UNIVERSITY OF CALIFORNIA, SAN DIEGO

Data Analysis Methods for Motion Segmentation and Material Reflectance

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Joshua J. Wills

Committee in charge:

Professor Serge Belongie, Chair
Professor Sam Buss
Professor Virginia de Sa
Professor Henrik Wann Jensen
Professor David Kriegman

2006

The dissertation of Joshua J. Wills is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2006

To Leandra

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I owe many people a large debt of gratitude for the help and support they have given me during the course of this dissertation.

I would like to thank my friends and family for their support and encouragement and for humoring me to the point of reading every one of my papers.

It has been a great privilege and a great pleasure to work with my advisor, Serge Belongie. His guidance and counseling has formed the foundations of both this thesis and my research career. I have also benefited from the knowledge and guidance I have received from David Kriegman and Henrik Wann Jensen. My work would not have been possible if not for invaluable conversations with Sameer Agarwal, Kristin Branson, Manmohan Chandraker, Piotr Dollar, Craig Donner, Satya Mallick, Ben Ochoa, and Eric Wiewora. I would also like to thank my earlier teachers, Jesus Jimenez, Gerald Lashley, and Jeff McKinstry, for instilling in me an appreciation for Computer Science and encouraging me to pursue graduate school.

Finally I would like to thank my wife, Leandra, to whom this dissertation is dedicated for her patience and support without which this dissertation would have proven impossible.

Portions of this dissertation are based on papers that I have co-authored with others. Listed below are my contributions to each of these papers.

1. Chapter 2 is based on work that has appeared as:

   - J. Wills, S. Agarwal and S. Belongie, "What Went Where," *CVPR*, 2003, pp. 37-44, vol. 1. [126]. I was the primary author and was responsible for the development of the method, completing the literature survey, performing the experiments and all implementation except the graph cut code.
   - J. Wills, S. Agarwal and S. Belongie, "A Feature-based Approach for Dense Segmentation and Estimation of Large Disparity Motion." *International Journal of Computer Vision*, 2006, pp. 125–143, vol. 68 (2) [127]. I was the primary author and was responsible for the development of the method, completing the literature survey, performing the experiments and all implementation except graph cut code.

2. Chapter 4 has been submitted for publication of the material as: J. Wills, S. Agarwal, D. Kriegman and S. Belongie, "Toward a Perceptual Space for Reflectance,"

ACM Transactions on Graphics, 2006, in review. I was the primary author and responsible for the design and implementation of the psychophysics experiment, the literature survey, and produced all renderings.

In addition, Chapter 2 is also based on work with my advisor and I that has appeared as:

- S. Belongie and J. Wills, "Structure from Periodic Motion," *SCVMA* (in conjunction with *ECCV*), 2004. [10].
- J. Wills and S. Belongie, "A Feature-Based Approach for Determining Long Range Optical Flow," *ECCV*, 2004, pp. 170-182, vol. 3 [128].

Figures 3.1 and 3.6 were rendered by Henrik Wann Jensen.

VITA

1978	Born, Twin Falls, Idaho
2001	B. A., *summa cum laude*, Point Loma Nazarene University
2004	M. S., University of California San Diego
2006	Ph. D., University of California San Diego

PUBLICATIONS

What Went Where (With S. Agarwal and S. Belongie.) *CVPR*, 2003, pp. 37-44, vol. 1.

Structure from Periodic Motion (With S. Belongie.) *SCVMA* (in conjunction with *ECCV*), 2004.

A Feature-Based Approach for Determining Long Range Optical Flow (With S. Belongie.) *ECCV*, 2004, pp. 170-182, vol. 3.

Periodic Motion Detection and Segmentation via Approximate Sequence Alignment (With I. Laptev, S. Belongie and P. Pérez.) *ICCV*, 2005, pp. 816-823, vol. 1.

A Feature-based Approach for Dense Segmentation and Estimation of Large Disparity Motion (With S. Agarwal and S. Belongie.) *International Journal of Computer Vision*, 2006, to appear.

Toward a Perceptual Space for Reflectance (With S. Agarwal, D. Kriegman and S. Belongie.) *ACM Transactions on Graphics*, 2006, in review.

ABSTRACT OF THE DISSERTATION

Data Analysis Methods for Motion Segmentation and Material Reflectance

by

Joshua J. Wills

Doctor of Philosophy in Computer Science

University of California San Diego, 2006

Professor Serge Belongie, Chair

Image analysis and image synthesis are the goals of computer vision and computer graphics, respectively. These research areas represent the domains into which the work presented in this dissertation fall. Specifically, we present work on three problems: segmentation and estimation of large disparity motion, simulating the reflectance for rough surfaces using microfacet models, and the perception of material reflectance.

We present a novel framework for motion segmentation that combines the concepts of layer-based methods and feature-based motion estimation. We demonstrate our approach on image pairs containing large inter-frame motion and partial occlusion. The approach is efficient, and it successfully segments scenes with inter-frame disparities beyond the scope of previous methods. We also present an extension that accounts for the case of non-planar motion, and applications of our method to automatic object removal and to structure from motion.

The Bidirectional Reflectance Distribution Function (BRDF) describes the way a surface reflects light. Microfacet reflectance models have been shown to work well for simulating the interaction of light with a rough surface. We give an overview of the existing techniques for reflection modeling and show how these techniques can be extended to handle refraction in a unified framework. To this end, two new derivations are presented for computing quantities required for refraction as well as a result that is (to our knowledge) previously unpublished.

While BRDFs allows for a complete radiometric description of light reflecting from a surface, they are complex mathematical objects that can be difficult to use in practice. Our aim is to construct a low-dimensional, perceptual space for BRDFs that can

be easily navigated. To this end, we design and carry out a comprehensive psychophysical study of the perception of measured reflectance. This is the largest study of its kind to date, and the first to use real material measurements. In addition, we introduce a new multidimensional scaling (MDS) algorithm for analyzing ordinal data that unlike existing methods is both efficient and optimal. We use the results of our study to construct a perceptual space of these BRDFs and introduce a new method for perceptual construction of novel BRDFs.

1

Introduction

The vast majority of motion pictures today contain some amount of computer generated imagery (CGI). In many cases, artificial elements are inserted into live-action footage to give the illusion that the scene contained both the artificial and real elements. To achieve this type of effect, the live-action scene must be analyzed and the locations and positions of objects in the scene as well as the camera position and properties must be estimated. In addition, the portions of the frames that contain the artificial elements must be synthesized taking into account the lighting of the scene, the properties of the elements, and the position of the camera. Image analysis and image synthesis are the goals of computer vision and computer graphics, respectively. These research areas represent the domains into which the work presented in this dissertation fall. Specifically, we present work on three problems:

- Large Disparity Motion Segmentation
- Microfacet-based Reflectance Models
- The Perception of Material Reflectance

While each of these problems are distinct, they share similar concepts and techniques. We will now give an introduction to each problem followed by a discussion of the correlation between problems.

The problem of motion segmentation requires the estimation of the motions present in a set of images as well as a segmentation of the pixels in each image to one of the estimated motions. Traditionally this segmentation has focused on video sequences where the inter-frame motion is quite small, and there are many approaches that provide satisfactory results. However, many of these techniques fail if the motion

between frames is too large. This type of motion – large disparity motion – is most commonly due to quickly moving objects and/or low frame-rate sequences. However, there are many other situations that give rise to large disparity motion such as, camera handoff or the registration of motion that is present in two widely separated cameras (e.g. two surveillance cameras on opposite sides of a corner) or dynamic scene registration (e.g. aligning images that capture the lighting of a scene to be used in rendering special effects with same scene as it appears in a sequence shot at a different time where many elements of the scene may have moved). The goal is to be able to find the coherently moving objects in a scene regardless of the magnitude of the motion between frames.

The goal of reflectance models is to model the effects produced by the interaction of light with a surface. These effects are what make velvet look like velvet and copper look like copper. In the case of microfacet-based reflectance models, we are trying to capture the effects for a rough surface. The earliest reflectance models in graphics were aimed at one of two extremes, either modeling surfaces that are extremely smooth [87] or extremely rough [62], though many real world materials fall somewhere in between. It is these types of materials that microfacet models were designed to simulate and they assume that the surface can be modeled as a collection of tiny facets that are usually assumed to be mirrors [108], though there has been work on surfaces with diffuse facets [84]. It is the aggregate reflection from these tiny facets that give a rough surface its reflective properties.

In many problems within computer graphics, the goal is to model the physics of the scene (especially the physics of the interaction of light and materials) as accurately as possible. However, in many problems (particularly though encountered when synthesizing images for a motion picture) the ultimate goal is to convince the viewers of the realism of the scene regardless of the physical or radiometric correctness. With this in mind, it is valuable to understand the perception of reflectance and to determine what aspects of reflectance are perceptually meaningful. The problem we address in this dissertation is the estimation of a space of reflectance for a set of materials, or how can we arrange the set of materials in a new space such that the relative distances between materials approximate the perceptual distances experienced by viewers, which we capture through a set of psychophysical experiments.

As mentioned previously, we can divide these three problems into the areas of computer vision and computer graphics with the first, motion segmentation, falling squarely into computer vision, the second, microfacet reflectance, falling squarely into computer graphics, and the third, the perception of reflectance, which straddles the

often blurry boundary between the two. Some of the similarities are quite obvious, for example, the object of interest in both the second and third problems is a function that models reflectance. Though most of the common themes are higher level. Two significant components that are present in all three pieces of work are techniques for the aggregation of data from multiple sources and the discovery of simplified representation of complex objects.

The first, the aggregation of information, is a primary challenge in all three problems. In our approach for motion segmentation there are two significant instances of techniques for data aggregation: using groups of pixel correspondences to estimate the motion present in the scene and using similarity of neighboring pixels to compute a smooth assignment to motion layers. In the case of microfacets, this principle is implicit as we are trying to model the net effect of reflection from many small facets, and it is only the aggregate reflectance that is important. Aggregation in the case of our perceptual analysis of reflectance is more explicit. Our method for capturing perceptual data requires many subjects to view many combinations of materials, and the output of our analysis is a representation of the data that best satisfies as many subjects for as many materials as possible.

The second, the discovery of simplified representations of complex objects, is common to many problems in computer vision and graphics including the three in which we are interested. The final goal of our approach to motion segmentation is to take a complex scene and decompose it into a handful of independent layers with associated motions. In most cases, we simplify this further by assuming the scene can be represented by planar layers, and we search for the best approximation. Similarly, microfacet reflectance functions work off of the assumption that while general reflectance functions have many dimensions, they can be well approximated by a combination of much simpler (and lower dimensional) functions. Finally, our perceptual analysis results in a low dimensional space that is able to capture the majority of the perceptual distances present in our subject data.

Since the problems are relatively distinct, each problem is presented in a separate chapter and each chapter is largely self-contained. The structure of this dissertation is as follows: we begin with a presentation of our approach to large disparity motion in Chapter 2, we present our work on microfacet reflectance models in Chapter 3, and our work on the perception of material reflectance is presented in Chapter 4.

2

Large Disparity Motion Segmentation

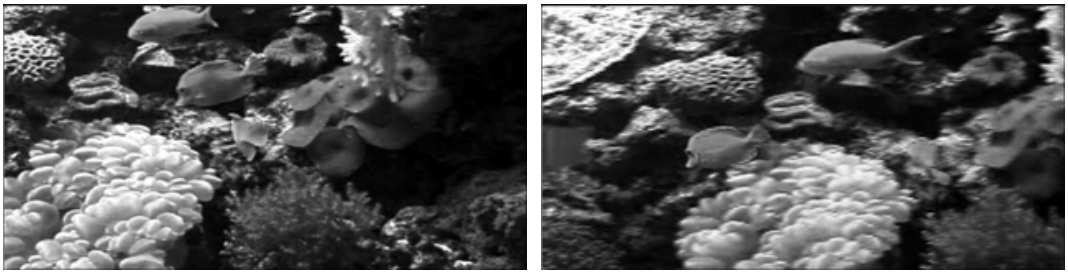


Figure 2.1 Two consecutive frames from a saltwater aquarium webcam.

Consider the pair of images shown in Figure 2.1. These two images were captured by an aquarium webcam on a pan-tilt head. For a human observer, a brief examination of the images reveals what happened from one frame to the next: the lower fish swam down and darted forward and the upper fish moved forward slightly; meanwhile, the camera panned to the left about a third of the image width. Even without color information, this is a simple task for the human visual system. The same cannot be said for any existing computer vision system. What makes this problem difficult from a computational perspective? There are number of complicating factors, including the following: (1) due to the low frame rate, the motion between frames is a significant fraction of the image size, (2) the moving objects are relatively small and have few features compared to the richly textured background, (3) the poses of the fish change as they swim, (4) because of the panning motion of the camera, the second frame has motion blur.

Finding out what went where in two frames of an image sequence is an instance of the motion segmentation problem. Formally, motion segmentation consists of (1) finding groups of pixels in two or more frames that move together, and (2) recovering the motion fields associated with each group. Motion segmentation has wide applicability in areas such as video coding, content-based video retrieval, and mosaicking. In its full generality, the problem cannot be solved since infinitely many constituent motions can explain the changes from one frame to another. Fortunately, in real scenes the problem is simplified by the observation that objects are usually composed of spatially contiguous regions, and the number of independent motions is significantly smaller than the number of pixels. Operating under these assumptions, we propose a new motion segmentation algorithm for scenes containing objects with large inter-frame motion. The algorithm leverages and builds upon established techniques for robust estimation of motion fields and discontinuity preserving smoothing in a novel combination that delivers the first dense, layer-based motion segmentation method for the case of large (non-differential) motions.

The structure of this chapter is as follows: we begin with a presentation of our approach to large disparity planar motion in section 2.1, we present our extension to non-planar motion in section 2.2, applications are presented in section 2.3, and conclusions and future work are presented in section 2.4.

2.1 Planar Motion Segmentation

2.1.1 Introduction

Early approaches to motion segmentation were based on estimating dense optical flow. The optical flow field was assumed to be piecewise smooth to account for discontinuities due to occlusion and object boundaries, see for example [6, 13, 82]. Darrell & Pentland [26] and Wang & Adelson [120] introduced the idea of decomposing the image sequence into multiple overlapping layers, where each layer is a smooth motion field. Weiss [124] extended this approach to account for flexible motion fields using regularized radial basis functions (RBFs).

Optical flow based methods are limited in their ability to handle large inter-frame motion or objects with overlapping motion fields. Coarse-to-fine methods are able to solve the problem of large motion to a certain extent (see for example [100, 101]), but the degree of sub-sampling required to make the motion differential places an upper

bound on the maximum allowable motion between two frames and limits it to about 15% of the dimensions of the image [52]. Also in cases where the order of objects along any line in the scene is reversed and their motion fields overlap, the coarse to fine processing ends up blurring the two motions into a single motion before optical flow can be calculated.

In this dissertation we are interested in the case of discrete motion, i.e. where optical flow based methods break down. Most closely related to our work is that of Torr [104]. Torr uses sparse correspondences obtained by running a feature detector and matching them using normalized cross correlation. He then processes the correspondences in a RANSAC framework to sequentially cover the the set of motions in the scene. Each iteration of his algorithm finds the dominant motion model that best explains the data and is simplest according to a complexity measure. The set of models and the associated correspondences are then used as the initial guess for the estimation of a mixture model using the Expectation Maximization (EM) algorithm. Spurious models are pruned and the resulting segmentation is smoothed using morphological operations.

In a more recent work [106], the authors extend the model to 3D layers in which points in the layer have an associated disparity. This allows for scenes in which the planarity assumption is violated and/or a significant amount of parallax is present. The pixel correspondences are found using a multiscale differential optical flow algorithm, from which the layers are estimated in a Bayesian framework using EM. Piecewise smoothness is ensured by using a Markov random field prior.

Neither of the above works demonstrate the ability to perform dense motion segmentation on a pair of images with large inter-frame motion. In both of the above works the grouping is performed in a Bayesian framework. While the formulation is optimal and strong results can be proved about the optimality of the Maximum Likelihood solution, actually solving for it is an extremely hard non-linear optimization problem. The use of EM only guarantees a locally optimal solution and says nothing about the quality of the solution. As the authors point out, the key to getting a good segmentation using their algorithm is to start with a good guess of the solution and they devote a significant amount of effort to finding such a guess. However it is not clear from their result how much the EM algorithm improves upon their initial solution.

2.1.2 Proposed Method

Our approach is based on a two stage process, the first of which is responsible for motion field estimation and the second of which is responsible for motion layer as-

signment. As a preliminary step we detect interest points in the two images and match them by comparing filter responses. We then use a RANSAC based procedure for detecting the motion fields relating the frames. Based on the detected motion fields, the correspondences detected in the first stage are partitioned into groups corresponding to each constituent motion field, and the resulting motion fields are re-estimated. Finally, we use a fast approximate graph cut based method to densely assign pixels to their respective motion fields. We now describe each of these steps in detail. A reference Matlab implementation of the steps described in this section is available for download at `http://vision.ucsd.edu/motion_seg.html`.

2.1.3 Interest point detection and matching

Many pixels in real images are redundant, or very similar to nearby pixels, so it is beneficial to find a set of points that reduce some of this redundancy. To achieve this, we detect interest points using the Förstner operator [37]. To describe each interest point, we apply a set of 76 filters (3 scales and 12 orientations with even and odd phase and an elongation ratio of 3:1, plus 4 spot filters) to each image. The filters, which are at most 31×31 pixels in size, are evenly spaced in orientation at intervals of 15°, and the changes in scale are half octave. For each of the scales and orientations, we use a quadrature pair of derivative-of-Gaussian filters corresponding to edge and bar-detectors respectively, as in [39, 54].

To obtain some degree of rotational invariance, the filter response vectors may be reordered so that the order of orientations is cyclically shifted. This is equivalent to filtering a rotated version of the image patch that is within the support of the filter. We perform three such rotations in each direction to obtain rotational invariance up to $\pm 45^\circ$.

We find correspondences by comparing filter response vectors using the L_1 distance. We compare each interest point in the first image to those in the second image and assign correspondence between points with minimal error. Since matching is difficult for image pairs with large inter-frame disparity, the remainder of our approach must take into account that the estimated correspondences can be extremely noisy.

Estimating Motion Fields

Robust estimation methods such as RANSAC [33] have been shown to provide very good results in the presence of noise when estimating a single, global transformation

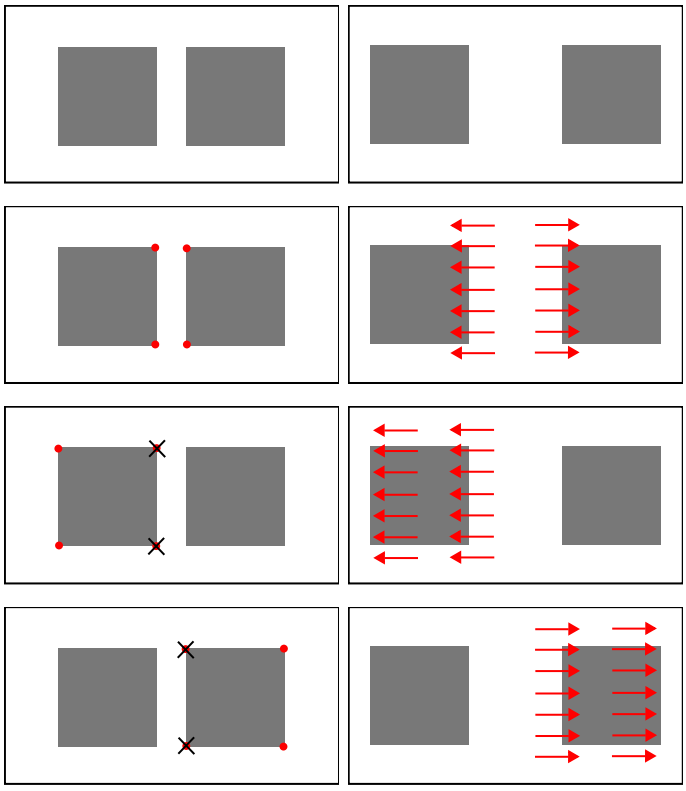


Figure 2.2 Phantom motion fields. (Row 1) Scene that consists of two squares translating away from each other. (Row 2) Under an affine model, triplets of points that span the two squares will incorrectly propose a global stretching motion. This motion is likely to have many inliers since all points on the inner edges of the squares will fit this motion exactly. If we then delete all points that agree with this transformation, we will be unable to detect the true motions of the squares in the scene (Rows 3 & 4).

between images. Why can't we simply apply these methods to multiple motions directly? It turns out that this is not as straightforward as one might imagine. Methods in this vein work by iteratively repeating the estimation process where each time a dominant motion is detected, all correspondences that are deemed inliers for this motion are removed [104].

There are a number of issues that need to be addressed before RANSAC can be used for the purpose of detecting and estimating multiple motions. The first issue is that combinations of correspondences – not individual correspondences – are what promote a given transformation. Thus when "phantom motion fields" are present, i.e., transformations arising from the relative motion between two or more objects, it is possible that the deletion of correspondences could prevent the detection of the true independent motions; see Figure 2.2. Our approach does not perform sequential deletion of correspondences and thus circumvents this problem.

Another consideration arises from the fact that the RANSAC estimation procedure is based on correspondences between interest points in the two images. This makes the procedure biased towards texture rich regions, which have a large number of interest points associated with them, and against small objects in the scene, which in turn have a small number of interest points. In the case where there is only one global transformation relating the two images, this bias does not pose a problem. However it becomes apparent when searching for multiple independent motions. To correct for this bias we introduce "perturbed interest points" and a method for feature crowdedness compensation.

Perturbed Interest Points

If an object is only represented by a small number of interest points, it is unlikely that many samples will fall entirely within the object. One approach for promoting the effect of correct correspondences without promoting that of the incorrect correspondences is to appeal to the idea of a stable system. According to the principle of perturbation, a stable system will remain at or near equilibrium even as it is slightly modified. The same holds true for stable matches. To take advantage of this principle, we dilate the interest points to be disks with a radius of r_p, where each pixel in the disk is added to the list of interest points. This allows the correct matches to get support from the points surrounding a given feature while incorrect matches will tend to have almost random matches estimated for their immediate neighbors, which will not likely contribute to a widely-supported warp. In this way, while the density around a valid motion is increased, we do not see the same increase in the case of an invalid motion; see Figure 2.3.

Feature Crowdedness

Textured regions often have significant representation in the set of interest points. This means that a highly textured object will have a much larger representation in the set of interest points than an object of the same size with less texture. To mitigate this effect, we bias the sampling. We calculate a measure of crowdedness for each interest point and the probability of choosing a given point is inversely proportional to this crowdedness score. The crowdedness score is the number of interest points that fall into a disk of radius r_c.

Partitioning and Motion Estimation

Having perturbed the interest points and established a sampling distribution on them, we are now in a position to detect the motions present in the frames. We do so using a two step variant of RANSAC, where multiple independent motions are explicitly handled, as duplicate transformations are detected and pruned in a greedy

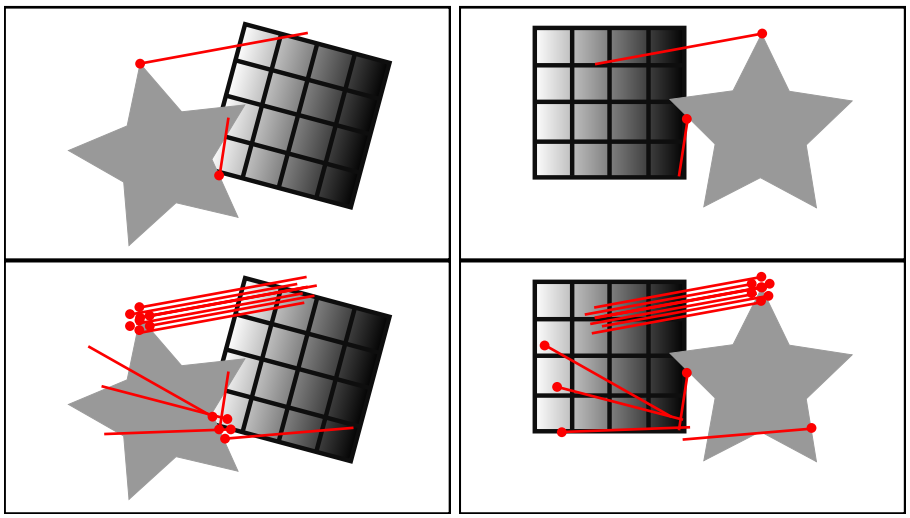


Figure 2.3 Perturbed Interest Points. Correspondences are represented by point-line pairs where the point specifies an interest point in the image and the line segment ends at the location of the corresponding point in the other image. (Row 1) We see one correct correspondence and one incorrect correspondence that is the result of an occlusion junction forming a white wedge. (Row 2) The points around the correct point have matches that are near the corresponding point, but the points around the incorrect correspondence do not.

manner. The first step provides a rough partitioning of the set of correspondences (motion identification) and the second takes this partitioning and estimates the motion of each group (motion refinement).

First, a set of planar warps is estimated by a round of standard RANSAC and inlier counts (using an inlier threshold of τ) are recorded for each transformation. In our case, we use planar homography which requires 4 correspondences to estimate, however similarity or affinity may be used (requiring 2 and 3 correspondences, respectively). The estimated list of transformations is then sorted by inlier count and we keep the first n_t transformations, where n_t is some large number (e.g. 300).

We expect that the motions in the scene will likely be detected multiple times, and we would like to detect these duplicate transformations. Comparing transformations in the space of parameters is difficult for all but the simplest of transformations, so we compare transformations by comparing the set of inliers associated with each transformation. If there is a large overlap in the set of inliers (more than 75%) the transformation with the larger set of inliers is kept and the other is pruned.

Now that we have our partitioning of the set of correspondences, we would like to estimate the planar motion represented in each group. This is done with a second round of RANSAC on each group with only 100 iterations. This round has a tighter

threshold to find a better estimate. We then prune duplicate warps a second time to account for slightly different inlier sets that converged to the same transformation during the second round of RANSAC with the tighter threshold.

The result of this stage is a set of proposed transformations and we are now faced with the problem of assigning each pixel to a candidate motion field.

Layer Assignment

The problem of assigning each pixel to a candidate motion field can be formulated as finding a function $l : I \rightarrow \{1, \ldots, m\}$, that maps each pixel to an integer in the range $1, \ldots, m$, where m is the total number of of motion fields, such that the reconstruction error

$$\sum_i [I(i) - I'(M(l(i), i))]^2$$

is minimized. Here $M(p, q)$ returns the position of pixel q under the influence of the motion field p.

A naïve approach to solving this problem is to use a greedy algorithm that assigns each pixel the motion field for which it has the least reconstruction error, i.e.,

$$l(i) = \underset{1 \leq p \leq m}{\operatorname{argmin}} [I(i) - I'(M(p, i))]^2 \tag{2.1}$$

The biggest disadvantage of this method as can be seen in Figure 2.4 is that for regions with relatively constant intensity it can produce unstable labellings, in that neighboring pixels that have the same brightness and are part of the same moving object can get assigned to different warps. What we would like instead is to have a labelling that is piecewise constant with the occasional discontinuity to account for genuine changes in motion fields.

The most common way this type of problem is solved (see e.g. [124]) is by imposing a smoothness prior over the set of solutions, i.e., an ordering that prefers piecewise constant labellings over highly unstable ones. It is important that the prior be sensitive to true discontinuites present in the image. In [18], for example, Boykov, Veksler and Zabih have shown that discontinuity preserving smoothing can be performed by adding a penalty of the following form to the objective function

$$\sum_i \sum_{j \in \mathcal{N}(i)} s_{ij} [1 - \delta_{l(i)l(j)}]$$

where $\delta_{(..)}$ is the Kronecker delta, equal to 1 when its arguments are equal. Given a measure of similarity s_{ij} between pixels i and j, it penalizes pixel pairs that have been assigned different labels. The penalty should only be applicable for pixels that are near each other. Hence the second sum is over a fixed neighborhood $\mathcal{N}(i)$. The final objective function we minimize is

$$\sum_i [I(i) - I'(M(l(i), i))]^2 + \lambda \sum_i \sum_{j \in \mathcal{N}(i)} s_{ij}[1 - \delta_{l(i)l(j)}]$$

where λ is the tradeoff between the data and the smoothness prior.

An optimization problem of this form is known as a *Generalized Potts model* which in turn is a special case of a class of problems known as metric labelling problems. Kleinberg & Tardos demonstrate that the metric labelling problems corresponds to finding the maximum *a posteriori* labelling of a class of Markov random field [58]. The problem is known to be NP-complete, and the best one can hope for in polynomial time is an approximation.

Recently Boykov, Veksler and Zabih (BVZ) have developed a polynomial time algorithm that finds a solution with error at most two times that of the optimal solution [19]. Each iteration of the algorithm constructs a graph and finds a new labelling of the pixels corresponding to the minimum cut partition in the graph. The algorithm is deterministic and guaranteed to terminate in $O(m)$ iterations.

Besides the motion fields and the image pair, the algorithm takes as input a

Figure 2.4 Example of naïve pixel assignment as in Equation 2.1 for the second motion layer in Figure 2.6. Notice there are many pixels that are erratically assigned. This is why smoothing is needed.

similarity measure s_{ij} between every pair of pixels i, j within a fixed distance of one another and two parameters, k the size of the neighborhood around each pixel, and λ the tradeoff between the data and the smoothness term. We use a Gaussian weighted measure of the squared difference between the intensities of pixels i and j,

$$s_{ij} = \exp\left[-\frac{d(i,j)^2}{2k^2} - (I(i) - I(j))^2\right]$$

where $d(i,j)$ is the distance between pixel i and pixel j.

We run the BVZ algorithm twice, once to assign the pixels in the image I to the forward motion field and again to assign the pixels in image I' to the inverse motion fields relating I' and I. If a point in the scene occurs in both frames, we expect that its position and appearance will be related as:

$$\begin{aligned} M(l(p), p) &= p' \\ M(l'(p'), p') &= p \\ I(M(l(p), p)) &= I'(p) \end{aligned}$$

Here, the unprimed symbols refer to image I and the primed symbols refer to image I'. Assuming that the appearance of the object remains the same across the images, the final assignment is obtained by intersecting the forward and backward assignments.

In this simple intersection step, occluded pixels are removed from further consideration. By reasoning about occlusion ordering constraints over more than than two frames, one can retain and explicitly label occluded pixels in the output segmentation; see for example the recent work of Xiao and Shah [130].

```
1.  Detect interest points in I
2.  Perturb each interest point
3.  Find the matching points in I'
4.  For i = 1:N_s
       Pick tuples of correspondences
       Estimate the warp
       Store inlier count
5.  Prune the list of warps
6.  Refine each warp using its inliers
7.  Perform dense pixel assignment
```

Figure 2.5 Algorithm Summary

2.1.4 Experimental Results

We now illustrate our algorithm, which is summarized in Figure 2.5, on several pairs of images containing objects undergoing independent motions. We performed all of the experiments on grayscale images with the same parameters[1].

Figure 2.6 Notting Hill sequence. (Row 1) Original image pair of size 311×552, (Rows 2-4) Pixels assigned to warp layers 1-3 in I and I'.

Our first example is shown in Figure 2.6. In this figure we show the two images, I and I', and the assignments for each pixel to a motion layer (one of the three detected motion fields). The rows represent the different motion fields and the columns represent

[1] $N_s = 10^4, n_t = 300, r_p = 2, r_c = 25, \tau = 10, k = 2, \lambda = .285$. Image brightnesses are in the range $[0, 1]$.

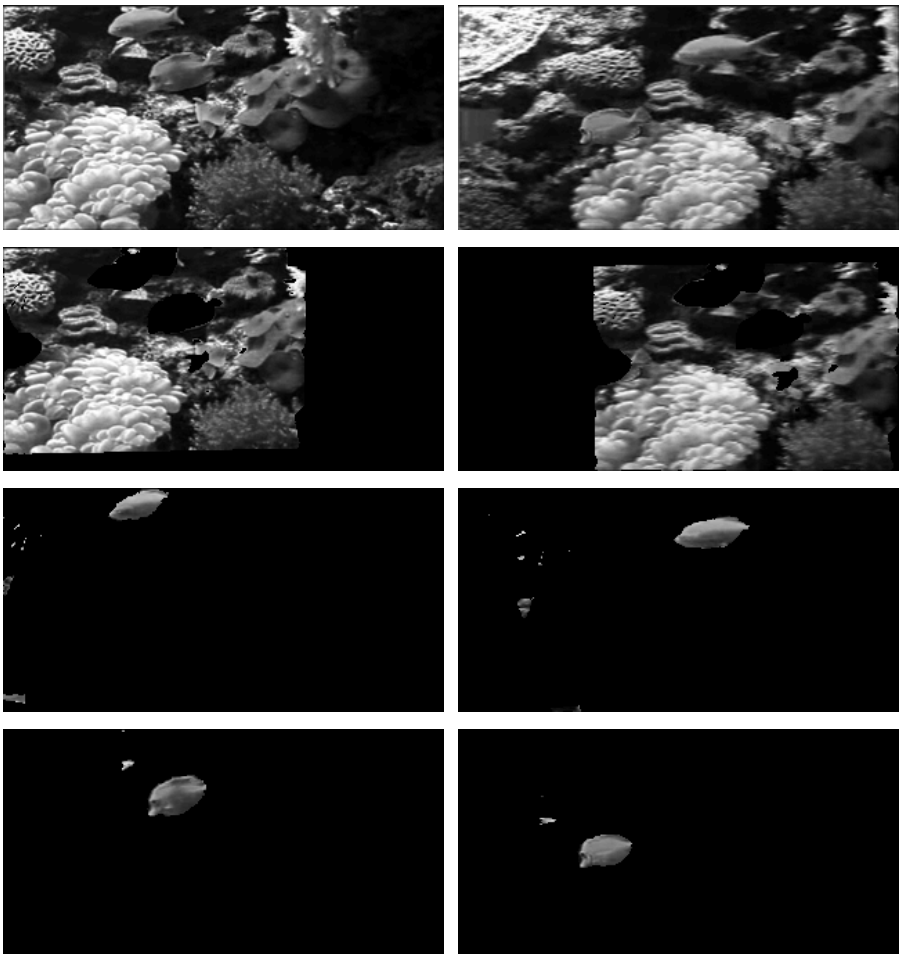


Figure 2.7 Fish sequence. (Row 1) Original image pair of size 162×319, (Rows 2-4) Pixels assigned to warp layers 1-3 in I and I'.

the portions of each image that are assigned to a given motion layer. The motions are made explicit in that the pixel support from frame to frame is related exactly by a planar homography. Notice that the portions of the background and the dumpsters that were visible in both frames were segmented correctly, as was the man. This example shows that in the presence of occlusion and when visual correspondence is difficult (i.e. matching the dumpsters correctly), our method provides good segmentation. Another thing to note is that the motion of the man is only approximately planar.

Figure 2.7 shows a scene consisting of two fish swimming past a fairly complicated reef scene. The segmentation is shown as in Figure 2.6 and we see that three motions were detected, one for the background and one for each of the two fish. In this scene, the fish are small, feature-impoverished objects in front of a large feature-rich

background, thus making the identification of the motion of the fish difficult. In fact, when this example was run without using the perturbed interest points, we were unable to recover the motion of either of the fish.

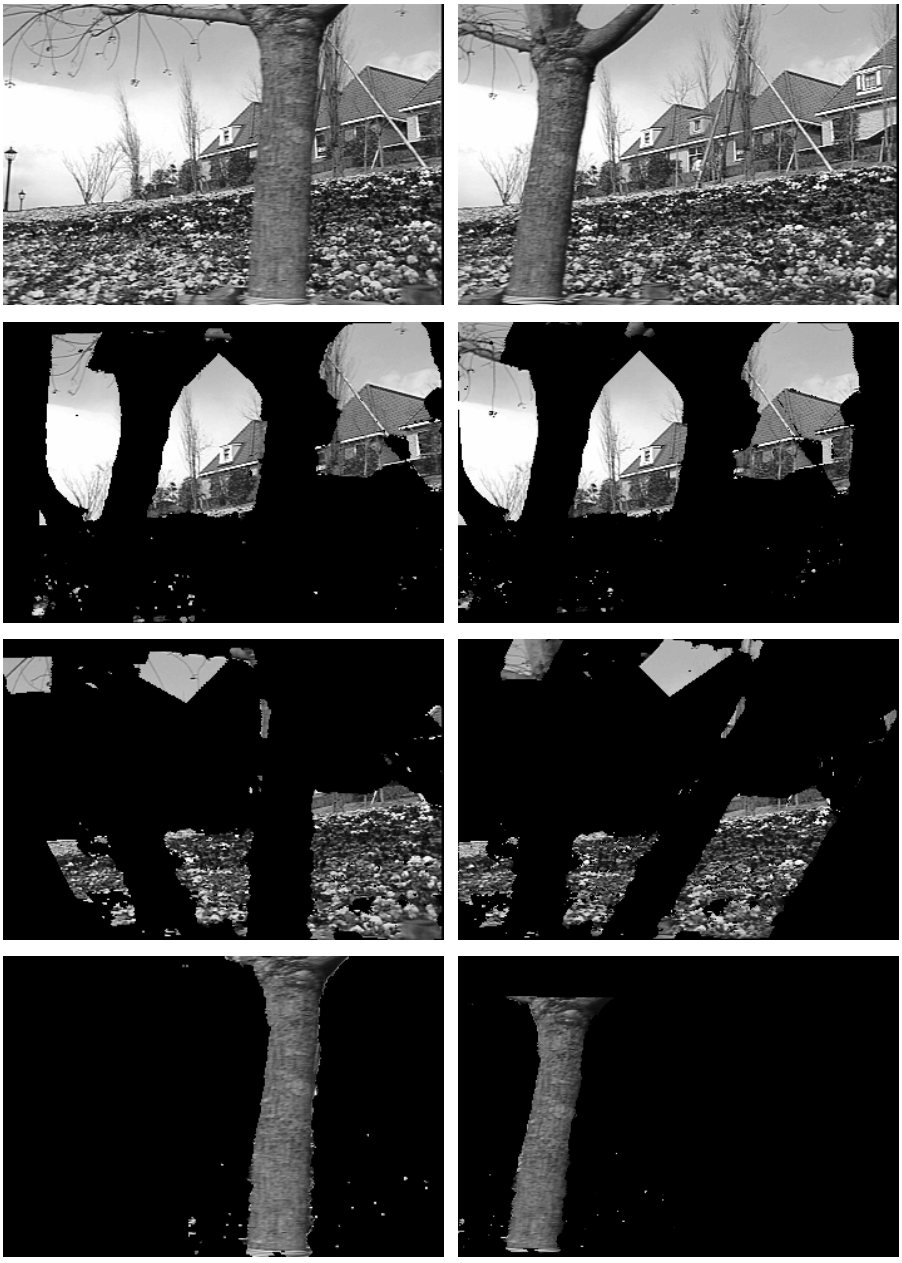


Figure 2.8 Flower Garden sequence. (Row 1) Original image pair of size 240×360, (Rows 2-4) Pixels assigned to warp layers 1-3 in I and I'.

Figure 2.8 shows two frames from a sequence that has been a benchmark for motion segmentation approaches for some time. Previously, only optical flow-based

techniques were able to get good motion segmentation results for this scene, however producing a segmentation of the motion between the two frames shown (1 and 30) would require using all (or at least most) of the intermediate frames. Here the only input to the system was the frame pair shown in Row 1. Notice that the portions of the house and the garden that were visible in both frames were segmented accurately as was the tree. This example shows the discriminative power of our filterbank as we were unable to detect the motion field correctly using correspondences found with the standard technique of normalized cross correlation. In addition, this example demonstrates the importance of the perturbed interest points and sampling based on feature crowdedness as the correct motions were not detected unless both of the techniques were used.

In Figure 2.9, a moving car passes behind a tree as the camera pans. Here, only two motion layers were recovered and they correspond to the static background and to the car. Since a camera rotating around its optical center produces no parallax for a static scene, the tree is in the same motion layer as the fence in the background, whereas the motion of the car requires its own layer. The slight rotation in depth of the car does not present a problem here.

2.1.5 Discussion

In this section we have presented a new method for performing dense motion segmentation and estimation in the presence of large inter-frame motion.

Like any system, our system is limited by the assumptions it makes. We make three assumptions about the scenes:

1. Identifiability
2. Constant appearance
3. Dominant planar motion.

A system is identifiable if its internal parameters can be estimated given the data. In the case of motion segmentation it implies that given a pair of images it is possible to recover the underlying motion. The minimal requirement under our chosen motion model is that each object present in the two scenes should be uniquely identifiable. Consider Figure 2.10; in this display, several motions can relate the two frames, and unless we make additional assumptions about the underlying problem, it is ill posed. Similarly in some of the examples we can see that while the segments closely match the individual objects in the scene, some of the background bleeds into each layer. Motion

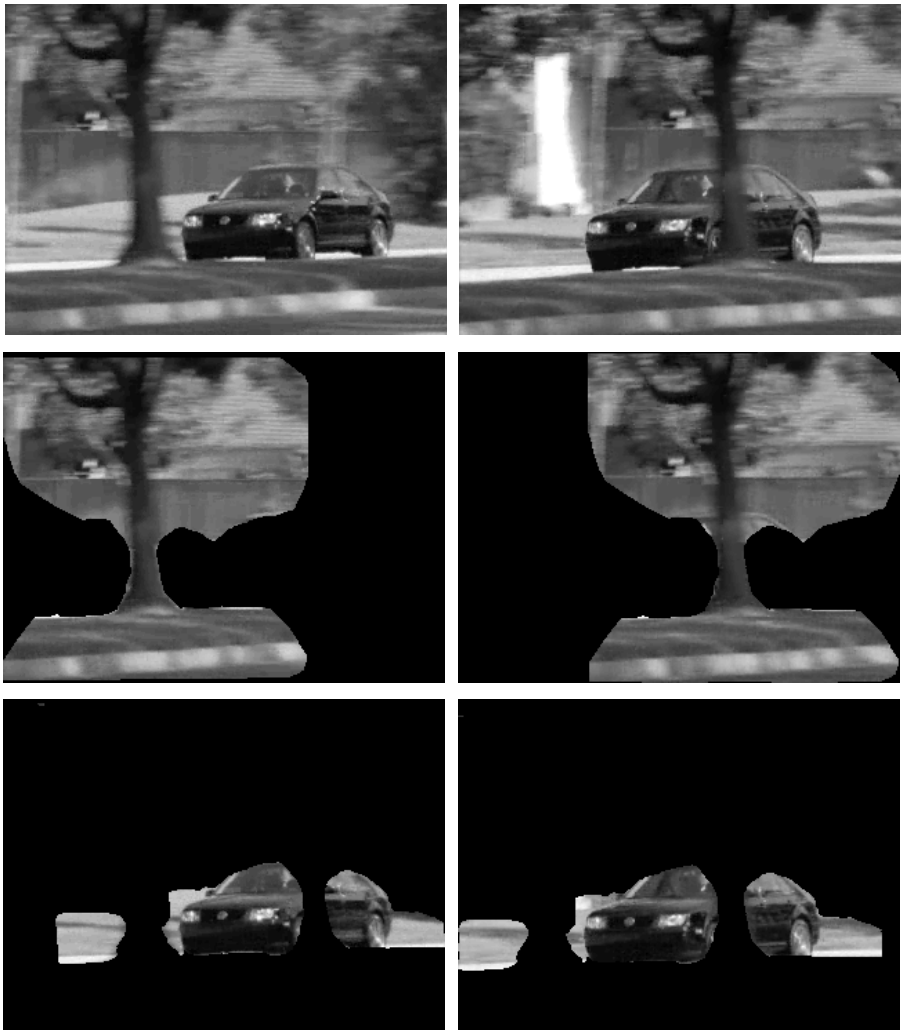


Figure 2.9 VW sequence. (Row 1) Original image pair of size 240×320, (Rows 2-3) Pixels assigned to warp layers 1-2 in I and I'.

is just one of several cues used by the human vision system in perceptual grouping and we cannot expect a system based purely on the cues of motion and brightness to be able to do the job. Incorporation of the various Gestalt cues and priors on object appearance will be the subject of future research.

Figure 2.10 Ternus Display. The motion of the dots is ambiguous; additional assumptions are needed to recover their true motion.

Our second assumption is that the appearance of an object across the two frames remains the same. While we do not believe that this assumption can be done away with completely, it can be relaxed. Our feature extraction, description, and matching is based

on a fixed set of filters. This gives us a limited degree of rotation and scale invariance. We believe that the matching stage of our algorithm can benefit from the work on affine invariant feature point description [76] and feature matching algorithms based on spatial propagation of good matches [66].

Our third assumption is that the individual motion fields are predominantly planar. This is not a strict requirement and is only needed insofar as we are able to obtain the initial planar fits. The actual motion estimate and segmentation is based on the more flexible spline based model.

2.1.6 Conclusion

In this section, we have presented a solution to the problem of motion segmentation for the case of large disparity motion and given experimental validation of our method. We have also presented an extension to handle non-planar/non-rigid motion as well as applications to automatic object deletion and structure from periodic motion. Our approach combines the strengths of the feature-based approaches (i.e., no limits on the disparity between frames) and the the direct, optical flow-based methods (i.e., provides a dense segmentation and correspondences).

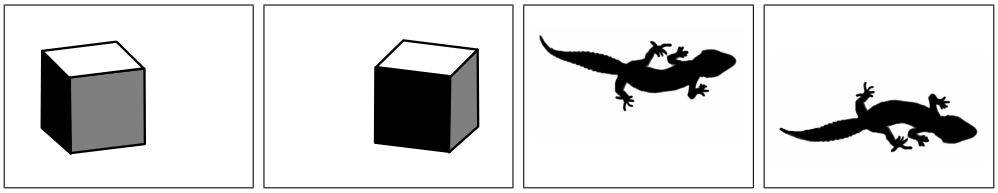


Figure 2.11 Non-planarity vs. non-rigidity: The left image pair shows a non-planar object undergoing 3D rigid motion; the right pair shows an approximately planar object undergoing non-rigid motion. Both examples result in residual with respect to a 2D planar fit.

2.2 Non-Planar Motion Segmentation

2.2.1 Introduction

Consider the image pairs illustrated in Figure 2.11. These have a significant component of planar motion but exhibit residual with respect to a planar fit because of either the non-planarity of the object (e.g. a cube) or the non-rigidity of the motion (e.g. a lizard). These are scenes for which the motion can be approximately described by a planar layer-based framework, i.e. scenes that have "shallow structure" [89]. In order to extend our approach to such scenes, we propose an additional stage consisting of a regularized spline model for capturing finer scale variations on top of an approximate planar fit. Our approach is related in spirit to the *deformotion* concept in [95], developed for the case of differential motion, which separates overall motion (a finite dimensional group action) from the more general deformation (a diffeomorphism).

It is important to remember that optical flow does not model the 3D motion of objects, but rather the changes in the image that result from this motion. Without the assumption of a rigid object, it is very difficult to estimate the 3D structure and motion of an object from observed change in the image, though there is recent work that attempts to do this [20, 21, 109–111, 118, 129]. For this reason, we choose to do all estimation in the image plane (i.e. we use 2D models), but we show that if the object is assumed to be rigid, the correspondences estimated can be used to recover the dense structure and 3D motion.

This approach extends the capabilities of feature-based scene matching algorithms to include dense optical flow without the limits on allowable motion associated with techniques based on differential optical flow. Previously, feature-based approaches could handle image pairs with large disparity and multiple independently moving objects, while optical flow techniques could provide a dense set of pixel correspondences

even for objects with non-rigid motion. However, neither type of approach could handle both simultaneously. Without the assumption of a rigid scene, existing feature-based methods cannot produce dense optical flow from the sparse correspondences, and in the presence of large disparity and multiple independently moving objects, differential optical flow (even coarse-to-fine) can break down. The strength of our approach is that dense optical flow can now be estimated for image pairs with large disparity, more than one independently moving object, and non-planar (including non-rigid) motion.

2.2.2 Related Work

The work related to our approach comes from the areas of motion segmentation, optical flow and feature-based (sparse) matching.

Several well known approaches to motion segmentation are based on dense optical flow estimation [6,13,82]; in these approaches the optical flow field was assumed to be piecewise smooth to account for discontinuities due to occlusion and object boundaries. Wang & Adelson introduced the idea of decomposing the image sequence into multiple overlapping layers, where each layer represents an affine motion field [120]. However their work was based on differential optical flow, which places strict limits on the amount of motion between two frames.

In [124], Weiss uses regularized radial basis functions (RBFs) to estimate dense optical flow; Weiss' method is based on the assumption that while the motion will not be smooth across the entire image, the motion is smooth within each of the layers. Given the set of spatiotemporal derivatives, he used the EM algorithm to estimate the number of motions, the dense segmentation and the dense optical flow. This work along with other spline-based optical flow methods [100, 101] however, also assumes differential motion and therefore does not apply for the types of sequences that we are considering.

In [107], Torr et al. show that the trifocal tensor can be used to cluster groups of sparse correspondences that move coherently. This work addresses similar types of sequences to those of our work in that it is trying to capture more than simply a planar approximation of motion, but it does not provide dense assignment to motion layers or dense optical flow. The paper states that it is an initialization and that more work is needed to provide a dense segmentation, however the extension of dense stereo assignment to multiple independent motions is certainly non-trivial and there is yet to be a published solution. In addition, this approach is not applicable for objects with non-rigid motion, as the fundamental matrix and trifocal tensor apply only to rigid motion.

Our work builds on the motion segmentation found via planar motion models as in [126], where planar transformations are robustly estimated from point correspondences in a RANSAC framework. A dense assignment of pixels to transformation layers is then estimated using an MRF. We refine the planar estimation produced by [126] using a regularized spline fit. Szeliski & Shum [101] also use a spline basis for motion estimation, however their approach has the same limitations on the allowable motion as other coarse-to-fine methods.

2.2.3 Our Approach for Non-planar Motion

When the scene contains objects undergoing significant 3D motion or deformation, the optical flow cannot be described by any single low dimensional image plane transformation (e.g., affine or homography). However, to keep the problem tractable we need a compact representation of these transformations; we propose the use of thin plate splines for this purpose. A single spline is not sufficient for representing multiple independent motions, especially when the motion vectors intersect [124]. Therefore we represent the optical flow between two frames as a set of disjoint splines. By disjoint we mean that the support of the splines are disjoint subsets of the image plane. The task of fitting a mixture of splines naturally decomposes into two subtasks: motion segmentation and spline fitting.

Ideally we would like to do both of these tasks simultaneously, however these tasks have conflicting goals. The task of motion segmentation requires us to identify groups of pixels whose motion can be described by a smooth transformation. Smoothness implies that each motion segment has the the same gross motion, however, except for the rare case in which the entire layer has exactly the same motion everywhere, there will be local variations. Hence the motion segmentation algorithm should be sensitive to inter-layer motion and insensitive to intra-layer variations. On the other hand, fitting a spline to each motion field requires attention to all the local variations. This is an example of different tradeoffs between bias and variance in the two stages of the algorithm. In the first stage we would like to exert a high bias and use models with a high amount of stiffness and insensitivity to local variations, whereas in the second stage we would like to use a more flexible model with a low bias.

We begin with the motion segmentation procedure of Section 2.1.2. The output of this stage, while sufficient to achieve a good segmentation, is not sufficient to recover the optical flow accurately. However, it serves two important purposes: firstly it provides

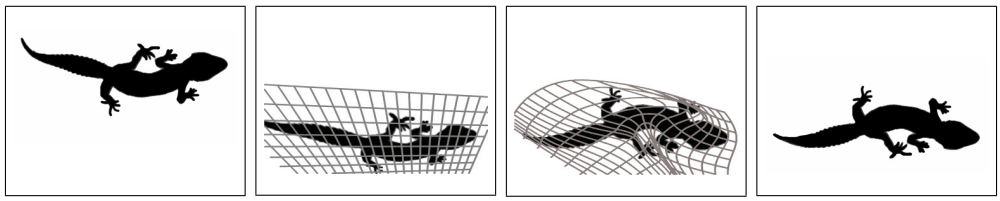


Figure 2.12 Determining Long Range Optical Flow. The goal is to provide dense optical flow from the first frame (1), to the second (4). This is done via a planar fit (2) followed by a flexible fit (3).

an approximate segmentation of the sparse correspondences that allows for coherent groups to be processed separately. This is crucial for the second stage of the algorithm as a flexible model will likely find an unwieldy compromise between distinct moving groups as well as outliers. Secondly, since the assignment is dense, it is possible to find matches for points that were initially mismatched by limiting the correspondence search space to points in the same motion layer. The second stage then bootstraps off of these estimates of motion and layer support to iteratively fit a thin plate spline to account for non-planarity or non-rigidity in the motion. Figure 2.12 illustrates this process.

We now describe the stages of the algorithm in detail.

Detecting Dominant Planar Motion

We begin by finding planar approximations of the motion in the scene as well as a dense assignment of pixels to motion layers. We use the motion segmentation algorithm presented in section 2.1. An example of this is shown in Figure 2.13

Example of Planar Fit and Segmentation

Figure 2.13 shows an example of the output from the planar fit and segmentation process. In this figure we show the two images, I and I', and the assignments for each pixel to a motion layer (one of the three detected motion fields). The columns represent the different motion fields and the rows represent the portions of each image that are assigned to a given motion layer. The motions are made explicit in that the pixel support from frame to frame is related exactly by a planar homography. Notice that the portions of the background and the dumpsters that were visible in both frames were segmented correctly, as was the man. The result of the spline fit for this example will be shown in Section 2.2.4.

Figure 2.13 Notting Hill sequence. (Row 1) Original image pair of size 311×552, (Row 2) Pixels assigned to warp layers 1-3 in I, (Row 3) Pixels assigned to warp layers 1-3 in I'.

Refining the Fit with a Flexible Model

The flexible fit is an iterative process using regularized radial basis functions, in this case Thin Plate Spline (TPS). The spline interpolates the correspondences to result in a dense optical flow field. This process is run on a per-motion layer basis.

Feature extraction and matching

During the planar motion estimation stage, only a gross estimate of the motion is required so a sparse set of feature points will suffice. In the final fit however, we would like to use as many correspondences as possible to ensure a good fit. In addition, since the correspondence search space is reduced (i.e. matches are only considered between pixels assigned to corresponding motion layers), matching becomes somewhat simpler. For this reason, we use the Canny edge detector to find the set of edge points in each of the frames and estimate correspondences in the same manner as in Section 2.1.2.

Iterated TPS fitting

Given the approximate planar homography and the set of correspondences between edge pixels, we would like to find the dense set of correspondences. If all of the correspondences were correct, we could jump straight to a smoothed spline fit to obtain

dense (interpolated) correspondences for the whole region. However, we must account for the fact that many of the correspondences are incorrect. As such, the purpose of the iterative matching is essentially to distinguish inliers from outliers, that is, we would like to identify sets of points that exhibit coherence in their correspondences.

One of the assumptions that we make about the scenes we wish to consider is that the motion of the scene can be approximated by a set of planar layers. Therefore a good initial set of inliers are those correspondences that are roughly approximated by the estimated homography. From this set, we use TPS regression with increasingly tighter inlier thresholds to identify the final set of inliers, for which a final fit is used to interpolate the dense optical flow. We now briefly describe this process.

Thin Plate splines are a family of approximating splines defined over $\mathbb{R}^d$. The theory for Thin Plate Splines was first developed by Duchon [29, 30] and subsequently by Meinguet [75]. Our presentation here follows follows Wahba [119].

Our task here is to construct smoothly varying functions that map pixel positions in one image to pixel positions in another. We use two splines, one each for the x and y mappings. Let $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ be the positions of the points in the first image. Let the target for the first spline be given by v_i and let f denote the transformation that we are trying to estimate. In two dimensions the smoothness penalty for Thin Plate Splines is given by

$$J_2 = \iint_{\mathbb{R}^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy$$

J_2 is also known as the bending energy. The minimization is performed over the space of functions χ whose partial derivatives of total order 2 are in $\mathcal{L}(R^2)$, i.e., the integral of square of every partial derivative of order 2 over $\mathbb{R}^2$ is bounded. Meinguet [75] provides a detailed description of this space. The functional $J_2(f)$ defines a semi-norm over χ.

The smoothing thin-plate spline is then defined to be the solution to the following variational problem:

$$\arg\min_{f \in \chi} \left[\frac{1}{n} \sum_{i}^{n} (v_i - f(x_i, y_i))^2 + \mu \iint_{\mathbb{R}^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy \right] \tag{2.2}$$

where the scalar μ is the tradeoff between fitting the target values v_i and the smoothness of the function f.

The null space of the penalty functional is a three dimensional space consisting of polynomials of degree less than or equal to one, i.e., the space of all functions spanned

by the basis functions

$$\phi_1(x,y) = 1, \qquad \phi_2(x,y) = x, \qquad \phi_3(x,y) = y$$

$$ax + by + c, \qquad a, b, c \in \mathbb{R}$$

Duchon [29] showed that if the points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ are such that the least squares regression on ϕ_1, ϕ_2, ϕ_3 is unique then the variational problem above has a unique solution f_μ and is given by

$$f_\mu(x,y) = \sum_i^n w_i G_i(x,y) + ax + by + c \tag{2.3}$$

Here $G(x,y)$ is the Green's function to the twice iterated Laplacian. It is also known as the fundamental solution to the bi-harmonic equation

$$\Delta^2 G = 0$$

where,

$$G_i(r) = r^2 \log r, \qquad r^2 = (x - x_i)^2 + (y - y_i)^2$$

Thus the calculation of the spline fit requires the estimation of the parameters w_i and a, b, c.

Now let K be an $n \times n$ matrix with entries given by

$$K_{ij} = G_i(x_j, y_j)$$

and let T be a $n \times 3$ matrix with rows given by

$$T_i = [\ x_i \quad y_i \quad 1\ ].$$

Also let $\mathbf{d}$ be the 3-vector

$$\mathbf{d} = [\ a \quad b \quad c\ ]^\top.$$

Then it can be shown that the optimal value for the coefficient vector $\mathbf{w} = [w_i]$ is given by the solution to the matrix equations [15, 88, 119]

$$(K + n\mu I)\,\mathbf{w} + T\mathbf{d} = 0 \tag{2.4}$$

$$T^{\top}\mathbf{w} = 0 \tag{2.5}$$

This is a simple linear system that can be solved using matrix inversion. For the case of $\mu = 0$ we obtain the interpolating Thin Plate Spine.

The complexity of matrix inversion scales as $O(n^3)$ in the number of rows. Thus as the number of points that we are fitting to goes up it is not practical to use these methods on the full dataset. In our experiments we take a naive subsampling based approach. Out of the 1200 points that we are required to fit to, we randomly subsampled 500 points and used them as the landmarks for the spline fitting procedure. We observe that this simple approach works well in practice. A number of researchers have explored more sophisticated and computationally attractive approaches to the problem [28, 41, 94, 124]. Any of these can be used as a replacement for our spline estimation procedure.

We estimate the TPS mapping from the points in the first frame to those in the second where μ_t is the regularization factor for iteration t. The fit is estimated using the set of correspondences that are deemed inliers for the current transformation, where τ_t is the threshold for the t^{th} iteration. After the transformation is estimated, it is applied to the entire edge set and the set of correspondences is again processed for inliers, using the new locations of the points for error computation. This means that some correspondences that were outliers before may be pulled into the set of inliers and vice versa. The iteration continues on this new set of inliers where $\tau_{t+1} \leq \tau_t$ and $\mu_{t+1} \leq \mu_t$. We have found that three iterations of this TPS regression with incrementally decreasing regularization and corresponding outlier thresholds suffices for a large set of real world examples. Additional iterations produced no change in the estimated set of inlier correspondences.

This simultaneous tightening of the pruning threshold and annealing of the regularization factor aid in differentiating between residual due to localization error or mismatching and residual due to the non-planarity of the object in motion. When the pruning threshold is loose, it is likely that there will be some incorrect correspondences that will pass the threshold. This means that the spline should be stiff enough to avoid the adverse effect of these mismatches. However, as the mapping converges we place higher confidence in the set of correspondences passing the tighter thresholds. This process is similar in spirit to iterative deformable shape matching methods [9, 22].

I. **Estimate planar motion**
1. Find correspondences between I and I'
2. Robustly estimate the motion fields
3. Densely assign pixels to motion layers

II. **Refine the fit with a flexible model**
4. Match edge pixels between I and I'
5. For t=1:3
6. Fit all correspondences within τ_t using TPS regularized by μ_t
7. Apply TPS to set of correspondences

Note: $(\tau_{t+1} \leq \tau_t, \mu_{t+1} \leq \mu_t)$

Figure 2.14 Algorithm Summary

2.2.4 Experimental Results

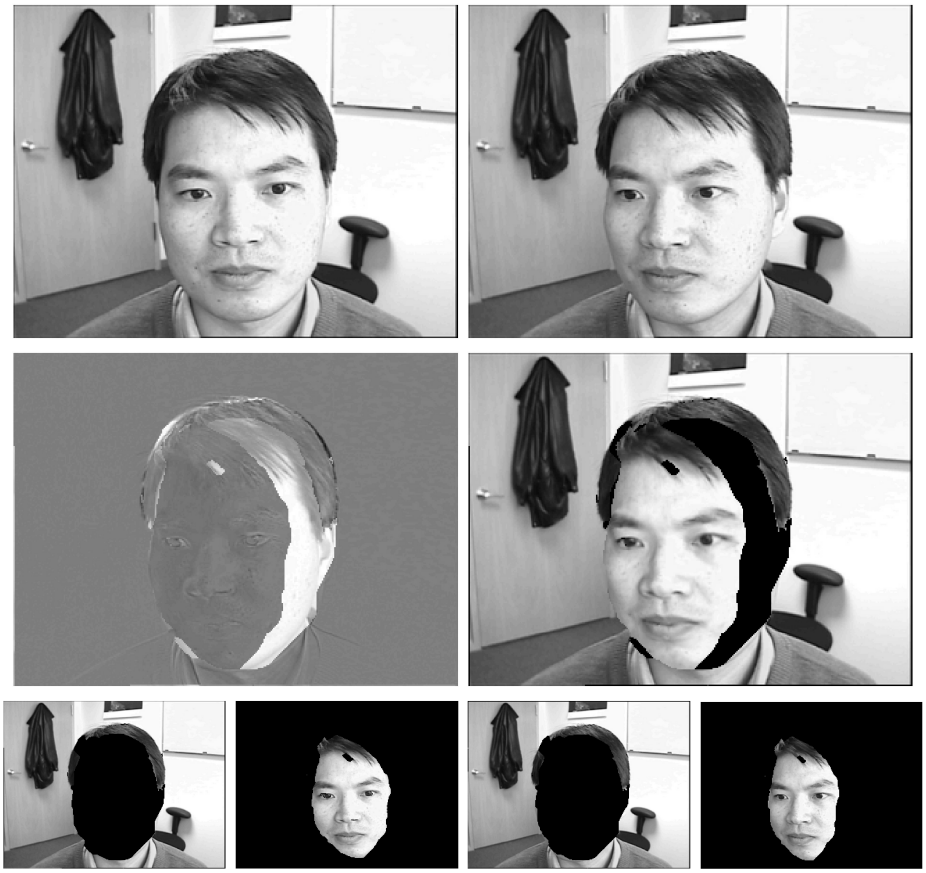


Figure 2.15 Face Sequence. (Row 1) The two input images, I and I' of size 240×320. (Row 2) The difference image is show first where grey regions indicate zero error regions and the reconstruction, $\mathcal{T}(I)$ is second. (Row 3) The initial segmentation found via planar motion.

We now illustrate our algorithm, which is summarized in Figure 2.14, on several pairs of images containing objects undergoing significant, non-planar motion. Since the motion is large, displaying the optical flow as a vector field will result in a very confusing

figure. Because of this, we show the quality of the optical flow in other ways, including (1) examining the image and corresponding reconstruction error that result from the application of the estimated transform to the original image (we refer to this transformed image as $\mathcal{T}(I)$), (2) showing intermediate views (as in [92]), or by (3) showing the 3D reconstruction induced by the set of dense correspondences. Examples are presented that exhibit either non-planarity, non-rigidity or a combination of the two. We show that our algorithm is capable of providing optical flow for pairs of images that are beyond the scope of existing algorithms. We performed all of the experiments on grayscale images using the same parameters[2].

Face Sequence The first example is shown in Figures 2.15 and 2.16. The top row of Figure 2.15 shows the two input frames, I and I', in which a man moves his head to the left in front of a static scene (the nose moves more than 10% of the image width). The second row shows first the difference image between $\mathcal{T}(I)$ and I' where error values are on the interval [-1,1] and gray regions indicate areas of zero error. This image is followed by $\mathcal{T}(I)$; this image has two estimated transformations, one for the face and another for the background. Notice that error in the overlap of the faces is very small, which means that according to reconstruction error, the estimated transformation successfully fits the relation between the two frames. This transformation is non-trivial as seen in the change in the nose and lips as well as a shift in gaze seen in the eyes, however all of this is captured by the estimated optical flow. The final row in Figure 2.15 shows the segmentation and planar approximation from Section 2.1.2, where the planar transformation is made explicit as the regions' pixel supports are related exactly by a planar homography. Dense correspondences allow for the estimation of intermediate views via interpolation as in [92]. Figure 2.16 shows the two original views of the segment associated with the face as well as a synthesized intermediate view that is realistic in appearance. The second row of this figure shows an estimation of relative depth that comes from the disparity along the rectified horizontal axis. Notice the shape of the nose and lips as well as the relation of the eyes to the nose and forehead. It is important to remember that no information specific to human faces was provided to the algorithm for this optical flow estimation.

Notting Hill Sequence The next example shows how the spline can also refine what is already a close approximation via planar models. Figure 2.17 shows a close up of the planar error image, the reconstruction error and finally the warped grid

[2] $k = 2, \lambda = .285, \tau_p = 15, \mu_1 = 50, \mu_2 = 20, \mu_3 = 1, \tau_1 = 15, \tau_2 = 10, \tau_3 = 5$. Here, k, λ, and τ_p refer to parameters in Section 2.1.2.

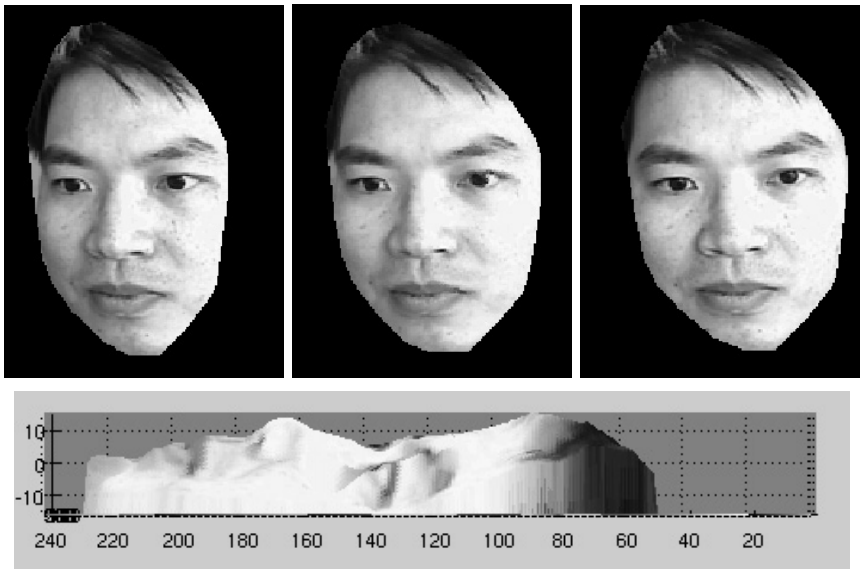


Figure 2.16 Face Sequence – Interpolated views. (Row 1) Original frame I', synthesized intermediate frame, original frame I, (Row 2) A surface approximation from computed dense correspondences.

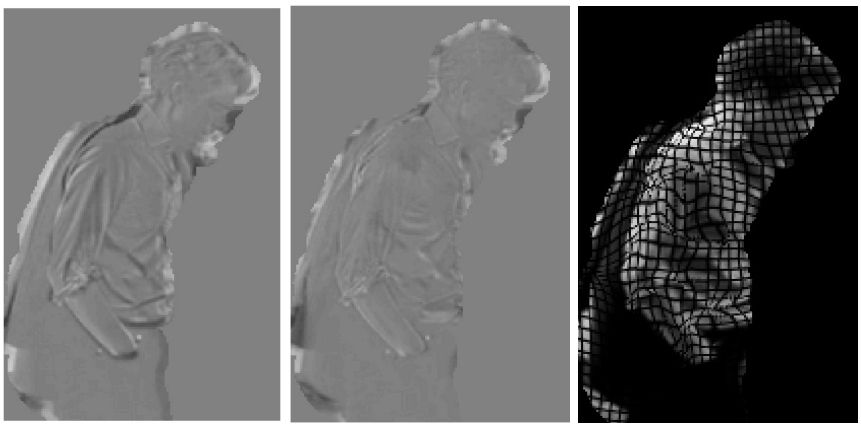


Figure 2.17 Notting Hill. Detail of the spline fit for a layer from Figure 2.6, difference image for the planar fit, difference image for the spline fit, grid transformation.

for the scene that was shown in Figure 2.6. The planar approximation was not able to capture the 3D nature of the clothing and the non-rigid motion of the head with respect to the torso, however the spline fit captures these things accurately.

Gecko Sequence The second example, shown in Figure 2.18, displays a combination of a non-planar object (a gecko lizard), undergoing non-rigid motion. While this is a single object sequence, it shows the flexibility of our method to handle complicated motions. In Figure 2.18(1), the two original frames are shown as well as a synthesized intermediate view (here, intermediate refers to time rather than viewing direction since

we are dealing with non-rigid motion) . The synthesized image is a reasonable guess at what the scene would look like midway between the two input frames. Figure 2.18(2) shows $\mathcal{T}(I)$ as well as the reconstruction error for the spline fit $(\mathcal{T}(I) - I')$, and the error incurred with the planar fit. We see in the second row of Figure 2.18(2) that the tail, back and head of the gecko are aligned very well and those areas have negligible error. When we compare the reconstruction error to the error induced by a planar fit, we see that the motion of the gecko is not well approximated by a rigid plane. Here, there is also some 3D motion present in that the head of the lizard changes in both direction and elevation. This is captured by the estimated optical flow.

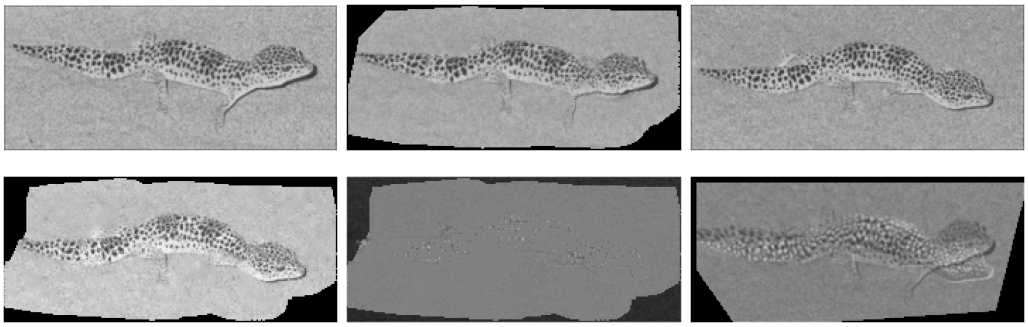


Figure 2.18 Gecko Sequence. (Row 1) Original frame I of size 102×236, synthesized intermediate view, original frame I'. (Row 2) $\mathcal{T}(I)$, Difference image between the above image and I' (gray is zero error), Difference image for the planar fit.

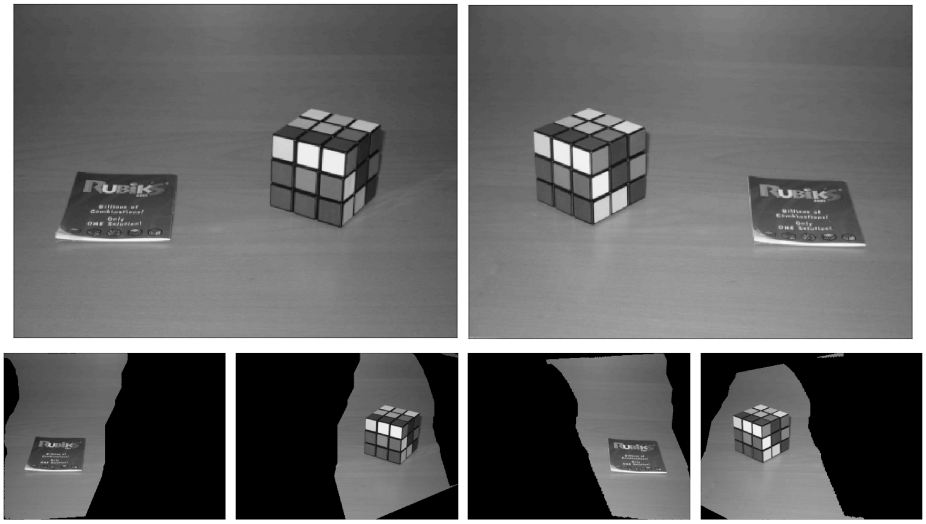


Figure 2.19 Rubik's Cube. (Row 1) Original image pair of size 300×400, (Row 2) assignments of each image to layers 1 and 2.

Rubik's Cube The next example shows a scene with rigid motion of a non-

planar object. Figure 2.19 displays a Rubik's cube and user's manual switching places as the cube rotates in 3D. Below the frames, we see the segmentation that is a result of the planar approximation. As can be seen the segmentation contains large chunks of the background along with the Rubik's Cube. While it is indeed desirable that the only pixels that we segment are those belonging to the Rubik's Cube, we must note that the background lacks any distinguishing features making its motion truly ambiguous. Hence without additional knowledge about the objects in the scene, any prior that we place on the scene while segmenting it will be the cause of some mistakes. In our MRF-based segmentation scheme we make the assumption that the layers are spatially contiguous, this coupled with the motion ambiguity mentioned earlier results in some portion of the background being interpreted as belonging to the same layer as the Rubik's cube. Figure 2.20 shows $\mathcal{T}(I)$, the result of the spline fit applied to this same scene. The first row shows a detail of the two original views of the Rubik's cube as well as a synthesized intermediate view. Notice that the rotation in 3D is accurately captured and demonstrated in this intermediate view. The second row shows the reconstruction errors, first for the planar fit and then for the spline fit, followed by $\mathcal{T}(I)$. Notice how accurate the correspondence is since the spline applied to the first image is almost identical to the second frame.

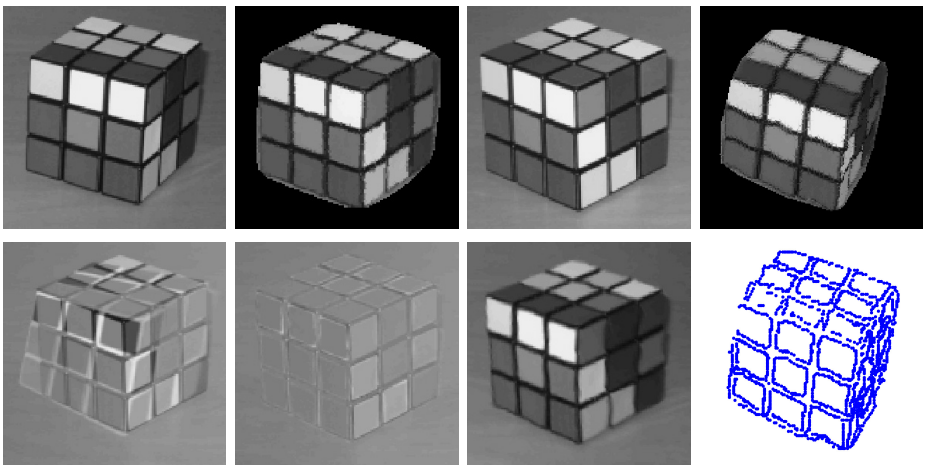


Figure 2.20 Rubik's Cube – Detail. (Row 1) Original frame I, synthesized intermediate frame, original frame I', A synthesized novel view, (Row 2) difference image for the planar fit, difference image for the spline fit, $\mathcal{T}(I)$, the estimated structure shown for the edge points of I. We used dense 3D structure to produce the novel view.

Correspondences between portions of two frames that are assumed to be projections of rigid objects in motion allow for the recovery of the structure of the object, at least up to a projective transformation. In [103], the authors show a sparse point-set

from a novel viewpoint and compare it to a real image from the same viewpoint to show the accuracy of the structure. Figure 2.20 shows a similar result, however since our correspondences are dense, we can actually render the novel view that validates our structure estimation. The novel viewpoint is well above the observed viewpoints, yet the rendering as well as the displayed structure is fairly accurate. Note that only the set of points that were identified as edges in I are shown; this is not the result of simple edge detection on the rendered view. We use this display convention because the entire point-set is too dense to allow the perception of structure from a printed image. However, the rendered image shows that our estimated structure was very dense. It is important to note that the only assumption that we made about the object is that it is a rigid, piecewise smooth object. To achieve similar results from sparse correspondences would require additional object knowledge, namely that the object in question is a cube and has planar faces. It is also important to point out that this is not a standard stereo pair since the scene contains multiple objects undergoing independent motion.

2.2.5 Discussion

Since splines form a family of universal approximators over $\mathbb{R}^2$ and can represent any 2D transformation to any desired degree of accuracy, it raises the question as to why one needs to use two different motion models in the two stages of the algorithm. If one were to use the affine transform as the dominant motion model, splines with an infinite or very large degree of regularization can indeed be used in its place. However, in the case where the dominant planar motion is not captured by an affine transform and we need to use a homography, it is not practical to use a spline. This is so because the set of homographies over any connected region of $\mathbb{R}^2$ are unbounded, and can in principal require a spline with an unbounded number of knots to represent an arbitrary homography. So while a homography can be estimated using a set of four correspondences, the corresponding spline approximation can, in principle, require an arbitrarily large number of control points. This poses a serious problem for robust estimation procedures like RANSAC since the probability of hitting the correct model decreases exponentially with increasing degrees of freedom.

Many previous approaches for capturing long range motion are based on the fundamental matrix. However, since the fundamental matrix maps points to lines, translations in a single direction with varying velocity and sign are completely indistinguishable, as pointed out, e.g. by [107]. This type of motion is observed frequently in motion

sequences. The trifocal tensor does not have this problem; however, like the fundamental matrix, it is only applicable for scenes with rigid motion and there is not yet a published solution for dense stereo correspondence in the presence of multiple motions.

2.3 Applications

2.3.1 Automatic Object Removal

We demonstrate an application of our algorithm to the problem of video object deletion in the spirit of [53,120]; see Figure 2.21. The idea of using motion segmentation information to fill in occluded regions is not new, however previous approaches require a high frame rate to ensure that inter-frame disparities are small enough for differential optical flow to work properly. Here the interframe disparities are as much as a third of the image width.

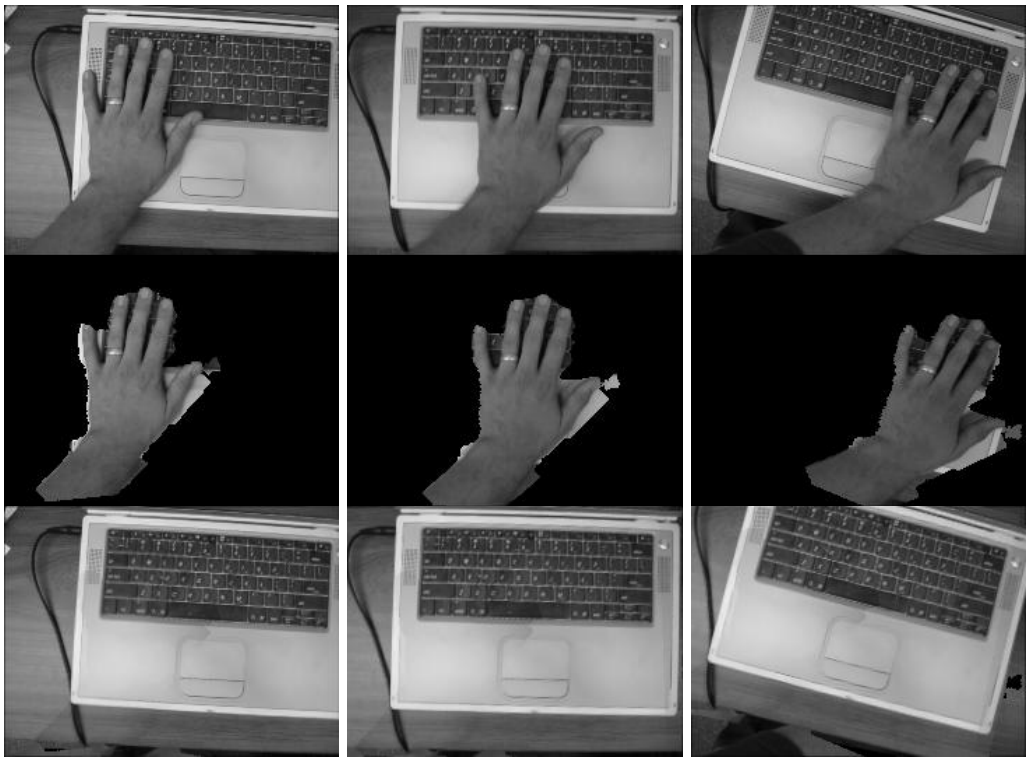


Figure 2.21 Illustration of video object deletion. (1) Original frames of size 180×240. (2) Segmented layer corresponding to the motion of the hand. (3) Reconstruction without the hand layer using the recovered motion of the keyboard. Note that no additional frames beyond the three shown were used as input.

2.3.2 Structure From Periodic Motion

We show how to exploit temporal periodicity of moving objects to perform 3D reconstruction. The collection of period-separated frames serve as a surrogate for multiple rigid views of a particular pose of the moving target, thus allowing the use of standard techniques of multiview geometry. We motivate our approach using human motion cap-

ture data, for which the true 3D positions of the markers are known. We next apply our approach to image sequences of pedestrians captured with a camcorder. Applications of our proposed approach include 3D motion capture of natural and manmade periodic moving targets from monocular video sequences.

Introduction

Periodic motion is ubiquitous in the physical world, from the oscillations of a pendulum to the gallop of a horse. The periodicity of moving objects such as pedestrians has been widely recognized as a cue for salient object detection in the context of tracking and surveillance, see for example [25,67]. In this section we focus on the use of periodicity for a different and, to our knowledge, novel purpose: 3D reconstruction. The key idea is very simple. Given a monocular video sequence of a periodic moving object, any set of period-separated frames represents a collection of snapshots of a particular pose of the moving object from a variety of viewpoints. This is illustrated in Figure 2.22. Thus each complete period in time yields one view of each pose assumed by the moving object, and by finding correspondences in frames across neighboring periods in time, one can apply standard techniques of multiview geometry, with the caveat that in practice such periodicity is only approximate. In this section we present this idea and apply it to the problem of estimating sparse 3D structure and dense disparity for walking humans.

The organization of this section is as follows. We review related work in Section 4.2. In Section 2.3.2 we discuss our approach. Experimental results appear in Section 4.3, and we conclude and discuss future work in Section 2.3.2.

Related Work

Periodicity is a kind of symmetry, and as such, its use in recovering 3D information is related to approaches that leverage other kinds of symmetry. An early example of work in this vein is Kanade's method of recovering 3D shape from a single view of a skew symmetric object [55]; more recent extensions of these ideas appear in [38, 43]. The periodicity we are concerned with is temporal; in contrast, spatial periodicity (together with homoegeneity and isotropy) has been exploited in several shape-from-texture approaches, e.g. [40,70], in which the periodicity pertains to texture elements on the surface of a curved object. While the periodicity of walking humans and animals has indeed been used for other purposes, e.g. pedestrian detection [25], to our knowledge the present work is the first to exploit it for 3D reconstruction.

Figure 2.22 Illustration of periodic motion for a walking person. Equally spaced frames from one second of footage are shown. The pose of the person is approximately the same in the first and last frames, but the position relative to the camera is different. Thus this pair of frames can be treated approximately as a stereo pair for purposes of 3D structure estimation. Note that while the folds in the clothing change over time, their temporal periodicity makes them rich features for correspondence recovery across periods.

Our Approach

In this section we describe our approach to estimating structure from periodic motion (SFPM). In illustrating the idea, we make use of motion capture (or *mocap*) data from [96]. We provide experimental results on regular video sequences in the following section.

Estimating the Period

In the present work we specify the period of the moving target manually. A number of approaches exist for estimating the period of a walking figure, e.g. [25]. As our focus is on the reconstruction problem, we have not investigated the use of these algorithms, though we do address the issue of error in the period estimation step in Section 4.3.

Multiview Geometry across Periods

The most elementary configuration for periodic structure from motion is the case of two views separated in time by one period. As is well known from [31, 46], the 3D structure of a rigid object can be estimated up to a projective transformation from two uncalibrated views. The periodic motion counterpart to this is illustrated in Figure 2.23(a,b), which depicts two 2D views of mocap data spaced apart one period T_o in time.

In this case, the camera is stationary and the walking figure has translated and rotated relative to the camera over the course of the period. These two views correspond approximately to a stereo pair of a particular pose of the walking figure. The

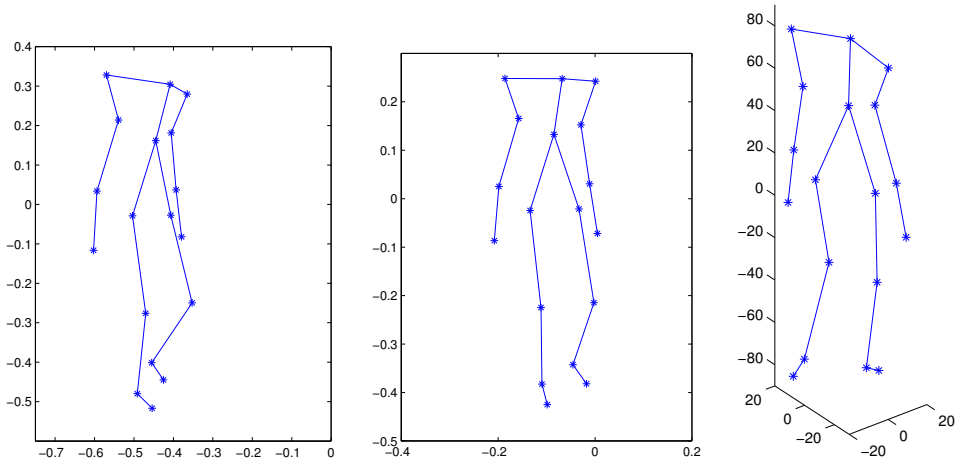


Figure 2.23 Illustration of structure from periodic motion using motion capture data: (a) view at time t, (b) view at time $t + T_o$, (c) 3D reconstruction from T_o-separated views.

reconstruction obtained from these two views is shown in Figure 2.23(c). Since we are using uncalibrated cameras, the reconstruction is arbitrary up to a 3D homography; our display shows the reconstruction using a least-squares homography estimated using the ground truth marker positions. Alternatively, if three or more views are available, one can employ autocalibration techniques such as [69]. Partial calibration information can also be obtained from knowledge about the scene (see e.g. [47] Ch. 18) or from known properties of the moving target, e.g. that it is a human of a certain aspect ratio.

As is the case in standard structure from motion (SFM), the underlying geometry is only part of the problem: one must solve for correspondences between views before estimating the structure.

Solving for Correspondences

In real video sequences, for which identified features are not available as in the mocap data, we can appeal to methods of interest point detection and correspondence recovery that are used in conventional SFM. In particular, we use a RANSAC-based approach [105] on interest points extracted using the Förstner operator [37]. We perform interest point description and matching using the method of [126], which uses the L_1-norm on the error between vectors of filter responses computed at each interest point.

In using RANSAC to estimate the epipolar geometry, we assume that the feature points on the moving object dominate those in the rest of the scene. Because of this simplification, we do not need a separate figure/ground motion segmentation step as preprocessing.

Computing Dense Disparity

Once the epipolar geometry is known for an image pair, a number of dense stereo correspondence algorithms can be applied along the epipolar lines. In this work we use the method of [60], which is an energy minimization based method using a graph cut approximation. The input to the algorithm is a pair of rectified images (with respect to the object of interest) and the output is a disparity array. For rectification, we use the algorithm described in [47], Sec. 10.12.

Experiments

Walking Person I

Figure 2.24(c) shows the sparse 3D structure recovered for the T_o-separated frames of a walking person shown in Figure 2.24(a,b). A detail of the head and left shoulder region is shown in Figure 2.24(d) from a viewpoint behind the person and slightly to the left. Here we can see that the qualitative shape of the head relative to the sleeve region is reasonable.

The set of points used here consists of (i) the Förstner interest points used to estimate the fundamental matrix and (ii) the neighboring Canny edges with correspondences consistent with the epipolar geometry. Many points appear around the creases in the clothing, but this leaves several blank patches around the lower shirt and the arm.

Walking Person II

In Figure 2.25 we show an example of dense disparity estimation for another T_o-separated frame pair of a walking person. The input frames are shown at the top, followed by the rectified image pair. The estimated disparity relative to the left rectified image is shown next; for purposes of visualization, in this figure we have manually masked out the region corresponding to the person. The disparities are shown as a gray level, with lighter shades indicating larger disparity. We observe that the individual's right leg has higher disparity than the left leg, which is consistent with their depth ordering relative to the image plane, and that the majority of the disparity estimates for the rest of the body fall somewhere in between these values. In the original image pair, the light colored top of the forearm bleeds into the bright background; this corrupts the disparity estimate in that region.

Sensitivity Study To conclude our experiments, we examine the sensitivity of the 3D reconstruction with respect to errors in the estimate of T_o. For this purpose, we again make use of the mocap data from Section 2.3.2.

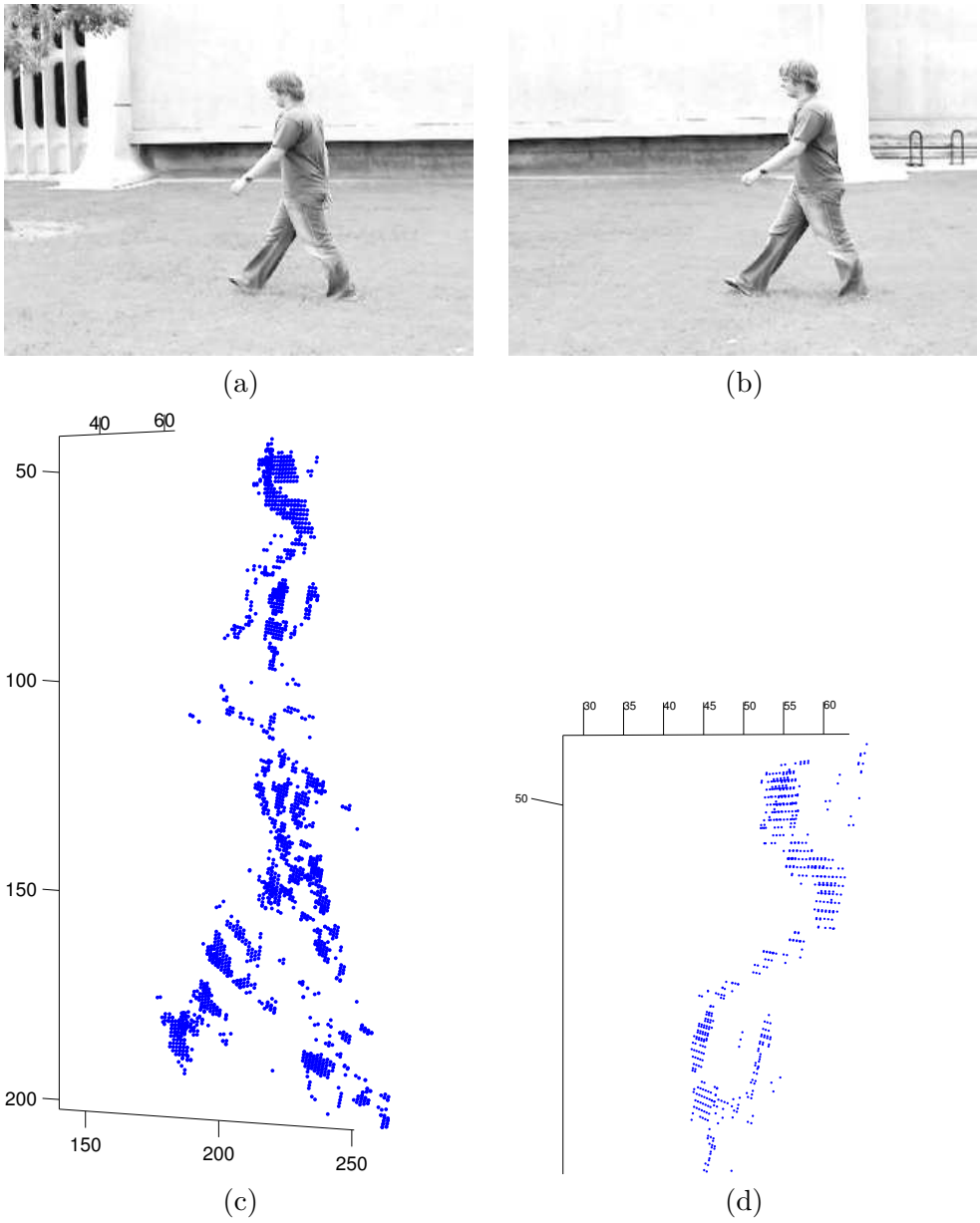


Figure 2.24 (a,b) T_o-separated input frames. (c) Estimated 3D structure for interest points. (d) Detailed view of head and shoulder region viewed from behind the person.

We consider 200 frames of a regular walking sequence captured at 60 fps with $T_o \approx 90$ frames [96]. Each frame is a 2D projection (cf. Figure 2.23(a,b)) of the recorded 3D positions (which are accurate to 1mm) of a set of markers rigidly attached to a subject's body. We selected a different 2D projection of frame 100 as a reference view. Using the reference view together with each of the previously mentioned 200 views, we computed the 3D reconstruction and the root-mean-square (RMS) error relative to the

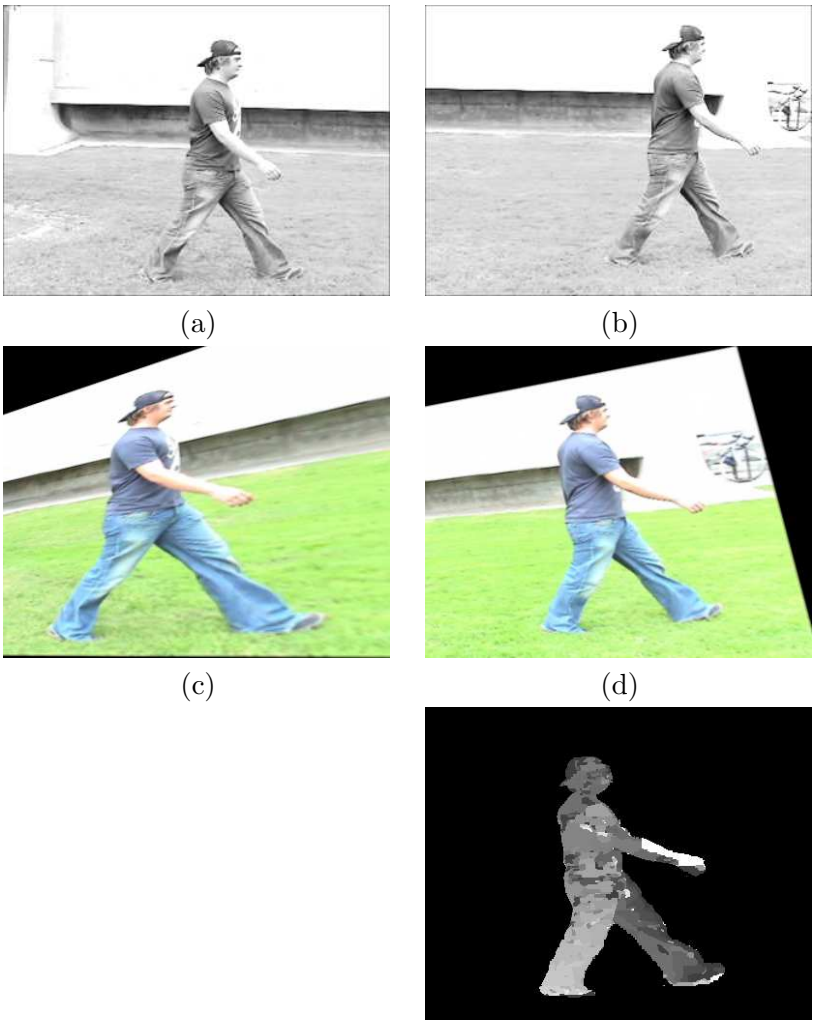
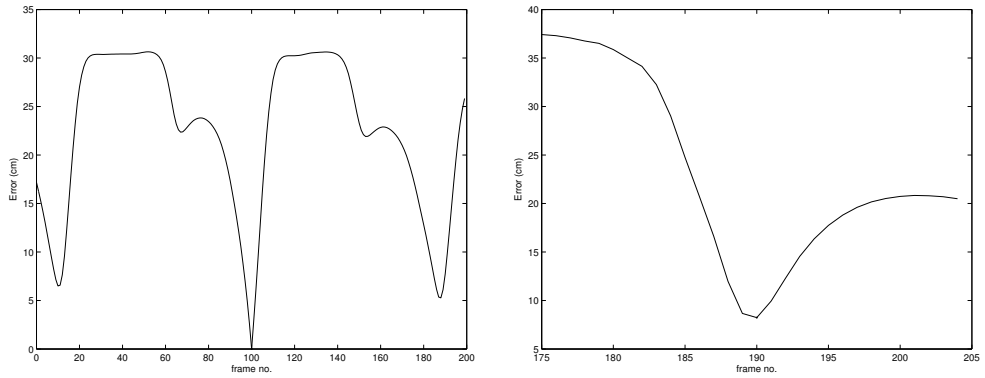


Figure 2.25 (a,b) T_o-separated input frames. (c,d) Rectified images computed with respect to estimated epipolar geometry of input frames. (e) Estimated disparity, masked out to show region of interest containing the person.

known 3D structure at the reference frame.

The error, which is plotted in Figure 2.26(a), is computed after solving for the least-squares homography aligning the projective reconstruction with the ground truth marker positions at the reference frame. The periodicity is evident in the dips that occur at ±90 frames on either side of 100. As expected, the error drops to zero at frame 100, at which point the reconstruction problem reduces to the case of exact stereo. The plot in Figure 2.26(b) shows a detail of the reconstruction error computed for 30 frames centered around frame 190; again the reference view is frame 100, but here the cameras specifying the 2D projections are the same for all the views. In each plot, it is evident



(a) (b)

Figure 2.26 Reconstruction error vs. frame number for mocap data of a walking person with $T_o \approx 90$ frames. (a) RMS error in units of cm between true 3D coordinates at frame 100 and the estimated 3D coordinates using one 2D view at frame 100 and a different 2D view at each of frames 1-200. (b) RMS error for frames 175-205 relative to frame 100, this time using the same 2D view for the reference frame as for frames 1-200.

that the error grows gradually with respect to displacements around the local optimum.

Conclusion and Future Work

We have presented an approach to 3D structure estimation based on monocular views of periodic motion. We demonstrated this approach using motion capture data and raw footage of pedestrians. Using the motion capture data, we explored the behavior of the reconstruction with respect to errors in the period estimation step.

The weakest part of the system is currently the correspondence estimation step. In theory, by the definition of periodicity, the problem treated in this work is identical to the classical SFM problem, provided the period estimate is correct. However, in practice, the correspondence problem is at least as hard as the usual stereo correspondence problem, and is in general harder, due to appearance variations across periods. In this regard, the correspondence problem associated with the SFPM problem lies somewhere in between the classical correspondence problem of wide-baseline stereo and the feature correspondence problem in 3D object recognition. We could therefore benefit from the use of methods designed with the latter problem in mind; in work with Ivan Laptev and Patrick Pérez [64], we have extended this approach to structure from periodic motion by incorporating spatio-temporal interest points.

2.4 Conclusion

In this section we begin with a summary of the chapter, followed by a discussion of the primary contributions of the work presented in this chapter and finally a discussion on the limitations of our method as well as future work.

2.4.1 Summary

In this chapter, we have presented a solution to the problem of motion segmentation for the case of large disparity motion and given experimental validation of our method. We have also presented an extension to handle non-planar/non-rigid motion as well as applications to automatic object deletion and structure from periodic motion. Our approach combines the strengths of the feature-based approaches (i.e., no limits on the disparity between frames) and the the direct, optical flow-based methods (i.e., provides a dense segmentation and correspondences).

2.4.2 Primary Contributions

Motion Segmentation Framework

The contribution of this work is primarily the framework proposed for large disparity motion that breaks the problem into three distinct stages:

1. **Compute Point Correspondences** This stage takes as input the two frames and outputs a list of point correspondences where each entry in the list specifies the location of a real world point in each of the two frames. This list will be very noisy as point correspondences for large disparity images are very difficult to estimate. In our work we use filter-based matching for corners to generate the correspondences, however any matching engine may be seamlessly substituted (e. g. the SIFT operator [68]).

2. **Estimate Motion Fields** This stage takes as input the list of point correspondences from stage 1 and outputs a list of motions that are present in the set of correspondences. This estimation should be robust to noise in the set of point correspondences and the list of motions should be ordered according to support in the set of correspondences. We use a RANSAC-based approach that has modifications to prune duplicate motions that are present.

3. **Densely Assign Pixels to Motion Layers** The final stage takes as input the two frames as well as the set of detected motions. The goal is to find a segmentation the minimizes the reconstruction error – how well the first image approximates the second when the motions are applied to the pixels in a given layer – while preserving the monocular smoothness in each of the frames – neighboring pixels with similar brightness are likely members of the same object. We use a graph cut-based algorithm that takes an explicit parameter to tradeoff the reconstruction error and the smoothness of the assignment.

Pseudo-code for our approach to planar motion segmentation appears in Appendix A.1.

Duplicate Motion Pruning

In order to avoid the effects of phantom motions we use a two step variant of RANSAC, where multiple independent motions are explicitly handled, as duplicate transformations are detected and pruned in a greedy manner. The first step provides a rough partitioning of the set of correspondences (motion identification) and the second takes this partitioning and estimates the motion of each group (motion refinement).

First, a set of planar warps is estimated by a round of standard RANSAC and inlier counts (using an inlier threshold of τ) are recorded for each transformation. In our case, we use planar homography which requires 4 correspondences to estimate, however similarity or affinity may be used (requiring 2 and 3 correspondences, respectively). The estimated list of transformations is then sorted by inlier count and we keep the first n_t transformations, where n_t is some large number (e.g. 300).

We expect that the motions in the scene will likely be detected multiple times and we would like to detect these duplicate transformations. Comparing transformations in the space of parameters is difficult for all but the simplest of transformations, so we compare transformations by comparing the set of inliers associated with each transformation. If there is a large overlap in the set of inliers (more than 75%) the transformation with the larger set of inliers is kept and the other is pruned.

Techniques for Feature-Impoverished Objects

Since our approach estimates motion based on consistency in point correspondences, finding the motion of an object containing many identifiable features is quite easy. The problem becomes more difficult for objects that have fewer features. We have

introduced two new techniques for detecting the motion of objects with few features even in complicated and feature-rich scenes.

The first of these techniques – Perturbed Interest Points – adds features in the neighborhood of an interest point and allows matches to be found that will support the true motion of the object, and the second – Sampling Based on Feature Crowdedness – increases the likelihood of features with few neighbors being chosen during the RANSAC stage of our algorithm.

2.4.3 Subsequent Work

Subsequent work has extended the framework presented in this chapter to handle objects exhibiting 3 dimensional motion as well as to handle occlusion explicitly.

In [12], Bhat et al. extend our approach to handle rigid objects exhibiting non-planar motion by using both planar homographies and fundamental matrices. They begin with point correspondences from SIFT features [68] and during the motion estimation stage the motion model (homography vs. fundamental matrix) is decided automatically by using the simpler of the two that fits the set of correspondences in question (this seems like a good idea with a lot of potential for mistakes - fmtx is too loose). Pixel assignment is decided in the same manner as in our work though the matches based on Fundamental matrices have potential matches along a line. To deal with this ambiguity without creating a large search space, the authors use a multi-scale approach to determine the window size for matches for each motion layer.

In [130], Xiao and Shah employ a similar framework to ours for motion segmentation that is aimed at intelligently handling occlusion. They work in a different domain in that the sequences are taken from video and do not exhibit large inter-frame disparity. The assignment to motion layers is achieved using a similar graph-cut based approach as in our case, but the assignment is done using a set of frames where the number of frames is automatically determined to ensure that individual pixel disparity does not exceed a pre-specified threshold. Occlusion is handled by explicitly labeling pixels as occluded during the graph-cut assignment and using the property that occlusion assignments should obey the depth ordering of the individual layers.

2.4.4 Limitations and Future Work

Running Time

In a effort to be as general as possible, we put no limit on the magnitude of the allowable motion (i.e. any pixel in the first frame is allowed to match with any pixel in the second frame). While this allows us to estimate correspondences for many possible motions, it greatly increases the running time of our algorithm and in practice, even with large disparity images, an individual pixel likely moves less than half of the image size between the two frames. If we used a threshold on the maximum disparity for a single pixel we could greatly reduce our running time.

Using a 2GHz desktop PC, the entire process for the image pair shown in Figure 2.9 takes 12 minutes and of those 12 minutes, 10 are spent in the loop that determines the nearest neighbor for each interest point. If the search space is limited we could reduce this time greatly and make our approach far more practical.

Also, since we are using a fairly simple point matching approach, our set of interest points must be quite large to ensure good matching. However, with a more sophisticated matching approach, this set could likely be greatly reduced.

Dangling Backgrounds

One of the artifacts present in some of our results are pieces of background that attach themselves to foreground motion layers. As we discussed in section 2.1, these pixels are problematic because they seem to fit the motion of the other pixels in the layer and it is only our higher level reasoning that allows us to determine that they have been erroneously assigned. Perhaps with more than two images, we could more accurately assign these pixels.

Occlusion Reasoning

Since we are working primarily with pairs of images, our occlusion reasoning is very simple – we only assign pixels to motion layers if we determine that they appear in both frames. Subsequent work has addressed this problem be incorporating information from more frames [130], but perhaps something can still be done using only two frames – e.g. use monocular segmentation to reason about pixels that have not been assigned to any motion layer. This could allow for better boundary separations in many cases.

Multiple Frames

Even though our approach is specifically designed for image pairs with large disparity which won't likely appear in high frame-rate video, a similar approach may work well for high frame-rate video. Correspondences (and motion models) could be generated using a standard point tracking algorithm [102] and then the image sequence could be segmented using our graph-cut based approach. Since the segmentation would likely be over many frames, we would have to include the possibility of occluded pixels. This method would not specifically account for out of plane effects, but decompose a video into a set of pixel trajectories that could be processed to get estimates of the actual 3-dimensional motion of the scene.

This type of segmentation would be useful in special effects especially when computer generated elements are going to be inserted into a complicated scene. It could be used for motion blur in the case of very high framerate cameras as well as for object removal (similar to the object removal presented in section 2.3. Here a scene could be decomposed into a collection of spatiotemporal layers where individual layers could be analyzed and manipulated independently.

Multi-pass Approaches

As we discussed in section 2.2, even objects that exhibit residual to a planar fit may be approximated by our method. This approach could be extended to allow for multipass approaches to motion segmentation. These passes could be with increasingly flexible models (as in section 2.2) or increasingly higher resolution versions of the images. This would allow for much faster matching and could allow for flexible models to be introduced as needed.

This chapter is based on published material that appears as:

- J. Wills, S. Agarwal and S. Belongie, "What Went Where," *CVPR*, 2003, pp. 37-44, vol. 1. [126]. I was the primary author and was responsible for the development of the method, completing the literature survey, performing the experiments and all implementation except the graph cut code.

- S. Belongie and J. Wills, "Structure from Periodic Motion," *SCVMA* (in conjunction with *ECCV*), 2004. [10].

- J. Wills and S. Belongie, "A Feature-Based Approach for Determining Long Range Optical Flow," *ECCV*, 2004, pp. 170-182, vol. 3 [128].

- J. Wills, S. Agarwal and S. Belongie, "A Feature-based Approach for Dense Segmentation and Estimation of Large Disparity Motion." *International Journal of Computer Vision*, 2006, pp. 125–143, vol. 68 (2) [127]. I was the primary author and was responsible for the development of the method, completing the literature survey, performing the experiments and all implementation except graph cut code.

3

Reflection and Refraction in Microfacet Reflectance Models

Microfacet reflectance models have been shown to work well for simulating the interaction of light with a rough surface. In this chapter, we give an overview of the existing techniques for reflection modeling and show how these techniques can be extended to handle refraction in a unified framework. We also show that existing models are simply special cases of this model. To this end, two new derivations are presented for computing quantities required for refraction as well as a result that is (to our knowledge) previously unpublished.

3.1 Introduction

The human visual system is quite proficient at discerning information about a material by examining the way it interacts with the light in the scene. When rendering images, we would like to model this interaction as accurately as possible to parallel the effects seen in the real world. Take for example, Figure 3.1, which shows a rendered image of a statue. Upon inspection, the statuette appears to be made of frosted glass. This impression results from the way the light interacts with the surface.

There are many subtle effects that a surface has on the way light is reflected/refracted. This is especially evident in dielectrics with rough surfaces (an example being frosted glass). A nice benefit of physically-based approaches is that if the surface is modeled accurately, many of these effects will be captured automatically. Notice in Figure 3.1 that there are bright regions on the back side of the statue directly opposite the light. These arise from a focussing effect the dielectric surface has on the incoming light. If we

Figure 3.1 A dragon made of frosted glass rendered using a microfacet model. For this material, reflection and refraction for a rough surface must both be modeled. (Rendered by Henrik Wann Jensen)

instead try to model this using an ad hoc reflectance model we would have a hard time finding parameters that capture these types of effects.

While the range of visual effects found in nature is quite wide, here the focus will be on what is known as geometrical optics - when the surface elements are quite large with respect to the wavelength of light - and ignore the wave effects of light, though there are nice reflectance models based on Gaussian hight fields [7, 49, 93] that capture many of these effects. For a good comparison of geometrical and physical optics see [77].

Microfacet models have been shown to model the effects that roughness has on geometrical optics quite well. These models will be the focus of this chapter. Specifically, we would like to simply model the interaction of light on a rough surface and capture effects resulting from reflection as well as transmission.

This chapter will be using the terms refraction and transmission interchangeably. While this may not be strictly correct (since refraction specifically refers to the change in direction that light undergoes during transmission), refraction is commonly used to refer to the counterpart to reflection and we will continue in that manner. Sim-

ilarly, in much of the discussion about microfacets, we will be explaining it for the case of reflection, though except where specifically stated (as in Section 3.3 which is devoted to the case of reflection) reflection will refer to both reflection and refraction.

In this chapter, we give an overview of the existing techniques for reflection and show how these techniques can be extended to handle refraction in a unified framework. We also show that this technique for refraction is in fact a generalization of the reflection models and the original models are simply special cases of this model. To this end, two new derivations are presented for computing quantities required for refraction as well as a result that is (to our knowledge) previously unpublished.

The structure of this chapter is as follows: we begin with an overview of relevant work, in Section 3.2 we give an overview of some basic concepts and notation, in Sections 3.3 and 3.4 we cover the cases of reflection and refraction, the geometric term is covered in Section 3.5, importance sampling is covered in Section 3.6, and conclusions are presented in Section 3.7.

3.1.1 Empirical Models

Some of the early models provided reasonable results but were based on purely empirical models, which led to a number of flaws with respect to physical validity. One of the earliest models, that of Lambert [62] presented a method where appearance is independent of viewing direction. Since this is probably the easiest Bidirectional Reflectance Distribution Function (BRDF) to work with, it is commonly assumed in computer vision and inverse rendering problems. However, this type of reflectance cannot result from solely surface interactions [45].

Gouraud [42] proposed a method to interpolate shading to render curved surfaces and to add a degree of realism, Phong [87] used a cosine lobe to simulate a highlight that changes with viewing direction and combined it with a diffuse (or Lambertian) term. One problem with this model is the fact that it can reflect more light than is incident on a surface. This isn't always a noticeable problem in rendering when only single bounces of light are considered, however when simulating multi-bounce phenomena, like indirect illumination, this can lead to infinite values and poor results.

The Schlick model [90] added efficiency and accuracy to the Phong model by incorporating Fresnel effects (and a nice technique for computing Fresnel coefficients for the case of unpolarized light).

Another empirical model is that of Ward [122], which handles effects produced

by anisotropic reflection. Instead of assuming an isotropic BRDF, Ward models it as an elliptical distribution with varying degrees of eccentricity. This intuitively corresponds to scratches on the surface.

Another problem with the empirical models is the interaction between reflection and refraction. There should be a significant focusing of the light for the case of refraction. The models that are cased on properties of the surface will demonstrate this focusing, however the empirical models would have to be hand-tuned to handle this effect.

3.1.2 Microfacet models

With microfacet models the surface is assumed to be composed of many tiny facets with some distribution over orientation. Light is reflected off of these facets and the amount of light that is reflected is proportional to the number of facets that are oriented in the direction required for the reflection from light to eye.

The first microfacet model is that of Torrance and Sparrow [108]. This paper introduced the various projection factors that appear in the BRDF as well a term to account for the shadowing of light from individual facets called a geometric term. To simplify the calculation of this geometric term, they assumed that the surface consists of symmetric v-grooves of infinite length in all directions - an assumption that makes this model not physically plausible since the surface cannot be constructed [59].

Blinn [14], introduced this technique to the graphics community and presented a conversion method for parameters used in the Phong model as well as for a different distribution function. Since Blinn was writing for the graphics community, he presented an algebraic formulation for the BRDF that was geared toward simple implementation. This was especially evident in the geometric and Fresnel terms.

Cook and Torrance [24] extended the Torrance and Sparrow model to include wavelength-dependent effects. In addition, they showed that reflectance predicted by their model very closely matches reflectance data for a variety of materials including metals and non-metals.

Oren and Nayar [84] extended these microfacet models for the case of diffuse facets. Since the facets are assumed Lambertian, they are modeling more than a surface reflection and even refer to the reflectance they model as body reflection. Because of this and the fact that the application of this model to refraction doesn't really make sense, we will not go into the details of this model.

$\vec{n}$	global surface normal
$\vec{n}_f$	facet normal
$\vec{\omega}_i$	incident direction
$\vec{\omega}_r$	reflected direction
$\vec{\omega}_t$	refracted direction
θ_i	angle of incident direction (w.r.t. $\vec{n}$)
θ_r	angle of reflected direction (w.r.t. $\vec{n}$)
θ_t	angle of refracted direction (w.r.t. $\vec{n}$)
θ'_i	local angle of incident direction (w.r.t. $\vec{n}_f$)
θ'_r	local angle of reflected direction (w.r.t. $\vec{n}_f$)
θ'_t	local angle of refracted direction (w.r.t. $\vec{n}_f$)
dA_s	infinitesimal surface area
α	angle between $\vec{n}_f$ and $\vec{n}$
$p(\alpha)$	number of facets per dA per $d\omega$
E_s	irradiance of the surface
L_i	radiance incident on the surface
L_r	radiance reflected from the surface
L_t	radiance transmitted through the surface
f_r	BRDF
η	ratio of indices of refraction (η_t/η_i)

Figure 3.2 Notation that will be used throughout this chapter.

Ashikhmin et al. [3] provide a method for surfaces whose microfacets have any distribution and also show a method for solving for the appropriate values for the shadowing function. This allows a relaxation of the symmetric groove assumption made initially in [108] and may be called physically plausible since a surface could be constructed. They show application of their model capturing the appearance of varying materials (including anisotropic reflection) with success.

Koenderink et. al. [59] present a method based on thoroughly pitted surfaces. While different and apparently difficult for refraction, there are nice points: shadowing is easy, multiple bounces is easy.

3.2 Basics

3.2.1 Notation and Terminology

Figure 3.2 gives a summary of the notational conventions that will be employed in this chapter.

In this section, we present definitions of radiometric terms that are useful in the study of surface reflection. Detailed derivations and descriptions of these terms are

given by Nicodemus et al. [79]. All directions are represented by the zenith angle θ and the azimuth angle ϕ. The light source is assumed to lie in the x-z plane and is therefore uniquely determined by its zenith angle θ_i, as shown in Figure 3.3. The monochromatic flux $d\Phi_i$ is incident on the surface area dA, from the direction θ_i, and a fraction of it, $d^2\Phi_t$, is reflected in the direction (θ_r, ϕ_r). The irradiance E_s of the surface is defined as the incident flux density:

$$E_s = \frac{d^2\Phi_i}{dA_s}$$

The radiance L_r, of the surface is defined as the flux emitted per unit fore-shortened area per unit solid angle. The surface radiance in the direction (θ_r, ϕ_r) is defined as:

$$L_r = \frac{d^2\Phi_r}{dA_s \cos\theta_r d\omega_r}$$

The BRDF f_r of a surface is a measure of how bright the surface appears when viewed from a given direction, when it is illuminated from another given direction. The BRDF is defined as:

$$f_r(x, \vec{\omega}_i, \vec{\omega}_r) = \frac{dL_r}{dE_s}$$

Using $d^2\Phi_i = L_i \cos\theta_i dA_s d\omega_i$, we can express this as:

$$f_r(x, \vec{\omega}_i, \vec{\omega}_r) = \frac{d^2\Phi_r}{L_i \cos\theta_i \cos\theta_r dA_s d\omega_i d\omega_r} \tag{3.1}$$

We will use this equation in the following sections to derive BRDFs that are specific to the cases of reflection and refraction from microfacet surfaces.

3.2.2 Important BRDF Properties

Conservation of Energy

Since light doesn't simply disappear or appear from nowhere, we must ensure that our BRDF doesn't introduce or remove light from the scene. This can be simply stated as:

$$\int_\Omega f_r(x, \vec{\omega}_i, \vec{\omega}_r) d\vec{\omega}_i d\vec{\omega}_r = 1$$

Note that this integral must be over the entire sphere Ω to account for transmission and absorption.

This property is especially important for global illumination applications.

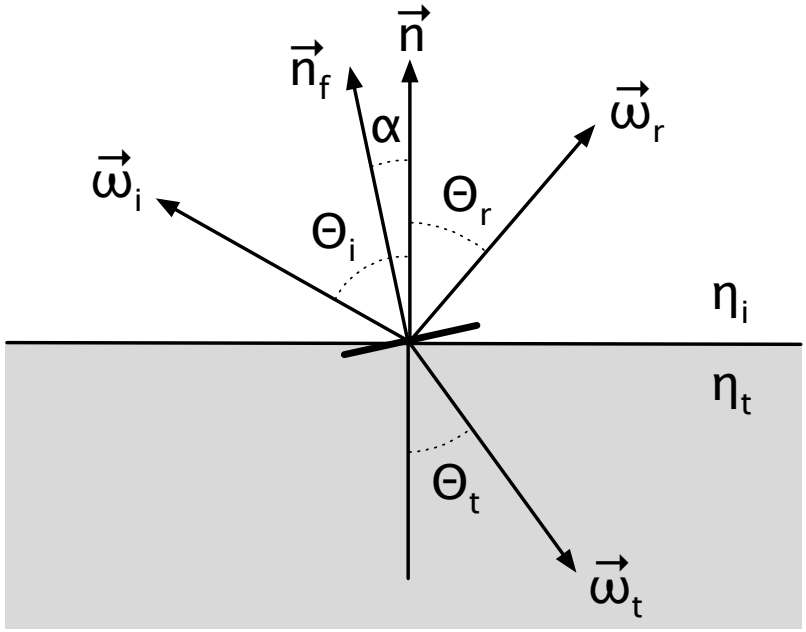


Figure 3.3 Reflection from and refraction through a microfacet on a rough surface.

Helmholtz Reciprocity

Over a century ago, Helmholtz described the interchangeable role of the light emitter and receiver with respect to reflectance. For a BRDF to be physically valid, it must fulfill this constraint, or specifically:

$$f_r(x, \vec{\omega}_i, \vec{\omega}_r) = f_r(x, \vec{\omega}_r, \vec{\omega}_i) \quad \forall\vec{\omega}_i \forall\vec{\omega}_r$$

This property is particularly important when ray-tracing since the interchangeability of the source and receiver is a core assumption to the technique.

3.2.3 Microfacet Model

Consider the geometry shown in Figure 3.3. The surface area dA_s, is located at the origin of the coordinate frame, and its surface normal points in the direction of the z-axis. The surface is illuminated by a beam of light coming in along $\vec{\omega}_i$. We are interested in determining the radiance of the surface in the direction $\vec{\omega}_r$. Only those planar micro-facets whose normal vectors lie within the solid angle $d\omega_f$ are capable of specularly reflecting light flux that is incident along $\vec{\omega}_i$ into the infinitesimal solid

angle $d\vec{\omega}_r$. From the vectors $\vec{\omega}_i$ and $\vec{\omega}_r$, we can determine the normal direction of the reflecting facets, $\vec{n}_f$ (the method for this computation is given for the cases of reflection and refraction in the following sections).

The number of facets per unit area of the surface that are oriented within the solid angle $d\omega_f$ is equal to $p(\alpha)d\omega_f$, where α is the angle between $\vec{n}$ and $\vec{n}_f$. Therefore, the number of facets in the surface area dA_s, that are oriented within $d\omega_f$ is equal to $p(\alpha)d\omega_f dA_s$. If we let a_f be the area of each facet, then the area of points in dA_s, that will reflect light from the direction $\vec{\omega}_r$ into the solid angle $d\omega_r$, is equal to $a_f p(\alpha)d\omega_f dA_s$. The flux incident on the set of reflecting facets is determined as:

$$d^2\Phi_i = L_i d\omega_i (a_f p(\alpha) d\omega_f dA_s) \cos\theta_i' \tag{3.2}$$

3.2.4 Distribution Function

Gaussian Distribution

The model proposed by Torrance and Sparrow [108] uses the Gaussian Distribution:

$$P(\alpha) = \frac{c_t}{\sqrt{2\pi}\sigma} e^{-(\alpha^2/2\sigma^2)}$$

where c_t is the normalization factor that accounts for the truncation of the distribution to $[-\pi/2, \pi/2]$ and is defined as:

$$c_t = \frac{1}{\left(\frac{1}{\sqrt{2\pi}\sigma}\int_{-\pi/2}^{\pi/2} e^{-\alpha/2\sigma^2} d\alpha\right)}$$

This normalization factor can be pre-computed using the error function and is constant for a given roughness value.

Beckmann Distribution

Beckmann and Spizzichino [8] provides a comprehensive theory for a variety of surface conditions ranging from smooth to very rough. The distribution function proposed and later used in [24] is:

$$P(\alpha) = \frac{1}{\sigma_b^2 \cos^4(\alpha)} e^{-(\tan(\alpha)/\sigma_b)^2}$$

While this expression is more difficult computationally, it is already normalized so there is no preprocessing needed.

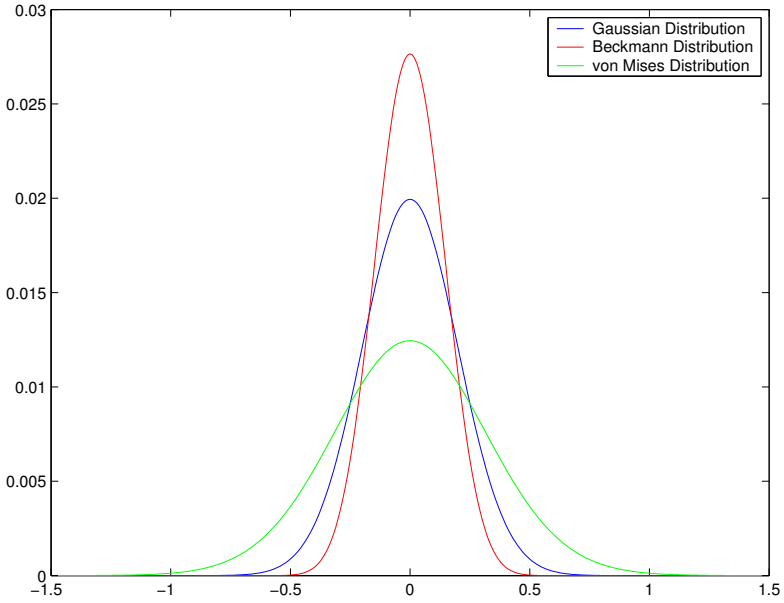


Figure 3.4 The different distribution functions for a single roughness value of $\sigma = 0.2$.

Von Mises Distribution

The Von Mises Distribution [34] is the circular analog to the Gaussian distribution on a line. It is defined on $[0, 2\pi)$ with probability density function

$$P(\alpha) = \frac{1}{2\pi I_0(\sigma_v)} e^{\cos(\alpha)\sigma_v}$$

where I_0 is a modified Bessel function of the first kind of order 0.

This covers the entire circle but can be made to cover only half of the circle if used to find the probability of 2α with a correction for the roughness parameter.

The Von Mises distribution can be sampled as in [11].

Discussion

Von Mises and Beckmann are theoretically nicer since they belong on the sphere but the Gaussian (even with truncation) is easier to compute and provides similar results.

Figure 3.4 shows plots of the different distribution functions. Notice that they are very similar in shape, and with appropriate roughness values they can be made to be quite close for most angles.

3.2.5 Fresnel Effects

Fresnel terms account for the tradeoff between reflection and transmission that occurs as the incident angle changes. As the incident angle becomes larger, more of the light is reflected. This effect can be seen in a glass of water, the portions of the glass in the center are virtually invisible, while around the edge reflections of the surroundings become quite visible.

This can be approximated as [90]:

$$F(\vec{\omega}_i, \vec{\omega}_t) = F_0 + (1 - F_0)(1 - \cos(\theta))^5$$

where F_0 is the reflectance at normal incidence:

$$F_0 = \left(\frac{\eta - 1}{\eta + 1}\right)^2$$

Where η is ratio of indices of refraction, η_t/η_i. Note that the larger of the two angles (i. e. $\theta = \max(\theta_i, \theta_r)$) should be used in this computation.

3.3 Reflection

Reflection is the process by which light that is incident on a surface leaves that surface on the same side. To compute the reflectance (that is the ratio of incident to reflected light) predicted by the microfacet model we need to compute the direction of the facet that will reflect the light in the viewing direction and the change in solid angle between the normal and the reflected direction. In the following sections we provide the details of these computations as well as the resulting BRDF.

3.3.1 Computing $\vec{n}_f$

Since the reflection will be symmetric about the normal of the reflecting facet, the required normal lies exactly halfway between $\vec{\omega}_i$ and $\vec{\omega}_r$:

$$\vec{n}_f = \frac{\vec{\omega}_i + \vec{\omega}_r}{|\vec{\omega}_i + \vec{\omega}_r|}$$

this normalization constant can be computed as $\sqrt{2 + 2(\vec{\omega}_i \cdot \vec{\omega}_r)}$.

3.3.2 Change in Solid Angle

The change in solid angle that results from a change in the normal is (as proven in [77]):

$$d\omega_f = \frac{d\omega_r}{4cos\theta_i'} \tag{3.3}$$

3.3.3 Bidirectional Reflectance Distribution Function

If we combine Equations 3.1 and 3.2 we get the following expression for the BRDF:

$$\begin{aligned} f_r(x, \vec{\omega}_i, \vec{\omega}_r) &= \frac{L_i d\omega_i (a_f p(\alpha) d\omega_f dA_s) \cos\theta_i'}{L_i \cos\theta_i \cos\theta_r dA_s d\omega_i d\omega_r} \\ &= \frac{a_f p(\alpha) \cos\theta_i'}{\cos\theta_i \cos\theta_r} \frac{d\omega_f}{d\omega_r} \end{aligned}$$

using the result from Equation 3.3 we get:

$$f_r(x, \vec{\omega}_i, \vec{\omega}_r) = \frac{a_f p(\alpha)}{4 \cos\theta_i \cos\theta_r}$$

and including the Fresnel and geometric terms $G(\cdot)$, the final expression for the BRDF becomes:

$$\boxed{f_r(x, \vec{\omega}_i, \vec{\omega}_r) = \frac{G(\vec{\omega}_i, \vec{\omega}_t) F(\vec{\omega}_i, \vec{\omega}_t) a_f p(\alpha)}{4 \cos\theta_i \cos\theta_r}}$$

We will now move on to the case of refraction where light will be transmitted between two mediums.

3.4 Refraction

Refraction (or transmission) is the process of light entering a new medium which results in certain changes in the direction and intensity. The primary difference stems from the fact that there are two media involved and therefore we must not ignore things like indices of refraction.

Refraction has a greater need for a theoretical model since it would be quite difficult to measure the effect of exactly one interface (since that would require the light and the sensor to be in different media.

As we did in Section 3.3, we provide the details of the computations necessary for the microfacet model as well as the resulting BRDF. In addition, we show that these new quantities are generalization of those in Section 3.3 that take indices of refraction

into account and show that when both rays have the same index of refraction, they reduce to those computed for reflection.

3.4.1 Computing $\vec{n}_f$

Using Snell's law we can compute the facet normal that will refract light from direction $\vec{\omega}_i$ to $\vec{\omega}_t$. The facet normal, $\vec{n}_f$, lies in the plane spanned by the incident and the refracted vectors, and it can be calculated as follows (a proof of which appears in Appendix A.2):

$$\vec{n}_f = \frac{\vec{\omega}_i + \eta\vec{\omega}_t}{\sqrt{\eta^2 - 2\eta\cos\gamma + 1}} \tag{3.4}$$

Note that $\sqrt{\eta^2 - 2\eta\cos\gamma + 1} = \sqrt{\eta^2 + 2\eta(\vec{\omega}_i \cdot \vec{\omega}_t) + 1}$ is the length of the vector $(\vec{\omega}_i + \eta\vec{\omega}_t)$ for any two non-colinear and normalized vectors $\vec{\omega}_i$ and $\vec{\omega}_t$. This is also true for the case of reflection where the numerator simplifies to $\vec{\omega}_i + \vec{\omega}_t$ and the denominator simplifies to $\sqrt{2 + 2(\vec{\omega}_i \cdot \vec{\omega}_t)}$.

This result (which I have not been able to find any published account) produces the same vector (up to a scale factor) as the method of [44]:

$$\vec{n}_f = \begin{cases} \vec{\omega}_i + (\vec{\omega}_i + \vec{\omega}_t)\frac{\eta_t}{\eta_i - \eta_t} & \text{if } \eta_i > \eta_t \\ -\vec{\omega}_t - (\vec{\omega}_i + \vec{\omega}_t)\frac{\eta_i}{\eta_t - \eta_i} & \text{if } \eta_t > \eta_i \end{cases} \tag{3.5}$$

This method simplifies to:

$$\vec{n}_f = \frac{\eta_i\vec{\omega}_i + \eta_t\vec{\omega}_t}{\eta_t - \eta_i} = \left(\frac{\eta_i}{\eta_t - \eta_i}\right)\vec{\omega}_i + \eta\vec{\omega}_t$$

While this gives the correct vector, it is not normalized and the scale factor introduced will cause problems for the case of reflection.

This vector always points in the direction of the vector associated with the smaller of the two indices of refraction. One test to check whether or not the refraction is possible is to test whether or not the negative of the dot product between the computed normal and the vector associated with the larger of the two indices of refraction is negative. If so, the refraction is not possible. This is summarized as:

if $\eta_i > \eta_t$ and $(\vec{n}_f \cdot \vec{\omega}_i) < 0$	possible
if $\eta_t > \eta_i$ and $(\vec{n}_f \cdot \vec{\omega}_t) < 0$	possible
else	impossible

3.4.2 Change in Solid Angle

To correctly calculate the number of facets that are oriented in the correct direction, we need to find the relation between a change in the solid angle around the normal and the corresponding change in the solid angle around the refracted direction. This ratio can be calculated as (a proof of which appears in Appendix A.3):

$$d\omega_f = \frac{\eta^2 \cos\theta_t'}{(\cos\theta_i' - \eta\cos\theta_t')^2} d\omega_t \tag{3.6}$$

One special case worth considering is that of specular reflection. In this case, $\theta_t' = \pi - \theta_i'$ and $\eta = 1$ and we get:

$$d\omega_f = \frac{\cos\theta_i'}{(2\cos\theta_i')^2} d\omega_t = \frac{d\omega_t}{4\cos\theta_i'}$$

which is a well-known result proven in [77]. In addition, for the case of transmission for an interface with η=1, we get $d\omega_t = 0$ which is also to be expected.

This term is also derived in [97], however the derivation is not as lucid as the one presented in this report.

3.4.3 Bidirectional Reflectance Distribution Function

As we did in Section 3.3.3, we can express the BRDF as (with the appropriate vector/angle substitutions):

$$f_r(x, \vec{\omega}_i, \vec{\omega}_t) = \frac{a_f p(\alpha)\cos\theta_i'}{\cos\theta_i \cos\theta_t} \frac{d\omega_f}{d\omega_t}$$

using the result from Equation 3.6 we get:

$$f_r(x, \vec{\omega}_i, \vec{\omega}_t) = \frac{a_f p(\alpha)\cos\theta_i' \eta^2 \cos\theta_t'}{\cos\theta_i \cos\theta_t (\cos\theta_i' - \eta\cos\theta_t')^2}$$

and including the Fresnel and geometric terms, the final expression for the BRDF becomes:

$$\boxed{f_r(x, \vec{\omega}_i, \vec{\omega}_t) = \frac{\eta^2 G(\vec{\omega}_i, \vec{\omega}_r) T(\vec{\omega}_i, \vec{\omega}_r) a_f p(\alpha)\cos\theta_i' \cos\theta_t'}{\cos\theta_i \cos\theta_t (\cos\theta_i' - \eta\cos\theta_t')^2}}$$

Where $T(\vec{\omega}_i, \vec{\omega}_r)$ is the Fresnel term for transmitted light. We will consider the geometric term G in the following section.

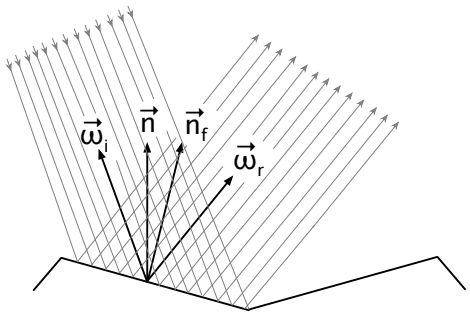
3.5 Geometric Term

Since the microfacet models expect that as the projected area from a surface patch increases, the number of visible facets (and thus reflected light) increases. For grazing angles, this projected area becomes infinite and thus the estimated irradiance becomes infinite. However, if we consider the interaction of facets at these grazing angles, we see that facets are much more likely to block each other as we approach these grazing angles. The function that models this effect is known as the geometric term. Some of the earliest work on this term comes from the physics community and was for general random surfaces [7, 93]. The term proposed by Torrance and Sparrow is probably the most widely used though the assumptions required make the model physically implausible. A more recent term was introduced by Ashikhmin et al. that doesn't make these limiting assumptions.

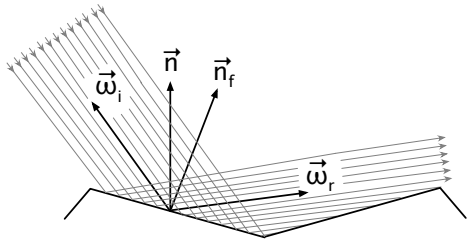
3.5.1 Torrance and Sparrow

The Torrance-Sparrow geometric term depends on two very important assumptions: the first is that the surface is comprised of long grooves that extend in all directions and the second is that these grooves are symmetric – that each side of the groove is at the same angle to the global surface normal. The first assumption is what makes the model physically implausible since no surface can be constructed that has infinite grooves that extend in all directions (though it is important to note that this assumption is not inherent to this microfacet model in general, but only to this term - without this term the Torrance-Sparrow model may be considered physically plausible). The second assumption simplifies the calculations and more importantly, it makes the case of grazing angles trivial.

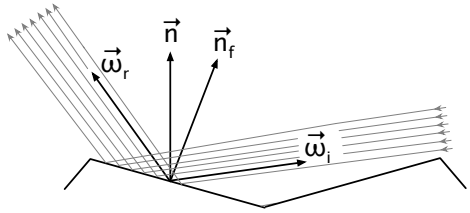
Figure 3.5 shows the three types of situations we would like to capture with the geometric term: (a) shows perfect reflection, or the case where no light is blocked by another facet before or after the reflection, (b) shows masking, when the light is blocked bay another facet after the reflection, and (c) shows shadowing (same as masking except that the roles of the light and the viewing direction are exchanged), when the light is blocked before the reflection. What is needed for each of these cases is the ratio of light that is not blocked to the light incident on the facet. This ratio is trivially 1 for (a) and the other two cases can be computed quite easily. Blinn [14] provides a nice derivation for each of these cases in terms of simple dot products. For the case of masking, as shown in Figure 3.5(b), the ratio of unblocked light to incident light, $G_b(\vec{\omega}_i, \vec{\omega}_r)$ can be



(a) Perfect reflection



(b) Masking



(c) Shadowing

Figure 3.5 Three cases for the geometric term: (a) perfect reflection, (b) masking, (c) shadowing. This figure is reproduced after [14]

expressed as:

$$G_b(\vec{\omega}_i, \vec{\omega}_r) = \frac{2(\vec{n}_f \cdot \vec{n})(\vec{\omega}_i \cdot \vec{n})}{(\vec{\omega}_r \cdot \vec{n}_f)}$$

And for the case of shadowing, as shown in Figure 3.5(c), the ratio of unblocked light to incident light, $G_c(\vec{\omega}_i, \vec{\omega}_r)$ can be expressed as:

$$G_c(\vec{\omega}_i, \vec{\omega}_r) = \frac{2(\vec{n}_f \cdot \vec{n})(\vec{\omega}_r \cdot \vec{n})}{(\vec{\omega}_i \cdot \vec{n}_f)}$$

Using these terms, we can calculate the ratio of unblocked as the minimum of these three cases:

$$G(\vec{\omega}_i, \vec{\omega}_r) = \min\left(1, G_b(\vec{\omega}_i, \vec{\omega}_r), G_c(\vec{\omega}_i, \vec{\omega}_r)\right)$$

One thing to notice about this geometric term is that it is independent of the roughness of the surface. This doesn't seem like a good property of a geometric term

since the higher the roughness value (that is the higher the likelihood of facets with large slopes) the more likely facets will be blocked as grazing angles are approached. Next, we will consider a geometric term that takes roughness into account and does not rely on the symmetric v-groove assumption.

3.5.2 Ashikhmin, Premoze and Shirley

If we do not assume that the surface is composed of v-grooves, but instead of randomly oriented facets we can think of the geometric term as capturing the ratio of shadowed area to projected area (though we will have to do this twice, once for shadowing and once for masking). Ashikhmin et. al. [3] proposed a method for computing this value that is dependent on the roughness of the surface. More precisely, this involves computing the expected value of the shadowed area for facets according to the surface roughness distribution:

$$G_p^*(\vec{\omega}) = \int_{\vec{n}_f \in \Omega} (\vec{\omega} \cdot \vec{n}_f) p(\alpha) d\omega_f \tag{3.7}$$

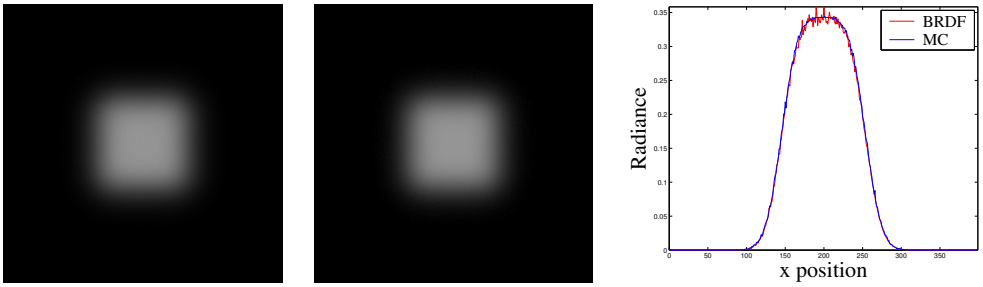
This is then computed for the incident and viewing direction and the value of $G_p(\vec{\omega}_i, \vec{\omega}_r) = \min(G_p^*(\vec{\omega}_i), G_p^*(\vec{\omega}_r))$.

This expression is evaluated using numerical integration, but can be precomputed and, since it is quite smooth, is suitable for interpolation for a set of sample points.

This method is quite nice theoretically since it eliminates the symmetric v-groove assumption (which is especially important for the case of refraction where the notion of symmetry within a groove becomes difficult when the groove will be viewed both from above and within the surface. However, the results of this geometric term are actually quite close to the results for the Torrance-Sparrow term and may not warrant the extra computation for many purposes.

3.6 Importance Sampling

Importance sampling is essential for efficient rendering with a BRDFs that have peaked distributions. In our case this happens when the roughness of the surface is low. Unfortunately, there has not been much progress in developing techniques for importance sampling of microfacet based models such as the Torrance-Sparrow model. The reason for this is that the final expressions are complex and difficult to integrate analytically. One possibility is to replace the BRDF with a simpler model such as the Lafortune model [61],



BRDF Monte Carlo

Figure 3.6 A square light seen through a dielectric surface with roughness $\sigma = 0.3$. The scene has been rendered using the BRDF (left) and brute-force Monte Carlo (middle) sampling of the microfacets. On the right the intensity values along a slice through the center of the images have been plotted for comparison. (Rendered by Henrik Wann Jensen)

which supports importance sampling. Another option is to directly simulate the physics of the microfacet distribution [56], which turns out to be surprisingly simple and efficient for sampling microfacet distributions. We start with the integral for reflected radiance

$$L_r = \int_{\vec{\omega}' \in \Omega} f_r L_i(\vec{n} \cdot \vec{\omega}')\, d\omega' \tag{3.8}$$

(which is just the evaluation of the BRDF over all incident directions). To evaluate this integral, we use the fundamental assumption that the microfacets are independent and distributed according to $p(\alpha)$. For simplicity, we assume that the distribution is Gaussian and isotropic, but any $p(\alpha)$ can be used. To sample the BRDF, we first generate a pair of Gaussian distributed random numbers (using the Box-Müller method [17]). We use the length of the vector for the elevation angle and the rotation for the azimuth. These random numbers are used to perturb the surface normal. Next, we compute the Fresnel term for the perturbed normal direction, and use random sampling to pick either a reflected or refracted direction. If the perturbed normal generates a valid reflected or refracted direction then we trace a ray in this direction — otherwise the result is zero. The resulting ray distribution accounts for both the facet distribution function, the Fresnel term, and the change in the solid angle.

In Figure 3.6 we compare the Monte Carlo sampling method with a direct evaluation of the BRDF for refraction of an area light source in a square. Note, how the results match quite nicely.

3.7 Conclusion

Microfacet models can be used to model the reflection from and the refraction through a rough surface. In this report, we have given an overview of the existing techniques for reflection and shown how these techniques can be extended to handle refraction in a unified framework. We have also shown that this technique for refraction is in fact a generalization of the reflection models. We have seen methods for computing the distribution of incident light for both of these cases as well as a possible direction for future research. At the cutting edge of rendering continues to approach photo-realism, these types of modeling will become increasingly important to be able to capture the many subtle appearance effects in the natural and man-made worlds.

4

Toward a Perceptual Space for Reflectance

The Bidirectional Reflectance Distribution Function (BRDF) describes the way a surface reflects light. BRDFs are complex mathematical objects that, while allowing for a complete radiometric description of light reflecting from a surface, can be difficult to use in practice. Recently there has been interest in understanding the perception of reflectance in a manner similar to the work done over the last two centuries on the perception of color. The aim is to construct a low-dimensional, perceptual space for BRDFs that can be easily navigated, similar to a perceptually uniform color space. To this end, we design and carry out a comprehensive psychophysical study of the perception of measured reflectance. This is the largest study of its kind to date, and the first to use real material measurements. In addition, we introduce a new multidimensional scaling (MDS) algorithm for analyzing ordinal data that unlike existing methods is both efficient and optimal. We use the results of our study to construct a perceptual space of these BRDFs and introduce a new method for perceptual construction of novel BRDFs.

4.1 Introduction

Photorealistically rendered images depend not only upon scene geometry and illumination, but also on the material models for each object. A local model of material reflectance is captured by a Bidirectional Reflectance Distribution Function (BRDF) [80]. There has been a great deal of progress in creating physically-based analytic BRDF models [3, 23, 50, 108, 121] and more recently measurement driven models [74]. For each of these reflectance models, there exists some underlying space of possible reflectances

whose dimension is given by the parameters of the model. Yet, these models and their resulting spaces do not account for the ways people actually perceive materials, e.g. which attributes are significant and which ones are ignored. In this chapter we analyze the perception of reflectance and introduce a methodology for deriving a space for reflectance from results of psychophysical experiments using measured BRDF data.

The study of the perception of reflectance holds promise for both computer graphics and vision researchers. As it stands today, a digital artist has to develop a feel for the parameters of the various analytical reflectance models before he can use them to produce the desired effect. This is due to the complex relationship between model parameters and the resulting perceptual sensation. Learning this relationship is a complicated and error prone process based on repeated trial and error. The challenge is even more acute for data-driven BRDF models where the parameter space is particularly large and unintuitive. Imagine trying to select a desired color by specifying parameters that define a spectral density function. A perceptual space for reflectance allows computer graphics artists to readily navigate the space of BRDFs and to work with BRDFs in a manner similar to how they work with various color spaces. Additionally, shading is a strong cue in human vision and an understanding of reflectance perception will give us insight into the priors and constraints used by humans to solve various shading related problems, e.g., shape from shading and recognition over variable and unknown lighting.

Reflectance can often be broken into two distinct components: chromatic and achromatic. In this study we will restrict our attention to the achromatic aspects of the BRDF, also known as gloss. Gloss was originally studied in the paper industry [51] and has been formalized by the American Society for Testing and Materials (ASTM). The ASTM defines gloss as "the angular selectivity of reflectance, involving surface-reflected light, responsible for the degree to which reflected highlights or images of objects may be seen as superimposed on a surface" [5].

We chose to consider only gloss because the largest publicly available database of reflectance measurements (the MIT-MERL database [74]) consists of only 55 usable isotropic BRDFs. This is a very small subset of the vast variety of reflectance functions. Color is such a strong perceptual cue that given the sparseness of our BRDF database, differences in color between two BRDFs will completely overwhelm differences due to the gloss. Thus, in the following, the term BRDF will refer to the achromatic aspects of reflectance; when we refer to the chromatic aspects, we will make specific note of it.

Our study contributes to the state of the art in perception research in computer graphics in three ways. First, we design and implement a comprehensive study of the

perception of measured reflectance. This is the largest study of its kind to date, and the first to use real material measurements. Second, we develop a new multidimensional scaling (MDS) algorithm for analyzing ordinal data. This algorithm is a replacement for the widely used weighted non-metric MDS algorithm [16]. The new algorithm is efficient and optimal, in that it finds the globally optimal solution in polynomial time, unlike the algorithm it replaces. Finally we use the results of our psychophysical study to analyze the perception of the achromatic aspects of reflectance. As part of this analysis we estimate the dimensionality of the space of reflectance perception and construct a perceptually meaningful embedding of these BRDFs. We also introduce a novel perceptual interpolation scheme that uses the embedding obtained from human subject responses and the geometry of the space of BRDFs to provide the user with an intuitive interface for navigating the space of reflectances and constructing new ones.

We begin with related works in section 4.2. In section 4.3 we present our experimental framework for measuring the perception of gloss. Our new method for analyzing perception of gloss is described in section 4.4. An analysis of our results and our method for perceptual interpolation are provided in section 4.5. We conclude with a discussion in section 4.6.

4.2 Related Works

With the increasing emphasis on photorealism, perception has become an active area of research in a number of areas of computer graphics including ray distribution for global illumination [98], tone mapping [65, 113], evaluation of translucency [36] and perception of reflectance [86, 125]. In this section we survey some of the recent work in computer graphics and vision science on the perception of reflectance. We refer the interested reader to [86] and [125] for a more complete survey of the historical developments in this area.

In computer graphics the study of the perception of reflectance was pioneered by Pellacini et al. in which they present a perceptually meaningful reparameterization of the Ward reflectance model [86]. The authors collected perceptual data by asking subjects to quantitatively rate the similarity between images generated with varying model parameters. MDS was used on these ratings to obtain a perceptual model with two parameters. The parameters roughly correspond to Hunter's contrast gloss and DOI gloss. The original Ward parameters for roughness and contrast gloss are independent, while contrast gloss depends on both specular and diffuse reflectance – most likely because the

human visual system is sensitive to relative luminance. The study was based entirely on a single empirical BRDF model. Our work, while similar in spirit, is based entirely on measured reflectances and a much larger set of subjects. We also will argue in the next section that our psychometric study based on paired comparisons is a significant improvement over the ratings system employed by Pellacini et al.

Westlund and Meyer used appearance standards to build a new method for representing BRDFs [125]. Each BRDF is represented by a set of two types of measurements: measurements for specular gloss and haze (ratio of the specular peak to the light a few degrees off specular) and measurements for flop (chromatic effects as seen in pearlescent and metallic Paints). Color was measured both at specular peak and off specular followed by interpolation in CIELAB space. The model allows for simple representation and much simpler measurement of materials, and inherits the psychophysically based qualities of the individual components (e.g., each step in the interval from 0 to 100 in the gloss dimension equals a uniform step in gloss space), though there is no attempt to capture the perceptual effects of different combinations of the dimensions.

In [81], Obein et al. estimate the perceptual scaling of gloss using a series of 10 black plates[1] arranged into pairs of pairs and users decided which of the two pairs were more similar. Since they are only interested in specular gloss, they assume the data is one dimensional and use maximum likelihood difference scaling (MLDS) [71] to get an appropriate scaling of the data that obeys the similarities observed by the subjects. They find that people are far more sensitive to small changes in low gloss samples and less sensitive in intermediate and high gloss samples.

A significant milestone in the availability of measured reflectance data was the work of [74] who followed up on the work of [72]. The authors developed a gantry and used it to measure a number of isotropic materials. A part of this data set is now publicly available. Our work is based on this database of measurements. As part of the same work, the authors also developed a new reflectance model that was based on their database of measured isotropic BRDFs. They explored both linear as well as non-linear representations. Their model had 45 dimensions in the linear case and 14 dimensions in the non-linear case. A user test was used to classify the BRDFs in a number categories e.g. blueness, goldness, metalness which were used to define trait vectors which were then used to navigate the space of BRDFs and assist a user in moving from one BRDF to another.

[1] Black was chosen so the specular highlight dominates the diffuse component.

In terms of methodology the work that is closest to ours is that of Ledda et al. [65]. The authors examined the perceptual performance of various tone mapping operators by doing paired comparisons on pairs of tone mapped images displayed on two low dynamic range displays and a high dynamic range display showing the original high dynamic range image. While similar in the methodology of collecting the data, our analysis methods are significantly different as they are only interested in questions of consistency and overall preferences.

The analysis of paired comparisons in statistics, the experimental paradigm used here, has a long history in statistics, psychometrics and biometrics. This includes work on producing rankings, measuring consistency within and across subjects and MDS methods for ordinal data [16, 27, 57]. In this study we are particularly interested in constructing an embedding from paired comparisons. The weighted non-metric MDS algorithm addresses this problem, however it has a number of shortcomings the most significant of which is that it is based on an iterative majorization procedure which can only find a locally optimal solution to the stress minimization problem it solves. Our work addresses this problem by formulating the ordinal MDS problem as a semidefinite programming problem, which can be solved optimally in polynomial time. This approach has its roots in the work on semidefinite embeddings [123] and distance function learning from relative comparisons [91].

4.3 Experimental Framework

The aim of our experiment is to capture the perceptual similarity for varying reflectance functions. Each participant was shown a series of triplets of rendered images with constant geometry and illumination, but with varying BRDFs and was asked to indicate whether the center image was more similar to the image on the left or to the image on the right (Figure 4.1 shows a screenshot from one such test). We chose this form of experiment, known as *paired comparison*, over the rating method based on a continuous slider in [86]. Rating methods using continuous intervals have been shown to have problems with validity and reliability and subjects usually require a fair amount of training before the experiment [57]. In particular, each subject it seems has his or her own internal continuous scaling function that confounds the process of integrating responses across subjects; despite its precision, its accuracy is questionable. Paired comparisons on the other hand offer a much simpler task and enjoy far more intra- and inter-subject consistency.

Figure 4.1 Screen capture from the distance comparison test. The subject is asked to click on the appropriate button to indicate which pair appears more similar, **Left**: Left + Middle, or **Right**: Middle + Right. This mode of input has a number of advantages over the conventional approach of asking the subject to provide a continuous measure of similarity using a slider: (1) the paired comparison is a subjectively easier task, (2) the additional information content in a human specified continuous dissimilarity measure is of questionable value, (3) the mapping between different subjects' similarity scales is unknown *a priori*.

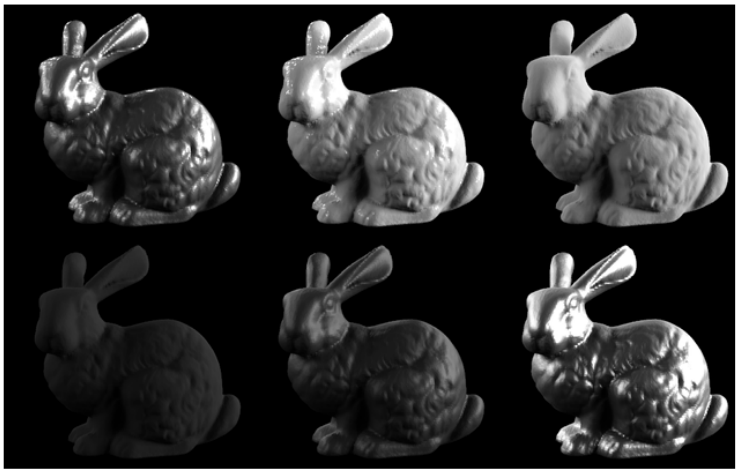


Figure 4.2 Example BRDFs. Six of the 55 images used in our psychophysics study. While monochromatic, they have widely varying gloss properties. The BRDFs used include metals, paints, fabrics, minerals, synthetics, and organic materials.

The images used in the experiment all contain the Stanford bunny [114] rendered under constant illumination and viewing direction with 55 BRDFs from the MIT/MERL BRDF database [74]. The database contains a large representative set of materials including metals, paints, fabrics, minerals, synthetics, and organic materials. Examples of some of these BRDFs appear in figure 4.2. We used natural illumination since it has been

shown that subjects have more discriminative power under this type of illumination than under simple and/or synthetic lighting [35]. We used the illumination conditions that worked best in their experiment. We chose the bunny model because it is simple yet provides a more varied distribution of surface normal/incident direction combinations than a sphere. Each image was rendered under the same high dynamic range illumination using structured importance sampling [1]. As in previous work [35, 86], we used Tumblin's rational sigmoid [112] to map the rendered high dynamic range images to our low dynamic range displays. The images were rendered in color and then converted to grayscale for our experiment. Our displays have a maximum brightness of 180 cd/m^2.

Previous studies on this subject had a rather small number of subjects [81, 86]. As there are over 78,000 possible triplets only a randomly sampled subset of comparisons could be performed. Our study has 75 subjects performing 200 comparisons for a total of 15,000 comparisons (there were a small number of repeated comparisons). None of the authors were subjects. All subjects were unaware of the aim of the experiment and all had normal or corrected to normal vision. The triplets were chosen at random for each subject.

4.4 Analyzing Paired Comparisons

One of the aims of this study is to construct a Euclidean Space in which the Euclidean distance between a pair of BRDFs corresponds to the perceptual distance between them. This is not to say that such a space necessarily exists. Indeed there is nothing that suggests *a priori* that human perception obeys the triangle law. However, the analytical, representational and computational simplicity of a linear space is attractive enough to warrant an attempt at discovering the best fitting Euclidean embedding.

Multidimensional scaling (MDS) refers to the general task of assigning coordinates to a set of objects in some Euclidean space such that a given set of dissimilarity, similarity or ordinal relations between the points are obeyed. This assignment of coordinates is also known as an embedding. The most well known of the various MDS algorithms is classical multidimensional scaling, where the dissimilarities between points are assumed to be actual Euclidean distances.

Let D be an $n \times n$ matrix of pairwise distances. The matrix D is symmetric with a zero diagonal. We are interested in finding a $d \times n$ matrix X where each column $\boldsymbol{x}_i$ is the representation of the point i in R^d and $\mathrm{D}_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$. Denote the inner product (or Gram matrix) for this set of points by $\mathrm{K} = \mathrm{X}^\top \mathrm{X}$. K is an $n \times n$ symmetric positive semi-

definite matrix. Let us now abuse notation and use D^2 to indicate the matrix of squared pairwise distances $\mathrm{K} = -\frac{1}{2}(\mathrm{I} - 11^\top)\mathrm{D}^2(\mathrm{I} - 11^\top)$. Here, I is the $n \times n$ identity matrix and 1 is the n-vector of all ones. In light of this solution to the Classical Multidimensional Scaling problem is straight forward. Given the eigenvalue decomposition $\mathrm{K} = \mathrm{U}\Sigma\mathrm{U}^\top$, it follows that $\mathrm{X} = \mathrm{U}\Sigma^{1/2}$. The solution so obtained is ambiguous to a global rotation.

The weighted ordinal multidimensional scaling algorithm [16] has been used in the past to construct Euclidean embeddings from paired comparisons or ranking data. This algorithm formulates the embedding as the minimum of a non-linear minimization problem and uses a combination of isotonic regression and iterative majorization to find a locally optimal solution. The algorithm has no stated time complexity or quality guarantees associated with it. In this section we present a new ordinal MDS algorithm that utilizes modern convex optimization theory to solve for an Euclidean embedding in polynomial time. The solution that it returns is guaranteed to be the globally optimal solution to the optimization problem that we formulate.

An important consideration when performing MDS is the issue of dimensionality, i.e., how many dimensions should the embedding exist in? Ideally we want the embedding of the smallest possible dimension. There are a number of reasons for this. An obvious one is computational complexity. A lower dimensional embedding is computationally easier to work with and to visualize. A more important reason however is that we want our embedding not only to explain the observed data but also to generalize well to unseen data. Statistical learning theory [117] informs us that for the same training error a simpler model is expected to perform better than a more complex one and should be preferred. For our analysis, the rank of the embedding is its complexity and thus we prefer lower rank embeddings to higher rank ones.

While the tools presented in this section were specifically developed for our study, we hope that they will find broader use in other psychometric and statistical studies.

4.4.1 MDS for Paired Comparisons

We are now ready to present our method for performing multidimensional scaling on relative comparisons. The method is related in spirit to the recent work on learning kernel matrices [63] and learning distance metrics from relative comparisons [91] but significantly different so as to warrant a detailed presentation.

We begin with some notation. We use lower case italicized roman symbols

$i, j, k, \ldots$ to indicate scalars and to index into the set of BRDFs. Lower case bold faced symbols $\boldsymbol{x}$ indicate vectors. Upper case symbols $\mathrm{P}, \mathrm{Q}, \mathrm{R}, \ldots$ are used to denote matrices.

The matrix X is used to indicate the embedding coordinates for the BRDFs. The matrix K denotes the Gram matrix, $\mathrm{K} = \mathrm{X}^\top \mathrm{X}$. K is a symmetric positive semi-definite matrix, denoted by $\mathrm{K} \succeq 0$.

$\mathcal{S}$ is the set of all collected observations consisting of 3-tuples (i, j, k) where the subject indicated that perceptually the image rendered using BRDF j was more similar to the one rendered using BRDF i than it was to the image rendered using the BRDF k. Let D_{ij} denote the perceptual distance between BRDFs i and j, then

$$\mathcal{S} = \{(i, j, k) | \mathrm{D}_{ij} < \mathrm{D}_{jk}\} \tag{4.1}$$

Note that while our experiments do not provide an estimate of D_{ij}, they do provide the inequality relation $\mathrm{D}_{ij} < \mathrm{D}_{jk}$. The set $\mathcal{S}$ is allowed to have repetitions and inconsistencies.

As in classical MDS we convert the problem into one that can be stated in terms of the Gram matrix K.

$$\begin{aligned} D_{ij}^2 = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 &= \boldsymbol{x}_i^\top \boldsymbol{x}_i - 2\boldsymbol{x}_i^\top \boldsymbol{x}_j + \boldsymbol{x}_j^\top \boldsymbol{x}_j \\ &= \mathrm{K}_{ii} - 2\mathrm{K}_{ij} + \mathrm{K}_{jj}, \end{aligned}$$

where, K_{ij} is the (i, j)-th element of of K. Since distances by definition are always non-negative we can without loss of generality replace the constraint $\mathrm{D}_{ij} < \mathrm{D}_{jk}$ with $\mathrm{D}_{ij}^2 < \mathrm{D}_{jk}^2$, which we can then write in terms of the inner product matrix K as

$$\mathrm{K}_{ii} - 2\mathrm{K}_{ij} + \mathrm{K}_{jj} < \mathrm{K}_{jj} - 2\mathrm{K}_{jk} + \mathrm{K}_{kk}$$

Now, our general aim is to find a Gram matrix, K, that satisfies inequality constraints of the above form for every triplet (i, j, k) that is a member of $\mathcal{S}$. As we noted earlier, K is symmetric positive semidefinite. This is a necessary and a sufficient condition for K to be the inner product matrix for some set of points.

The set of inequality constraints above are not sufficient to determine a positive semidefinite matrix K uniquely. This is because the relative comparison constraint has a scale, translation and rotation ambiguity. Translating a point set in space does not change the inter-point distances and scaling the entire point set preserves the relative

ordering of every pair of distances. While doing nothing to change the geometry of the solution, this can lead to numerical instabilities in the convex solver. Finally, even though the embedding we construct is ambiguous up to a rotation as is the case with classical MDS, the Gram matrix K is rotation invariant.

The translation ambiguity is eliminated by demanding that the embedding be centered at the origin, i.e., $\forall a = 1, \ldots, n,\ \sum_b \mathrm{X}_{ab} = 0$, which can be restated as

$$\begin{aligned} \sum_a \left(\sum_b \mathrm{X}_{ab} \right)^2 &= 0, \\ \sum_{bc} \sum_a \mathrm{X}_{ab}\mathrm{X}_{ac} &= 0, \\ \sum_{bc} \mathrm{K}_{bc} &= 0. \end{aligned} \tag{4.2}$$

This is a linear equation in the entries of the matrix K.

Handling the scale ambiguity is a bit more complicated. To prevent the embedding from collapsing into the origin, we constrain the scale of the embedding from below. We will demand that for a relative comparison to be valid the two distances should be different by at least 1 unit distance.

$$\mathrm{K}_{ii} - 2\mathrm{K}_{ij} + \mathrm{K}_{jj} + 1 \leq \mathrm{K}_{jj} - 2\mathrm{K}_{jk} + \mathrm{K}_{kk}. \tag{4.3}$$

Two things should be noted here. What was a strict inequality earlier has now been converted into a non-strict one. Secondly, the choice of 1 as the minimum difference between pairs of distances is arbitrary and does not affect the quality of the embedding. The choice of any other constant would result in a uniform scaling of the embedding. This form of the constraint only bounds the scale of the embedding from below. We have not constrained the scale of the embedding from above yet. We will deal with this shortly.

As we noted earlier, we are not interested in just any embedding that obeys the data constraints, but the one with the minimal dimension. The dimensionality of the embedding is the same as the rank of the matrix X which is in turn the same as the rank of the matrix K. Thus in the ideal case in which we have data that is completely noise free and there exists a Euclidean space in which it can be embedded we would like

to solve the following optimization problem

$$\begin{aligned} &\arg\min_{\mathrm{K}} \ \operatorname{rank}(\mathrm{K}) \\ \forall (i,j,k) \in \mathcal{S} \quad &\mathrm{K}_{kk} - \mathrm{K}_{ii} + 2\mathrm{K}_{ij} - 2\mathrm{K}_{jk} \geq 1 \\ &\sum_{ab} \mathrm{K}_{ab} = 0, \quad \mathrm{K} \succeq 0 \end{aligned} \tag{E1}$$

The above formulation has two problems. First, for the optimization problem to be feasible, there should be a positive semidefinite matrix that satisfies every relative comparison in the collected data. This is clearly not true in general. Second, the rank of a matrix is a non-convex function and thus the above is a non-convex optimization problem. Indeed, minimizing the rank of a symmetric positive semidefinite matrix subject to linear inequality constraints is an NP-hard problem [32].

To get around the first problem we introduce slack variables ξ_{ijk} in every inequality constraint which allow for violations of the inequality and augment the objective function to minimize the total violation:

$$\begin{aligned} &\arg\min_{\mathrm{K},\boldsymbol{\xi}} \sum_{(ijk)\in\mathcal{S}} \xi_{ijk} + \lambda \operatorname{rank}(\mathrm{K}) \\ \forall (i,j,k) \in \mathcal{S} \quad &\mathrm{K}_{kk} - \mathrm{K}_{ii} + 2\mathrm{K}_{ij} - 2\mathrm{K}_{jk} \geq 1 - \xi_{ijk}, \\ &\xi_{ijk} \geq 0, \quad \sum_{ab} \mathrm{K}_{ab} = 0, \quad \mathrm{K} \succeq 0. \end{aligned} \tag{E2}$$

This introduction of slack variables is very similar to the formulation of soft-margin support vector machines. λ is a positive scalar that controls the tradeoff between the violations and the rank of the matrix, i.e., the complexity of our model.

To deal with the problem of non-convexity of the objective function we use what is now a standard tool from the convex programming literature. Instead of solving the original problem, we solve a convex relaxation. The convex envelope of the rank of a symmetric positive semidefinite matrix is its trace [32]. In light of this we have the following semidefinite program (SDP).

$$\begin{aligned} &\arg\min_{\mathrm{K},\boldsymbol{\xi}} \sum_{(ijk)\in\mathcal{S}} \xi_{ijk} + \lambda \operatorname{tr}(\mathrm{K}) \\ \forall (i,j,k) \in \mathcal{S} \quad &\mathrm{K}_{kk} - \mathrm{K}_{ii} + 2\mathrm{K}_{ij} - 2\mathrm{K}_{jk} \geq 1 - \xi_{ijk}, \\ &\xi_{ijk} \geq 0, \quad \sum_{ab} \mathrm{K}_{ab} = 0, \quad \mathrm{K} \succeq 0. \end{aligned} \tag{E3}$$

Reformulating the objective function in terms of the slack variables and the trace of the matrix has an additional benefits. It constrains the scale of K from above since its a minimization problem.

There is an intuitive explanation for using the trace of the matrix K as the convex regularizer. The rank of a symmetric matrix can be restated as the number of non-zero eigenvalues, or the L_0(counting) norm of the vector of its eigenvalues. A commonly used convex relaxation for problems involving finding the sparsest vector is to replace the objective with the L_1 norm of this vector. For a symmetric positive semidefinite matrix the trace is exactly that, the L_1 norm of the vector of eigenvalues. Another way of interpreting the regularizer is to note that the sum of the eigenvalues of K is the variance of the embedding. Thus it implies that for all embeddings with the same slack violation we will choose the one that has the lowest variance.

The optimization problem (E3) is a semidefinite program (SDP). SDPs are convex optimization problems that are generalizations of linear programming problems and can be solved efficiently and optimally using interior-point methods similar to the ones used for solving linear programs [78, 116]. Efficient solvers exist for solving SDPs [99]. Thus the proposed MDS algorithm is both efficient and optimal in the solution that it reports.

Once K is computed, the embedding itself can be recovered from the eigendecomposition of K in the same manner as in classical multidimensional scaling as shown in Appendix 4.4.

4.5 Experiments and Analysis

In this section we present our analysis of the human subject data. We describe the various sources of error in a data set like ours and the results of three independent experiments that we performed to estimate these errors. We describe the perceptual space that results from performing MDS on the set of paired comparisons and discuss its properties. We show how the embedding correlates with existing ASTM standard measurements for gloss. Finally we describe a perceptual interpolation process that allows the user to navigate the space of BRDFs and generate novel intermediate BRDFs corresponding to their position in the perceptual space. We begin with a short discussion of generalization error, parameter selection and k-fold cross-validation.

Despite the set of 55 BRDFs of the MIT-MERL database being a big step forward in terms of measured data availability, this is a fraction of the space of BRDFs

and even so we can only measure a sampling of them. It is therefore important that care is taken before making any inferences from it. The inferences we make should not just explain the observed data points, but their expected performance over the unobserved portions of the space should be good as well. Only then can we be sure that our conclusions are not just an artifact of our particular data set but say something more general about perception. Given a set of subject responses to paired comparisons on the same 55 BRDFs, we measure the error of an embedding as the average number of paired comparisons that are violated if we use the pairwise distance between BRDFs in the embedded space as our estimate of the distance between them.

The expected error of an estimator over an independent test set is called the *test error* or *generalization error* [48]. The training error is smaller than testing error, in fact it can be arbitrarily smaller than it. Thus, when reporting the performance of our statistical estimates from the data, it is important to report an estimate of the generalization error and not just the training error.

Another problem that one faces in problems like the one we are solving is that of model selection. In the last section we argued that simpler models or lower complexity estimates are to be preferred to higher complexity ones. However, it is also the case that higher complexity models typically fit the training data better than lower complexity models. In our case the regularization parameter λ controls the complexity of our embedding. But how does one choose the optimal value of λ? If one could estimate the generalization error for the various choices of λ then one could choose that λ for which the error was the lowest.

The most widely used method for estimating generalization error is cross-validation [48]. In k-fold cross-validation the data set (the set of human responses) is split into k roughly equal parts. At the i^{th} iteration, the model is fitted (embedding is learned) using $k-1$ parts of the data excluding the i^{th} part which is then used for measuring the prediction error. The final prediction error estimate is the mean of the k error estimates obtained in this manner. Typical choices of k are 5 or 10. We use 10-fold cross-validation in this study.

4.5.1 Sources of Error

As with any study involving human responses, our data is prone to errors and inconsistencies. This can be due to inconsistency of response across subjects as well as inconsistencies across comparisons performed by a single subject. These errors

are not just a function of the set of subjects used for our study but of the particular experimental setup that we use to collect the data, including but not limited to the choice of the BRDFs, geometry and illumination. In this section we describe three experiments, one that examines inter-subject inconsistencies and two that look at the two types of inconsistencies exhibited by a single subject. These experiments, each carried out with 12 different subjects, were independent of our main experiment with 75 subjects.

Inter-subject Consistency

It is possible that given the same triplet of images such as the one in Figure 4.1, there will be variability in the response of the subjects to it. To get an estimate of how often subjects came to the same conclusion on our data set, we did a separate smaller study in which 12 subjects evaluated the same set of 120 randomly chosen comparisons. The reason for a separate study was that due to the large size of the set of possible comparisons ($55\binom{54}{2} \approx 78000$) and our main data set does not contain a significant number of overlap across all the users.

We found significant agreement across subjects. The majority vote accounts for about 85% of our total data in this experiment.

Repeated-Trial Consistency

It is the case that sometimes when asked to make the same paired comparison multiple times, the same subject will show variability in his or her response across trials. It is important to estimate this variability to get an idea of the repeatability of the experiment. High variability reduces our trust in the data and the conclusions we can draw from it.

We conducted a study where 30 random comparisons were chosen and presented 4 times, randomizing both the order of the outer images as well as the order of presentation. Twelve subjects were used for this study. On average 87% of the time subjects gave the same answer.

Circular Preferences

One of the aims of this study is to construct a Euclidean space in which distances correspond to perceptual dissimilarity. In a Euclidean space every set of unique pairwise distances between a set of points can be ordered without ambiguity, thus if a subject

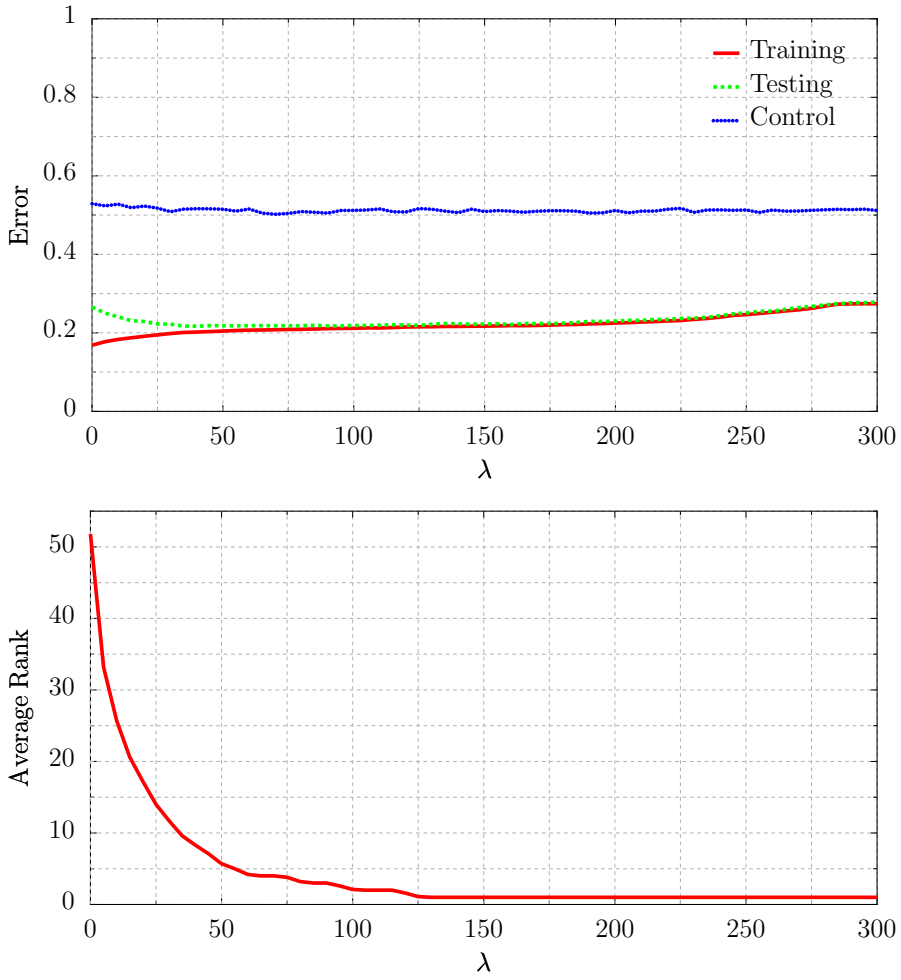


Figure 4.3 Cross Validation and Rank. (a) Training (red) and Testing (green) error curves for varying choices of the regularization parameter λ for our MDS algorithm. Testing error (blue) for the randomized control set. (b) Average rank as a function of the regularization parameter.

expresses preference for three BRDFs as $D_{ij} < D_{jk} < D_{ki} < D_{ij}$, they are being circular in their preferences and these preferences are not representable in a Euclidean space.

Since this type of violation can only be detected if a subject is given all three distance evaluations for a given set of 3 BRDFs, we conducted a study where for each of the 12 subjects a different random set of 40 triplets of BRDFs was randomly selected and the subjects evaluated each of the three distance comparisons.

We found that the violation rate was 1.5% on average with a median of 1%.

We note that each of these error estimates are specific to our dataset, the performance of the same subjects over different tasks and datasets will be different.

4.5.2 The space of gloss perception

In this section we present and analyze the optimal embedding that was obtained by applying the MDS algorithm described in 4.4.1 to our data set.

We ran our MDS algorithm on the data for varying values of λ between 0 and 300, and performed 10-fold cross-validation for each value of λ. Figure 4.3(a) plots the training (Red) and testing error (Green) as a function of the parameter λ. Figure 4.3(b) plots the average rank of the embedding as a function of λ. As expected the rank of the embedding goes down as the regularization is increased.

As an additional check for the fact that our data does indeed contain structure and we are learning from it, we performed the following control experiment. We generated a new dataset by taking each triplet (i, j, k) in our dataset and randomly swapping i and k. This is equivalent to a random observer's response if he were shown exactly the same set of comparisons. We then learnt an embedding for varying values of λ and measured the test error using cross-validation. In Figure 4.3(a) the blue curve plots this error. As can be seen the test error never goes below 50%. The consistent and significant gap between the blue and the green curves indicates that our data set is far from purely random.

The choice of a non-zero λ indicates a tradeoff between the rank of the matrix K and total amount of violation in the paired comparisons. Setting $\lambda = 0$ would focus the attention of the MDS algorithm entirely on reducing the violations. Doing so results in matrix K that has a training error of 17%. The resulting embedding has 53 dimensions (which is only 2 less than the maximum). The cross-validation error for $\lambda = 0$ is 27%. This is a significant gap and indicates the poor generalization ability of this embedding. The algorithm without any regularization is allowed to come up with a complex model that overfits to the noise in the training data resulting in poor performance on test data. As the regularization increases, the training error increases, but the testing error decreases at first and then starts to go back up again. This is because as we increase the penalty for higher rank embeddings, the algorithm trades model complexity for training error. The simpler lower dimensional model does not overfit to the noise leading to an increase in generalization performance. However as the regularization parameter continues increasing the algorithm is biased too strongly towards choosing a low rank embedding , ultimately being restricted to one dimension which is not enough to explain the set of relative comparisons leading to a high test error.

The embedding with the best cross-validation error has a training error of 21.9%

and a test error of 21.3%. The embedding has over 95% of the variance contained in the first two dimensions. Truncating the embedding at two dimensions increased the test error by 0.5%. We do not consider this significant. This embedding is a significant improvement in terms of test error as well as the complexity of the embedding. To put these numbers in perspective, a trivial upper bound on the test error of an embedding is 50% since a purely random predictor or even one that gives the same answer every time will on average get half of the paired comparisons right. Using the L_2 distance between sampled BRDF vectors as the distance function results in 37.5% error and the inter-subject error was 17%.

Further, we analyzed the stability of this embedding. We constructed 55 different embeddings corresponding to leaving the response data corresponding to one of the BRDFs out at a time. Each of the embeddings produced in this manner was then aligned upto a similarity transformation to its corresponding 54 points in the final embedding reported above, and the average squared distortion was measured [115]. Paired comparisons are invariant to similarity transformations. To establish a scale for these errors, the average distance between pairs of points in the global embedding was calculated.

The root mean squared distortion was 2.7e-2 and the average distance between points in the global embedding was 8.7e-1. This is an error of 3% or an order of magnitude difference. This is indicative of the stability of the embedding produced by analyzing our data using the algorithm proposed in this chapter.

Figure 4.4(a) shows the optimal 2-D embedding with cropped windows of the BRDF images displayed in the locations of the BRDF in the new space. Notice the clustering of the BRDFs into two distinct clumps and the similarity amongst the images corresponding to them in each clump. There are also two pronounced trends in the embedding, a vertical trend with the darker BRDFs at the top gradually getting brighter with the brightest BRDFs at the bottom. The other roughly breaks the BRDFs into two clusters: the primarily diffuse BRDFs and those that have a strong glossy or specular component. It is also interesting that the metallic BRDFs are all in the lower left corner and the fabrics are in the upper right corner. This embedding is based entirely on the user preference data, no BRDF or image data was used, which points to the significant descriptive power contained in the paired comparison data.

Figure 4.4(b)-(d) show plots of three of the ASTM gloss dimensions in our embedding space. The position of each circle corresponds to one of the BRDFs in the embedding space and the diameter corresponds to the measurement of the BRDF in the ASTM gloss dimension. We chose to plot contrast gloss, specular gloss and haze since

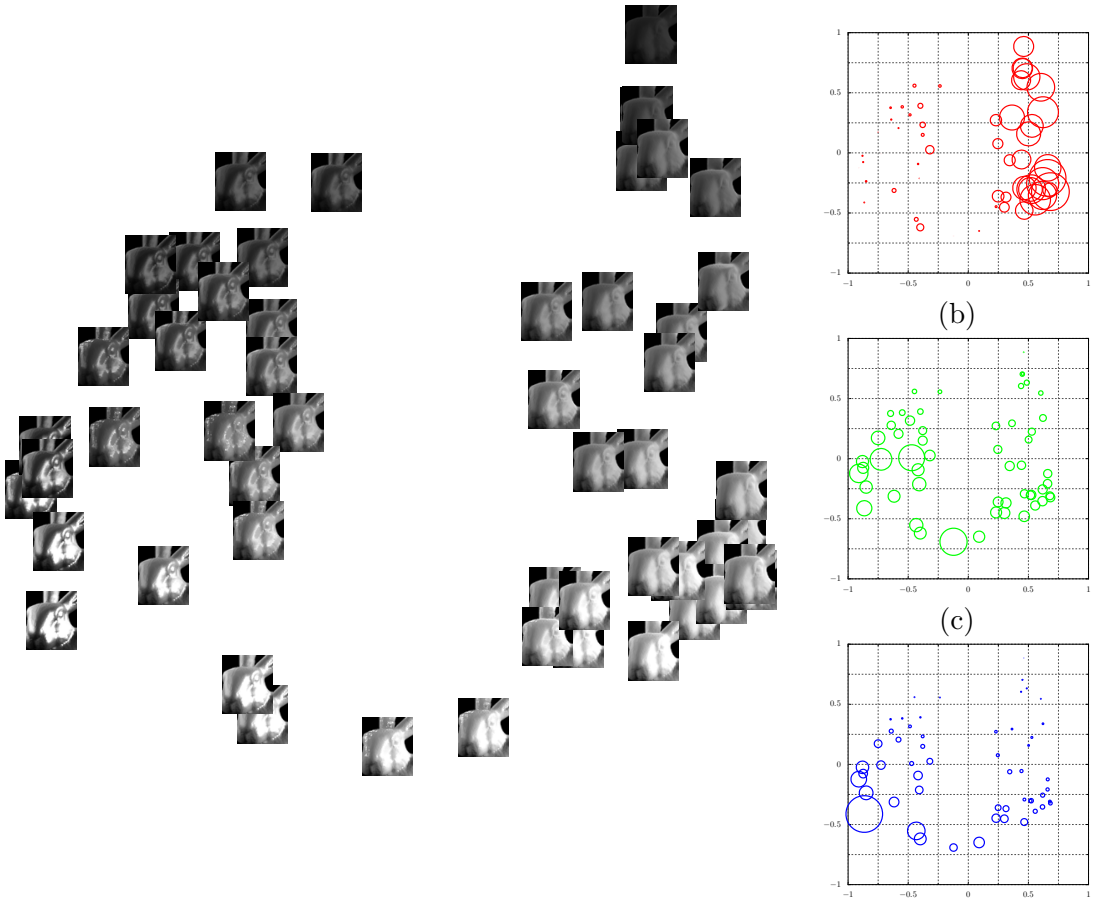


Figure 4.4 Perceptual Embedding. (a) The optimal 2-D embedding with cropped windows of the BRDF images displayed in the locations of the BRDF in the new space. (b-d): Contrast Gloss (b), Specular Gloss (c) and Haze (d) values shown for BRDFs in the embedding. The diameter of the circles corresponds to the value for each property.

they were the ASTM dimensions mentioned as significant in previous work [86, 125].

Figure 4.4(b) shows the measurements of each BRDF for contrast gloss. Notice that there is a strong horizontal trend with contrast gloss increasing from left to right. Since contrast gloss is the ratio of light reflected far from the specular direction to the light reflected in the specular direction, it will be higher for matte materials and lower for materials with a strong specular component.

Figure 4.4(c) shows the measurements of each BRDF for specular gloss at 20°. The measurements exhibit a trend increasing from the lower left corner to the upper right corner. This correlates to the trend we noticed before with the glossy materials on the left and the metallic materials in the lower left corner.

Figure 4.4(d) shows the measurements of each BRDF for haze. There is a

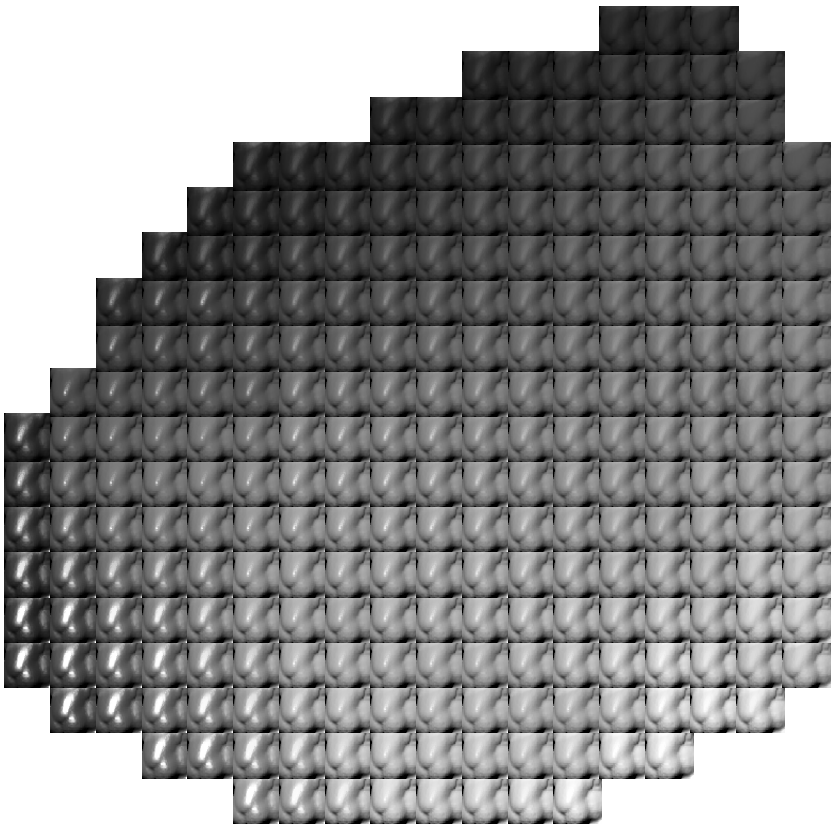


Figure 4.5 Uniform perceptual sampling. The convex hull of the perceptual embedding was resampled, for each point a new BRDF was generated using the perceptual interpolation procedure which was used then to render the nose of the Stanford bunny.

strong trend increasing from the lower left corner to the upper right corner. This is a measure of the light that is reflected 5° off specular and may be less sensitive to noise, which may explain the fewer number of outliers when compared to the measurements taken on specular.

4.5.3 Perceptual Interpolation

As pointed out by Matusik in his thesis, mathematically the set of BRDFs is convex [73], given any two BRDFs $\boldsymbol{x}$ and $\boldsymbol{y}$ and a scalar $0 \leq \mu \leq 1$, $\mu\boldsymbol{x} + (1-\mu)\boldsymbol{y}$ is a mathematically valid BRDF. Thus given a set of BRDFs, measured or otherwise, a simple way to generate new BRDFs is by taking all convex combinations of them. This approach, however, has two problems. First, arbitrary convex combinations, while mathematically correct, can result in physically implausible BRDFs [74]. Second, one is typically interested in producing materials with properties close to some known collection

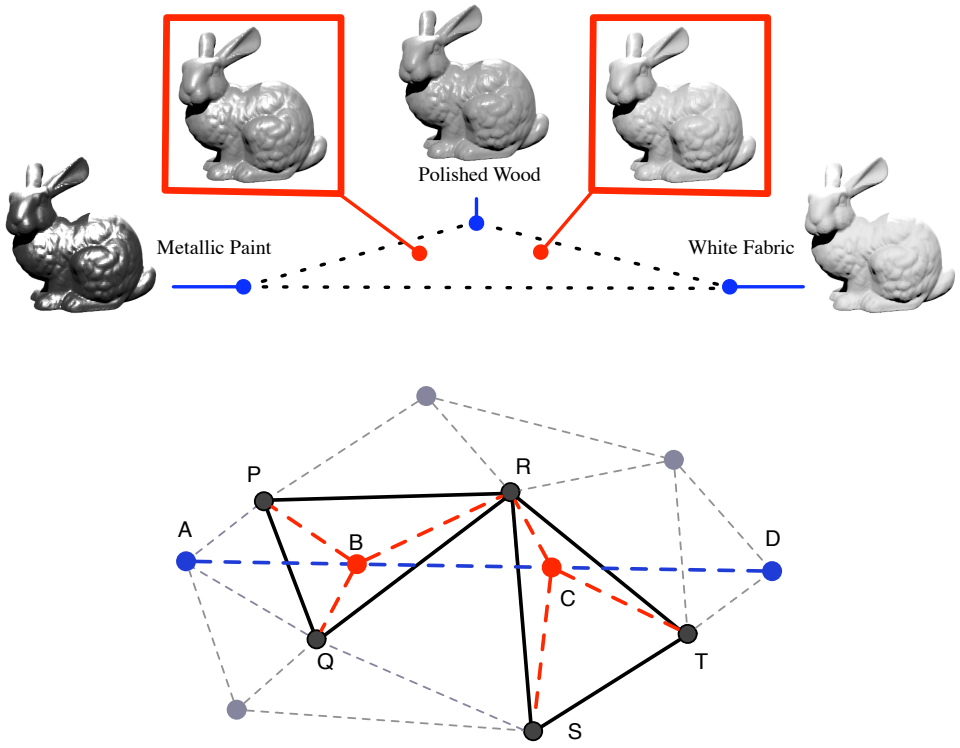


Figure 4.6 Perceptual interpolation. Figure (left) illustrates how a user might use perceptual interpolation to design new materials. He begins by specifying a set of base materials (in this case three) indicated by blue dots and then chooses a point relative to them (indicated in red). This relative position in the perceptual space is used to perceptually interpolate a new BRDF for that point. The image corresponding to the interpolated BRDFs are shown in the red boxes. Figure (right) illustrates the underlying geometrical method used for performing the perceptual interpolation between two BRDFs. Notice that in this case neither of the two end points (A and D) are used in the interpolation process and only one BRDF (R) is shared for the two interpolations (B and C). This illustrates the locally linear yet highly non-linear nature of perceptual interpolation.

of BRDFs; in such case one would like that the combination of weights to correlate with perceptual distance to the basis BRDFs, thus making the combination process intuitive and useful in practice. However, there is nothing to suggest that a linear algebraic combination of two BRDFs translates into a perceptual combination of their properties. Once again it is useful to make an analogy with color perception. There are colors that appear to be both red and blue (purples), both blue and green (blue-greens) and both yellow and green (yellow-greens). There is no color, however, that is subjectively the combination of red and green [85]. Thus we must be careful in our use of linearity

when combining perceptual properties, in particular we should avoid linearly combining objects that are perceptually far apart.

Having a low dimensional perceptual space in which known BRDFs are embedded offers a solution to the problem of perceptual BRDF design. An artist wanting to design a new material can now easily move around in this space and indicate the perceptual position of the BRDF that he wants by indicating how close it is to the known BRDFs. Of course this requires the ability to generate a BRDF from its position in the perceptual embedding. Since we are only given a sparse sampling of the space of BRDFs, we must construct a perceptual interpolation scheme that uses the geometry of the embedding to interpolate over the measured BRDF data.

Given a point in the perceptual space, one naive solution would be to select the k-nearest neighbors of that point from amongst the set of BRDFs. The distance used for determining the neighbors is the Euclidean distance in the perceptual space. The point corresponding to the desired BRDF may or may not lie in the convex hull of its nearest neighbors and its not clear what weighting scheme should be used to actually interpolate between the BRDFs.

Our solution to the problem is to start by first constructing a Delaunay triangulation of the space [83]. Delaunay triangulation constructs a natural neighborhood structure on the embedding by maximizing the minimum angle of all angles of the triangulation. This ensures that long thin triangles connecting far ends of the embedding are avoided. Now when the user specifies a point in the space, we select the Delaunay triangle containing the point and use its Barycentric coordinates to linearly interpolate between the vertex BRDFs. Barycentric coordinates sum to one, thus the resulting interpolant is a convex combination of the vertex BRDF and thereby it is a mathematically valid BRDF. This means for any point inside the convex hull of the embedding, we need at most three BRDFs to generate a perceptual interpolation for it. Figure 4.6(a) illustrates this process.

An interesting consequence of this interpolation scheme is that even if a point in the perceptual space lies on the line joining two BRDFs, the interplolated value of a BRDF could be the result of three entirely different BRDFs. Figure 4.6(b) illustrates this phenomenon. Thus, even though our interpolation process is locally linear, the overall interpolation is a non-linear process. Figure 4.8 shows a comparison between our perceptual interpolation and simple linear interpolation. Note that the primary difference in this case is in the specular peak and the overall impression of glossy vs. matte. In the perceptual interpolation the change in specularity is more gradual whereas the linear

interpolation jumps to a glossy material in just one step from the matte material.

Figure 4.5 shows a collage of images obtained by uniformly sampling within the convex hull of the perceptual embedding. At each point we calculate the interpolated BRDF using the perceptual interpolation procedure described above and use it to render the corresponding image.

We note that if the computational overhead of using measured BRDFs is too much for a particular application, it is simple to replace each measured BRDF with the best fitting empirical model and return to the user the parameters of the best fitting BRDFs at the vertices of the chosen Delaunay triangle and the three combination weights. As our understanding of the space of perception improves and we construct more detailed and perhaps higher dimensional models, our perceptual interpolation scheme will extend naturally. In the higher dimensional case, one can replace the Delaunay triangulation in the plane with the n-space generalization [83], though navigation of a space with more than three dimensions is a tricky user interface design problem.

4.5.4 Integration with Color

While we are aware that the perception of gloss is affected by the surface color, often one can decouple the chromatic and achromatic parts of the BRDF. Following [86], we can integrate our perceptual model of surface gloss with color by assuming that gloss and chromaticity are approximately independent [2,4]. We use our method to interpolate the L channel in perceptual space and then choose two example BRDFs to use for ends of color interpolation in the a and b channels of the CIELAB color space. Figure 4.7 shows an example of this interpolation. Note that while the color interpolation is a simple linear interpolation in color space based on the two endpoint BRDFs, the interpolation for gloss is based on interpolation between 5 intermediate BRDFs as shown in Figure 4.6(b).

4.6 Discussion

In this study we have presented the results of a psychophysical study of the perception of achromatic isotropic reflectance. The study uses the largest publicly available data set of measured reflectances. We introduced a novel MDS algorithm for analyzing the data we collected. This algorithm is a replacement for the widely used weighted non-metric MDS algorithm. It is efficient and optimal in the sense that it find the global minimum to the resulting optimization problem. Analysis of our dataset using this algorithm revealed a two dimensional perceptual embedding. The embedding captures

Figure 4.7 Perceptual Interpolation: four Buddhas rendered in the Galileo environment. The images on the far right and left are rendered from real measured BRDFs (Aluminum Bronze and Teflon, repectively). The images in between are rendered using BRDFs that are constructed by perceptually interpolating the measured BRDFs in our perceptual space.

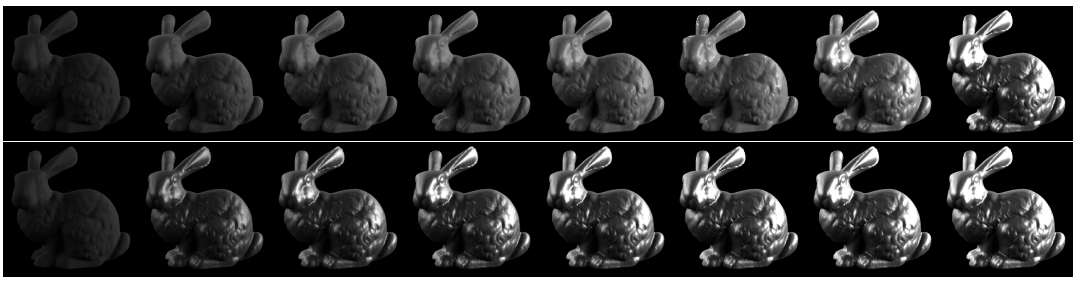


Figure 4.8 Perceptual vs. Linear Interpolation. The top row shows a perceptual interpolation (left to right) from a very matte material (black fabric) to one that is very specular (silver metallic paint). The bottom row shows the linear interpolation. The primary difference in this case is in the specular peak and the overall impression of glossy vs. matte. In the perceptual interpolation (top) the change in the specularity is more gradual whereas linear interpolation (bottom) jumps to a glossy material in just one step from the matte material.

a large fraction of human subject responses indicating that at least the gross structure of the perceptual space of reflectance can be captured by approximating it with a low-dimensional Euclidean space. We also introduced a novel perceptual interpolation scheme that uses the geometric structure of this embedding to perceptually interpolate between BRDFs. This procedure is computationally efficient and locally linear. We showed how this scheme performs better than just linearly interpolating from one target BRDF to another.

We are aware that the small size of the BRDF database places a restriction on the strength of the conclusions we can draw from it. However, our aim in this chapter not just an investigation of the phenomenology of the perception of reflectance but also to propose a methodology and framework for doing such an investigation in the future. We hope that this will set the stage for larger and more elaborate studies of the perception of reflectance.

There are a number of very interesting avenues for future work; we briefly mention some of them here. One interesting direction to explore is the effect of motion on the perception of reflectance. It is often the dynamic reflectance that makes it easier to spot fakes in rendered scenes. It may be interesting to show users rotating geometry and/or cameras to see the effect of the discrimination by the user. How does the scale and magnification of the object affect the ability to discriminate between materials. This has important implications on Level-of-Detail research. Also in talking to the test subjects after the experiment, many of them mentioned that they became quite proficient at quickly examining certain portions of each of the images. It will be interesting to monitor the eye movement of the subjects and study how reflectance and image saliency are related.

The perceptual interpolation procedure described in this chapter is a first step towards constructing an easily navigable space for reflectance. Part of the ease of navigation is due to its two dimensional structure. As we model finer scale structure variations, the dimensionality of this space will likely go up, which will necessitate novel user interfaces for navigating them.

The MDS algorithm we have proposed in this work is a novel and general tool that we expect to have applications in various fields including psychophysics, vision and graphics. It is simple to extend it to other experimental setups like pairs of pairs and complete ranking. A similar formulation can perhaps be used to learn empirical distance functions that will allow us to measure perceptual distance between two previously unseen BRDFs without performing an additional psychophysical study.

This chapter, in full, has been submitted for publication of the material as:
J. Wills, S. Agarwal, D. Kriegman and S. Belongie, "Toward a Perceptual Space for Reflectance," *ACM Transactions on Graphics*, 2006, in review.
I was the primary author and responsible for the design and implementation of the psychophysics experiment, the literature survey, and produced all renderings.

A

Appendices to the Dissertation

A.1 Pseudo-code for Motion Segmentation

In this appendix, we provide pseudo-code with syntax that is similar to Matlab code for the approach to planar motion segmentation that is presented in section 2.1. We will provide the top level motion segmentation routine as well as functions for each of the three stages in our algorithm (point correspondences, motion estimation, and pixel assignment) and for each function the inputs and outputs will be defined.

A.1.1 Motion Segmentation

The function `MotionSeg` computes the layer decomposition for two input images based on the parameters `k` and `lambda`, which control the pixel assignment.

```
[Motions,Assign1,Assign2]=MotionSeg(Im1,Im2,k,lambda)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function MotionSeg computes the layer decomposition for
%     two input images
% Input:
%     Images to segment: Im1, Im2
%     k stdev of gaussian for neighborhood weighting
%     lambda tradeoff between reconstruction error and smoothness
% Output:
%     Array of transformations: Motions
%     Assignment matrices for Im1 and Im2: Assign1, Assign2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Find point correspondences
  [pts1,pts2]=PointCorrespondences(Im1,Im2)

  % Find motions present in set of correspondences
  Motions=EstimateMotion(pts1,pts2)

  % Densely assign pixels to layers
  [Assign1,Assign2]=AssignPixels(Im1,Im2,Motions,k,lambda)
```

A.1.2 Point Correspondences

The function `PointCorrespondences` computes interest points and correspondences for two input images and returns the results as a pair of lists of points where each entry specifies the position of a given point in each of the input images. This list is assumed to be very noisy.

```
[pts1,pts2]=PointCorrespondences(Im1,Im2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function PointCorrespondences computes interest points
%     and correspondences for two input images
% Input:
%     Images to find point correspondences for: Im1, Im2
% Output:
%     Point Locations: pts1, pts2
%     pts1(i) and pts2(i) are corresponding points
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Find corners
  [corners1,corners2]=DetectCorners(Im1,Im2)

  % Perturb interest points
  [pts1,pts2]=PerturbPoints(corners1,corners2)

  % Filter images for point description at corners
  [FR1,FR2]=FilterImages(Im1,Im2, pts1,pts2)

  % Find nearest match in Im2 for each point in Im1
  [pts1,pts2]=PointMatch(FR1, FR2, pts1, pts2)
```

A.1.3 Motion Estimation

The function `EstimateMotion` computes a set of motions for a set of input point correspondences. It assumes there are routines that can compute crowdedness, compute random samples based on the crowdedness at each point, estimate a least squares estimate of a planar homography between 2 sets of points, and count inliers for a given transformation.

```
Motions=EstimateMotion(pts1,pts2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function EstimateMotion computes a set of motions for
%     a set of point correspondences
% Input:
%     Set of point correspondences: pts1, pts2
% Output:
%     Array of transformations: Motions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Compute feature crowdedness
  [crowdedness]=ComputeCrowded(Im1,pts1)

  % Choose a set of random samples based on crowdedness
  %  RandSams is a list of 4-tuples of points (assuming homography)
  RandSams=GetRandSams(pts1, crowdedness)

  % RANSAC
  Foreach tuple in RandSams
      % estimate homography for tuple
      H=EstHomography(pts1(tuple),pts2(tuple)

      % count inliers
      InlierCount=GetInlierCount(pts1,pts2,H)

      % store H and InlierCount
      HResults(count++)=[H,InlinerCount]
```

```
% Sort Hs by inlier counts
Motions=sort(HResults)

% Prune duplicate warps
InlierSet={}
Foreach H in Motions
    inliers=GetInliers(pts1,pts2,H)
    if(overlap(inliers,InlierSet)<OverlapTau)
        Add inliers to InlierSet
    else
        Prune H from Motions
```

A.1.4 Pixel Assignment

The function `AssignPixels` computes the layer assignment for two input images and a set of motions. This function uses a routine that implements the algorithm of Boykov, Veksler, and Zabih [19] and assumes there is a routine to compute reconstruction errors for a set of motion layers an another that computes the intersection of two layer assignments given a set of motions.

```
[Assign1,Assign2]=AssignPixels(Im1,Im2,Motions,k,lambda)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function AssignPixels computes the layer assignment for
%     two input images and a set of motions
% Input:
%     Images to use for assignment: Im1, Im2
%     Array of transformations: Motions
%     k stdev of gaussian for neighborhood weighting
%     lambda tradeoff between reconstruction error and smoothness
% Output:
%     Assignment matrices for Im1 and Im2: Assign1, Assign2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Compute reconstruction error for each motion
  ReconErrs1=ComputeReconErrs(Im1,Im2, Motions)

  % Compute reconstruction error for inverse motions
  ReconErrs2=ComputeReconErrs(Im2,Im1, Inv(Motions))

  % Build similarity matrix
  Loop over all pixels i
      Loop over all pixels j in neighborhood of i
         Sij1=exp( - (distance(i,j)^2)/2k^2  -  pixelDiff(i,j,Im1)^2)
         Sij2=exp( - (distance(i,j)^2)/2k^2  -  pixelDiff(i,j,Im2)^2)

  % Densely assign pixels to layers for forward motion
  [Assign1a]=BVZGraphCut(Im1,Im2,ReconErrs1,Sij1,lambda)
```

```
% Densely assign pixels to layers for backward motion
[Assign2a]=BVZGraphCut(Im2,Im1,ReconErrs2,Sij2,lambda)

% Compute Intersection between forward and backward motions
[Assign1,Assign2]=IntersectAssigns(Assign1a,Assign2a,Motions)
```

A.2 Derivation for the Computation of Normal

We can determine the local angle of incidence θ_i' and the slope of the facet α from incident and refracted vectors, $\vec{\omega}_i$ and $\vec{\omega}_t$. Using the ratio of the indices of refraction $\eta = \eta_t/\eta_i$ and the angle between $\vec{\omega}_i$ and $\vec{\omega}_t$: $\gamma = \theta_i' - \theta_t'$ we will first solve for θ_i'. We begin with Snell's Law:

$$\sin\theta_i' = \eta \sin\theta_t' = \eta \sin(\theta_i' - \gamma)$$

Letting $x = \sin\theta_i'$, we get:

$$\begin{aligned}
x &= \eta x \cos\gamma - \eta\sqrt{1-x^2}\sin\gamma \\
x(1-\eta\cos\gamma) &= -\eta\sqrt{1-x^2}\sin\gamma \\
x^2(1-\eta\cos\gamma)^2 &= \eta^2\sin^2\gamma - x^2\eta^2\sin^2\gamma \\
x^2 &= \frac{\eta^2\sin^2\gamma}{(1-\eta\cos\gamma)^2 + \eta^2\sin^2\gamma} \\
x^2 &= \frac{\eta^2\sin^2\gamma}{1-2\eta\cos\gamma+\eta^2}
\end{aligned}$$

which gives us the following expression for θ_i':

$$\theta_i' = \sin^{-1}\left(\sqrt{\frac{\eta^2\sin^2\gamma}{1-2\eta\cos\gamma+\eta^2}}\right) \tag{A.1}$$

Using θ_i' and $\theta_t' = \theta_i' - \gamma$, we can calculate the new normal, $\vec{n}_f$, as follows: Assuming that $\vec{\omega}_i$ and $\vec{\omega}_t$ are not collinear (a case which makes the determination of $\vec{n}_f$ trivial and unnecessary), $\vec{n}_f$ lies in the plane of these two vectors and we can express it as:

$$\vec{n}_f = x_r\vec{\omega}_t + x_i\vec{\omega}_i$$

We also know that since $\vec{n}_f$ lies on the unit sphere $\|\vec{n}_f\| = 1$. This can be expressed as:

$$\|\vec{n}_f\| = x_r^2 + x_i^2 + 2x_r x_i(\vec{\omega}_t \cdot \vec{\omega}_i) = 1 \tag{A.2}$$

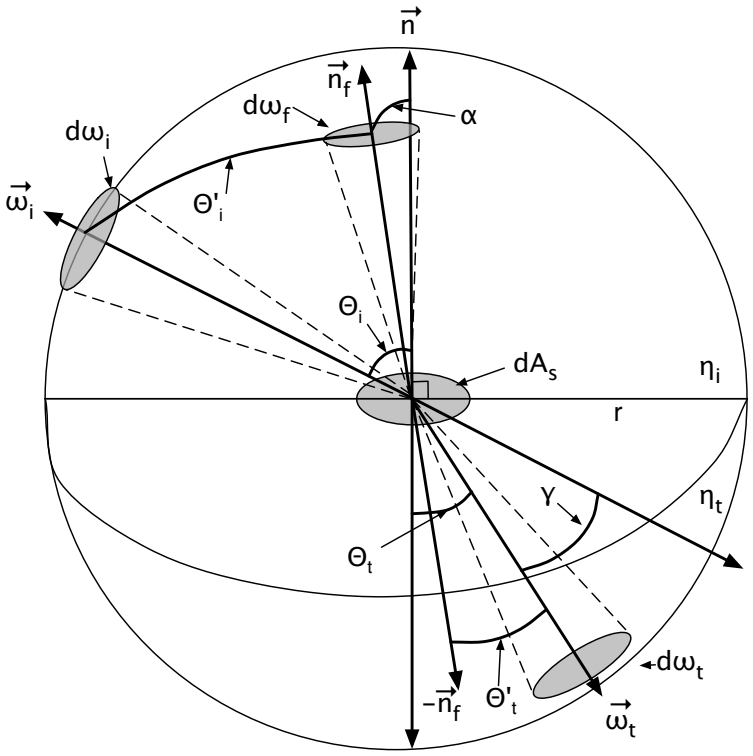


Figure A.1 This is the geometry used in the calculation of θ_i' and α. θ_i' is the local angle of incidence, θ_t' is the local angle of refraction, α is the slope of the reflecting/refracting facet, and $\vec{\omega}_i$ and $\vec{\omega}_t$ are the incident and refracted vectors respectively.

Since $(\vec{n}_f \cdot \vec{\omega}_i) = \cos\theta_i'$,

$$\begin{aligned} x_r(\vec{\omega}_t \cdot \vec{\omega}_i) + x_i &= \sqrt{1 - \frac{\eta^2(1-\cos^2\gamma)}{1+\eta^2-2\eta\cos\gamma}} \\ &= \sqrt{\frac{(\eta\cos\gamma - 1)^2}{1+\eta^2-2\eta\cos\gamma}} \end{aligned}$$

If we let $c_f = \sqrt{\eta^2 - 2\eta\cos\gamma + 1}$ we can solve for x_i in terms of x_r:

$$\begin{aligned} x_i &= \frac{(\eta\cos\gamma - 1)}{c_f} - x_r(\vec{\omega}_t \cdot \vec{\omega}_i) \\ &= \frac{(\eta\cos\gamma - 1)}{c_f} + x_r\cos\gamma \end{aligned}$$

Inserting this value into equation A.2 we get:

$$\begin{aligned} x_r^2 + \left(\frac{\eta\cos\gamma - 1}{c_f} + x_r\cos\gamma\right)^2 + & \\ 2x_r(-\cos\gamma)\left(\frac{\eta\cos\gamma - 1}{c_f} + x_r\cos\gamma\right) &= 1 \\ x_r^2 + x_r^2\cos^2\gamma - 2x_r^2\cos^2\gamma + \left(\frac{\eta\cos\gamma - 1}{c_f}\right)^2 &= 1 \\ (1-\cos^2\gamma)x_r^2 + \left(\frac{\eta\cos\gamma - 1}{c_f}\right)^2 &= 1 \end{aligned}$$

We can then solve for x_r:

$$\begin{aligned} x_r &= \sqrt{\frac{c_f^2 - (\eta\cos\gamma - 1)^2}{c_f^2(1-\cos^2\gamma)}} \\ &= \sqrt{\frac{\eta^2(1-\cos^2\gamma)}{c_f^2(1-\cos^2\gamma)}} \end{aligned}$$

which gives the following values for x_i and x_r:

$$\begin{aligned} x_r &= \frac{\eta}{\sqrt{\eta^2 - 2\eta\cos\gamma + 1}} \\ x_i &= \frac{1}{\sqrt{\eta^2 - 2\eta\cos\gamma + 1}} \end{aligned}$$

This gives us the final expression for $\vec{n}_f$:

$$\vec{n}_f = \frac{\vec{\omega}_i + \eta\vec{\omega}_t}{\sqrt{\eta^2 - 2\eta\cos\gamma + 1}} \tag{A.3}$$

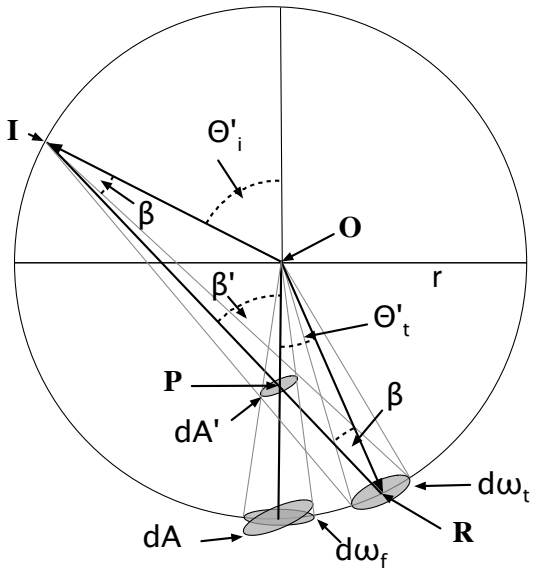


Figure A.2 This is the geometry that is used to calculate $d\omega_f/d\omega_t$. $d\omega_f$ is the solid angle around the normal, $d\omega_t$ is the solid angle around the refracted direction, θ_i' is the angle between the incident direction and the normal, θ_t' is the angle between the refracted direction and the normal

A.3 Derivation for the Solid Angle Relations

We would like to find the relation between the solid angle around the normal and the solid angle around the refracted direction during refraction. To do so, we extend the elegant proof of the relation for the case of reflection by Nayar et al. [77] to the case of refraction. We begin by considering the relation between the solid angle $d\omega_f$ around the normal to the solid angle $d\omega_t$ around the outgoing direction for two arbitrary angles θ_i' and θ_t', where θ_i' is the angle between the incident direction and the normal, and θ_t' is the angle between the outgoing direction and the normal. We will use the geometry of figure A.2.

Since the triangle joining the incident and outgoing directions is isosceles, $\beta = \left(\frac{\theta_i'-\theta_t'}{2}\right)$ which leads to $\beta' = \left(\frac{\theta_t'+\theta_i'}{2}\right)$. In addition, we find that the angle at **IOR** is $\theta_t' + \pi - \theta_i'$.

Using the following relations:

$$\begin{aligned} dA' &= \left(\frac{|\mathbf{IP}|}{|\mathbf{IR}|}\right)^2 d\omega_t \\ dA &= \left(\frac{r}{|\mathbf{OP}|}\right)^2 dA' \\ d\omega_f &= \cos\theta_t' dA \end{aligned}$$

we can arrive at a relation between the solid angles around the normal and the refracted direction:

$$d\omega_f = \cos\theta'_t \left(\frac{r}{|\mathbf{OP}|}\right)^2 \left(\frac{|\mathbf{IP}|}{|\mathbf{IR}|}\right)^2 d\omega_t \tag{A.4}$$

We can then compute the lengths we need:

Using two applications of the Pythagorean theorem:

$$\begin{aligned} |\mathbf{IR}|^2 &= 2r^2 - 2r^2\cos(\pi - \theta'_i + \theta'_t) \\ &= 2r^2(1 - \cos(\pi - \theta'_i + \theta'_t)) \\ &= 4r^2\sin^2\left(\frac{\pi}{2} - \frac{\theta'_i - \theta'_t}{2}\right) \\ &= 2r\cos\left(\frac{\theta'_i - \theta'_t}{2}\right) \end{aligned}$$

using the law of sines:

$$\begin{aligned} |\mathbf{IP}| &= r\sin(\pi - \theta'_i)/\sin\beta' \\ &= r\sin\theta'_i/\sin\left(\frac{\theta'_t + \theta'_i}{2}\right) \\ |\mathbf{OP}| &= r\sin\beta/\sin\beta' \\ &= r\sin\left(\frac{\theta'_i - \theta'_t}{2}\right)/\sin\left(\frac{\theta'_t + \theta'_i}{2}\right) \end{aligned}$$

Substituting the computed values into equation A.4 leads to:

$$d\omega_f = \frac{\cos\theta'_t\sin^2\theta'_i}{4\cos^2\left(\frac{\theta'_i-\theta'_t}{2}\right)\sin^2\left(\frac{\theta'_i-\theta'_t}{2}\right)} d\omega_t = \frac{\cos\theta'_t\sin^2\theta'_i}{\sin^2(\theta'_i - \theta'_t)} d\omega_t$$

We now have the expression for the case of two independent angles. If we then use our knowledge of the relation between the two angles for the case of refraction, namely, $\sin\theta'_i = \eta\sin\theta'_t$, where η is the ratio of indices of refraction, this becomes the following expression:

$$d\omega_f = \frac{\eta^2\cos\theta'_t}{(\cos\theta'_i - \eta\cos\theta'_t)^2} d\omega_t \tag{A.5}$$

Bibliography

[1] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. *SIGGRAPH '03*, 22(3):605–612, 2003.

[2] T. Aida. Glossiness of colored papers and its application to specular glossiness measuring instruments. *Systems and Computers in Japan*, 28(1):95–112, 1997.

[3] M. Ashikhmin, S. Premoze, and P. Shirley. A microfacet-based brdf generator. In *SIGGRAPH*, pages 65–74, 2000.

[4] ASTM. *D3134-97(2003): "Standard Practice for Establishing COLOR and GLOSS TOLERANCEs"*. ASTM International, 2003.

[5] ASTM. *E284-05a: "Standard Termninology of Appearance"*. ASTM International, 2005.

[6] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV 95*, pages 777–784, 1995.

[7] P. Beckmann. Shadowing of random rough surfaces. *IEEE Transactions on Antennas and Propagation*, 13:384–388, 1965.

[8] P. Beckmann and A. Spizzichino. *The Scattering of Electromagnetic Waves from Rough Surfaces*. MacMillan, New York, first edition, 1963.

[9] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[10] S. Belongie and J. Wills. Structure from periodic motion. In *Spatial Coherence for Visual Motion Analysis*, Prague, Czech Republic, May 2004.

[11] D. Best and N. Fisher. Efficient simulation of the von mises distribution. *Applied Statistics*, 24:152–157, 1979.

[12] P. Bhat, K. C. Zheng, N. Snavely, A. Agarwala, M. Agrawala, M. F. Cohen, and B. Curless. Piecewise image registration in the presence of multiple large motions. In *IEEE Conference on Computer Vision Pattern Recognition*, 2006.

[13] M. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *T-PAMI*, 18:972–986, 1996.

[14] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *Siggraph 1977*, volume 11, pages 192–198, July 1977.

[15] F. L. Bookstein. Principal warps: thin-plate splines and decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[16] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Series in Statistics. Springer Verlag, 1997.

[17] G. E. P. Box and M. E. Müller. A note on the generation of random normal deviates. *Ann. Math. Stat.*, 29:610–611, 1958.

[18] Y. Boykov, O. Veksler, and R. Zabih. Approximate energy minimization with discontinuities. In *IEEE International Workshop on Energy Minimization Methods in Computer Vision*, pages 205–220, 1999.

[19] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1222–1239, 2001.

[20] M. Brand. Morphable 3d models from video. In *CVPR01*, pages II:456–463, 2001.

[21] M. Brand and R. Bhotika. Flexible flow for 3d nonrigid tracking and shape recovery. In *CVPR01*, pages I:315–322, 2001.

[22] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 44–51, June 2000.

[23] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *Computer Graphics (SIGGRAPH 1981)*, 15(4):187–196, 1981.

[24] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 15(3):7–24, 1982.

[25] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 2000.

[26] T. Darrell and A. Pentland. Robust estimation of a multi-layer motion representation. In *Proc. IEEE Workshop on Visual Motion*, Princeton, NJ, 1991.

[27] H. A. David. *The Method of paired comparisons*. Chapman and Hall, London, second edition, 1988.

[28] G. Donato and S. Belongie. Approximate thin plate spline mappings. In *Proc. 7th Europ. Conf. Comput. Vision*, volume 2, pages 531–542, 2002.

[29] J. Duchon. Fonction-spline et esperances conditionnelles de champs gaussiens. *Ann. Sci. Univ. Clermont Ferrand II Math.*, 14:19–27, 1976.

[30] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In W. Schempp and K. Zeller, editors, *Constructive Theory of Functions of Several Variables*, pages 85 –100. Berlin: Springer-Verlag, 1977.

[31] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *European Conference on Computer Vision*, pages 563–578. Springer, 1992.

[32] M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In *Proceedings of Americal Control Conference*, June 2004.

[33] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, vol. 24:381–95, 1981.

[34] N. I. Fisher. *Statistical analysis of circular data*. Cambridge University Press, Cambridge, England, first edition, 1995.

[35] R. W. Fleming, R. O. Dror, and E. H. Adelson. Real-world illumination and the perception of surface reflectance properties. *J. Vis.*, 3(5):347–368, 7 2003.

[36] R. W. Fleming, H. W. Jensen, and H. H. Bulthoff. Perceiving translucent materials. In *APGV '04*, pages 127–134, New York, NY, USA, 2004. ACM Press.

[37] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, 1987.

[38] A. R. François, G. G. Medioni, and R. Waupotitsch. Mirror symmetry $\Longrightarrow$ 2-view stereo geometry. *Image and Vision Computing*, 21(2):137–143, February 2003.

[39] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.

[40] J. Gårding. Surface orientation and curvature from differential texture distortion. In *Proc. 5th Int'l Conf. on Computer Vision, Boston*, pages 733–739, 1995.

[41] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

[42] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, 20(6):623–629, June 1971.

[43] A. Gross and T. E. Boult. Analyzing skewed symmetries. *Int'l. Journal of Computer Vision*, 13(1):91–111, 1994.

[44] R. Hall. *Illumination and Color in Computer Generated Imagery*. Springer-Verlag, New York, NY, 1989.

[45] P. Hanrahan and W. Krueger. Reflection from layered surfaces due to subsurface scattering. In *Computer Graphics* Proceedings, Annual Conference Series, ACM SIGGRAPH, pages 165–174, August 1993.

[46] R. I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc Conf. Computer Vision and Pattern Recognition*, 1992.

[47] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[48] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer Verlag, 2001.

[49] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg. A comprehensive physical model for light reflection. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 175–186. ACM Press, 1991.

[50] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg. A comprehensive physical model for light reflection. *Computer graphics (SIGGRAPH 1991)*, 25(4):175–186, 1991.

[51] R. S. Hunter and R. W. Harold. *The measurement of appearance*. Wiley, New York, 1987.

[52] M. Irani and P. Anandan. All about direct methods. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*. Springer-Verlag, 1999.

[53] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, December 1993.

[54] D. Jones and J. Malik. Computational framework to determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10):699–708, Dec. 1992.

[55] T. Kanade. Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence*, 17:409–460, 1981.

[56] C. Kelemen and L. Szirmay-Kalos. A microfacet based coupled specular-matte brdf model with importance sampling. In *Eurographics Conference*, pages 25–34, 2001.

[57] M. Kendall and K. D. Gibbons. *Rank Correlation Methods*. Oxford University Press, 1990.

[58] J. Kleinberg and E. Tardos. Approximate algorithms for classification problems with pairwise relationships: Metric labelling and markov random fields. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 1999.

[59] J. J. Koenderink, A. J. van Doorn, K. J. Dana, and S. K. Nayar. Bidirectional reflection distribution function of throughly pitted surfaces. *International Journal of Computer Vision*, 31(2/3):129–144, July 1999.

[60] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proc. 8th International Conference on Computer Vision*, 2001.

[61] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. *Computer Graphics*, 31(Annual Conference Series):117–126, 1997.

[62] J. H. Lambert. *Photometria, sive De mensura et gradibus luminis, colorum et umbrae*. Number 31–33 in Ostwald's Klassiker der exakten Wissenschaften. W. Engelmann, Leipzig, 1892.

[63] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[64] I. Laptev, S. Belongie, P. Pérez, and J. Wills. Periodic motion detection and segmentation via approximate sequence alignment. In *Proc. ICCV*, 2005.

[65] P. Ledda, A. Chalmers, T. Troscianko, and H. Seetzen. Evaluation of tone mapping operators using a high dynamic range display. In *SIGGRAPH '05*. ACM Press, August 2005.

[66] M. Lhuillier and L. Quan. Match propagation for image-based modeling and rendering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1140–1146, 2002.

[67] Y. Liu, R. Collins, and Y. Tsin. Gait sequence analysis using Frieze patterns. In *Proc. 7th Europ. Conf. Comput. Vision*, 2002.

[68] D. Lowe. Demo software: Invariant keypoint detector. http://www.cs.ubc.ca/spider/lowe/keypoints/.

[69] R. K. M. Pollefeys and L. V. Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *Int'l. Journal of Computer Vision*, 32(1):7–25, 1999.

[70] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *Int'l. Journal of Computer Vision*, 23(2):149–168, 1997.

[71] L. T. Maloney and J. N. Yang. Maximum likelihood difference scaling. *J. Vis.*, 3(8):573–585, 10 2003.

[72] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, and K. E. Torrance. Image–bassed bidirectional reflectance distribution functiona measurement. *Applied Optica-OT*, 39(16):2592–2592600, June 2000.

[73] W. Matusik. *A Data-Driven Reflectance Model*. PhD thesis, MIT, 2003.

[74] W. Matusik, H. Pfister, M. Brand, and L. McMillian. A data-driven reflectance model. In *SIGGRAPH '03*, pages 759–769, New York, NY, USA, 2003. ACM Press.

[75] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *J. Appl. Math. Phys. (ZAMP)*, 5:439–468, 1979.

[76] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142. Springer, 2002. Copenhagen.

[77] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflections: Physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(7):611–634, 1991.

[78] Y. N. A. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. SIAM, Philadelphia, 1994.

[79] F. E. Nicodemus, J. C. Richmond, J. J. Hisa, I. W. Ginsberg, and T. Limperis. *Geometrical Considerations and Nomenclature for Reflectance*. Monograph number 160. National Bureau of Standards, Washington, DC, 1977.

[80] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometric considerations and nomenclature for reflectance. Monograph 161, National Bureau of Standards (US), Oct. 1977.

[81] G. Obein, K. Knoblauch, and F. Vienot. Difference scaling of gloss: Nonlinearity, binocularity, and constancy. *J. Vis.*, 4(9):711–720, 8 2004.

[82] J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2):143–155, 1998.

[83] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tesselations: Concepts and Applications of Voronoi Daigrams*. Wiley, 1992.

[84] M. Oren and S. K. Nayar. Generalization of Lambert's reflectance model. *Computer Graphics*, 28(Annual Conference Series):239–246, 1994.

[85] S. E. Palmer. *Vision Science: Photons to Phenomenology*. MIT Press, 1999.

[86] F. Pellacini, J. A. Ferwerda, and D. P. Greenberg. Toward a psychophysically-based light reflection model for image synthesis. In *SIGGRAPH '00*, pages 55–64., New York, NY, USA, 2000. ACM Press.

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

[88] M. J. D. Powell. A thin plate spline method for mapping curves into curves in two dimensions. In *Computational Techniques and Applications (CTAC95)*, Melbourne, Australia, 1995.

[89] H. S. Sawhney and A. R. Hanson. Trackability as a cue for potential obstacle identification and 3D description. *International Journal of Computer Vision*, 11(3):237–265, 1993.

[90] C. Schlick. An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum*, 13(3):233–246, 1994.

[91] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2003.

[92] S. Seitz and C. Dyer. View morphing. In *SIGGRAPH*, pages 21–30, 1996.

[93] B. Smith. Geometrical shadowing of a random rough surface. *IEEE Transactions on Antennas and Propagation*, 15:668–671, 1967.

[94] A. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *ICML*, 2000.

[95] S. Soatto and A. J. Yezzi. DEFORMOTION: Deforming motion, shape average and the joint registration and segmentation of images. In *European Conference on Computer Vision*, pages 32–47. Springer, 2002. Copenhagen.

[96] Y. Song, L. Goncalves, and P. Perona. Unsupervised learning of human motion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(7):814–827, 2003.

[97] J. Stam. An illumination model for a skin layer bounded by rough surfaces. In *Rendering Techniques '01*, pages 39–52, 2001.

[98] W. A. Stokes, J. A. Ferwerda, B. Walter, and D. P. Greenberg. Perceptual illumination components: A new approach to efficient, high quality global illumination rendering. In *SIGGRAPH '04*, pages 742–749, New York, NY, USA, 2004. ACM Press.

[99] J. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

[100] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. In *IEEE Conference on Computer Vision Pattern Recognition*, pages 194–201, Seattle, Washington, 1994.

[101] R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199–1210, 1996.

[102] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[103] C. Tomasi and T. Kanade. Factoring image sequences into shape and motion. In *Proc. IEEE Workshop on Visual Motion*. IEEE, 1991.

[104] P. H. S. Torr. Geometric motion segmentation and model selection. In J. Lasenby, A. Zisserman, R. Cipolla, and H. Longuet-Higgins, editors, *Philosophical Transactions of the Royal Society A*, pages 1321–1340. Roy Soc, 1998.

[105] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int Journal of Computer Vision*, 24(3):271–300, 1997.

[106] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. In *Seventh International Conference on Computer Vision*, volume 2, pages 983–991, 1999.

[107] P. H. S. Torr, A. Zisserman, and D. W. Murray. Motion clustering using the trilinear constraint over three views. In R. Mohr and C. Wu, editors, *Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision*, pages 118–125. Springer–Verlag, 1995.

[108] K. E. Torrance and E. M. Sparrow. Theory off off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57:1105–1114, Sept. 1967.

[109] L. Torresani, C. Bregler, and A. Hertzmann. Learning non-rigid 3d shape from 2d motion. In *NIPS 2003*, 2003.

[110] L. Torresani and A. Hertzmann. Automatic non-rigid 3d modeling from video. In *ECCV04*, pages Vol II: 299–312, 2004.

[111] L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modelling non-rigid objects with rank constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 493–500, Kauai, Hawaii, 2001.

[112] J. Tumblin, J. K. Hodgins, and B. K. Guenter. Two methods for display of high contrast images. *ACM Trans. Graph.*, 18(1):56–94, Jan. 1999.

[113] J. Tumblin and H. E. Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6):42–48, Nov. 1993.

[114] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *SIGGRAPH '94*, pages 311–318, 1994.

[115] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE-PAMI*, 13(4):376–380, April 1991.

[116] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[117] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[118] R. Vidal and Y. Ma. A unified algebraic approah to 2-d and 3-d motion segmentation. In *Proc. European Conf. Comput. Vision*, Prague, Czech Republic, May 2004.

[119] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[120] J. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 361–366, 1993.

[121] G. J. Ward. Measuring and measurinmodelling anisotropic reflection. *Computer Graphics (SIGGRAPH 1992)*, 26(2):265–272, 1992.

[122] G. J. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics (Siggraph '92 Proceedings)*, 26(3):265–272, 1992.

[123] K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *ICML '04*, pages 839–846, Banff, Canada, 2004.

[124] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 520–526, 1997.

[125] H. B. Westlund and G. W. Meyer. Applying appearance standards to light reflection models. In *SIGGRAPH '01*, pages 501–51., New York, NY, USA, 2001. ACM Press.

[126] J. Wills, S. Agarwal, and S. Belongie. What went where. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, volume 1, pages 37–44, Madison, WI, June 2003.

[127] J. Wills, S. Agarwal, and S. Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. *Int'l. Journal of Computer Vision*, 68(2):125–143, June 2006.

[128] J. Wills and S. Belongie. A feature-based approach for determining long range correspondences. In *Proc. European Conf. Comput. Vision*, volume 3, pages 170–182, Prague, Czech Republic, May 2004.

[129] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *Proc. European Conf. Comput. Vision*, Prague, Czech Republic, May 2004.

[130] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. In *CVPR04*, Washington, D. C. 2004.