible correct parameter values for surface classification.

This thesis addresses the issue of uncertainty in surface geometry reconstruction. In the mainstream literature of BRDF extraction from known scene geometry, it is always assumed that the scene is known exactly. However, techniques such as reconstruction via dense stereo estimation are never exact, and the uncertainty in the derived dense stereo maps should be projected as uncertainties into the scene, and into the corresponding estimation of the BRDF parameters. These uncertainties in the scene geometries with respect to the extraction of BRDF models have to our knowledge never been considered or incor-

porated into a model/paremeter estimation scheme. In this thesis, we propose a method for including the uncertainty in reconstructed surface geometry, when calculated using dense stereo reconstruction algorithms on a stereo image pair, within our MRF parameter estimation method, through the use of a pseudolikelihood measure.

The uses for parametric BRDF data, which is calculated by estimating corresponding MRF parameters are as follows:

- Sample sets representing the BRDF/MRF parameters may be used as features for surface classification

- Estimates for the MRF parameters can be used in shape from shading algorithms which use MRF optimization given MRF/BRDF parameters ([73], [72], [71])

- A stereo reconstruction scheme could use MRF/BRDF parameter estimates for improved reconstruction

- The MRF/BRDF parameters could be used in an iterative reconstruction/BRDF parameter estimation scheme

- Advances in MRF parameter estimation methods could be applied to this method for BRDF estimation, once the framework for casting BRDF estimation into MRF estimation is established.

It is worth noting that one of our goals in creating a framework where BRDF based MRF potential terms are used is to obviate the need for using pairwise MRF interaction terms in stereo reconstruction. These pairwise MRF potential terms are often defined in an ad-hoc manner for smoothing of disparity maps to enforce smoothness constraints on the object which is being reconstructed. MRF potential terms on point triplets based on BRDF parameters could provide a less artificial way of doing such smoothing. Although this is one potential use for BRDF based MRF potential functions, this particular use is not explored in this thesis.

Now that some possible uses for such a parametric BRDF model for material surfaces have been described, the novel theoretical innovations presented in this thesis may be summarized as follows:

- The estimation of parametric BRDF model parameters within a Markov random field pseudolikelihood framework with the explicit incorporation of the uncertainties in the reconstructed scene geometry

- The use of Population Monte Carlo sampling and Dynamically weighted MCMC sampling for MRF parameter estimation

Figure 1.1: A froth flotation cell, with a video camera and digital light projector visible in the scene. The froth surface in the flotation cell is one of the object surfaces which we analyse and classify according to its BRDF characteristics.

- A new similarity measure for comparing sample sets which represent posterior distributions, called the namely the Average Gaussian Mixture Model Probability

- A new framework for embedding shape from shading in a MRF MAP estimation problem (originally proposed by us in [73])

- A new framework for casting the BRDF parameter estimation problem as a MRF parameter estimation problem

- The use of posterior distributions on MRF/BRDF parameters as features in a classification framework.

This is the first classification algorithm (to our knowledge) that uses posterior distributions in BRDF parameter space as features for classification and it is the first time (to our knowledge) that the same equipment (namely a digital light projector) has been used both for structured lighting based dense stereo reconstruction and also as the light source in a BRDF extraction framework. An example of a flotation cell is shown in Fig. 1.1, where some of the equipment required to create a dense stereo surface reconstruction (video cameras and digital light projector) is also visible.

The rest of this thesis is divided into the following chapters, the contents of which are summarized as follows:

In chapter two we review the relevant literature in the fields of BRDFs, Markov random field (MRF)

parameter estimation, the Markov Chain Monte Carlo (MCMC) sampling methods, particle filtering, and Population Monte Carlo (PMC) methods.

In chapter three we describe the two parametric BRDF models which our algorithm is tested on, namely the isotropic Ward model, and the anisotropic Phong model.

In chapter four we describe the MRF formulation applied to reflectance modelling. We describe how the BRDF parameter estimation problem may be posed as a MRF parameter estimation problem, and why this is appropriate given the data we have used to reconstruct our material surfaces. Also described is the pseudolikelihood function which is used to weight the samples of the probability distributions in the BRDF parameter space.

In chapter five we describe importance sampling, the particle filter, partitioned particle filter, Markov Chain Monte Carlo (MCMC) sampler, dynamically weighted MCMC sampler, and Population Monte Carlo (PMC) sampler. We also describe the types of importance and softening functions we use to derive the posterior distributions on the parametric BRDF models.

In chapter six we describe the method we used to calculate a probabilistic 3-D surface reconstruction, which is used as input data for the calculation of the parametric BRDFs for surfaces. The pseudolikelihood function developed in chapter four is adjusted to incorporate the uncertainty in the dense stereo reconstruction.

In chapter seven we report results of our algorithms operating on synthetic data. The algorithms tested for calculating the BRDF parameters, using our MRF parameter estimation framework, are the Levenberg-Marquardt (LM) nonlinear optimizer, the dynamically weighted MCMC sampler, and the PMC sampler.

In chapter eight we test the PMC sampler on real data, using the Ward and anisotropic Phong models. The data are collected using a digital light projector and a stereo camera rig. Forty-eight material surfaces are extracted using this setup, and the posterior distributions in the parameter space are used as features in intra and extra-class similarity matrices. It is found that the posterior distributions on the BRDF parameter spaces are reasonable candidates for features for material surface classification.

In chapter nine we conclude the thesis and provide a discussion of our results.

The appendices are summarized as follows:

- Appendix A contains a brief description of sampling from a multinomial distribution,

- Appendix B contains a description of some pixel similarity measures which can be used in dense stereo reconstruction,

- Appendix C contains the mathematical formulation of the thin plate spline method of interpola-

tion, which we used to generate synthetic data,

- Appendix D contains a description of nonlinear optimization methods, particularly the Levenberg-Marquardt method, which was tested for convergence performance against the Population Monte Carlo method to estimate Markov Random Field parameters,

- Appendix E contains tables of similarity measurements between different real material surfaces, using different probabilistic distance measures operating on the sample sets, which are calculated using the Population Monte Carlo algorithm.

Chapter 2

Literature Review

This chapter reviews some of the relevant literature in the field of BRDF extraction, Population Monte Carlo, Markov Chain Monte Carlo and iterative Bayesian sampling methods, and some of the established methods for Markov Random Field parameter estimation. These are the core components in the novel aspects of the BRDF estimation methodology. The other software modules required to calculate the reflectance parameters from these probabilistic surfaces include everything required to create a probabilistic dense surface reconstruction, such as camera calibration, rectification, sparse feature correspondence resolution and dense stereo reconstruction algorithms.

2.1 BRDF parameterisations and extraction

This section examines some previous attempts at estimating BRDFs and object reflectance characteristics.

In [115], Torrance and Sparrow do a study in interpreting the reflectance of surfaces via an analysis of the surface roughness. Their reflectance model assumes that the surface consists of small, randomly oriented, mirror-like facets. Specular reflection occurs against these facets, and diffuse reflection is due to multiple reflections and internal scattering between the facets. Shadowing and masking of adjacent facets is included in the model. The model they derive from their assumptions predicts the off-specular peak phenomenon, which causes an increase in reflection as the angle of incidence increases. The model holds well for many metallic and non-metallic surfaces.

In [33], the concept of the Bidirectional Reflectance Distribution function is introduced. The BRDF is a distribution function, which relates the irradiance incident from one given direction to its contribution to the reflected radiance in another direction. The work gives much of the currently used nomenclature including concepts, terms, symbols, and units, for categorizing and specifying reflectance quantities

for a variety of different beam configurations (both incident and reflected beams). These concepts are defined and interrelated in terms of the BRDF.

The imaging gonioreflectometer is introduced in [122], and the Ward BRDF model is presented. The model has three parameters: a diffuse reflectance coefficient, a specular reflectance coefficient, and a surface roughness parameter which is the standard deviation of the surface slope. The model is also extended to surfaces with uncorrelated slope distributions along orthogonal directions along its surface.

In [86], the Lambertian reflectance model is generalized (extended). This model accounts for such radiometric phenomena as masking, shadowing and surface interreflections. A surface is modelled as a collection of symmetric "V-cavities", each of which has two opposing facets (this is the same surface model as used in [114]). In [86], each facet is assumed to be Lambertian. A more complex diffuse reflectance term is derived, which can provide reasonably good reflectance models for a variety of rough surfaces, without including specular reflection in the BRDF at all.

Sato et al. in [106] present a method for extracting the geometry and some reflectance characteristics of everyday objects from images. The geometry of the surface of the object is first found by integrating range data, then each component of the reflectance (diffuse and specular) is separated from the image data (which consists of images of the object taken from different locations). The specular component is estimated by identifying suitable points on the object's surface, where there are clearly visible specular highlights.

Rusinkiewicz in [102] develops a method for decomposing BRDF data into re-parameterized data, transformed into functions of the halfangles and a difference angle, rather than the usual angle of incidence and reflection. It is shown that this transformed data is more compact (requiring less storage), due to the nature of the features of standard BRDFs, where specular lobes and retro-reflective peaks are often aligned with the newly transformed coordinate axes.

In [99] and [98], Dror et al. develop an algorithm for learning relationships between surface reflectance and certain features from a single image, by exploiting the statistical regularities in the spatial structure of real world illumination, which translate into relationships between surface reflectance and statistical features of the image. This is done in an attempt to facilitate visual material recognition, to improve reconstruction algorithms, and to overcome the limitations of shape from shading algorithms (which assume known surface reflectance properties), and of classical stereo and motion estimation algorithms (which assume Lambertian reflectance). The material recognition algorithm is verified using a set of photographs of physical spheres of known reflectance.

In [70], a method for calculating bidirectional texture functions (BTFs) from a surface is developed. BTFs are six-dimensional functions that model a surface texture as a function of illumination and viewing directions (the additional dimensions are for texture data). In [70] it is shown how to develop novel views given data points of the BTF at various known values.

In [2], Ashikhmin et al. propose an anisotropic Phong BRDF model which conserves energy, is reciprocal, and has a non-Lambertian diffuse reflectance term. It is also suited for use in Monte Carlo renderers. We selected this anisotropic Phong model as one of the models to use for testing our parameter estimation algorithm on synthetic data, due to its simplicity and its performance characteristics.

In [79], a method which simultaneously estimates the BRDF (Ward parameterisation) and the texture of a surface is developed. An image is acquired where half the image is of a plane of a reference material, and the other half is of the material under investigation. The line separating the two materials must form a plane with the light source and camera locations. First a least squares approach is used to find the BRDF Ward parameters (as in [58]), after which the texture information is expressed as the difference in the measured BRDF values at a region in the image, and the expected value given the calculated Ward reflectance parameters.

In [56], a new reflectance model is developed which posits that any BRDF is in fact a simplification of the Bidirectional Surface Scattering Distribution Function (BSSRDF). The BSSRDF allows for light transport between any two points on the surface of the material. Integration of the BSSRDF, single scattering, diffusion approximation, and texture handling are all incorporated in a ray tracing framework using Monte Carlo sampling for integration.

In [65], an algorithm is developed which robustly detects the different materials of real objects, and fits an average BRDF to each of them. This is done by clustering the acquired surface reflectance samples into groups of similar materials, using the Lafortune [61] parametric BRDF model.

In [94], Ramamoorthi introduces a signal processing framework for BRDF estimation, which describes the reflected light as a convolution of a lighting function and the object's BRDF. This is expressed mathematically as the product of spherical harmonic coefficients of the BRDF, with the lighting. The inverse rendering problem is then cast as a deconvolution problem.

In [40], a method is developed for estimating the shape and reflectance of a surface from a small number if images. The surface is represented by a set of discrete cosine transform (DCT) coefficients. The algorithm alternates between doing steepest descent on the DCT coefficients (using a photoconsistency measure), updating the surface Lambertian albedo characteristics (per point) using a linear least squares method, solving numerically (Newton's method) for the light source directions, and solving (Newton's method) for the specular reflectance and surface roughness.

In [117], images of an opaque object are taken with different polarizers over the camera lens. Using the fact that specular light tends to be polarized, and diffuse not, Umeyama uses an Independent Component Analysis algorithm to calculate the diffuse and specular reflectance components of the object's surface reflection.

In [131], a View Independent Reflectance Map (VIRM) is proposed, which is calculated from calibrated

images of the object. The VIRM is a simplified version of the Torrance-Sparrow model, which only models a single material under distant lighting conditions. An iterative method solves for the geometry and the VIRM using a photoconsistency measure. In [94], it is noted that there is an inherent ambiguity when trying to recover both the lighting and the BRDF, since a blurred light source and a sharp BRDF give the same results as a sharp light source and a low-pass BRDF. [131] takes advantage of this by replacing specular reflection with the combination of a circular low pass filter and a perfect specular (mirror) reflectance at each point on the surface. It is essentially a simplified model for reflectance under the assumptions that all parts of the surface are of the same material, with distant lighting, and that no self shadowing or self-reflection occurs. The advantage of this method is that many of the model parameters may be calculated linearly.

In [126], four parametric BRDF models are compared, namely those of Phong, Ward, He-Torrance and Lafortune. In this comparative study, different real world objects have their reflectances measured using a gonioreflectometer. Each model is then fitted via nonlinear optimization with the L^2 norm of the BRDF error, weighted by the cosine of the reflectance angle. It was found that the Ward and Phong models perform best when there is less dependence of the reflectance on the angle of incidence. The He-Torrence model works well for some surfaces, and the Lafortune model performs well on all the surfaces.

In [43], a method is developed for calculating the BRDF and the shape of a material from a set of images. The algorithm iterates between optimization of BRDF parameters, computing the surface normals and material weight maps, and enforcing the integrability of the resultant normal map (by solving a Poisson equation). The material weights indicate the relative amount of each of two "fundamental" materials which comprise the surface subregion.

In [119], the BRDF is estimated by sampling frontier points on the object contour, and under assumptions of fixed or varying illumination, the parameters are found by formulating the problem as one of blind deconvolution, and using a minimization technique. Frontier points are points on the object such that the plane formed by the point and the two camera centers is tangential to the object. The algorithm iteratively re-estimates the illumination and surface reflectance given several images of the object and frontier points. In the fixed-illumination moving-camera setting, the algorithm iterates between fixing the BRDF parameters and recovering the light distribution by using the Lucy-Richardson algorithm, then fixing the light distribution and estimating the BRDF (with gradient descent). In the fixed camera varying lighting setting, a cost function is minimized with gradient descent. For more precise geometry recovery, the algorithm of [51] is used.

In [12], a variational method is developed which alternates between estimating the surface geometry (using a photoconsistency measure) and the reflectance parameters. The initial geometry is estimated using a shape-from-silhouette method, which takes as inputs the captured images of the object rotating on a turntable, and the projection matrices for each image.

In [64], an MCMC ray-tracing rendering framework is developed which uses importance sampling to focus samples on the ray-traced paths which carry higher amounts of energy. It is assumed that parametric BRDFs for all surfaces are known, and that the direction of each ray from the surface point to the camera is known. The importance sampling occurs over the incident light direction from the light sources onto the material. To facilitate importance sampling over the incident directions, the BRDF for the surface is factorized and reparameterised. This results in a more efficient MCMC ray tracing renderer. In some sense there is a similarity between this paper and our work: whereas in [64] it is assumed that the direction of the incident light is unknown and the parametric BRDF is known exactly, but the incident light directions are many and must be integrated over with MCMC sampling. In contrast, we assume that the direction vectors of the incident and outgoing light path are known exactly, and a probabilistic representation for the parametric BRDF is to be found using sampling methods.

2.2 MRF parameter estimation

In this section we review some of the history of MRF parameter estimation over both continuous and discrete MRFs, as well as some of the more recent methods. This serves as background to the novel MRF parameter estimation method which we develop in this thesis.

Besag introduced the coding method in [6], where nodes on a lattice are separated into disjoint sets, called codings. The conditional densities within a coding are independent. Two codings are needed for a 4 neighbourhood system, and four codings for an 8 neighbourhood system. Each of the Maximal Likelihood MRF parameter estimates are based only on the information from one coding. The estimates from each coding may be averaged (one method) to estimate the overall best MRF parameters.

In [7], Besag introduced the pseudolikelihood function. This method incorrectly assumes that lattice node labels are independent, but allows us to avoid calculating the partition function when estimating the probability of a labelling on a lattice for a particular set of MRF parameters. In [8] the pseudolikelihood parameter estimation technique, is further tested. We selected this method to derive probability estimates for BRDF model parameters as is seen in later chapters. In [9], a parameter estimation method for MRFs on images is outlined, which iterates between making an Iterative Conditional Modes (ICM) estimate of the correct label for each pixel, and using these label estimates to maximize the MRF parameters using a pseudolikelihood measure.

The roots of Mean Field approaches to MRF parameter estimation (e.g. [37] and [136]) are in statistical mechanics. The Mean Field estimation approach takes a similar form to the pseudolikelihood approach, but uses a mean field approximation for the values of the neighbouring random variable node values (labels) instead of the neighbouring site labels. Different mean field algorithms estimate the mean field approximations differently, for example by iteratively averaging the neighbouring pixels intensities

[37], or by using a saddle point approximation [136].

In [52], a method of estimating Gibbs hyperparameters in a positron emission tomography (PET) setting is developed. It relies on simultaneously deriving a posterior distribution on both the state values on the lattice and on the parameters of the MRF. Using samples from this posterior, it is simple to estimate quantities like the mean of the image (lattice), averaged over the (hyper)parameter uncertainties. Note that in this way it is similar to our method, which estimates reflectance parameters, given a per-lattice-site range map, with associated certainties for each height, for each point on the lattice (this probabilistic range map is derived from a posterior distribution on the pixel disparities using a stochastic dense stereo method). These parameters may also be estimated from the samples of this posterior distribution. The partition function, required to calculate the probability for each sample set of image realizations and MRF parameters is estimated using Geyer's reverse logistic regression method [41], which is an improvement on Bennett's ratio method [4], which is itself an advance on the importance sampling method of [42].

In [46], the GMRF parameter estimation via the histogramming method of [24] which uses lattice codings is investigated, and the relationship between this and the Maximum Likelihood method is explored. A method for reducing bias in the histogramming method is also proposed.

In [105], a Maximum Likelihood method is developed for estimating the shape and temperature parameters of a Generalized Gaussian Markov Random Field (GGMRF), although the approach may be extended to general MRFs. The method relies on efficient estimation of the partition function. This is done by estimating the expectation value of the derivative of the partition function and integrating it by fitting a second order spline to it. The expectation value is estimated using a Metropolis-Hastings style algorithm, and a careful study of the prior distribution values involved has allowed for the construction of a proposal distribution which is tuned to the problem and improves the estimates derived from the sampling procedure.

In [17], the idea developed in [25] and [46] is extended. The authors seek to produce a general method for parameter estimation, using image analysis to model pixel (node) interactions on a lattice only as a case study. A grey level image with G levels is divided into G binary planes. Each plane is modelled using a stochastic spatial model, from which the MRF parameters are estimated using a weighted least squares method.

In [104], an efficient scheme for estimating the shape parameter for the generalized Gaussian MRF (GGMRF) is proposed. To do this, an offline numerical computation of the log of the partition function is done. The domain of application for this technique is tomography, and the parameters depend also on unknown image data. An EM method was designed to do the parameter estimation with missing data. It is also shown that the temperature parameter for the GGMRF has a simple closed form solution.

In [69], a novel stochastic approach is used for parameter estimation in multiresolution MRFs (MRMRF).

Images are decomposed by wavelets and the MRF parameters are estimated using the subbands as input texture data. The idea is that these estimated MRF parameters may be used to classify the texture of the image. The method proposed for estimating the MRF parameters is a Markov Chain Monte Carlo (MCMC) method, using the pseudolikelihood of the image given the proposed MRF parameters as a probability, for each sample in the chain. These samples yield a posterior distribution on the probability of the MRF parameters. This paper was one of the main inspirations for our MRF parameter estimation method, which uses a Population Monte Carlo sampler, rather than an MCMC sampler to estimate MRF parameters.

In [95], 3-D Gaussian MRF models are used for texture segmentation in 3-D MRI images. These MRF models parameters are estimated by solving a system of standard normal equations for each of a set of voxels from MRI texture data.

In [59], an MRF segmentation model which combines colour and texture features (given by a set of Gabor filters) is introduced. The parameters for the model (which include Gaussian Mixture Model coefficients) are estimated with an EM algorithm, to handle incomplete training data (unknown labelling of input pixels in the unsupervised learning case).

In [85], a "homotopy calculation method" is developed for image restoration under specific independent Gaussian noise and neighbourhood system assumptions. The method alternates between updating MRF parameters and updating (restored) image labels.

In [121], Wainwright does an analysis a method which uses the same convex variational relaxation to construct an M-estimator to fit MRF parameters (in a general setting) and to perform approximate marginalization in the prediction (restoration/labelling) step. It is proved to be beneficial to use an inconsistent M-estimator that returns an incorrect model since the errors in the M-estimator can offset errors in the prediction step. (The prediction step may be equivalent to denoising, smoothing or interpolating new noisy data).

In [129], a method for texture analysis which combines filter theory and random fields, using a Maximum Entropy approach, is developed. The parameters for this MRF are estimated using MCMC.

In [137], a method for estimating the best MRF parameters in a dense stereo algorithm is developed. The method alternates between finding the disparity map given current MRF parameters, and optimizing the MRF parameters given the input images and the disparity map. An EM algorithm is used to estimate the MRF parameters given a disparity map, and belief propagation is used to estimate the disparity map given the MRF parameters.

In [47], Haindl presents a method for the unsupervised segmentation of colour image textures. The texture segmentation is done in the MRF parameter space. A recursive maximum pseudolikelihood parameter estimation procedure for a GMRF model is derived, and used to evaluate the segmentations.

This is analogous to our work: we ultimately seek to create "reflectance features", which are sample based posterior distributions on the parameter space of a BRDF, that are extracted from a (probabilistic) object surface.

In [142], two recursive-in-order least squares algorithms for estimating the parameters of a 2-D GMRF are presented. The first uses auxiliary vectors to find recursive forms of the parameter vector for a noncausal GMRF, the second replaces the GMRF model with a noncausal nonsymmetric model; the computational complexity is about half that of the conventional Least Squares estimator.

In [81], a method is presented for estimating the clique potentials in a pairwise GMRF on the lattice of pixels on an image. Since the energy of the GMRF will take the form of a weighted sum of Gaussian kernels, the authors use Support Vector Machine (SVM) learning (the Mean Field theory SVM learning algorithm [82]), for estimating this energy function. The potential function of each clique shape in the GMRF is modeled as a Gaussian shaped kernel. The intended domain of application for this method is image texture modelling, and the method is satisfactory for synthetic data.

There are several investigations in the literature about using the Expectation Maximization (EM) method to optimize MRF parameters. For example, [135] develops a method for blind image restoration using a coupled Markov random field. Other EM based methods for MRF parameter estimation include [138], [22], [134], [16], [38], [92], and [104].

In [135], Zhang develops an algorithm to do blind image restoration, using two coupled Markov random fields to model the original image: a compound Gauss Markov (CGM) field, and a line process. The method uses a mean field algorithm in an Expectation Maximization framework. There are two sets of model parameters which relate to the noise in the Gauss Markov field, and to the expected level of image blurring (each pixel in the observed image is treated as a locally-blurred transform of the corresponding region in the original (unrectified) image). After each set of mean field iterations, the blur parameters are estimated (as coefficients of a discrete cosine transform (DCT) on a "blurring kernel"). The noise parameters for the CGM model are not iteratively re-estimated in this algorithm, and reasonably good initial values must be given for these parameters if the algorithm is to converge correctly. Zhang points out that whereas the blur parameters may be estimated by calculating the coefficients of a DCT, the noise parameters in the CGM model would have to be estimated using a nonlinear gradient descent type algorithm. This introduces additional control parameters and may produce bad local minima. There is some similarity between this method and ours: in [135], noise parameters for an observation model and correct image data (MRF field labelling) are estimated given observed data, whereas we are dealing with the problem of parameter estimation of an MRF model, given probabilistic information on possible individual labels at points on a disparity map. However, [135] avoids doing parameter estimation on the actual potential function parameters of the MRF.

In [116], the progress with the EM mean field algorithm is continued, this time with application to

magnetic resonance imaging (MRI) data. The observed MRI volumetric intensity image is modelled as being the result of a true image with spatially interacting pixels, and spatially low frequency bias, which is the result of the operation of the MRI scanner. As in the previous example, this is solved by iteratively estimating the bias parameters and the estimate of the original image, in an Expectation Maximization manner.

In [104], a Maximum Likelihood approach is used for the recovery of the shape parameter in a Generalized Gaussian Markov Random Field (GGMRF). Recovery of the temperature parameter was shown in [16] to have a simple closed-form solution, although estimation of the shape parameter is made difficult by the need to calculate the partition function. An optimal Maximum Likelihood method is developed which calculates the logarithm of the partition function, offline. An alternative method was suggested in [93], which estimates the shape parameter through the computation of the kurtosis of differences over neighbouring pixels.

In [103], a general framework for a multi resolution non-homogeneous Generalized Gaussian Markov Random Field (GGMRF) parameter estimation method is developed. It is shown to be applicable to the domains of optic flow and PET (tomography). At each resolution (in a coarse to fine approach) the noise variance parameters at each pixel are estimated (the MRF field is non-homogeneous, so the noise parameter varies across the image, or across the data). The temperature of the MRF across the data is also re-estimated at each resolution.

2.3   Shape from shading in a stochastic MRF framework

In [23], the shape from shading problem is for the first time treated in a stochastic framework, using Markov Random Fields in a Bayesian formulation, with simulated annealing. The stochastic method outperforms a deterministic method to minimize the associated energy function, although it is much slower. A hybrid method is also proposed, which combines the deterministic and stochastic approaches, in a multi-resolution framework. Our method uses a different control point parameterisation of the surface, and solves the inverse problem (i.e. reflectance parameter estimation).

2.4   Partitioned Particle Filtering (PPF) and Importance Sampling

The bootstrap filter was developed by Gordon in [45] and [44], and forms the basis for the particle filter which is also known as the conditional density propagation (condensation) filter. The condensation (conditional density propagation) algorithm was developed in [14], to track the parametrization of a shape contour in a recursive Bayesian manner. In [74], MacCormick and Blake introduce partitioned particle filtering, which is in essence a repeated importance sampling algorithm, using a different impor-

tance sampling distribution for each partition. It may be used to refine or improve the sample locations of a sample set representing a probability distribution without introducing bias. In [75], this method is used for tracking the contours of a human hand.

In [84] it is shown that all particle filtering algorithms, Kalman filtering algorithms, and Hidden Markov Models inference algorithms are special cases of inference done on a (Dynamic) Bayesian Network (DBN).

2.5 Population Monte Carlo methods

In [63], the Population Markov Chain Monte Carlo, or simply Population Monte Carlo (PMC) sampler is introduced. This sampler uses statistical information from a group of Metropolis-Hastings samplers to calculate better proposal distributions for each of the samplers. It is found that the Population MCMC method performs better than an evolutionary algorithm, and better than a single Metropolis-Hastings sampler in a problem involving exploration of arc frequencies in networks.

The Population Monte Carlo (PMC) method is (apparently independently) also introduced in [19]. It is shown that iterated importance sampling can produce more accurate approximations to a distribution than the sequential sampling of a MCMC algorithm. The PMC method is a combination of the MCMC method (for the construction of the proposal distribution), importance sampling, SIR (sample-importance-resampling for sample equalization), and iterated particle systems. In their excellent introduction, they point out that early literature on MCMC methods attempted to dissociate the method from importance sampling, despite the commonality of sampling from a different distribution to the target distribution. Only recently have the two approaches (importance sampling and MCMC) been coupled, (see [76], [68], and the iterated particle systems of [27]). The PMC method is tested by simulating from a Gaussian Mixture Model, and an ion-channel model. The PMC sampler is found to perform much better than an equivalent Metropolis Hasting sampler. The PMC algorithm of [19] includes a resampling step of all samples at each iteration, which is not included in the PMC algorithm of [63].

One of the first iterated particle systems was the bootstrap filter [45], which propagated samples in a Bayesian manner. Additional sampling steps were introduced in [5]. The auxiliary particle filter of [90] added an auxiliary variable to improve the sampling efficiency and robustness to outliers.

In [76], a transformation called an "importance link function" is used to adjust the samples taken from the proposal distribution, to resemble more closely the target distribution. The goal of reducing the importance weight variances is achieved in three case studies which include an analysis of a case-deletion influence analysis for a linear and nonlinear model, using this method. Unfortunately the construction of the importance link functions requires a detailed understanding of both the posterior and the importance distributions.

In [54], a short cross-disciplinary survey of Monte-Carlo and importance sampling style algorithms, as well as genetic algorithms, and sampling techniques in physics systems such as lattice systems and polymer models is presented. It is found that many of the techniques related to these models may be described as Population Monte Carlo methods.

In [26], a theoretical advance is made for general PMC schemes: given a function of interest, it is found that the asymptotic variance for this function may be minimized by iteratively minimizing the parameters on a kernel of importance functions (referred to as a D-Kernel). This is relevant to variance reduction in general Monte Carlo experiments and simulations, where much effort has been put into reducing the number of samples required to represent a posterior distribution by sampling efficiently and avoiding sampling in areas where the function has uninteresting values. However, the adaptivity of this PMC model is limited to the modification of the weights of each of the kernels.

Fan [32] uses the PMC framework in a photo-realistic rendering application, where it is known that Monte Carlo integration algorithms on a per pixel rendering basis can produce physically accurate lighting models. This method requires the calculation of integrals of two types: hemispheric integrals and image plane integrals. Two PMC samplers are used in this method. The first is used for hemispheric rendering integrals, where incident light directions are sampled from some importance function. The appropriate choice of an an importance function to sample from will allow for more realistic and faster rendering if for example, the importance function allows us to ignore a light source that is occluded from a particular pixel. The second sampler in this method, is the "adaptive image plane sampler", which is used for the selection of pixels for refinement, based on perceptually weighted variance criteria. The hemispheric integral sampler allows samples to be guided towards important illumination directions, without introducing bias.

In [62], the work of [32] is continued. A Population Monte Carlo algorithm is developed that improves the rendering efficiency of their algorithm, which concentrates computation on important light source paths, in rendering a scene. The PMC energy redistribution algorithm allows for sample reuse, and the consequent propagation of important samples for calculating light transport enables more intelligent calculation of path integrals. The PMC algorithm in this context iterates over a population of light transport paths, where each path is calculated using a ray tracing algorithm.

In [18], an algorithm is developed that iteratively updates the mixture weights and parameters of a mixture of importance sampling probability density functions, to optimize the performance of an importance sampler. A Rao-Blackwellisation step is incorporated, for updating the weights and density parameters. It is also pointed out that one of the benefits of importance sampling (or Population Monte Carlo) schemes is the parallelizability, which allows it to take advantage of parallel or multiple-core architectures.

In [20], a PMC method (which the authors refer to as both Population Monte Carlo and Iterated Im-

portance Sampling algorithm) is tested on missing data problems. It is found that the PMC method outperforms the MCMC method, and can provide a good representation of distributions of interest in just a few iterations. While the selection of good importance functions is still important, the iterative parallel multi-sample nature of the PMC algorithm makes it more robust to poor choices of importance function. In fact, the PMC framework allows for the inclusion of multiple importance kernels, each suited differently to the missing-data problem. A Rao-Blackwellization step is included and offers some benefit.

Chapter 3

Parametric BRDFs for Reflectance Modelling

An early method for capturing the BRDF of a material was to use a device called a gonioreflectometer, which has a moving photometer and light source. Data points are collected by moving the light source and the photometer and repeatedly taking measurements. The imaging gonioreflectometer, an improvement which allows the device to capture an entire hemisphere of surface reflections simultaneously, was introduced in [122]. This device incorporates a hemispherical mirror and a charge coupled device (CCD) with a fish eye lens, to collect all the incident light rays. In the same paper a parametric model is introduced for anisotropic reflectance, known as the Ward model. Although no parametric model can ever capture a material's reflectance completely, the Ward model is simple and accurate, and is still used today with many variations. In [122], the model's correctness is validated using the imaging gonioreflectometer. For our parametric BRDF estimation method we explore two possible BRDF parameterisations, namely the Ward model of [122], and the anisotropic Phong model of [2]. However, our method would extend just as well to most BRDF parameterisations.

In general BRDFs are written in the form $\rho(\theta_i, \phi_i; \theta_r, \phi_r)$, where ρ is a fraction of the reflected energy, which depends on the incident angles (θ_i, ϕ_i) (between the surface normal and the light source), and the angles of reflectance (θ_r, ϕ_r), the angles between the surface normal and the camera sensing element. The azimuthal angle ϕ is taken about the surface normal at the point of incidence, and the angle of elevation θ is taken against the surface plane. Sometimes the incident and reflected angles are described in vector form, as is done in the description of the anisotropic Phong model.

A BRDF function should actually be a 5-dimensional function $\rho(\theta_i, \phi_i; \theta_r, \phi_r, \lambda)$, where the λ parameter refers to the wavelength of the light (polarization should also be included explicitly, but seldom is). The function is called "bidirectional", because the incident and reflected angles can be exchanged

for one another, and the function will return the same value (due to the physics of light). This is why backwards ray tracing, for rendering, works. With colour cameras, we could theoretically develop a reflectance estimation technique that works across three colour channels. Since this is a trivial extension, we do not explore the three colour BRDF parameterisation estimation problem, but rather work on the average intensity across colour channels. In each of the two models described here, the following notation is used:

- $\mathbf{a} \cdot \mathbf{b}$ is the scalar product of vectors $\mathbf{a}$ and $\mathbf{b}$,

- $\hat{\mathbf{n}}$ is the surface normal,

- $\mathbf{h}$ is the normalized halfway vector between the incident light direction vector $\mathbf{k}_1$ and the reflected light direction vector $\mathbf{k}_2$,

- $\mathbf{u}, \mathbf{v}$ are tangent vectors along the surface which, with $\hat{\mathbf{n}}$, form an orthonormal basis.

3.1 The Isotropic Gaussian Ward reflectance model

This section describes the isotropic Ward BRDF model, as expounded in [122]. The isotropic Ward model is given by

$$\rho_{bd,iso}(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{\rho_d}{\pi} + \rho_s \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \frac{\exp\left(-\tan^2\delta/\alpha^2\right)}{4\pi\alpha^2}, \qquad (3.1)$$

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 anisotropic Ward model is

$$\rho_{bd}(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{\rho_d}{\pi} + \rho_s \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \frac{\exp(-\tan^2\delta(\cos^2\phi/\alpha_u^2 + \sin^2\phi/\alpha_v^2))}{4\pi\alpha_u\alpha_v}, \qquad (3.2)$$

where ϕ is the azimuth angle of the half vector projected onto the surface plane (as seen in Fig. 3.1). The previous equation may be computed more efficiently by avoiding some of the trigonometric calculations, and restating Eqn. 3.2 as:

$$\rho_{bd}(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{\rho_d}{\pi} + \rho_s \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \frac{1}{4\pi\alpha_u\alpha_v} \exp\left(-2\frac{\frac{\mathbf{h}\cdot\mathbf{u}}{\alpha_u} + \frac{\mathbf{h}\cdot\mathbf{v}}{\alpha_v}}{1 + \mathbf{h}\cdot\hat{\mathbf{n}}}\right), \qquad (3.3)$$

where

$$\mathbf{h} \cdot \mathbf{u} = \frac{\sin \theta_r \cos \theta_r + \sin \theta_i \cos \theta_i}{||\overrightarrow{h}||}, \tag{3.4}$$

$$\mathbf{h} \cdot \mathbf{v} = \frac{\sin \theta_r \sin \theta_r + \sin \theta_i \sin \theta_i}{||\overrightarrow{h}||} \tag{3.5}$$

and

$$\mathbf{h} \cdot \hat{\mathbf{n}} = \frac{\cos \theta_r + \cos \theta_i}{||\overrightarrow{h}||}, \tag{3.6}$$

with

$$||\overrightarrow{h}|| = \left[2 + 2 \sin \theta_r \sin \theta_i (\cos \phi_r \cos \phi_i + \sin \phi_r \sin \phi_i) + 2 \cos \theta_r \cos \theta_i \right]^{\frac{1}{2}}. \tag{3.7}$$

The following substitutions are used for vector calculations:

$$\overrightarrow{h} = \mathbf{k}_1 + \mathbf{k}_2, \tag{3.8}$$

$$\mathbf{h} = \frac{\overrightarrow{h}}{||\overrightarrow{h}||}, \tag{3.9}$$

$$\cos(\theta_r) = \mathbf{k}_1 \cdot \hat{\mathbf{n}}, \tag{3.10}$$

$$\cos(\theta_i) = \mathbf{k}_2 \cdot \hat{\mathbf{n}}. \tag{3.11}$$

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 [122] 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 $\alpha > 0.2$.

3.2 The Anisotropic Phong model

The anisotropic Phong model, due to [2] (a more detailed derivation of this model is given in [3]), parameterizes a BRDF with four parameters, referring approximately to diffuse and spectral reflectivity and horizontal and vertical surface roughness. The model has four parameters, namely R_s and R_d, the specular and diffuse reflection coefficients, and n_u and n_v, the surface roughness coefficients. As with the Ward model, the reflectance is modeled as a sum of diffuse and specular components:

$$\rho(\mathbf{k}_1, \mathbf{k}_2) = \rho_d(\mathbf{k}_1, \mathbf{k}_2) + \rho_s(\mathbf{k}_1, \mathbf{k}_2). \tag{3.12}$$

The specular component of the reflectance function is

$$\rho_s(\mathbf{k}_1, \mathbf{k}_2) = \frac{\sqrt{(n_u + 1)(n_v + 1)}}{8\pi} \frac{(\hat{\mathbf{n}} \cdot \mathbf{h})^{n_u \cos^2 \phi + n_v \sin^2 \phi}}{(\mathbf{h} \cdot \mathbf{k}) \max((\hat{\mathbf{n}} \cdot \mathbf{k}_1), (\hat{\mathbf{n}} \cdot \mathbf{k}_2))} F(\mathbf{k} \cdot \mathbf{h}), \tag{3.13}$$

where

- n_u and n_v are the surface roughness parameters,

- ϕ is the angle between $\mathbf{u}$ and the halfvector $\mathbf{h}$ projected onto the surface (see Fig. 3.1),

- $\mathbf{k}_1$ is the vector from the surface to the light source, normalized to unit length,

- $\mathbf{k}_2$ is the vector from the surface to the camera center, normalized to unit length,

- $\rho(\mathbf{k}_1, \mathbf{k}_2)$ is the specified BRDF function,

- $\mathbf{k}$ indicates that either $\mathbf{k}_1$ or $\mathbf{k}_2$ can be used,

- $F(\cos\theta)$ is the Fresnel reflectance given an incident angle θ.

The approximation to the Fresnel reflection term is

$$F(\mathbf{k} \cdot \mathbf{h}) = R_s + (1 - R_s)(1 - (\mathbf{k} \cdot \mathbf{h}))^5. \tag{3.14}$$

The above approximation, where R_s is the material's reflectance for the normal incidence, is due to [107]. For more efficient computation, it is shown that the computation of trigonometric functions may be avoided by rewriting Eqn. 3.13 as

$$\rho_s(\mathbf{k}_1, \mathbf{k}_2) = \frac{\sqrt{(n_u + 1)(n_v + 1)}}{8\pi} \frac{(\hat{\mathbf{n}} \cdot \mathbf{h})^{\frac{n_u(\mathbf{h}\cdot\mathbf{u})^2 + n_v(\mathbf{h}\cdot\mathbf{v})^2}{1 - (\mathbf{h}\cdot\hat{\mathbf{n}})^2}}}{(\mathbf{h} \cdot \mathbf{k})\max((\hat{\mathbf{n}} \cdot \mathbf{k}_1), (\hat{\mathbf{n}} \cdot \mathbf{k}_2))} F(\mathbf{k} \cdot \mathbf{h}). \tag{3.15}$$

The diffuse term (which does not depend on n_u or n_v) for Eqn. 3.12 is given as

$$\rho_d(\mathbf{k}_1, \mathbf{k}_2) = \frac{28 R_d}{23\pi}(1 - R_s)(1 - (1 - \frac{\hat{\mathbf{n}} \cdot \mathbf{k}_1}{2})^5)(1 - (1 - \frac{\hat{\mathbf{n}} \cdot \mathbf{k}_2}{2})^5). \tag{3.16}$$

The leading factor of $\frac{28}{23\pi}$ is for total energy conservation. A diagram of the vectors in this model is shown in Fig. 3.1. The anisotropic Phong model has many advantages over the Ward model: it obeys energy conservation and reciprocity laws and allows for different reflectance characteristics in different directions (as a result of the surface roughness which has two spatial parameters associated with it). It also models Fresnel reflectance, where as the angle of incidence decreases the specularity increases. The roughness coefficients n_u and n_v have no physical meaning in the anisotropic Phong model (unlike the Ward model).

Although in the anisotropic Phong model, the diffuse term may be replaced with another one (e.g. the Lambertian reflectance function, which is wholly diffuse), the authors of [2] prefer to use an angle dependent diffuse term which balances the energy between diffuse and specular reflection for better energy conservation.

The two reflectance models described here are widely known, especially the Ward model. Demonstrating our method using each of these two methods will help in proving the genericity of our solution.
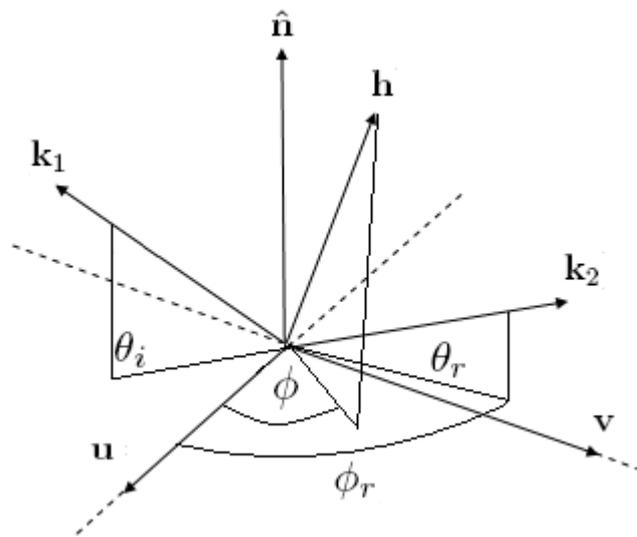
Figure 3.1: BRDF geometry.

Chapter 4

Markov random field model for Shape from Shading

The Markov random field model allows for local probabilistic interactions between nodes on a lattice. By associating high probabilities to surface geometries which are congruent with the available information about lighting, camera and surface reflectance, we are able to use the MRF framework to estimate the surface which is most likely to have created the reflectance measurements.

4.1   Markov random field framework

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 our previous research in [73], [72] and [71], where we cast the shape from shading problem into a MRF framework.

Surface geometries are calculated given the reflectance parameters of the surfaces, by minimizing local clique energy over all the individual cliques that make up the surface. This is done using Loopy Belief Propagation, and Gibbs sampling. This led us to hypothesize that the same reflectance parameters of the surface can be estimated using the same (much expanded) MRF framework, by treating a probabilistic surface reconstruction as a posterior distribution on a MRF. We now describe MRFs in general, and then our particular use of them.

If a Markov random field model is defined on a lattice of connected sites (or nodes), each of which represents a discrete random variable that can take on one of a set of possible labels, it means that the probability of a node taking on any particular label depends only on the labels of its neighbouring nodes. In particular, the sites of a MRF on a lattice $\mathbf{S}$ are said to be connected according to a neighbourhood system $\mathcal{N}$, such that

$$\mathcal{N} = \{\mathcal{N}_i | \forall i \in \mathbf{S}\}, \tag{4.1}$$

where $\mathcal{N}_i$ is the set of sites neighbouring node i. A clique is a set of nodes which are all neighbours of each other. If the conditional probability function for the label of node i given all other site labellings is equal to the probability of the site given the labels of its neighbouring nodes only, i.e. if

$$p(f_i | f_{\mathbf{S}-\{i\}}) = p(f_i | f_{\mathcal{N}_i}), \tag{4.2}$$

then the random field is a Markov random field.

Usually MRFs are described using Gibbs distributions on the lattice. If potential energy terms are defined for each clique, for each possible set of labellings on those cliques, then the probability of a particular lattice labelling can be defined as (Gibbs random field formulation):

$$p(f) = Z^{-1} \exp(-U(f)/T), \tag{4.3}$$

where Z is the normalizing partition function, T is a temperature variable, and U is the potential energy on the lattice given a realization (or instance) of labels, f. The constant

$$Z = \sum_{f \in F} \exp(-U(f)/T), \tag{4.4}$$

known as the partition function, ensures that the probability of Eqn. 4.3 is normalized. The temperature T affects the relative probabilities of high and low energy lattice labellings. As the temperature becomes higher, all node labellings become closer to having the same likelihood. Algorithms using simulated annealing rely on decreasing the temperature gradually to find the highest probability labelling (maximum a posteriori or MAP labelling) for the lattice. Often T is not used at all; similarly for the purposes of this thesis we set $T = 1$, and ignore it from here onwards.

The energy function $U(f)$ is a sum of clique potential functions $\psi_c(f)$ over the set of cliques $\mathbf{C}$ (cliques for which potential energy interaction terms are defined) on the lattice $\mathbf{S}$. This is defined as

$$U(f) = \sum_{c \in \mathbf{C}} \psi_c(f). \tag{4.5}$$

All energies must be positive to ensure correct normalization by Z. It should be clear at this point that higher energies in the potential functions correspond to lower probabilities for the labellings which produce them. The order of a clique is the number of nodes in that clique. In their original unpublished

paper [57], Hammesley and Clifford showed that any Gibbs random field is a Markov random field, and for completeness, we include a proof of this theorem. First we examine the probability at site i on a lattice, given the labels on all other sites:

$$p(f_i|f_{\mathbf{S}-\{i\}}) = \frac{p(f_i, f_{\mathbf{S}-\{i\}})}{p(f_{\mathbf{S}-\{i\}})} = \frac{p(f)}{\sum_{f_i' \in \mathcal{L}} p(f')}, \tag{4.6}$$

where $f' = \{f_1, \ldots, f_{i-1}, f_i', f_{i+1}, \ldots, f_m\}$ (i.e. all labels except f_i are fixed), m is the number of sites on the lattice, and $\mathbf{S} - \{i\}$ indicates all sites on the lattice except site i. Now

$$p(f) = Z^{-1}\exp(-\sum_{c \in \mathbf{C}} \psi_c(f)), \tag{4.7}$$

so

$$p(f_i|f_{\mathbf{S}-\{i\}}) = \frac{\exp(-\sum_{c \in \mathbf{C}} \psi_c(f))}{\sum_{f_i'} \exp(-\sum_{c \in \mathbf{C}} \psi_c(f'))}. \tag{4.8}$$

Next $\mathbf{C}$ is divided into two sets $\mathbf{A}$ and $\mathbf{B}$, with $\mathbf{A}$ consisting of cliques which have site i as a member and $\mathbf{B}$ consisting of sites which do not have site i as a member. Now Eqn. 4.8 can be rewritten as

$$p(f_i|f_{\mathbf{S}-\{i\}}) = \frac{\exp(-\sum_{c \in \mathbf{A}} \psi_c(f))\exp(-\sum_{c \in \mathbf{B}} \psi_c(f))}{\sum_{f_i'} \left\{ \exp(-\sum_{c \in \mathbf{A}} \psi_c(f'))\exp(-\sum_{c \in \mathbf{B}} \psi_c(f')) \right\}}. \tag{4.9}$$

The probability depends only on the potentials of cliques containing site i, since $\psi_c(f) = \psi_c(f')$ for cliques not containing i. Thus all other energy terms cancel out, as they appear in both the numerator and the denominator, leaving

$$p(f_i|f_{\mathbf{S}-\{i\}}) = \frac{\exp(-\sum_{c \in \mathbf{A}} \psi_c(f))}{\sum_{f_i'} \exp(-\sum_{c \in \mathbf{A}} \psi_c(f'))}. \tag{4.10}$$

Thus a Gibbs random field is a Markov random field.

4.2 Markov random field formulation applied to the Shape from Shading problem

This section explains how the Markov random field methodology can be used with a surface intensity image (reflectance map), given lighting and camera information, both to recover its shape and to estimate the MRF parameterisation (in this case the MRF parameters will correspond to the BRDF parameters). The MRF is constructed by identifying "corner vertex nodes" on an image lattice. These corner vertex nodes are situated at the corners of pixels. This is a departure from usual MRF image processing methods, where the nodes or sites typically represent the states on the pixels themselves. We then form a set of clique potential functions ψ which are functions of triplets of corner vertex nodes.
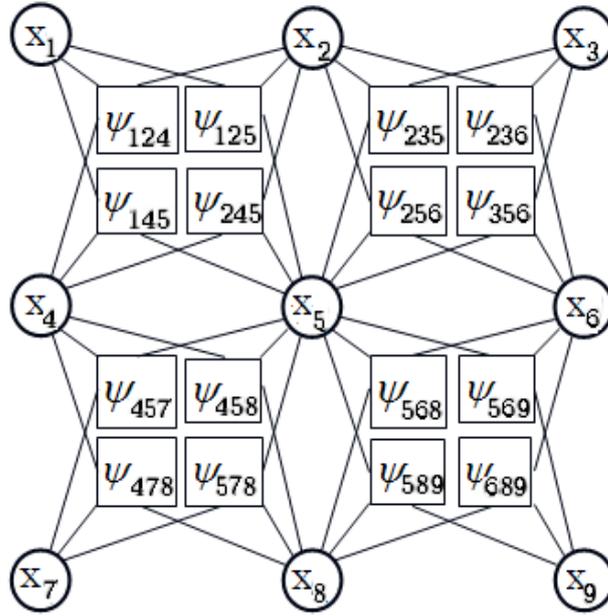
Figure 4.1: This diagram illustrates the energy terms (square nodes) on the triplets of random variables representing the labels on corner vertex nodes (round nodes labeled x_i). If a loopy belief propagation approach is used to calculate the posterior distribution (as in [73]), the square nodes would represent factor nodes connected to the variable nodes (round nodes).

A diagram for the topology for this scheme with pixels, corner vertex nodes, and the corresponding energy terms for each triplet, is shown in Fig. 4.1. With the inclusion of explicit range data on each corner vertex node, the additional energy terms are shown in Fig. 4.2.

The shape from shading algorithms we developed in [73], [72] and [71] calculate (approximately) an optimal set of labels for the height at each corner vertex on a surface, given a reflectance map of that surface with known lighting and camera information. As in these three papers, the state on a corner vertex node represents the range or depth of the surface at that location. Each triplet of vertices describes a unique plane passing through the corresponding 3D locations on the surface. The orientation of that plane relative to the direction of the light source allows a probability to be assigned to that configuration for that triplet, given the observed intensities in the corresponding image region. It is also assumed without loss of generality to the functioning of the algorithm that there is a single light source at infinity, and therefore that all the light rays are parallel (the algorithm can be easily adjusted to incorporate light sources within the scene).

The plane generated by the triplet of corner vertex 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
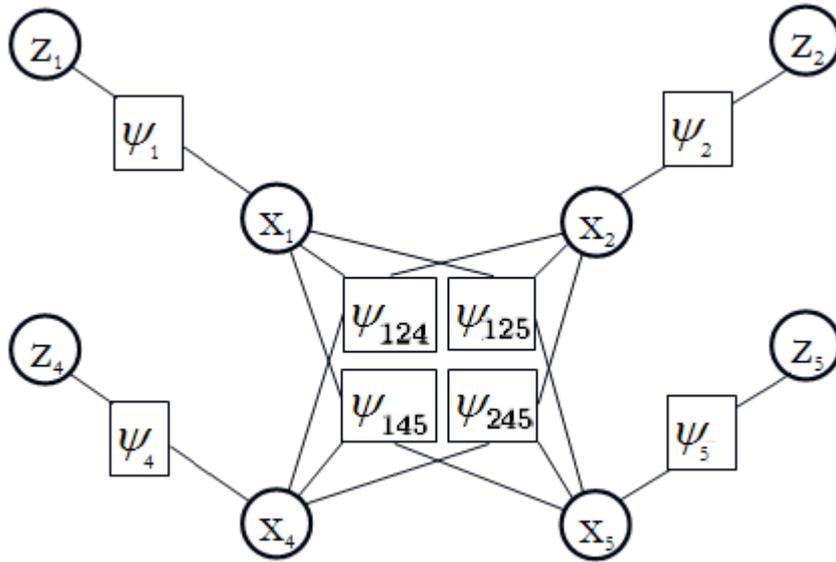
Figure 4.2: This diagram illustrates the energy terms (square nodes) on the triplets of random variables representing the labels on corner vertex nodes (round nodes labeled $\mathbf{x}_i$) as well as the corresponding dependence of the corner vertex node labels $\mathbf{x}_i$ on the range data for each corner vertex node contained in $\mathbf{z}_i$. The round nodes labeled $\mathbf{z}_i$ are visible nodes giving range or depth data.
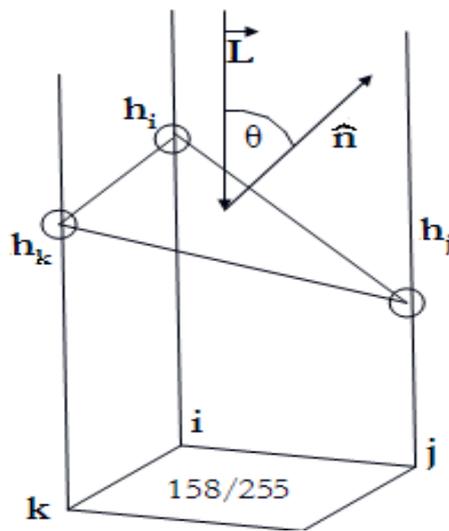
Figure 4.3: A triangular plane is generated by the three points on the surface, each corresponding to a label value for the site of one of the corner vertex nodes surrounding the pixel. The normal to the plane $\hat{\mathbf{n}}$ and the incident light source direction $\vec{L}$ are indicated. The elevation of a node i, when taking on a label x_i, is h_i (it will be seen later that this example is for an affine camera). The square region interior to the four corner vertex nodes is a pixel, in this case with intensity $158/255$

the reflectance function used. This is shown in Fig. 4.3, 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:

$$p(\mathbf{X}|\mathbf{Y}, \mathbf{Z}, \theta) \propto \prod_{\substack{i,j,k \\ i<j<k}} \exp(-\psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta)) \prod_i \exp(-\psi_i(x_i, z_i)) \qquad (4.11)$$

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

$$\psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta) = |y_{ijk} - \mathrm{L}(i, j, k, x_i, x_j, x_k, \vec{L}, \mathbf{P}, \theta)|/\sigma_b \qquad (4.12)$$

where L is the radiance of the surface point into the camera lens, and 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 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

$$\psi_i(x_i, z_i) = |z_i - u_i(x_i)|/\sigma_r, \qquad (4.13)$$

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. 4.11 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 [139], [29]. 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):

$$p = \partial u/\partial x \qquad \text{and} \qquad q = \partial u/\partial y, \qquad (4.14)$$

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

$$\hat{\mathbf{n}} = (-p, -q, 1)/\sqrt{(p^2 + q^2 + 1)}. \tag{4.15}$$

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

$$\mathrm{BRDF}(i, j, k, x_i, x_j, x_k, \vec{L}, \mathbf{P}, \theta) = \frac{\rho_d}{\pi}, \tag{4.16}$$

and the surface radiance is

$$\mathrm{L}_r(i, j, k, x_i, x_j, x_k, \vec{L}, \mathbf{P}, \theta) = \frac{\rho_d}{\pi} I |\hat{\mathbf{n}} \cdot \vec{L}|, \tag{4.17}$$

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), L is the (isotropic) radiance emitted by the surface element, and I is the intensity of the light source. 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

$$\hat{\mathbf{n}} = (\mathbf{Z}_j - \mathbf{Z}_i) \times (\mathbf{Z}_k - \mathbf{Z}_i)/||(\mathbf{Z}_j - \mathbf{Z}_i) \times (\mathbf{Z}_k - \mathbf{Z}_i)||, \tag{4.18}$$

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 $\mathbf{P}$.

As in [73] we can extend the energy function to include static scene/moving light source information (on the assumption that all points on the surface are always visible to both the camera and to all light sources). We develop a simple occlusion formulation in [72], but we do not explore that formulation in this thesis, although it is applicable). For multiple reflectance maps, Eqn. 4.12 above is adjusted to:

$$\psi_{ijk}(x_i, x_j, x_k, \vec{y}_{ijk}, \theta) = \sum_{m=1}^{M} |y_{m,ijk} - \mathrm{L}_{rm}(i, j, k, x_i, x_j, x_k, \vec{L}, \mathbf{P}, \theta)|/\sigma_b, \tag{4.19}$$

where θ is a parameter set which contains the BRDF parameter set. M is the number of images from the same camera (one image taken for each light source), m indexes each of the images, and $y_{m,ijk}$
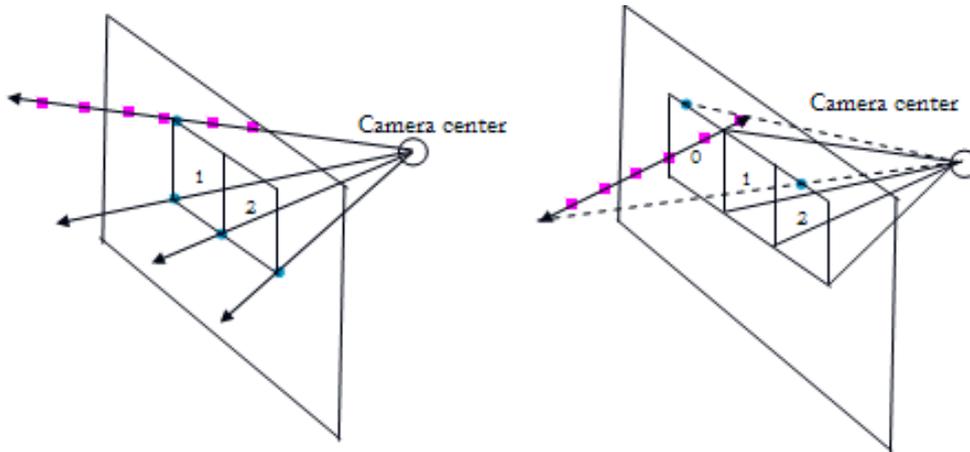
Figure 4.4: Two possible surface depth or range parameterisations. In the first parameterisation (left), the site label for a surface point (corner vertex node) corresponds to a distance of the point from the camera center. In the second parameterisation (right), the label value on a node corresponds to the depth of the surface point behind the camera plane.

is the intensity of the image region contained by the projection of the 3-D points corresponding to the three corner vertex nodes i, j, k of the first image, in the mth image. $\overrightarrow{L}_m$ is the light source direction in the mth image, and $\text{L}_{rm}(\cdot)$, is now a function which returns the intensity value for the mth image. This is the basic formulation which we explored in [73] and [72].

4.2.1 Depth or Range parameterisation for shape from shading

At this point we note that if the label of a corner vertex node refers to its elevation, there are at least two obvious parameterisations to consider. These are depicted in Fig. 4.4. The first parameterisation (Fig. 4.4, left), is such that the label on a corner vertex node corresponds to its range from the (first) camera center. The second parameterisation (Fig. 4.4, right) is such that the label of a corner vertex node corresponds to the perpendicular depth of the surface point behind the image plane.

Note that in the second height parameterisation $y_{m,ijk}$ does not relate to a single pixel for $m = 1$ (except in the case of an affine camera), but rather to a projected region in the first image (see Fig. 4.5). The calculation of the expected intensity value over this region is more complicated than the examination of a single pixel intensity for $m = 1$. Furthermore, the first parameterisation corresponds to the geometry of 3-D points reconstructed from stereo image point correspondences. Therefore, in this thesis we only explore the use of a single intensity image (corresponding to the first camera) for BRDF parameter estimation, using the first parameterisation.
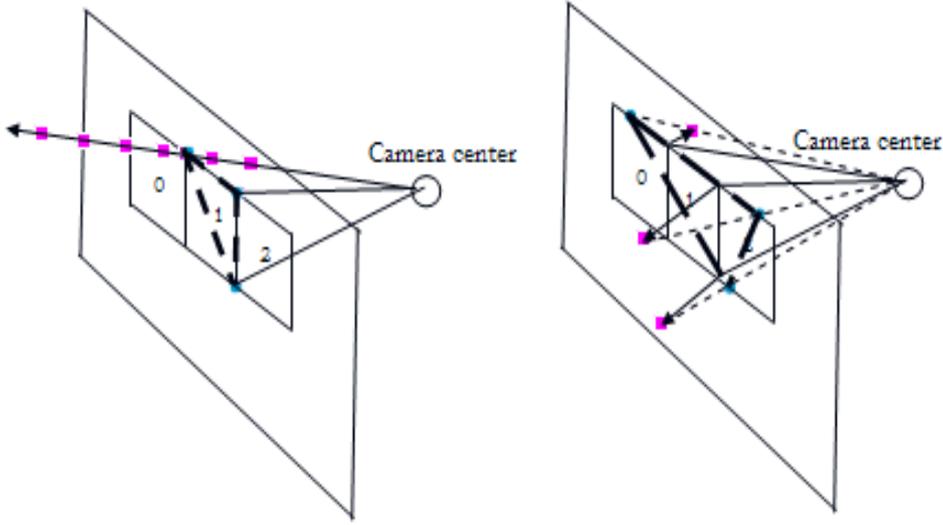
Figure 4.5: In this figure we see how in the second parameterisation, different height labels for the same triplet of corner vertex nodes would require us to evaluate the expected intensity over a region in the image of the first camera which does not correspond to a single pixel. The advantage offered by the first parameterisation (left) is that only single pixels around a site need to be inspected to calculate the energy functions.

4.3 MRF model for surface reflectance using Ward and Phong reflectance models

Now that the general MRF framework for BRDF extraction has been established, it is possible to consider individual reflectance models, such as the Ward and Phong models. To incorporate the Ward reflectance model, Eqn. 4.12 is adjusted to read:

$$\psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta) = |y_{ijk} - \text{WARD}(i, j, k, x_i, x_j, x_k, \overrightarrow{L}, \mathbf{P}, \theta)|/\sigma_b \qquad (4.20)$$

where $\theta = (\rho_d, \rho_s, \alpha)$ contains the Ward reflectance parameters. WARD$(\cdot)$ is a function which takes as parameters the locations on the lattice i, j, k and calculates the 3-D points on the material surface given the labels on the random variables x_i, x_j, x_k. The angles or vectors of incidence and reflectance are calculated for this 3-D surface triangle given the location of the camera center (calculated linearly using the first camera matrix contained in $\mathbf{P}$, see [49] for details) and the light source direction $\vec{L}$. Then Eqn. 3.1 can be used to determine the expected intensity of the image region at the point on the surface. Similarly, for the anisotropic Phong model:

$$\psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta) = |y_{ijk} - \text{PHONG}(i, j, k, x_i, x_j, x_k, \overrightarrow{L}, \mathbf{P}, \theta)|/\sigma_b, \qquad (4.21)$$

where $\theta = (R_d, R_s, n_u, n_v)$, or simply $\theta = (R_d, R_s, n)$, since we have set $n_u = n_v$. This geometry is shown in Fig. 4.6, where the normal $\hat{\mathbf{n}}$ calculated for the surface triangle corresponding to particular
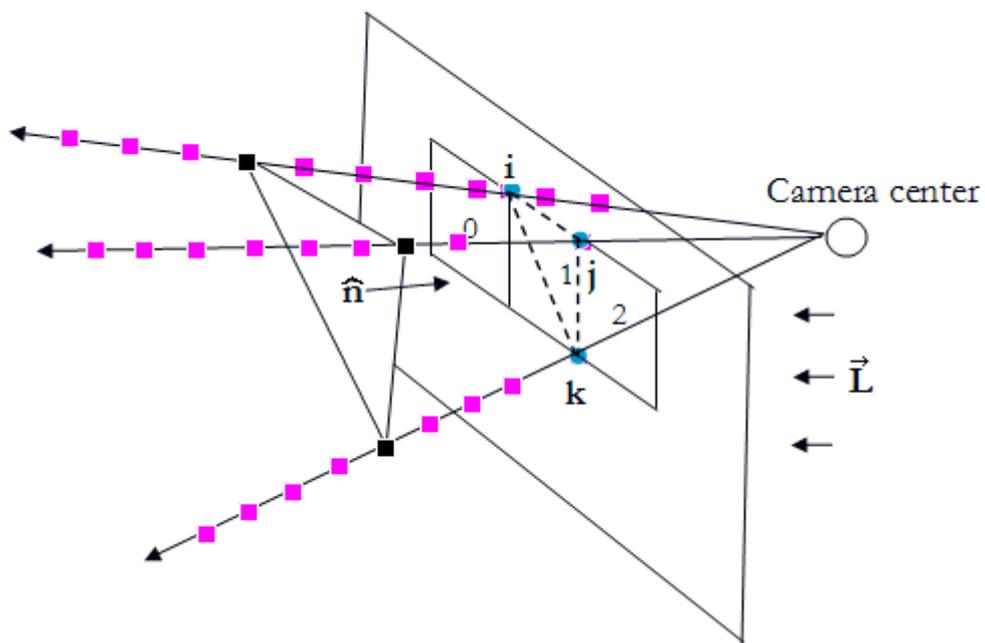
Figure 4.6: In this figure is shown a surface triangle in the scene, which corresponds to the corner vertex nodes i, j, k taking on labels which correspond ranges indicated by the black squares. The triangle normal $\hat{n}$ is shown. The reflectance intensity of the surface triangle is viewed in the upper right half of pixel 1 (the region's perimeter is drawn with a broken line). The light source direction is also shown.

labels chosen for the random variables x_i, x_j, x_k at corner vertex nodes i, j, k is added. This is described in section 6.4.2 after triangulation has been discussed. The reflectance intensity of the surface triangle is projected into the upper right half of pixel 1 (the region's perimeter is drawn with a broken line). The light source direction $\vec{L}$ is also shown (the angles/directions in Eqns. 3.1 and 3.12 assume the direction is from the surface to the camera/light source). The geometry of Fig. 3.1 is used on the triangular surface patch to calculate the intensity of the corresponding area in the image plane.

4.4 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

$$p(f|\theta) = Z(\theta)^{-1} \exp(-U(f,\theta)) = \frac{\exp(-U(f,\theta))}{\sum_{s \in F} \exp(-U(s,\theta))}, \qquad (4.22)$$

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 [7]. The conditional probability per site on a graph is estimated using only its immediate (Markov) neighbours. This is now expressed in general terms, with a general potential function $U(\cdot)$. Note that the pseudolikelihood equals the true likelihood when all the labels are independent. For a single site i (and neighbourhood) labelling,

$$p(f_i|f_{\mathcal{N}_i}, \theta) = \frac{\exp(-U(f_i, f_{\mathcal{N}_i}, \theta))}{\sum_{f_s \in \mathcal{L}} \exp(-U(f_s, f_{\mathcal{N}_s}, \theta))}, \qquad (4.23)$$

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

$$PL(f|\theta) = \log \prod_{i \in S} p(f_i|f_{\mathcal{N}_i}, \theta). \qquad (4.24)$$

This pseudolikelihood estimate is necessary to approximate

$$p(\theta|f) = p(f|\theta) \frac{p(\theta)}{p(f)}. \qquad (4.25)$$

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

$$p(\theta|f) = \exp(PL(f|\theta)), \qquad (4.26)$$

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 [69].

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

$$U_i(x_i, x_{\mathcal{N}_i}, \theta) = \sum_{\substack{j \in \mathcal{N}_i, k \in \mathcal{N}_i \\ k > j}} \psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta), \qquad (4.27)$$

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. 4.2 are being used, the potential term U 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

$$\theta_{PL}^* = \arg\max_\theta PL(f|\theta). \qquad (4.28)$$

A stereo reconstruction may be modelled as a realization of a Markov random field, since it is a function of the disparity map between the pair of stereo images. This disparity map is an instance of the set of possible disparity maps, given the number of potential candidate matches in the second image for points in the first image.

The Markov Random Field framework for using corner vertex nodes with potential energies on their triplets is novel to this thesis, as is the notion of using control points which are no the pixels themselves, but rather the corner vertex nodes which lie at their intersections. We are now equipped to begin a discussion of some of the sampling algorithms and optimization methodologies which may be used to do parameter estimation on this Markov Random Field arrangement.

Chapter 5

MCMC, Population Monte Carlo, and iterative Bayesian resampling methods

This chapter describes the theory of sampling techniques such as particle filtering, partitioned particle filtering (PPF), MCMC, dynamically weighted MCMC and Population Monte Carlo. Ultimately we would like to use MCMC style sampling methods to do parameter estimation on the MRFs as described in the previous chapter. Resampling and importance sampling are also described as they are important techniques used by the other sampling methods. Particle filtering and partitioned particle filtering are reviewed because the partitioned sampling in PPF is similar to PMC, (additionally it is shown in [84] that inference done on a Markov network using particle filtering is one type of message passing done for inference on a Bayesian network, and that a MRF is one type of Bayesian network). The concepts are thus interrelated. The chapter ends with a discussion of the pdf softening methodology used on the target distribution on the BRDF parameters. This is necessary since the target posterior distribution on the BRDF parameters is too sharp to use PMC effectively to calculate a representative set of weighted samples for the pdf.

Our PMC method may be seen as a compromise between doing an exhaustive search for the optimal values (MAP parameter values), and returning an unbiased posterior distribution on these parameters. We also identify that Partitioned Particle Filtering (due to [74]) over a single time step is a special case of Population Monte Carlo sampling. In the notation used here

- $\mathbf{X}^t$ indicates a random variable at time t,

- $\mathcal{X}^t$ indicates the set of random variables $\{\mathbf{X}^i\}$ for $i = 0, \dots, t$,

- x_i^t is the ith sample at time t from $\mathbf{X}^t$,

- π_i^t is the weight on sample x_i^t,

- $\overrightarrow{x}^t$ is the set of samples x_i^t, $i = 1, \ldots, M$, if there are M samples.

- $\overrightarrow{\pi}^t$ is the set of sample weights π_i^t, $i = 1, \ldots, M$, if there are M samples.

5.1 Particle Filtering

This section describes particle filtering, also known as the conditional density propagation (condensation) algorithm, or more generally as iterative nonlinear Bayesian filtering. To do so, consider Bayes' theorem

$$p(\mathbf{X}|\mathbf{Z}) = \frac{p(\mathbf{Z}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Z})}, \tag{5.1}$$

where in this context $\mathbf{X}$ refers to a state estimate, and $\mathbf{Z}$ to the observed data, which are used to make state estimates. The chain rule of probability

$$p(\mathbf{A}, \mathbf{B}, \mathbf{C}) = p(\mathbf{A}|\mathbf{B}, \mathbf{C})p(\mathbf{B}|\mathbf{C})p(\mathbf{C}), \tag{5.2}$$

is used to derive the recursive formulation of the particle filter equations. The Markov property in stochastic dynamics implies that the state variables form a temporal Markov chain:

$$p(\mathbf{X}^t|\mathbf{X}^{t-1}) = p(\mathbf{X}^t|\mathcal{X}^{t-1}). \tag{5.3}$$

where $\mathcal{X}^{t-1} = \{\mathbf{X}^0, \ldots, \mathbf{X}^{t-1}\}$. Observations (posterior probability correction at each time step, for each particle, given the observed data) are assumed to be mutually independent, and also independent of the dynamical process:

$$p(\mathcal{Z}^{t-1}, \mathbf{X}^t|\mathcal{X}^{t-1}) = p(\mathbf{X}^t|\mathbf{X}^{t-1}) \prod_{i=1}^{t-1} p(\mathbf{Z}^i|\mathbf{X}^i), \tag{5.4}$$

where $\mathcal{Z}^{t-1} = \{\mathbf{Z}^0, ..\mathbf{Z}^{t-1}\}$ and $\mathcal{X}^{t-1} = \{\mathbf{X}^0, ..\mathbf{X}^{t-1}\}$. Since observation densities $p(\mathbf{Z}^t|\mathbf{X}^t)$ are modeled as independent,

$$p(\mathcal{Z}^t|\mathcal{X}^t) = \prod_{i=1}^{t} p(\mathbf{Z}^i|\mathbf{X}^i), \tag{5.5}$$

where $\mathcal{Z}^t = \{\mathbf{Z}^0, ..\mathbf{Z}^t\}$ and $\mathcal{X}^t = \{\mathbf{X}^0, ..\mathbf{X}^t\}$. Finally, by combining the previous equations, the particle filter equations given observation densities are:

$$p(\mathbf{X}^t|\mathbf{Z}^t) = k^t p(\mathbf{Z}^t|\mathbf{X}^t)p(\mathbf{X}^t|\mathbf{Z}^{t-1}) \tag{5.6}$$

where

$$p(\mathbf{X}^t|\mathbf{Z}^{t-1}) = \int_{\mathbf{X}^{t-1}} p(\mathbf{X}^t|\mathbf{X}^{t-1})p(\mathbf{X}^{t-1}|\mathbf{Z}^{t-1})d\mathbf{X}^{t-1} \tag{5.7}$$
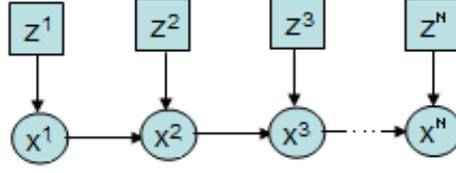
Figure 5.1: Bayesian network of the basic particle filter, which is one of many inference algorithms which can be run on a Bayesian network. This illustrates the conditional dependencies between the state variables (hidden nodes, circles), and the observations taken at each time step (visible nodes, squares).

in which $p(\mathbf{X}^t|\mathbf{X}^{t-1})$ represents the state dynamics, which can be defined uniquely for each time step t. The observation density at time t is $p(\mathbf{Z}^t|\mathbf{X}^t)$. Even if there are no dynamics on the state vector over time, additive noise may be inserted via the term $p(\mathbf{X}^t|\mathbf{X}^{t-1})$, in which case it is referred to as "stochastic diffusion". Adding random noise to evolving particle sets is important to keep exploring new parts of the parameter space. The observation density may also change over time, as does the data $\mathbf{Z}^t$ against which the state vector $\mathbf{X}^t$ is to be compared. This latter probability is the answer to the question "how likely is it that a particular state vector gave rise to the observed data". The normalizing term k^t is expanded as

$$k^t = p(\mathbf{Z}^t|\mathcal{Z}^{t-1}). \tag{5.8}$$

One way to solve these equations algorithmically is to represent the prior (and posterior) distribution at each time step using a weighted set of samples, or particles: $\{x_i^t, \pi_i^t\}$, with $i = 1, 2, ..., N$, where x_i^t represents the ith sample at time t, and π_i^t represents the probability of this sample. In this thesis, $\{x_i^t, \pi_i^t\}$, for $i = 1, 2, ..., N$ is also written $\{\vec{x}^t, \vec{\pi}^t\}$. The Bayesian network depicting the basic particle filter is shown in Fig. 5.1. An algorithm which describes the basic particle filter follows:

1. For $t = 0, \ldots, T$
2. Generate distribution of samples $\mathbf{X}^t$, represented by $(\vec{x}^t, \vec{\pi}^t)$ by sampling
    from $p(\mathbf{X}^t|\mathbf{Z}^{t-1})$ (note that $p(\mathbf{X}^t|\mathbf{Z}^{t-1}) = p(\mathbf{X}^0)$, if $t = 0$)
    This can be done by selecting sample x_i^{t-1} with probability π_i^{t-1}, and generating a new
    sample $x_i^t \sim p(\mathbf{X}^t|\mathbf{X}^{t-1} = x_i^{t-1})$
3. For $i = 1, \ldots, M$ (M is the number of samples representing the distribution $\mathbf{X}^t$
4.   Weight the ith sample according to its observation probability:

$$\pi_i^t = p(\mathbf{Z}^t|\mathbf{X}^t = x_i^t) \tag{5.9}$$

5. end (3)
6. Do a resample operation on the particles currently representing $\mathbf{X}^t$,
    i.e. $(\vec{x}^t, \vec{\pi}^t) = \text{resampleParticles}(\vec{x}^t, \vec{\pi}^t, \sigma)$

       where σ is a parameter (standard deviation) for the addition of Gaussian noise

6. end (1).

5.1.1 Resampling

In the resampling step, samples are taken from the distribution $p(\mathbf{X}^t)$, which is already represented by the weighted sample set $\{\overrightarrow{x}^{t}, \overrightarrow{\pi}^{t}\}$, by choosing sample x_i^t with probability π_i^t, and giving the resulting sample the probability $1/M$ where M is the number of samples. There may be several samples with the same value, and the same probability. Gaussian noise is often added to these samples to ensure that the samples keep exploring the target distribution. A concise way of writing a resampling step is as follows:

1. Normalize $\overrightarrow{\pi}$, if it is not already normalized
2. For $i = 1..N$ (N is the number of samples)
3. $(I_i) \sim \mathcal{M}(1, \overrightarrow{\pi})$
4.   If Gaussian noise is added, $n \sim \mathcal{N}(0; \Sigma)$, otherwise $n = 0$
5. $x_{i,\text{new}} = x_{I_i} + n$
6. $\pi_{i,\text{new}} = 1/N$
7. end (2)

where n is a sample drawn from the Gaussian distribution with mean 0 and covariance Σ. $\mathcal{M}$ is the multinomial sampling function, which draws one index from the available indices, which have corresponding probabilities in the normalized vector $\overrightarrow{\pi}$ (see Appendix A for a more detailed explanation of multinomial sampling). I_i indicates the index chosen for sample i. Hereafter, the above steps are indicated by the notation

$$p(\mathbf{X})^{\text{new}} = \text{resample}(p(\mathbf{X})),\qquad(5.10)$$

where $p(\mathbf{X})$ is a pdf, represented by a sample set. Alternatively we may state the operation as

$$(\overrightarrow{x}^{\text{new}}, \overrightarrow{\pi}^{\text{new}}) = \text{resampleParticles}(\overrightarrow{x}, \overrightarrow{\pi}),\qquad(5.11)$$

where $\overrightarrow{x}$ is the set of samples representing $p(\mathbf{X})$ and $\overrightarrow{\pi}$ contains all the corresponding sample weights. If we add Gaussian noise to the new sample values, we write

$$(\overrightarrow{x}^{\text{new}}, \overrightarrow{\pi}^{\text{new}}) = \text{resampleParticles}(\overrightarrow{x}, \overrightarrow{\pi}, \sigma)\qquad(5.12)$$

where $\Sigma = \sigma \mathbb{I}_{d \times d}$, where d is the dimension of a sample vector, and $\mathbb{I}_{d \times d}$ is the identity matrix of dimension $d \times d$ (corresponding to the case of a spherical Gaussian distribution).

This resampling operation is referred to in the literature variously as branching, reweighting, reconfiguration, rejuvenation, resampling, prune-enrichment, selection. It is not probabilistically correct to add sample noise in the resampling stage without justifying it with a stochastic diffusion term $p(\mathbf{X}^t|\mathbf{X}^{t-1})$. However, it is often algorithmically more convenient to add sample noise at the resampling stage than to add it at the importance sampling stage, if importance sampling via importance reweighting is included in the particle filtering scheme.

5.2 Importance sampling

Suppose that within a recursive Bayesian particle filtering framework, one prefers not to sample from the prior distribution $p(\mathbf{X}^{t-1}|\mathbf{Z}^{t-1})$, but rather from an alternative proposal distribution $g(\mathbf{X}^t|\mathbf{Z}^t)$. The expectation of a function $f(\mathbf{X})$ given a prior distribution on the random variable $\mathbf{X}$ is expressed as

$$E[f(\mathbf{X}^t)] = \int f(\mathbf{X}^t)\frac{p(\mathbf{X}^t|\mathbf{Z}^t)}{g(\mathbf{X}^t|\mathbf{Z}^t)}g(\mathbf{X}^t|\mathbf{Z}^t)d\mathbf{X}^t \tag{5.13}$$

$$= \int f(\mathbf{X}^t)\frac{p(\mathbf{Z}^t|\mathbf{X}^t)p(\mathbf{X}^t)}{p(\mathbf{Z}^t)g(\mathbf{X}^t|\mathbf{Z}^t)}g(\mathbf{X}^t|\mathbf{Z}^t)d\mathbf{X}^t \tag{5.14}$$

$$= \int f(\mathbf{X}^t)\frac{\pi^t(\mathbf{X}^t)}{p(\mathbf{Z}^t)}g(\mathbf{X}^t|\mathbf{Z}^t)d\mathbf{X}^t \tag{5.15}$$

where

$$\pi^t(\mathbf{X}^t) = \frac{p(\mathbf{Z}^t|\mathbf{X}^t)p(\mathbf{X}^t)}{g(\mathbf{X}^t|\mathbf{Z}^t)} \tag{5.16}$$

represents the weight of the sample/particle drawn from the proposal distribution, after it has been renormalized to remove the bias introduced by sampling from the proposal distribution, and after it has been observed against the data $\mathbf{Z}^t$. Then

$$E[f(\mathbf{X}^t)] = \frac{1}{p(\mathbf{Z}^t)}\int f(\mathbf{X}^t)\pi^t(\mathbf{X}^t)g(\mathbf{X}^t|\mathbf{Z}^t)d\mathbf{X}^t \tag{5.17}$$

$$= \frac{\int f(\mathbf{X}^t)\pi^t(\mathbf{X}^t)g(\mathbf{X}^t|\mathbf{Z}^t)d\mathbf{X}^t}{\int p(\mathbf{Z}^t|\mathbf{X}^t)p(\mathbf{X}^t)\frac{g(\mathbf{X}^t|\mathbf{Z}^t)}{g(\mathbf{X}^t|\mathbf{Z}^t)}d\mathbf{X}^t} \tag{5.18}$$

$$= \frac{\int f(\mathbf{X}^t)\pi^t(\mathbf{X}^t)g(\mathbf{X}^t|\mathbf{Z}^t)d\mathbf{X}^t}{\int \pi^t(\mathbf{X}^t)g(\mathbf{X}^t|\mathbf{Z}^t)d\mathbf{X}^t} \tag{5.19}$$

$$= \frac{E_{g(\mathbf{X}^t|\mathbf{Z}^t)}[\pi^t(\mathbf{X}^t)f(\mathbf{X}^t)]}{E_{g(\mathbf{X}^t|\mathbf{Z}^t)}[\pi^t(\mathbf{X}^t)]}. \tag{5.20}$$

This shows that it is necessary to normalize the weights at each time step, across the new particle weights $\pi(\mathbf{X}^t)$ to form a posterior from which an expectation can be taken.

Drawing samples $x_1^t, \ldots, x_N^t$ from $g(\mathbf{X}^t|\mathbf{Z}^t)$, expectations of interest can be approximated by

$$E[f(\mathbf{X}^t)] = \frac{\frac{1}{N}\sum_{i=1}^N \pi^t(x_i^t)f(x_i^t)}{\frac{1}{N}\sum_{i=1}^N \pi^t(x_i^t)} \approx \sum_{i=1}^N \tilde{\pi}^t(x_i^t)f(x_i^t), \qquad (5.21)$$

where the normalized weights are given by

$$\tilde{\pi}^t(x_i^t) = \frac{\pi^t(x_i^t)}{\sum_{j=1}^N \pi^t(x_j^t)}, \qquad (5.22)$$

and where, in terms of the notation used elsewhere in this thesis,

$$\pi_i^t = \pi^t(x_i^t). \qquad (5.23)$$

Particle filtering where the samples are generated from a proposal distribution $g(\mathbf{X}|\mathbf{Z})$ is expressed by the equation

$$p(\mathbf{X}^t|\mathbf{Z}^t) = k^t p(\mathbf{Z}^t|\mathbf{X}^t)\frac{g(\mathbf{X}^t|\mathbf{Z}^t)}{g(\mathbf{X}^t|\mathbf{Z}^t)}p(\mathbf{X}^t|\mathbf{Z}^{t-1}), \qquad (5.24)$$

which indicates that all samples are drawn from $g(\mathbf{X}^t|\mathbf{Z}^t)$ and reweighted by $k^t p(\mathbf{Z}^t|\mathbf{X}^t)\frac{p(\mathbf{X}^t|\mathbf{Z}^{t-1})}{g(\mathbf{X}^t|\mathbf{Z}^t)}$.

It was stated earlier that it is often algorithmically more convenient to add sample noise at this stage than to add it at the importance sampling stage, if importance sampling is included in the particle filtering scheme.

This is because importance sampling may take the form either of simulating from a specified importance sampling distribution which can be based on the previous sample set (in which case noise is added through the importance function, which is usually a Gaussian distribution centered on the expected mean of the state variable $\mathbf{X}$), or of multiplication (reweighting) of the incoming sample set's weights (which often represents the uniform distribution) by each sample's probability evaluated by the importance function, followed by a resample operation (this is called importance reweighting). If the latter is done, as in [75], the only logical place to add the noise is after the resampling stage.

5.2.1 Classic Sample Importance Resample (SIR) algorithm

The usual form of the SIR filter, which incorporates a resample step after the importance sampling step, is described as follows.

1. For $t = 0..T$
2.   Simulate a sample set $\vec{x}^t$ from distribution g^t, with

$$x_i^t \sim g^t(\mathbf{X}^t|\mathbf{Z}^t), \qquad (1 \leq i \leq N). \qquad (5.25)$$

3.   Calculate the probability weight $\hat{\pi}$ for each sample in $\overrightarrow{x}^t$:

$$\hat{\pi}_i^t = \frac{p(\mathbf{Z}^t|\mathbf{X}^t = x_i^t)p(\mathbf{X}^t = x_i^t|\mathbf{Z}^{t-1})}{g^t(\mathbf{X}^t = x_i^t|\mathbf{Z}^t)} \tag{5.26}$$

4.   Normalize the sample weights:

$$\pi_i^t = \frac{\hat{\pi}_i^t}{\sum_{j=1}^N \hat{\pi}_j^t} \tag{5.27}$$

5.   Resample the sample set:

$$(\overrightarrow{x}^t, \overrightarrow{\pi}^t)^{\text{new}} = \text{resampleParticles}(\overrightarrow{x}^t, \overrightarrow{\pi}^t). \tag{5.28}$$

6. end(1).

In the SIR algorithm described above, step 2 indicates that the ith of N samples is drawn from the importance function g^t. In step 3 the new probability weights of each of the samples are calculated. Note that each new probability depends on that particle's previous probability, and has as a denominator the probability of that sample having been drawn from the proposal distribution g^t.

5.2.2   Alternative SIR algorithm

An alternative way of doing Sample Importance Resampling, using importance reweighting, is described in the following algorithm:

1. For $t = 1..T$

2.   Reweight the samples in the sample set $(\overrightarrow{x}^U, \overrightarrow{\pi}^U)$ representing the uniform distribution U over the parameter space:

$$\pi_i \leftarrow g^t(\mathbf{X} = x_i^U|\mathbf{Z}^t), \qquad (1 \le i \le N). \tag{5.29}$$

3. $(\overrightarrow{x}^t, \overrightarrow{\pi}^t) = \text{resampleParticles}(\overrightarrow{x}^U, \overrightarrow{\pi}, \sigma)$

4.   Reweight the samples in the sample set $(\overrightarrow{x}^t, \overrightarrow{\pi}^t)$:

$$\hat{\pi}_i^t \leftarrow \frac{\pi_i^t p(\mathbf{Z}^t|\mathbf{X}^t = x_i^t)p(\mathbf{X}^t = x_i^t|\mathbf{Z}^{t-1})}{g^t(\mathbf{X}^t = x_i^t|\mathbf{Z}^t)}, \qquad (1 \le i \le N). \tag{5.30}$$

5.   Normalize the sample weights:

$$\pi_i^t = \frac{\hat{\pi}_i^t}{\sum_{j=1}^N \hat{\pi}_j^t} \tag{5.31}$$

6. end(1).

In the above algorithm, σ is the standard deviation of the additive Gaussian noise in the resampling operation. If $t = 1$, $p(\mathbf{X}^{t-1}|\mathbf{Z}^{t-1}) = p(\mathbf{X}^0)$, which can be represented by a set of samples evenly distributed in the parameter space to represent the uniform distribution (if there is no prior information about the distribution $p(\mathbf{X}^0)$). In step 2, samples representing the uniform distribution can be generated by choosing sample points at regular intervals in the parameter space.

5.3 Partitioned Particle Filtering

Partitioned Particle Filtering (PPF), due to [74], is a method for concentrating samples in a particular region in the parameter space, within the particle filtering framework. The method uses iterative importance reweighting, which is probabilistically equivalent to importance sampling. The partitioned particle filtering scheme may be expressed by adjusting Eqn. 5.6 to read

$$p(\mathbf{X}^t|\mathbf{Z}^t) = k^t p(\mathbf{Z}^t|\mathbf{X}^t) \prod_{n=1}^{N} \frac{g^n(\mathbf{X}^t|\mathbf{Z}^t)}{g^n(\mathbf{X}^t|\mathbf{Z}^t)} p(\mathbf{X}^t|\mathbf{Z}^{t-1}), \tag{5.32}$$

where N is the number of partitions on the state vector $\mathbf{X}$ (the rest of the equations are the same as in the basic particle filter). This algorithm may be computed as follows:

1. For $t = 0, \dots, T$
2.   Generate distribution of samples $\mathbf{X}^t$, represented by $(\overrightarrow{x}^t, \overrightarrow{\pi}^t)$ by
           sampling from $p(\mathbf{X}^t|\mathbf{Z}^{t-1})$ (note that $p(\mathbf{X}^t|\mathbf{Z}^{t-1}) = p(\mathbf{X}^0)$, if $t = 0$)
3.   For $n = 1, \dots, N$ partitions
4.   For $i = 1, \dots, M$ samples representing the distribution $\mathbf{X}^t$

$$\pi_i^t \leftarrow \pi_i^t g^n(\mathbf{X}^t = x_i^t|\mathbf{Z}^t) \tag{5.33}$$

5.   end(4).
6.   Do a resampling step on the samples currently representing $\mathbf{X}^t$,
          i.e. $(\overrightarrow{x}^t, \overrightarrow{\pi}^t) = \text{resampleParticles}(\overrightarrow{x}^t, \overrightarrow{\pi}^t, \sigma)$,
         where σ is a parameter (standard deviation) for the addition of Gaussian
         noise,
7.   For $i = 1, \dots, M$ samples representing the distribution $\mathbf{X}^t$
8.   Reweight

$$\pi_i^t \leftarrow \pi_i^t / g^n(\mathbf{X}^t = x_i^t|\mathbf{Z}^t) \tag{5.34}$$

9.   end (7)
10.   end (3)
11.   Weight each of the M samples according to its observation probability:

$$\pi_i^t \leftarrow \pi_i^t p(\mathbf{Z}^t|\mathbf{X}^t = x_i^t) \tag{5.35}$$

11. end (1).

In [74], the partitioned particle filtering scheme is represented diagrammatically as follows (in this example there are two partitions, one for each dimension of the random variable $\mathbf{X}$, with associated
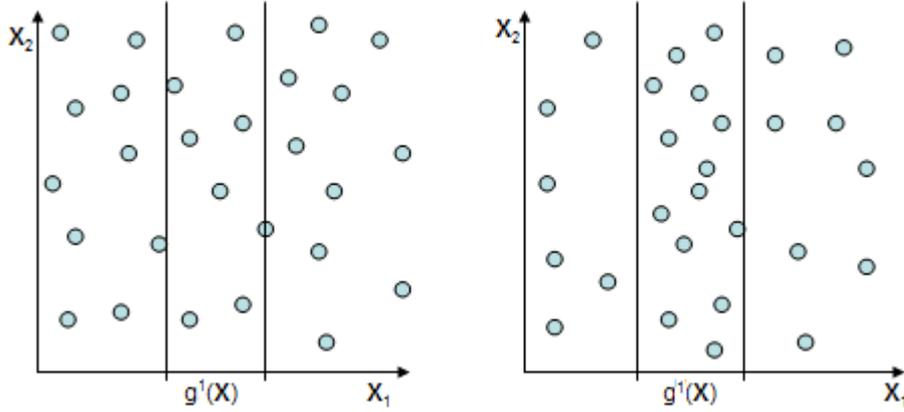
Figure 5.2: The sample set is resampled using importance function g^1 (left). The new sample set after the importance resampling stage (with additive Gaussian noise applied to the new samples) is shown on the right.

importance sampling functions g^1 and g^2):

$$
\begin{aligned}
p(\mathbf{X}|\mathbf{Z}^{t-1}) \rightarrow \quad \sim \quad &\rightarrow *h_1(\mathbf{X}'|\mathbf{X}) \rightarrow \sim g^1 \\
&\rightarrow *h_2(\mathbf{X}''|\mathbf{X}') \rightarrow \sim g^2 \\
\rightarrow *h_3(\mathbf{X}''|\mathbf{X}) \rightarrow &\times f(\mathbf{Z}^t|\mathbf{X}'') \rightarrow p(\mathbf{X}|\mathbf{Z}^t)
\end{aligned}
\tag{5.36}
$$

Note that before each importance sampling partition, the samples are passed through local dynamic distributions $*h_n(\mathbf{X}'|\mathbf{X})$, which usually take the form of linear transition kernels. Although this step could be absorbed into the importance sampling stage, in the motion tracking literature (where partitioned particle filtering was introduced), the dynamics are separated from the importance sampling process. In this diagram, $\sim$ indicates a resampling operation, and the operation $\sim g^n$ refers to a single importance weighting-resampling step, performed on the incoming sample set.

Note that in the scheme shown in Eqn. 5.36, the symbol g^n indicates a weighted resampling operation where the samples have their weights multiplied by an importance function, after which the weighted samples are resampled, and then the weights are multiplied by the inverse of the probability of the importance function for each sample. This ensures that the distribution remains unbiased after any PPF importance weighted resampling operation. Figs. 5.2 and 5.3 show evolution of a particle set over two importance resampling iterations. These diagrams also describe the effects of the importance sampling operations of PMC.
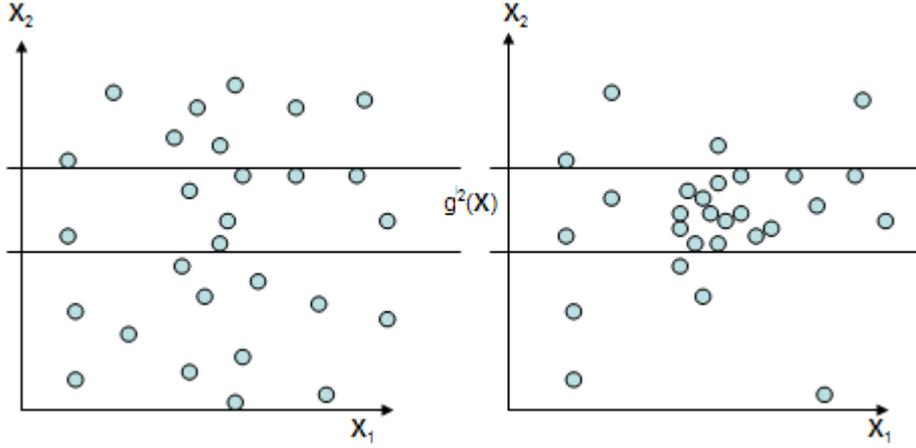
Figure 5.3: The result of the previous importance resampling operation (Fig. 5.2) is resampled using importance function g^2 (left). The new sample set after the importance resampling stage (with additive Gaussian noise applied to the new samples).

5.4   Markov Chain Monte Carlo (MCMC)

MCMC is a popular method for exploring probability distributions and for integrating functions. According to Bayes' theorem the following equation can be written which describes the posterior distribution of θ conditional on $\mathbf{X}$, and which assumes no prior information about the distribution of $\mathbf{X}$:

$$p(\theta|\mathbf{X}) = \frac{p(\theta)p(\mathbf{X}|\theta)}{\int p(\theta)p(\mathbf{X}|\theta)d\theta} \propto p(\theta)p(\mathbf{X}|\theta). \tag{5.37}$$

Any feature of a posterior distribution may be used for Bayesian inference, including n^{th} order moments, highest posterior density regions, and quantiles. These can be expressed in terms of a posterior distribution on θ. The posterior expectation of a function $g(\theta)$ is

$$E(g(\theta|\mathbf{X})) = \frac{\int g(\theta)p(\theta)p(\mathbf{X}|\theta)d\theta}{\int p(\theta)p(\mathbf{X}|\theta)d\theta}, \tag{5.38}$$

which can be simplified to

$$E(g(\theta)) = \frac{\int g(\theta)p(\theta)d\theta}{\int p(\theta)d\theta}. \tag{5.39}$$

The integrals in such an expression are hard to evaluate, but can be evaluated by using a Markov chain. A sequence of random variables $\{\theta_0, \theta_1, \dots, \theta_N\}$ is thus generated: each state from time $i > 0$ is a sample from a distribution $p(\theta_{i+1}|\theta_i)$, which depends only on the state at time i. If it is assumed that the Markov chain is time homogeneous, then the sequence converges to some stationary distribution $\varrho(\cdot)$. When sampling with Markov chains, the first m samples are usually discarded (this is known as the

burn-in), so that $\{\theta_i, i = m+1, \dots, n\}$ are dependent samples approximately from $\varrho(\cdot)$. Therefore,

$$\theta = \frac{1}{n-m} \sum_{i=m+1}^{n} \theta_i \tag{5.40}$$

is an ergodic average. Such a Markov chain may be constructed using the Metropolis-Hastings algorithm. For each time i, the next state θ_{i+1} is chosen by sampling from a proposal distribution $g(\theta'|\theta_i)$. The candidate is accepted with probability

$$\alpha(\theta_i, \theta') = \min\left(1, \frac{p(\theta'|\mathbf{X})g(\theta'|\theta_i)}{p(\theta_i|\mathbf{X})g(\theta_i|\theta')}\right). \tag{5.41}$$

If the new sample candidate is accepted, the next state is $\theta_{i+1} = \theta'$, otherwise $\theta_{i+1} = \theta_i$. Assuming the prior distribution $p(\theta)$ is flat (e.g. if prior information is unavailable), and if $p(\theta|\mathbf{X}) \propto p(\theta)p(\mathbf{X}|\theta)$, then

$$\alpha(\theta_i, \theta') = \min\left(1, \frac{p(\theta'|\mathbf{X})g(\theta'|\theta_i)}{p(\theta_i|\mathbf{X})g(\theta_i|\theta')}\right) = \min\left(1, \frac{p(\mathbf{X}|\theta')g(\theta'|\theta_i)}{p(\mathbf{X}|\theta_i)g(\theta_i|\theta')}\right) \tag{5.42}$$

since

$$\frac{p(\theta'|\mathbf{X})}{p(\theta_i|\mathbf{X})} = \frac{p(\mathbf{X}|\theta')\frac{p(\theta')}{p(\mathbf{X})}}{p(\mathbf{X}|\theta_i)\frac{p(\theta_i)}{p(\mathbf{X})}} = \frac{p(\mathbf{X}|\theta')p(\theta')}{p(\mathbf{X}|\theta_i)p(\theta_i)} = \frac{p(\mathbf{X}|\theta')}{p(\mathbf{X}|\theta_i)}. \tag{5.43}$$

The last step follows because there is no prior probability on the observation labels $\mathbf{X}$, nor on the parameters θ. With a symmetric proposal distribution, the acceptance probability formula for the corresponding Metropolis algorithm is simplified to

$$\alpha(\theta_i, \theta') = \min\left(1, \frac{p(\mathbf{X}|\theta')}{p(\mathbf{X}|\theta_i)}\right). \tag{5.44}$$

After describing the basis of the algorithm mathematically, the Metropolis algorithm [80] can be described algorithmically as:

1. Start with θ_0, then iterate:
2. For $i = 1, \dots, N-1$
3. Propose θ' from $g(\theta_i, \theta')$
4. Calculate ratio:

$$a = \frac{p(\mathbf{X}|\theta')}{p(\mathbf{X}|\theta_i)} \tag{5.45}$$

5. If $a > 1$, accept $\theta_{i+1} = \theta'$
    else accept with probability a
    if rejected: $\theta_{i+1} = \theta_i$
6. end(2)

Similarly, the Metropolis-Hastings algorithm [50], which has an asymmetrical proposal distribution, can be described in the following algorithm:

1.   Start with θ_0, then iterate:
2.   For $i = 1, \ldots, N - 1$
3.   Propose θ' from $g(\theta_i, \theta')$
4.   Calculate ratio:

$$a = \frac{p(\mathbf{X}|\theta')g(\theta'|\theta_i)}{p(\mathbf{X}|\theta_i)g(\theta_i|\theta')} \tag{5.46}$$

5.   If $a > 1$, accept $\theta_{i+1} = \theta'$
          else accept with probability a
          if rejected: $\theta_{i+1} = \theta_i$
6.   end (2)

5.5 The dynamically weighted MCMC sampler

After preliminary testing of a basic Metropolis type sampler on the distribution of the BRDF pseudo-likelihood function, it was found that the acceptance ratio was too low, due to the sharp nature of the target distribution on the BRDF parameters. Since the goal is to explore the distribution as well as to find a good MAP value, we implemented the dynamically weighted MCMC sampler, due to [68]. Some analysis of the convergence characteristics of this sampler is included, as a way of describing its behaviour, which may not be well known or understood.

As described in [76], it is possible to include importance sampling in a MCMC sampler for better exploration of a target distribution. In [68], a dynamic weighting scheme for MCMC samplers is introduced, which is essentially an augmentation of MCMC with importance reweighting. The goal is to allow the sampler to move more freely through the target distribution, and to prevent it from being trapped in local probability maxima. The notation used to describe the dynamically weighted MCMC sampler is the same as in [68], and the arguments and proofs for the behaviour of the algorithm are taken directly. This is done to facilitate easy comparison with the contents of the original paper. The notation follows:

- $\varrho(x)$ is the target distribution

- $\mathcal{X}$ is the space on which $\varrho(x)$ is defined

- X_i is a state variable, also defined on $\mathcal{X}$

- Π_i is a dynamic weight, taking values in the range $(0, \infty)$

- $T(x, y)$ is a proposal transition function which is aperiodic and irreducible

- $g(x)$ is an invariant measure of $T(x, y)$

- $g(x, y) = g(x)T(x, y)$

- $U(a, b)$ is the uniform distribution between a and b.

Note that a measure g is an invariant measure for a sequence of random variables $(X_i)_{i \geq 0}$ if, whenever the initial condition X_0 is distributed according to g, so is X_i for $i > 0$.

There are two types of sample transition rule for the dynamically weighted MCMC sampler, namely the Q-type and the R-type moves. The essential idea of each is to increase the weight of a sample until it is accepted. Although the sample weights can increase very far above unity, the weighted posterior distributions (sample sets) still form approximately unbiased representations of the target distributions, up to a scale factor. Both the R-type and the Q-type sampler associate weights with each of the samples, which is atypical for MCMC samplers. The sample weights are used to store information which affects the probability of making a transition (accepting a new, different sample). If the sampler is stuck in a local maximum value for the probability in the target distribution, the weight of the sample increases until the transition is accepted. The two algorithms are now described (taken directly from [68]):

5.5.1 Q-Type dynamic weighted MCMC sampler

Suppose that the initial state is $(X_0, \Pi_0) = (x, \pi)$, i.e. X_0 denotes the original system state at time $i = 0$ and W_0 its weight.

1. For $i = 0, \ldots, N - 1$

2. Propose the next state $Y = y$ from the proposal distribution $T(x, y)$, and compute the Metropolis ratio:

$$r(x, y) = \frac{\varrho(y)T(y, x)}{\varrho(x)T(x, y)} \tag{5.47}$$

3. Choose $\theta = \theta(x, \pi) \geq 0$ and draw $u \sim U(0, 1)$

4. Update (X_i, Π_i) to (X_{i+1}, Π_{i+1}) according to

$$(X_{i+1}, \Pi_{i+1}) = \begin{cases} (y, \max(\theta, \pi r(x, y))) & \text{if } u \leq \min(1, \pi r(x, y)/\theta) \\ (x, a\pi) & \text{otherwise,} \end{cases} \tag{5.48}$$

where $a > 1$ can be a constant or an independent random variable

5. end(1)

5.5.2 R-Type dynamic weighted MCMC sampler

Suppose that the initial state is $(X_0, \Pi_0) = (x, \pi)$, i.e. X_0 denotes the original system state at time $i = 0$ and W_0 its weight.

1. For $i = 0, \ldots, N - 1$

2.   Draw $Y = y$ from $T(x, y)$ and compute Metropolis ratio $r(x, y)$

3.   Choose $\theta = \theta(x, \pi) \geq 0$, and draw $u \sim U(0, 1)$

4.   Update (X_i, Π_i) to (X_{i+1}, Π_{i+1}) according to

$$(X_{i+1}, \Pi_{i+1}) = \begin{cases} (y, \pi r(x, y) + \theta) & \text{if } u \leq \frac{\pi r(x,y)}{\pi r(x,y)+\theta} \\ (x, \pi(\pi r(x, y) + \theta)/\theta) & \text{otherwise,} \end{cases} \quad (5.49)$$

      where $a > 1$ can be a constant or an independent random variable

5. end(1).

In each of these dynamically weighted MCMC sampler variants, θ is a variable parameter that can depend on previous values of (X, Π). The augmented sample chains eventually escape local modes (maxima in the probability distribution) by increasing the sample's associated weight Π. In [68] it is assumed that $\theta =$ constant, although theoretically it may be varied. The new acceptance/rejection steps of these two sampling types cause the equilibrium distribution (if it exists) of the samples to deviate from the target distribution ϱ. However, the principle of invariance with respect to importance weighting (IWIW), introduced in [127], is applicable to the R-type sampling move (and approximately, for the Q-type move).

The IWIW principle (taken from [68]) states that the joint distribution $f(x, \pi)$ of (X, Π) is "correctly weighted" with respect to ϱ, if $\sum_\pi \sum \pi f(x, \pi) \propto \varrho(x)$. Therefore the target distribution can be represented by the weighted samples, up to a scale factor. A transition rule is said to be IWIW if it maintains the correctly weighted property for the joint distribution of (x, π), when the initial joint distribution is correctly weighted.

In [68], it is shown that the R-type move is IWIW. Since we wish to compare PMC to MCMC for exploring the posterior distribution of our BRDF parameters, we implemented the R-type and Q-type dynamically weighted MCMC samplers and compared the performance with the other methods. The proof that the IWIW property applies to the R-type sampler relies on taking an expectation over the possible sample weights for the new sample at each iteration. The joint probability distribution over possible values for (x, π) is $f_1(x, \pi)$, while the probability for values (y, π') (the next sample in the chain) is $f_2(y, \pi')$. The joint transition probability for a sample x moving to y is $T(x, y)$. $I(\cdot)$ is the indicator function which returns 1 when its argument is true. The proof offered in [68] that R-type

sampling is IWIW follows:

$$
\begin{aligned}
&\sum \pi' f_2(y, \pi') \\
&= \sum_{\pi'} \Big\{ \sum_x \sum_\pi \pi' f_1(x, \pi) I[\pi' = \pi r(x, y) + \theta] \\
&\quad \times T(x, y) \frac{\pi r(x,y)}{\pi r(x,y)+\theta} + \sum_z \sum_\pi \pi' f_1(y, \pi) \\
&\quad \times I\Big[\pi' = \frac{\pi(\pi r(y,z)+\theta)}{\theta}\Big] T(y, z) \frac{\theta}{\pi r(y,z)+\theta} \Big\} \\
&= \sum_x \sum_\pi f_1(x, \pi) T(x, y) \frac{\pi r(x,y)}{\pi r(x,y)+\theta} (\pi r(x, y) + \theta) \\
&\quad + \sum_z \sum_\pi f_1(y, \pi) T(y, z) \frac{\theta}{\pi r(y,z)+\theta} \frac{\pi(\pi r(y,z)+\theta)}{\theta} \\
&= \sum_x \sum_\pi \pi f_1(x, \pi) \frac{\varrho(y)T(y,x)}{\varrho(x)} + \sum_\pi \sum_z \pi f_1(y, \pi) T(y, z) \\
&= \sum_x c_1 \varrho(y) T(y, x) = c_1 \varrho(y) = 2 c_1 \varrho(y).
\end{aligned}
\tag{5.50}
$$

This proves that R-type sampling generates a Markov Chain with sample weights in proportion to the probability values of the target distribution at the locations of the samples. In [68], a similar type of reasoning is presented to establish that Q-type sampling, when $\theta > 0$, generates a distribution which *approximately* satisfies the IWIW principle. This reasoning follows:

$$
\begin{aligned}
&\sum \pi' f_2(y, \pi') \\
&= \sum_{\pi'} \Big\{ \sum_x \sum_\pi f_1(x, \pi) T(x, y) \min(1, \tfrac{\pi r(x,y)}{\theta}) \pi' + \\
&\qquad \sum_\pi \sum_z a \pi f_1(y, \pi) T(y, z) q_\pi(y, z) \Big\} \\
&= \sum_x \Big\{ \sum_\pi f_1(x, \pi) T(x, y) \pi r(x, y) \Big\} + a \sum_\pi q_\pi(y) f_1(y, \pi) \\
&= \sum_x c_1 \varrho(x) T(x, y) \frac{\varrho(y)T(y,x)}{\varrho(x)T(x,y)} + R_a = c_1 \varrho(y) + R_a,
\end{aligned}
\tag{5.51}
$$

where $q_\pi(y, z)$ is the probability of rejection when the chain proposes a move from y to z, and $q_\pi(y)$ is the total rejection probability for moving away from sample location y, with

$$
R_a = a \sum_\pi \pi q_\pi(y) f_1(y, \pi).
\tag{5.52}
$$

Following the analysis of [68], when $q_\varrho(y)$ is approximately constant, then $R_a \approx a c_1 \varrho(y)$, which approximately satisfies the IWIW property. Also, if $\sum_\pi \pi f_1(y, \pi)$ is large, the residue $R_a \approx 0$, in which case the IWIW property is also approximately met. In the case where $\theta = 0$, all the sample moves are accepted, and the two moves become equivalent. In this case, the IWIW property is satisfied, as is shown by

$$
\begin{aligned}
\sum \pi' f_2(y, \pi') &= \sum_x \sum_\pi f_1(x, \pi) T(x, y) \pi r(x, y) \\
&= \sum_x c_1 \varrho(x) \frac{\varrho(y)T(y,x)}{\varrho(x)T(x,y)} = c_1 \varrho(y).
\end{aligned}
\tag{5.53}
$$

Interestingly, one of the differences with setting $\theta > 0$ is that the normalizing constant changes from c_1 to $2c_1$, although the IWIW property is maintained in both cases.

Next, as in [68], the behaviour of these sampling steps for $\theta = 0$ and $\theta = 1$, for the case where $T(x, y)$ is reversible, is described. The descriptions are taken almost exactly from [68].

5.5.3   Case 1: $\theta = 0$

When $\theta = 0$, the Q-type and R-type moves become identical. In this case, all proposed moves are accepted. The weight of the new sample Π' given the previous sample weight Π is

$$\Pi' = \Pi r(x, y). \tag{5.54}$$

Given that $g(x)$ is the invariant distribution of $T(x, y)$, and $g(x, y) = g(x)T(x, y)$ is the joint transition distribution of the two steps taken consecutively, if we define the weighting function to be $u(x) = \varrho(x)/g(x)$, where $g(x)$ is the importance sampling distribution, the update formula for the sample weights may be written as

$$\Pi' = \Pi \frac{u(y)g(y, x)}{u(x)g(x, y)}. \tag{5.55}$$

Therefore, if one has a reversible Markov chain where $g(x, y) = g(y, x)$, and if the first sample is $(X_0 = x_0, \Pi_0 = c_0 u(x_0))$, then for any $i > 0$, it is true that $\Pi_i = c_0 u(X_i)$. The weights in this Monte Carlo Markov Chain are identical to those generated by a standard importance sampler using the importance function g.

5.5.4   Case 2: $\theta = 1$

With $\theta = 1$, the Q-type sampler converges to a standard importance sampler, with a trial pdf $g(x)$. Any weight Π corresponding to sample X will converge to a degenerate distribution about $c_0 u(x)$ (again $u(x) = \varrho(x)/g(x)$). This is because if $u_0 = \min_y(\varrho(y)/g(y))$, once the sample (x, π) satisfies $\pi = c_0 u(x)$, with $c_0 u_0 \geq 1$ for some c_0, the proposed transition y will always be accepted. The new weight will be $c_0 u(y)$. The weight will stabilize at $\pi(x) = c_0 u(x)$ as soon as $c_0 u_0 \geq 1$, and the equilibrium distribution of x will be $g(x)$. The weighting process over the sample chain for any starting value Π will therefore climb until

$$\Pi_i \geq \max_{y:T(X_i, y) > 0} \frac{u(X_i)}{u(y)}, \tag{5.56}$$

after which the weight process stabilizes. It is shown by a simple example that the weights of the R-type sampler tend to increase monotonically. With $T(x, y)$ symmetric and $\varrho(\cdot)$ uniform on $\mathcal{X}$, the weights are

$$\Pi' = \begin{cases} \pi + 1 & \text{if } u \leq \frac{\pi}{\pi+1} \\ \pi(\pi + 1) & \text{otherwise,} \end{cases}$$

in which case the Π process increases to infinity with probability 1. By multiplying with a random value $V \sim (1 - \delta, 1 + \delta)$, (if δ is chosen such that $E(\log(V))$ is not too small), the stability of the

weight process Π can be recovered:

$$\Pi_{i+1} = \begin{cases} V(\pi r(x,y)+1) & \text{if accepted} \\ V(\pi(\pi r(x,y))+1) & \text{if rejected.} \end{cases}$$

An argument for this is supplied in [68].

5.6 The theory of Population Monte Carlo

According to [19], 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 the MCMC and importance sampling were combined (e.g. [76], [68]).

In our implementation, there is little difference between partitioned particle filtering (over a single time step) and PMC. In [19], 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:

$$\varrho^{\otimes n}(\theta_1,...,\theta_n) = \prod_{i=1}^{n} \varrho(\theta_i). \tag{5.57}$$

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 $\theta^t = (\theta_1^t, ..., \theta_n^t)$, (each sample has been drawn from g_i^t) is reweighted by

$$\pi_i^t = \frac{\varrho(\theta_i^t)}{g_i^t(\theta_i^t)}, \qquad i = 1, ..., n, \tag{5.58}$$

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

$$\mathfrak{J}^t = \frac{1}{n} \sum_{i=1}^{n} \pi_i^t h(\theta_i^t) \tag{5.59}$$

are unbiased estimators of $E_\varrho[h(\theta)]$ at every iteration t, for every integrable function $h(\cdot)$. In fact,

$$
\begin{aligned}
E[\pi_i^t h(\theta_i^t)] &= \int \int \frac{\varrho(\theta)}{g_i^t(\theta|\mathfrak{C})} h(\theta) g_i^t(x|\mathfrak{C}) dx \ u(\mathfrak{C}) d\mathfrak{C} \\
&= \int \int h(x) \varrho(\theta) d\theta \ u(\mathfrak{C}) d\mathfrak{C} = E_\varrho[h(X)],
\end{aligned}
\tag{5.60}
$$

where $\mathfrak{C}$ indicates the vector of past random variables which affect g_i^t, and $u(\mathfrak{C})$ refers to its distribution. Assuming that the variances

$$
\text{var}(\pi_i^t h(\theta_i^t))
\tag{5.61}
$$

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

$$
\text{var}(\mathfrak{J}^t) = \frac{1}{N^2} \sum_{i=1}^N \text{var}(\pi_i^t h(\theta_i^t)),
\tag{5.62}
$$

implies that the importance-weighted terms are always uncorrelated. Since distributions ϱ are often unscaled and unnormalized, one can instead use

$$
\pi_i^t \propto \frac{\varrho(\theta_i^t)}{g_i^t(\theta_i^t)}, \qquad i = 1, \ldots, N.
\tag{5.63}
$$

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 [96], 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 $(\theta_1^t, ..., \theta_N^t)$ 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 $(y_1^t, ..., y_N^t)$ resulting from this resampling operation is similar to an i.i.d. sample taken from the distribution $\varrho^{\otimes n}(\theta_1, ..., \theta_N)$.

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 [97], 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 [26]) 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. Generate $(I_i^0)_{1 \leq i \leq N} \sim M(1, \overrightarrow{\pi}^0)$

    4. Set $(\theta_{i,\text{new}}^0)_{1 \leq i \leq N} = (\theta_{I_i^0}^0)_{1 \leq i \leq N}$.

- 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. Generate $(I_i^n)_{1 \leq i \leq N} \sim M(1, \overrightarrow{\pi}^n)$

    4. Set $(\theta_{i,\text{new}}^n)_{1 \leq i \leq N} = (\theta_{I_i^n}^n)_{1 \leq i \leq N}$.

As before, $\overrightarrow{\pi}^n$ indicates the normalized set of sample weights π_i^n, for $1 < i < N$. I is a sample index, and M is the multinomial sampling function (see Appendix A). Note that the mathematical functioning of partitioned particle filtering over a single time step with N iterations, assuming there are no time dynamics for state vector transitioning (as indicated by the $*h(\cdot)$ in the previous section on PPF), is identical to N iterations of Population Monte Carlo, if the incident distribution over $\mathbf{X}$, i.e. $p(\mathbf{X}^t|\mathbf{X}^{t-1})$, is uniform. In PPF, the motion prior (or state dynamics) for each time step, at $t > 0$, takes the form $p(\mathbf{X}^t|\mathbf{X}^{t-1})$, and at $t = 0$, $p(\mathbf{X}^t|\mathbf{X}^{t-1}) = p(\mathbf{X}^0)$.

The objective in using the Population Monte Carlo sampler for parametric BRDF extraction is to simultaneously derive a sample set that is a good representation on the underlying BRDF parameters, and one in which the best (maximum a posteriori) sample is reliably close to the true parameter values for the parametric BRDF model. Therefore, using synthetically generated surfaces, one can compare a PMC sampler with a MCMC sampler in terms of the rate at which the MAP sample will reliably come within a threshold of the true parameter values.

To do importance sampling on the distribution of the parameters of the BRDF model, we use the pseudolikelihood of Eqn. 4.24 as a target distribution. Thus

$$p(\theta) = \exp(PL(\mathbf{X}|\theta)) = \prod_{i \in S} p(x_i|x_{\mathcal{N}_i}, \theta) = \prod_{i \in S} \frac{\exp(-U(x_i, x_{\mathcal{N}_i}, \theta))}{\sum_{s \in \mathcal{L}} \exp(-U(s, x_{N_i}, \theta))}, \qquad (5.64)$$

where $\mathcal{L}$ is the set of possible labels which the random variable x_i can take on, and where for each corner vertex node i,

$$U(x_i, x_{\mathcal{N}_i}, \theta) = \sum_{\substack{j \in \mathcal{N}_i, \\ k \in \mathcal{N}_i, k > j}} \psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta) \qquad (5.65)$$

with

$$\psi_{ijk}(x_i, x_j, x_k, y_{ijk}, \theta) = |y_{ijk} - \text{BRDF}(i, j, k, x_i, x_j, x_k, \vec{L}, \mathbf{P}, \theta)|/\sigma_b, \tag{5.66}$$

where σ_b is a user-definable parameter which governs the sharpness of the energy function.

If the number of corner vertex nodes is large, it can be inefficient to use an importance function which calculates the pseudolikelihood over the entire lattice of corner vertex node sites, if it requires too much computation. A single location on the lattice does not provide a strong enough importance function if it is simply used directly. We propose using a "sub" pseudolikelihood, which uses a product of the individual site probabilities over a certain part of the lattice, for example $k = 1000$ to 20000 sites. An importance sampling step can be done using these k sites, after which the next importance sampling step is done on the next k. This is expressed as

$$g^n(\theta) = \exp(PL^n(\mathbf{X}|\theta)) = \prod_{i=nk}^{(n+1)k-1} p(x_i|x_{\mathcal{N}_i}, \theta)$$

$$= \prod_{i=nk}^{(n+1)k-1} \frac{\exp(-U(x_i, x_{\mathcal{N}_i}, \theta))}{\sum_{s\in\mathcal{L}} \exp(-U(s, x_{N_i}, \theta))}. \tag{5.67}$$

One can also create subsets of the original set $\mathbf{S}$ of corner vertex nodes, which include every k^{th} site on the lattice. The first subset S^1 could include sites $\{0, k, 2k, \dots\}$, the second subset S^2 could then include sites $\{1, k+1, 2k+1, \dots\}$, the third subset S^3 includes sites $\{2, k+2, 2k+2, \dots\}$, and so on. If this is the case, the pseudolikelihood can be written

$$g^n(\theta) = \exp(PL^n(\mathbf{X}|\theta)) = \prod_{i\in S^n} p(x_i|x_{\mathcal{N}_i}, \theta)$$

$$= \prod_{i\in S^n} \frac{\exp(-U(x_i, x_{\mathcal{N}_i}, \theta))}{\sum_{s\in\mathcal{L}} \exp(-U(s, x_{N_i}, \theta))}. \tag{5.68}$$

For multiple geometric surfaces under the same lighting conditions (different reconstructions from different synchronized image pairs), one can move progressively through each surface.

5.7   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

$$\pi_i^{\text{new}} \propto (\pi_i^{\text{old}})^l, \tag{5.69}$$

i.e. for each sample, take the weight of the ith sample and take it to the lth power. After all the weights have been adjusted in this way, the weights $\overrightarrow{\pi}^{\text{new}}$ 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

$$p^{\text{new}}(\theta) = s(p(\theta)). \tag{5.70}$$

The operation of this algorithm on a set of weighted samples is denoted by

$$\{\overrightarrow{\theta}, \overrightarrow{\pi}\}^{\text{new}} = s_p(\{\overrightarrow{\theta}, \overrightarrow{\pi}\}^{\text{old}}), \tag{5.71}$$

or (overloading the notation), its performance on a single sample is written $\pi_i^{\text{new}} = s_s(i, \pi_i^{\text{old}})$.

Using this, one can construct an importance function which is a compounded function of softening and pseudolikelihood, i.e.

$$g^n(\theta|\mathbf{X}) = s(\exp(PL^n(\mathbf{X}|\theta))), \tag{5.72}$$

remembering that we are using $p(\theta|\mathbf{X}) = \exp(PL(\mathbf{X}|\theta))$, assuming there are uniform priors on $\mathbf{X}$ and θ.

This function does not involve probabilistic information on the label nodes $\mathbf{X}$ calculated from a probabilistic dense stereo correspondence algorithm, which we believe it should. In later chapters we modify this pseudolikelihood function to include information on the label probabilities, which are derived from the corresponding dense stereo uncertainty for each reconstructed point, as calculated by a Loopy Belief Propagation algorithm.

Now that we have expounded some sampling methodologies and described their usefulness for evaluating Pseudolikelihoods, we turn to the framework we have used for calculating probabilistic dense disparity fields for stereo pairs of images.

Chapter 6

Stereo reconstructions and uncertainty

In this chapter we show how we construct (smoothed) probabilistic surface reconstructions using Loopy Belief Propagation (LBP), and how to incorporate the uncertainties in the surface reconstruction into the potential energy terms that are used to estimate the MRF parameters which are also the BRDF parameters of the surface.

In this thesis, the method used to derive smooth probabilistic disparity maps, which are used to generate 3-D surface reconstructions of object surfaces, is the accelerated LBP algorithm due to Tappen [112]. The LBP method for dense stereo calculates a posterior density estimation on a MRF, defined on the lattice of corner vertex nodes (which occur between pixels). The basic Loopy Belief Propagation algorithm for dense stereo correspondence calculation is due to [109] and is the source of many of the modern belief propagation based stereo algorithms ([111], [108], [128], [60], [66], [133]).

The approximation of the posterior distributions can be accelerated using the method of [112]. Our implementation of this algorithm varies from the original in that we find the posterior distribution on the disparities of the corner vertex nodes, i.e. the points at the intersection of pixels quadruplets (and pixel pairs if the point is on the edge of the image), rather than on the image pixels themselves, as is usually done in dense stereo correspondence algorithms. One could define the lattice as having one node for each pixel, but having lattice nodes on corner vertex nodes allows for smoother transitioning from the notation developed in previous chapters (and from the notation in [71], [73], [72]).

Belief propagation methods are better suited to reconstructing bubble surfaces, and surfaces with varying height profiles, than graph-cut based methods [112],[110]. This is because the regularity of the energy functions required for graph-cut based optimization causes a bias towards fronto-parallel planar surfaces, i.e. for neighbouring pixels to have identical disparities. Bubbles have smooth curvature, and we found that the bubble and material surface reconstructions calculated using belief propagation methods are visibly more accurate than reconstructions based on dense correspondences calculated

using graph-cut methods.

In addition, most dense stereo correspondence algorithms derive a final "best disparity solution", whereas we believe a probabilistic solution gives a truer account of the uncertainties involved in a surface reconstruction. Our aim is to include these probabilistic disparities in our BRDF parameter estimation method. Our BRDF estimation algorithm would also work using one of the algorithms which solve only for the final MAP estimate for the stereo reconstruction of a surface (such as dense stereo reconstruction via graph-cuts), since the state space for each node label is the same, but in that case we cannot include soft probabilistic information (calculated by the dense stereo correspondence algorithm) about the likelihood of the alternative corner vertex node labels. For example, a graph-cut or Gibbs sampling based MRF optimizer may be used to obtain the MAP labelling on the set of nodes, in which case the corresponding probabilities on the node labels can be set to the uniform distribution if there is no other information available. The pseudolikelihood summation of Eqn. 4.23 would still occur over the local configuration space of each node.

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

This section outlines the general process for creating 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 $\mathbf{A}$ of calibration object
2. Capture stereo images $\mathbf{B}$ of scene under structured lighting (checkerboard pattern)
3. Capture stereo images $\mathbf{I}$ of scene illuminated by white light
4. Calculate and calibrate projection matrices of cameras $\mathbf{P}$ in stereo pair using images $\mathbf{A}$
5. Rectify images $\mathbf{B}$ giving images $\mathbf{I}^R$ using calibrated projection matrices $\mathbf{P}$ (and point correspondence data for images $\mathbf{A}$), using the rectification methods of [36] and [55]
6. Do dense stereo correspondence on rectified images $\mathbf{I}^R$, calculating disparity field $\mathbf{X}$ and probabilities (beliefs) on this field $\mathbf{b}$
7. Use probabilistic dense stereo correspondence information $\mathbf{X}$ and $\mathbf{b}$, along with image intensity information (images $\mathbf{I}$), 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.

In this thesis, steps 1 to 5 were done using standard algorithms. For step 4 (calculation and calibration of camera projection matrices), we used the method of [140]. For step 5 (rectification of stereo images), we used a combination of [55] and [36].
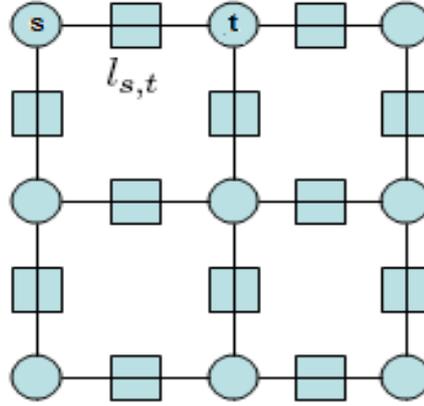
Figure 6.1: The lattice of nodes in $\mathbf{X}$ (round nodes) and the nodes on the dual lattice (square nodes). A node on the dual lattice $l_{s,t}$ is situated between nodes s and t.

We thus proceed immediately to describe step 6, which is done using the dense stereo correspondence algorithm of [109], which has for a long time been a competitive alternative on the Middlebury rankings of stereo reconstruction algorithms (see [53], where its performance may be examined, and where comparisons of alternative dense stereo correspondence algorithms can be viewed).

6.2 MRF formulation for dense stereo correspondence estimation

The description of [109] is followed almost exactly in describing the LBP application to dense stereo correspondence estimation. In this description, the reference view is the first image in the image pair. Stereo vision is modeled by three coupled MRFs defined on the lattice $\mathbf{S}$ of corner vertex nodes. It will be shown how these three MRFs, defined on sets of random variables $\mathbf{D}, \mathbf{L}$ and $\mathbf{O}$, can be combined into a single MRF. $\mathbf{D}$ is the disparity field of the reference view (the first image in the pair), $\mathbf{L}$ is a spatial line process on the dual lattice (shown in Fig. 6.1), and represents explicitly the presence or absence of depth discontinuities in the reference view. $\mathbf{O}$ is a spatial binary process, which indicates occluded points in the reference view. Note that the nodes on the lattice $\mathbf{S}$ are ordered.

Using Bayes' rule, we specify the joint posterior probability over $\mathbf{D}, \mathbf{L}$ and $\mathbf{O}$, given a pair of rectified stereo images, $\mathbf{I}^R = (\mathbf{I}^{R1}, \mathbf{I}^{R2})$, where $\mathbf{I}^{R1}$ is the reference (left) image. This gives the Bayesian formulation:

$$p(\mathbf{D}, \mathbf{L}, \mathbf{O}|\mathbf{I}^R) = \frac{p(\mathbf{I}^R|\mathbf{D}, \mathbf{L}, \mathbf{O})p(\mathbf{D}, \mathbf{L}, \mathbf{O})}{p(\mathbf{I}^R)}. \tag{6.1}$$

This model draws on the model of [39], where $\mathbf{D}$ and $\mathbf{L}$ are two coupled MRFs. The disparity field $\mathbf{D}$ represents the set of random variables to estimate, and $\mathbf{L}$ represents the discontinuities of those

variables, across the lattice. Assuming that $\mathbf{I}^R$ is independent of $\mathbf{L}$ (since the line discontinuities are not observed explicitly in the image), this becomes

$$p(\mathbf{I}^R|\mathbf{D}, \mathbf{L}, \mathbf{O}) = p(\mathbf{I}^R|\mathbf{D}, \mathbf{O}). \tag{6.2}$$

For the sake of simplicity one may ignore the dependence between $\mathbf{O}$ and $\{\mathbf{D}, \mathbf{L}\}$ although in fact the occlusion between image regions is dependent on the relative depths of the surfaces in the images. Nevertheless, with this assumption,

$$p(\mathbf{D}, \mathbf{L}, \mathbf{O}) = p(\mathbf{D}, \mathbf{L})p(\mathbf{O}). \tag{6.3}$$

The stereo model is now

$$p(\mathbf{D}, \mathbf{L}, \mathbf{O}|\mathbf{I}^R) = \frac{p(\mathbf{I}^R|\mathbf{D}, \mathbf{O})p(\mathbf{D}, \mathbf{L})p(\mathbf{O})}{p(\mathbf{I}^R)}. \tag{6.4}$$

A match measure $F(s, d_s, \mathbf{I}^R)$ is defined, which has as arguments a site s, a disparity d_s, and the image data $\mathbf{I}^R$, and returns a dissimilarity measure on the regions (windows) about s and d_s:

$$F(s, d_s, \mathbf{I}^R) = 1 - d(s, d_s, \mathbf{I}^R), \tag{6.5}$$

where $d(s, d_s, \mathbf{I}^R)$ is a match measure between sites s in the first image $\mathbf{I}^{R1}$ and the corresponding site given the disparity d_s in the second image $\mathbf{I}^{R2}$, given the image data $\mathbf{I}^R$. The disparity value d_s is measured in pixels, and in our implementation takes on a value in the range $-20 \leq d_s \leq 20$, $d_s \in \mathbb{Z}$. In [109], the pixel dissimilarity measure of [11] is used. For our purposes any method which returns a value in the range [0;1] is reasonable for calculating $d(s, d_s, \mathbf{I}^R)$: the method must return values between 1 (for a perfect match) and 0 (for a poor match). Some region matching measures are described in Appendix B. This gives

$$p(\mathbf{I}^R|\mathbf{D}, \mathbf{O}) \propto \prod_{s \notin O} \exp(-F(s, d_s, \mathbf{I}^R)/\sigma_f), \tag{6.6}$$

where $s \notin O$ indicates that the probability is calculated using only the non-occluded sites, (O is the set of occluded sites), and σ_f is a factor which makes large pixel dissimilarities have a low probability.

The first order neighbourhood of sites is defined as the Ising neighbourhood, where a site's neighbours include only the site immediately above it, to the left of it, to the right of it and below it, as depicted in Fig. 6.2. The first order neighbourhoods G_s and $\mathcal{N}_s = \{t|t > s, t \in G_s\}$ are thus defined for each site s on the lattice $\mathbf{S}$. The ordering on sites s on the lattice $\mathbf{S}$ allows the expression $\{t|t > s, t \in G_s\}$ to include only sites which are in G_s and which are also higher in the ordering than s. The use of this expression thus allows one to avoid double counting pairwise interaction potential terms.

The line process which is used to preserve continuity in the disparity field is defined as a set of binary random variables $l_{s,t}$, $s \in \mathbf{S}, t \in \mathcal{N}_s$ on the dual lattice. If $l_{s,t} = 1$, this indicates the presence of a
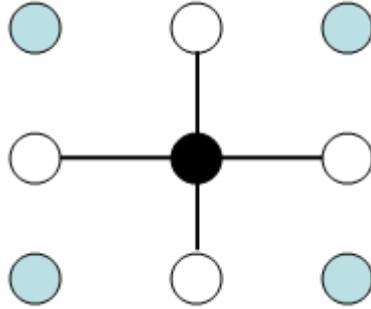
Figure 6.2: The white round nodes are the Ising neighbours of the black node.
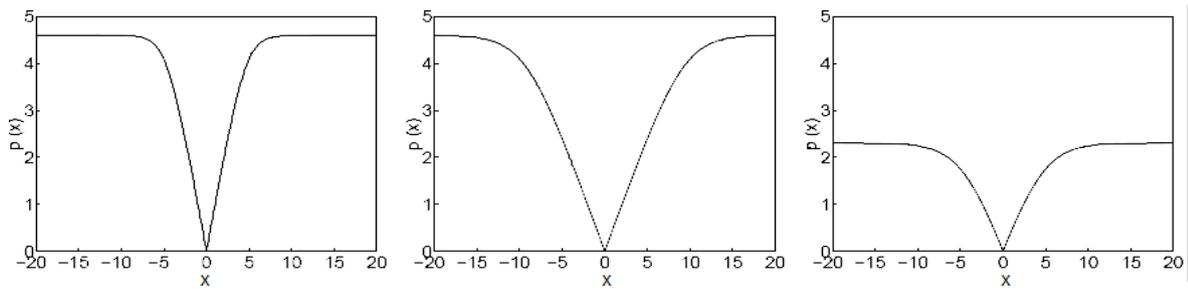
Figure 6.3: Robust function $\rho(x) = -\ln((1-e)\exp(-\frac{|x|}{\sigma}) + e)$, derived from the Total Variance model [101] (this diagram taken from [109]). The function plotted on the left has $e = 0.01$, $\sigma = 1.0$. The function shown in the middle has $e = 0.01$, $\sigma = 2.0$, the function shown on the right has $e = 0.1$, $\sigma = 2.0$.

disparity discontinuity between the labels d_s and d_t; if $l_{s,t} = 0$, there is no disparity discontinuity. The outlier process $\mathbf{O}$ is also a set of binary random variables. The prior distribution of Eqn. 6.4 is expanded as:

$$p(\mathbf{D}, \mathbf{L}, \mathbf{O}) \propto \left(\prod_s \prod_{t \in \mathcal{N}_s} \exp(-\psi_c(d_s, d_t, l_{s,t})) \right) \left(\prod_s \exp(-n_c(o_s)) \right), \qquad (6.7)$$

where $\psi_c(d_s, d_t, l_{s,t})$ is the joint clique potential function (spatial interaction term) of d_s, d_t and $l_{s,t}$, and $n_c(o_s)$ is the clique potential function of o_s. Spatial interaction may be incorporated by defining

$$\psi_c(d_s, d_t, l_{s,t}) = \psi(d_s, d_t)(1 - l_{s,t}) + \gamma(l_{s,t}). \qquad (6.8)$$

The functions $\psi(d_s, d_t)$ and $n_c(o_s)$ can be specified manually. In the above, $\psi(d_s, d_t)$ penalizes the different assignments of neighbouring sites when no discontinuity is present between them, and $\gamma(l_{s,t})$ penalizes the occurrence of a discontinuity between sites s and t. It is usual to set $\gamma(l_{s,t}) = 0$. When the previous formulae are combined,

$$p(\mathbf{D}, \mathbf{O}, \mathbf{L}|\mathbf{I}^R) \propto \left(\prod_{s \notin O} \exp(-F(s, d_s, \mathbf{I}^R)) \right) \left(\prod_s \exp(-n_c(o_s)) \right)$$
$$\left(\prod_s \prod_{t \in \mathcal{N}_s} \exp(-(\psi(d_s, d_t)(1 - l_{s,t} + \gamma(l_{s,t})))) \right). \qquad (6.9)$$

The term $n_c(o_s)$ is then simplified by ignoring the spatial interaction of the occlusion sites, giving

$$n_c(o_s) = n(o_s). \qquad (6.10)$$

In [109] it is noted that the form of $n_c(o_s)$ should be

$$n_c(o_s) = n(o_s) + \sum_{t \in \mathcal{N}_s} n'(o_s, o_t), \qquad (6.11)$$

where $n(o_s)$ is a single-site clique potential function, which penalizes occlusion, and $n'(o_s, o_t)$ penalizes different occlusion labels o_s and o_t for sites s and t. However, this is ignored.

Using the methodology of [13] for combining line processes with robust statistics, the MAP formulation of the problem is rewritten in [109] as

$$\max_{\mathbf{D}, \mathbf{L}, \mathbf{O}} p(\mathbf{D}, \mathbf{L}, \mathbf{O}|\mathbf{I}^R) = \max_{\mathbf{D}} \left\{ \max_{\mathbf{O}} \prod_s \exp(-(F_s, d_s, \mathbf{I}^R)(1 - o_s) + n(o_s)) \right.$$
$$\left. \max_{\mathbf{L}} \prod_s \prod_{t \in \mathcal{N}_s} \exp(-(\psi(d_s, d_t)(1 - l_{s,t}) + \gamma(l_{s,t}))) \right\}. \qquad (6.12)$$

With the problem expressed in this form, the task is to find the set of labels for $\mathbf{D}$, $\mathbf{L}$, and $\mathbf{O}$, which maximizes the probability $p(\mathbf{D}, \mathbf{O}, \mathbf{L}|\mathbf{I}^R)$ in Eqn. 6.9. However, the problem can be simplified. The

spatial binary processes of $l_{s,t}$ and o_s are converted to analog processes, with $0 \leq l_{s,t}^a \leq 1$ and $0 \leq o_s^a \leq 1$, following the paradigm of [13]. It is possible to make the simplification

$$\max_{\mathbf{O}} \prod_s \exp(-(F(s, d_s, \mathbf{I}^R)(1 - o_s^a) + n(o_s^a)))$$
$$= \exp(-\min_{\mathbf{O}} \sum_s (F(s, d_s, \mathbf{I}^R)(1 - o_s^a) + n(o_s^a))), \qquad (6.13)$$

where

$$\min_{\mathbf{O}} \sum_s (F(s, d_s, \mathbf{I}^R)(1 - o_s^a) + n(o_s^a)) \qquad (6.14)$$

is an objective function, describing a robust estimator.

The robust potential energy functions for the robust estimator of Eqn. 6.12 which use the above analog processes [13] are defined as:

$$\psi_d^{\min}(s, d_s) = \min_{o_s^a}(F(s, d_s, \mathbf{I}^R)(1 - o_s^a) + n(o_s^a)) \qquad (6.15)$$

and

$$\psi_p^{\min}(d_s, d_t) = \min_{(l_{s,t}^a)}(\psi(d_s, d_t)(1 - l_{s,t}^a) + \gamma(l_{s,t}^a)). \qquad (6.16)$$

Note that the robust function $\psi_p^{\min}(\cdot, \cdot)$ is not dependent on the precise location on the lattice. Finally, the posterior probability of the disparity field is defined by including the two robust functions:

$$p(\mathbf{D}|\mathbf{I}^R) \propto \left(\prod \exp(-\psi_d^{\min}(s, d_s)) \right) \left(\prod_s \prod_{t \in \mathcal{N}_s} \exp(-\psi_p^{\min}(d_s, d_t)) \right). \qquad (6.17)$$

In this way, the two analog processes are absorbed into a single outlier process, and three coupled Markov random fields are converted into a single Markov random field, using robust estimators.

6.2.1 Robust statistics

Robust estimators are now briefly described, since the potential functions of our Markov random field are defined using this technique (we do not use the robust functions of Eqn. 6.15 and 6.16, they were derived for illustration). Robust estimators are described in detail in [13], and arose out of the need to prevent too great an influence of outliers on an objective function. Hampel et al. in [48] describe the goals of robust statistics as

1. to describe the structure best fitting the data

2. to identify the outlying data points.

The task of robust estimation is to find the values in a parameter vector $\mathbf{a} = [a_0 \dots a_N]$ that provide the best fit for a model $u(s, \mathbf{a})$, given data measurements $\mathbf{d} = \{d_0 \dots d_S\}$. The model may be fitted by minimizing the residual errors $d_s - \mathbf{u}(s, \mathbf{a})$:

$$\min_{\mathbf{a}} \sum_{s \in S} \rho(d_s - \mathbf{u}(s; \mathbf{a}), \sigma_s), \tag{6.18}$$

where s iterates over the data points, σ_s is a variable scale parameter, and ρ is the error norm. If the measurement errors are normally distributed, the optimal ρ-function is quadratic in the residual errors:

$$\rho(d_s - \mathbf{u}(s; \mathbf{a}), \sigma_s) = \frac{(d_s - \mathbf{u}(s; \mathbf{a}))^2}{2\sigma_s^2}. \tag{6.19}$$

This is the standard least-squares estimation problem. If a quadratic error function is used, then the influence of outliers increases without bound. Many robust functions are designed to have cost functions that fall away from the quadratic, such as the truncated quadratic estimator of [15]. The choice of different robust functions results in different robust estimators. An estimator that is less sensitive to outliers is said to be more robust. When the quadratic function of Eqn. 6.19 is used, the outliers contribute too much to the error. In [48], the behaviour of robust functions is analysed using influence functions which are proportional to the derivative of the ρ-function. If the quadratic ρ-function is x^2, then the influence function λ is $2x$. This means that for least-squares estimators, the influence of outliers increases linearly and without bound. One alternative to the least squares estimator is to use Huber's minimax estimator [91]:

$$\rho_\epsilon(x) = \begin{cases} x^2/2\epsilon + \epsilon/2 & \text{if } |x| \le \epsilon \\ |x| & \text{if } |x| > \epsilon \end{cases} \tag{6.20}$$

with

$$\lambda_\epsilon(x) = \begin{cases} x/\epsilon, & \text{if } |x| \le \epsilon \\ \text{sign}(x) & \text{if } |x| > \epsilon. \end{cases} \tag{6.21}$$

In this estimator, the ρ-function increases like x^2 for small errors, and increases like $|x|$ for larger errors. The Lorentzian estimator has a different characteristic:

$$\rho(x, \sigma) = \log\left(1 + \frac{1}{2}(\frac{x}{\sigma})^2\right), \qquad \lambda(x, \sigma) = \frac{2x}{2\sigma^2 + x^2}, \tag{6.22}$$

which allows the influence of residual errors to decrease when the absolute value is greater than a certain threshold.

Two alternative robust functions which are derived from the Total Variance model of [101] are used here for dense stereo correspondence (they are also used in [109]). They are derived from an objective function with the potential function specified as $\rho(x) = |x|$, and are truncated into the functions:

$$\psi_d(s, d_s) = -\ln\left((1 - e_d) \exp\left(-\frac{|F(s, d_s, \mathbf{I}^R)|}{\sigma_d}\right) + e_d\right), \tag{6.23}$$

and

$$\psi_p(d_s, d_t) = -\ln\left((1 - e_p)\exp\left(-\frac{|d_s - d_t|}{\sigma_p}\right) + e_p\right), \tag{6.24}$$

where $e_d, \sigma_d, e_p, \sigma_p$ are definable parameters for the robust functions. If these functions are substituted into Eqn. 6.17, then a function of similar form to that maximized in Eqn. 6.12 is obtained. Plots of these robust functions (Eqns. 6.23 and 6.24) for different parameter values are shown in Fig. 6.3. These are the functions used to generate smooth probabilistic disparity maps in our experiments (and in [109]), and to calculate beliefs on the states of any corner vertex node.

The posterior probability distribution over the disparity field $\mathbf{D}$ is a Markov random field. This Markov random field on $\mathbf{D}$ is now converted into a loopy Bayesian network/Markov network consisting of the set of hidden nodes $\mathbf{X}$ and visible (observation nodes) $\mathbf{Y}$. Each variable node x_s in the Markov network is a discrete random variable representing the disparity of the corner vertex node at site s. Each such node also has connected to it (topologically) a visible node which represents a direct observation on the probability distribution over the labels of the discrete random variable x_s. The visible nodes are so called because the probability vectors they contain are known exactly, unlike the variable nodes where the state probabilities must be calculated through inference algorithms [84].

Each observation node (visible node) y_s typically contains probabilistic values for match measures for site s in the reference image corresponding to various points (in the horizontal scan line, within the range of disparities) in the rectified target image, as indicated by the disparity value (node label) x_s. We denote $\mathbf{X} = \{x_s\}$, i.e. $\mathbf{X}$ is the set of random variables $\{x_s\}\forall s \in \mathbf{S}$, such that each x_s can take on any label in $\mathcal{L}$, and each label corresponds to a disparity value. We also define the set of visible data nodes $\mathbf{Y} = \{y_s\}$, i.e. each random variable x_s has an associated visible data node y_s for $1 \leq s \leq N$, where N is the number of sites on the lattice $\mathbf{S}$ (see Fig. 6.4 for a graphical depiction of the connectivity of the visible and hidden nodes). The visible data y_s for a site s on the lattice $\mathbf{S}$ is a vector of probabilities. The MRF may be defined using clique potential functions:

$$p(\mathbf{X}|\mathbf{Y}) \propto \left(\prod_{s,t:s>t,t\in\mathcal{N}_s} v_{st}(x_s, x_t)\right)\left(\prod_s v_s(x_s, y_s)\right) \tag{6.25}$$

where $\mathbf{X}$ represents the set of random variables describing the disparity field, so

$$v_{st}(x_s, x_t) = \exp(-\psi_p(x_s, x_t)) \tag{6.26}$$

and $\mathbf{Y}$ is effectively the image data, thus

$$v_s(x_s, y_s) \propto \exp(-\psi_d(s, x_s)), \tag{6.27}$$

which is (almost exactly) the original formulation of [109]. The multiplication iterators $s, t : s > t, t \in \mathcal{N}_s$ indicate a use of the ordering on the node sites which prevents the double counting of potential
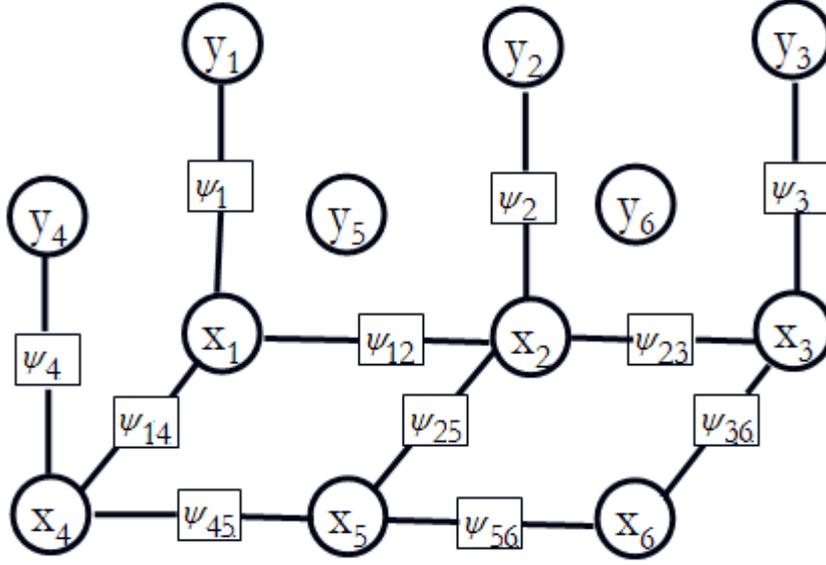
Figure 6.4: A depiction of the pairwise clique potential system used to smooth probabilistic disparity maps. The round nodes labeled **x** are the random variables (corner vertex nodes) in the rectified image; the round nodes labeled **y** are the local (visible) observation nodes for each corner vertex node **x**. Pairwise potential terms are ψ_{ij}, local evidence potential terms are ψ_i.

energy terms. However, in keeping with our established notation for clique potentials (ψ) and total clique potentials per node per clique type (U), the previous three equations may be rewritten as

$$p(\mathbf{X}|\mathbf{Y}) \propto \left(\prod_s \exp(-U_s^B) \right) \left(\prod_s \exp(-U_s^C) \right) \tag{6.28}$$

where

$$U_s^B = \sum_{t:s>t,t\in\mathcal{N}s} \psi_p(x_s, x_t) \tag{6.29}$$

and

$$U_s^C = \psi_d(s, x_s). \tag{6.30}$$

In the above, $\upsilon_{st}(x_s, x_t)$ becomes a compatibility matrix between adjacent nodes x_s and x_t. The second function $\upsilon_s(x_s, y_s)$ is often referred to as the evidence or local evidence on node x_s. Thus, three coupled MRFs defined on the sets of random variables $\mathbf{D}, \mathbf{L}, \mathbf{O}$ have been converted into a single MRF on the random variables $\mathbf{X}$ and image data $\mathbf{Y}$ using the robust functions of Eqns. 6.23 and 6.24.

6.3   Loopy Belief Propagation for dense stereo estimation

In this section the Loopy belief propagation (LBP) and accelerated LBP algorithms for approximate inference on Bayesian networks are described. Belief propagation is an iterative algorithm for approximating the posterior distributions on loopy or non-loopy Bayesian networks. If the graph or network has no cycles, then the inference is exact (i.e. the posterior distributions obtained when the algorithm converges are probabilistically correct). Loopy belief propagation, introduced by [89], works by generalizing the forwards-backwards message passing algorithm to graphs which have cycles. Theoretically, the algorithm is prone to double counting information, to converging to the wrong posterior distribution, and failing to converge. It is found that in many applications, including dense stereo matching, the algorithm works very well. A range of variations on the algorithm has been developed, including ones which guarantee convergence (such as [132],[125]), and ones which improve the accuracy of the posterior distribution by considering embedded trees [120].

If there are cycles in the graph, the algorithm converges instead to something close to the correct posterior distribution. Some work has been done on the convergence properties of the Loopy Belief Propagation (LBP) algorithm on cyclic graphs under various constraints. It is shown in [124] that in loopy networks where the variable nodes are Gaussian, the inference is exact. The inference is also exact when dealing with a network containing a single loop, as is shown in [123]. In some cases, a loopy Bayesian network will exhibit oscillatory behaviour in the posterior distribution on any node and on the messages, as the same information is passed around the loops in the network. One effective message damping method, which reduces the oscillatory behaviour, is proposed in [83], where the messages from each node are adjusted to be a combination of the existing message and the usual update message. In [130] it is shown that the LBP algorithm does not in general converge to the correct posterior in a loopy network, but rather to an equivalent result to the Kikuchi approximation to the Gibbs Free Energy on a lattice of random variables. It is shown in [130] that Belief Propagation is only one of a series of message passing variants. The authors go on to define a more general message passing algorithm called Generalized Belief Propagation.

However, we choose to use an accelerated version of the classical dense stereo Belief Propagation algorithm of [109]. This offers faster convergence with similar performance. We further motivate our choice for using this method over alternative LBP and non-LBP based dense stereo estimation algorithms, by noting that the belief propagation method of [109] was for a long time one of the best performers on the Middlebury stereo data sets [53]. Moreover, many of the best Belief Propagation-based algorithms to appear on the Middlebury rankings were derived from the Belief Propagation method of [109].

6.3.1　Factor node algorithm for Loopy Belief Propagation

A concise description of the factor node algorithm for approximating posterior distributions on loopy networks is the factor graph algorithm, (see [84] for a complete description). Briefly, the variable nodes first send messages to their neighbouring factor nodes:

$$\mu_{x \to f}(x) = \prod_{g \neq f} \mu_{g \to x}(x). \tag{6.31}$$

Then each factor node sends a message to each of its neighbouring variable nodes:

$$\mu_{f \to x}(x) = \sum f(u) \prod_{y \neq x} \mu_{y \to f}(y). \tag{6.32}$$

The above two steps are iterated until convergence or some other stopping criterion is reached. In these equations x and y are variable nodes (corner vertex nodes), while g and f are factor nodes. Each factor node represents a clique potential function on its connected variable nodes. This is the algorithm we used in [72] to obtain an approximate solution to the shape from shading problem, using the same connectivity on the corner vertex nodes as is depicted in Fig. 4.1, where the square nodes are factor nodes and the round nodes are variable nodes.

6.3.2　Loopy Belief Propagation: max-product, parallel message passing protocol

It is possible to do loopy belief propagation without factor nodes, as is described in [89]: this is the method followed here, and in [109].

Given the two MRF potential energy functions $\psi_{st}(x_s, x_t)$ and $\psi_s(x_s, y_s)$, the parallel message passing schedule can be described, using the max-product update rule for approximating the posterior distributions $p(\mathbf{X}|\mathbf{Y})$ for the random variable labels at each of the sites. If $m_{st}(x_s, x_t)$, abbreviated $m_{st}(x_s)$, is the message being sent to node x_t from node x_s, and $m_s(x_s, y_s)$ (abbreviated to $m_s(x_s)$) is the message that observed node y_s sends to node x_s, and if $b_s(x_s)$ is the belief at node x_s, (note that x_s refers both to the node in the network, and to the random variable which that node represents) then the max-product algorithm for belief propagation on the lattice of corner vertex nodes may be described as follows:

Max-product algorithm

1. Initialize all messages to the uniform distribution
2. For $i = 1 \dots T$

　　Update all the messages:

$$m_{st}^{i+1}(x_t) \leftarrow \kappa \max_{x_s} \left(\upsilon_{st}(x_s, x_t) m_s^i(x_s) \prod_{k \in \mathcal{N}_s \setminus t} m_{ks}^i(x_s) \right) \tag{6.33}$$
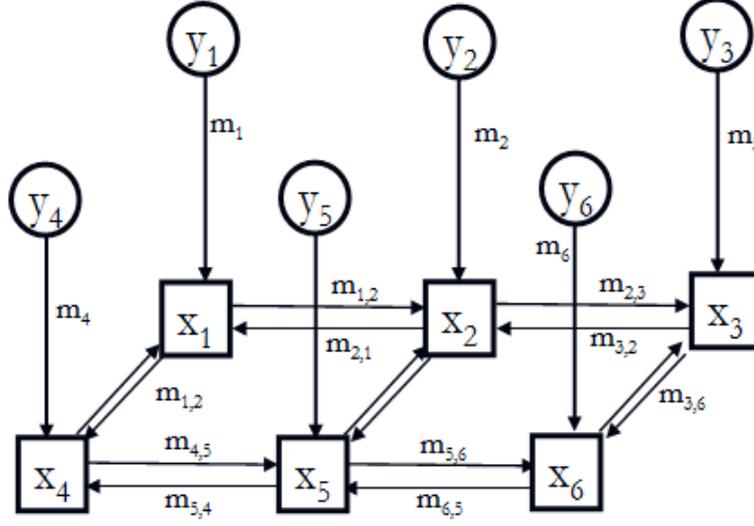
Figure 6.5: A depiction of the message passing algorithm on a lattice. Messages are passed between nodes which are topologically connected. The prior data is contained in the visible (round) nodes labeled 'y', which contain the image data. The discrete random variables (square nodes) represent the disparities at the corner vertex node at each point (corner vertex node) on the lattice. The square region interior to four 'x' nodes represents a pixel. This diagram is taken from [109].

    end(2)

3. Compute final beliefs at time T

$$b_s(x_s) \leftarrow \kappa m_s(x_s) \prod_{k \in \mathcal{N}_s} m_{ks}^T(x_s). \tag{6.34}$$

Then the maximum a posteriori label for a node x_s is found by

$$x_s^{MAP} = \arg \max_{x_k \in \mathcal{L}} b_s(x_k). \tag{6.35}$$

The iterator $k \in \mathcal{N}_s \setminus t$ indicates that messages from all the random variables neighbouring the random variable (corner vertex node) x_s at site s are included, except x_t. A diagram showing the message passing scheme is shown in Fig. 6.5.

In the above algorithm, κ is a normalizing constant which normalizes the message vectors m (makes its elements, which are probabilities, sum to unity). Since all final beliefs $b_s(\cdot)$ are normalized (the elements must sum to unity, since it is a probability vector), the multiplication of intermediate messages by κ can usually be omitted, although it can help to prevent underflow. T is the number of iterations of belief propagation, so step 3 uses the final messages at time T to calculate the posterior beliefs on
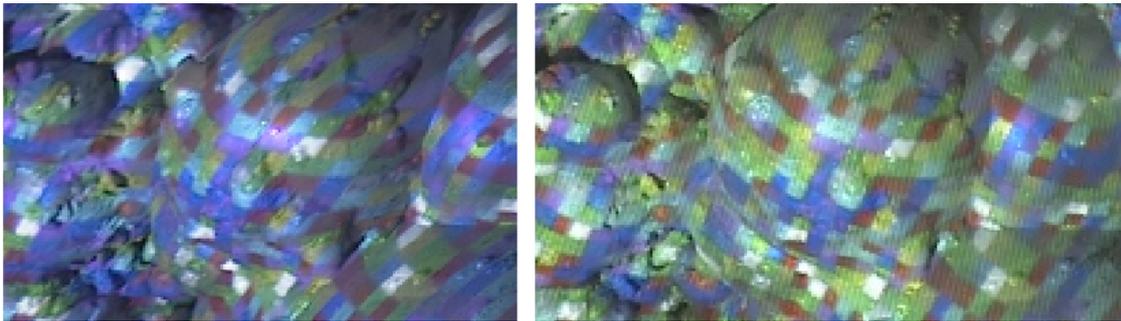
Figure 6.6: A user-selected subregion of the rectified image pair of Fig. 6.7. The right sub-window is wider because of the possibility of matches in the range of disparity, which in this example is 30 pixels.

the node labels. $\mathcal{L}$ is the set of possible labels that a corner vertex node may adopt, and depends on the disparity range selected for the image pair. The term $\upsilon_{st}(x_s, x_t)$ is a square compatibility matrix which is populated with elements calculated according to Eqn. 6.26. The algorithm may also be run until it is detected that there is little or no change in the messages. Note that the product of two messages (vectors containing probabilities) is a componentwise product.

After the Belief Propagation algorithm is run there exists a probability on the disparity for each site on the lattice, that is contained in $b_s(x_s)$. Our MRF method for BRDF extraction uses these probabilities on the labels of corner vertex nodes. In Fig. 6.9 are shown the resulting maximum a posteriori disparity values per pixel in a disparity map for the rectangular sub-window in a rectified image of froth shown in Fig. 6.6. The posterior distribution from which the disparity map was extracted, was calculated using the LBP algorithm which was just described. These sub-windows were extracted from the rectified froth pair shown in Fig. 6.7. It is easy to see that the disparity map generated with LBP is an improvement on the disparity map calculated by using only the window correlation measure, shown in Fig. 6.8. Whether it is more accurate is unknown since there is no other method available to us which gives a perfect groundtruth reconstruction for surfaces which change their shape quickly over time against which we can compare this reconstruction method. It is clear that the LBP smoothed disparity maps are less noisy, and in this sense it is a definite improvement.

An excellent compendium of belief propagation style-algorithms for static and dynamic Bayesian networks may be found in [84].

6.3.3   Match measure

There is a number of different algorithms which can be used for calculating the match measure for Eqn. 6.5, which is incorporated in the observation potential of Eqn. 6.27. These include 2-D correlation, Kullback-Leibler divergence, histogram intersection, mutual information (MI), Earth Mover's Distance
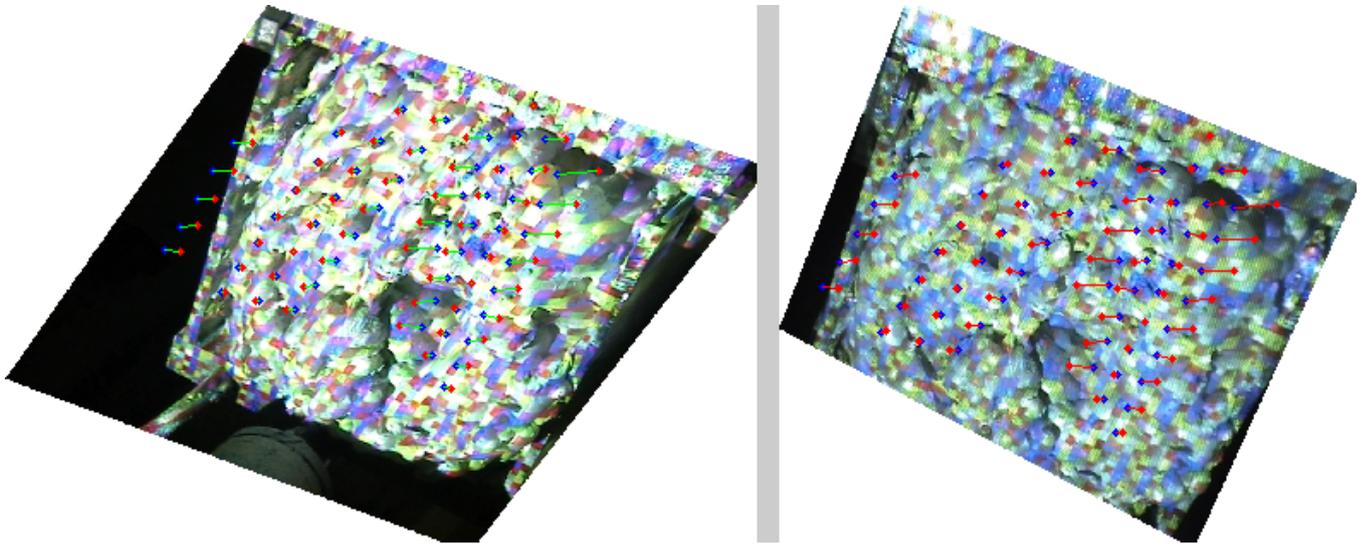
Figure 6.7: A rectified pair of froth surfaces. These surfaces are illuminated by a checkerboard pattern of coloured light, generated by a digital light projector. This coloured pattern facilitates the calculation of accurate disparity maps. The images were rectified using the method of [55]. The final transformed sparse feature correspondences are superimposed on the images.
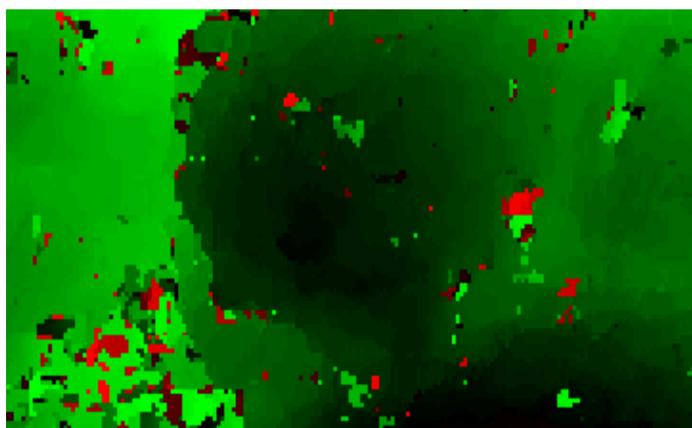
Figure 6.8: MAP estimate for the disparity map, using a 2-D correlation window.
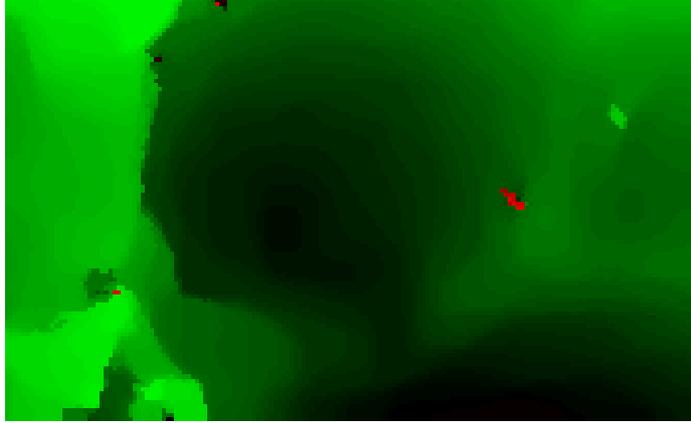
Figure 6.9: The probability map corresponding to the disparity map of Fig. 6.8 is smoothed using Loopy Belief propagation, (max-product updates, parallel message passing schedule). The MAP disparity values are shown above.

and the inner product. Each of these measures performs better under different conditions. In our experiments, we use a dissimilarity measure which uses the simple 2-D correlation measure:

$$r(A, B) = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{\sum_m \sum_n (A_{mn} - \overline{A})^2 \sum_m \sum_n (B_{mn} - \overline{B})^2}}, \qquad (6.36)$$

where $\overline{A}$ and $\overline{B}$ are the averages of pixels in the first and second rectangular image windows respectively. The correlation is converted into a measure which is in the range $[0; 1]$:

$$d(s, d_s, \mathbf{I}^R) = (r + 1)/2, \qquad (6.37)$$

where the image windows A and B are the rectangular image regions about site s in the first rectified image $\mathbf{I}^{R1}$ and the corresponding region in the second rectified image $\mathbf{I}^{R2}$, given the disparity d_s.

The interested reader is referred to Appendix B, where brief descriptions of some alternative window match measures are described.

6.3.4 Accelerated message passing

An enhancement described in [112] is to update all messages in each column and row, traversing the image four times per iteration. First we start at the rightmost column and update the messages for each column left of the rightmost column, in a leftwards direction. An analagous procedure is repeated in the other three directions (down, right, up). This allows the messages calculated at the previous row/column to be used immediately on the next row/column, without waiting for the next time step. In this way, information spreads across the lattice much faster than it does when using the parallel

update schedule of [109], described in the previous section. This accelerated method converges to a local energy minimum faster than the standard LBP algorithm, which works on all nodes on the lattice in parallel. The following algorithm replaces step 2 in the max-product parallel algorithm in Section 6.3.2.

Accelerated Loopy Belief Propagation message update schedule for dense correspondence

In this algorithm, the message $m_{\text{right}}(x_j)$ indicates the message from the node to the right of node x_j, to node x_j (similarly for $m_{\text{up}}(x_j), m_{\text{left}}(x_j), m_{\text{down}}(x_j)$). It is assumed that the lattice has R rows and C columns. A message is excluded if it refers to a node that does not exist.

1. For $n = 1, \ldots, T$ (T is the number of LBP iterations)
2.   For $c = C - 1 \ldots 1$, (from right to left)
3.   For $x_j \in$ variable nodes in column c
4.   Update message from right (if node x_i is to the right of node x_j):

$$m_{\text{right}}(x_j) \leftarrow \max_{x_i} \left(v_{ij}(x_i, x_j) m_i(x_i) m_{\text{right}}(x_i) m_{\text{up}}(x_i) m_{\text{down}}(x_i) \right) \tag{6.38}$$

5.   end(3)
6.   end(2)
7.   For $r = 2 \ldots R$, (from top to bottom)
8.   For $x_j \in$ variable nodes in row r
9.   Update message from above (if node x_i is above node x_j):

$$m_{\text{up}}(x_j) \leftarrow \max_{x_i} \left(v_{ij}(x_i, x_j) m_i(x_i) m_{\text{right}}(x_i) m_{\text{up}}(x_i) m_{\text{left}}(x_i) \right) \tag{6.39}$$

10.   end(8)
11.   end(7)
12.   For $c = 2 \ldots C$, (from left to right)
13.   For $x_j \in$ variable nodes in column c
14.   Update message from left (if node x_i is to the left of node x_j):

$$m_{\text{left}}(x_j) \leftarrow \max_{x_i} \left(v_{ij}(x_i, x_j) m_i(x_i) m_{\text{left}}(x_i) m_{\text{up}}(x_i) m_{\text{down}}(x_i) \right) \tag{6.40}$$

15.   end(13)
16.   end(12)
17.   For $r = R - 1 \ldots 1$, (from bottom to top)
18.   For $x_j \in$ variable nodes in row r
19.   Update message from below (if node x_i is below node x_j):

$$m_{\text{down}}(x_j) \leftarrow \max_{x_i} \left(v_{ij}(x_i, x_j) m_i(x_i) m_{\text{down}}(x_i) m_{\text{left}}(x_i) m_{\text{right}}(x_i) \right) \tag{6.41}$$
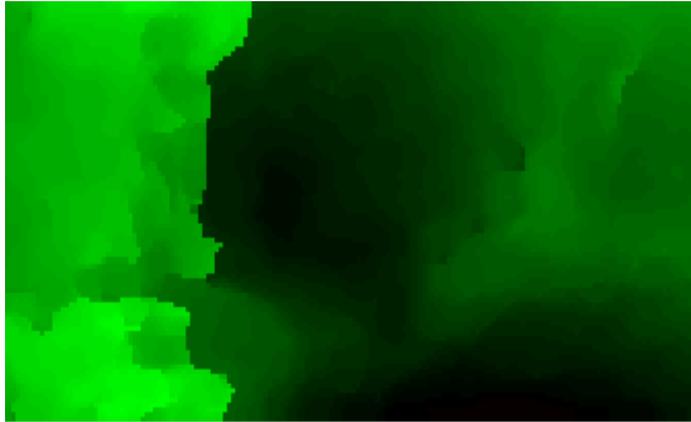
Figure 6.10: The MAP disparity map corresponding to the same disparity probability field as represented in Fig. 6.8 is smoothed using Loopy Belief propagation, (max-product updates, accelerated message passing schedule).

20.   end(18)

21.   end(17)

22. end(1).

For a parallel message passing schedule (as in [109]), with an image of width C, it would take C iterations for information to pass from one side of the image to the other. With accelerated message passing, this happens in one iteration. All other aspects of the accelerated LBP and parallel message passing (previous section) LBP are similar. In Fig. 6.11, we show the resulting maximum a posteriori disparity values per pixel for a rectangular sub-window in a rectified image of froth, calculated using the accelerated LBP algorithm just described. The matter presented in this chapter thusfar does not represent original work, and is presented merely to motivate the use of the accelerated LBP algorithm for dense stereo reconstruction rather than the parallel LBP algorithm. In the latter parts of this chapter we use the derived probabilities using the accelerated LBP method on a stereo pair to define a new potential function on a MRF which includes the BRDF related MRF energy functions as described in Chapter four.

6.4   Calculation of 3-D points from corner vertex node labels

In this section it is shown how to triangulate a pair of points from a set of dense stereo correspondences into the scene. This procedure will be done for each disparity label, for each point in the reference view (first image). The corresponding probability for a particular disparity, for a particular point, is taken to be the probability of the surface passing through the corresponding point in the scene.

In dense stereo reconstruction algorithms it is common, after calculating the MAP disparity map for

the rectified image pair, to project each of the pixels/points into 3-D space. This is necessary for our algorithm since the BRDF functions WARD($\cdot$) and PHONG($\cdot$) assume the ability to calculate the 3-D point locations given the labellings on corner node vertex triplets. Given $\mathbf{H}_1$ and $\mathbf{H}_2$, the rectifying homographies for the cameras for the first and second images respectively, and given $\mathbf{P}_1$ and $\mathbf{P}_2$, the corresponding calibrated projection matrices for the first and second images, one may calculate the location of the 3-D point corresponding to a disparity value of $d_{r(i)}^{R1}$ in the rectified image, (corresponding to corner vertex node i in the unrectified image) using a linear triangulation method.

First it is necessary to find the location of the corner vertex node in the rectified image $\mathbf{I}^{R1}$ corresponding to corner vertex node i in the unrectified image $\mathbf{I}^1$. Assume there is a point in the first unrectified image at pixel (x_i, y_i). First, this point is converted into a vector of homogeneous coordinates $\mathbf{X}_i^{I1} = [x_i \ y_i \ 1]^T$. The point is then transformed by the same rectifying homography that was applied to the first image to rectify it:

$$\mathbf{Y}_{r(i)}^{R1} \simeq \mathbf{H}_1 \mathbf{X}_i^{I1}, \tag{6.42}$$

where the symbol $\simeq$ indicates that the vector to its left is homographically equal to the vector to its right, i.e. when each vector is multiplied by a (different) scalar value, to make its last element unity, then the vectors are the same.

Following this formulation, $r(i)$ is the index of the corner vertex node at $\mathbf{Y}_{r(i)}^{R1}$, in the rectified image. It is possible that due to inaccuracy in calibration or in the calculation of the rectifying homographies, the location contained in the coordinates of $\mathbf{Y}_{r(i)}^{R1}$ does not correspond exactly to a corner vertex node location, i.e. a point between four pixels in the rectified image. If this is the case, the nearest corner vertex node to the coordinates in $\mathbf{Y}_{r(i)}^{R1}$ is selected as the node corresponding to corner vertex node i in the unrectified image. Illustrations of this are given in Figs. 6.12 and 6.13. The inverse relationship also exists

$$\mathbf{X}_i^{I1} \simeq \mathbf{H}_1^{-1} \mathbf{Y}_{r(i)}^{R1}. \tag{6.43}$$

In reality, one may also need to store the offsets of the warped image boundaries when homographically warping an image. If the image is warped such that some of the image points fall on negative coordinates in the image plane, then the offsets must be stored if the parts of the warped image with negative image plane coordinates are not to be discarded. If the offsets for the warped image are (o_x, o_y), (i.e. the image point at $(0, 0)$ corresponds to the image plane point (o_x, o_y) (the warped image is shifted to have all its points at positive pixel coordinates), then the previous two equations become

$$\text{(image plane point)} \quad \mathbf{y} \simeq \mathbf{H}_1 \mathbf{X}_i^{I1}, \tag{6.44}$$

$$\text{(image point)} \quad \mathbf{Y}_{r(i)}^{R1} = \mathbf{y} - [o_x \ o_y \ 0]^T, \tag{6.45}$$

and

$$\mathbf{X}_i^{I1} \simeq \mathbf{H}_1^{-1}(\mathbf{Y}_{r(i)}^{R1} + [o_x \ o_y \ 0]^T). \tag{6.46}$$
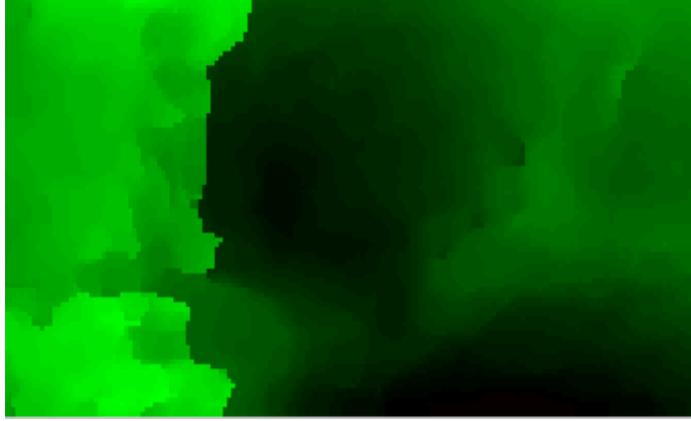
Figure 6.11: The MAP disparity map corresponding to the same disparity probability field as represented in Fig. 6.8 is smoothed using Loopy Belief propagation, (max-product updates, accelerated message passing schedule).

In Eqns. 6.45 and 6.46 it is assumed that $\mathbf{y}$ and $\mathbf{Y}^{R1}_{r(i)}$ have been homogeneously scaled so that the last element is unity.

This is illustrated in Fig. 6.12, where a homographically warped image is drawn, in which parts of the warped image are at negative coordinates on the image plane. The use of offsets is also appropriate if the image is warped (rectified) onto a region in the image plane which is positive and displaced from the origin. Then the image is translated so that as little image space as possible is wasted: the image is translated so that the topmost point of the warped image is on the horizontal axis (the zero vertical image coordinate), and the leftmost point of the warped image is at the vertical axis (zero horizontal image coordinate). Therefore the offsets are nearly always stored and used as per Eqns. 6.45 and 6.46, in our implementation.

Thus if $\mathbf{I}^1$ and $\mathbf{I}^2$ are the first and second images in the stereo pair, and if $\mathbf{I}^{R1}$ is the rectified first image and $\mathbf{I}^{R2}$ is the rectified second image, then $r(i)$ in image $\mathbf{I}^{R1}$ is at homogeneous image coordinates $\mathbf{Y}^{R1}_{r(i)} = Q(\mathbf{X}^{I1}_i, o^1_x, o^1_y, \mathbf{H}_1)$ in the first rectified image and this corresponds to coordinates $\mathbf{Y}^{R2} = \mathbf{Y}^{R2}_b + [d^{R1}_{r(i)} \ 0 \ 0]^T$ where $\mathbf{Y}^{R2}_b = \mathbf{Y}^{R1}_{r(i)} + [o^1_x - o^2_x \ o^1_y - o^2_y \ 0]^T$ in the second rectified image, given that $d^{R1}_{r(i)}$ is the disparity label for corner vertex node $x^{R1}_{r(i)}$ in $\mathbf{I}^{R1}$.

The function $\mathbf{Y}^{R1}_{r(i)} = Q(\mathbf{X}^{I1}_i, o^1_x, o^1_y, \mathbf{H}_1)$ is described algorithmically rather than mathematically, as a homogeneous scaling operation is involved, which would make a mathematical description difficult:

1. $\mathbf{y}_i \leftarrow \mathbf{H}_1 \mathbf{X}^{I1}_i$
2. $\mathbf{y}_i \leftarrow \text{homo}(\mathbf{y}_i)$
3. $\mathbf{Y}^{R1}_{r(i)} \leftarrow \mathbf{y}_i - [o^1_x \ o^1_y \ 0]^T$

where "homo$(\mathbf{x})$" is a function that takes as an argument the homogeneous vector $\mathbf{x} = [x_1 \ x_2 \ x_3]$ and scales it so that the last element is unity, i.e. $\mathbf{x}/x_3 = \text{homo}(\mathbf{x})$.
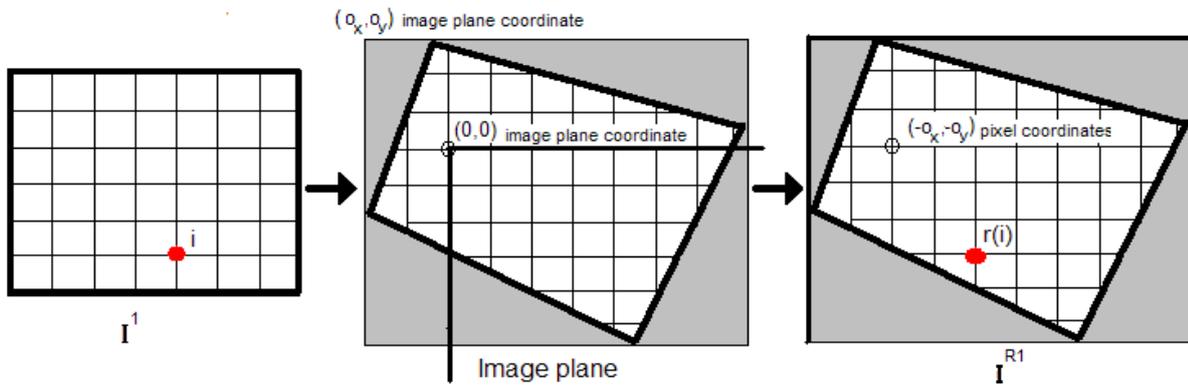
Figure 6.12: An image ($\mathbf{I}^1$) is depicted on the left and the same image homographically warped (rectified) by matrix $\mathbf{H}_1$ on the right ($\mathbf{I}^{R1}$). The middle figure is the image plane, with the origin indicated. The image on the right is obtained by viewing the image plane in image coordinates. The origin in the image plane has pixel coordinates $(-o_x, -o_y)$, if the image is translated so that all points have positive pixel coordinates. Each square indicates a pixel. The corner vertex node i (circle) on the left corresponds to the corner vertex node $r(i)$ on the right. The corner vertex nodes may be numbered row by row, starting at the top left of the image and rectified image. This is the same homographical warping done to obtain rectified image pairs, such as those shown in Fig. 6.7. The random variables on the lattice for the corner vertex nodes for the image on the left are labelled x_i, the random variables for the corresponding corner vertex nodes for the homographically transformed image on the right are labelled x_i^{R1}. The point coordinates (0,0) are indicated in the second image. The offsets (o_x, o_y) indicate the displacements between the pixel locations and the corner vertex node locations on the image plane. If there is an offset of (o_x, o_y), it means that the pixel at (x, y) corresponds to the coordinates $(x + o_x, y + o_y)$ in the image plane.
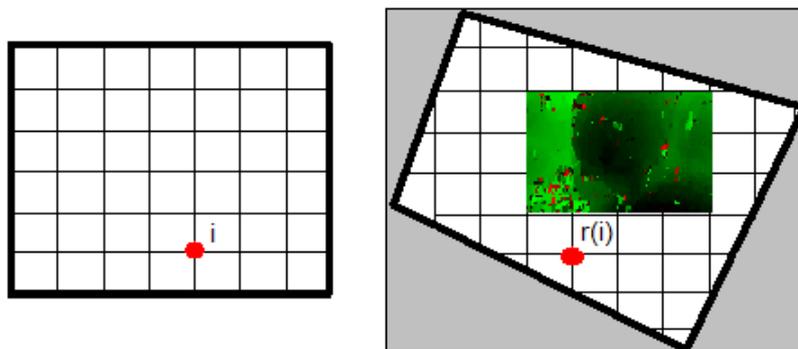
Figure 6.13: The disparity map from Fig. 6.8 is superimposed on the rectified image grid to show that the disparity maps come from cropped subwindows within the rectified images.

The 3-D coordinates $\mathbf{Z}_i$ of the point corresponding to corner vertex node $r(i)$ given a disparity $d_{r(i)}^{R1}$ may be found using a linear 3-D triangulation algorithm, such as the Singular Value Decomposition (SVD) method described in [49]:

$$\mathbf{Z}_i = \text{triangulate}(\mathbf{H}_1, \mathbf{H}_2, o_x^1, o_y^1, o_x^2, o_y^2, \mathbf{Y}_{r(i)}^{R1}, \mathbf{Y}_b^{R2}, d_{r(i)}^{R1}, \mathbf{P}_1, \mathbf{P}_2), \tag{6.47}$$

where o_x^1 and o_y^1 are the offsets for $\mathbf{I}^{R1}$, and o_x^2 and o_y^2 are the offsets for the $\mathbf{I}^{R2}$. The triangulation algorithm takes a point correspondence between points in the rectified images $\mathbf{I}^{R1}$ and $\mathbf{I}^{R2}$, and finds the corresponding original positions of the points in the unrectified images $\mathbf{I}^1$ and $\mathbf{I}^2$ via Eqn. 6.43. In fact, the triangulation function can take the following parameters:

$$\mathbf{Z}_i = \text{triangulate}(\mathbf{X}_i^{I1}, \mathbf{H}_2, o_x^2, o_y^2, \mathbf{Y}_b^{R2}, d_{r(i)}^{R1}, \mathbf{P}_1, \mathbf{P}_2), \tag{6.48}$$

since it is already known that the point $\mathbf{Y}_{r(i)}^{R1}$ in $\mathbf{I}^{R1}$ corresponds to $\mathbf{X}_i^{I1}$ in $\mathbf{I}^1$. Since there is a probability $p(x_{r(i)}^{R1} = s)$ on the corner vertex node $x_{r(i)}^{R1}$ taking on a value s, this translates into a probability on the location of the 3-D point $\mathbf{Z}_i$. Since point $r(i)$ with disparity s in $\mathbf{I}^{R1}$ triangulates into

$$\mathbf{Z}_i^s = \text{triangulate}(\mathbf{X}_i^{I1}, \mathbf{H}_2, o_x^2, o_y^2, \mathbf{Y}_b^{R2}, s, \mathbf{P}_1, \mathbf{P}_2), \tag{6.49}$$

we can say

$$p(\mathbf{Z}_i^s) = p(x_{r(i)}^{R1} = s). \tag{6.50}$$

These points $\mathbf{Z}_i^s$ occur along the same ray. A disparity s at node $r(i)$ in the first rectified image $\mathbf{I}^{R1}$ corresponds to corner vertex node i in the first unrectified image $\mathbf{I}^1$ having a label s. The probabilities are equivalent (excluding information derived from the BRDF estimates):

$$p(x_i = s) \equiv p(x_{r(i)}^{R1} = s). \tag{6.51}$$

What we have done is simply equated the probability of a corner vertex node $x_{r(i)}^{R1}$ having a particular disparity in the rectified image pair, to the probability of the same corner vertex node being triangulated to the corresponding point into the scene. The number of points in the scene which a corner vertex node in $\mathbf{I}^1$ can be triangulated into is the same as the number of possible disparity values $d_{r(i)}^{R1} \in \mathcal{L}$.

Later in this chapter a methodology for incorporating this probabilistic information on the probability of a corner vertex node label in $\mathbf{I}^{R1}$ into the estimation of the probability of the state of a corner vertex node label in $\mathbf{I}^1$ is developed, which includes energy terms associated with BRDF information (Eqn. 4.12), using Gibbs style potential interactions within a pseudolikelihood framework.

6.4.1   Triangulating points in 3-D given image points and calibrated projection matrices.

Assume that the location of a point in each of the original (unrectified) images has been calculated: $\mathbf{X}_i^{I1}$ in $\mathbf{I}^1$ and $\mathbf{X}_i^{I2}$ in $\mathbf{I}^2$, using Eqn. 6.43 or 6.46 on some pair of corresponding points $\mathbf{Y}_{r(i)}^{R1}$ and $\mathbf{Y}^{R2} = \mathbf{Y}_b^{R2} + [d_{r(i)}^{R1} \ \ 0 \ \ 0]^T$ in the rectified images $\mathbf{I}^{R1}$ and $\mathbf{I}^{R2}$.

Included here is a description of a linear algorithm for triangulating image points into world coordinates (in $\mathbb{R}^3$). The method assumes that the projection matrices contained in $\mathbf{P}$ have all been found through calibration. If in image n there is a point $\mathbf{X}^n = \mathbf{P}_n \mathbf{Z}$, and $\mathbf{Z}$ is the 3-D world coordinate of the point which is projected as $\mathbf{X}^n$ in image n, then it is possible to rewrite the coefficients to form a linear system of the form $A\mathbf{Z} = 0$. For convenience, we write $\mathbf{X}^n = (x, y, 1)$, the location of the projected point in the first image. The homogeneous scale factor is eliminated by taking the cross-product $\mathbf{X}^n \times (\mathbf{P}_n \mathbf{Z})$ to give three linearly dependent equations:

$$x(\mathbf{p}^{3T}\mathbf{Z}) - (\mathbf{p}^{1T}\mathbf{Z}) = 0$$
$$y(\mathbf{p}^{3T}\mathbf{Z}) - (\mathbf{p}^{2T}\mathbf{Z}) = 0$$
$$x(\mathbf{p}^{2T}\mathbf{Z}) - y(\mathbf{p}^{1T}\mathbf{Z}) = 0. \tag{6.52}$$

One can now assemble a matrix of the form

$$A = \begin{bmatrix} x\mathbf{p}^{3T} - \mathbf{p}^{1T} \\ y\mathbf{p}^{3T} - \mathbf{p}^{2T} \\ x'\mathbf{p}'^{3T} - \mathbf{p}'^{1T} \\ y'\mathbf{p}'^{3T} - \mathbf{p}'^{2T} \end{bmatrix} \tag{6.53}$$

where $\mathbf{p}^{iT}$ is the ith row of $\mathbf{P}_1$ and $\mathbf{p}'^{iT}$ is the ith row of $\mathbf{P}_2$. This linear system can be solved in many ways, for example by using the SVD.

The function of Eqns. 6.48 and 6.49 ($\mathbf{Z}_i = \text{triangulate}(\mathbf{X}_i^{I1}, \mathbf{H}_2, o_x^2, o_y^2, \mathbf{Y}_b^{R2}, d_{r(i)}^{R1}, \mathbf{P}_1, \mathbf{P}_2)$) can therefore be described in the following algorithm:

1. $\mathbf{Y}^{R2} \leftarrow \mathbf{Y}_b^{R2} + [d_{r(i)}^{R1} \ \ 0 \ \ 0]^T$
2. $\mathbf{y}_2 \leftarrow \mathbf{Y}^{R2} + [o_x^2 \ \ o_y^2 \ \ 0]^T$
3. $\mathbf{x}_2 \leftarrow \mathbf{H}_2^{-1}\mathbf{y}_2$
4. $\mathbf{X}_i^{I2} \leftarrow \text{homo}(\mathbf{x}_2)$
5. Use linear algorithm described above to find 3-D point $\mathbf{Z}_i$ given $\mathbf{X}_i^{I1}, \mathbf{X}_i^{I2}, \mathbf{P}_1, \mathbf{P}_2$,

where as before $\text{homo}(\mathbf{x})$ is a function that takes as an argument the homogeneous vector $\mathbf{x} = [x_1 \ \ x_2 \ \ x_3]^T$ and scales it so that the last element is unity, i.e. $\mathbf{x}/x_3 = \text{homo}(\mathbf{x})$.

Note that if the projection matrices are uncalibrated, the reconstructed points are reconstructed only up to a projective ambiguity (see [49] for details). Therefore the cameras must be accurately calibrated. An example of part of the maximum a posteriori surface generated by the disparity map of Fig. 6.11 is shown in Fig. 6.14.

6.4.2 Calculation of BRDF potential terms using triangulation

It was stated earlier that the MRF clique potential terms of Eqns. 4.20 and 4.21 are calculated using a method which involves projection of the corner vertex nodes into the scene. The algorithm for calcu-
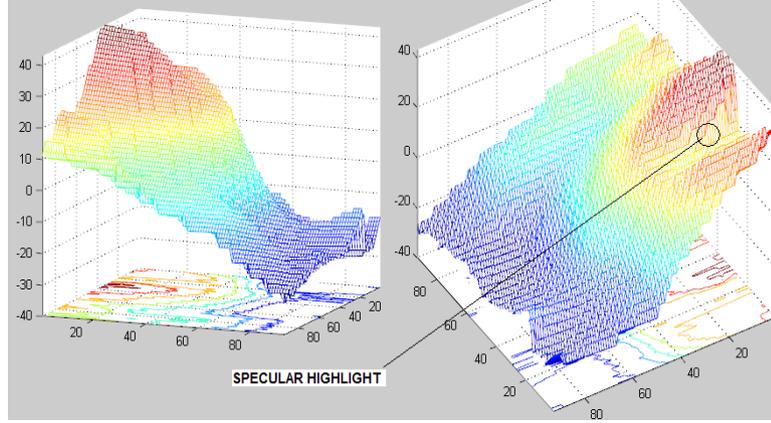
Figure 6.14: The MAP surface reconstruction of a surface over a small image region. In this case, a specular highlight on part of the surface has led to bad matching for pixels corresponding to the highlighted area, and a corresponding bad MAP estimate for this region of the surface reconstruction.

lating $\text{BRDF}(i, j, k, x_i, x_j, x_k, \overrightarrow{L}, \mathbf{P}, \theta)$ follows:

1. $\mathbf{Z}_i = \text{triangulate}(\mathbf{X}_i^{I1}, \mathbf{H}_2, o_x^2, o_y^2, \mathbf{Y}_b^{R2}, x_i, \mathbf{P}_1, \mathbf{P}_2)$
2. $\mathbf{Z}_j = \text{triangulate}(\mathbf{X}_j^{I1}, \mathbf{H}_2, o_x^2, o_y^2, \mathbf{Y}_b^{R2}, x_j, \mathbf{P}_1, \mathbf{P}_2)$
3. $\mathbf{Z}_k = \text{triangulate}(\mathbf{X}_k^{I1}, \mathbf{H}_2, o_x^2, o_y^2, \mathbf{Y}_b^{R2}, x_k, \mathbf{P}_1, \mathbf{P}_2)$
4. Calculate normal $\hat{\mathbf{n}}$ given the 3-D surface triangle formed by $\mathbf{Z}_i, \mathbf{Z}_j, \mathbf{Z}_k$
5. Calculate all other intermediate variables such as $\mathbf{h}$, $\mathbf{k}_1$ and $\mathbf{k}_2$
6. Calculate expected intensity as viewed by $\mathbf{P}_1$ according to Eqn. 3.1 or 3.12.

Note that the sixth parameter in the triangulate$(\cdot)$ functions above are states on the corner vertex nodes of the first unrectified image $\mathbf{I}^1$. In this way the labels for a corner vertex node i in $\mathbf{I}^1$ are used directly in the triangulation of a point into the scene, via the disparity of the corresponding point $r(i)$ in the rectified images $\mathbf{I}^R$. The label on the corner vertex node i in $\mathbf{I}^1$ is used as the disparity value of corner vertex node $r(i)$ in $\mathbf{I}^{R1}$.

6.5 Incorporating dense stereo uncertainty into Markov random field pseudolikelihood model for estimating BRDF parameters

In a previous section, it was shown how a probabilistic dense stereo map can be estimated for a given surface. The true value for any corner vertex node label is taken to be the MAP value for the corner vertex node label (i.e. the label with the highest belief associated with it according to Eqn. 6.34). However, in the calculation of the pseudolikelihood, we take into account not only the other possible labels (and the corresponding depth or range) of the surface at each point, but also the probabilities on

these labels as calculated by a loopy belief propagation based dense stereo correspondence algorithm.

Recalling Eqns. 6.28, 6.29 and 6.30:

$$p(\mathbf{X}|\mathbf{Y}) \propto \left(\prod_s \exp(-U_s^B)\right)\left(\prod_s \exp(-U_s^C)\right) \tag{6.54}$$

where

$$U_s^B = \sum_{t:s>t,t\in\mathcal{N}s} \psi_p(x_s, x_t) \tag{6.55}$$

and

$$U_s^C = \psi_d(s, x_s), \tag{6.56}$$

one can form potential energy terms appropriate for calculating the pseudolikelihood (the potential energy terms for Markov neighbours of a site are used to calculate the site's probability of taking on a specific label). These energy terms are for nodes in the rectified first image $\mathbf{I}^{R1}$, where corner vertex node i in the first image corresponds to corner vertex node or point $r(i)$ in the rectified first image:

$$V_i^B = \sum_{t\in\mathcal{N}_{r(i)}} \psi_p(x_{r(i)}^{R1}, x_t^{R1}) \tag{6.57}$$

and

$$U_i^C = \psi_d(r(i), x_{r(i)}^{R1}). \tag{6.58}$$

These are energy terms relating to the labels $x_{r(i)}^{R1}$ on corner vertex nodes in the first rectified image $\mathbf{I}^{R1}$. They use the local clique structure of the transformed corner vertex locations in the rectified image. Note that the potential term of Eqn. 6.57 differs from the potential term of Eqn. 6.29, because the calculation of the pseudolikelihood (Eqn. 4.24) requires the summation of all neighbouring pairwise potential terms, even though this means using pairwise potential terms twice.

This pseudolikelihood can be reformulated to include the label probabilities for each site, given the compatibility cliques on the neighbouring pairwise nodes $r(i)$ in the rectified first image, and the energy associated with the local evidence (robust function with pixel dissimilarity measure). This is written as

$$PL_2(\mathbf{X}|\theta) = \log \prod_{i\in S} p(x_i|x_{\mathcal{N}_i}, \theta)$$
$$= \log \prod_{i\in S} \frac{\exp(-U_i^A(x_i, x_{\mathcal{N}_i}, \theta) - V_i^B(x_i, x_{\mathcal{N}_i}) - U_i^C(x_i, \mathbf{I}^R))}{\sum_{s\in\mathcal{L}} \exp(-U_i^A(s, x_{\mathcal{N}_i}, \theta) - V_i^B(s, x_{\mathcal{N}_i}) - U_i^C(s, \mathbf{I}^R))}, \tag{6.59}$$

where the potential term U^A is the BRDF-based potential energy term called U in Eqn. 5.65, using the methodology of section 6.4.2, specifically encoding the energy terms associated with the corner vertex nodes' agreement with the image data.

It is sensible to omit the V^B potential term if there is no reason to assume any spatial continuity charac-
teristics (as are usually encoded in this term), e.g. if the surface of the object is not predictably smooth
or homogenous. One problem with this formulation is that, by itself, it does not allow the rectified
probabilistic disparity map to be smoothed: in typical LBP or general MRF methods to denoise or
smooth a probabilistic disparity map, the algorithm is allowed to run for many iterations, and informa-
tion flows from one side of the disparity map to the other. With the pseudolikelihood in the form of Eqn.
6.59, only local interactions are considered, with no information flow across non-adjacent (non-Markov
neighbouring) nodes or pixels; this is undesirable since information from non-adjacent nodes should be
incorporated into the probabilistic reconstruction of any point on the surface.

In our implementation, the probability on the range label x_i of corner vertex node i in the unrectified
first image depends on the belief on the disparity $b(x_{r(i)}^{R1})$ of corner vertex node $r(i)$ in the rectified
first image $\mathbf{I}^{R1}$, calculated using a dense stereo LBP algorithm. This is because there is a one-to-one
correspondence between the disparity x_i for corner vertex node $r(i)$, and a range label for corner vertex
node i in the unrectified first image $\mathbf{I}^1$. We seek to include both information from the reflectance and
from the dense stereo correspondence in the calculation of the probability of a corner vertex node label.

It may seem artificial to include a posterior distribution derived from an LBP calculated approximation
for the beliefs on the disparities of the corner vertex nodes in the rectified first image $\mathbf{I}^{R1}$, without
also using a LBP type algorithm to calculate the beliefs on the corner vertex node states using BRDF
information. However, this is easily justified by noting that the BRDF data is not yet available at this
stage. In fact, it is the BRDF parameters that we seek to estimate. The beliefs on the disparity field
as calculated by the LBP algorithm can best be regarded as a range sensor reading for each site on the
lattice, even though it is the result of processing such raw probabilistic data on the corner vertex node
states.

This situation may also arise if for example an LBP algorithm for dense stereo was implemented locally
on the hardware of a stereo camera pair, and transmitted to a computer for BRDF estimation. In general,
the probabilities/beliefs for the disparity field may be calculated by any non-Bayesian method, and it is
this general case that our algorithm caters for.

It is therefore preferable to use the final posterior belief on the 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 this can be converted into a potential term:

$$U_i^D(x_i, \mathbf{I}^R) = \psi_i^D(x_i) = -T \log(b(x_{r(i)}^{R1})/k_1), \tag{6.60}$$

where T is the temperature and k_1 is a normalization constant. This formulation permits the incor-
poration 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. 6.60 is

$$
\begin{aligned}
PL_3(\mathbf{X}|\theta) &= \log \prod_{i \in S} p(x_i|x_{\mathcal{N}_i}, \theta) \\
&= \log \prod_{i \in S} \frac{\exp(-U_i^A(x_i, x_{\mathcal{N}_i}, \theta) - U_i^D(x_i, \mathbf{I}^R))}{\sum_{s \in \mathcal{L}} \exp(-U_i^A(s, x_{\mathcal{N}_i}, \theta) - U_i^D(s, \mathbf{I}^R))}.
\end{aligned}
\tag{6.61}
$$

For notational convenience we also write the function

$$
\begin{aligned}
ePL_3(\mathbf{X}|\theta) &= \exp(PL_3(\mathbf{X}|\theta)) \\
&= \prod_{i \in S} \frac{\exp(-U_i^A(x_i, x_{\mathcal{N}_i}, \theta) - U_i^D(x_i, \mathbf{I}^R))}{\sum_{s \in \mathcal{L}} \exp(-U_i^A(s, x_{\mathcal{N}_i}, \theta) - U_i^D(s, \mathbf{I}^R))}.
\end{aligned}
\tag{6.62}
$$

Now that the third pseudolikelihood function PL_3 has been defined, the importance function of Eqn. 5.72 can be reformulated:

$$
G(\theta|\mathbf{X}) = s(ePL_3(\mathbf{X}|\theta)),
\tag{6.63}
$$

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. 5.70). This is the importance function used for the PMC based MRF parameter estimation. As with the previous pseudolikelihood functions, one can construct importance functions G^n based on iteration over a certain part the lattice, using

$$
ePL_3^n(\mathbf{X}|\theta) = \prod_{i \in S^n} p(x_i|x_{\mathcal{N}_i}, \theta),
\tag{6.64}
$$

where S^n is the nth 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

$$
G^n(\theta|\mathbf{X}) = s(ePL_3^n(\mathbf{X}|\theta)).
\tag{6.65}
$$

6.6 Population Monte Carlo for BRDF parameter estimation

A valid implementation of the generic PMC algorithm to approximate a posterior distribution on a BRDF parameterisation can now be given as:

6.6.1 Population Monte Carlo algorithm (I)

- n=0: Initialize sample locations and probability weights:

    1. Generate samples $(\theta_i^0)_{1 \le i \le N} \sim G^0$

    2. Compute weights $(\pi_i^0)_{1 \le i \le N} = (ePL_3(\mathbf{X}|\theta_i^0)/G^0(\theta_i^0|\mathbf{X}))_{1 \le i \le N}$

    3. $\{\overrightarrow{\theta}^0, \overrightarrow{\pi}^0\} = \text{resampleParticles}(\{\overrightarrow{\theta}^0, \overrightarrow{\pi}^0\})$

- For $n > 0$

    1. Generate samples $(\theta_i^n)_{1 \le i \le N} \sim G^n(\{\overrightarrow{\theta}^{n-1}, \overrightarrow{\pi}^{n-1}\}|\mathbf{X})$

    2. Compute weights $(\pi_i^n)_{1 \le i \le N} = (ePL_3(\mathbf{X}|\theta_i^n)/G^n(\theta_i^n|\mathbf{X}))_{1 \le i \le N}$

    3. $\{\overrightarrow{\theta}^n, \overrightarrow{\pi}^n\} = \text{resampleParticles}(\{\overrightarrow{\theta}^n, \overrightarrow{\pi}^n\})$.

Step 1 in both cases needs to be clarified. Given an analytical or functional expression for a pdf, it is sometimes easy to sample from it (for example from the normal distribution), but often it is not. In our case, where the importance functions G are not known analytically (they are evaluated per sample given the data), the distributions must be generated or sampled from in a different way. When $n = 0$, step 1, ("Generate samples $(\theta_i^0)_{1 \le i \le N} \sim G^0$"), is performed as follows:

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

2. $\{\overrightarrow{\theta}^0, \overrightarrow{\pi}^0\} = \text{resampleParticles}(\{\overrightarrow{u}^0, \overrightarrow{v}^0\}, \sigma_0)$.

When ($n > 0$), step 1 ("Generate samples $(\theta_i^n)_{1 \le i \le N} \sim G^n(\{\overrightarrow{\theta}^{n-1}, \overrightarrow{\pi}^{n-1}\}|\mathbf{X})$"), can be performed in the following way:

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

$$v_i = \pi_i^{n-1} G^n(\theta_i^{n-1}|\mathbf{X})/ePL_3(\mathbf{X}|\theta_i^{n-1}) \tag{6.66}$$

2. $\{\overrightarrow{\theta}^n, \overrightarrow{\pi}^n\} = \text{resampleParticles}(\{\overrightarrow{\theta}^{n-1}, \overrightarrow{v}\}, \sigma_n)$.

The additive Gaussian noise at the resampling stage is parameterized by σ_n, which is the standard deviation of the noise, and which decreases at each iteration of the PMC algorithm. It is likely that the function $ePL_3(\cdot)$ is far more expensive to evaluate than any $G^n(\cdot)$, (this motivates us to create importance functions $G^n(\cdot)$ which use only part of the data and are therefore computed faster). Furthermore,

the function $ePL_3(\cdot)$ is too sharp, and sampling from it would cause too many particles to be resampled from the MAP sample, resulting in degeneracy. For this reason propagation of samples which are weighted directly with the $ePL_3(\cdot)$ function as a factor is avoided (although to recover the correct posterior distribution at any time it is necessary to correctly weight the samples with $ePL_3(\cdot)$).

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:

6.6.2 Population Monte Carlo algorithm (II)

- 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^0

    4. If the correct posterior distribution $ePL_3(\mathbf{X}|\theta)$ is needed:

    5.   Compute $(\pi_i^0)_{1\leq i\leq N} = (ePL_3(\mathbf{X}|a_i^0)/G^0(a_i^0|\mathbf{X}))_{1\leq i\leq N}$

    6. $\{\overrightarrow{\theta}^0, \overrightarrow{\pi}^0\} = \text{resampleParticles}(\{\overrightarrow{a}^0, \overrightarrow{\pi}^0\})$

    7. end.

- For $n > 0$

    1. Generate samples $(a_i^n)_{1\leq i\leq N} \sim G^n(\{\overrightarrow{a}^{n-1}, \overrightarrow{c}^{n-1}\}|\mathbf{X})$

    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^n

    4. If the correct posterior distribution $ePL_3(\mathbf{X}|\theta)$ is needed:

    5.   Compute $(\pi_i^n)_{1\leq i\leq N} = (ePL_3(\mathbf{X}|a_i^n)/G^n(a_i^n|\mathbf{X}))_{1\leq i\leq N}$

    6. $\{\overrightarrow{\theta}^n, \overrightarrow{\pi}^n\} = \text{resampleParticles}(\{\overrightarrow{a}^n, \overrightarrow{\pi}^n\})$

    7. end.

Once again 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 $c_i^0 = G^0(u_i^0|\mathbf{X}), \quad (1 \leq i \leq N)$

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

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

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

$$w_i = c_i^{n-1} G^n(a_i^{n-1}|\mathbf{X}) \tag{6.67}$$

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

As before, the additive Gaussian noise at the resampling stage is parameterized by σ_n, which is the standard deviation of the noise, and which decreases 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 (resampleParticles$(\cdot, \sigma_n)$) is initialized with $\sigma_0 = 0.07$, and decreases linearly over the PMC iterations until finally $\sigma_T = 0$.

The MRF framework for estimating BRDFs through MRF parameters has been established, and methods for sampling its parameters have been described, along with a Pseudolikelihood based cost function to take into accoutn uncertainties in the reconstructed surface. The next two chapters deal with convergence results for synthetic data and classification results on real surface data.

Chapter 7

BRDF parameter estimation on synthetic data

Our algorithm was tested using both synthetically generated surface and illumination data, and on real data comprising of reconstructed surfaces of materials and of froth surfaces. The testing on synthetic data was done to demonstrate the performance of the algorithm, and its convergence to the correct posterior distribution and to the true reflectance parameters which were used to generate each synthetic surface reflectance image. Three algorithms are compared for convergence to the true reflectance parameters, namely a Levenberg-Marquardt nonlinear optimizer, a dynamically weighted MCMC sampler, and a PMC sampler. Some statistics are reported, concerning the convergence results for different algorithm running parameters. The parameters which are varied, in order to explore the performance of the parametric BRDF estimation algorithms, are

1. The number of possible labels for each synthetic corner vertex node

2. The number of samples to propagate in the PMC method

3. The number of seeds in the Multiple-Seed LM algorithm

4. The BRDF model used to render the synthetic surfaces (Ward and anisotropic Phong)

Samples sets which have been passed through one or more PMC iterations are referred to as "evolved", or "developed" samples.

7.1   Reparameterisation of Ward and Phong models

In the course of our preliminary experiments on synthetic data, it was found that the best range of parameters for the Ward model, for accurate recovery, are as follows:

$$\rho_d \in [0, 1], \qquad \rho_s \in [0, 1], \qquad \alpha \in (0, 0.5]. \tag{7.1}$$

In [122] it is noted that for a physically valid reflectance function, $\rho_s + \rho_d < 1$, and that α should not be "too large" (in their experiments α is always less than 0.5).

We set the parameter space for the samples or particles in our MCMC and PMC samplers and seeds in the LM minimizer to be in the range of values in the unit cube. Given samples in this range and assuming the Ward BRDF parameterisation, the calculation of α is done by taking the last component of any particle (or seed) and multiplying it by 0.5 to recover the BRDF parameter vector. Similarly, the optimal anisotropic Phong parameter values were found to be in the ranges:

$$R_d \in [0, 1], \qquad R_s \in [0, 1], \qquad n_u \in (0, 25], \qquad n_v \in (0, 25]. \tag{7.2}$$

We tested only the isotropic version of the anisotropic Phong model [2], setting $n_u = n_v$, which means that the samples and seeds exist in $\mathbb{R}^3$. Since the optimal range for these parameters is (0,25] (values greater than 25 are difficult to estimate accurately and do not generate realistic looking surfaces), we multiply the last component of any sample vector by 25 to get the anisotropic Phong parameter set. This enables us to initialize all samples at evenly spaced points on the unit cube, for both the Ward and anisotropic Phong parameter models (which is not necessary mathematically, but it allows reuse of software modules, including those for visualizing the samples and cost functions).

7.2   Synthetic Data

The convergence results of the MCMC, PMC and LM algorithms were tested by varying a number of the internal and external algorithm parameters. The MRF based BRDF probability model as we have described it allows the following internal parameters to be varied:

- The number of samples in the PMC sampler

- The amount of additive Gaussian noise used when sampling from the importance function

- The number of corner vertex nodes to iterate through before doing a resample operation

- The degree to which the sample set should be sharpened or softened (annealed), if at all

- The number of possible labels per corner vertex node (in the synthetic data)

- The prior distribution on the labeling of each corner vertex node.

The external factors which affect the algorithm running on real or synthetic data are

- The reflectance model used

- The number of simultaneous light sources.

The last two items on internal parameter variability refer to the simulated disparity noise. In stereo images, disparity noise can lead to depth ambiguity of points on the reconstructed surface. To test each algorithm's ability to recover the correct BRDF parameters, using synthetic data, a synthetic probabilistic range map was generated for each synthetic surface. Normalized probabilities were associated to each of the range values corresponding to the labels of each corner vertex node, according to the following distribution, which has a maximum probability on the true depth or range value of the point:

$$p_i(x_i, z_i) \propto \exp(-\psi_i(x_i, z_i)), \tag{7.3}$$

where

$$\psi_i(x_i, z_i) = |z_i - r_i(x_i)|/\sigma. \tag{7.4}$$

Here z_i is the true range of the synthetic surface at point i, $r_i(x_i)$ is the corresponding range of corner vertex node i given a labelling of x_i on that node and the projective camera data, and σ is a definable parameter for generating priors on the height profiles of these synthetic surfaces. We set each incremental label on the corner vertex nodes to correspond to an increment of 0.3 units (pixel widths) of range from the camera center, with $\sigma = B/5$, where B is the number of possible disparity labels which a corner vertex node can take on. Note that in our synthetic data experiments, the surfaces are generated without doing dense stereo correspondence estimation on a pair of images. Therefore the pseudolikelihood function used in the synthetic data experiments is

$$
\begin{aligned}
PL_s(\mathbf{X}|\theta) &= \log \prod_{i \in S} p(x_i | x_{\mathcal{N}_i}, \theta) \\
&= \log \prod_{i \in S} \frac{\exp(-U_i^A(x_i, x_{\mathcal{N}_i}, \theta) - U_i^R(x_i, z_i))}{\sum_{s \in \mathcal{L}} \exp(-U_i^A(s, x_{\mathcal{N}_i}, \theta) - U_i^R(s, z_i))},
\end{aligned} \tag{7.5}
$$

where

$$U_i^R = \psi_i(x_i, z_i). \tag{7.6}$$

The target distribution is then

$$
\begin{aligned}
ePL_s(\mathbf{X}|\theta) &= \exp(PL_s(\mathbf{X}|\theta)) \\
&= \prod_{i \in S} \frac{\exp(-U_i^A(x_i, x_{\mathcal{N}_i}, \theta) - U_i^R(x_i, z_i))}{\sum_{s \in \mathcal{L}} \exp(-U_i^A(s, x_{\mathcal{N}_i}, \theta) - U_i^R(s, z_i))}.
\end{aligned} \tag{7.7}
$$

In the above, the energy term U_i^R referring to the range of each surface point is not dependent on the posterior distribution calculated using a Loopy Belief Propagation algorithm, as in Eqn. 6.61. Only a single image $\mathbf{I}^1$ is used, since there is no dense stereo correspondence done.

To calculate the softening functions, the set of corner vertex nodes on the lattice is divided into three subsets: S^0, S^1, S^2, such that $S^n = \{n, n+k, n+2k, \dots\}$ with $k = 3$. The synthetic pseudolikelihood function is then

$$
\begin{aligned}
ePL_s^n(\mathbf{X}|\theta) &= \exp(PL_s^n(\mathbf{X}|\theta)) \\
&= \prod_{i \in S^n} \frac{\exp(-U_i^A(x_i, x_{\mathcal{N}_i}, \theta) - U_i^R(x_i, z_i))}{\sum_{s \in \mathcal{L}} \exp(-U_i^A(s, x_{\mathcal{N}_i}, \theta) - U_i^R(s, z_i))},
\end{aligned}
\tag{7.8}
$$

as in Eqn. 6.65. Since 20 PMC iterations are done, 20 importance functions G^n are required. The three functions are reused, so

$$
G_s^n(\theta|\mathbf{X}) = s(ePL_s^j(\mathbf{X}|\theta)),
\tag{7.9}
$$

where $j = \mathrm{mod}(n, 3)$.

For the synthetic data it is assumed that the only light source is a point light source at infinity. The algorithm may be extended to other types of light sources. The convergence of the PMC algorithm to the known correct parameters is compared to the convergence results of a dynamically weighted MCMC based algorithm, which also returns a sample set representing a pdf over all the BRDF parameters, and to the convergence results for a multiple-seed Levenberg-Marquardt minimization algorithm.

To generate synthetic surfaces for testing the algorithms using the Ward parameterisation, twenty points in a cube of space are randomly chosen. Interpolation is done between these points using thin plate spline interpolation (due to [28] and discussed in [30]). This generates a smooth and natural-looking surface, which is good for recovery of Ward parameters. The interested reader is referred to Appendix C, where the algorithm for thin plate spline interpolation is described. Since the surface is known precisely, it can be rendered from any viewpoint, with any reflectance model. Examples of such surfaces are shown in Fig. 7.1.

Experimentation showed that such smooth flat synthetic surfaces are not optimal for recovery of the anisotropic Phong parameterizations. This is because it is difficult to separate the effects of the surface roughness parameters (n_u, n_v) and the diffuse/specurity parameters (R_d, R_s) when using surfaces which do not have many parts which are at relatively oblique angles to the camera. The Ward model behaves differently: to separate the specular reflectance coefficient ρ_s from the surface roughness α, a surface which has less curvature away from the image plane is required (the size of the specular lobe is an indicator of the amount of data available to separate the effects of the ρ_s and α terms).

Each model is therefore tested synthetically on its own optimal surface type for BRDF parameter estimation. Therefore, to recover the anisotropic Phong model parameters in our synthetic experiments, we
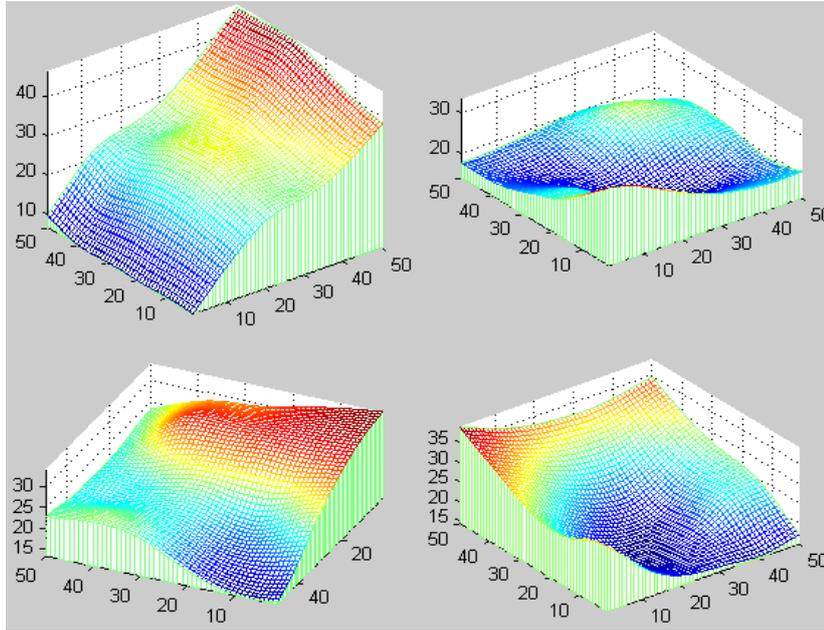
Figure 7.1: Synthetically generated surfaces (generated using thin plate spline interpolation) for testing of Ward BRDF parameter recovery algorithms.

used jagged surfaces, with more extreme angles between the surface normals and the incident lighting and camera direction angles. These surfaces were generated by making the depth or range at each point on the surface a sample from a uniform distribution over the range [0,6], and adding a constant depth value. Such a surface is shown in Fig. 7.2.

When testing the BRDF extraction algorithms for both the Ward and anisotropic Phong reflectance models, it is assumed that the light source is positioned at infinity. Multiple Levenberg-Marquardt nonlinear minimizations on a probability error function derived from the pseudolikelihood function are also done. The parameter set for each minimization is initialized on a parameter vector taken from a set of regularly-spaced points within the BRDF parameter space. The experimentation methodology for testing the algorithms on the synthetic data is as follows:

Experimental methodology using synthetic data

1. For $i=1, \dots$,number of trials
2.   Generate random synthetic surface (smooth or jagged),
3.   Render surface with random light source (at infinity) and camera at the origin, with Ward
        or anisotropic Phong reflectance model, giving reflectance image $\mathbf{I}^1$,
4.   Create range disparity field $\mathbf{X} = \{x_i\}$, where x_i can take on labels which correspond to
        discrete ranges about the true range of the surface at corner vertex node i. Then $\mathbf{S}$ contains the
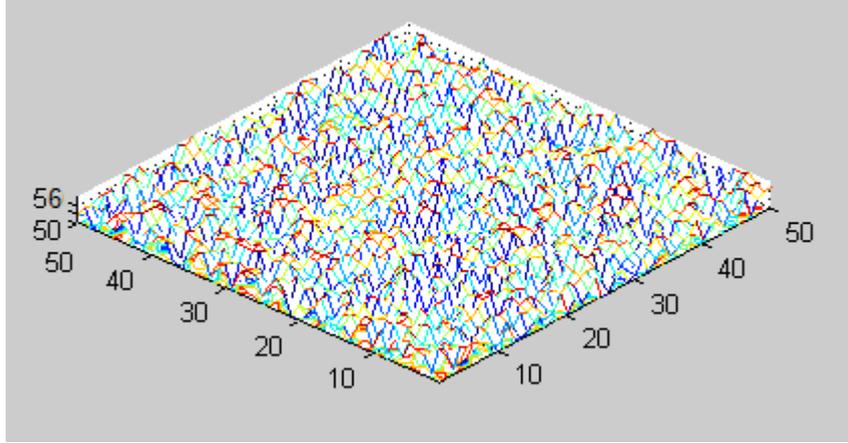
Figure 7.2: Synthetically generated rough surface for testing of Phong BRDF parameter recovery algorithms.

     labels corresponding to the true range of the surface at each corner vertex node i,

5.   Run the LM, MCMC and PMC parameter estimation algorithms, storing the results,

6. end (1)

7. From the resulting data, analyse the convergence characteristics, and tabulate or plot
    graphically the results for each method.

7.3   Multiple-seed Levenberg-Marquardt, dynamically weighted MCMC and PMC convergence results for synthetic data

In this section we report on convergence results for three different strategies for estimating BRDF parameters, namely the Multiple-seed LM minimization method, the dynamically weighted MCMC method, and the Population Monte Carlo method.

7.3.1   Multiple-seed Levenberg-Marquardt minimization

The first method is a Levenberg-Marquardt nonlinear optimization based approach (see Appendix D for a summary of nonlinear optimization methods and a description of the LM method in particular). The error function supplied to this method is

$$\epsilon(i) = 1/p(x_i|x_{\mathcal{N}_i}, \theta) = 1/\frac{\exp(-U_i^A(x_i, x_{\mathcal{N}_i}, \theta) - U_i^R(x_i, z_i)}{\sum_{s \in \mathcal{L}} \exp(-U_i^A(s, x_{\mathcal{N}_i}, \theta) - U_i^R(s, z_i)}. \tag{7.10}$$

Interestingly, the convergence results were found to be superior to those obtained when using an error function of

$$\epsilon(i) = 1 - p(x_i | x_{\mathcal{N}_i}, \theta). \tag{7.11}$$

Therefore, the function shown in Eqn. 7.10 was used for the LM convergence experiments.

There are as many elements in the error vector ϵ as there are corner vertex nodes in the image. The LM function, which returns both a locally optimal parameter value and its corresponding error, is

$$[\theta_{\min}, \text{err}] = \text{LM}(\text{seed}, \mathbf{S}, \mathbf{X}, \vec{L}, \mathbf{P}, \mathbf{I}^1), \tag{7.12}$$

where "seed" indicates the initial parameter set for θ, $\mathbf{S}$ is the set of labels for $\mathbf{X}$ corresponding to the true surface, $\mathbf{X}$ is the range disparity field, and $\mathbf{I}^1$ is the reflectance image rendered using the true BRDF parameter set. As before, $\vec{L}$ is the light source direction, and $\mathbf{P}$ is the projective camera matrix data (for one camera), and reflectance images are rendered given a BRDF model for each intermediate parameter set in the LM algorithm. These parameters are used to calculate the potential energy terms for Eqn. 7.10, so that the parameter set $\theta_{\min}$, which contains the BRDF parameters producing the smallest error may be calculated.

The LM method does not return any kind of posterior distribution, only the parameter vector which is a local (and hopefully global) minimum of the error function (within the specified tolerance of error and the user-specified limit of LM iterations).

The convergence results of the Multiple-seed LM method tested on synthetic data are shown in Figs. 7.3 to 7.6. To ameliorate the problem of the LM minimizer finding bad local probability maxima, we implemented a multiple-seed version of the algorithm, initializing the BRDF parameter set at regularly spaced points (seeds) within the space of possible BRDF parameter values (distributed as the samples in Fig. 7.15), then running the LM minimizer once for each of these seeds. The more seeds tested, the more likely it is that the resulting minimum-error parameter set is in fact the global minimum for the cost function.

The results of the iterative minimizations for the Multiple-seed LM minimizer, using the error function of Eqn. 7.10, are shown in Figs. 7.3 to 7.6. The LM algorithm was run 20 times on synthetic data, with 512 seeds. To test the convergence stability of the algorithm, subsets of the seeds and their corresponding final optimized values were randomly selected from the 512 seeds. The number of seeds in these subsets ranges from 1 to 512. The plots in Figs. 7.3 and 7.4 show the average distance between the true BRDF parameter set and the best LM-optimized sample per sample subset. The Multiple-seed LM algorithm is described as follows:

7.3.2   Multiple-seed Levenberg-Marquardt for MRF parameter estimation

1. Initialize 512 seeds evenly spaced on the BRDF parameter space.

2. For n=1, . . . ,N

3.   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

4.   Generate random BRDF parameter vector $\vec{b}^n$, and render intensity map $\mathbf{I}^1$ of surface
        given lighting, camera and reflectance parameters

5.   For i=1, . . . ,512

6.   Generate seed_i (seeds are positioned at discrete intervals in the cube of parameter space)

7.   Calculate $[\theta_i^n, \text{err}_i^n] = \text{LM}(\text{seed}_i, \mathbf{S}, \mathbf{X}, \vec{L}, \mathbf{P}, \mathbf{I}^1)$

8.   end(5)

9. end(2)

10. For n=1, . . . ,N

11.   For j=1, . . . ,512

12.   Sample a random subset u^n consisting of j non-duplicate elements from the set (θ^n, err^n)

13.   Find the parameter vector θ_{min}^n in u^n corresponding to the minimum value of err^n in u^n

14.   Calculate $d_{j,\text{min}}^n = ||\theta_{\text{min}}^n - \vec{b}^n||$

15.   end(11)

16. end(10)

17. Calculate $\overline{d}_{j,\text{min}} = \frac{1}{N} \sum_n d_{j,\text{min}}^n$.

After running this battery of minimizations, $\overline{d}_{j,\text{min}}$ contains the average minimum error difference between the parameter vector of the LM calculated optimum value and the true BRDF parameter vector used to render the image of the synthetic surface. This value is calculated for the range of subset sizes (each subset is populated with randomly chosen entries from the set of LM-minimized parameter vectors from the 512 seeds). The values for $\overline{d}_{j,\text{min}}$ for the Ward parameterisation, plotted against the number of seeds j in the subset, are shown in Fig. 7.3. The results using the anisotropic Phong parameterisation, with the rough synthetic surfaces shown in Fig. 7.2, are shown in Fig. 7.4.

It can be seen by examining Figs. 7.3 and 7.4 that the more seeds that the multiple-seed LM algorithm is initialized with, the greater the likelihood of having a low minimum error. The smaller the number of seeds, the greater the risk that the LM algorithm will converge into bad local minima (represented by the peaks in these two sets of graphs). The number of disparity labels for the synthetic surface data is indicated in the graphs by "labels". The errors tend to decrease with more disparity labels.

Also shown in Figs. 7.5 and 7.6 are the convergence results for a single Multiple-seed LM run, with 512 seeds (a different, random parameter vector was used to render the surfaces on which the LM algorithm
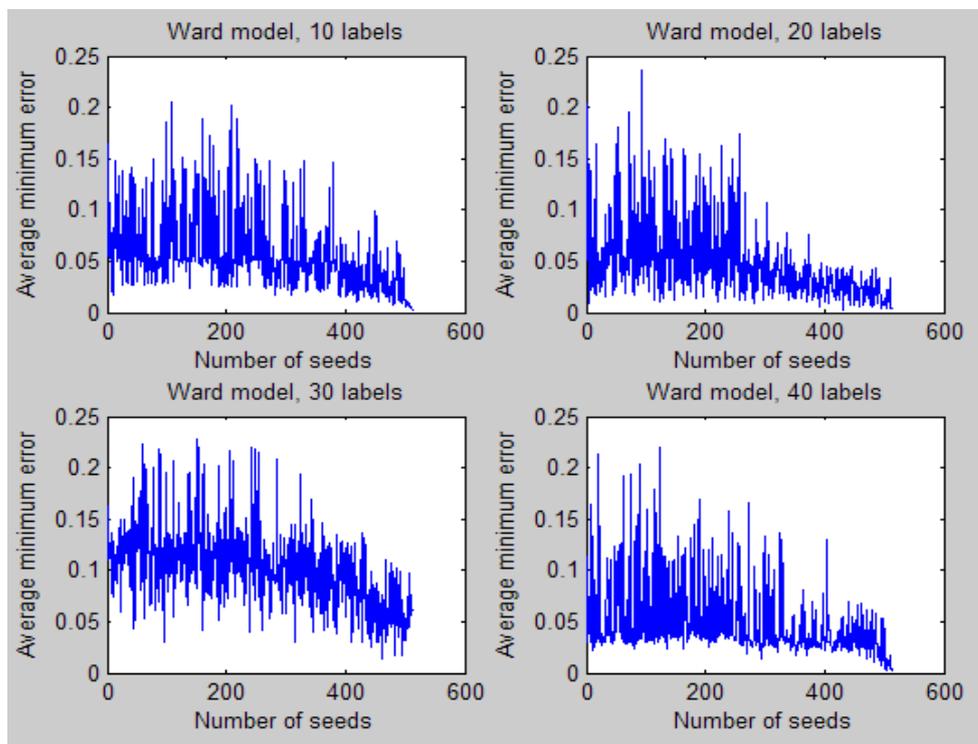
Figure 7.3: Average convergence results over 20 runs for the LM algorithm, run using the Ward reflectance model to render randomly generated surfaces.
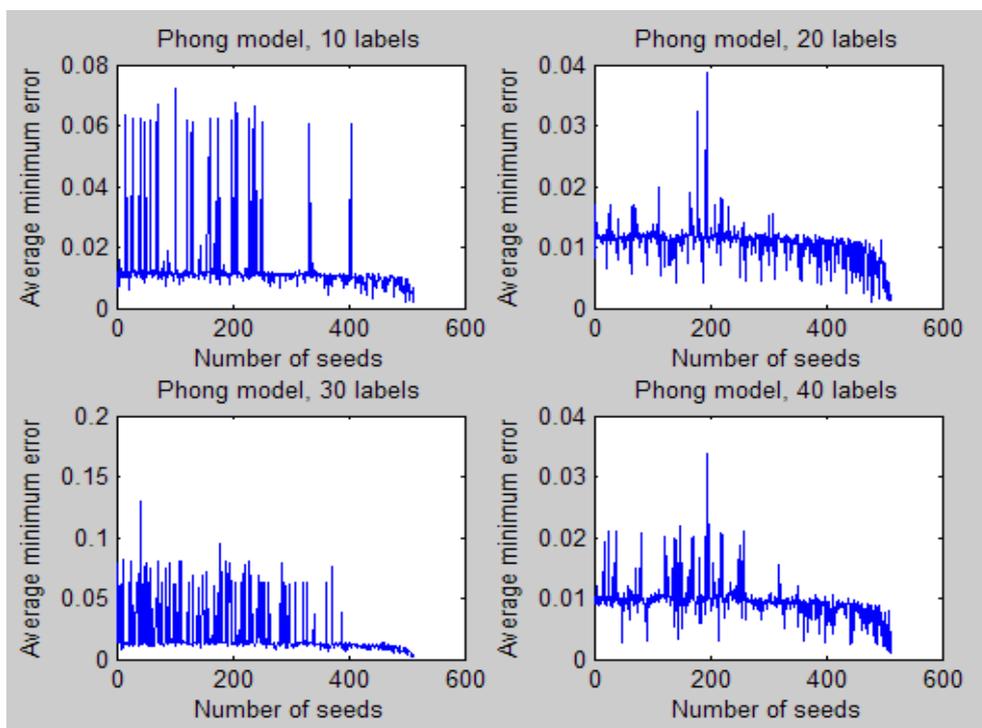
Figure 7.4: Average convergence results over 20 runs for the LM algorithm, run using the anisotropic Phong model to render randomly generated surfaces.
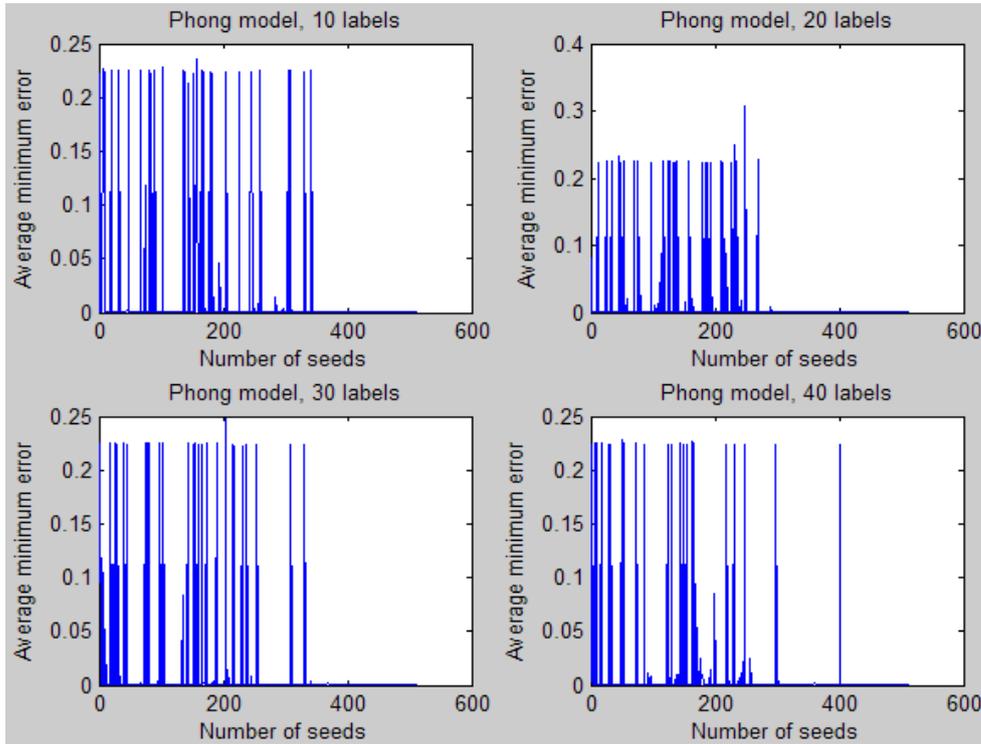
Figure 7.5: Convergence results for a single randomly selected run for the Multiple-seed LM algorithm, run using the anisotropic Phong reflectance model to render randomly generated surfaces.

was run in each of the four subgraphs in Figs. 7.5 and 7.6). The results are for the same algorithm as is presented in section 7.3.2, but with N=1 (in Figs. 7.3 and 7.4, N=20). A comparison of Figs. 7.5 and 7.6 with Figs. 7.3 and 7.4 shows us that there are more peaks in Figs. 7.3 and 7.4, which is expected since these graphs are averaged results over many runs. In the example Multiple-seed LM runs shown in Figs. 7.5, the optimal parameter sets are found reliably using subsets of 400 seeds or more. In the example Multiple-seed LM runs shown in Figs. 7.6, the optimal parameter set is found within a parameter seed subset of size 220 for the case of 10 disparity labels (upper left quadrant), within a subset size of 350 seeds for 20 disparity labels (upper right quadrant). For the case of 30 disparity labels, a parameter seed subset size greater than 150 is sufficient to find a good solution (one within 0.015 units Euclidean distance from the true Ward parameters). Finally, for the case of 40 disparity labels (bottom right quadrant), the final peak is at a parameter seed subset size of 250. Note however that the Multiple-seed LM algorithm as described in section 7.3.2 indicates that for displaying the results, the subsets are chosen randomly. Therefore the results for an individual run of the Multiple-seed LM algorithm with 512 seeds are of limited benefit, although they do give some kind of insight about the distribution of erroneous results (the peaks in the plots), and can be used to estimate the number of parameter seeds needed to reduce the risk of deriving a best Multiple-seed parameter set that is at a bad local probability maximum.
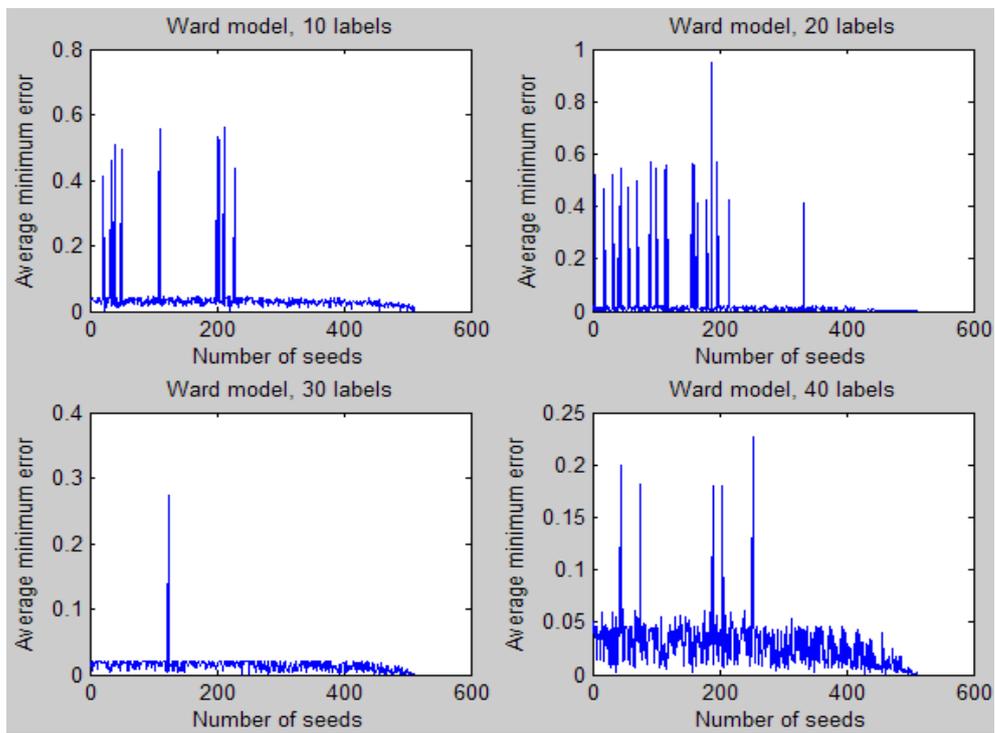
Figure 7.6: Convergence results for a single randomly selected run for the Multiple-seed LM algorithm, run using the Ward reflectance model to render randomly generated surfaces.

7.4 R-type and Q-type dynamically weighted MCMC convergence results

The MCMC method we implemented was the dynamically weighted MCMC sampler of [68], which, like the PMC method, uses the pseudolikelihood function for each sample as the target distribution probability. This dynamically weighted variant was chosen because it was found that the classical MCMC sampler did not explore the parameter space in a satisfactory way. The dynamically weighted MCMC sampling method returns a posterior distribution on the BRDF parameters which is more densely sampled than the Population Monte Carlo method, but is less efficient in finding the global maxima of the distributions.

The posterior distribution of Eqn. 7.7 for a range of synthetic surfaces was explored as the target distribution $\varrho(\cdot)$ using the R-type and Q-type dynamically weighted MCMC samplers, with $\theta = 1$. The proposal distribution $T(x, y)$ is a Guassian kernel with $\sigma = 0.05$ in $\mathbb{R}^3$, centered on the previous sample. The initial sample in each experiment is initialized with θ_0 (we use θ instead of X in Eqns. 5.48 and 5.49, to represent the BRDF parameter set) sampled from the uniform distribution in the unit cube in $\mathbb{R}^3$, and $\Pi_0 = \varrho(\theta_0)$, i.e.

$$\varrho(\theta_0) = ePL_s(\mathbf{X}|\theta_0), \tag{7.13}$$

where as before $\mathbf{X}$ is the set of random variables representing the states on the corner vertex nodes in image $\mathbf{I}^1$. Sample proposals outside the unit cube of parameter space are automatically rejected. In each case, the sampler was allowed to run for 20000 iterations. The computational load is therefore approximately equivalent to the PMC algorithm run with 1000 samples per sample set over 20 iterations. For each sample run, the distance between the parameter set of the MAP sample and the true BRDF parameters used to generate the surface, at each time step, is stored. These MAP distances for each time step are averaged over 100 MCMC runs. This enables visualization of the average convergence rates for each of these MCMC samplers. The methodology for collecting and displaying the average results for the dynamically weighted MCMC sampler follows:

1. For n=1,...,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 Q-type or R-type dynamically weighted MCMC algorithm for 20000 iterations, using Eqn. 7.7 as the target distribution, storing each sample set (θ_i^n, Π_i^n), while maintaining a running store of the sample $\theta_{i,\max}^n$ which has the highest probability $\varrho(\theta_i^n)$, up to the current value i.
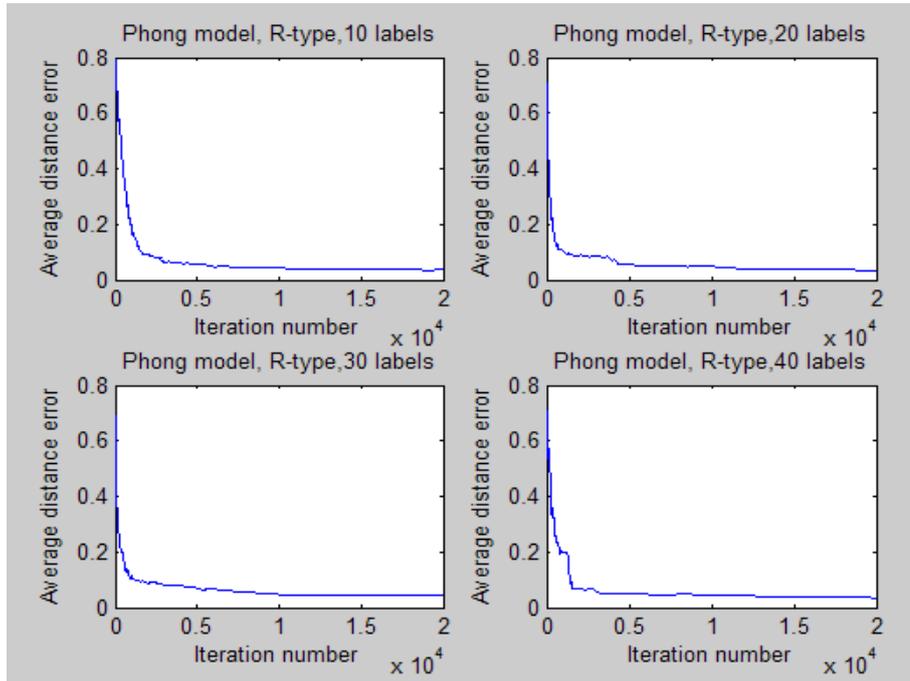
Figure 7.7: Average convergence results for the dynamically weighted MCMC algorithm with R-type moves using the Phong model.

5. end(1).

6. For i=1,...,20000

7.   Calculate $d_i = \frac{1}{N} \sum_{n=1}^{N} ||\vec{b}^n - \theta_{i,\max}^n||$.

8. end(6).

In the above, we have N=100 experiments for each combination of BRDF model (Ward or Phong) and number of labels (10,20,30,40), for both the Q-type and R-type samplers. The graphs shown in Figs. 7.7 to 7.10 plot the average values d_i against the sample number i.

Fig. 7.7 shows the average convergence results using synthetic data for the R-type sampler, using the anisotropic Phong reflectance model. In Fig. 7.8, the average convergence results using synthetic data for the R-type sampler, using the Ward reflectance model, are shown. Figs. 7.9 and 7.10 show the same data, but using the Q-type sampler. The convergence results are very similar. Generally, the error value approaches 0.1 (which is an error value similar to the error value obtained by the LM or PMC based methods), and on average this takes about 20000 samples to reach. Furthermore, 20000 weighted samples is too great a number to be used effectively in many of the pdf comparison measures used for classification of surfaces (described in the next chapter).

In Figs. 7.11, 7.12, 7.13 and 7.14 are shown more convergence results for the dynamically weighted MCMC sampler. Each plot shows the convergence behaviour for a single randomly chosen sample
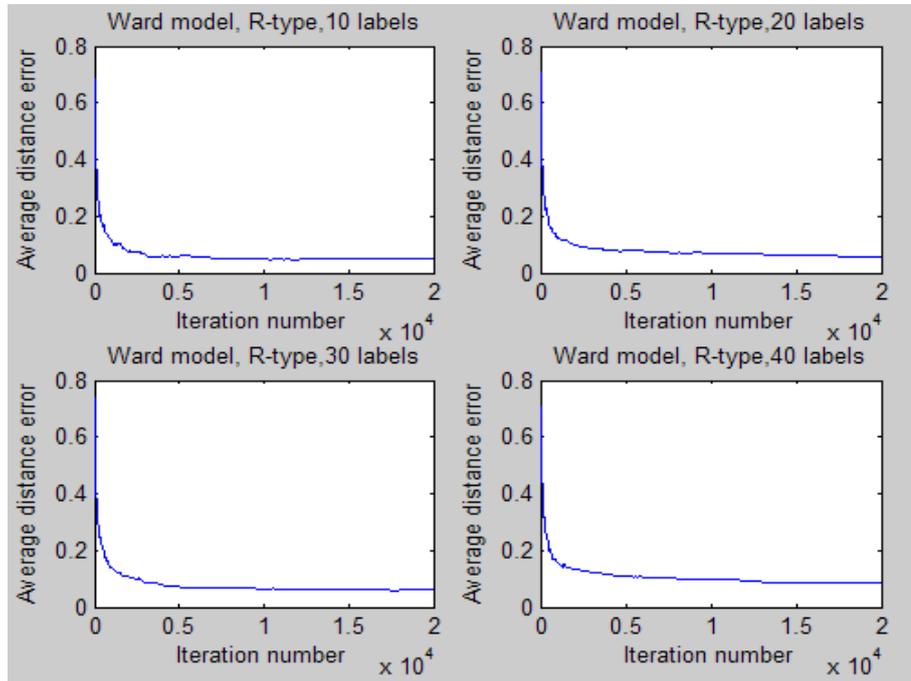
Figure 7.8: Average convergence results for the dynamically weighted MCMC algorithm with R-type moves using the Ward model.
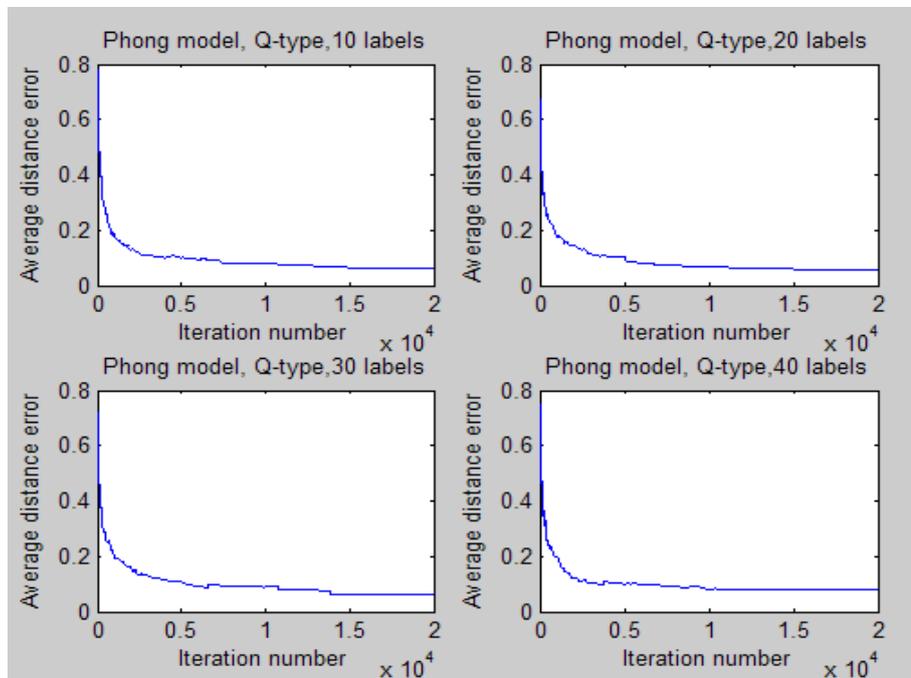
Figure 7.9: Average convergence results over 100 runs for the dynamically weighted MCMC algorithm with Q-type moves, using the Phong model.
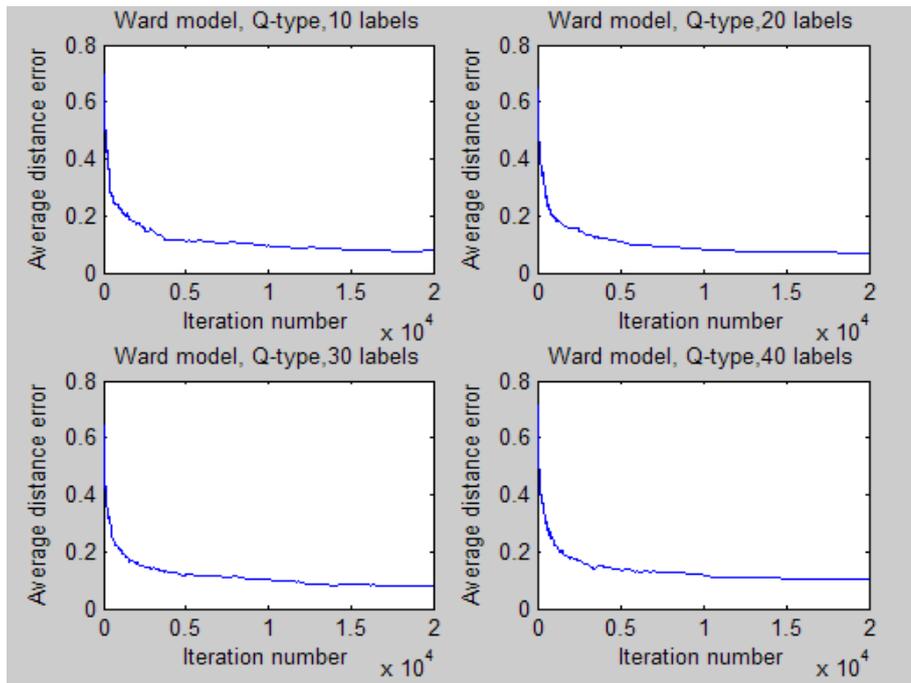
Figure 7.10: Average convergence results over 100 runs for the dynamically weighted MCMC algorithm with Q-type moves, using the Ward model.
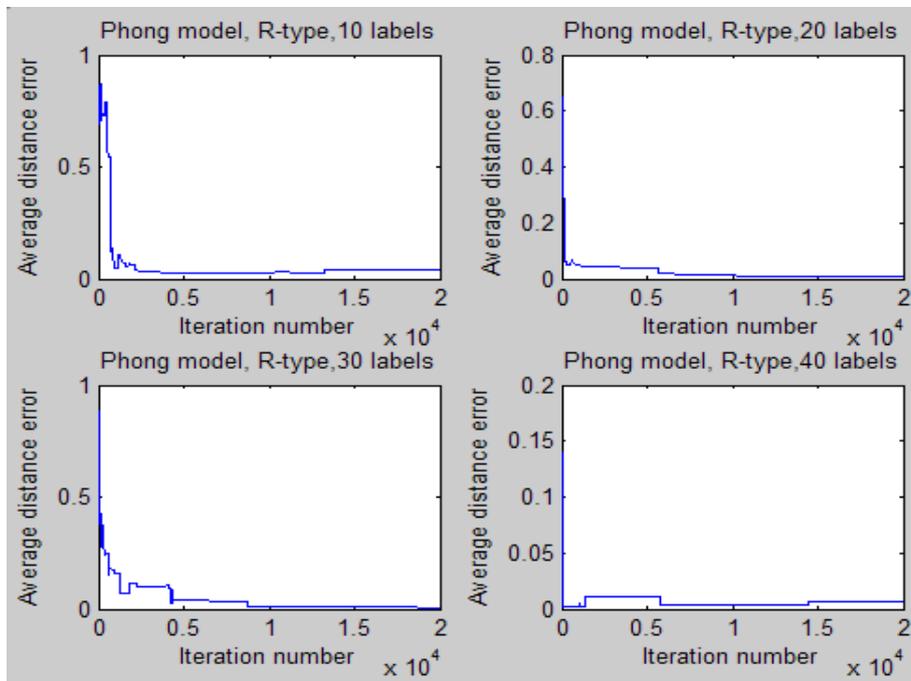
Figure 7.11: Convergence results for one randomly selected example, run using the dynamically weighted MCMC algorithm with R-type moves using the Phong model.
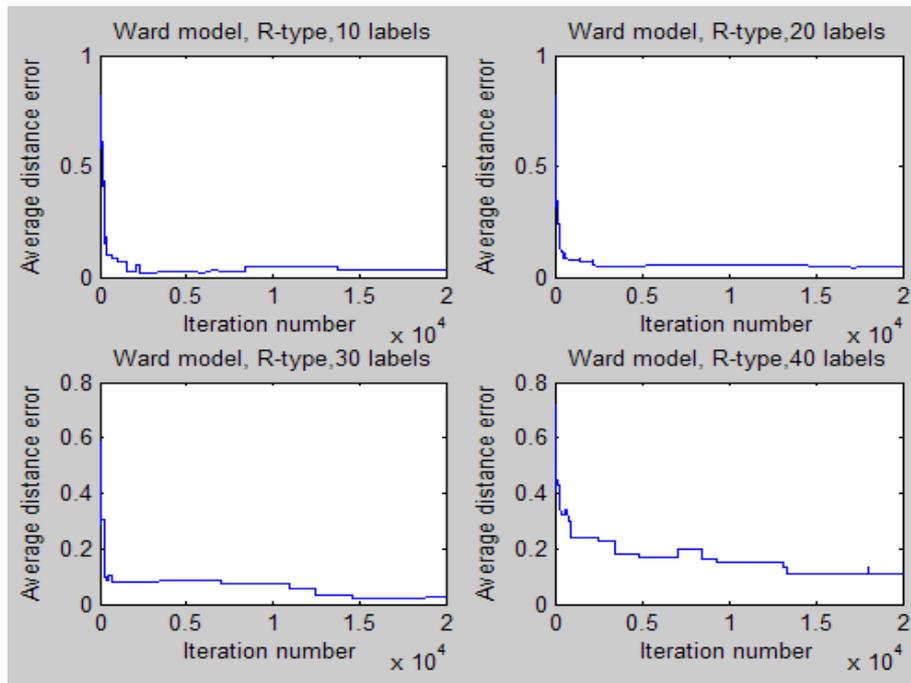
Figure 7.12: Convergence results for one randomly selected example, run using the dynamically weighted MCMC algorithm with R-type moves using the Ward model.
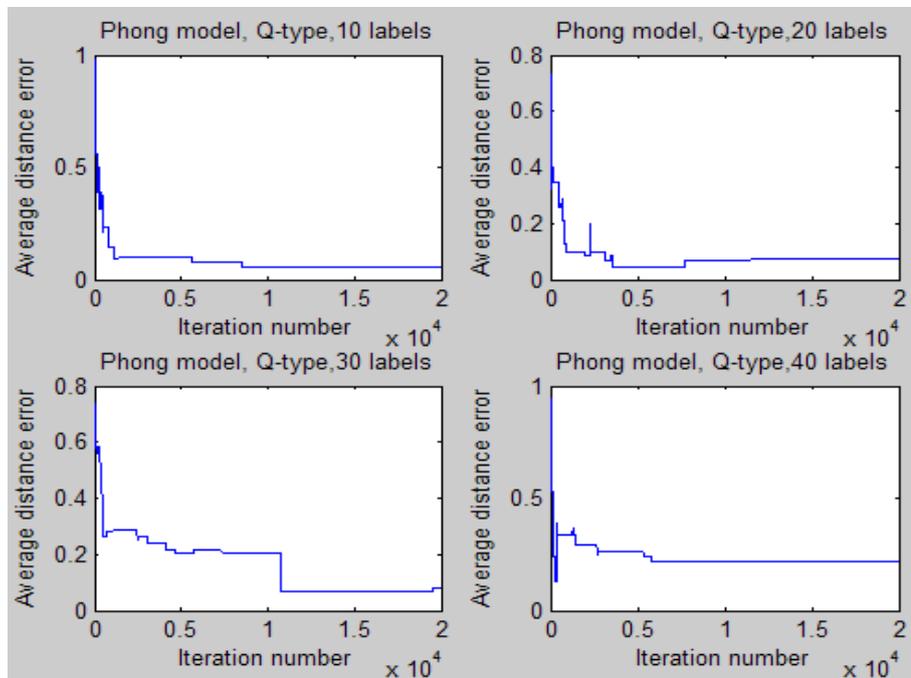
Figure 7.13: Convergence results for one randomly selected example, run using the dynamically weighted MCMC algorithm with Q-type moves, using the Phong model.
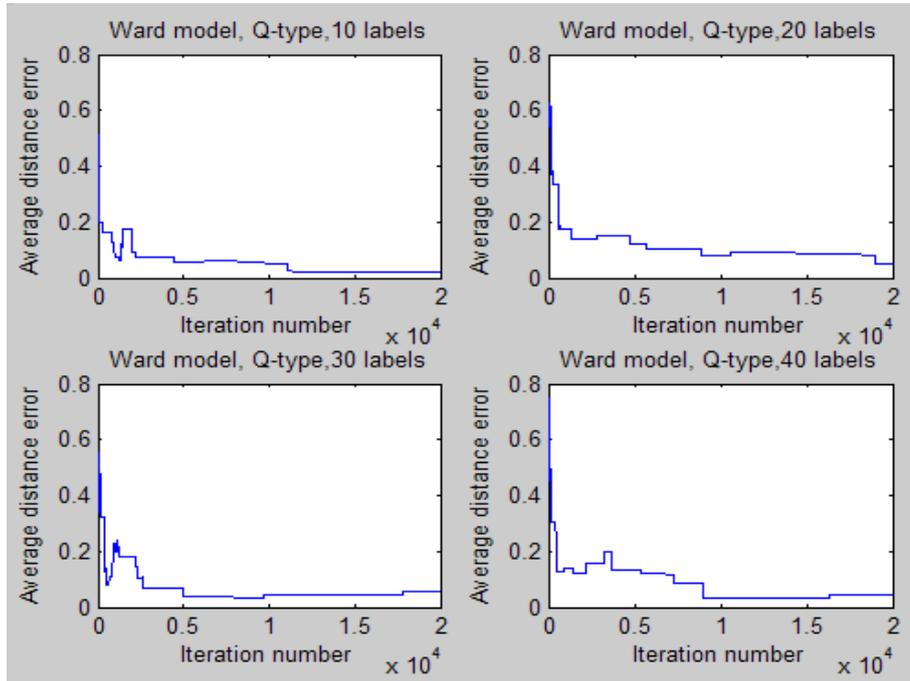
Figure 7.14: Convergence results for one randomly selected example, run using the dynamically weighted MCMC algorithm with Q-type moves, using the Ward model.

run. It is seen that the MAP samples do not move steadily towards the true parameter set used to generate the synthetic surfaces, but sometimes move away from it (upwards sloping parts of graphs in Figs. 7.11..7.14). This is indicative of the nature of the target distribution probability function of the pseudolikelihood (Eqn. 7.7), which has erroneous local maxima. When a MAP sample is close to one of the false local maxima, it is counted as the best sample, but it is actually further from the true parameter set.

7.4.1 Convergence results for the PMC method

The posterior distribution of Eqn. 7.7 for a range of synthetic surfaces was also 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,. . .,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. 7.7 as the target distribution, storing each sample set $(\overrightarrow{\theta}, \overrightarrow{\pi})^{n,i}$ 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 7.1 to 7.4, per PMC iteration, in the columns labelled "MAP dist".

7. The variance of the sample set weights $\overrightarrow{\pi}^{n,i}$ is calculated for each iteration, and averaged over
　　the number of trial runs (N=100). These are also recorded in Tables 7.1 and 7.2, 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).

Some visualizations of the PMC algorithm in operation are shown:

Visualization of PMC algorithm operation

In Figs. 7.15 to 7.28, volumetric plots show the distribution of the pseudolikelihood function $ePL_3(\mathbf{X}|\theta)$ of Eqn. 6.62, sampled at regular points in the cube of parameter space, with the locations of the samples at different iterations of the PMC algorithm superimposed. The data is plotted at intervals of 5 iterations of the PMC algorithm's operation. It can be seen how the samples evolve and change position as the PMC algorithm iterates. The volumetric slices in these plots are chosen (almost randomly) to assist in the visualization of the probability density function. The particle weights in these plots are not directly visible. Figs. 7.15 to 7.19 show five iterations of the PMC algorithm, run on a surface rendered with the Ward reflectance model. Figs. 7.20 to 7.24 show another example using the Ward reflectance model, with a different synthetic surface and parameter set. Figs. 7.25 to 7.28 show five iterations of the PMC algorithm run on a surface rendered using the anisotropic Phong model.
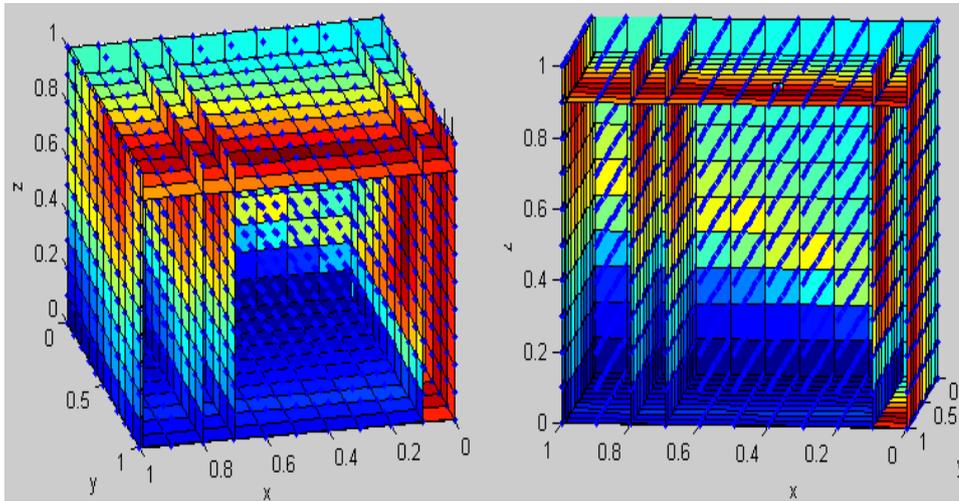
Figure 7.15: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.7, 0.4, 0.9) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 1. Two different views are shown.
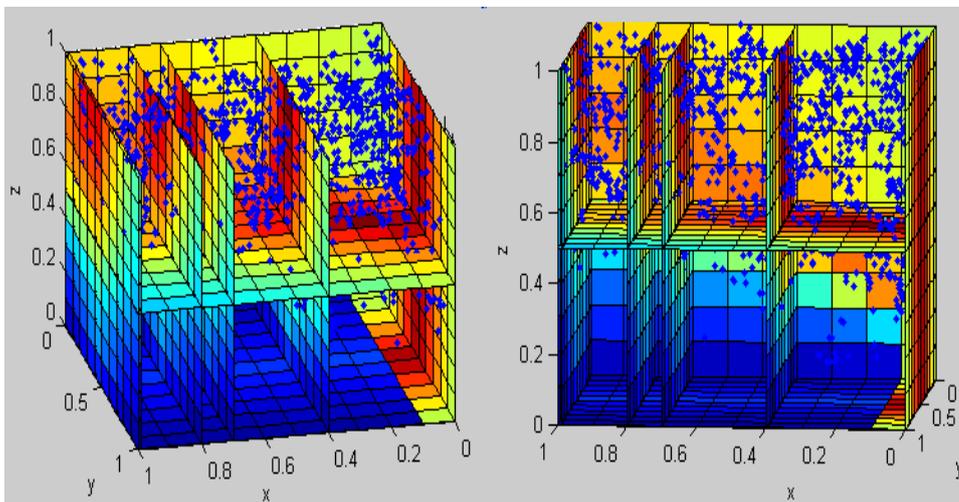
Figure 7.16: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.7, 0.4, 0.9) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 5. Two different views are shown.
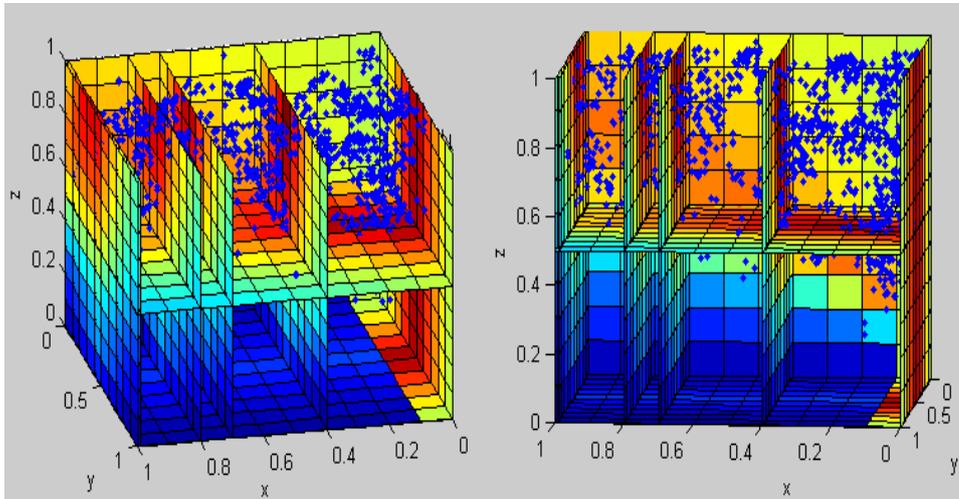
Figure 7.17: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.7, 0.4, 0.9) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 10. Two different views are shown.
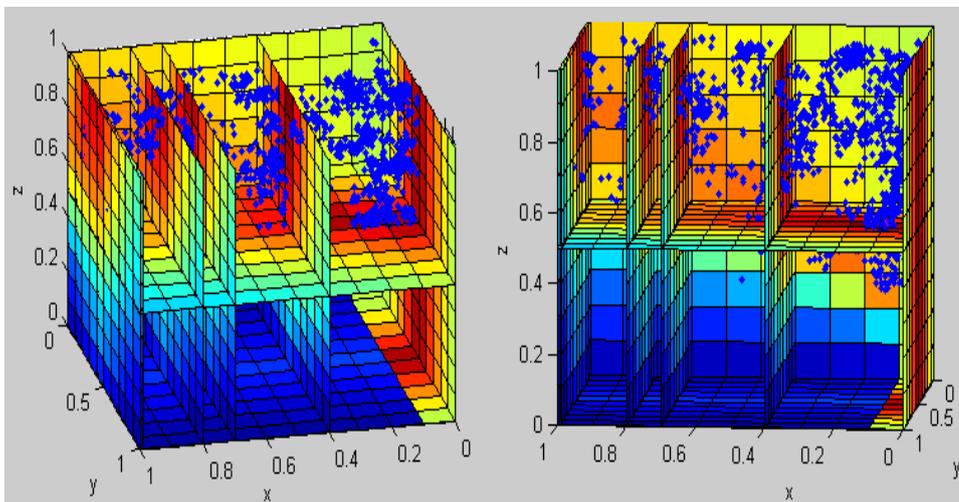
Figure 7.18: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.7, 0.4, 0.9) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 15. Two different views are shown.
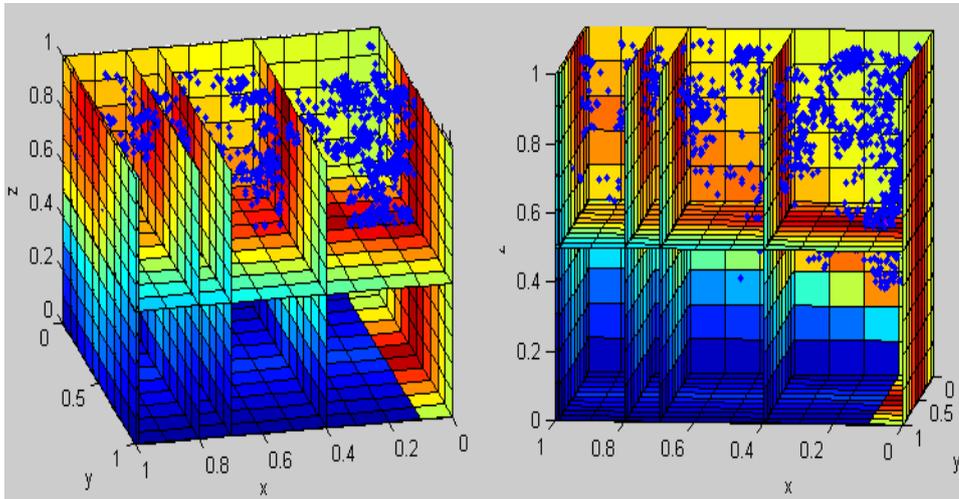
Figure 7.19: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.7, 0.4, 0.9) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 20. Two different views are shown.
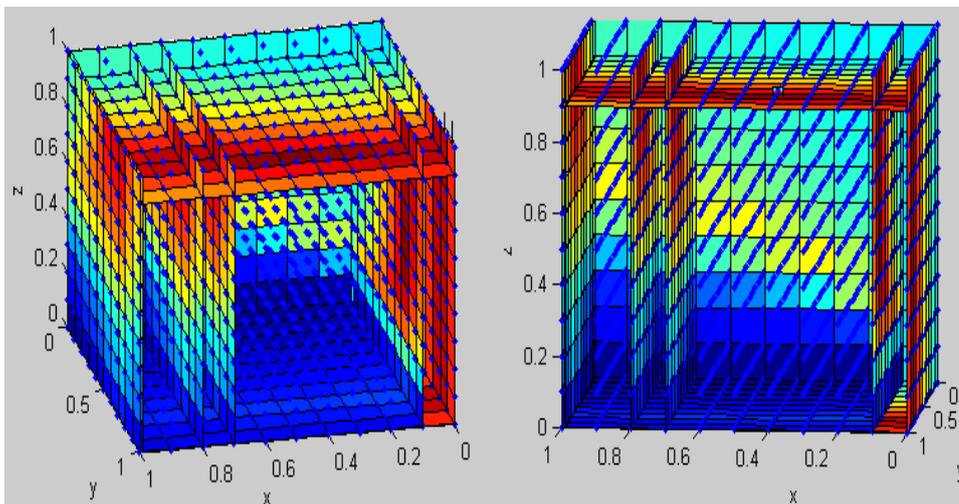
Figure 7.20: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.8, 0.9, 0.5) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 1. Two different views are shown.
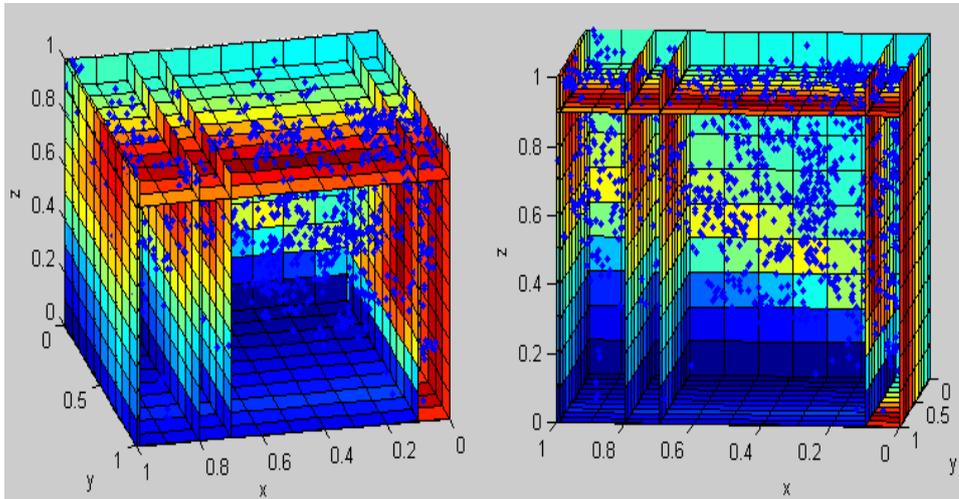
Figure 7.21: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.8, 0.9, 0.5) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 5. Two different views are shown.
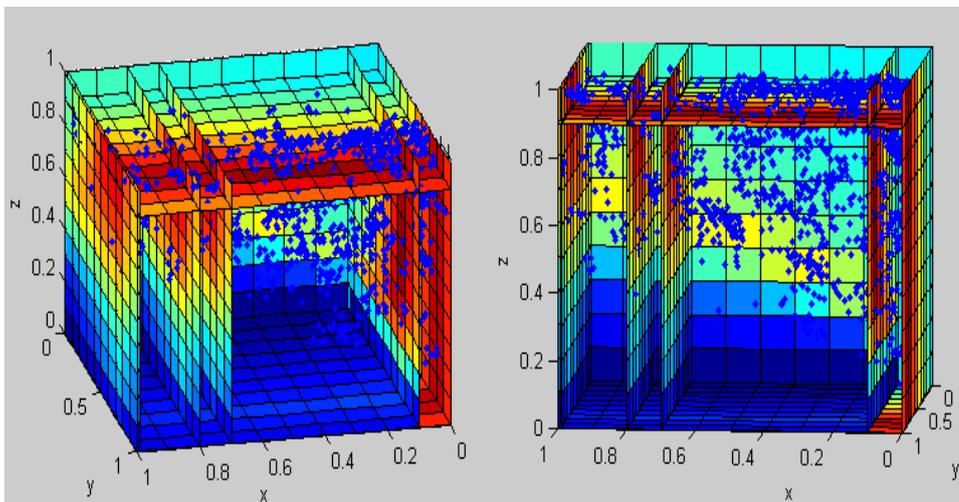
Figure 7.22: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.8, 0.9, 0.5) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 10. Two different views are shown.
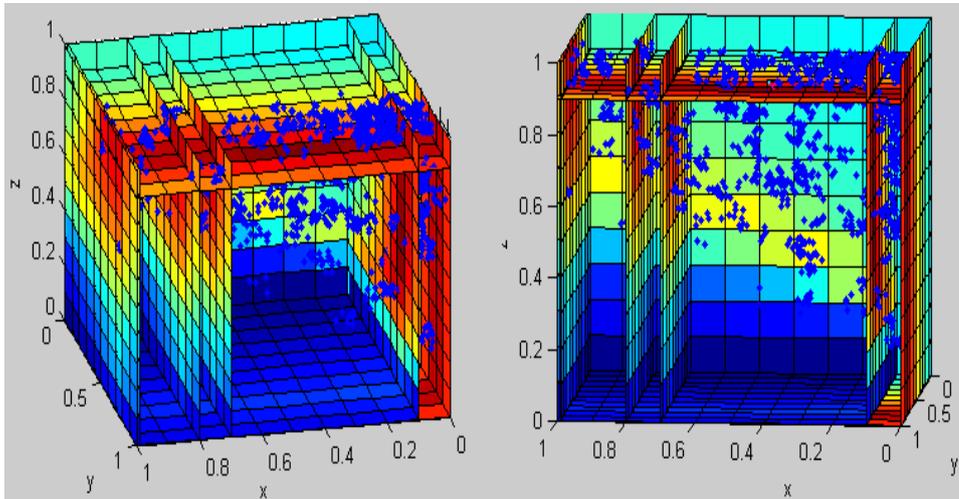
Figure 7.23: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.8, 0.9, 0.5) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 15. Two different views are shown.
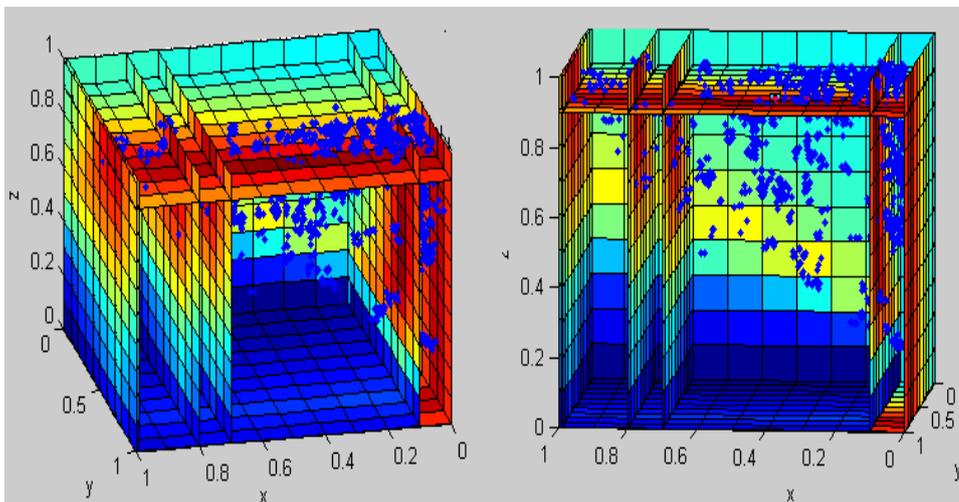
Figure 7.24: Posterior distribution of Ward parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.8, 0.9, 0.5) = (\rho_d, \rho_s, 2\alpha)$. The points indicate samples for PMC iteration number 20. Two different views are shown.
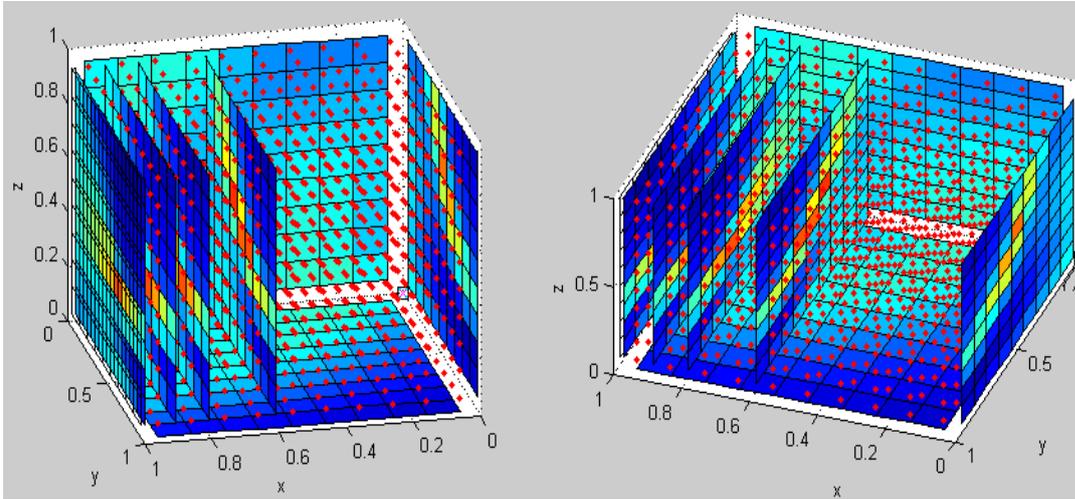
Figure 7.25: Posterior distribution of anisotropic Phong parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.8, 0.9, 0.5) = (R_d, R_s, n/25)$, where $n = n_u = n_v$. The points indicate samples for PMC iteration number 1. Two different views are shown.
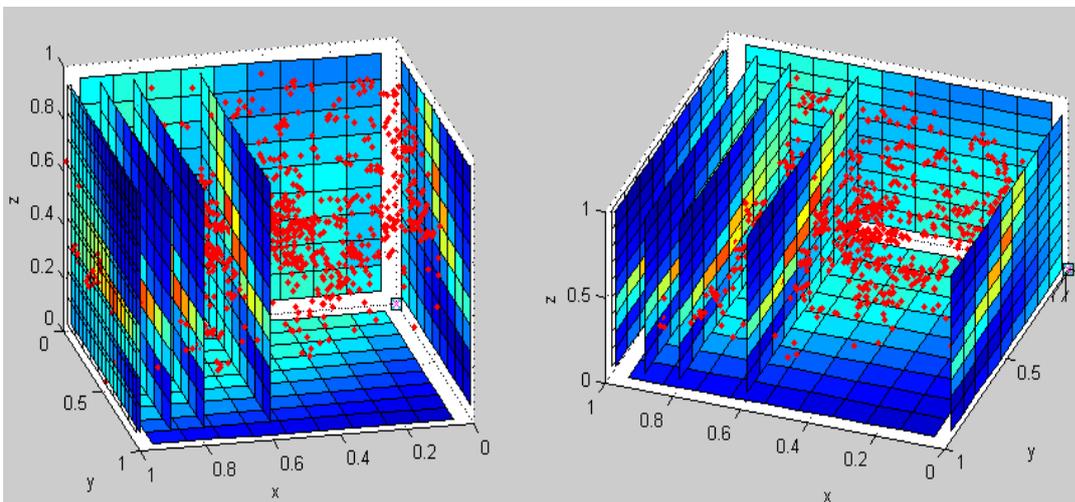
Figure 7.26: Posterior distribution of anisotropic Phong parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.8, 0.9, 0.5) = (R_d, R_s, n/25)$, where $n = n_u = n_v$. The points indicate samples for PMC iteration number 5. Two different views are shown.
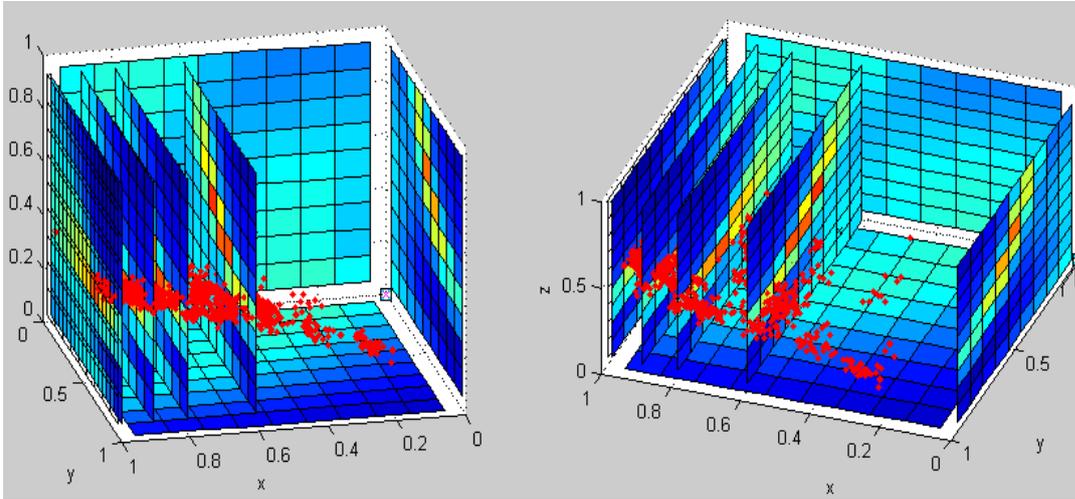
Figure 7.27: Posterior distribution of anisotropic Phong parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.8, 0.9, 0.5) = (R_d, R_s, n/25)$, where $n = n_u = n_v$. The points indicate samples for PMC iteration number 10. Two different views are shown.
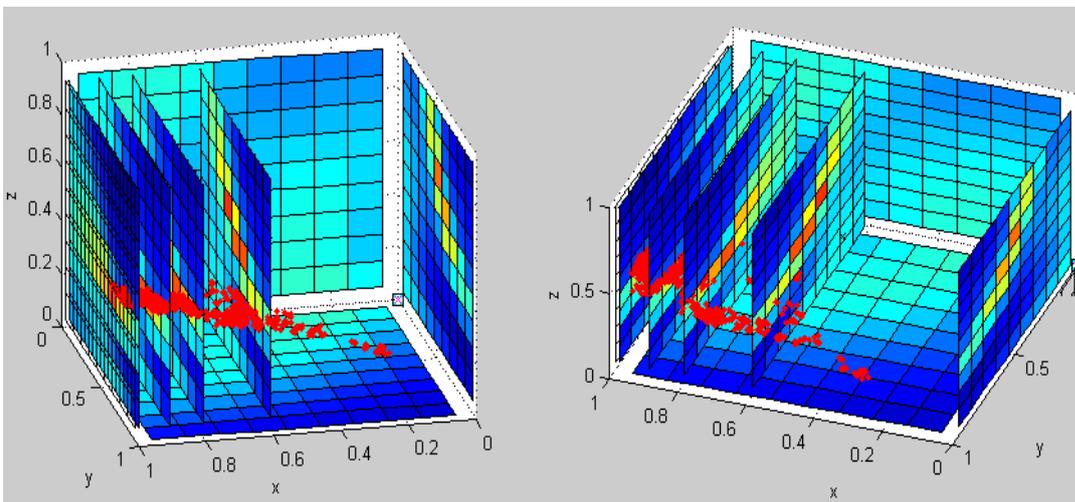
Figure 7.28: Posterior distribution of anisotropic Phong parameterisation calculated from synthetically generated probabilistic surface. The true parameter values in this plot are $(x, y, z) = (0.8, 0.9, 0.5) = (R_d, R_s, n/25)$, where $n = n_u = n_v$. The points indicate samples for PMC iteration number 20. Two different views are shown.

The convergence results of the PMC BRDF parameter estimation method are shown in Tables 7.1 to 7.4. The PMC algorithm was run using 1000, 125, and 64 samples (recorded in the tables as "numSamples"), over 20 iterations, for different numbers of possible synthetic corner vertex node labels. The importance functions of Eqn. 7.9 were used for this, in the PMC algorithm described in section 6.6.2.

The number of labels per synthetic corner vertex node could take on the values of 10,20,30 and 40. A value of 40 for the number of labels ("numLabels") corresponds to the equivalent of a disparity range of 40 pixels, in the case of real stereo image data (we are however using synthetic data at this stage, so the number of labels corresponds to the number of discretizations of the range of the point along the ray through the camera center). The Tables 7.1 to 7.4 also record convergence information with respect to the average Euclidean distance of the MAP sample from the true value (in the column "MAP dist"), as well as changes in the average sample variance (in the column "var", where "$\times 10^{-6}$" indicates that the values in the column should be multiplied by $\times 10^{-6}$ to obtain the correct variances).

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 7.1 to 7.4 are average values over 100 runs. The convergence results for the "MAP dist" statistic appears to be better when using the anisotropic Phong reflectance model than when using the Ward reflectance model. This is because the algorithms tested using the anisotropic Phong models were run using jagged surface data (Fig. 7.2), whereas the Ward model experiments were done on smooth surfaces. These results are presented in table form since there are too many variables to easily distinguish graphically plotted data.

Tables 7.5, 7.6, 7.7 and 7.8 contain PMC convergence statistics for the Ward and anisotropic Phong models, for the case where a single randomly chosen run is examined (rather than averaging the results over 100 runs, as is done in Tables 7.1,7.2,7.3 and 7.4). This data gives insight about the behaviour that one could expect for a particular PMC run. Note that in all the Tables 7.1..7.8 there is a sudden drop in the sample variance at the 20th PMC iteration (last row in each table). This is because after the last iteration, the samples are reweighted with the unsoftened pseudolikelihood functions of Eqn. 7.7, which is a sharper function than the functions to weight the samples in the previous iterations (Eqn. 7.9).

When examining Table 7.5, one can see by inspection of columns 2,4 and 6 that the Euclidean distances of the MAP particle to the true BRDF parameter set, after 20 iterations, are 0.026, 0.21 and 0.35. These distances are large, indicating that this table represents a poor sample run for those values (numLabels = 10, numSamples = {1000,125,64}). The errors improve for the example sample runs where "numLabels" is greater than 10 (see last row corresponding to 20th PMC iteration. The sample runs described in

numLabels	10	10	10	10	10	10	20	20	20	20	20	20
numSamples	1000	1000	125	125	64	64	1000	1000	125	125	64	64
PMC iters	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$
1	0.3439	18.2	0.4054	38.2	0.489	671	0.2863	3.609	0.4547	343.1	0.5139	507.7
2	0.23	1.975	0.3763	33.35	0.4878	191.9	0.1587	0.8979	0.3991	113.3	0.4925	210.2
3	0.1872	0.887	0.355	31.79	0.4643	104.1	0.1356	0.2929	0.3955	61.44	0.4851	91.51
4	0.1864	0.594	0.3419	29.01	0.459	47.61	0.1252	0.1789	0.3754	60	0.4571	54.43
5	0.1829	0.4873	0.3318	17.94	0.4664	31.86	0.1266	0.148	0.362	54.08	0.4574	47.1
6	0.1693	0.456	0.318	54	0.463	37.39	0.1115	0.1217	0.3278	56.76	0.4523	42.57
7	0.1595	0.4411	0.3122	92.38	0.4647	49.42	0.1142	0.09438	0.3217	39.96	0.4475	39.58
8	0.155	0.4097	0.2992	65.05	0.4398	34.72	0.1258	0.08784	0.3239	49.76	0.4356	51.65
9	0.1531	0.3693	0.306	90.41	0.435	60.79	0.1221	0.07107	0.3243	36.71	0.4413	33.72
10	0.1499	0.3678	0.2959	70.44	0.4397	35.9	0.1126	0.0616	0.3233	35.01	0.4348	48.48
11	0.1578	0.3439	0.2875	39.84	0.4389	20.02	0.1109	0.05896	0.3187	34.53	0.4268	14.03
12	0.1454	0.3274	0.2733	60.29	0.4335	30.38	0.1219	0.05774	0.3146	32.28	0.4288	19.21
13	0.1446	0.3042	0.2752	41.95	0.4262	20.47	0.1109	0.04885	0.3066	27.7	0.4211	26.8
14	0.1365	0.2913	0.2734	31.16	0.4221	14.33	0.119	0.04144	0.3105	22.14	0.4183	20.45
15	0.1432	0.2662	0.2731	48.82	0.4332	22.6	0.1026	0.03878	0.3111	22.42	0.4192	25.01
16	0.1511	0.2603	0.2775	38.36	0.4219	25.88	0.1146	0.05723	0.3027	19.77	0.407	27.55
17	0.1367	0.2577	0.2765	30.33	0.4325	10.96	0.1025	0.06599	0.3042	30.07	0.4007	15.97
18	0.1312	0.2016	0.2771	47.14	0.4256	33.04	0.1036	0.05574	0.3022	22.81	0.4092	22.34
19	0.1276	0.1784	0.2792	41.68	0.4275	35.52	0.09577	0.02362	0.2935	18.92	0.4075	41.43
20	0.1324	0.1509	0.2831	21.13	0.4231	2.829	0.1043	0.01832	0.2979	16.03	0.4029	1.434

Table 7.1: 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.

numLabels	30	30	30	30	30	30	40	40	40	40	40	40
numSamples	1000	1000	125	125	64	64	1000	1000	125	125	64	64
PMC iters	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$
1	0.3525	13.2	0.4325	17.7	0.5345	1685	0.3622	4.176	0.4881	30.34	0.4715	3690
2	0.226	0.9745	0.4232	17.62	0.5501	119.3	0.261	1.146	0.4814	20.39	0.4758	167.5
3	0.2024	0.5346	0.4254	24.9	0.5205	49.54	0.1926	1.15	0.4788	18.7	0.4628	91.95
4	0.1746	0.483	0.4374	11.18	0.5403	60.63	0.1956	0.9106	0.4711	10.45	0.4487	72.02
5	0.166	0.4922	0.4129	9.79	0.5344	62.12	0.1605	0.7327	0.473	12.08	0.4274	62.59
6	0.1462	0.4152	0.402	8.759	0.5293	44.85	0.1283	0.6255	0.463	8.152	0.4315	65.71
7	0.1585	0.3615	0.3994	5.312	0.5183	44.23	0.1353	0.5818	0.4613	3.844	0.4241	69.01
8	0.138	0.3642	0.4078	5.761	0.5147	40.84	0.1265	0.5739	0.4504	2.467	0.4263	46.67
9	0.1363	0.4002	0.3966	8.043	0.49	47.86	0.138	0.463	0.4576	7.981	0.4171	61.87
10	0.1396	0.3736	0.4102	10.24	0.4928	24.3	0.138	0.4517	0.4545	4.122	0.4056	54.2
11	0.1383	0.3318	0.3995	6.799	0.4934	30.02	0.1302	0.4178	0.4618	6.183	0.4063	32.29
12	0.131	0.2832	0.41	7.085	0.4963	73.06	0.1279	0.4353	0.4633	11.42	0.4071	49.19
13	0.1424	0.3007	0.3982	4.16	0.4899	41.57	0.1281	0.3804	0.4667	9.817	0.393	34.92
14	0.1359	0.2658	0.3936	5.581	0.4919	33.66	0.1286	0.3689	0.4668	8.161	0.4012	32.39
15	0.1354	0.2186	0.3962	5.621	0.4808	27.12	0.1327	0.3284	0.4662	8.779	0.4064	37.01
16	0.1253	0.2101	0.4009	4.37	0.4815	29.05	0.12	0.2988	0.4699	8.499	0.4017	22.18
17	0.1178	0.1805	0.3984	4.038	0.4811	40.7	0.1157	0.2744	0.4694	5.663	0.3964	32.38
18	0.1319	0.16	0.3966	2.45	0.482	26.7	0.1305	0.2462	0.4712	12.83	0.3867	21.32
19	0.1165	0.1442	0.3972	5.208	0.4778	28.23	0.1284	0.2299	0.4725	9.248	0.3946	15.64
20	0.1188	0.1157	0.3984	0.0005852	0.4815	3.233	0.1335	0.2167	0.467	0.001259	0.3942	7.014

Table 7.2: 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	10	10	10	10	10	10	20	20	20	20	20	20
numSamples	1000	1000	125	125	64	64	1000	1000	125	125	64	64
PMC iters	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$
1	0.1795	0.8904	0.2465	36.15	0.3519	113.3	0.2234	1.217	0.2713	27.24	0.3388	186.9
2	0.139	0.3015	0.1858	55.57	0.3223	64.9	0.1499	1.168	0.2253	19.65	0.3704	136.9
3	0.1563	0.3128	0.1736	33.8	0.3074	60.52	0.1078	0.7134	0.2138	21.78	0.2592	40.93
4	0.1294	0.2932	0.1603	13.41	0.2898	90.01	0.09533	0.7099	0.1898	20.27	0.2906	25.63
5	0.1253	0.3842	0.1585	11.48	0.2742	85.06	0.08339	0.623	0.1681	11.87	0.2612	23.16
6	0.1132	0.399	0.1545	11.4	0.2664	80.64	0.07423	0.624	0.1601	12.82	0.227	45.45
7	0.1037	0.2723	0.1525	9.954	0.253	170.8	0.05957	0.4831	0.1358	7.895	0.2275	38.69
8	0.07545	0.2984	0.1467	7.33	0.2508	46.54	0.05457	0.4188	0.1355	10.6	0.2386	37.11
9	0.0908	0.4696	0.1408	7.767	0.2412	71.77	0.04967	0.4033	0.1408	8.923	0.2587	47.7
10	0.08269	0.3232	0.1223	8.123	0.2407	59.95	0.0501	0.4691	0.1259	11.84	0.2385	44.16
11	0.07377	0.4734	0.1318	11.7	0.2334	34.56	0.04934	0.3547	0.1139	9.815	0.2096	72.25
12	0.06823	0.3687	0.1242	8.291	0.2313	26.48	0.04109	0.3591	0.1085	5.499	0.2077	53.11
13	0.06078	0.6598	0.1144	6.772	0.2239	29.77	0.0422	0.3062	0.1055	7.303	0.2045	24.51
14	0.06593	0.2872	0.1181	5.618	0.2187	28.93	0.03383	0.3279	0.09987	10.36	0.2072	14.96
15	0.04666	0.4632	0.1058	9.705	0.2119	30.32	0.03618	0.2373	0.09868	7.476	0.1884	20.57
16	0.05153	0.5783	0.1029	7.163	0.21	48.4	0.03579	0.2792	0.09587	7.341	0.1873	30.73
17	0.04271	0.5383	0.1002	4.441	0.2097	32.32	0.02893	0.3314	0.09573	9.429	0.2006	24.24
18	0.04678	0.3965	0.1057	5.579	0.2057	21.66	0.0261	0.2101	0.09648	4.017	0.1999	14.42
19	0.03458	0.4505	0.1086	5.828	0.2034	31.26	0.02508	0.2299	0.09485	4.912	0.1784	32.66
20	0.04566	0.002106	0.11	0.2994	0.2021	0.6664	0.02658	0.00356	0.08941	0.1715	0.1773	0.1784

Table 7.3: Convergence results of the PMC algorithm when using the anisotropic Phong 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	40	40	40	40	40	40
numSamples	1000	1000	125	125	64	64	1000	1000	125	125	64	64
PMC iters	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-6})$
1	0.1828	2.094	0.2565	28.7	0.3674	130.2	0.1937	1.524	0.2886	77.58	0.353	127.6
2	0.1166	1.948	0.2161	18.6	0.3486	78.5	0.1645	1.158	0.2576	86.93	0.3273	151.7
3	0.1112	1.449	0.2157	16.02	0.3146	66.5	0.1244	0.6507	0.2503	22.89	0.3047	211.4
4	0.09082	1.01	0.1832	18.66	0.3012	69.94	0.1255	0.5472	0.2251	4.055	0.2836	110.5
5	0.08589	0.4637	0.1677	14.17	0.2938	61.51	0.07503	0.629	0.197	6.878	0.262	52.27
6	0.0761	0.3838	0.1612	19.05	0.2636	57.4	0.07258	0.6713	0.1766	5.772	0.2551	50.04
7	0.06001	0.357	0.1744	25.53	0.2569	40.2	0.06998	0.6258	0.1881	7.928	0.2445	28.68
8	0.05305	0.3329	0.1499	23.11	0.2562	24.3	0.06695	0.6558	0.1794	7.173	0.242	27.21
9	0.04989	0.3237	0.1425	12.43	0.2539	26	0.05039	0.56	0.1636	6.844	0.2358	47.06
10	0.05222	0.3424	0.1401	10.27	0.2467	25.61	0.04837	0.6176	0.1578	4.977	0.2335	40.95
11	0.0436	0.2651	0.1453	11.28	0.2313	33.58	0.04664	0.4883	0.1565	6.839	0.2212	59.94
12	0.04978	0.4266	0.1385	8.441	0.236	48.4	0.04759	0.4076	0.1525	3.447	0.2198	35.92
13	0.05267	0.3163	0.1367	3.826	0.2237	21.91	0.03595	0.6366	0.1388	4.72	0.2159	32.97
14	0.03302	0.2544	0.135	5.948	0.2259	15.91	0.04676	0.4036	0.1441	4.112	0.2092	42.32
15	0.0393	0.3448	0.129	3.438	0.2114	27.52	0.03554	0.3983	0.1359	8.125	0.2052	32.46
16	0.02465	0.2298	0.1208	10.51	0.2098	47.67	0.04451	0.4037	0.1424	5.453	0.2037	51.84
17	0.03896	0.3499	0.1218	7.47	0.215	14.75	0.05564	0.2726	0.1381	8.625	0.2075	28.76
18	0.04136	0.2983	0.1185	5.528	0.2124	39.14	0.03834	0.3052	0.132	4.137	0.1993	29.61
19	0.03825	0.3004	0.1246	7.979	0.2093	34.31	0.02899	0.4205	0.1305	7.654	0.196	28.87
20	0.03095	0.01321	0.1209	0.2453	0.2089	0.3795	0.03218	0.04817	0.1231	0.1989	0.1951	0.6611

Table 7.4: Convergence results of the PMC algorithm when using the anisotropic Phong 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	10	10	10	10	10	10	20	20	20	20	20	20
numSamples	1000	1000	125	125	64	64	1000	1000	125	125	64	64
PMC iters	MAP dist	var $(\times 10^{-7})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-4})$
1	0.48	17	0.45	12	0.57	25	0.27	18	0.46	21	0.37	32
2	0.22	15	0.45	2.6	0.57	7.3	0.2	1.8	0.39	3.8	0.35	0.82
3	0.24	11	0.46	1.9	0.54	6.8	0.21	0.91	0.43	2.6	0.35	0.35
4	0.16	8.5	0.44	2.9	0.56	2.9	0.21	0.62	0.4	2	0.35	0.28
5	0.076	6.5	0.41	4	0.52	1.6	0.2	0.57	0.19	1.5	0.35	0.25
6	0.13	5.4	0.25	4.2	0.53	3.3	0.07	0.48	0.23	1.8	0.36	0.26
7	0.11	4.8	0.46	3.5	0.56	3.6	0.056	0.42	0.25	1.8	0.35	0.32
8	0.061	4.4	0.23	3.9	0.33	1.1	0.054	0.4	0.42	1.4	0.34	0.3
9	0.057	4.4	0.22	3.3	0.52	5	0.19	0.35	0.45	0.97	0.35	0.35
10	0.085	4	0.21	2.3	0.48	3.5	0.0068	0.3	0.45	1	0.37	0.31
11	0.084	3.6	0.25	3	0.4	9.4	0.036	0.3	0.43	0.97	0.39	0.4
12	0.096	3.6	0.2	2.1	0.39	48	0.038	0.27	0.42	1.1	0.36	0.31
13	0.092	3.5	0.2	2.2	0.38	24	0.13	0.24	0.43	0.71	0.37	0.34
14	0.058	3.2	0.21	1.9	0.37	5.1	0.12	0.23	0.44	0.55	0.39	0.4
15	0.096	2.8	0.24	1.9	0.37	2.2	0.047	0.21	0.43	0.46	0.35	0.47
16	0.052	2.6	0.25	1.6	0.35	2.3	0.12	0.17	0.41	0.56	0.35	0.64
17	0.039	2.3	0.23	1.8	0.38	1.8	0.072	0.15	0.43	0.6	0.3	0.2
18	0.062	2.3	0.21	1.2	0.35	3	0.11	0.13	0.41	0.72	0.32	0.35
19	0.037	2.1	0.22	1.2	0.34	2.2	0.032	0.1	0.41	0.65	0.33	0.27
20	0.026	2	0.21	1.8	0.35	0.3	0.11	0.085	0.43	0.32	0.32	0.066

Table 7.5: Convergence results of the PMC algorithm when using the isotropic Ward reflectance model. A single example run is shown over the 20 iterations of the PMC algorithm. The image size is 100x100 pixels. The table is populated with values for a single randomly chosen run of the PMC algorithm.

numLabels	30	30	30	30	30	30	40	40	40	40	40	40
numSamples	1000	1000	125	125	64	64	1000	1000	125	125	64	64
PMC iters	MAP dist	var $(\times 10^{-7})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-7})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-5})$
1	0.18	16	0.3	3.3	0.63	9.4	0.071	16	0.13	2.5	0.32	15
2	0.054	12	0.23	4.2	0.52	6.2	0.14	18	0.13	2	0.3	17
3	0.11	10	0.22	3.4	0.39	4.6	0.14	18	0.15	1.3	0.29	10
4	0.15	9.5	0.11	1.3	0.4	6.6	0.14	10	0.032	2.4	0.17	3.9
5	0.063	8	0.095	0.88	0.51	5.7	0.096	7.2	0.051	1.5	0.14	8.7
6	0.061	7.6	0.091	0.86	0.33	3.9	0.1	5.1	0.013	1.3	0.12	7.9
7	0.034	6.9	0.079	1.3	0.21	3.9	0.1	3.9	0.041	0.88	0.12	5.3
8	0.05	3.8	0.097	0.93	0.21	2.7	0.027	3.8	0.0064	0.71	0.11	8.1
9	0.043	5.3	0.07	0.35	0.17	10	0.087	3.3	0.027	1.6	0.11	4.7
10	0.041	4.1	0.093	0.62	0.16	9.1	0.066	3.7	0.017	0.91	0.1	2.4
11	0.03	4.8	0.072	0.51	0.17	6.6	0.067	2.6	0.031	0.83	0.095	3.7
12	0.044	3.8	0.059	0.48	0.15	4	0.063	3.1	0.022	1.2	0.11	3.7
13	0.044	3	0.058	1	0.16	4.7	0.052	2.5	0.0027	0.69	0.098	6
14	0.039	2.4	0.048	0.76	0.16	4.1	0.022	3.2	0.011	0.82	0.097	4
15	0.043	2.5	0.036	0.75	0.15	3	0.022	3.8	0.0062	0.84	0.089	3.7
16	0.045	2.8	0.028	0.67	0.15	4.4	0.073	2.1	0.0061	0.7	0.075	2.3
17	0.059	2	0.029	0.59	0.14	3.2	0.048	2.8	0.026	0.96	0.075	3.4
18	0.036	2.7	0.03	0.88	0.13	5.2	0.1	3.2	0.0072	0.56	0.077	6.1
19	0.041	3.2	0.048	1	0.13	2.7	0.015	3.1	0.014	0.52	0.073	5.4
20	0.038	0.21	0.047	0.014	0.14	0.052	0.015	0.22	0.03	0.029	0.08	0.033

Table 7.6: Convergence results of the PMC algorithm when using the isotropic Ward reflectance model. A single example run is shown over the 20 iterations of the PMC algorithm. The image size is 100x100 pixels. The table is populated with values for a single randomly chosen run of the PMC algorithm.

numLabels	10	10	10	10	10	10	20	20	20	20	20	20
numSamples	1000	1000	125	125	64	64	1000	1000	125	125	64	64
PMC iters	MAP dist	var $(\times 10^{-7})$	MAP dist	var $(\times 10^{-6})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-7})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-5})$
1	0.19	13	0.15	24	0.28	9.2	0.38	9.8	0.12	2.8	0.14	3.2
2	0.12	9.1	0.14	20	0.28	9.4	0.13	5.9	0.14	2	0.13	2.5
3	0.24	6.6	0.13	17	0.28	7	0.15	4.9	0.14	1.7	0.13	1.8
4	0.26	8.6	0.11	16	0.28	6.1	0.021	7.1	0.1	1.3	0.12	1.8
5	0.17	7.8	0.11	15	0.27	2	0.04	5.8	0.085	1.7	0.11	2.05
6	0.052	8.4	0.084	8.8	0.27	2.7	0.098	4.6	0.079	1.05	0.079	1.2
7	0.14	10	0.099	5	0.27	1.6	0.026	3.4	0.07	0.78	0.7	1.1
8	0.022	6.1	0.089	8.8	0.22	4.5	0.084	2.4	0.06	1.4	0.06	1.4
9	0.054	5.4	0.076	8.8	0.24	4.5	0.024	3.4	0.048	0.56	0.08	0.74
10	0.013	3.9	0.072	12	0.23	3.3	0.093	3	0.043	0.67	0.05	0.96
11	0.1	3.6	0.07	8.6	0.24	2.7	0.085	3.5	0.029	0.4	0.06	0.64
12	0.056	3.1	0.056	6.4	0.24	1.5	0.027	3.6	0.036	0.71	0.036	0.72
13	0.031	2.6	0.06	5.4	0.24	4.1	0.037	3.3	0.027	0.92	0.027	0.85
14	0.097	3	0.051	4.7	0.24	2.7	0.011	2.5	0.021	0.72	0.019	0.88
15	0.025	3.1	0.057	5.4	0.21	6.1	0.017	3.1	0.023	0.74	0.016	0.63
16	0.03	2.9	0.043	7.8	0.19	6.5	0.038	3.2	0.026	0.9	0.026	1.1
17	0.031	3.6	0.044	11	0.19	3.4	0.039	4.1	0.018	0.6	0.021	0.71
18	0.089	2.5	0.055	6.3	0.19	5.4	0.026	3.5	0.014	0.81	0.012	0.53
19	0.035	2.6	0.043	4.5	0.19	4.4	0.0056	5.4	0.014	0.72	0.011	0.46
20	0.13	0.063	0.039	0.12	0.2	0.027	0.013	0.018	0.017	0.022	0.019	0.022

Table 7.7: Convergence results of the PMC algorithm when using the anisotropic Phong reflectance model. A single example run is shown over the 20 iterations of the PMC algorithm. The image size is 100x100 pixels. The table is populated with values for a single randomly chosen run of the PMC algorithm.

numLabels	30	30	30	30	30	30	40	40	40	40	40	40
numSamples	1000	1000	125	125	64	64	1000	1000	125	125	64	64
PMC iters	MAP dist	var $(\times 10^{-7})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-7})$	MAP dist	var $(\times 10^{-5})$	MAP dist	var $(\times 10^{-5})$
1	0.18	16	0.3	3.3	0.63	9.4	0.071	16	0.13	2.5	0.32	15
2	0.054	12	0.23	4.2	0.52	6.2	0.14	18	0.13	2	0.3	17
3	0.11	10	0.22	3.4	0.39	4.6	0.14	18	0.15	1.3	0.29	10
4	0.15	9.5	0.11	1.3	0.4	6.6	0.14	10	0.032	2.4	0.17	3.9
5	0.063	8	0.095	0.88	0.51	5.7	0.096	7.2	0.051	1.5	0.14	8.7
6	0.061	7.6	0.091	0.86	0.33	3.9	0.1	5.1	0.013	1.3	0.12	7.9
7	0.034	6.9	0.079	1.3	0.21	3.9	0.1	3.9	0.041	0.88	0.12	5.3
8	0.05	3.8	0.097	0.93	0.21	2.7	0.027	3.8	0.0064	0.71	0.11	8.1
9	0.043	5.3	0.07	0.35	0.17	10	0.087	3.3	0.027	1.6	0.11	4.7
10	0.041	4.1	0.093	0.62	0.16	9.1	0.066	3.7	0.017	0.91	0.1	2.4
11	0.03	4.8	0.072	0.51	0.17	6.6	0.067	2.6	0.031	0.83	0.095	3.7
12	0.044	3.8	0.059	0.48	0.15	4	0.063	3.1	0.022	1.2	0.11	3.7
13	0.044	3	0.058	1	0.16	4.7	0.052	2.5	0.0027	0.69	0.098	6
14	0.039	2.4	0.048	0.76	0.16	4.1	0.022	3.2	0.011	0.82	0.097	4
15	0.043	2.5	0.036	0.75	0.15	3	0.022	3.8	0.0062	0.84	0.089	3.7
16	0.045	2.8	0.028	0.67	0.15	4.4	0.073	2.1	0.0061	0.7	0.075	2.3
17	0.059	2	0.029	0.59	0.14	3.2	0.048	2.8	0.026	0.96	0.075	3.4
18	0.036	2.7	0.03	0.88	0.13	5.2	0.1	3.2	0.0072	0.56	0.077	6.1
19	0.041	3.2	0.048	1	0.13	2.7	0.015	3.1	0.014	0.52	0.073	5.4
20	0.038	0.21	0.047	0.014	0.14	0.052	0.015	0.22	0.03	0.029	0.08	0.033

Table 7.8: Convergence results of the PMC algorithm when using the anisotropic Phong reflectance model. A single example run is shown over the 20 iterations of the PMC algorithm. The image size is 100x100 pixels. The table is populated with values for a single randomly chosen run of the PMC algorithm.

Tables 7.6, 7.7 and 7.8 also seem to describe good runs, with the final "MAP dist" (after 20 iterations) errors always below 0.14.

7.4.2 Conclusion

The conclusion of this chapter is that, according to an analysis of convergence results for the dynamically weighted MCMC, PMC, and Multiple-seed LM algorithms, only the PMC sampler can provide a unique combination of convergence to a good MAP sample value, while also returning a sample set which represents the posterior distribution. It also does so with relatively few samples and within relatively few iterations, compared to the dynamically weighted MCMC sampling algorithm.

It is seen in Tables 7.1 and 7.2 that the convergence statistics when using either 125 or 64 samples are much worse than when using 1000, with average errors in the region of 0.4 units from the true BRDF parameters. However, in general, and when using smaller numbers of samples per sample set in the PMC method, the sample set variances tend to decrease steadily over the 20 PMC iterations. This indicates that the BRDF probability functions have bad local maxima, into which the samples are collecting (this was verified in our experiments).

This is consistent with our finding that the PMC algorithm run with larger numbers of samples has both a reduced average MAP distance, and a lower sample variance: the larger number of samples has allowed the algorithm to find the correct local maxima more reliably. This clearly indicates that a higher number of samples (1000 or greater) is required to achieve decent convergence results when using the PMC method. It was also discovered that there were regions in the parameter space close to the limits in the unit cube which gave poorer convergence results. When true BRDF parameter sets were sampled from these regions, all three BRDF estimation methods (Multiple-seed LM, dynamic MCMC and PMC) suffered, giving unreliable convergence to the true BRDF parameter set used to render the synthetic image $\mathbf{I}^1$.

One can compare the PMC sampler results when using 125 samples with the dynamic MCMC sampling results after 1300 sampling steps (the computational loads are approximately equivalent, since the PMC method uses the pseudolikelihood over all sites on the lattice (Eqn. 7.7) only at the end of 20 iterations, whereas the dynamic MCMC samplers use this pseudolikelihood over all sites for every sample). Similarly, one can compare the PMC sampler when using 64 samples with the dynamic MCMC sampler, after 700 samples, in Figs. 7.7 to 7.14. When this comparison is done, it is seen that the dynamically weighted MCMC sampler sometimes returns an MAP sample with a smaller distance to the true BRDF parameter set. This suggests that if a very small amount of time is available, and only a few samples can be processed, the dynamically weighted MCMC sampler is a reasonable alternative choice. However, we could easily adjust the PMC sampler to store the best MAP sample through all the iterations, and return that sample as the MAP sample. In addition the PMC sampler would still have the advantage of

returning a sample set small enough to be used as features in classification (less than 1000 samples for efficient calculation, although this depends on the method used for inter-feature distance calculations).

Although we have no way to verify the correct convergence of the model's MRF parameters to a real material's BRDF parameters given the uncertainty in its reconstruction, the fact that there are local maxima in the probability distribution in the BRDF of the real material, and that these local maxima are distributed in a particular way, is represented by the PMC sample set. The more samples are used to represent this distribution, the more accurately it will be represented by the sample set. Therefore, even though the PMC samples after 20 iterations may have missed an optimal solution for the BRDF parameterisation for a given real surface, the samples do still represent the parametric MRF/BRDF probability distribution, and can thus be used as features nonetheless.

Chapter 8

Testing PMC and LM algorithm on real data: Classification results

This chapter describes the classification experiments done on real data captured using a stereo projector and digital light projector, using the parametric BRDF posterior distributions extracted from the stereo data as features. The goal is to test whether samples representing parametric BRDFs, calculated using the PMC algorithm developed in this thesis, can be used to classify materials, and to discover which sample set classification method gives the best classification results. This is done by calculating probabilistic reconstructions on a number of real world objects, namely different material surfaces. The cloth materials are draped over a hemispherical object, to allow the extraction of BRDF data over a wide range of incident and reflected angles. The plastic materials are either naturally curved or are flexible and are bent into a curving shape. The froth surface appears to have sufficient curvature on the bubbles to allow parametric BRDF estimation.

The testing on real data is done firstly to show that the resulting posterior distributions developed are reasonably congruent with the visible surface characteristics, but more importantly to show that the sample sets representing the parametric BRDFs can be used as features for material surface classification. The correctness of these distributions is not verified using any other method, as there is no other method available which calculates the posterior distributions on surface parametric BRDFs given probabilistic reconstruction information. Since it is actually the MRF parameters which are estimated, standard parametric BRDF parameter estimation methods such as [122] could not be used to verify this method, since the cost functions being minimized are themselves different (our cost functions involve pseudolikelihoods and MRF potential functions, rather than simple least-squares fitting between predicted and measured pixel intensity values, as is usually done in BRDF parameter estimation algorithms).
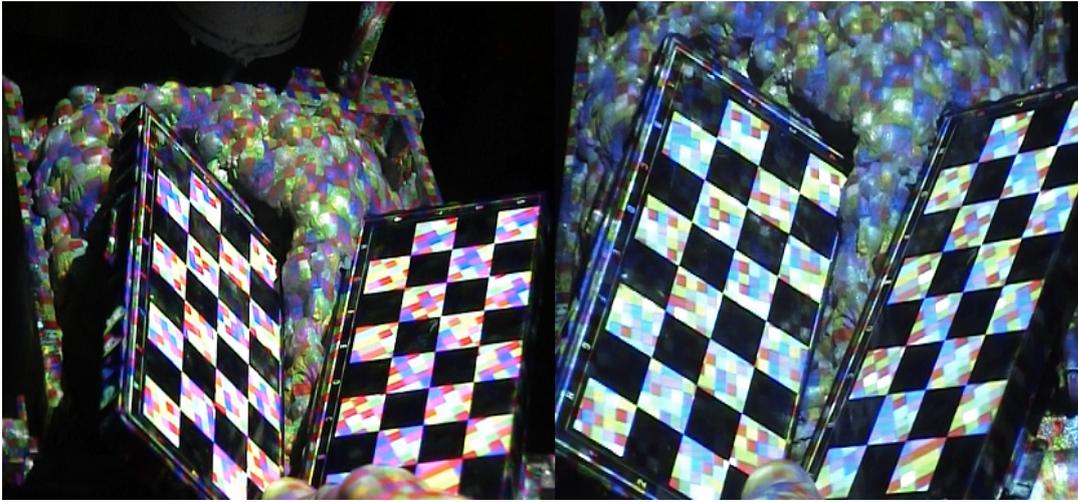
Figure 8.1: The checkerboard used to calibrate the cameras is seen held over the froth surface, from each camera in the stereo rig.

8.1 Real data

Images of real froth surfaces were acquired using a stereo pair of video cameras which recorded footage of the froth of a flotation cell. For the material surfaces, which are static, we used two high resolution still cameras to capture the stereo images. A digital light projector (DLP) was used to project a checkered pattern of coloured squares onto the material surface, to allow identification of the pixel correspondences with better accuracy. The DLP was positioned about 1.5m from the surface. The projector was configured to alternate between shining a coloured checkerboard pattern for half a second, and then shining white light onto the froth surface. On the assumption that the froth does not change shape significantly faster than the projector transitions from shining the coloured pattern to shining the white light, the intensity image of the surface under the white light corresponds to the same surface reconstructed using the coloured checkerboard pattern of projected light. The reflectance intensity image for the reconstructed surface geometries is obtained as an image from one of the video cameras at a moment when the projector is shining a white screen (one or two frames after the checkerboard pattern is projected).

The calibration checkerboard is shown in Fig. 8.1, in the left and right camera views, with the cameras in similar configurations as when taking footage of the froth surfaces. Fig. 8.2 shows the left and right views under white illumination from the digital light projector (DLP). Fig. 8.3 shows images taken an instant before the images of Fig. 8.2, of the same froth surface under a colourful checkerboard lighting pattern projected by the DLP.

The purpose of gathering data in this way is to have an accurate probabilistic stereo reconstruction.
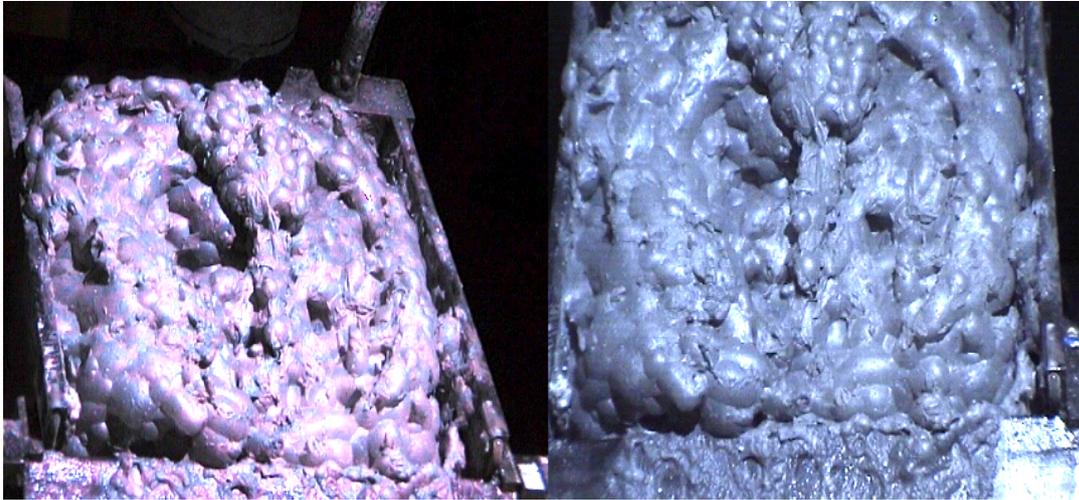
Figure 8.2: White light is projected by the projector onto the froth surface. This view is shown from each camera in the stereo rig.
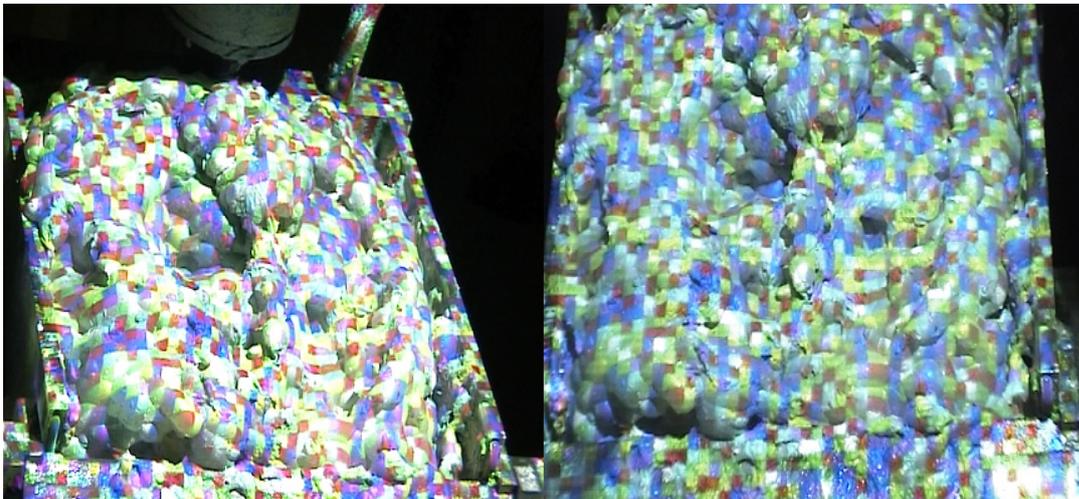
Figure 8.3: A checkerboard pattern is projected onto the froth surface. This pattern gives better stereo reconstruction, as it allows greater certainty on pixel matching.

The first set of images which include a physical checkerboard are taken in order to calibrate the stereo cameras. Images where the projector shines a coloured light pattern onto the material surface are used to create an accurate probabilistic reconstruction. The images captured when the projector is shining white light are used to estimate the parametric BRDF, given the probabilistic surface reconstruction.

As with the unstable froth surfaces, the stereo data for the material surfaces was captured with structured lighting shone onto each material surface, followed by a screen of white light from the same projector source, which allows calculation of the parameterized BRDF posterior distributions. Images of the materials analysed are shown in Figs. 8.4 to 8.15. These materials exhibit a wide range of reflectance characteristics, such as glossiness, shininess (laminated paper surface, leather), and fuzziness (cloth materials). After the stereo data is captured, probabilistic reconstruction is done using the dense stereo algorithms described in Chapter 6.

Developed (evolved) sample sets which represent the posterior pseudolikelihood function on the BRDF parameterisation for a surface are calculated for each material, including the froth surface. This is done using the PMC algorithm described in section 6.6.2, with the pseudolikelihoods and importance functions defined using three subsets of corner vertex nodes on the lattice: S^0, S^1, S^2, such that $S^n = \{n, n+k, n+2k, \dots\}$ with $k = 3$. The pseudolikelihood function over any subregion is therefore

$$
\begin{aligned}
ePL_3^n(\mathbf{X}|\theta) &= \exp(PL_3^n(\mathbf{X}|\theta)) \\
&= \prod_{i \in S^n} \frac{\exp(-U_i^A(x_i, x_{\mathcal{N}_i}, \theta) - U_i^D(x_i, \mathbf{I}^R))}{\sum_{s \in \mathcal{L}} \exp(-U_i^A(s, x_{\mathcal{N}_i}, \theta) - U_i^D(s, \mathbf{I}^R))}.
\end{aligned}
\tag{8.1}
$$

Since 20 PMC iterations are done, 20 importance functions G^n are required. The three functions are reused, so

$$
G^n(\theta|\mathbf{X}) = s(ePL_3^j(\mathbf{X}|\theta)),
\tag{8.2}
$$

where $j = \mod(n, 3)$.

Then a number of similarity measures used to compare probability density functions represented by sample sets drawn from normal distributions are used for the comparison of these probability density functions. Some measures which are not usually used for comparison of probability density functions are also used to compare the sample sets. Classification statistics are given for each such similarity measure.

8.2   Comparing sample sets

For the classification of surfaces through the comparison of their BRDFs, 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.
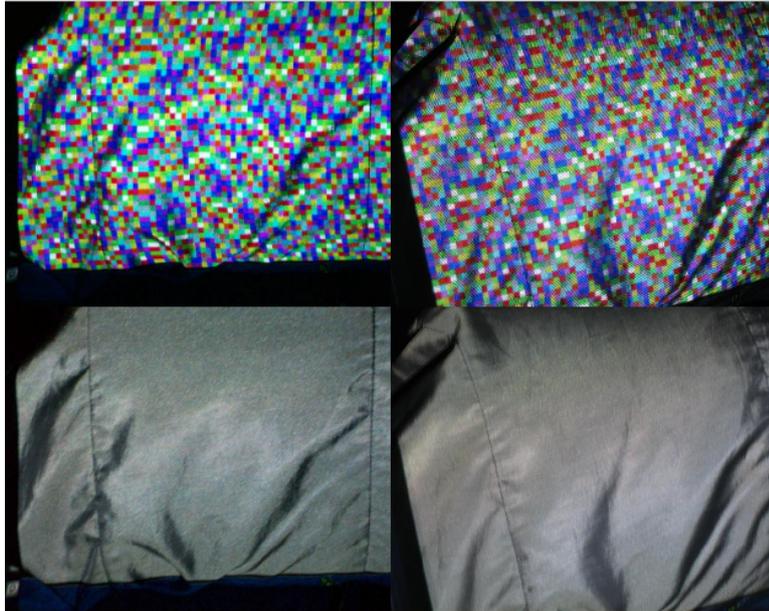
Figure 8.4: Material surface #1 (shiny fabric).
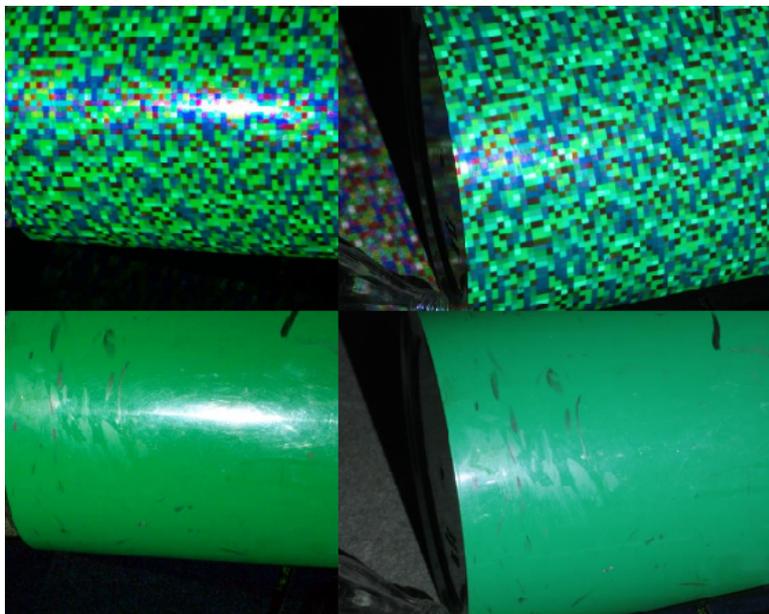
Figure 8.5: Material surface #2 (shiny plastic).
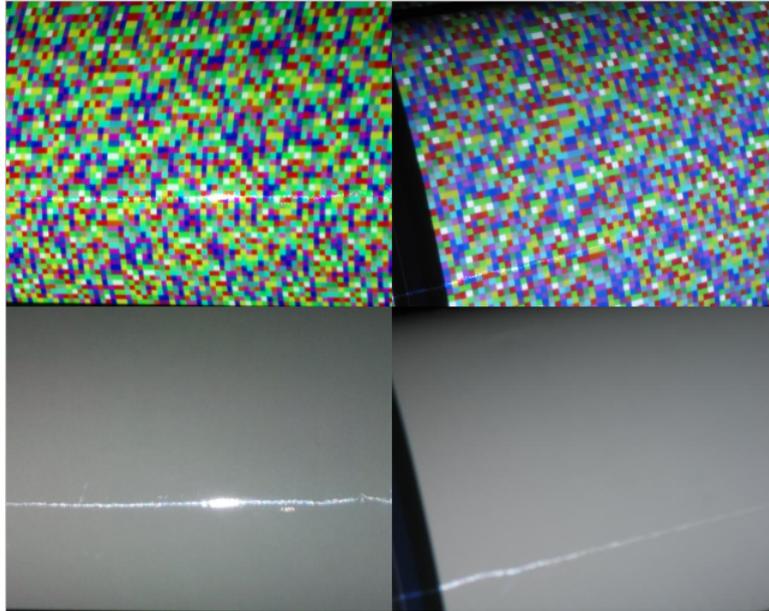
Figure 8.6: Material surface #3 (laminated paper).
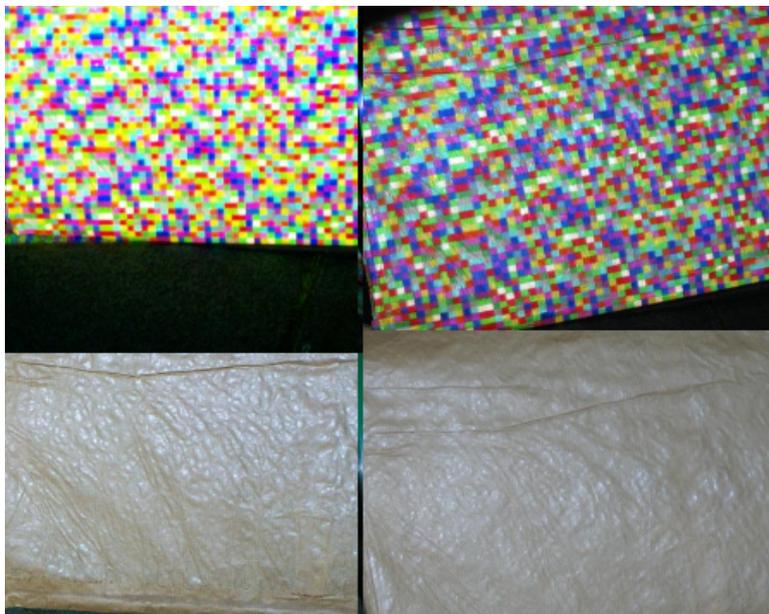
Figure 8.7: Material surface #4 (glossy brown paper).
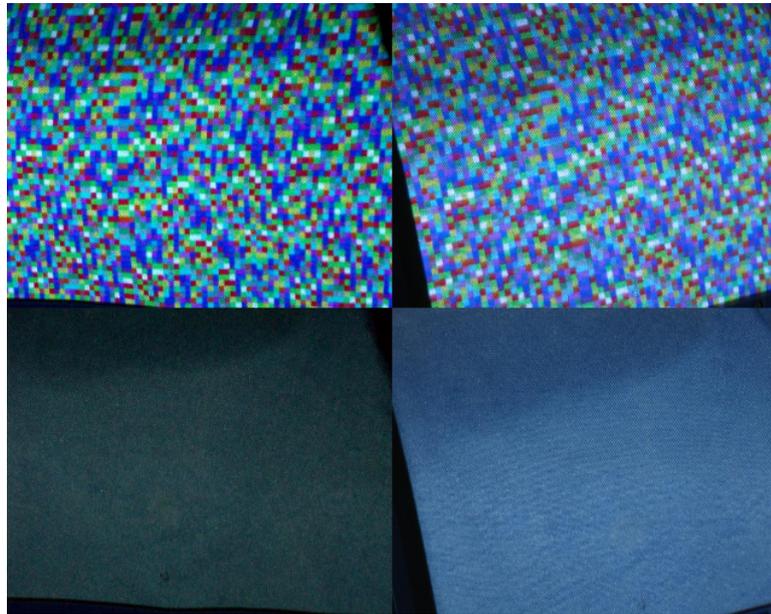
Figure 8.8: Material surface #5 (canvas).
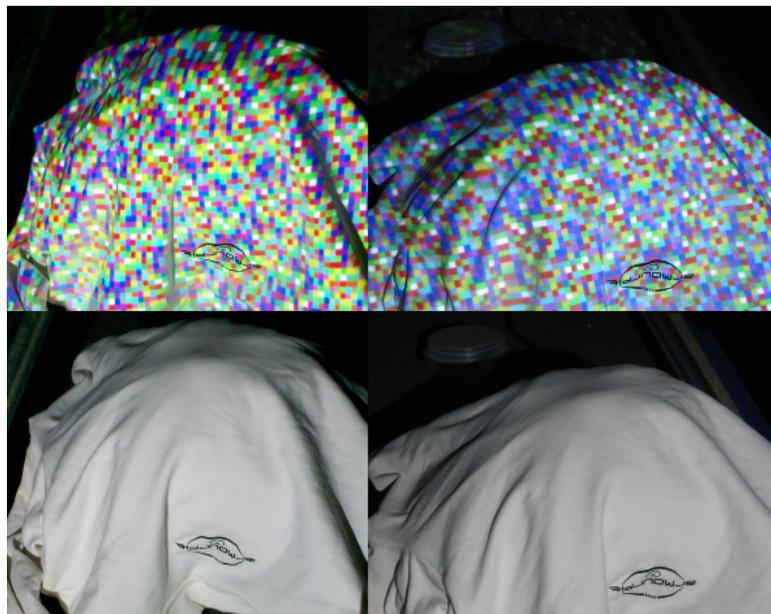
Figure 8.9: Material surface #6 (smooth cloth).
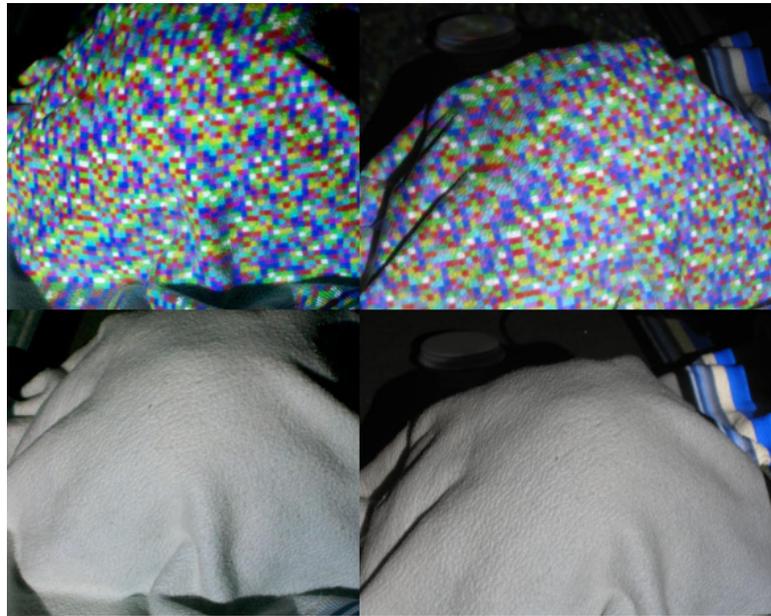
Figure 8.10: Material surface #7 (fuzzy cloth).
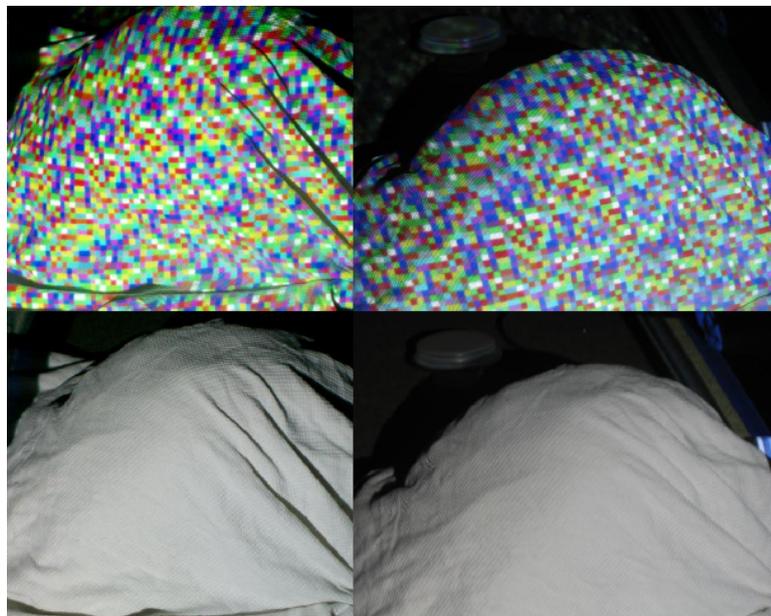
Figure 8.11: Material surface #8 (t-shirt cloth).
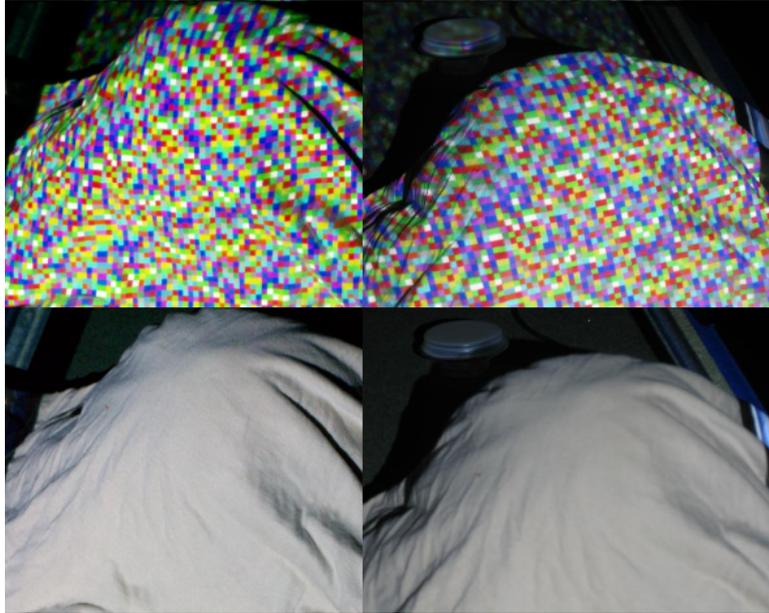
Figure 8.12: Material surface #9 (t-shirt cloth).
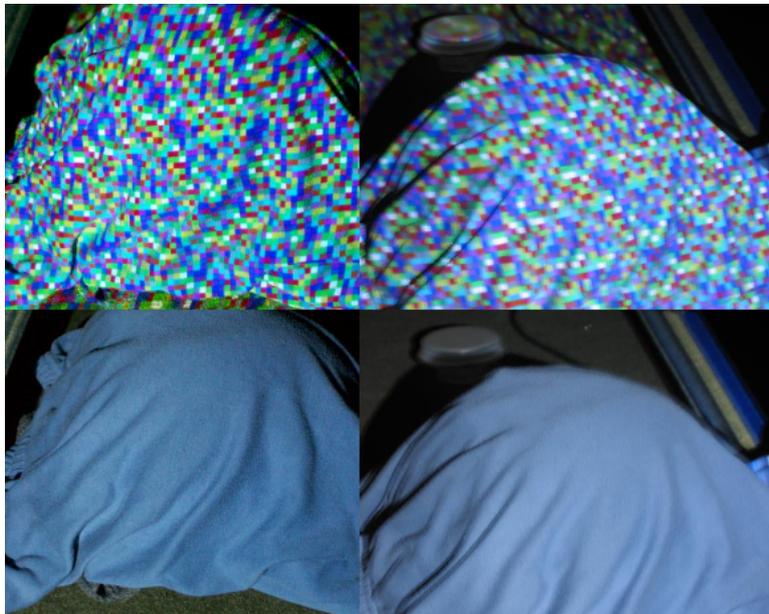
Figure 8.13: Material surface #10 (smooth blue cloth).
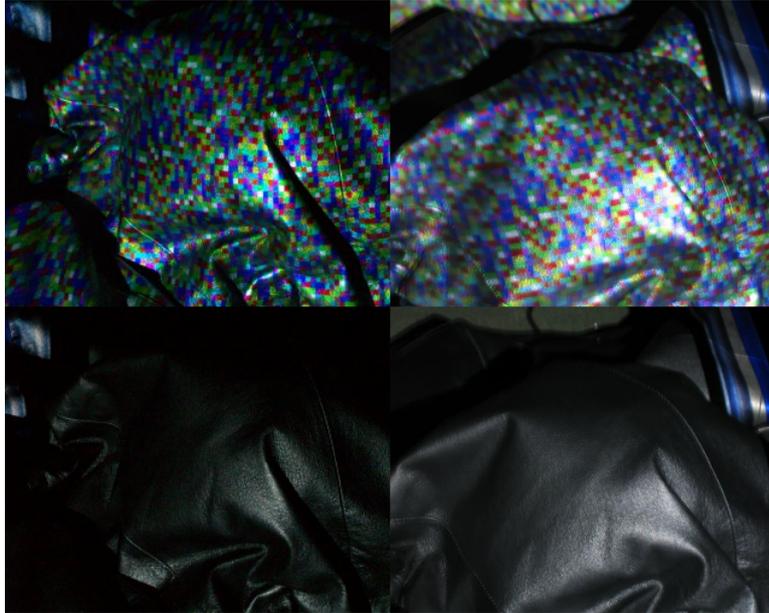
Figure 8.14: Material surface #11 (black leather).
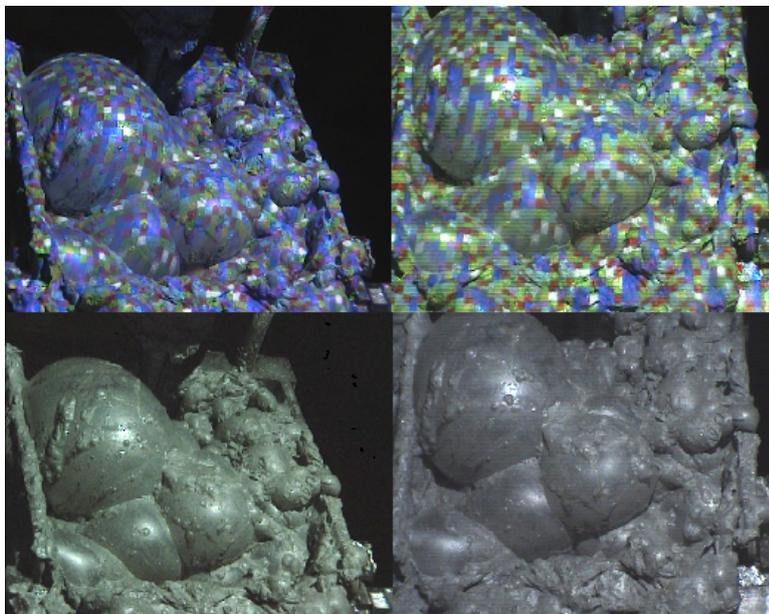
Figure 8.15: Surface #12 (Froth).

8.2.1 Comparing sample sets assuming Gaussian noise: Average Gaussian Mixture Model Probability

This section introduces our "Average Gaussian Mixture Model probability" (AGMMP) measure. This measure is reminiscent of sample classification using Parzen windows [87] but is closer to sample classification using Gaussian Mixture Modelling [31]. The GMM framework is modified to compare two weighted samples sets, since the Parzen windows and GMMs can only classify an unweighted sample set given a weighted sample set. Using the AGMMP measure, it is possible to compare two probability density functions which are represented by sample sets with associated weights in a simple manner. This is done by making assumptions about the samples' dependencies, and about the type of noise about each sample. It is assumed that samples across sets have generated each other as if sampling from Gaussian mixture models which have a single mode on each point in the sample set. The multivariate normal distribution equation, which describes the joint probability of an N dimensional sample x, given a Gaussian distribution with covariance Σ centered at u, is

$$\mathcal{N}_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-0.5(x-\mu)^T\Sigma^{-1}(x-\mu)\right), \tag{8.3}$$

where $\mathcal{N}$ is the multivariate Gaussian distribution, d is the size of the random vector (in this case, the size of the parameter vector for the BRDF model being tested), and $|\Sigma|$ is the determinant of covariance matrix Σ.

Assume that one is given two weighted sample sets: $\{\vec{x}, \overrightarrow{\pi}_x\}$, where $\{\pi_x^i\}$ is the normalized probability weight on sample x^i, and $\{\vec{y}, \overrightarrow{\pi}_y\}$, where π_y^j is the normalized probability weight for sample y^j. Assume also that the conditional probability of any two samples is governed by a multivariate Gaussian distribution, so that the probability of a sample x^i in the first set arising from y^j in the second set is

$$p(x^i|y^j) = \mathcal{N}_{y^j,\Sigma}(x^i), \tag{8.4}$$

where the covariance of any Gaussian mixture center $\Sigma = \sigma\mathbb{I}_{d\times d}$, where $\mathbb{I}_{d\times d}$ is a $d \times d$ identity matrix.

Suppose $p_x(w_i)$ is the probability weight of the ith sample in the first sample set, so $p_x(w_i) = \pi_i^x$. The probability of any x^i being generated by the set of samples $\vec{y}$ is

$$p(x^i|\overrightarrow{y}) = \frac{1}{M}\sum_{j=1}^{M} p(x^i|w_j, y^j)p_y(w_j), \tag{8.5}$$

where $p_y(w_j)$ is the probability weight of the jth sample in the second sample set, so $p_y(w_j) = \pi_j^y$. M is the number of samples in $\overrightarrow{y}$. If one assumes independence between the samples x^i, then

$$p(\overrightarrow{x}|\{\overrightarrow{y}, \overrightarrow{\pi}^y\}, \sigma) = \prod_{i=1}^{N} p(x^i, \overrightarrow{y}), \tag{8.6}$$

where N is the number of samples in $\vec{x}$. This gives the usual equation for the probability of data given a Gaussian mixture model with prior probabilities $p_y(w_j)$ on each center j:

$$p(\vec{x}|\{\vec{y}, \vec{\pi}^y\}, \sigma) = \frac{1}{M} \prod_{i=1}^{N} \sum_{j=1}^{M} \mathcal{N}_{y^j, \Sigma}(x^i)\pi_j^y. \tag{8.7}$$

This probability measure has taken into account the sample weights on the second set, but not on the first set. To incorporate this information into our measure, we create a symmetrical measure:

$$\mathrm{AGMMP}(\{\vec{x}, \vec{\pi}^x\}, \{\vec{y}, \vec{\pi}^y\}, \sigma) = \frac{p(\vec{x}|\{\vec{y}, \vec{\pi}^y\}, \sigma) + p(\vec{y}|\{\vec{x}, \vec{\pi}^x\}, \sigma)}{2}. \tag{8.8}$$

8.2.2 Comparing unevolved sample sets using a correlation measure and the inner product

Assume the samples are in their initial positions at evenly spaced positions on a grid in the parameter space, and that they have been given normalized weights corresponding to the observation density ePL_3 as described in Eqn. 6.62. Then we may intuitively apply a correlation measure to the sample weights, with the summations and pairwise products on the sample weights done in the same order as that in which the particles are indexed. The equation for the correlation coefficient between two vectors is

$$r = \frac{\sum_m (\pi_1^m - \mathrm{mean}(\vec{\pi}_1))(\pi_2^m - \mathrm{mean}(\vec{\pi}_2))}{\sqrt{(\sum_m (\pi_1^m - \mathrm{mean}(\vec{\pi}_1))^2)(\sum_m (\pi_2^m - \mathrm{mean}(\vec{\pi}_2))^2)}}, \tag{8.9}$$

where π_1^m indicates the weight of the mth sample in the first sample set.

In a similar manner, we may use the inner product (referred to as "IP" in the tables which follow) to compare two sample sets, if the samples are in the same locations in parameter space:

$$r = \sum_m (\pi_1^m \pi_2^m). \tag{8.10}$$

In both cases, the sets of sample weights $\vec{\pi}_1^m$ and $\vec{\pi}_2^m$ are assumed to be normalized in the sense that the probabilities sum to 1.

8.2.3 Comparing sample sets using the Earth Mover's Distance

Another measure we used with some success for defining distances between sample sets is the Earth-Mover's distance. An efficient optimization method for solving this class of transport problem was developed in [100], in which each feature or data point (called a "signature") is given a mass, and a function is defined which returns the amount of work required to transport mass from any signature in the source distribution to any signature in the target distribution. The algorithm finds the optimal way

to move all mass from the first signature set to the second signature set. The amount of work required to do this is called the Earth Mover's Distance (EMD). A low EMD between two signature sets thus indicates that they are similar.

To apply this technique to the task of comparing probability distributions which are represented by weighted sample sets, each sample is treated as a signature. The signature weight is set equal to the sample weight (sample probability). In this way, particle sets representing the same distribution tend to have low EMDs, and particle sets representing different distributions would have high EMDs. More information on the EMD may be found in Appendix B.

Although we cannot support the use of the Earth Mover's Distance (EMD), which is a linear transport optimization algorithm, as a probabilistic distance measure, this method was found to be effective as a measure of similarity for the comparison of weighted sample sets. Again the task is to compare the probability density functions of parametric BRDFs represented by weighted sample sets. To compare such sample sets, a measure is required that returns a high value to indicate similarity between sample sets. Low values thus indicate large differences between sample sets which represent different underlying posterior distributions on the BRDF of the measured surfaces.

The EMD measure gives the opposite of this (a small distance corresponds to a similar pair of sample sets), but it is easy to correct it by inverting or negating the output and adding a constant value. There are in fact not many alternatives available for comparing non-Gaussian distributions represented by weighted particles: methods such as correlation and mutual information rely on the pdfs being defined at all points on the domain (i.e. not represented by a collection of isolated samples, but rather known at intervals throughout the domain of the distribution). The Chi-squared method partitions the pdf samples first, but it is not obvious how the sample space should be partitioned (binned).

8.2.4 Comparing sample sets: analytical expressions for BRDF posterior distributions which are assumed to be normally distributed

There are many theoretical probabilistic distances which can be used to compare two pdfs. The mathematical expressions for the Kullback-Leibler divergence and symmetric Kullback-Leibler divergence [113], and the Chernoff [21], Bhattacharyya [10], Matusita [78], Patrick-Fisher [88], Mahalanobis [77], Lissack-Fu [67] and Kolmogorov [1] distances, are shown in [141], and reproduced in Table 8.1.

In [141], a method is developed which can calculate the similarity between non-Gaussian sample sets using Chernoff, Bhattacharyya, Matusita, Lissack-Fu, and other distances, by transforming the sample sets into Reproducing Kernel Hilbert Space. This method does not cater for probabilistically weighted sample sets in its current form: it assumes the samples are independently drawn from the original "ensembles", or probability densities. This method of [141] allows for good classification and discrim-

KL-divergence	$J_R(p_1 \| p_2) = \int_X p_1(X) \log\{\frac{p_1(X)}{p_2(X)}\} dX$
Symm. KL divergence	$J_D(p_1, p_2) = \int_X [p_1(X) - p_2(X)] \log\{\frac{p_1(X)}{p_2(X)}\} dX$
Chernoff	$J_C(p_1, p_2) = -\log\{\int_X p_1^{\alpha_2}(X) p_2^{\alpha_1}(X)\} dX$
Bhattacharyya	$J_B(p_1, p_2) = -\log\{\int_X [p_1(X) p_2(X)]^{1/2} dX\}$
Matusita	$J^t(p_1, p_2) = \{\int_X [\sqrt{p_1(X)} - \sqrt{p_2(X)}]^{1/2} dX\}^{1/2}$
Patrick-Fisher	$J_P(p_1, p_2) = \{\int_x [p_1(X)\pi_1 - p_2(X)\pi_2]^2 dX\}^{1/2}$
Lissack-Fu	$J_L(p_1, p_2) = \int_X |p_1(X)\pi_1 - p_2(X)\pi_2|^{\alpha_1} [p_1(X)\pi_1 + p_2(X)\pi_2]^{\alpha_2} dX$
Kolmogorov	$J_K(p_1, p_2) = \int_X |p_1(X)\pi_1 - p_2(X)\pi_2| dX$

Table 8.1: A list of distances between probability density functions, taken from [141], where $0 \le \alpha_1, \alpha_2 \le 1$ and $\alpha_1 + \alpha_2 = 1$. π_1 and π_2 are the prior probabilities on the distributions.

Chernoff	$J_C(p_1, p_2) = \frac{1}{2}\alpha_1\alpha_2(\mu_1 - \mu_2)^T [\alpha_1\Sigma_1 + \alpha_2\Sigma_2]^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\log\frac{|\alpha_1\Sigma_1 + \alpha_2\Sigma_2|}{|\Sigma_1|^{\alpha_1}|\Sigma_2|^{\alpha_2}}$
Bhattacharyya	$J_B(p_1, p_2) = \frac{1}{8}(\mu_1 - \mu_2)^T [\frac{1}{2}(\Sigma_1 + \Sigma_2)]^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\log\frac{\frac{1}{2}|\Sigma_1 + \Sigma_2|}{|\Sigma_1|^{1/2}|\Sigma_2|^{1/2}}$
KL divergence	$J_R(p_1 \| p_2) = \frac{1}{2}(\mu_1 - \mu_2)^T \Sigma_2^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\log\frac{\Sigma_2}{\Sigma_1} + \frac{1}{2}\text{tr}[\Sigma_1\Sigma_2^{-1} - I_d]$
Symm. KL divergence	$J_D(p_1, p_2) = \frac{1}{2}(\mu_1 - \mu_2)^T (\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2) + \frac{1}{2}\text{tr}[\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 - 2I_d]$
Patrick-Fisher	$J_P(p_1, p_2) = [(2\pi)^d|2\Sigma_1|]^{-1/2} + [(2\pi)^d|2\Sigma_2|]^{-1/2}$ $-2[(2\pi)^d]|\Sigma_1 + \Sigma_2|]^{-1/2}\exp\{-\frac{1}{2}(\mu_1 - \mu_2)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2)\}$
Mahalanobis	$J_M(p_1, p_2) = (\mu_1 - \mu_2)^T \Sigma (\mu_1 - \mu_2)$

Table 8.2: Some analytical expressions for probabilistic distances between Gaussian probability density functions, where $0 \le \alpha_1, \alpha_2 \le 1$ and $\alpha_1 + \alpha_2 = 1$. $|\Sigma|$ indicates the determinant of Σ. I_d is the identity matrix.

ination of sample sets which are drawn from highly non-Gaussian distributions, but is more complex than the simple analytical expressions for the same distances (Chernoff, Bhattacharyya). Since the posterior distributions on the parametric BRDFs of the real material surfaces appear to be generally unimodal, we use the analytical expressions shown in Table 8.2, making the simplifying assumption that the evolved weighted sample sets are normally distributed.

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 8.2. 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. 8.2.

In the similarity matrix shown in Tables 8.3 to 8.5, and in the similarity matrices shown in Appendix E, results are tabulated for the similarities between weighted sample sets. Some of the matrices contain results from comparisons done on the original, evenly-spaced weighted sample sets, distributed

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	993.7	990.5	992	971.7	975	985.9	964.6	989.1	988.8	991.8	965.7	985.4	985.1	979	978.5	974.5
2		994.7	990.4	968	972.8	987.3	962.9	988.6	984.5	987.9	964.4	983.5	986.3	979.3	979.3	973.8
3			993.1	961.5	979.7	986.7	971.2	986.4	984.5	990.1	956.1	987.6	980.6	973	971.1	968.5
4				999.4	935.5	968.1	912.3	996.2	995.2	990.7	999.4	961.7	997.1	997.7	997.7	997.8
5					993.2	974.7	991	982	953.1	975.7	882.9	989.6	942.6	930.2	924.8	915.6
6						994.2	970.5	983.6	974.9	983.9	932.3	981.6	977	969.5	966.1	960
7							992.6	975.3	934	964.6	847.3	983.3	925.3	906.9	902.1	887.9
8								994.5	990.4	988.9	985.8	971	993	991	990.8	989.4
9									996.3	994.6	990.6	976.7	991.8	989.4	989.3	987.8
10										994.8	975.4	984.5	985.7	982.2	982.3	977.8
11											999.9	959.7	997.3	998.4	998.3	998.5
12												993.6	962.4	954.3	953.2	943.5
13													997.1	996.1	996.4	995.3
14														998.7	998.4	998.5
15															998.7	998.4
16																999.4
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																

Table 8.3: Columns 1 to 16 of similarity matrix for parametric BRDF surfaces of real materials, using the AGMMP measure with $\sigma = 0.005$, with evolved sample sets (20 PMC iterations) in the range ($\rho_d \in [0,1], \rho_s \in [0,1], \alpha \in [0,0.5]$) in Ward parameter space. These similarity values are derived from probabilistic distance measures on parametric BRDF distributions corresponding to the target distributions of $ePL_3(\mathbf{X}|\theta)$ (Eqn. 6.62). (The data in this similarity matrix has been transformed linearly for better visibility; the ordering for similarities between material pairs is preserved).

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	742.4	745.4	778.6	659.2	954.7	973	948.6	965.8	983.3	989.5	986.2	986.8	981.7	978.7	982.9	984
2	747.3	754.6	785.9	665.2	960.3	977	956.1	971.8	985.7	992	988	989.4	985.2	980	982.7	984.6
3	768.2	769.8	802.1	692	946.3	969.3	940.3	962.6	980.4	988	984.4	984.8	981	973	979.1	980.4
4	600.9	609.8	652.1	478.1	987.3	991.3	986.3	991.3	990.3	995.3	996.2	997.2	977.5	996.6	996.6	997.9
5	863.7	865.8	890.9	812.9	875.2	917.9	861.9	902	949.4	967.7	961	965.5	947.3	931.3	946	951.4
6	792.2	802.1	829.7	736.9	939.9	966.4	930.9	955.2	981.9	988.3	987.7	986.3	985.2	971	980.6	980.8
7	886.6	889.6	912	845.4	840.5	890	822.6	868.7	932.2	959.8	946.7	955.5	928.9	909.2	925.8	937.2
8	667.3	670	709.6	564.6	979.1	988	977.1	985.5	991.1	993.3	992.4	993.3	985.4	990.7	992.1	992.3
9	663.1	665	705	551.9	971.2	981	968.9	978.6	985	992.2	989.3	991.3	976	987.1	988.4	989.9
10	709.1	711.1	747.8	618.1	957.1	973	952.3	966.7	982.7	990.4	986.6	987.8	978.1	981.5	983.2	985.5
11	595.9	606.1	648.5	472.6	988.9	992.3	988	992.5	990.7	995.5	996.8	997.6	977.3	997.4	997.3	998.5
12	815.7	817.5	846.5	750.5	913.7	942.9	904	930.9	964.7	977.5	971.4	973.6	963	956.4	962.3	966.5
13	648.2	652.7	693.2	539.1	987.8	993.6	986.9	992.6	994.4	995.6	995	996.7	987.4	995.6	995.6	996
14	610.2	619.4	660.8	494.3	993	996.1	992.6	996	994.5	995.3	997.2	997.5	984.4	998.5	998.3	998.6
15	620.6	630.3	671.2	505.6	992.5	995.7	992	995.6	994.5	995.8	997.2	997.8	984.3	998.3	997.9	998.5
16	593.1	602.9	645.3	473.3	994.5	996.5	994.1	996.9	994	994.8	997.6	997.5	982.8	999.2	998.8	999.1
17	994.5	992.6	989.7	986.9	559.2	718	540.9	713.1	764.8	896.1	911.5	855.3	830	667	745	773.3
18		994.4	986.4	989	554	704.5	537.3	709.5	759	886	927.6	860.8	822.7	669.8	745.5	779.6
19			994.3	978.8	639	766.3	621.5	764.9	811.9	917.4	931.7	887.4	865	737	801.1	826.9
20				995.5	471.7	618.5	457.1	634.4	690.3	823.4	917.4	819.1	751.3	613.2	688.4	733.1
21					999.8	997	999.9	998.4	990.8	984.6	991	989	981.1	995	993.8	992.5
22						997.4	995	997.1	995.3	991	993.5	993.4	989.2	995.7	995.5	994.2
23								998.6	991.3	985	991.3	989.3	981.8	995.3	994.1	992.7
24								998.7	994.6	990	994	993.4	986.8	997	996.4	995.3
25									996.3	992.6	992.8	993.1	994.1	992.7	994.3	992.4
26										994.7	992.8	993.6	986.2	988.8	990.4	991.8
27											996	993.8	986.9	992.1	993.7	994.1
28												996.6	985.7	994.5	995.1	995.9
29													996.2	986.5	991	987.8
30														998.8	998.4	998.4
31															997.5	996.9
32																998.4
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																

Table 8.4: Columns 17 to 32 of similarity matrix for parametric BRDF surfaces of real materials, continued from Table 8.3.

	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
1	979.8	980.9	982.4	984.6	987.9	991.8	992	989.9	484.2	306.9	328.6	495.5	738	817	811.9	667.5
2	983.5	985.1	982.9	985.9	983.7	987.4	989.5	987.6	483.9	296	317.5	506	747	824.4	822.4	668.2
3	978.1	978.5	978.8	982.7	989.1	991	991.8	990.6	520.9	347.6	369.7	533.7	765.4	839.6	833.1	698.5
4	986.3	986.8	994.3	992.8	969.4	985.3	984.9	979.2	270.2	34.51	53.16	275.4	585.7	683.8	690.4	458.4
5	940.6	948.8	947	952.1	990.2	981.7	985.2	986.9	663.4	514.7	538.4	675	872.3	921.2	918.8	822.8
6	979.7	981.7	982.1	985.9	984.2	986.2	985.9	984.5	568.8	428.5	450.3	609	807.4	880.4	866.6	778.4
7	919.6	928.8	932	934.7	986.3	973.3	977.9	980.9	701.5	572.8	596.7	716	899	941.3	937.9	860.3
8	988.7	989.2	991.2	991.1	974.1	985.8	986.4	981.2	373.4	165.9	186.5	385.7	657.1	749.5	745.9	566.6
9	981	980.8	985.6	986.2	982.3	991.7	991.4	988.6	358.6	144	162.6	356	645.3	733.9	737.8	533.4
10	978.7	979.5	980.4	982.9	987.4	993.5	992.8	991.4	433.6	249.6	271.8	437.8	702.1	785.8	781.2	621.6
11	986.9	987.1	995.1	993.5	967.7	984	983.7	977.7	261.9	23.35	42.34	269.4	581.9	680.5	688.3	451.5
12	957.5	963.7	960.4	965.8	991.9	988.5	989.9	991.6	591.5	432.9	456.4	597.7	818.5	879.8	875.9	758.6
13	992	992.7	994.4	993.4	971.6	985.1	986.3	980.3	342	121.8	142.1	355.1	636.9	732.4	730.5	535.6
14	992.3	992.1	997.1	995.2	964	980.6	981.4	974.2	288	59.13	79.28	304.4	601.2	701.1	702.4	489.4
15	992.2	991.9	996.4	994.7	966	981.7	982.8	976.1	299.7	69.07	89.06	316	611.8	709.6	712.2	496.1
16	991.5	991.8	997.5	995.3	960.6	978.6	979.3	971.5	263.9	30.22	50.46	279.8	583.5	685.6	687.9	468.2
17	792.6	751	787.2	858.6	918.9	825.7	864.4	866.4	898.1	777.1	802.9	916.4	989.7	984.6	982.7	934.6
18	779.2	738.8	775.7	842	904.5	815.4	844.1	854.9	920.5	825.7	846.2	943.5	992.1	986.3	987.8	962.2
19	831.1	801.1	828.4	876.7	938	875.3	897.1	905.7	889.4	776.6	799.4	907.6	988	989.7	987.6	941
20	694.3	662.5	696	748.4	837.4	759.9	763.2	800	969.6	920.8	932.4	980.2	990.8	976.6	982.9	983.7
21	990.6	990.6	994.5	991	934.5	957.2	958.9	947.9	220.6	23.47	18.31	255.5	552.4	664.9	659.4	458.3
22	994.7	994.9	995.8	993.8	954.4	971.7	973.5	965	304.8	93.37	112.9	332.5	610.8	714.5	706.5	525.8
23	991	991	994.7	991.2	935.3	957.8	959.5	948.6	224	4.207	22.7	258.6	554.7	666.9	661.1	461.4
24	993.8	994.2	996.5	993.7	948.8	968.3	969.9	960.3	269.3	52.76	72.28	297.2	585	692.7	685.6	495.1
25	995.9	995.9	995	994.8	966.8	980.4	981.3	974.2	370.5	179.2	200.7	393	651.2	750.6	737.5	590.3
26	988.1	988.4	989.1	990.6	977.9	987.1	987.8	984.3	413.6	219.6	241	434.2	693.8	781.6	777.2	611.6
27	990.7	990.4	992.9	993.4	970.1	983.3	983	977.3	363.5	173	195.2	391	658.3	754.5	747.3	584
28	990.9	991	993.7	993.1	971.6	984.7	985.6	980.2	348	131.8	152.2	364.9	645.1	738.8	737.8	544.1
29	994.9	994.2	993.7	995.2	972.3	981.7	982.8	976.2	445.3	277.1	296.9	472.7	704	796	780.5	662.5
30	993	993.2	997.5	995.3	960	977.8	978.7	970.6	279.9	53.03	74.14	298.9	594.4	696.9	695.6	489.5
31	993.2	993.3	997	995.4	962.9	979.3	980.1	972.4	303.3	86.64	107.1	322.4	608.8	709.8	706.2	514.2
32	991.7	991.6	996.6	995	965.3	981.3	982	975	294.9	69.77	89.77	311.8	606.7	705.7	706.7	497.2
33	996.8	995.9	995.2	995.3	965.2	978.3	979.2	971.6	380	200.5	221.7	407.9	657.3	758	742.9	608.8
34		996.4	994.3	994	964.4	978	979.1	971.7	374.4	186.6	207.7	401.4	655.1	754.9	740.2	599.4
35			996.4	995.4	963	977.8	978.8	970.9	338.3	137.7	158.6	363.3	632.6	732	725	557.2
36				995.1	970.3	981.4	982.4	975.7	408.5	223.4	246.2	435.3	681.6	774.9	764	622.9
37					992.5	987.9	989.5	990.2	599.3	439.3	462.4	609.5	823.3	883.9	879.6	760
38						994.2	992.4	991.9	483.7	326.5	352.1	492.9	739.1	820.5	809.9	687.9
39							993.4	991.5	519.5	335	356.9	528.7	767.3	839.1	836.6	687.9
40								993.2	521.1	353.4	378.2	529.2	772.7	843.8	839.1	703.1
41									997	992.2	993.1	993.7	957.8	920.4	948.4	960.9
42										999.3	998.4	997.5	934.3	884.3	924.7	939.2
43											998.2	996.3	933.5	884.5	922.9	937.9
44												997.1	953.1	913.7	945.5	957.2
45													995.4	986.5	989.1	978.1
46														993.9	992	971.2
47															993.9	981.5
48																996

Table 8.5: Columns 33 to 48 of similarity matrix for parametric BRDF surfaces of real materials, continued from Table 8.4.

for example as shown in Fig. 7.15, while others refer to comparisons done on PMC evolved sample sets, calculated over 20 iterations, distributed as shown (for example) in Fig. 7.19. In all cases, 1000 samples were used over PMC 20 iterations. In these similarity matrices, entry (i, j) in any table is the similarity between material $\#i$ and $\#j$. The similarities (distances) for the EMD measure have been inverted, since a high EMD between materials indicates a low similarity and vice versa. Some of the matrices have been linearly transformed, to make them easier to read (linear scaling does not affect the classification results at the algorithmic level, and preserves the relative similarity ordering between any pair of distances).

Tables 8.3, 8.4, 8.5 show a similarity matrix between sample sets representing estimated parametric BRDF distributions on real material surfaces, using our AGMMP similarity measure. In particular, there are 12 different material types. In this similarity matrix (and in those in Appendix E), columns $4n + 1, 4n + 2, 4n + 3, 4n + 4$, for $n = 0, \ldots, 11$, belong to the same material class, with the BRDF posterior estimated for different shapes and orientations of the same material (and the same is true for the rows). In the tables, materials belonging to different classes are separated by vertical or horizontal lines. The similarity measures in these tables are expected to be higher when materials of the same class are being compared.

In Tables 8.6 to 8.11, 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 similarity coefficient with the current material that is higher than the lowest similarity coefficient among the intra-class members of the current material. For example, on examination of material $\#1$ using the AGMMP measure with $\sigma = 0.005$ in Table 8.3 it is clear that the lowest intra-class similarity value for material $\#1$ (first four entries in the first row: the lowest value among entries $(1, 2)(1, 3)(1, 4)$, which are $(990.5, 992, 971.7)$ in Table 8.3) is 971.7. The "Miss" total is incremented for every entry in the first row $(1, 5)(1, 6)...(1, 48)$, which is above 971.7. Then, the next material type is considered, maintaining a running total for the Miss count, and so on. 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 [141].

Classification results using the AGMMP distance measure are shown in Table 8.6, for the case where the samples are in the initial locations, before any PMC sampling iterations have been done, with each sample weighted directly by the pseudolikelihood measure $ePL_3(\mathbf{X}|\theta)$ of Eqn. 6.62. In Table 8.6, classification error statistics are shown for sample sets compared using the AGMMP measure, for varying

σ	5e-1	1e-1	2e-1	3e-1	5e-2	1e-2	5e-3	1e-3	5e-4	1e-4	MAP	EMD	corr	IP
Miss	817	857	827	842	838	838	838	838	838	838	194	91	85	131
WNN	44	44	44	44	47	44	44	44	44	44	44	36	36	36
NN1	44	44	44	44	44	44	44	44	44	44	31	17	16	23

Table 8.6: Classification error statistics for the 48 surfaces, using the Ward reflectance model with samples in original positions. Statistics are shown for the classification done using the AGMMP measure with varying values of σ (first 10 columns), the MAP sample, the EMD, the correlation as defined in Eqn. 8.9 (called "corr"), and the inner product defined in Eqn. 8.10 (called "IP").

σ	5e-1	1e-1	2e-1	3e-1	5e-2	1e-2	5e-3	1e-3	5e-4	1e-4	MAP	EMD	corr	IP
Miss	812	854	812	831	827	827	827	827	827	827	216	84	93	147
WNN	44	47	48	48	48	48	48	48	48	48	40	35	37	37
NN1	44	44	45	45	45	45	45	45	45	45	29	16	17	25

Table 8.7: Classification error statistics for the 48 surfaces, using the anisotropic Phong reflectance model with samples in original positions. Statistics are shown for the classification done using the AGMMP measure with varying values of σ (first 10 columns), the MAP sample, the EMD, the correlation as defined in Eqn. 8.9 (called "corr"), and the inner product defined in Eqn. 8.10 (called "IP").

values of σ, as it appears in Eqn. 8.8. Finding a good value for σ is a task which depends on the data, and is done manually in our experiments. The values for σ which are tested in all PMC experiments are $0.5, 0.3, 0.2, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001$. Tables 8.6 and 8.7 show that the AGMMP gives extremely poor classification results for unevolved, weighted sample sets (the samples are in the original positions, regularly spaced on the cube of BRDF parameter space). Classification results when using the distances between MAP sample values are also poor when run on unevolved sample sets, (this statistic is shown in the column labelled "MAP"). The EMD measure gives good classification results (NN1 gives 17 out of a possible 48 errors), and the correlation gives good classification results also.

In Tables 8.6 and 8.7, error counts in the column labelled "corr" refer to comparison using correlation over the weights of the samples at their initial locations, evenly spaced on the unit cube of parameter space. Therefore, with the samples at identical locations on the parameter space for the two materials to be compared, one can treat the sets of sample weights $\vec{\pi}_1$ and $\vec{\pi}_2$ as vectors, since the particles are at corresponding locations in each of the unit cubes in the BRDF parameter space.

Tables 8.8 to 8.11 contain classification error results using PMC developed sample sets, evolved over 20 iterations, as features for the material surfaces.

In Tables 8.8 and 8.9, it is seen that the AGMMP distance measure gives good NN1 classification results. In Table 8.8, the best performance is achieved (for the NN1 statistic), with a minimum of 23 errors out of a possible 48 errors, when $\sigma = 0.1$ and $\sigma = 0.001$. In Table 8.9, the best NN1 classification results

σ	5e-1	1e-1	2e-1	3e-1	5e-2	1e-2	5e-3	1e-3	5e-4	1e-4
Miss	211	189	165	134	135	138	132	191	132	191
WNN	39	43	42	40	40	40	39	40	39	40
NN1	27	25	26	27	26	26	26	25	26	25

Table 8.8: Classification error statistics for the 48 surfaces, using the Ward reflectance model with PMC developed particle sets. The statistics are shown for the AGMMP distance measure, with varying values of σ.

σ	5e-1	1e-1	2e-1	3e-1	5e-2	1e-2	5e-3	1e-3	5e-4	1e-4
Miss	234	216	193	159	159	175	171	231	171	231
WNN	43	40	39	37	37	37	37	38	37	38
NN1	24	23	24	24	23	24	25	24	25	24

Table 8.9: Classification error statistics for the 48 surfaces, using the anisotropic Phong reflectance model with PMC developed particle sets. The statistics are shown for the AGMMP distance measure, with varying values of σ.

are 25 out of a possible 48 errors, obtained when $\sigma \in \{0.1, 0.001, 0.0001\}$.

In Tables 8.10 and 8.11, classification results are shown when using the different probabilistic distances shown in Table 8.2. In both tables, the best NN1 classification results are given when using the symmetric Kullback-Leibler divergence, which in each case gave 15 out of a possible 48 errors. An important result is that classification performance is always better when using a set of samples, rather than using only the distances between the MAP samples. The classification results obtained when using the Euclidean distances between the best parameter sets for the different materials, calculated using the Multiple-Seed LM method, with 512 seeds, is worse than any of the other classification methods, with 38 out of 48 NN1 errors for the anisotropic Phong model experiment, and 33 out of 48 NN1 errors for the Ward model experiment.

Notice also that the "corr" measure in Table 8.6, calculated using the original evenly spaced samples, gives the lowest Miss total for classification using unevolved sample sets. Interestingly, the EMD measure for the evenly spaced samples, weighted according to Eqn. 6.62, gives a lower (better) score than the corresponding EMD similarities for the sample sets developed over 20 PMC iterations (Tables 8.10 and 8.11). The AGMMP measure works better on the evolved sample sets than on the evenly spaced sample sets, but the classification results are generally worse for AGMMP than when using the other distance measures on the developed sample sets.

It should be clear that the "Miss" statistic is quite an aggressive statistic, since it counts every case where the similarity of a material surface with an extra-class material surface is better than the worst intra-class similarity. Moreover, with 48 surfaces and 12 classes of material, the highest possible Miss statistic for a given similarity metric (Gaussian similarity, EMD, correlation, etc.) is Miss = $44 \times 48 = 2112$,

	Chern	Bhatta	KL	Symm KL.	PF	Mahalanobis	MAP	EMD	LM512
Miss	157	155	134	119	198	160	174	161	412
WNN	39	38	37	37	47	38	39	37	48
NN1	22	22	19	15	26	24	25	23	38

Table 8.10: 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 Chern, with α_1 arbitrarily set to 0.7), the Bhattacharyya distance (Bhatta), the Kullback-Leibler Divergence (KL), the Symmetric KL Divergence (Symm KL), the Mahalanobis distance, 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.

	Chern	Bhatta	KL	Symm KL.	PF	Mahalanobis	MAP	EMD	LM512
Miss	117	116	98	80	165	121	157	135	241
WNN	42	41	38	38	47	41	42	39	42
NN1	23	24	18	15	26	26	27	25	33

Table 8.11: Classification error statistics for the 48 surfaces, using the Ward reflectance model with PMC developed particle sets. The statistics are shown for the Chernoff distance (labelled Chern, with α_1 arbitrarily set to 0.7), the Bhattacharyya distance (Bhatta), the Kullback-Leibler Divergence (KL), the Symmetric KL Divergence (Symm KL), the Mahalanobis distance, 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.

which would occur if every one of the materials had one extremely dissimilar within-class candidate material. A Miss statistic of 400 is therefore quite low, since it means that only 19% of the possible class violations occurred. This measure gives an indication of how reliably the materials can be clustered based on such similarity measures.

Upon viewing coarse volumetric plots of the pseudolikelihood function on the BRDF parameters (Eqn. 6.61) of three of the materials, it is clear why misclassifications may occur. The volumetric pseudo-likelihood plots over the BRDF parameters for materials #8 and #9, shown in Fig. 8.16 are similar in appearance, and it is difficult to distinguish between them with the naked eye. Material #11, shown in Fig. 8.17 clearly has a different BRDF, which makes its classification easier. Note also that the high probability regions as indicated in the plots agree with our expectations about the Ward reflectance properties of the material surfaces. Materials #8 and #9 are expected to exhibit little specular reflectance and a large amount of diffuse reflectance. Material #11 is expected to show little diffuse reflectance, high specular reflectance, and a very high roughness parameter α, since the specular lobes are large over regions of varying inclination. These characteristics are evident in the plots of Figs. 8.16 and 8.17.

The statistics for "Miss", "NN1" and "WNN" over the forty-eight surfaces, and for each of these probabilistic distances are shown in Tables 8.7, 8.9 and 8.10 for the anisotropic Phong parameterisation, and in Tables 8.6, 8.8, 8.11 for the Ward parameterisation. It is shown in these tables that the Ward parameterisation gives better classification results. This is consistent with our discovery (using synthetic data) that smooth flat surfaces are worse for recovering the anisotropic Phong parameterisations than jagged surfaces. Since the real surfaces we reconstructed and classified using our method were smooth and comparatively flat, it is consistent that the Ward parameterisation should give (moderately) better classification.
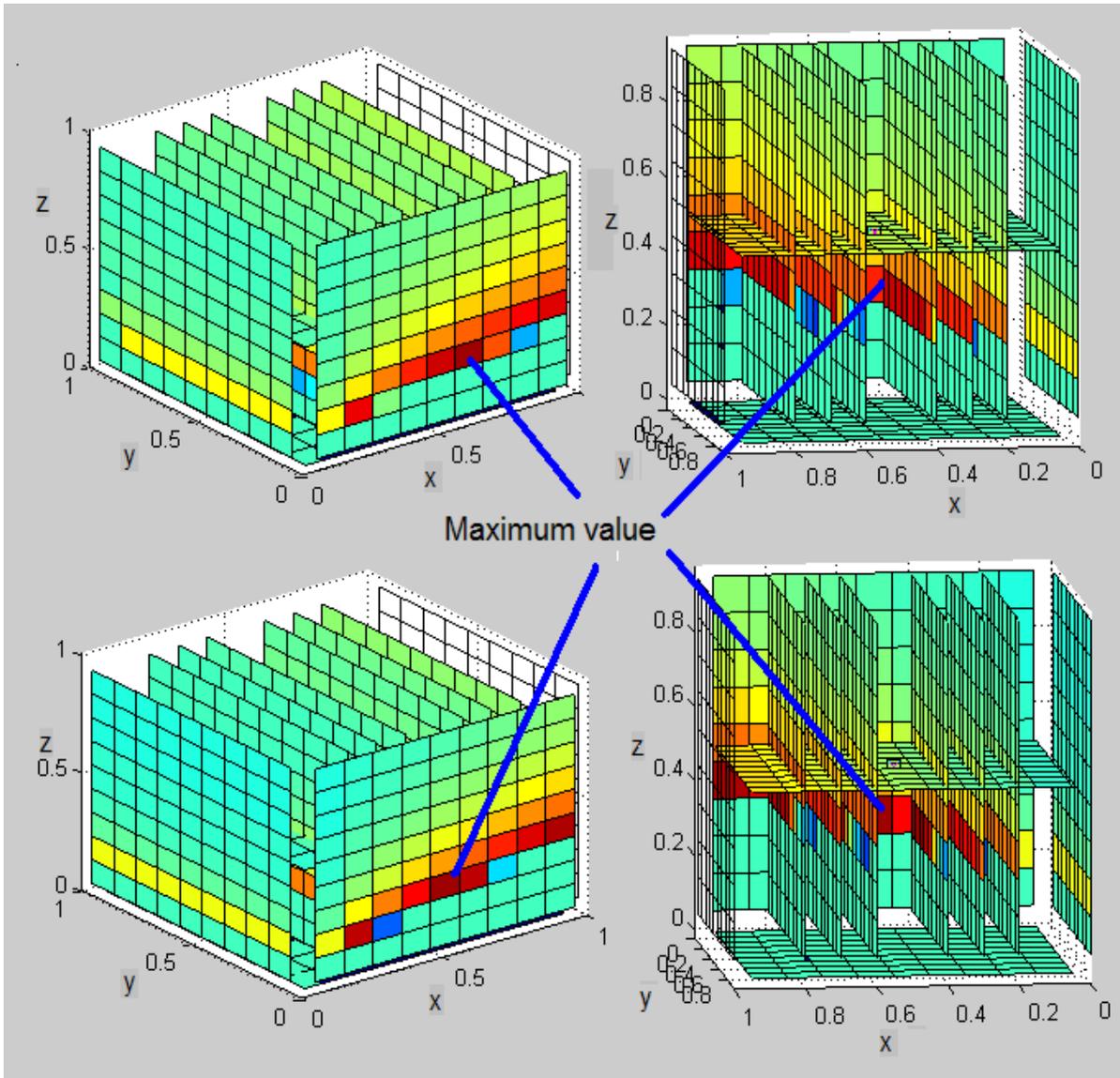
Figure 8.16: Volumetric plot of the pseudolikelihood estimate $PL_3(\mathbf{X}|\theta)$ for the probabilities on the Ward BRDF parameters on materials #8 and #9. The pdf has been evaluated at the same locations where a sample set for the PMC method would be initialized, but the samples have not been superimposed onto these plots. $(x, y, z) = (\rho_d, \rho_s, 2\alpha)$.
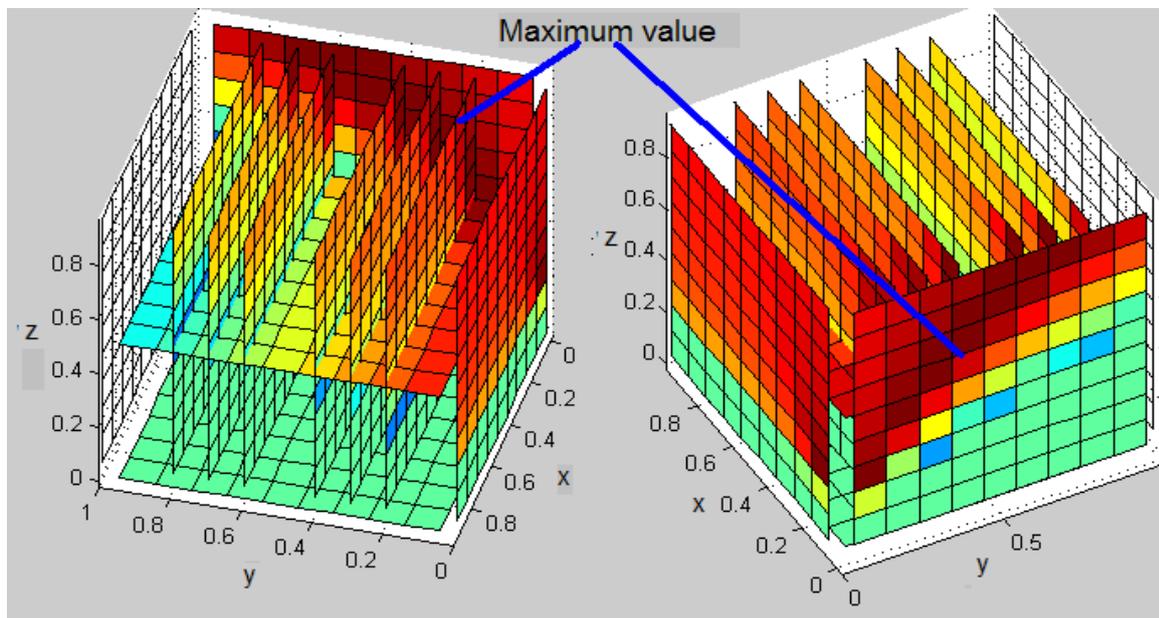
Figure 8.17: Volumetric plot of the pseudolikelihood estimate $PL_3(\mathbf{X}|\theta)$ for the probabilities on the Ward BRDF parameters on material #11. The pdf has been evaluated at the same locations where a sample set for the PMC method would be initialized, but the samples have not been superimposed onto these plots. $(x, y, z) = (\rho_d, \rho_s, 2\alpha)$.

numLabels	LM (512 seeds, 10 iterations)	R-type MCMC (20000 iterations)	PMC (1000 samples, 20 iterations)
10	1.3 hours	9 minutes	4 minutes
20	2.6 hours	18 minutes	8 minutes
30	3.8 hours	27 minutes	12 minutes
40	5 hours	36 minutes	16 minutes

Table 8.12: Approximate average running times for the BRDF estimation algorithms. The processor used was an AMD Athlon 4600.

8.3   Running times of MRF/BRDF parameter estimation algorithms

The estimation algorithms proposed for parametric BRDF estimation (multiple-seed Levenberg Marquardt, dynamically weighted MCMC, and PMC) require varying amounts of time to converge. To compare the computational load for each of these methods, the running times of the three algorithms are presented in Table 8.12. These running times are for both the real and synthetic data, for the cases where each corner vertex node can take on any of 10,20,30 or 40 labels corresponding to 3-D locations or disparity values. The computational load increases approximately linearly with the number of possible labels for the corner vertex nodes since the number of labels affects the computational complexity of calculation of the denominator in the pseudolikelihood calculations of Eqns. 6.62 and 7.7 linearly. The multiple-seed LM algorithm was run with 512 seeds, the MCMC algorithm was run for 20000 iterations, and the PMC algorithm was run for 20 iterations with 1000 samples per iteration. The Multiple-seed LM algorithm was significantly slower than the other two algorithms. The reason that 512 seeds were chosen as the upper limit is that it is at this point that the average error function for this algorithm run on synthetic data reliably reaches a good minimum value when run on synthetic data, as seen in Figs. 7.3 and 7.4.

In the case that only a single MAP value is required, the data shows that if less than 15 minutes is available for calculation, the PMC algorithm is appropriate. If 5 hours is available, then the Multiple-seed LM algorithm run with 512 seeds is likely to give better results. The PMC algorithm was not tested under operating parameters which could take advantage of the greater amount of time. It would be possible to process more samples than the current maximum of 1000 samples for each of 20 PMC iterations, or to run the PMC algorithm for more iterations, or to process more samples over more iterations in the given time, but this was not tested (to run a battery of tests each of which takes five hours is extremely time consuming). To extract BRDF posterior distribution samples for features for classification, the PMC algorithm is the only possible choice.

Chapter 9

Conclusions

We have described a method for extracting BRDF features from calibrated image data. The image data is obtained using a stereo camera pair and a digital light projector that is used to improve the dense stereo correspondence estimation using structured lighting, as well as to supply a light source which is used to estimate the BRDF of the reconstructed surface. These features consist of the posterior probability distributions on MRF parameters, which are actually BRDF parameters which have been used as MRF potential function parameters. This framework can parameterize the reflectance characteristics of a wide set of material surfaces, but cannot, in its present form, handle material surfaces that exhibit complicated radiometric phenomena such as translucence, radiosity, and subsurface scattering.

Our method uses a Population Monte Carlo iterative resampling scheme to simultaneously approach the optimal set of parameters, while also returning a posterior distribution over the BRDF parameters in this parameter space. This method can take into account the probabilistic uncertainties of a surface which has been reconstructed from a dense stereo correspondence algorithm. This is also the first classification algorithm (to our knowledge) which uses posterior distributions in BRDF parameter space as features for object surface classification.

Our method regards the reconstructed surface as a realization of a Markov random field in the space of possible depth/disparity fields for the reconstructed surface, and treats the parameter estimation of the BRDF model as a MRF parameter estimation problem.

Two novel methods for doing Markov random field parameter estimation were proposed and investigated in this thesis, namely a Population Markov Chain Monte Carlo sampler, and a dynamically weighted Markov Chain Monte Carlo sampler.

A pseudolikelihood measure is used to weight the relative "fitness" (i.e. the congruence of the parameterisation with the observed image data and known light source and camera configurations) of each of the tested parameter sets (samples/particles). In this way posterior distributions on the BRDF

parameterisations are derived, using both dynamically weighted MCMC and Population Monte Carlo samplers. The final posterior distributions on the BRDF parameters are intended for use in surface categorisation/classification, but could also be used as parameters for a MRF based dense stereo reconstruction algorithm, to give a probability on the configuration of triplets of corner vertex nodes on the object surface, and in this way improve surface reconstruction estimates.

We tested the convergence of different sampling algorithms for reaching the MAP value for the reflectance parameterisation, using synthetic data, with the Ward and anisotropic Phong BRDF models. We compared a Levenberg-Marquardt nonlinear optimizer initialized on multiple seeds, dynamically-weighted MCMC samplers, and a Population Monte Carlo sampler, to test the relative performances of these algorithms for finding the true BRDF parameterisations which were used to generate the images of the synthetic surfaces. For the MCMC and PMC algorithms, a sample set representing the posterior distribution on the BRDF parameters is also generated in addition to a good MAP estimate.

Using synthetic data, it is found that the Multiple-seed LM minimizer, minimizing an error function (Eqn. 7.10) which maximizes a pseudolikelihood corresponding to the fitness of a sample representing the BRDF parameter vector, converges less reliably when given fewer seeds for initialization. The reliability of the Multiple-seed LM algorithm for finding the correct BRDF parameters is improved by increasing the number of seeds on which the algorithm is initialized (the best result is the one which is retained), however this results in an increase in computational load. When 512 seeds are used, the Multiple-seed LM algorithm has improved reliability of convergence to the correct parameter set (using synthetic data), but takes about 20 times longer than either the PMC or dynamically weighted MCMC sampling algorithms. The Multiple-seed LM method also does not yield a posterior distribution on the BRDF parameters, which we have shown to be valuable information for material surface classification.

The dynamically weighted MCMC samplers (Q-type and R-type) have more stable convergence characteristics than the LM method, but by the time the MAP sample is within a reasonable distance (Euclidean distance < 0.1 in the Ward parameter space, which is taken to be the unit cube in $\mathbb{R}^3$) of the true BRDF parameter set for the synthetic surface image, there are too many samples (about 20000) for the distributions to be compared efficiently using available pdf comparison techniques. This caused us to test only the results of the LM and PMC methods for performing real material classification.

When testing the Population Monte Carlo (PMC) method with 1000 samples over 20 iterations, for parametric BRDF estimation on synthetic data, it was found that the MAP sample reliably converged to within a Euclidean distance of 0.14 units from the true parameter values in the unit cube of parameter space, in the experiments using the Ward model. The convergence in the case of the anisotropic Phong model was better, with an average distance of < 0.05 units (the difference is due to the fact that the anisotropic Phong experiments were run on jagged synthetic surfaces: these differences are also present in the dynamically weighted MCMC convergence results and the Multiple-seed LM convergence results, where the convergence results when using the anisotropic Phong model are slightly

better). There is also a steady decrease in sample variance for all disparity ranges tested, for both reflectance models.

The posterior distributions on the BRDF parameters, as calculated using the PMC method, were then tested for their potential use as features for classification, using a variety of inter-feature distance measures including the Earth Mover's Distance and our new symmetrical per-sample Gaussian Mixture Model based distance measure (called the Average Gaussian Mixture Model Probability measure), using both Ward and anisotropic Phong reflectance models of the real material surfaces. Chernoff, Bhattacharyya and Patrick-Fisher distances, as well as the Kullback-Leibler and Symmetric Kullback-Leibler divergences were also tested for their use as inter-feature distances in comparing the evolved particle sets, on the simplifying assumption that the parametric BRDF samples were generated from normal distributions. A first-nearest-neighbour (NN1) and worst-nearest-neighbour (WNN) misclassification testing methodology was used to reveal the types of inter-feature distances which gave the fewest nearest neighbour misclassifications. It was found that the Symmetric Kullback-Leibler divergence is the inter-feature distance measure which gave the fewest NN1 misclassifications, on our real surface data. Few nearest-neighbour errors indicate good clustering of the data using the selected features.

We also tested the classification methods on the real materials using as features the BRDF parameter sets derived using the best Multiple-seed LM parameter sets (i.e. from all of the LM minimization operations done on the set of parameter seeds, the solution with the lowest error was chosen).

The results show that the PMC-based classification method outperforms the Multiple-seed LM method when the calculated features for each method (the weighted sample set for the PMC method and the lowest error parameter vector for the Multiple-seed LM method) are used for classification on real material surface data, reconstructed using a stereo camera pair with structured lighting for dense stereo reconstruction.

The PMC method is therefore a suitable method for doing MRF parameter estimation, for calculating posterior distributions on parametric BRDFs for material surfaces, given probabilistic surface reconstructions. The sample sets calculated using the PMC method are reasonably good features for material surface classification.

9.1 Novel contributions

The novel theoretical contributions of the thesis include the estimation of parametric BRDF model parameters within a Markov random field pseudolikelihood framework with the explicit incorporation of the uncertainties in the reconstructed scene geometry. This is the first time, to our knowledge, that a Population Monte Carlo sampler, rather than a Markov Chain Monte Carlo sampler, has been

used for Markov random field parameter estimation in any setting. There appears to be no reference in the literature that a dynamically weighted MCMC sampler has ever been used for MRF parameter estimation: this is one of the techniques we explore. In Chapter 8, we have developed a new similarity measure for comparing sample sets which represent posterior distributions, which to our knowledge has not been used before, namely the Average Gaussian Mixture Model Probability (AGMMP) measure.

9.2 Final criticisms

In the framework of monocular vision, many ambiguities occur in reconstructing the 3D-surface of a scene from one 2D-image. In such situations, the stochastic methods which use a multihypotheses algorithm are often efficient. This is the basis of the work we have presented here. The problem of 3D-reconstruction admits a variety of approaches: there can be one or several images of the same scene, taken either under the same point of view but different lighting conditions (photometric stereo) or under the same lighting but different points of view. There may or may not be boundary conditions or range data available for the surface to be reconstructed. If it is available, it may only be given as soft probabilistic constraints. The framework as presented here caters for either case.

The photometric characteristics of the materials i.e., their BRDFs, can also be known, or not. With more information, fewer ambiguities occur, and the 3D-surface should be better reconstructed. The real advantage of the stochastic methods using a multi-hypotheses algorithm is that they allow the formulation of all the variants of 3D reconstruction in a common formalism.

The framework we have presented allows any sampling algorithm to be used to extract the parametric BRDF of a surface since it has been embedded in a Markov Random Field parameter estimation problem. This version of the problem naturally allows one to include hard or soft constraints on the depths of any of the surface points. This is the core advantage of the framework, since any improved sampling or parameter estimation problems may immediately be applied using our framework. The second major theoretical benefit of using MRF potential terms based on the BRDF of the surface to be reconstructed, is that it allows a physically meaningful constraint to be imposed, in contrast to the many existing MRF based surface smoothing algorithms, where the potential terms are often ad-hoc and physically meaningless.

A final criticism pertaining to the utility of this method in favour of others for the purpose of froth classification is that the framework introduced is overly complex, and that better classification results may be found using much simpler statistical methods which do not rely on 3D reconstruction or BRDF estimation.

While it is true that classifiers such as support vector machines and neural networks could have better classification accuracy when given the task of labeling images taken from froth footage (or surfaces of

other types), our method has both classification power, but also calculates geometry and BRDF features. Our research group has hypothesized that knowledge of the geometry of individual froth bubbles may give insight about the the level of material loading on the froth surface. In addition, knowledge about the reflectance characteristics of the froth can give insight as to the quality and type of the mineral ore particles contained on the froth surface. Therefore the intermediate calculations are also of potential value in the automation of the froth flotation control process.

9.3 MRF parameters and their relationship to BRDF parameters

It is difficult to relate the concept of a BRDF parameter such as the alpha parameter in the Ward model to its corresponding variable in an MRF pseudolikelihood equation such as Eqn. 6.61.

While the two alphas mean the same thing in each case (i.e. the standard deviation of the surface slope at any point), it is correct to say that the two alphas are not identical, since the estimation of the posterior distributions on the Markov Random Field parameters involve a cost function which involves the pseudolikelihood concept developed in chapters 4 and 6. Furthermore it can involve other potential function weighting terms such as the temperature and normalization constant of Eqn. 6.60.

In its present form, the probability distribution on the BRDF parameters is calculated using assumptions about the scene geometry and the relative certainties on each of the reconstructed points. If the assumptions are accurate, then the probability on any MRF/BRDF parameter is well justified, and the posterior distribution on the MRF parameter becomes as accurate as the model with its parameters allows. If, however the potential function and other pseudolikelihood parameters are not well chosen, the resulting posterior distribution will not give a fair indication of the BRDF parameters calculated, for that scene.

9.4 Future work

The framework we have provided for approaching the shape from shading problem, and for estimating BRDF parameters by solving for the MRF parameters on this structure allows for many further avenues of exploration. It may be possible to use other sampling techniques, including genetic algorithms, to improve the posterior estimate on the MRF parameters. It may also improve the computation time and accuracy to divide the calculation of the parameters into a two-step solution, in an expectation maximization framework, where the probabilities on the disparities are calculated alternately with the MRF parameter estimates. Using our framework, it should be possible to solve for the incident light direction, or the camera parameter, if the surface and BRDF characteristics are known. Our approach can be extended to include multiple light sources, in a stereo or monocular vision setting. Finally in the classification stage, more effective methods can be used to compare the posterior distributions on

the BRDF parameters, and the distances between them. A way of incorporating self-occlusion in the surface (from the light source) has not been developed.

Appendix A

Multinomial Distribution and Multinomial Sampling

In a random process where

- there are n independent trials

- each trial produces one event $E_1, E_2, \ldots, E_k$, where these events are mutually exclusive and collectively exhaustive, so $\pi_1 + \pi_2 + \ldots + \pi_k = 1$

- on each trial, E_j occurs with probability $\pi_j, j = 1, 2, \ldots, k,$

we can define random variables $\mathbf{X}_1, \mathbf{X}_2, ..\mathbf{X}_k$ such that $\mathbf{X}_n$ is the number of trials in which E_n occurs. Then, $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_k\}$ is said to have a multinomial distribution with index n and parameter $\pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$. Usually n is fixed and known.

The individual components of a multinomial random vector are binomial and have a binomial distribution: $X_n \sim \text{Bin}(n, \pi_n)$. Each trial is independent, but the groups of events are not independent, since given a known number of events of one type we have some information about the number of other types of events. If $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_k\}$ is multinomially distributed with index n and parameter vector $\pi = \{\pi_1, \pi_2, ..., \pi_k\}$, it may be written as

$$\mathbf{X} \sim \mathcal{M}(n, \pi) \tag{A.1}$$

This notation may be used to indicate a single sample taken from a set of k samples (particles) with probabilities $\pi_1, \pi_2, ...\pi_k$, by writing $\mathbf{X} \sim \mathcal{M}(1, \pi)$, and is used in this thesis for describing resampling operations.

Appendix B

Pixel Match Measures

This appendix describes some of the pixel/corner match measures which are used in disparity calculation algorithms, and which may be used in the BRDF parameter estimation algorithm proposed in this thesis.

B.1   2-D Correlation

Given two image regions A and B, the two dimensional correlation between the two regions is:

$$r = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \overline{A})^2)(\sum_m \sum_n (B_{mn} - \overline{B})^2)}}. \tag{B.1}$$

This value is in the range $[-1, 1]$. To convert this into an (approximately) probabilistic value, the values are scaled to be in the range $[0, 1]$.

B.2   Sum of Square Differences

The sum of square difference (SSD) measure is as follows:

$$r = \sum_m \sum_n (A_{mn} - B_{mn})^2. \tag{B.2}$$

This measure returns values which are larger the greater the difference between the image regions. To convert this into a probabilistic value one can invert the values, or take the negative of the values and add a constant, then scale them to be in the range [0,1].

B.3 Linear optimization of transport problem with Earth Mover's Distance

This is a method designed to solve the linear transport problem using linear optimization. Given two distributions, or sets of features (in [100] these are called "signatures"), with costs, or distances between each member of the first set to each member of the second set, the Earth Mover's Distance represents the cost of transporting mass from the first distribution to the second distribution, along the paths with the specified costs. This algorithm also allows for partial matching, where the distributions have different masses. This algorithm is well suited to comparing vary abstract dataset features, provided that the similarity between each type of feature can be described mathematically. It is also easy to extend this algorithm into a local window or region matching framework. The linear optimization algorithm on this cost function is sometimes called "EMD", but this is a misnomer.

B.4 Colour Histogram Intersection

Histogram intersection methods are useful for region matching because they are robust against objects being in different orientations. Provided there is enough colour separation in the pixels, a histogram intersection based feature measure can accurately match features which are rotated by 180 degrees in the image plane, and is even fairly robust to small rotations about the principal axes, provided that all parts of the feature remain visible (again, assuming good colour separation across features). The basic histogram intersection equation is

$$H(\mathbf{I}^1) \cap H(\mathbf{I}^2) = \sum_{i=1}^{n} \min(H_i(\mathbf{I}^1), H_i(\mathbf{I}^2)), \tag{B.3}$$

where H_i is a function which counts the number of pixels in the image $\mathbf{I}$ which are in a certain range of values. This is extensible to colour images. If the image region which is to be compared is small, the buckets should be made large.

B.5 Mutual Information

Mutual information is an information theoretic measure of the similarity between data sets. Assuming a random variable $\mathbf{A}$, the entropy is calculated as

$$h(\mathbf{A}) = -\int p(\mathbf{A}) \ln p(\mathbf{A}) d\mathbf{A}. \tag{B.4}$$

If there are two random variables $\mathbf{A}$ and $\mathbf{B}$, the joint entropy is

$$h(\mathbf{A}, \mathbf{B}) = -\int p(\mathbf{A}, \mathbf{B}) \ln p(\mathbf{A}, \mathbf{B}) d\mathbf{A} d\mathbf{B}. \tag{B.5}$$

The mutual information of the two random variables is

$$I(\mathbf{A}, \mathbf{B}) \equiv h(\mathbf{A}) + h(\mathbf{B}) - h(\mathbf{A}, \mathbf{B}). \tag{B.6}$$

It is clear that to use a mutual information measure it is necessary to assert some way of finding the probability of a random variable (pixel value in a region), as well as the joint probability of two such random variables (pixel values) in the data (image regions) are being compared. One method is to normalize the pixel intensity co-occurrence matrix, so that it can be treated as a joint pdf. For an example case study, see [118], where mutual information is used to register MRI data. The mutual information must also be converted into some kind of probability value if it is to be used in a probabilistic pixel match measure algorithm.

Appendix C

Thin plate splines

This appendix briefly describes the thin plate spline (TPS) interpolation method, which we use in our synthetic data testing methodology to generate synthetic surface data to test the convergence characteristics of the BRDF estimation algorithms. Given control points, (which in our scenario are randomly generated from a cube in $\mathbb{R}^3$), the thin plate spline algorithm fits a surface to these points which minimizes a specified bending energy. Although the idea was introduced in [28], we follow the formulation of [30].

One seeks to construct an interpolating function $f(x, y)$ which can generate smooth surfaces. The bending energy of the thin plate spline surface is

$$I[f(x,y)] = \int \int_{\mathbb{R}}^{2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dx dy. \tag{C.1}$$

If the data comes in the form of a set of points $\{(x_i, y_i, z_i)\}_{i=1}^{N}$, the minimizing function can be written as

$$f(x,y) = \sum_{j=1}^{n} a_j E(||(x - x_j, y - y_j)|| + b_0 + b_1 x + b_2 y), \tag{C.2}$$

where

$$E(r) = r^2 \ln(r^2). \tag{C.3}$$

The coefficients a_j and b_j are determined by requiring exact interpolation. This requirement implies

$$z_i = \sum_{j=1}^{n} E_{ij} a_j + b_0 + b_1 x_i + b_2 y_i, \tag{C.4}$$

for $1 < i < n$, where $E_{ij} = E(||(x_i - x_j, y_i - y_j)||)$. Writing this in matrix form,

$$\vec{z} = A\vec{a} + B\vec{b}, \tag{C.5}$$

where $A = [E_{ij}]$ is an $n \times n$ matrix and where B is an $n \times 3$ matrix, the rows of which are $[1 \quad x_i \quad y_i]$. An additional implication is that $B^T \vec{a} = \vec{0}$. The two vector equations can be solved to obtain:

$$\vec{a} = A^{-1}(\vec{z} - B\vec{b}) \text{ and } \vec{b} = (B^T A^{-1} B)^{-1} B^T A^{-1} \vec{z}. \tag{C.6}$$

These coefficients can then be used for interpolation between the data points.

Appendix D

Levenberg-Marquardt function optimization

In this appendix we summarize some nonlinear minimization methods, including Newton's method, the Gauss-Newton method, the basic gradient descent method and the Levenberg-Marquardt (LM) method. We follow almost exactly the description of [49], Appendix 6. Assume we are given a functional relation $\mathbf{X} = f(\mathbf{P})$ with $\mathbf{X}$ a measurement vector in $\mathbb{R}^N$ and $\mathbf{P}$ a parameter vector in $\mathbb{R}^M$. However, the measurement $\mathbf{X}$ is only an approximation of the true value $\overline{\mathbf{X}}$. The goal is to find a vector $\hat{\mathbf{P}}$ which most closely satisfies this functional relation. We can pose this goal as one of minimizing an error $||\epsilon||$ in the equation

$$\mathbf{X} = f(\mathbf{P}) - \epsilon. \tag{D.1}$$

When f is not a linear function, the estimate of the function f is improved iteratively, under the assumption that f is locally linear. If the initial guess for $\mathbf{P}$ is $\mathbf{P}_0$, then the error corresponding to this initial estimate is $\epsilon_0 = \mathbf{f}(\mathbf{P}_0) - \mathbf{X}$. It is further assumed that the function in the region around $\mathbf{P}_0$ is linearly estimated as $\mathbf{f}(\mathbf{P}_0 + \Delta) = \mathbf{f}(\mathbf{P}_0) + J\Delta$, where J is the Jacobian matrix calculated as $J = \partial \mathbf{f}/\partial \mathbf{P}$. The next point $\mathbf{P}_1$ is found by $\mathbf{P}_1 = \mathbf{P}_0 + \Delta$, which minimizes the term $\mathbf{f}(\mathbf{P}_1) - \mathbf{X} = \mathbf{f}(\mathbf{P}_0) + J\Delta - X = \epsilon_0 + J\Delta$. Finding Δ to minimize $\epsilon_0 + J\Delta$ is a linear problem. At this point the normal equation is introduced:

$$J^T J \Delta = -J^T \epsilon_0, \tag{D.2}$$

which is the general equation which is the basis for many gradient descent style minimization problems, including the LM method.

D.1 Newton's method

In Newton's method, the solution vector $\mathbf{P}$ is calculated iteratively according to $\mathbf{P}_{i+1} = \mathbf{P}_i + \Delta_i$, where Δ_i minimizes $J\Delta_i = -\epsilon_i$, and where J is the Jacobian $\partial \mathbf{f}/\partial \mathbf{P}$ at $\mathbf{P}_i$ and $\epsilon_i = f(\mathbf{P}_i) - \mathbf{X}$.

D.2 Gradient descent

The downhill gradient vector $-g_p$ where $-g_p = -\epsilon_P^T \epsilon$ is the direction of the most rapid decrease in the cost function. One of the key issues in gradient descent algorithms is the size of the step to take, in the direction of $-g_p$. The step size may be fixed, but some algorithms use a line search method in the direction of $-g_p$ in order to minimize the cost function. In this case, the parameter increment Δ is calculated from $\lambda\Delta = -g_p$ (λ controls the step size). The key difference between this method and Newton's method is that the Hessian of Newton's method is here replaced by λI, where I is the identity matrix, with width and height equal to the size of the parameter vector.

D.3 Gauss-Newton

Suppose one is considering a function $g(\mathbf{P})$. If one expands this function about $\mathbf{P}_0$ as a Taylor polynomial, then

$$g(P_0 + \Delta) = g + g_P\Delta + \Delta^T g_{PP}\Delta/2 + \ldots . \tag{D.3}$$

where the subscript P or PP indicates the order of the derivative. Since the goal is to find the minimum of this function (ignoring the higher order terms), it is differentiated with respect to Δ, and equated to zero, giving the equation $g_P + g_{PP}\Delta = 0$, or

$$g_{PP}\Delta = -g_P. \tag{D.4}$$

Here, g_{PP} is the Hessian matrix (matrix of second derivatives). Now if the function $g(\mathbf{P})$ is in fact the squared norm of the error function:

$$g(\mathbf{P}) = \frac{1}{2}||\epsilon(\mathbf{P})||^2 = \epsilon(\mathbf{P})^T \epsilon(\mathbf{P})/2, \tag{D.5}$$

with $\epsilon(\mathbf{P}) = f(\mathbf{P}) - \mathbf{X}$, then the cost function matches the type of least squares minimization we are considering.

The gradient vector $g_P = \epsilon_P^T \epsilon_P$, but if we note that $\epsilon_P = f_P = J$, then $g_{\mathbf{P}} = J^T \epsilon$. Now, if g_P is differentiated again, one obtains

$$g_{PP} = \epsilon_P^T \epsilon_P + \epsilon_{PP}^T \epsilon. \tag{D.6}$$

The second half ($\epsilon_P^T P \epsilon$) vanishes since it is assumed that $f(\mathbf{P})$ is linear.

$$g_{PP} = \epsilon_P^T \epsilon_P = J^T J. \tag{D.7}$$

Finally, substituting the gradient and the Hessian in Eqn. D.4, then $J^T J \Delta = -J^T \epsilon$, which is the normal equation. When the Hessian is approximated with $J^T J$, the method is known as the Gauss-Newton method.

D.4   The Levenberg-Marquardt algorithm

This method is an extension of the Gauss-Newton method, in which the normal equations are replaced by augmented normal equations. Instead of $J^T J \Delta = -J^T \epsilon$, then the augmented equations

$$(J^T J + \lambda I)\Delta = -J^T \epsilon. \tag{D.8}$$

The value of λ can vary with the iterations of the algorithm, and is often increased steadily if proposal values for P_{i+1} are worse than the best previous minimum. The values are therefore varied to escape local minima. The LM method in operation varies, depending on the value of λ, from behaving like the Gauss-Newton method (for small λ), and the gradient descent method (for large λ), and in performance is generally more efficient than either one of them.

Appendix E

Similarity results for posterior distributions on parametric BRDFs

This appendix contains a subset of classification results consisting of matrices of similarities between sample sets for each of the 48 different materials. The tables include results for the AGMMP and other pdf similarity measures, including the Earth Mover's distance and the correlation measure for samples in their original positions evenly spaced in the BRDF parameter space.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1000	997.7	999.8	993.9	995.4	996.6	993.9	999.2	999.5	999.7	992.5	996.2	987.6	980.7	982.8	983.4
2		1000	997	998.6	993.3	999.2	992.4	999.8	997.1	995.9	998.3	993.2	997.9	994.8	995.7	994.8
3			1000	992.8	996.8	996	995.2	998.9	998.9	999.8	991.3	997.7	987.2	980.7	982.6	982.8
4				1000	962.5	988.4	960.6	999.6	995.1	990.6	999.9	972.6	999.1	998	998.4	997.8
5					1000	988	999.4	995.5	986.2	993.1	956.7	999.2	973	958.2	957.8	954.4
6						1000	985.7	998.3	995.5	995.1	985.9	990	994.3	988.2	987.8	984.7
7							1000	994	983.7	991.1	955.6	997.6	971.1	957.6	956.9	953.1
8								1000	999.1	998.8	999.6	996.8	999.9	999.6	999.7	999.5
9									1000	998.9	993.7	989.5	990.1	983.4	985.6	985.6
10										1000	988.6	994.9	983.6	974.8	977	978.5
11											1000	968.6	999.3	998.7	999	998.5
12												1000	975	961.5	962.5	961.8
13													1000	998.8	999.3	998.5
14														1000	999.9	999.9
15															1000	999.7
16																1000
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																

Table E.1: Columns 1 to 16 of similarity matrix for parametric BRDF surfaces of real materials, using similarities derived from the Mahalanobis distance, with PMC developed sample sets, calculated over 20 iterations. (The data in this similarity matrix has been transformed linearly for better visibility; the ordering for similarities between material pairs is preserved). These similarity values are derived from probabilistic distance measures on parametric BRDF distributions corresponding to the target distributions of $ePL_3(\mathbf{X}|\theta)$ (Eqn. 6.62).

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	956.1	948	959.7	940	968.5	971.7	929.9	967.2	979	994.9	988.9	989.7	979.5	972.4	979.6	984.2
2	961.2	956.6	964.8	949.9	982.7	986.1	963.8	983.6	991.7	999.5	996.1	997.9	990.5	990.3	992.7	995.5
3	962.7	956.6	966.8	945	968	971.9	936	967.7	979.2	994	987.8	988.9	980.3	973.4	979.4	983.7
4	835.1	800.5	748.3	716.9	987.7	990.5	978.6	989.3	993.9	999.3	997.5	999.1	989.3	995.3	996.4	998
5	961.2	953.9	961.2	921.2	936.9	961.8	907.5	952.1	972.8	990.7	987.6	983.8	977.4	960.2	974	975.2
6	927.6	905.8	930.2	842.4	971.8	987.3	955.9	981.8	994.3	999.2	998.7	997.6	996.5	988.4	994	994.3
7	975.6	972.5	978.4	950.3	936.5	960.8	915.6	951.5	971.9	990.1	987.8	982.8	976.7	959.7	973.3	973.9
8	970.5	970.6	974.8	964.5	996.7	998.4	996.2	998.1	999.3	1000	999.7	999.9	998.8	999.4	999.5	999.7
9	904.1	883.3	896.3	866.6	971	974.6	934.9	970.8	981.3	994.8	988.3	990.8	979.9	976.2	982.2	986.3
10	926.7	911.1	926.5	903.3	963.3	967.6	916.5	962.2	975.4	992.9	986.9	987	974.5	967.7	976.6	981.4
11	823.2	782.7	712.6	674.7	989.4	992.2	981.9	991.2	995	999.3	998.3	999.4	989.7	996.6	997.5	998.8
12	946.3	934.6	944.3	910.8	945.4	962.6	908	954.3	972.8	990.3	986.2	983.6	976.8	961	973.3	975.5
13	895.8	869	889.4	821.7	988.2	992	972.5	990.2	996.5	999.6	999	1000	994.5	996.1	997.6	999.2
14	849.1	809.5	811.1	753.6	993.1	996.6	985.2	995.5	998.9	997.6	999.8	999.1	996.8	999.5	999.8	1000
15	852	809.9	814.6	729.5	992.3	995.9	983.6	994.7	998.5	998.2	999.8	999.5	995.8	999	999.5	1000
16	838.6	801.6	780.1	717	994.6	997.3	990.1	996.6	998.5	997.4	999.6	998.9	994.8	999.8	999.9	999.8
17	1000	999.7	998.5	998.2	813.6	884.7	768.1	859.6	905	955.2	952.8	935.3	923.8	861.8	908.7	902.1
18		1000	998.3	997.7	776.7	860.6	687	828.7	883.7	950.2	948.1	925	904	830.2	892.4	886.5
19			1000	990.8	735	878.4	647.9	839.5	903.4	958.9	956.2	938.6	928.5	839.1	908.8	900
20				1000	680	814.7	545.1	767.1	848.6	944.7	943.9	913	847.2	792.1	876.3	871.7
21					1000	998.3	1000	999	993.8	986.4	991.9	988.9	990.1	994.6	994	992.1
22						1000	997	999.8	998	989.8	995.6	992.5	996.8	997.7	997.5	995.5
23							1000	998.3	987	970.4	981.8	974.4	981.8	987	986.5	982.3
24								1000	996.7	987.7	994.1	990.6	994.6	996.8	996.4	994.2
25									1000	994.9	999.1	997	999.6	999.3	999.7	998.6
26										1000	998.2	999.5	993.7	994.3	996	998
27											1000	999.3	998.4	999	999.6	999.8
28												1000	995.6	996.8	998	999.4
29													1000	997.5	998.8	997
30														1000	999.9	999.1
31															1000	999.6
32																1000
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																

Table E.2: Columns 17 to 32 of similarity matrix for parametric BRDF surfaces of real materials, using similarities derived from the Mahalanobis distance, with PMC developed sample sets, calculated over 20 iterations. (The data in this similarity matrix has been transformed linearly for better visibility; the ordering for similarities between material pairs is preserved). These similarity values are derived from probabilistic distance measures on parametric BRDF distributions corresponding to the target distributions of $ePL_3(\mathbf{X}|\theta)$ (Eqn. 6.62).

	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
1	972	975.7	978.8	985.4	996.3	999.1	999.3	997.9	962.8	957.4	957.2	953.8	945.9	978.6	975.4	980.6
2	986.9	988.4	989.2	993.3	994.1	995.4	996.4	995	964.9	959.8	959.5	959.8	953.8	979.4	975.9	981.3
3	973	976.2	978.6	985	997.9	999.7	999.9	999	961.9	959.2	958	958.6	954.1	980.5	977.7	979.6
4	989.3	990.8	993.1	994.6	975.7	986.7	988.4	982.4	580.9	135.8	244.9	768.9	815.1	891.1	906.2	752.3
5	968.4	971.7	978.9	983.8	999.4	995.5	996.1	997.8	917.1	884.5	883.7	937.4	953.3	982.1	983.4	975.4
6	992.6	993.3	995.7	997.9	990.6	994.1	995.2	992.9	879.6	861.1	855.1	903	892.6	947.6	943.6	938.3
7	967.9	970.9	979.1	983.8	998	993.6	994.1	996.1	945.7	927.9	927.1	955.7	971.8	989.7	989.8	984.7
8	998.8	998.9	999.2	999.3	996.8	998.4	998.5	997.7	953.2	944.5	944.8	958.7	972.7	982.5	982.2	974
9	975	977.5	978.8	984.3	990.2	997	997.4	994.1	881.2	821.4	829.5	875.6	895.1	951.2	951.4	945.5
10	968.1	971.6	975.3	981.8	995	999.5	999.5	997.7	921.2	885.6	889.6	907.3	919.6	969.2	968.1	970.4
11	990.7	992.2	994.8	995.8	972.4	984.3	986.3	979.8	453.3	67.78	64.88	762.4	794	876.8	895.1	652.3
12	967.8	971.3	977.3	982.6	1000	997.3	997.8	999.3	922.5	896	896.2	927.3	940	977.7	979	973.6
13	992.6	993.7	994.7	997	975.9	980.6	983.5	980.2	823.7	802.7	794.7	877.8	863.4	925.5	916.1	886.5
14	996.9	997.5	998.6	999	963.4	970	974.4	969.2	746.9	632.7	645.8	816.6	817.3	898.4	895.3	844.7
15	996.1	996.8	998.1	998.7	964.5	972.2	976.2	971	678.1	613.7	596.7	826.2	809	891.8	887.6	791
16	996.3	997.2	999	998.9	964.2	973.9	977	971.1	576.5	410.6	400.2	801.3	807.3	883.4	889.3	731.8
17	896	910.1	935.8	947.2	959.5	936.8	950.1	945.4	991	984.4	984.6	994.2	999.5	996.6	997	996.1
18	869.6	889.4	925.1	938.6	950.6	921.7	938.2	934.1	991.9	985.2	985.3	994.7	999.9	996.8	997.7	997.4
19	892.5	911.8	939.2	951	963.2	937.5	953.6	947.1	970.4	933.7	936.6	985.5	998.4	998.6	998.4	995.1
20	821.7	845.9	909.4	922.3	929.7	910.3	921.4	923.3	994.2	986.5	986.5	997.3	997.1	988	990.9	993.8
21	994.4	995.2	995.4	993.7	948.2	958.5	961.7	955.6	427.2	148.5	114.9	771	782.3	859.1	869.2	629.4
22	998.9	999.1	998.8	997.6	963.8	965.8	969.4	967.4	795	765.1	756	865.7	856.1	913.6	903.1	867.5
23	989.4	990.9	989.1	986.6	913.7	910	923.7	919.9	371.4	109.2	74.23	736.7	686.4	819	808.9	579.3
24	997.7	998.1	997.8	996.3	955.6	959.3	963.5	960.5	707.4	656.6	640.5	836.5	826.2	893.6	884.3	810.7
25	999.4	999.5	999.8	999.9	974	974.5	978	976.6	857.2	831.6	827.9	889.5	877.8	932.8	922.3	914
26	990.8	991.9	992.7	995.9	991.1	992.4	993.5	992.2	960.1	951.9	952.3	952.8	947.6	975.9	971.8	978.7
27	996.9	997.3	998.1	999.3	987.3	986.9	988.4	988.3	959	949.7	950.4	950.6	944.6	974.1	969.6	978.1
28	993.3	994.3	995.1	997.5	984.5	986	987.9	986.4	933.2	921.6	921.6	929	921.5	959.7	952.3	959
29	999.3	999.3	999.5	999.8	979.8	976.3	981.1	980.1	829.8	812.7	800.2	899.8	884.6	938.5	918.1	894.4
30	997.8	998.3	999.5	999.4	962.2	964.3	969.2	966.5	784.8	682.7	696.7	830.3	837.2	908.9	902.6	873.9
31	998.4	998.7	999.4	999.7	974.1	975	977.9	976.7	896	866.3	869	894.6	892	941.6	932.4	936.4
32	996.1	996.7	997.8	998.8	975.9	979	981.3	979.2	876.1	811.9	823	874.2	892.2	941.4	936.1	934.1
33	1000	1000	999.7	999.1	969.5	967.8	972.4	971.9	819	783.4	776.7	877.5	860.1	923	910.1	891.4
34		1000	999.8	999.3	973.3	971.8	975.9	975	845.3	825.4	818.3	891.7	877.7	931.1	916.9	902.7
35			1000	999.6	979.8	976.7	980.1	980.1	932.7	927.5	925.9	932	916.3	955	943.3	952.3
36				1000	985.4	982.9	986	985.2	942.9	940.6	938.4	943.4	929.3	962.8	952.7	960.3
37					1000	997.2	997.8	999.3	952	946.7	945.3	947.5	950.8	980.9	980.4	978.5
38						1000	999.8	999.2	936.3	918.8	919.5	925.4	927.1	972.7	971.9	973.1
39							1000	999.4	948.8	943.1	942	942.3	936.7	974	971.8	973.7
40								1000	946.2	931	931.7	933.9	940.5	979.1	978.8	979.5
41									1000	997.5	996.8	999.6	987.9	976.1	986.5	965
42										1000	999.8	999.7	978.2	962.3	980.3	924.9
43											1000	999.5	978	961.7	979.7	915.8
44												1000	993.8	985.8	990.5	992.1
45													1000	996.7	997.5	997.1
46														1000	999.8	997.5
47															1000	998.9
48																1000

Table E.3: Columns 33 to 48 of similarity matrix for parametric BRDF surfaces of real materials, using similarities derived from the Mahalanobis distance, with PMC developed sample sets, calculated over 20 iterations. (The data in this similarity matrix has been transformed linearly for better visibility; the ordering for similarities between material pairs is preserved). These similarity values are derived from probabilistic distance measures on parametric BRDF distributions corresponding to the target distributions of $ePL_3(\mathbf{X}|\theta)$ (Eqn. 6.62).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	999.6	993.5	996.4	929.9	994.7	993.2	993.2	990.9	987.2	995.6	908.8	995.4	967.3	941.4	944.9	921.6
2		999.7	994.2	926.6	994.7	991.8	993.4	990.3	981	990.8	906.5	995	969.5	947.5	949.7	926.9
3			999.6	913.4	995	993.1	993.7	987.5	984	994.5	888.8	995.2	958.1	923.8	928.1	902.7
4				999.8	996.4	989.5	987.3	999.2	999.1	998.7	999.4	996.4	998.4	996.7	997.4	995.4
5					999.6	963.4	992.4	945.5	918.2	959.2	800.1	989.3	862.8	821	828.5	800.9
6						999.6	993	977.3	956.9	982.6	836.4	987.8	919.8	880.7	886.6	860.8
7							999.6	928.9	888.8	938.1	787.4	981.7	843.4	811	819.9	791.5
8								999.6	993.7	996.7	969.9	995.1	992.3	986.8	987.4	976.2
9									999.7	998.4	980.3	996.1	987.8	978.2	980	970.5
10										999.6	937	995.5	970.4	949.9	953	935.7
11											999.8	996.8	998.9	997.8	998.3	996.8
12												999.6	922.1	875.1	883.3	857.1
13													999.7	997.4	997.5	994.5
14														999.7	999.1	998.2
15															999.7	998.8
16																999.8
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																

Table E.4: Columns 1 to 16 of similarity matrix for parametric BRDF surfaces of real materials, using the AG-MMP measure with $\sigma = 0.05$, with PMC developed sample sets, calculated over 20 iterations. (The data in this similarity matrix has been transformed linearly for better visibility; the ordering for similarities between material pairs is preserved). These similarity values are derived from probabilistic distance measures on parametric BRDF distributions corresponding to the target distributions of $ePL_3(\mathbf{X}|\theta)$ (Eqn. 6.62).

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	983.1	962.5	983.1	751	832.9	938.4	834	914.7	971.5	992.5	987.3	985.6	962.1	958.6	971.4	964.7
2	981	964.4	981.7	740.2	866.8	956.6	868.2	936.8	980.2	992.9	988.3	985.6	979.4	965.1	976.7	966.9
3	985.9	970.6	982.9	795.5	802.9	926.8	805.5	899	968.1	991.3	985.4	981.4	961.1	951.2	965.7	954.5
4	933.1	893	957.9	334.7	954.9	973.6	952.7	971.1	982.8	998.6	994.5	998.3	967.1	991.4	992.8	996.8
5	982.1	977.3	983.2	886.3	655.7	811.1	656.7	776.8	889.6	951.5	944.9	928	894.9	851.5	879.2	866.4
6	972.7	976.7	985.4	870.1	764.2	905	771	867.6	964.4	989.7	987.7	969.5	974	925.5	947.6	923.6
7	978.7	980.1	985.4	914.7	664.6	796.9	664	772.8	868.3	934.4	927	912.7	874.4	833.6	860.3	852
8	971.3	940	975.5	573.2	940.7	979.1	936.8	971.8	987.5	997	994.1	995.7	979.9	988	991.9	991.2
9	968.6	931.7	974.7	523.6	901.6	943.8	898.2	936.7	959.5	992.3	989	987.3	940.4	970.1	975.1	980.9
10	979.7	953.9	980.8	665.4	844.7	928.2	843.4	912.1	960.5	991.6	985.4	983.2	945.4	957.9	967	965.1
11	928.5	887.6	955.7	313.8	960.1	977.2	958.1	975	985.8	999	996	998.9	971.2	993.4	994.6	997.9
12	987.3	977.3	986	851.2	724.2	869.3	730.8	840.2	930.9	972.7	967.2	956.3	923.3	906.9	927.4	912.2
13	965.1	932.6	973.8	500.1	971.7	990	970	987.5	994.8	998.8	998.1	998.8	986.5	996.8	998.1	997.5
14	942.8	907.9	963.5	395.1	983.1	993.9	981.2	992.6	997	999.1	999.2	999.2	990.5	999	999.3	999.3
15	947.5	913.3	966.4	407.2	981.4	992.8	979.8	991.6	996.5	999.1	999.1	999.4	989.3	998.9	999.1	999.2
16	930.5	893.7	956.7	341.1	985.2	994.5	983.9	993.5	997.5	999.1	999.3	999.4	991.4	999.4	999.5	999.5
17	999.6	991.1	990.4	944.4	610	662.2	607.4	643.9	716.2	793.9	805.1	706	731.6	642.5	693.7	654.5
18		999.6	991.6	976.3	685.7	712	680.2	700.7	749.5	810	818.7	741.9	755.7	689.7	732.7	698.2
19			999.6	954.1	715.7	778.4	709.2	764.2	825.7	889.3	893.9	847.2	827.1	782	815.8	800.9
20				999.6	828.7	798.8	820.7	798.1	798.8	831.5	846.8	768.4	800.6	747.9	783	750.1
21					999.8	998.7	999.6	998.8	997.6	997.8	998.2	992.2	998.3	991.4	995.2	989
22						999.7	993.1	998.2	998.8	998.2	998.4	997	998.5	996.5	998.2	995
23							999.8	999.1	997.9	998.4	998.6	992.2	998.6	991.8	995.7	989.2
24								999.7	999	998.3	998.9	997.2	998.7	997.2	998.3	995.5
25									999.7	997.6	995.3	995.5	997.7	991.6	996.1	989.9
26										999.6	994.8	994.2	984.2	981.9	989	983.9
27											999.7	992.3	990.5	983.9	989.5	982.9
28												999.7	986.5	993.3	995.3	995.6
29													999.7	979.9	990	972.6
30														999.8	999.3	998.8
31															999.7	997.7
32																999.7
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																

Table E.5: Columns 17 to 32 of similarity matrix for parametric BRDF surfaces of real materials, continued from Table E.4.

	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
1	956.5	961.8	972.5	986.4	996.3	996	996.3	996.2	231.1	169.8	201.9	176.2	842.7	928.8	919.3	378.3
2	973.9	976	982.1	990.8	994.5	992.7	995.1	995.3	212.5	156.1	181.5	164.2	838.7	934.4	921.6	363.6
3	952.9	958.3	970.8	985	995.7	995.5	995.8	995.6	290.1	216.5	260.3	231.3	871	942.7	927.4	446.8
4	971.1	973.7	984.8	992.5	991.3	999.1	997.3	998.2	24.29	27.92	13.62	22.85	568	789.2	785.2	31.23
5	861.4	873	899	938	986	969.2	978.6	984.4	564.5	496.4	525.7	475.1	928.7	961.3	948.2	666.5
6	953.5	961.4	968.6	985.9	983.3	980	983.1	988.5	465.3	396.7	435.6	396.1	918.3	956.6	946.2	601.6
7	836.6	851.3	879.4	916.8	976	951.8	966.3	975.2	639.1	581.9	612.4	557	948.9	971.7	960.7	731.4
8	980.8	982.9	989.8	993.5	994.8	997	996.9	997.1	65.91	30.34	48.2	46.13	724.9	881.5	873	142.9
9	943.1	947.7	963.3	990.4	995	998.1	997.6	998.1	21.63	2.934	10.94	18.33	690.8	865.9	855.4	100.2
10	943	948.4	962.8	986	995.9	997.3	996.9	996.8	151.4	104.6	134.1	120	788.2	906.1	895.6	270.9
11	975	977.3	987.5	992.9	990.5	999.1	997.2	998.2	27.75	2.819	16.25	25.58	554.8	781.8	776.1	35.46
12	907.5	915.6	934	966.7	993.4	987.5	990.8	993.4	415.6	352.7	389	356.1	911.1	961.2	945.4	573.3
13	988.9	990.3	995.7	996.6	995.3	998.6	997.9	998.2	35.84	4.576	24.12	31.85	685.7	865.5	860.5	70.24
14	992.7	993.8	997.9	998.3	993.3	998.8	997	998.2	39.54	8.406	29.93	38.71	610.2	815.7	814.3	31.62
15	991.5	992.7	997.3	997.8	993.9	998.9	997.4	998.3	39.42	7.597	29.62	37.28	623.5	825.4	822.6	25.82
16	993.4	994.5	998.3	998.4	991.3	999	996.9	998	41.7	9.778	31.86	39.5	573	792.7	789.1	33.06
17	698.7	717	721	769.6	725.4	692.4	730.6	750.7	739.7	699.1	719.6	700.9	972.1	979	974.2	828.8
18	727.5	744.3	754.1	784.1	746.4	712.9	748	768.8	868.2	826.1	853.6	835.8	987.3	984.6	983.4	913.2
19	798.9	815.5	831	877.1	873.9	843.3	869.6	885.2	784.4	743.5	768.5	737.2	974.5	985.8	981.6	858.9
20	792.6	801.7	800.3	809.5	739.2	705.5	746.2	766	978.6	942.9	968.5	970.7	994	992.6	993.5	986.5
21	998.5	998.4	998	998	970.3	976.6	979	988.7	44.32	38.35	43.2	44.15	539.9	774.3	771.7	40.7
22	999	998.9	998.9	998.9	986.3	989.8	988.4	994.3	43.12	27.93	38.02	40.04	659.4	848.8	846.5	53.64
23	998.8	998.7	998.5	998.4	970.7	977	978.9	989.5	44.14	38.47	43.24	44.22	545.8	780.6	775.4	39.65
24	999.2	999.2	999.2	999.1	984.4	989	987.6	994.5	43.33	29.74	42.5	43.11	610.8	818.2	815.9	30.27
25	997.9	998.1	997.5	998.3	990.8	992.9	992.1	994.5	81.19	53.28	73.7	66.66	752	899.1	895.5	192
26	982.3	985	990.1	994.5	994.8	995.6	996	996.2	123.3	72.06	102.1	90.2	773.9	900.5	888.5	231.1
27	988.7	990	993.3	996.4	991.6	993.7	993.6	995.8	128.5	88.02	105.9	99.27	716.6	869	853.8	189.8
28	987.4	989.1	994.5	996.3	995.1	997.6	997.3	997.5	61.57	26.74	49.32	48.31	690.8	860.7	853.7	92.98
29	997	996.6	996	998.1	986	985.7	986.4	993.4	194.7	161.2	181.1	169.5	848.7	944.1	941	367.4
30	996.7	997.2	999.1	998.9	991.9	998.3	996	997.3	46.01	15.51	36.22	42.73	605.4	813.5	809.7	39.32
31	995.6	996.4	998.6	998.7	992.4	997.2	995.9	997.2	43.74	15.74	38.94	42.2	640.7	835.8	829.1	70.32
32	991.7	992.8	997.5	997.9	993.4	998.4	996.9	998.1	42.02	11.54	29.8	38.17	619.9	820.4	814.8	48.45
33	999.7	998.2	998	998.6	986	986.7	986.9	993.9	111.4	87.48	102	97.27	775.3	911.1	907.1	233.1
34		999.7	997.9	998.5	987	988.5	988.3	993.9	78.15	56.28	70.87	65.17	759.6	904.4	899.4	205.8
35			999.7	998.4	989.6	992.7	992	995.3	68.2	40.4	58.18	56.54	699	867.2	861	131.5
36				999.7	989.6	989.8	990.2	994.4	143.5	107.8	117.7	112	789.5	913.4	905.6	267.2
37					999.6	990.8	992.3	993.6	427	348.9	401.1	349.3	920.5	955.6	944.6	582.8
38						999.6	995.3	996.1	303.7	249.2	285.2	270.4	831.3	920	910.1	415.8
39							999.6	996	245.7	178.5	219	189.3	863.6	941.1	926.5	411.5
40								999.6	290.1	230.9	263.7	228.8	859.9	934.7	922.2	433.9
41									999.7	989.6	993.2	997.1	992.5	989.4	990.8	985.7
42										999.8	998.8	998.2	994.9	980.2	982.6	958.6
43											999.8	998.2	994.1	980	981.5	957.7
44												999.7	993.2	982.8	986.1	975.7
45													999.7	991.4	991	971.5
46														999.6	991.9	946.5
47															999.6	969.5
48																999.6

Table E.6: Columns 33 to 48 of similarity matrix for parametric BRDF surfaces of real materials, continued from Table E.5.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	0.6802	0.9668	0.4056	0.2939	0.4943	0.1619	0.5014	0.609	0.8714	0.1117	0.4266	0.04584	-0.03106	-0.03631	-0.04317
2		1	0.5658	0.5587	0.1644	0.5119	0.05721	0.5024	0.7217	0.6674	0.3497	0.2458	0.3397	0.06643	0.07705	0.06542
3			1	0.3408	0.2798	0.4371	0.1467	0.4608	0.4617	0.8333	0.04158	0.4513	-0.003521	-0.05843	-0.06132	-0.06557
4				1	0.05527	0.8777	-0.0452	0.8133	0.6572	0.3998	0.9398	0.1344	0.7405	0.7598	0.7546	0.7833
5					1	0.1979	0.8805	0.4188	0.09188	0.1857	-0.01895	0.6842	0.05047	-0.005583	0.002381	-0.01579
6						1	0.03585	0.9064	0.7522	0.4774	0.7985	0.3368	0.7139	0.7429	0.7329	0.7065
7							1	0.178	-0.01762	0.06712	-0.08759	0.3853	-0.01723	-0.06519	-0.05529	-0.0716
8								1	0.6191	0.4507	0.731	0.5925	0.5952	0.6484	0.6334	0.643
9									1	0.6609	0.5009	0.1535	0.5237	0.3354	0.3436	0.304
10										1	0.1469	0.2899	0.1748	0.0411	0.05074	0.03548
11											1	0.01683	0.7626	0.8748	0.8601	0.9058
12												1	0.08971	0.02468	0.03244	0.01511
13													1	0.7472	0.8293	0.7166
14														1	0.9738	0.9647
15															1	0.9426
16																1
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																

Table E.7: Columns 1 to 16 of similarity matrix for parametric BRDF surfaces of real materials, using the correlation measure ("corr", Eqn. 8.9), with all samples in the sample set evenly spaced over the range ($\rho_d \in [0, 1], \rho_s \in [0, 1], \alpha \in [0, 0.5]$) in parameter space. These similarity values are derived from probabilistic distance measures on parametric BRDF distributions corresponding to the target distributions of $ePL_3(\mathbf{X}|\theta)$ (Eqn. 6.62).

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	-0.123	-0.139	-0.087	-0.138	-0.080	-0.074	-0.076	-0.075	-0.042	0.230	0.080	0.060	-0.007	-0.008	-0.016	0.071
2	-0.096	-0.111	-0.070	-0.116	-0.082	-0.078	-0.072	-0.079	-0.039	0.754	0.250	0.316	-0.007	0.081	0.021	0.184
3	-0.112	-0.125	-0.077	-0.126	-0.083	-0.079	-0.078	-0.079	-0.055	0.151	0.049	0.029	-0.0151	-0.029	-0.034	0.045
4	-0.199	-0.211	-0.150	-0.195	0.444	0.472	0.443	0.4541	0.554	0.578	0.497	0.5495	0.322	0.663	0.568	0.623
5	-0.221	-0.242	-0.147	-0.244	-0.0303	-0.021	-0.043	-0.022	0.009	0.088	0.058	0.048	0.02389	0.027	0.027	0.095
6	-0.228	-0.250	-0.163	-0.242	0.408	0.451	0.365	0.4306	0.595	0.473	0.476	0.480	0.342	0.660	0.609	0.627
7	-0.189	-0.212	-0.116	-0.223	-0.065	-0.058	-0.074	-0.0584	-0.035	0.014	0.011	-0.009	-0.002	-0.027	-0.020	0.034
8	-0.278	-0.294	-0.207	-0.272	0.405	0.433	0.390	0.414	0.514	0.447	0.407	0.412	0.296	0.573	0.502	0.535
9	-0.144	-0.161	-0.100	-0.161	0.114	0.138	0.08378	0.132	0.246	0.541	0.387	0.401	0.152	0.3547	0.310	0.4161
10	-0.106	-0.121	-0.066	-0.131	-0.032	-0.016	-0.039	-0.021	0.041	0.325	0.180	0.178	0.064	0.101	0.078	0.1738
11	-0.181	-0.192	-0.136	-0.177	0.577	0.602	0.572	0.5828	0.667	0.4941	0.487	0.533	0.377	0.741	0.646	0.6388
12	-0.170	-0.188	-0.114	-0.188	-0.005	0.002	-0.020	0.001	0.034	0.1304	0.086	0.0800	0.038	0.057	0.053	0.1239
13	-0.121	-0.140	-0.080	-0.153	0.175	0.235	0.152	0.205	0.581	0.675	0.780	0.85	0.417	0.821	0.759	0.891
14	-0.150	-0.169	-0.103	-0.172	0.577	0.673	0.527	0.625	0.872	0.260	0.506	0.469	0.533	0.885	0.842	0.728
15	-0.142	-0.160	-0.097	-0.165	0.474	0.556	0.432	0.510	0.86	0.329	0.611	0.586	0.582	0.928	0.894	0.821
16	-0.159	-0.176	-0.112	-0.174	0.679	0.734	0.654	0.704	0.852	0.268	0.499	0.466	0.513	0.864	0.796	0.694
17	1	0.964	0.728	0.777	-0.125	-0.1206	-0.129	-0.118	-0.116	-0.076	-0.073	-0.083	-0.066	-0.1231	-0.104	-0.1135
18		1	0.778	0.708	-0.140	-0.136	-0.141	-0.135	-0.135	-0.089	-0.08	-0.097	-0.079	-0.143	-0.122	-0.132
19			1	0.189	-0.079	-0.074	-0.090	-0.072	-0.074	-0.053	-0.04502	-0.064	-0.047	-0.083	-0.067	-0.072
20				1	-0.152	-0.150	-0.143	-0.149	-0.145	-0.101	-0.099	-0.1018	-0.079	-0.151	-0.133	-0.146
21					1	0.921	0.946	0.947	0.477	-0.060	-0.033	-0.046	0.093	0.339	0.241	0.089
22						1	0.817	0.981	0.603	-0.055	-0.021	-0.032	0.170	0.433	0.340	0.154
23							1	0.849	0.421	-0.042	0.011	-0.008	0.120	0.324	0.214	0.096
24								1	0.545	-0.057	-0.030	-0.039	0.104	0.3859	0.279	0.117
25									1	0.047	0.440	0.283	0.760	0.8774	0.909	0.664
26										1	0.55	0.739	0.040	0.341	0.2076	0.478
27											1	0.876	0.305	0.774	0.698	0.909
28												1	0.221	0.678	0.544	0.839
29													1	0.660	0.756	0.514
30														1	0.955	0.918
31															1	0.878
32																1
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																

Table E.8: Columns 17 to 32 of similarity matrix for parametric BRDF surfaces of real materials, continued from Table E.7.

	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
1	-0.026	-0.033	0.151	-0.002	0.538	0.876	0.848	0.564	-0.107	-0.038	-0.060	-0.124	-0.146	-0.163	-0.197	-0.075
2	-0.031	-0.039	0.207	0.043	0.277	0.511	0.533	0.264	-0.095	-0.042	-0.051	-0.105	-0.123	-0.161	-0.173	-0.082
3	-0.030	-0.039	0.124	-0.018	0.578	0.943	0.887	0.644	-0.097	-0.033	-0.056	-0.113	-0.133	-0.156	-0.181	-0.063
4	0.367	0.422	0.567	0.517	0.119	0.303	0.320	0.110	-0.156	-0.074	-0.086	-0.192	-0.200	-0.285	-0.284	-0.24
5	0.0102	0.011	0.074	0.03403	0.7217	0.3079	0.5629	0.723	-0.1986	-0.09551	-0.1104	-0.2301	-0.243	-0.036	-0.278	-0.045
6	0.366	0.431	0.585	0.545	0.3128	0.4164	0.4939	0.2841	-0.1962	-0.09324	-0.1083	-0.2262	-0.2464	-0.3273	-0.348	-0.264
7	-0.017	-0.021	0.01	-0.008	0.4878	0.1588	0.3838	0.5763	-0.1828	-0.08728	-0.1017	-0.201	-0.2204	0.072	-0.206	0.038
8	0.331	0.3838	0.520	0.450	0.552	0.4686	0.5973	0.4953	-0.2186	-0.1079	-0.1211	-0.2735	-0.2753	-0.3115	-0.375	-0.270
9	0.130	0.1617	0.3644	0.295	0.134	0.385	0.439	0.085	-0.131	-0.061	-0.072	-0.143	-0.164	-0.231	-0.235	-0.175
10	0.041	0.042	0.231	0.099	0.372	0.859	0.781	0.371	-0.107	-0.043	-0.059	-0.105	-0.136	-0.179	-0.192	-0.093
11	0.434	0.505	0.581	0.576	-0.046	0.021	0.055	-0.075	-0.142	-0.070	-0.078	-0.174	-0.180	-0.265	-0.25	-0.246
12	0.026	0.0303	0.122	0.01	0.944	0.486	0.685	0.804	-0.153	-0.073	-0.084	-0.173	-0.19	-0.162	-0.257	-0.108
13	0.342	0.403	0.616	0.745	0.013	0.043	0.122	-0.017	-0.125	-0.045	-0.067	-0.110	-0.159	-0.224	-0.228	-0.172
14	0.613	0.697	0.713	0.754	-0.059	-0.039	-0.003	-0.085	-0.141	-0.063	-0.077	-0.138	-0.177	-0.256	-0.254	-0.225
15	0.582	0.667	0.742	0.824	-0.047	-0.032	0.009	-0.074	-0.135	-0.057	-0.074	-0.131	-0.170	-0.246	-0.244	-0.210
16	0.580	0.668	0.673	0.718	-0.064	-0.043	-0.013	-0.089	-0.141	-0.062	-0.077	-0.147	-0.18	-0.261	-0.254	-0.233
17	-0.076	-0.086	-0.108	-0.089	-0.167	-0.1216	-0.176	-0.184	-0.1286	-0.116	-0.144	0.141	0.554	-0.049	0.132	0.253
18	-0.088	-0.100	-0.126	-0.106	-0.184	-0.1351	-0.196	-0.201	-0.122	-0.110	-0.139	0.154	0.674	-0.032	0.181	0.305
19	-0.051	-0.055	-0.069	-0.055	-0.109	-0.08036	-0.114	-0.120	-0.217	-0.115	-0.126	-0.087	0.738	0.182	0.3534	0.560
20	-0.089	-0.106	-0.135	-0.117	-0.191	-0.1393	-0.204	-0.205	0.092	-0.056	-0.085	0.337	0.208	-0.260	-0.1087	-0.145
21	0.316	0.392	0.212	0.143	-0.080	-0.071	-0.062	-0.098	-0.126	-0.050	-0.069	-0.106	-0.158	-0.229	-0.227	-0.201
22	0.546	0.608	0.362	0.251	-0.069	-0.062	-0.046	-0.088	-0.124	-0.049	-0.066	-0.100	-0.157	-0.226	-0.225	-0.191
23	0.228	0.277	0.174	0.133	-0.085	-0.073	-0.073	-0.100	-0.116	-0.053	-0.064	-0.112	-0.148	-0.215	-0.212	-0.201
24	0.419	0.503	0.254	0.174	-0.071	-0.064	-0.050	-0.090	-0.124	-0.050	-0.067	-0.100	-0.155	-0.223	-0.223	-0.190
25	0.703	0.810	0.825	0.847	-0.034	-0.024	0.007	-0.057	-0.118	-0.045	-0.063	-0.101	-0.151	-0.214	-0.216	-0.177
26	-0.002	-0.003	0.251	0.228	0.106	0.174	0.237	0.075	-0.083	-0.035	-0.042	-0.095	-0.104	-0.143	-0.148	-0.083
27	0.090	0.134	0.432	0.696	0.054	0.082	0.134	0.029	-0.079	-0.032	-0.0408	-0.089	-0.100	-0.137	0.141	-0.087
28	0.082	0.101	0.356	0.546	0.045	0.072	0.131	0.020	-0.078	-0.021	-0.037	-0.086	-0.105	-0.146	-0.149	-0.105
29	0.628	0.705	0.830	0.777	0.007	0.016	0.041	-0.006	-0.062	-0.016	-0.0304	-0.061	-0.082	-0.113	-0.116	-0.091
30	0.555	0.641	0.780	0.904	-0.008	0.010	0.058	-0.035	-0.122	-0.046	-0.0649	-0.118	-0.156	-0.221	-0.223	-0.175
31	0.587	0.679	0.837	0.959	-0.009	0.002	0.046	-0.033	-0.107	-0.039	-0.056	-0.095	-0.138	-0.194	-0.196	-0.152
32	0.360	0.418	0.655	0.860	0.080	0.0897	0.157	0.056	-0.119	-0.045	-0.0630	-0.115	-0.148	-0.177	-0.195	-0.125
33	1	0.959	0.825	0.640	-0.011	-0.005	0.020	-0.0262	-0.071	-0.013	-0.034	-0.059	-0.09499	-0.135	-0.136	-0.108
34		1	0.849	0.699	-0.016	-0.010	0.018	-0.034	-0.085	-0.019	-0.042	-0.069	-0.112	-0.1598	-0.160	-0.125
35			1	0.874	0.088	0.150	0.190	0.069	-0.110	-0.039	-0.057	-0.105	-0.141	-0.193	-0.1996	-0.137
36				1	0.009	0.021	0.068	-0.013	-0.094	-0.029	-0.048	-0.083	-0.122	-0.171	-0.174	-0.1247
37					1	0.615	0.8	0.918	-0.154	-0.066	-0.086	-0.164	-0.195	-0.133	-0.252	-0.065
38						1	0.930	0.678	-0.113	-0.047	-0.062	-0.121	-0.145	-0.162	-0.196	-0.072
39							1	0.84	-0.167	-0.072	-0.092	-0.173	-0.210	-0.186	-0.278	-0.095
40								1	-0.165	-0.072	-0.091	-0.182	-0.209	-0.092	-0.249	-0.040
41									1	0.214	0.253	0.686	-0.144	-0.281	-0.238	-0.340
42										1	0.868	0.460	-0.087	-0.137	-0.127	-0.168
43											1	0.461	-0.101	-0.159	-0.143	-0.204
44												1	-0.056	-0.312	-0.229	-0.235
45													1	0.198	0.4683	0.444
46														1	0.900	0.843
47															1	0.846
48																1

Table E.9: Columns 33 to 48 of similarity matrix for parametric BRDF surfaces of real materials, continued from Table E.8.

Bibliography

[1] B. Adhikara and D. Joshi. Distance discrimination et resume exhaustif. *Publs. Inst. Statis.*, 5:49–55, 1936.

[2] Michael Ashikhmin and Peter Shirley. An Anisotropic Phong BRDF Model. *Journal of Graphics Tools*, 5:25–32, 2000.

[3] Michael Ashikhmin and Peter Shirley. An Anisotropic Phong Light Reflection Model. *Technical Report*, 2000.

[4] C. H. Bennett. Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics*, 22:245–268, 1976.

[5] C. Berzuini, N. G. Best, W. R. Gilks, and C. Larizza. Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods. *JASA*, 92:590–599, 1997.

[6] J. Besag. Spatial interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society B*, 36(2):192–236, 1974.

[7] J. Besag. Statistical Analysis of Non-Lattice Data. *The Statistician*, 24(3):179–195, 1975.

[8] J. Besag. Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrika*, 64(3):616–618, 1977.

[9] J. Besag. On the statistical analysis of dirty pictures. *Journal of Royal Statistical Society B*, 48:259–302, 1986.

[10] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.

[11] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.

[12] Neil Birkbeck, Dana Cobzas, Peter Sturm, and Martin Jagersand. Variational Shape and Reflectance Estimation Under Changing Light and Viewpoints. *European Conference on Computer Vision*, 2006.

[13] M. J. Black and A. Rangarajan. On the Unification of Line Processes, Outlier Rejection, and Robust Statistics with Applications in Early Vision. *Int. Journal of Computer Vision*, 19(1):57–92, 1996.

[14] A. Blake, M.A. Isard, and D. Reynard. Learning to track the visual motion of contours. *Artificial Intelligence 78:101-134*, 1995.

[15] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, Massachusetts, 1987.

[16] C. A. Bouman and K. Sauer. Maximum likelihood scale estimation for a class of Markov random fields. *In Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 5:537–540, 1994.

[17] B. Calder, L. Linnett, S. Clarke, and D. Carmichael. Improvements in MRF Parameter Estimation. *IEE Colloquium on Multiresolution Modelling and Analysis in Image Processing and Computer Vision*, 21:3/1–3/6, 1995.

[18] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. Adaptive Importance Sampling in General Mixture Classes. *Technical report, Institut National de Recherche en Informatique et Automatique*, 2007.

[19] O. Cappé, A. Guillin, J. Marin, and C. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

[20] Gilles Celeux, Jean-Michel Marin, and Christian P. Robert. Iterated importance sampling in missing data problems. *IEEE on Trans. Image Processing*, 4(1):19–33, 1995.

[21] H. Chernoff. A measure of asymptotic efficiency of tests for a hypothesis based on a sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[22] M. L. Comer and E. J. Delp. Parameter estimation and segmentation of noisy or textured images using the EM algorithm and MPM estimation. *In Proc. of IEEE Int. Conf. on Image Processing*, II:650–654, 1994.

[23] A. Crouzil, X. Descombes, and J.-D. Durou. A Multiresolution Approach for Shape from Shading Coupling Deterministic and Stochastic Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1416–1421, Nov 2003.

[24] H. Derin and W. S. Cole. Segmentation of textured images using Gibbs random fields. *Computer Vision, Graphics, Image Processing*, 35:72–98, 1984.

[25] H. Derin and H. Elliot. Modelling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1), 1987.

[26] R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. Minimum variance importance sampling via Population Monte Carlo. *CMAP, Ecole Polytechnique, Palaiseau*, 2005.

[27] A. Doucet, N. de Freitas, and N. Gordon. *Sequential MCMC in Practice*. Springer-Verlag, New York, 2001.

[28] J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *RAIRO Analyse Numerique*, 10:5–12, 1976.

[29] Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical Methods for Shape from Shading: A survey with Benchmarks. *CVIU*, 2004.

[30] David Eberley. Thin Plate Splines. *Technical Report, Geometric Tools Inc.*, 1996.

[31] B.S. Everitt and D.J. Hand. *Finite Mixture Distributions*. Chapman & Hall, 1981.

[32] Shaohua Fan, Yu-Chi Lai, and Stephen Chenney. Population Monte Carlo Samplers for Rendering. *Technical Report, ID: TR1613*, 2007.

[33] F.E.Nicodemus, J.C.Richmond, J.J. Hsia, I.W.Ginsberg, and T.Lamperis. Geometrical Considerations and Nomenclature for Reflectance. *NBS Monograph 160, National Bureau of Standards, Washington, D.C.*, 1977.

[34] Gordon Forbes. *Texture and Bubble Size Measurements for Modelling Concentrate Grade in Flotation Froth Systems*. PhD thesis, Department of Chemical Engineering, University of Cape Town, 2007.

[35] J. Francis. *"Machine Vision for Froth Flotation"*. PhD thesis, Department of Electrical Engineering, University of Cape Town, 2001.

[36] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12:16–22, 2000.

[37] D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRFs: Surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):401–412, 1991.

[38] D. Geman. Statistical methods for tomographic image reconstruction. *Bull. Int. Stat. Inst.*, LII-4:5–21, 1987.

[39] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 721–741, 1984.

[40] A. S. Georghiades. Recovering 3-D Shape and Reflectance From a Small Number of Photographs. *European Conference on Computer Vision*, 2006.

[41] C. J. Geyer. Estimating normalizing constants and reweighting mixtures in Markov chain Monte Carlo. *Technical Report*, 568, 1991.

[42] C. J. Geyer and E. A. Thompson. Constrained Monte Carlo maximum likelihood for dependent data (with discussion). *Journal of the Royal Statistical Society (B)*, 54:657–699, 1992.

[43] Dan B. Goldman, Brian Curless, Aaron Hertzman, and Steven M. Seitz. Shape and Spatially-Varying BRDFs from Photometric Stereo. *International Conference on Computer Vision*, 2005.

[44] N. Gordon, D. Salmond, and C. Ewing. Bayesian state estimation for tracking and guidance using the bootstrap filter. *Journal of Guidance, Control and Dynamics*, 18:1434–1443, 1995.

[45] N. Gordon, D. Salmond, and A. F. M. Smith. Novel approach to nonlinear/Gaussian Bayesian state estimation. *IEEE Proc. F*, 140:107–113, 1993.

[46] M. Grelli and L. Onural. On a Parameter Estimation Method for Gibbs-Markov Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4), 1994.

[47] Michal Haindl. Texture Segmentation Using Recursive Markov Random Field Parameter Estimation. *Proceedings of the 11th Scandinavian Conference on Image Analysis*, pages 771–776, 1999.

[48] F. R. Hampel, E. M. Rochetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley and Sons, New York, NY, 1986.

[49] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[50] W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57:97–109.

[51] A. Hertzmann and S. M. Seitz. Shape and Materials by Example: A Photometric Stereo Approach. *CVPR*, 2003.

[52] D. M. Higdon, J. E. Bowsher, V. E. Johnson, T. G. Turkington, D. R. Gilland, and R. J. Jaszczak. Fully Bayesian Estimation of Gibbs Hyperparameters for Emission Computed Tomography Data. *IEEE Trans. on Medical Imaging*, 16(5):516–526, 1997.

[53] http://vision.middlebury.edu/ schar/stereo/web/results.php. Middlebury stereo vision homepage.

[54] Yukito Iba. Population Monte Carlo algorithms. *Transactions of the Japanese Society for Artificial Intelligence*, 16(2):279–286, 2000.

[55] F. Isgro and E. Trucco. Projective Rectification without Epipolar Geometry. *CVPR*, 1999.

[56] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. *In Proceedings of SIGGRAPH*, pages 511–518, 2001.

[57] J.M.Hammersley and P.Clifford. Markov Fields on finite graphs and lattices. *Unpublished*, 1971.

[58] K. F. Karner, H. Mayer, and M. Gervautz. An image based measurement system for anisotropic reflection. *Eurographics*, 15:119–128, 1996.

[59] Zoltan Kato and Ting-Chuen Pong. A Markov random field image segmentation model for color textured images. *IVC*, 2001.

[60] A. Klaus, M. Sormann, and K. Kerner. Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure. *18th International Conference on Pattern Recognition*, 2006.

[61] E. Lafortune, S. Foo, K. Torrance, and D. Greenberg. Non-Linear Approximations of Reflectance Functions. *In Proc. SIGGRAPH*, 1997.

[62] Yu-Chi Lai, Shaohua Fan, Stephen Chenney, and Charles Dryer. Photorealistic Image Rendering with Population Monte Carlo Energy Redistribution. *Eurographics Symposium on Rendering*, pages 287–296, 2007.

[63] K. B. Laskey and J. Myers. Population Markov Chain Monte Carlo. *Machine Learning*, 50:175–196(22), 2003.

[64] J. Lawrence, Szymon Rusinkiewicz, and Ravi Rammamoorthi. Efficient BRDF Importance Sampling Using a Factored Representation. *SIGGRAPH*, 23:3:496–505, 2004.

[65] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-Based Reconstruction of Spatially Varying Materials. *Rendering Techniques '01*, pages 104–115, 2001.

[66] Gang Li. Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity measure. *CVPR*, pages 2355–2362, 2006.

[67] T. Lissack and K. Fu. Error estimation in pattern recognition via L-distance between posterior density functions. *IEEE Trans. Information Theory*, 22:33–45, 1976.

[68] 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.

[69] Jun Liu, Lei Wang, and Stan Z. Li. MRMRF Texture Classification and MCMC Parameter Estimation. *Vision Interface, Canada*, 1999.

[70] Xinguo Liu, Yizhou Yu, and Heung-Yeung Shum. Synthesizing Bidirectional Texture Functions for Real-World Surfaces. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 97 – 106, 2001.

[71] M. Louw and F. Nicolls. A faster MRF optimization based method for Shape from Shading using Gibbs sampling with quadruplet cliques. *In Proc. Pattern Recognition Association of South Africa.*, 2007.

[72] 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.

[73] 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.

[74] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *In Proc. 7th Int. Conf. on Computer Vision*, pages 572–578, 1999.

[75] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface quality hand tracking. *European Conference on Computer Vision*, 2000.

[76] 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.

[77] P. Mahalanobis. On the generalized distance in statistics. *Proc. National Inst. Sci. (India)*, 12:49–55, 1936.

[78] K. Matusita. Decision rules based on the distance for problems of fit, two samples and estimation. *Ann. Math. Stat.*, 26:631–640, 1955.

[79] H. Mayer. Image-Based Texture Analysis for Realistic Image Synthesis. *Proceedings XIII Brazilian Symposium on Computer Graphics and Image Processing*, pages 219 – 226, 2000.

[80] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 1953.

[81] Refaat M. Mohamed, Ayman El-Baz, and Aly A. Farag. Image Modeling Using Gibbs-Markov Random Field and Support Vector Machines Algorithm. *Int. Journal of Information Technology*, 1/4.

[82] Refaat M. Mohamed and Aly A. Farag. Mean Field Thoery for Density Estimation Using Support Vector Machines. *7th Int. Conf. on Information Fusion*, pages 495–501, July 2004.

[83] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: an empirical study. *UAI*, 1999.

[84] Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[85] P. K. Nanda, U. B. Desai, and P. G. Poonacha. A Homotopy Continuation Method for Parameter Estimation in MRF Models and Image Restoration. *IEEE International Symposium on Circuits and Systems*, pages 273–276, 1994.

[86] Michael Oren and Shree K. Nayar. Generalization of the Lambertian Model and Implications for Machine Vision. *Int. Journal of Computer Vision*, 1995.

[87] E. Parzen. On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1064–1076, 1962.

[88] E. Patrick and F. Fisher. Nonparametric feature selection. *IEEE Trans. Information Theory*, 15:577–584, 1969.

[89] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[90] M. Pitt and N. Shephard. Filtering via Simulation: Auxiliary Particle Filters. *JASA*, 94:1403–1412, 1999.

[91] P.J.Huber. *Robust Statistics*. John Wiley and Sons, New York, NY, 1981.

[92] W. Pun and B. Jeffs. Shape parameter estimation for generalized Gaussian Markov random field models used in MAP image restoration. *In 29th Asilomar Conference on Signals, Systems and Computers*, 1995.

[93] W. Pun and B. Jeffs. Shape parameter estimation for generalized Gaussian Markov random field models used in MAP image restoration. *In Proc. 29th Asilomar Conference on Signals, Systems, and Computers*, 1995.

[94] Ravi Ramamoorthi and Pat Hanrahan. A Signal-Processing Framework for Inverse Rendering. *In Proc. SIGGRAPH*, 2001.

[95] E. Ranguelova and A. Quinn. Difference Field Estimation for Enhanced 3-D Texture Segmentation. *British Machine Vision Conference*, 2002.

[96] C. P. Robert and G. Casella. *Multiple Imputation for Nonresponse in Surveys*. John Wiley, New York, 1987.

[97] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 1999.

[98] R.O.Dror, E. Adelson, and A. Willsky. Estimating Surface Reflectance Properties from Images under Unknown Illumination. *SPIE*, 2001.

[99] R.O.Dror, E. Adelson, and A. Willsky. Surface Reflectance Estimation and Natural Illumination Statistics. *SCTV*, 2001.

[100] Y. Rubner, C. Tomasi, and L. J. Guibas. A Metric for Distributions with Applications to Image Databases. *International Conference on Computer Vision*, pages 59–66, 1998.

[101] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 27(60):259–268, 1992.

[102] Szymon Rusinkiewicz. A New Change of Variables for Efficient BRDF Representation. *In Proc. Eurographics Workshop on Rendering*, pages 11–22, 1998.

[103] Suhail S. Saquib, Charles A. Bouman, and Ken Sauer. A non-homogeneous MRF model for multiresolution Bayesian estimation. *In Proc. of IEEE Int. Conf. on Image Processing*, 2:445–448, 1996.

[104] Suhail S. Saquib, Charles A. Bouman, and Ken Sauer. Efficient ML Estimation of the Shape Parameter for Generalized Gaussian MRF. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 4:2227–2230, 1996.

[105] Suhail S. Saquib, Charles A. Bouman, and Ken Sauer. ML Parameter Estimation for Markov Random Fields with Applications to Bayesian Tomography. *IEEE Trans. Image Processing*, 7(7):1029–1044, 1998.

[106] Yoichi Sato, Mark D. Wheeler, and Katsushi Ikeuchi. Object Shape and Reflectance Modelling from Observation. *In Proc. SIGGRAPH*, 1997.

[107] C. Schlick. An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum*, 13(3), 1994.

[108] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. *CVPR*, 2005.

[109] Jian Sun, Heung-Yeung Shum, and Nan-Ning Zheng. Stereo Matching Using Belief Propagation. *European Conference on Computer Vision*, 2002.

[110] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields. *European Conference on Computer Vision*, 2006.

[111] K.-L. Tang, C.-K. Tang, and T.-T. Wong. Dense photometric stereo using tensorial belief propagation. *CVPR*, 2005.

[112] 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.

[113] T.M.Cover and J.A.Thomas. *Elements of Information Theory*. Wiley, 1991.

[114] K. Torrance and E. Sparrow. Theory for off-specular reflections from rough surfaces. *Journal of the Optical Society of America*, 57:1105–1114, 1967.

[115] K. E. Torrance and E. M. Sparrow. Theory for Off-Specular Reflection From Roughened Surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114, 1967.

[116] A. Tsai, J. Zhang, and A. Willsky. Multiscale Methods and Mean Field Theory in EM Procedures for Image Processing. *Eighth IEEE Digital Signal Processing Workshop*, 1998.

[117] Shinji Umeyama. Separation of Diffuse and Specular Components of Surface Reflection by Use of Polarization and Statistical Analysis of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:639–647, 2004.

[118] P. Viola and W. M. Wells III. Alignment by Maximization of Mutual Information. *International Journal of Computer Vision*, 24(2), 1997.

[119] George Vogiatzis, Paolo Favaro, and Robert Cipolla. Using Frontier Points to Recover Shape, Reflectance and Illumination. *International Conference on Computer Vision*, 2005.

[120] M. Wainwright, T. Jaakkola, and A. Willsky. Tree-based reparameterisation for approximate estimation on loopy graphs. *NIPS-11*, 2001.

[121] Martin J. Wainwright. Estimating the "wrong" Markov random field: Benefits in the computation-limited setting. *Advances in Neural Information Processing Systems*, pages 1425–1432, 2006.

[122] Gregory J. Ward. Measuring and Modeling Anisotropic Reflection. *SIGGRAPH*, 1992.

[123] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1-41.

[124] Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *NIPS-12*, 1999.

[125] M. Welling and Y-W Teh. Belief optimization for binary networks: a stable alternative to loopy belief propagation. *UAI*, 2001.

[126] Stephen H. Westin, Hongsong Li, and Kenneth E. Torrance. A Comparison of Four BRDF Models. *Technical report PCG-04-02, Program of Computer Graphics, Cornell University.*, 2004.

[127] W. H. Wong and F. Liang. Dynamic Weighting in Monte Carlo and Optimization. *Proceedings of the National Academy of Sciences*, pages 14220–14224, 1994.

[128] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo Matching with Color-Weighted Correlation, Heirarchical Belief Propagation and Occlusion Handling. *CVPR*, 2006.

[129] Xiangyu Yang and Jun Liu. Maximum Entropy Random Field Model for Texture Analysis. *In Proc. Vision Interface*, 1999.

[130] Jonothan S. Yedidia, William T. Freeman, and Yair Weiss. Bethe free energy, Kikuchi approximations and belief propagation algorithms. *SCTV*, 2001.

[131] Tianli Yu, Ning Xu, and Narendra Ahuja. Shape and View Independent Reflectance Map from Multiple Views. *European Conference on Computer Vision*, 2004.

[132] A. Yuille. A double loop algorithm to minimize the Bethe and Kikuchi free energies. *NIPS*, 2000.

[133] H. Zhang and S. Negahdaripour. BC and GC-Based Dense Stereo By Belief Propagation. *ICVS*, 2006.

[134] J. Zhang. The Mean Field Theory in EM procedures for Markov Random Fields. *IEEE Trans. Signal Processing*, 40(10):2570–2583, 1992.

[135] J. Zhang. The Mean Field Theory in EM procedures for blind Markov Random Field image restoration. *IEEE Trans. Iamge Processing*, 2(1):27–40, 1993.

[136] J. Zhang and G. G. Hanauer. The application of mean field theory to image motion estimation. *IEEE on Trans. Image Processin*, 4(1):19–33, 1995.

[137] Li Zhang and Steven M. Seitz. Estimating Optimal Parameters for MRF Stereo from a Single Image Pair. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2), 2007.

[138] M. C. Zhang, R. M. Haralick, and J. B. Campbell. Multispectral image context classification using stochastic relaxation. *IEEE Trans. on Systems, Man and Cybernetics*, 20(1):128–140, 1990.

[139] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from Shading: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[140] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[141] 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.

[142] C. R. Zou, Z. Y. He, E. I. Plotkin, and M. N. S. Swamy. Recursive-in-order Least Squares Parameter Esimation for 2-D Gaussian Markov Random Field Model. *IEEE International Symposium on Circuits and Systems*, 2:734–737, 1992.