ABSTRACT

Title of dissertation: RECOGNITION AND MATCHING IN THE PRESENCE OF DEFORMATION AND LIGHTING CHANGE

Sameer Sheorey, Doctor of Philosophy, 2008

Dissertation directed by: Professor David W. Jacobs
Department of Computer Science

Natural images of objects and scenes show a fascinating amount of variability due to different factors like lighting change, viewpoint change, occlusion and even articulation and non-rigid deformation. Various techniques for object recognition and image matching either try to model these changes or are insensitive to them. There are certain cases like recognition of specular objects and images with arbitrary deformations where existing techniques do not perform well. We aim to develop new techniques to deal with some of these cases.

We propose two different approaches for attacking deformation in images. The first approach is based on matching keypoints in images using histogram descriptors, while the second approach is based on a completely deformation invariant representation for images.

Histograms are a powerful statistical representation for keypoint matching and content based image retrieval. The earth mover's distance

(EMD) is an important perceptually meaningful metric for comparing histograms, but it suffers from high ($O(n^3 \log n)$) computational complexity. We propose a novel linear time algorithm for approximating the EMD for low dimensional histograms using the sum of absolute values of the weighted wavelet coefficients of the difference histogram. EMD computation is a special case of the Kantorovich-Rubinstein transshipment problem, and we exploit the Hölder continuity constraint in its dual form to convert it into a simple optimization problem with an explicit solution in the wavelet domain. We prove that the resulting wavelet EMD metric is equivalent to EMD, i.e. the ratio of the two is bounded and provide estimates for the bounds. The weighted wavelet transform can be computed in time linear in the number of histogram bins, while comparison is about as fast as for the normal Euclidean distance or χ^2 statistic. We experimentally show that wavelet EMD is a good approximation to EMD, has similar performance, but requires much less computation. The same algorithm can be used to compare histograms with unequal mass. We also provide an algorithm that computes the best match between a histogram and a scaled version of another histogram. For practical evaluation of these techniques, we have a C++ implementation of the fast Lifting Wavelet transform algorithm for arbitrary dimensional histograms.

An image of a non-planar object can undergo a large non-linear deformation due to a viewpoint change. Complex deformations occur in images of non-rigid objects, for example, in medical image sequences. We

propose using the *contour tree* as a novel framework invariant to arbitrary (smooth) deformations for representing and comparing images. The contour tree encodes the arrangement of the iso-intensity contours of an image and is invariant to arbitrary deformations since it does not depend on the shape of the contours. It represents all the deformation invariant information in an image. Computing the edit distance between two trees gives us a measure of the deformation invariant distance between the two corresponding images. This distance measure can also take into account various other difficulties of image matching, such as noise, occlusion and lighting changes.

Lighting changes greatly affect the appearance of all objects and make recognition difficult. Recognition of specular objects is particularly difficult because their appearance is much more sensitive to lighting changes than that of Lambertian objects. We consider an approach in which we use a 3D model to deduce the lighting that best matches the model to the image. In this case, an important constraint is that incident lighting should be non-negative everywhere. We propose a new method to enforce this constraint and explore its usefulness in specular object recognition, using the spherical harmonic representation of lighting. The method follows from a novel extension of Szego's eigenvalue distribution theorem to spherical harmonics, and uses semidefinite programming to perform a constrained optimization. The new method is faster as well as more accurate than previous methods. Experiments on both syn-

thetic and real data indicate that the constraint can improve recognition of specular objects by better separating the correct and incorrect models.

RECOGNITION AND MATCHING IN THE PRESENCE OF DEFORMATION AND LIGHTING CHANGE

by

Sameer Sheorey

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

Advisory Committee:
Professor David W. Jacobs, Chair/Advisor
Professor Eitan Tadmor, Dean's representative
Professor Larry Davis
Professor Rama Chellappa
Professor Ramani Duraiswami

Dedication

To my mother

Acknowledgments

The five years that I spent at Maryland for my dissertation have revealed a new and fascinating intellectual world to me. I wish to thank many people without whose moral and intellectual support this thesis would not have been possible.

Most of all, I am deeply grateful to my advisor Prof. David Jacobs who has given me more patience and freedom than can be expected from any advisor. He encouraged me to spend time pursuing my own research problems even when he was unsure of any productive outcomes. His perceptive insights into the fairly muddled situations I often found myself in have rescued me innumerable times. I am also thankful to him for the financial support that has enabled me to complete my degree. I also enjoyed working on the Electronic Field Guide project with Profs. Peter Belhumeur, Steven Feiner and Ravi Ramamoorthi and with John Kress, Ida Lopez, Sean White, Haibin Ling, Nandan Dixit, Daozheng Chen, Anne Jorsted and others.

I am grateful to my committee members Profs. Larry Davis, Eitan Tadmor, David Mount, Ramani Duraiswami, Yiannis Aloimonos and Rama Chellappa for their questions and comments on my thesis. I also appreciate the faculty members who taught me various courses. I particularly enjoyed the courses I took with Profs. David Jacobs, David Mount, Howard Elman, Rama Chellappa and P.S. Krishnaprasad.

My life at Maryland was made very simple by the people who kept reminding of all the forms I needed to fill and everything else I needed to do as a PhD student. I am especially grateful to Fatima Bangura, Janice Perrone, Mary Kay, Arlene Gonsales, Yerty Valenzuela, Margit Gedra, Mary Lewis, Sue Blandford, Jennifer Story, Brenda Chick and Jessica Touchard.

The friends, colleagues and companions I found at Maryland will always be in my memory. I have enjoyed talking with them about research and all the mundane things in life. I especially want to thank Kaushik, Soma, Gaurav, Gaurav, Narayanan, Shiv, Ashok, Dikpal, Haibin, Ashwin, Jagan and Arun. I will always remmeber the afternoon foosball games with Gaurav, Soma, Kaushik and Dikpal that were so much fun.

I especially want to thank my roommates Prithvi, Alap, Arun, Vinay, Vijay, Christopher, Ashok, Abhi and Som for cooking for me and consenting to eat the food that I cooked. I am thankful to Prithvi for sharing my enthusiasm for long drives, hiking and nature.

Finally, I am deeply grateful to my family – my mother, brother and Sarika; without whom I cannot imagine my life and who deserve most of the credit for this work.

Table of Contents

List of Tables vii

List of Figures viii

1 Introduction 1
1.1 Matching images with deformations . . . 2
1.1.1 Mass transportation problems . . . 3
1.1.2 General deformation invariance . . . 7
1.2 Recognition of specular objects . . . 8

2 Histogram comparison: Approximate EMD in linear time 11
2.1 Introduction . . . 11
2.2 Related Work . . . 18
2.3 Theory . . . 22
2.3.1 Continuous EMD and its dual . . . 23
2.3.2 EMD in the wavelet domain . . . 26
2.3.3 Why not the Fourier transform ? . . . 30
2.4 EMD for partial histograms . . . 32
2.4.1 Kantorovich-Rubinstein extension . . . 33
2.4.2 Rubner's EMD for signatures . . . 34
2.4.3 Hanin's partial EMD formulation . . . 36
2.4.4 WEMD for partial histograms . . . 38
2.4.5 Best partial match . . . 41
2.5 Improving WEMD consistency . . . 45
2.6 Experiments . . . 49
2.6.1 Some implementation notes . . . 49
2.6.2 Which wavelets ? . . . 53
2.6.3 Image retrieval: colour histograms . . . 57
2.7 Conclusion and future work . . . 62
2.7.1 Image registration . . . 63
2.A Proof of Lemma (1) . . . 67
2.B WEMD with biorthogonal wavelets . . . 75
2.C Wavelet transform implementation . . . 79

3 General deformation invariant matching 82
3.1 Introduction . . . 82
3.2 Related work . . . 85
3.3 The contour tree . . . 87
3.3.1 Construction . . . 90
3.3.2 Relation to the GIH . . . 92
3.3.3 Factors affecting the contour tree . . . 93
3.4 Comparing contour trees: edit distance . . . 96
3.5 Future work . . . 100

3.5.1 Experiments . . . 100
3.5.2 Multiresolution contour tree . . . 100

4 Recognition of specular objects 102
4.1 Introduction . . . 102
4.2 Past Work . . . 106
4.3 The non-negativity constraint . . . 108
4.3.1 The Fourier case . . . 109
4.3.2 Spherical Harmonics . . . 113
4.4 Recovering Lighting from an Image: Semidefinite Programming . . . 125
4.5 Experiments . . . 127
4.5.1 Implementation . . . 127
4.5.2 Experiments on Synthetic Images . . . 131
4.5.2.1 Variation of error with model specularity and query image frequency . . . 131
4.5.2.2 Fooling LIN . . . 133
4.5.3 Experiments on Real Images . . . 133
4.5.3.1 Reflectance Model Construction . . . 134
4.5.3.2 Shiny rubber ball . . . 136
4.5.3.3 Ceramic shaker . . . 138
4.6 Conclusion and Future Work . . . 139
4.A Spherical Harmonics . . . 140
4.A.1 Complex Spherical Harmonics . . . 140
4.A.2 Real Spherical Harmonics . . . 142
4.B Semidefinite Programming . . . 144
4.C Theorem of Carathéodory . . . 146

Bibliography 147

List of Tables

2.1 Correspondence between EMD for signatures, discrete EMD and continuous EMD for probability distributions 19

2.2 Theoretical (loose) error bound estimates 54

2.3 EMD approximation error for random histograms 56

2.4 Hanin (partial) EMD extension approximation error for random histograms 58

2.5 Error and time requirements for colour histograms 59

4.1 Speed comparison of SDP and Delta function method [Basri and Jacobs, 2003]. 130

4.2 Matching error: correct vs. uniform model 138

List of Figures

2.1 Some applications of histogram descriptors 13

2.2 Computation of wavelet EMD 14

2.3 Equivalent measure to Rubner's EMD extension 40

2.4 Subdifferential 44

2.5 Wavelet best partial match EMD 46

2.6 Undecimated WEMD and histogram shift 47

2.7 Lifting wavelet transform 52

2.8 Sample images of the different classes in from the SIMPLIcity dataset. 59

2.9 WEMD vs. Indyk-Thaper: error 60

2.10 WEMD vs. Indyk-Thaper: error in ordering 61

2.11 Colour histograms for content based image retrieval: wavelet EMD performance compared to other EMD methods 66

3.1 Examples of image deformation 83

3.2 A simple synthetic image and its contour tree 87

3.3 Converting a 2D square grid to a simplicial mesh 91

3.4 Different images but same GIH 93

4.1 Recognition error with specular objects 104

4.2 Specular Object Recognition Algorithm 129

4.3 Error vs query image frequency and model specularity 132

4.4 Shiny Rubber Ball 136

4.5 Ceramic shaker (LIN vs SDP) 137

Chapter 1

Introduction

Object recognition and image matching are difficult problems in computer vision because of the immense variability displayed by images of natural scenes. These large changes are caused by various factors like lighting, viewpoint change and changes in the imaged scene itself. A variety of techniques have been developed to deal with these variations. For example, changes due to lighting can be addressed by modelling the reflectance properties of the object and representing lighting by spherical harmonics [Basri and Jacobs, 2003], [Zhang and Samaras, March 2006]. In image matching, pose variation and articulation are manifested as deformations in the image and are usually dealt with using interest points and feature descriptors [Lowe, 2004], [Matas et al., 2002], etc. Matching interest points using their descriptors gives a sparse set of point correspondences that can be used to infer the pose or articulation state of the object. However, there are various missing links in this set of techniques. Recognition of objects under specular (mirror-like or directional) reflection, recognition of translucent objects and recognition in the presence of cast shadows are quite difficult using current techniques. Similarly, matching images under arbitrary deformations is also a difficult problem.

We aim to develop new techniques for a few of these missing links in this dissertation.

A proper choice of representation for the involved quantities goes a long way in solving a problem. *Harmonic analysis* is an important branch of mathematics that deals with the representation of functions in terms of a basis. It encompasses, among other things, Fourier series for representing periodic functions, spherical harmonics for representing functions defined on the sphere and wavelets for compactly representing functions belonging to many different classes. We will use some nice properties of these basis functions to help us deal with variability in images.

Let us now look at the specific problems that we aim to address in this dissertation.

1.1 Matching images with deformations

Factors like viewpoint change, articulation and non-rigid deformation that affect images can be treated under the common approach of deformation invariant image matching. In this proposal, we consider two different ways of dealing with deformations. The first approach is based identifying corresponding points between image pairs using histogram based feature descriptors. This is a popular approach for image matching [Mikolajczyk and Schmid, 2005]. We propose a new fast method for

matching histogram descriptors based on mass transportation problems. We also propose to explore new applications of this approach to image registration, which is a closely related problem to deformation invariant image matching. The second approach involves computing an image descriptor that is completely invariant to all smooth one to one deformations but retains all the deformation invariant information in the image.

1.1.1 Mass transportation problems

A mass transportation problem is the problem of determining how to *move* a probability distribution as economically as possible so that it coincides with another. An obvious but naive way of determining if two images are related by a deformation is to use appropriately normalized versions of the two images as probability distributions in a mass transportation problem The solution will then tell us how to deform one image into another. This image registration technique was used by [Haker et al., 2004]. An appropriate metric between the two images can reveal if the images are actually related by a deformation. This approach is not likely to work well in the presence of additional factors of variability like lighting and noise. However, we can use mass transportation problems in vision in two ways:

1. To compare histogram based feature point descriptors and obtain a sparse feature correspondence between two images. We can com-

pare histogram descriptors by solving a mass transportation problem since they are probability distributions. The solution is known as the earth mover's distance [Rubner et al., 2000] (EMD) between histograms. We describe a fast approximation algorithm for the EMD in this proposal.

2. Most image registration algorithms align images by maximizing the dependence between some feature distributions around corresponding points [Zitova and Flusser, 2003]. Mutual information is typically used as the measure of dependence. This can be substituted by the EMD between the joint distribution and product of marginal distributions [Chefd'hotel and Bousquet, 2007]. Using our fast EMD algorithm for registration is future work.

Comparing histogram descriptors: Histogram descriptors are a powerful representation for matching and recognition. Their statistical nature gives them sufficient robustness while maintaining discriminative power. They have been used extensively in vision applications like shape matching [Belongie et al., 2002], keypoint matching [Lowe, 2004], texture analysis [Lazebnik et al., 2005] and 3D object recognition [Johnson and Hebert, 1999]. Colour and texture histograms [Rubner et al., 2000] are also used for content based image retrieval. These descriptors are often compared using binwise dissimilarity measures like Euclidean or other L_p norms or the χ^2 statistic. While these measures can be computed very fast and of-

ten give good results, they do not take into account all possible variations in the random variables whose distributions they compare. On the other hand, crossbin distance measures consider the fact that histograms are based in feature space and it is possible for histogram mass to *move* between bins in feature space. They penalize this movement according to the distance covered, called the *ground distance*. The earth mover's distance (EMD) is a natural and intuitive metric between histograms if we think of them as piles of sand sitting on the ground (feature space). Each grain of sand is an observed sample. To quantify the difference between two distributions, we can measure how far the grains of sand have to be moved so that the two distributions coincide exactly. *EMD is the minimal total ground distance travelled weighted by the amount of sand moved* (called *flow*). EMD has been successfully used for image retrieval by comparing colour and texture histograms [Rubner et al., 2000], contour matching [Grauman and Darrell, 2004], image registration [Chefd'hotel and Bousquet, 2007], [Haker et al., 2004] and pattern matching in medical images [Holmes et al., 2002a], [Holmes et al., 2002b]. However, a major hurdle to using EMD is that it is computed by solving a linear program called the *transportation simplex* with a computational complexity of $O(n^3 \log n)$ (for an n-bin histogram).

We propose a novel method for approximating the EMD for histograms using a new metric on the weighted wavelet coefficients of the difference histogram. We show that this is equivalent to EMD, i.e. the

ratio of EMD to wavelet EMD is always between two constants. Although our estimates for these constants are loose, we will show experimentally that our metric follows EMD closely and can be used instead without any significant performance difference. The wavelet EMD metric can be computed in $O(n)$ time. We arrive at this approximation when we look at the dual of the transportation simplex. The objective function is now an inner product and the constraint is that the slack variable is a Hölder continuous function. Both can be expressed in the wavelet domain: the objective function exactly, but the constraint only approximately. The resulting wavelet domain optimization problem has a simple explicit solution.

Intuitively speaking, the wavelet transform splits up the difference histogram according to scale and location. Each wavelet coefficient represents an EMD subproblem that is solved separately. The sum of all distances is an approximation to EMD. This turns out to be a good approximation because the wavelet transform is well suited for splitting up a function according to scale and location.

We also show that the same algorithm can be used to compare histograms of unequal mass. This is required when the two histograms to be compared are constructed from a different number of samples, for example colour histograms of images of different sizes. It may not be a good idea to simply normalize the two histograms to the same total mass since the larger image may contain the smaller image as a part. The two images may be at different scales as well. So, we also give a fast algo-

rithm to compare a smaller histogram with an optimal proportion of the larger histogram. This is the best partial match algorithm [Holmes et al., 2002b].

Future Work : We want to test this for matching keypoint descriptors and for image registration.

1.1.2 General deformation invariance

An image of a curved object can undergo large non-linear deformations due to a viewpoint change. Non-linear deformations also occur because of changes in the object itself, for example in medical images of the body organs. Deformations can be treated by selecting distinctive interest points and computing descriptors around them. The descriptors can be made invariant to affine deformations [Mikolajczyk and Schmid, 2005]. The geodesic intensity histogram (GIH) [Ling and Jacobs, 2005] is a descriptor invariant to arbitrary deformations. However, this descriptor does not use all the deformation invariant information in the image. Matching a sparse set of descriptors will not provide us with a dense correspondence field either.

We propose to use a tree based descriptor that captures all the deformation invariant information in an image. If we consider the image as a surface embedded in 3D space, the *contour tree* [Carr et al., 2000] describes the topology of this surface, i.e. the local extrema values and

their relative locations. The contour tree is invariant to arbitrary image deformations and contains all the deformation invariant information about the image. We can compare two images in a deformation invariant way by computing the tree edit distance between their trees. This also enables us to deal with occlusion by using partial tree matching.

Future work : We want to make the image comparison more robust to noise, lighting changes and spatial discretization effects by appropriately modifying the tree edit distance. Further, we want to specialize the algorithm to certain types of deformations – for example those caused by viewpoint changes.

1.2 Recognition of specular objects

Lighting changes significantly affect the appearance of objects. Specular or shiny objects are affected much more than objects with diffuse (for example Lambertian) reflection. We consider the problem of identifying an object from a single image with unknown lighting, from several different objects of known structure and reflectance properties. This problem can be solved by treating reflection in a signal processing framework. The incident light (*input signal*) gets reflected from the object (*filtered*) to form the image (*output signal*). This process is easy to analyze in the frequency domain. Lighting is defined on the sphere since it is a function of direction in 3D space. Spherical harmonics are the appropriate basis

functions for representing functions defined on the sphere and are analogous to the Fourier complex exponential basis for periodic functions (i.e. functions defined on the circle).

Lambertian reflectance acts as a low pass filter and only the lower harmonics of the incident light are reflected. Specular reflection also reflects a lot of higher order harmonics. This difference is clear if we look at a ceramic pot. Before glazing, it appears almost uniformly bright, and it is hard to say much about the light sources illuminating it. High frequency details are suppressed. After it is glazed, most of the surrounding room is reflected quite clearly in it. As a filter, it now reflects high frequency details without much attenuation.

Identifying the object that produced the query image is an inverse problem. For each object in our model database, we compute the lighting that creates an image closest to the query image using least squares optimization over all frequency domain lighting. The object that produces the closest image is the most likely match. Since Lambertian objects heavily attenuate higher order harmonics, we only need to compute low frequency lighting. Experimentally, lighting up to order 2 (the first 9 spherical harmonic components) has been found to be sufficient [Basri and Jacobs, 2003]. For specular objects, the same method can be used since specular reflection is also a low pass filter, although one with much slower attenuation. We need to compute lighting up to a much higher order.

Unfortunately, the greater degrees of freedom available to lighting is abused to produce low errors for the wrong objects as well. The image of a white ball with a dark spot is well approximated by a plain white ball with high frequency lighting that contains a negative spot. This can be prevented by ensuring that the lighting is always non-negative. We propose a method to do this in the frequency domain during optimization. We experimentally show that this enables better discrimination between the correct and incorrect models, especially in the presence of image noise.

The non-negativity constraint is based on Szego's eigenvalue distribution theorem [Grenander and Szego, 1958] for Fourier series. If we form a Toeplitz matrix from the first few Fourier series coefficients of a function, then its eigenvalues, on the average, take on values picked uniformly from the function values. By constraining this matrix to be positive semidefinite, we can ensure that the function values are non-negative for some possible completion of the Fourier series. We extend this result to spherical harmonic series and use the non-negativity constraint in the optimization. Instead of a linear least squares problem, we now have a semidefinite programming (SDP) problem, that can still be solved efficiently (polynomial time) using standard SDP software.

Chapter 2

Histogram comparison: Approximate EMD in linear time

2.1 Introduction

Histogram descriptors: Histogram descriptors are a powerful representation for matching and recognition. Their statistical nature gives them sufficient robustness while maintaining discriminative power. They have been used extensively in vision applications like shape matching [Belongie et al., 2002], keypoint matching [Lowe, 2004], texture analysis [Lazebnik et al., 2005], 3D object recognition [Johnson and Hebert, 1999] and tracking [Zivkovic and Kröse, 2004]. Colour and texture histograms [Rubner et al., 2000] are also used for content based image retrieval. Figure (2.1) illustrates some of these applications in computer vision. These descriptors are often compared using binwise dissimilarity measures like Euclidean or other L_p norms or the χ^2 statistic. While these measures can be computed very fast and often give good results, they do not take into account all possible variations in the random variables whose distributions they compare. These unmodelled variations may lead to large measure values for changes in the distribution that are perceived to be small. For example, suppose we take two photos of a plain wall with strong and weak sunlight and compare their colour histograms. The histograms are

shifted delta functions and have large binwise differences. Consequently, all of these measures will give large values. The popular SIFT descriptor [Lowe, 2004] is a gradient orientation – location histogram. A similar histogram shifting will occur if the keypoint is not localized accurately. This problem is reduced by using a low number of bins (just 4×4 position and 8 orientation). As indicated in [Lowe, 2004], performance decreases if the number of bins are increased. This can be avoided by using a cross bin histogram distance like EMD.

Earth mover's distance: Cross-bin distance measures take into account the fact that histograms are based in feature space and it is possible for histogram mass to *move* between bins in feature space. They penalize this movement according to the distance covered, called the *ground distance*. The earth mover's distance (EMD) is a natural and intuitive metric between histograms if we think of them as piles of sand sitting on the ground (feature space). Each grain of sand is an observed sample. To quantify the difference between two distributions, we can measure how far the grains of sand have to be moved so that the two distributions coincide exactly. *EMD is the minimal total ground distance travelled weighted by the amount of sand moved* (called *flow*). EMD makes sure that shifts in sample values are not penalized excessively. For the example of a shifted delta function, the EMD is simply the shift amount. EMD has been successfully used for image retrieval by comparing colour and

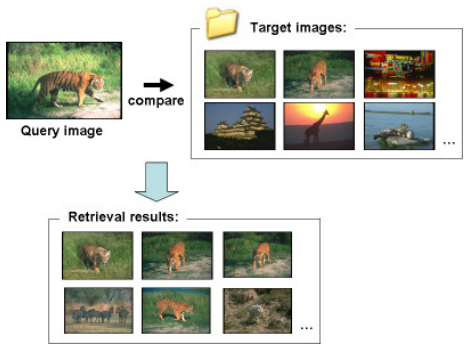
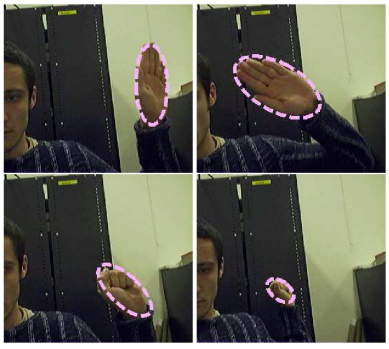
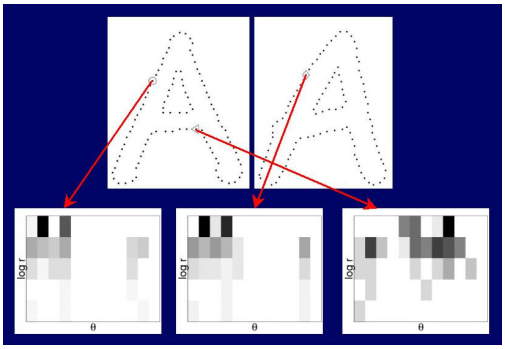


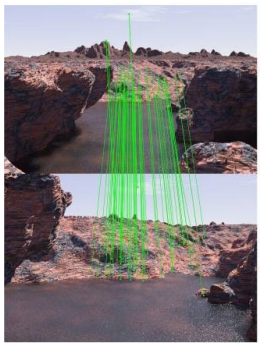
Image retrieval



Tracking



Shape matching



Keypoint matching

Figure 2.1: Some applications of histogram descriptors. Images courtesy of [Wang et al., 2001], [Belongie et al., 2002], A. Vedaldi and [Zivkovic and Kröse, 2004]

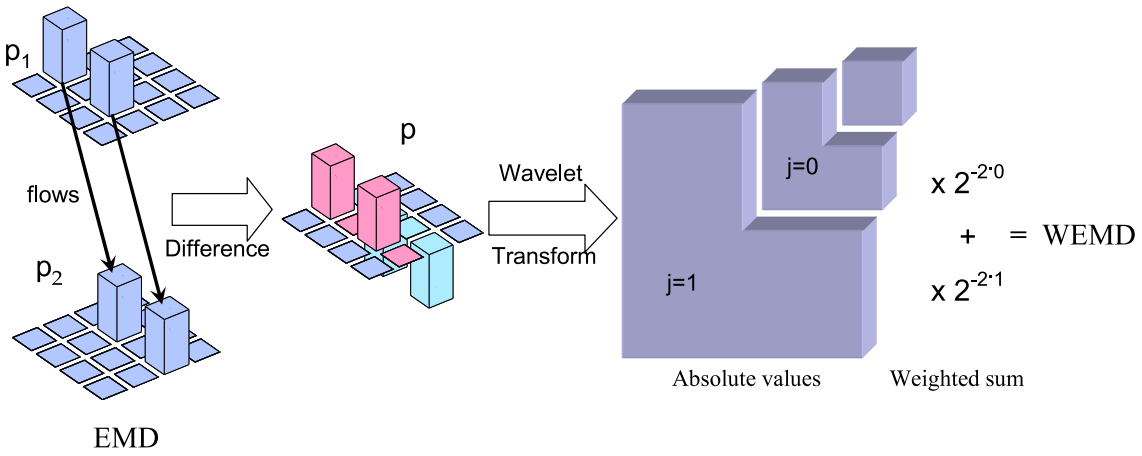


Figure 2.2: Computation of wavelet EMD

texture histograms [Rubner et al., 2000], contour matching [Grauman and Darrell, 2004], image registration [Chefd'hotel and Bousquet, 2007], [Haker et al., 2004] and pattern matching in medical images [Holmes et al., 2002a], [Holmes et al., 2002b]. However, a major hurdle to using EMD is its $O(n^3 \log n)$ computational complexity (for an n-bin histogram).

Wavelet EMD: We present a novel method for approximating the EMD for histograms p_1 and p_2 using a new metric on the weighted wavelet coefficients of the difference histogram. We show that this is equivalent to EMD, i.e. the ratio of EMD to wavelet EMD is always between two constants. Although our estimates for these constants are loose, we will show experimentally that our metric follows EMD closely and can be used instead without any significant performance difference. The wavelet EMD metric can be computed in $O(n)$ time.

EMD can be computed as the minimal value of a linear program. The Kantorovich-Rubinstein (KR) transshipment problem [Rachev and Rüschendorf, 1998] is the corresponding problem for continuous distributions. Both problems admit duals with the same optimal value. The important insight in our algorithm is that the dual of the KR problem has a wavelet domain representation with a simple explicit solution.

In the primal form, the objective function is the total flow–weighted ground distance between all bin pairs. See table (2.1) for exact definitions. The flows must make up for the difference between the histograms at each corresponding bin. In the dual form, the optimization is over a potential f assigned to each bin. For a difference histogram $p := p_1 - p_2$, the dual EMD is given by :

$$\text{Dual EMD} := \sup_f \int f(x)p(x)dx \tag{2.1}$$

subject to the constraint that the difference between two bin potentials is bounded by the ground distance $c(x, y) = \|x - y\|$, i.e. $f(x) - f(y) \leq \|x - y\|$. The objective function is the maximum inner product between the potential function and the difference histogram and is easily represented in the wavelet domain, since orthonormal wavelets preserve inner products. The constraint means that f cannot grow faster than a diagonal line at any point. This is actually a Hölder (or Lipschitz) continuity condition and is somewhat between continuity and differentiability. The wavelet coefficients of a Hölder continuous function decay exponentially at fine

scales, since fine scale wavelets represent rapid changes in the function. We thus have an equivalent constraint in the wavelet domain. The resulting optimization is trivial and gives an explicit solution :

$$d(p)_{wemd} := \sum_{\lambda} 2^{-j(1+n/2)} |p_\lambda| \tag{2.2}$$

p is the n dimensional difference histogram and p_λ are its wavelet coefficients. The index λ includes shifts and the scale j. We will call this the *wavelet EMD* between two histograms. This is clearly a metric. This is not exactly equal to the EMD since the Hölder continuity constraint can't be transformed exactly into the wavelet domain.

This surprising formula for approximating the EMD with wavelet coefficients of the difference histogram is one of the main contributions of this thesis. By using appropriate wavelets, we can approximate EMD very well. Since the wavelet transform is a common linear time operation, we can compute this in time linear in the number of bins for uniform histograms. Figure (2.2) explains the EMD approximation algorithm in 2D.

Intuitively speaking, the wavelet transform splits up the difference histogram according to scale and location. Each wavelet coefficient represents an EMD subproblem that is solved separately. The sum of all distances is an approximation to EMD. This is a good approximation because wavelet transforms are well suited for separating local variations according to scale and position. For a single wavelet coefficient, the mass

to be moved is proportional to $|p_\lambda|2^{-jn/2}$, since this would be the wavelet coefficient if we use wavelets normalized by total mass, i.e. $\int|\psi_\lambda| = 1$. The distance travelled is proportional to the span of the wavelet 2^{-j} (According to Meyer's [Meyer, 1992] convention, a wavelet at scale j is the mother wavelet squeezed 2^j times.) The total EMD is thus approximated by equation (2.2).

Approximation by scale and location separation is similar to the way packages are shipped over large distances. The total journey is broken into several hops – short and long. Short hops connect the source and destination to shipping hubs, while long hops connect the shipping hubs themselves. Packages from nearby towns merge at shipping hubs to travel together. Thus, the package journey is split into multiple scales, and the sum of the distances travelled is an approximation to the actual distance.

Next in section (2.4), we look at different extensions of EMD to the case of partial histograms, i.e. when we are not able to gather sufficient samples to construct a full histogram. Rubner's approach in this case does not give us a metric and we are not able to approximate it using wavelets. However, another approach presented by Hanin [Hanin, 1997] produces a metric that can be approximated by the same algorithm. We will also present an $O(N \log N)$ algorithm to match an optimal fraction of the larger histogram to the smaller histogram. In section (2.5), we will look at ways to improve the approximation using the undecimated or

stationary wavelet transform. Finally, section (2.6) presents experiments to validate this theory.

2.2 Related Work

The earth movers distance was introduced in vision by Werman *et al.* [Werman et al., 1985], though they did not use this name. Rubner *et al.* [Rubner et al., 2000] extended this to comparing *signatures*: adaptive histograms of varying mass represented by weighted clusters. They computed the EMD using a linear program called *transportation simplex* and used it for content based image retrieval by comparing colour signatures. They obtained better performance than binwise measures. This method has an empirical time complexity between $O(n^3)$ and $O(n^4)$. EMD being a transportation problem, can also be modelled as a network flow problem ([Korte and Vygen, 2000] chapter 9) in graph theory. The two histograms are represented by a single graph with a vertex for each bin and ground distances as the edge weights. The two histogram vertices act as sources and sinks respectively with bin contents as values. Computing EMD is now an *uncapacitated minimum cost flow problem* and can be solved by Orlin's algorithm ([Korte and Vygen, 2000] section 9.5) in $O(n^3 \log n)$ time.

Various approximation algorithms have been suggested to speed up the computation of EMD. Ling and Okada [Ling and Okada, 2006b] empirically showed that EMD could be computed in $O(N^2)$ time if an L_1

EMD for signatures [Rubner et al., 2000]	**Discrete EMD for histograms**	**Continuous EMD for distributions**
Signatures $f(i;1), f(i,2)$	Histograms $f(i;1), f(i,2)$	Distributions $p_1(x), p_2(x)$
In general, $\sum_i f(i;1) \neq \sum_i f(i;2)$	$\sum_i f(i;1) = \sum_i f(i;2) = 1$	$\int p_1(x)dx = \int p_2(x)dx = 1$
	Difference $f(i) := f(i;1) - f(i;2)$	Difference $p(x) := p_1(x) - p_2(x)$
Ground distance $d_{ij} \geq 0$	Ground distance $d_{ij} \geq 0$	Cost function $c(x,y) \geq 0$
Flow (from bin i to bin j) $g_{ij} \geq 0$	Flow (from bin i to bin j) $g_{ij} \geq 0$	Joint distribution $q(x,y) \geq 0$
	Potential π_i	Potential $f(x)$
EMD $:= \min \frac{\sum_{ij} g_{ij} d_{ij}}{\sum_{ij} g_{ij}}$	EMD $:= \min \sum_{ij} g_{ij} d_{ij}$	EMD $:= \inf \int c(x,y) q(x,y) dxdy$
s.t. $\sum_j g_{ij} \leq f(i;1), \sum_i g_{ij} \leq f(i;2)$,	s.t. $\sum_i g_{ik} - \sum_j g_{kj} = f(k)$	s.t. $\int q(u,y)dy - \int q(x,u)dx = p(u)$
$\sum_{ij} g_{ij} = \min(\sum_i f(i;1), \sum_i f(i;2))$		
	Dual EMD $:= \max \sum_i \pi_i f(i)$	Dual EMD $:= \sup \int f(x)p(x)dx$
	s.t. $\pi_i - \pi_j \leq d_{ij}$	s.t. $f(x) - f(y) \leq c(x,y)$

Table 2.1: Correspondence between EMD for signatures, discrete EMD and continuous EMD for probability distributions

ground distance is used instead of the usual Euclidean distance. They used the EMD for comparing different histogram descriptors and noted improved performance compared to χ^2 and Euclidean distance.

Indyk and Thaper [Indyk and Thaper, 2003] use a randomized multiscale embedding of histograms into a space equipped with the l_1 norm. The multiscale hierarchy is obtained by a series of random shifting and dyadic merging of bins. The histogram levels are weighted by powers of 2, with more weight at the coarser levels. They show that the l_1 norm computed in this space, averaged over all random shifts, is equivalent to the EMD. They do not prove this for individual random embeddings, and also do not estimate the constants that bound the ratio of this norm to EMD. They couple this with locality sensitive hashing for fast nearest neighbour image retrieval using colour signatures. Grauman and Darrell's pyramid match kernel [Grauman and Darrell, 2005] is based on this method. They use histogram intersection instead of l_1 distance at each level and inverted weights to obtain a similarity measure useful for matching partial histograms instead of a metric. Both these methods have a time complexity of $O(Tdm \log D)$ for d dimensional histograms with diameter D and m bins. The random embeddings are computed T times. Although these algorithms are fast, our algorithm gives deterministic error bounds. We will also show empirically that our algorithm is more accurate.

The diffusion distance introduced by Ling and Okada in [Ling and

Okada, 2006a] is computed by constructing a Gaussian pyramid from the difference histogram and summing up the L_1 norms of the various levels. Although this has some similarities with our algorithm, it is not an approximation to the EMD and may behave differently.

Holmes and Taylor [Holmes et al., 2002b], [Holmes et al., 2002a] use partial signature matching based on the EMD for identifying mammogram structures. They embed signatures into a learned Euclidean space to speed up computation. They find the *best partial match* that matches a fraction of one signature to another. We will show a fast way of computing the best partial match using our wavelet approximation.

The continuous EMD problem and its generalizations are based in probability theory for comparing distributions and have been studied since Nobel prize winner L. V. Kantorovich's [Rachev and Rüschendorf, 1998] first formulation of the problem as a linear program and the study of its duality in 1942. In this area, various equivalent formulations of EMD are minimal l_1 metric, Kantorovich-Rubinstein (KR) metric [Rachev and Rüschendorf, 1998], Wasserstein distance and Mallows distance [Levina and Bickel, 2001]. General mass transportation problems have wide applications in mathematical economics, recursive stochastic equations for studying convergence of algorithms and stochastic differential equations. Hanin [Hanin, 1997] proposed an extension of the KR metric to the case of partial histograms that preserves most of the properties of the original KR metric.

Wavelets have been shown to characterize a large and varied set of functions spaces [Meyer, 1992], i.e. a function belongs to a specific class if and only if a particular norm defined only on the magnitude of its wavelet coefficients is finite. This remarkable property of wavelets, particularly applicable to Hölder spaces, allows us to use wavelets in approximating the KR metric. The most popular application of wavelets is in compression and de-noising. It has been shown [Coifman and Donoho, 1995] that using the undecimated wavelet transform results in improved de-noising. We will use this result and show that we can improve the consistency of our approximation using the same technique.

2.3 Theory

The earth mover's distance is a metric between two probability distributions for metric ground distances. It is a special case of a class of optimization problems in applied probability theory called *mass transportation problems*. We will first look at the analogy between discrete and continuous EMD and state the dual form (section 2.3.1). Then, in section (2.3.2), we will describe how to convert the dual form into the wavelet domain. The wavelet domain dual problem has an explicit solution.

2.3.1 Continuous EMD and its dual

The wavelet domain connection of the EMD problem becomes clear only when we look at EMD for continuous distributions. Table (2.1) lists analogous terms between EMD for signatures and discrete and continuous versions of the EMD problem for distributions. The problem is simpler for histograms than for signatures because they must add up to 1. The objective function is simpler because the total flow $\sum_{ij} g_{ij} = 1$. The constraint is simpler as well and means that the flows must make up the difference between the two histograms. This is a mass conservation constraint. We will now formally state the continuous domain EMD problem [Rachev and Rüschendorf, 1998], summarized in the third column of table (2.1).

Let P_1 and P_2 be probability distributions with densities p_1 and p_2 respectively, defined on a compact space $S \subset \mathbb{R}^n$. More generally, In general, we can consider P_1 and P_2 to be non-negative *Borel* measures on S, i.e. $P_1, P_2 \in M_+(S)$. c is a continuous cost function on the Cartesian product space $S \times S$. Here, we will restrict c to be of the form $\|x - y\|^s$ with $0 < s \leq 1$. $s = 1$ gives us the usual Euclidean ground distance. Thus, c is always a norm. The Kantorovich-Rubinstein transshipment problem (KRP) is to find

$$\dot{\mu}_s = \inf_q \int ||x - y||^s \, q(x, y) dx dy \tag{2.3}$$

where the infimum is over all joint probability distributions Q with den-

sity q on $S \times S$. Q is analogous to flow in the discrete EMD problem and specifies how the source density p_1 is moved to the target density p_2. Thus the joint density q must satisfy the mass conservation constraint :

$$p_1(u) - p_2(u) = \int q(u, y)dy - \int q(x, u)dx \tag{2.4}$$

$p := p_1 - p_2$ is a difference density with the property that $\int p = 0$. The corresponding distribution P thus belongs to the class of Borel measures $M_0(S)$ with a total measure 0. The *Kantorovich–Rubinstein theorem* states that the problem admits the dual representation :

$$\dot{\mu}_s = \sup_f \int f(x)(p_1(x) - p_2(x))dx \tag{2.5}$$

with the same optimal value. The supremum is over all bounded continuous functions f on S (called potentials) satisfying the order s Hölder continuity condition

$$f(x) - f(y) \leq ||x - y||^s \quad \text{for all } x, y \in S \tag{2.6}$$

In the dual form, the EMD is the supremum of inner products of the difference density with a suitably smooth function.

Going back to the piles of sand, in the primal form, we try to find the flows q to convert p_1 into p_2 that move the sand by the least amount (2.3). In the dual form, we try to assign heights or potentials f to the various bins that will drive these flows. If we limit the change in the potentials by the ground distance (2.6), we can measure the total sand movement by the change in total height of the sand pile (2.5).

The potential f thus belongs to a *homogeneous Hölder space* of functions of order s denoted by $\dot{C}^s$. This is also referred to as a *homogeneous Lipschitz space* denoted by $\dot{\text{Lip}}s$. Hölder space membership is an indication of the global smoothness of a function. For $0 < s < 1$, a bounded, continuous function f belongs to the homogeneous Hölder class $\dot{C}^s(\mathbb{R}^n)$ if the following supremum exists and is finite :

$$C_H(f) := \sup_{x \neq y} \frac{|f(x) - f(y)|}{\|x - y\|^s} \tag{2.7}$$

This defines the Hölder seminorm of f. This is not a norm because it assigns zero length to all constants. We can now state the constraint (2.6) simply as

$$C_H(f) < 1 \tag{2.8}$$

The corresponding *inhomogeneous Hölder space* is equipped with the following norm:

$$\|f\|_{C^s} := \max\{C_H(f), \max_x |f(x)|\} \tag{2.9}$$

A simpler way of expressing the KR duality is in the form of the following Cauchy-Schwartz inequality :

$$|\langle f, p \rangle| := \left| \int f(x)p(x)dx \right| \leq \dot{\mu}_s(p) C_H(f) \tag{2.10}$$

The KR metric is a norm in the space of probability distributions $M_0(S)$ while C_H is a seminorm for the homogeneous Hölder space. The KR duality thus establishes an isometry between these two spaces, i.e. we can obtain the norm of a function in one space as the maximum inner product with all functions of unit norm in the corresponding dual space.

2.3.2 EMD in the wavelet domain

Now we will look at expressing the dual problem in the wavelet domain. We can identify the various classes that a function belongs to by observing the rate of decay of its wavelet coefficients [Meyer, 1992] (Chapter 6). For our application, we are interested in the wavelet characterization of Hölder spaces, since the potential f belongs to one. First we will explain some notation about the wavelet series representation of a function.

A function f in $\mathbb{R}^n$ can be expressed in terms of a wavelet series (Meyer [Meyer, 1992] Chapter 2) as:

$$f(x) = \sum_k f_k \phi(x-k) + \sum_\lambda f_\lambda \psi_\lambda(x) \tag{2.11}$$

ϕ and ψ are the scaling function and wavelet respectively. k runs through all integer n–tuples and represents shifts, and $\lambda := (\epsilon, j, k)$. In n dimensions, we need $2^n - 1$ different wavelet functions which are indexed by ϵ. They are usually constructed by a tensor product of 1D wavelet functions along individual dimensions. For example, in 2D, we have horizontal ($\epsilon = 1$: $\psi(x)\phi(y)$), vertical ($\epsilon = 2$: $\phi(x)\psi(y)$) and diagonal ($\epsilon = 3$: $\psi(x)\psi(y)$) wavelets. j represents the scale and is a non-negative integer. Larger values of j mean finer scales with shorter wavelet functions. The set of all possible λ for a scale $j \geq 0$ is denoted by Λ_j and Λ is the union of all Λ_j. We thus have

$$\psi_\lambda(x) := 2^{nj/2} \psi^\epsilon(2^j x - k) \tag{2.12}$$

A wavelet ψ has regularity $r \in \mathbb{N}$ if it has derivatives up to order r and

all of them (including ψ itself) have *fast decay*, i.e. they decay faster than any reciprocal polynomial for large x. For orthonormal wavelets, the coefficients can be computed as

$$f_k = \int f(x)\bar{\phi}(x-k)dx, \quad k \in \mathbb{Z}^n \tag{2.13}$$

$$f_\lambda = \int f(x)\bar{\psi}_\lambda(x)dx, \quad \lambda \in \Lambda, \quad j \geq 0 \tag{2.14}$$

$\bar{\phi}$ and $\bar{\psi}$ are complex conjugates of ϕ and ψ respectively.

The following theorem from Meyer ([Meyer, 1992] section 6.4) can be used to characterize functions in $C^s(\mathbb{R}^n)$:

Theorem 1. *A function $f \in L^1_{loc}(\mathbb{R}^n)$, (i.e. $|f|$ is integrable over all compact subsets of $\mathbb{R}^n$) belongs to $C^s(\mathbb{R}^n)$ if and only if, in a wavelet decomposition of regularity $r \geq 1 > s$, the approximation coefficients f_k and detail coefficients f_λ satisfy*

$$|f_k| \leq C_0, \quad k \in \mathbb{Z}^n \quad \textit{and}$$

$$|f_\lambda| \leq C_1 2^{-j(n/2+s)}, \quad \lambda \in \Lambda_j, \quad j \geq 0 \tag{2.15}$$

for some constants C_0 and C_1.

A little modification to the proof of this theorem (see 2.A) gives the following lemma:

Lemma 1. *For $0 < s < 1$, if the wavelet series coefficients of the function f are bounded as in (2.15), then $f \in C^s$ with $C_H(f) < C$ such that*

$$a_{12}(\psi; s)C_1 \leq C \leq a_{21}(\psi; s)C_0 + a_{22}(\psi; s)C_1 \tag{2.16}$$

for some positive constants a_{12}, a_{21} *and* a_{22} *that depend only on the wavelet and* s*. For discrete distributions defined on a lattice, the same condition holds for* $s = 1$ *as well.*

The constants a_{12}, a_{21} and a_{22} are estimated in 2.A. Now we have all the ingredients necessary for our main result :

Theorem 2. *Consider the KR problem with the cost function* $c(x, y) = \|x - y\|^s,\ s < 1$*. Let* p_k *and* p_λ *be the wavelet transform coefficients (approximation and detail, respectively) of the difference density* p *generated by the orthonormal wavelet-scaling function pair* ψ *and* ϕ *with regularity* $r \geq 1 > s$*. Then for any non-negative constants* C_0 *and* $C_1 > 0$*,*

$$\hat{\mu}_s = C_0 \sum_k |p_k| + C_1 \sum_\lambda 2^{-j(s+n/2)} |p_\lambda| \tag{2.17}$$

is an equivalent metric to the KR metric $\dot{\mu}_s$*; i.e. there exist positive constants* C_L *and* C_U *(depending only on the wavelet used) such that*

$$C_L \hat{\mu}_s \leq \dot{\mu}_s \leq C_U \hat{\mu}_s \tag{2.18}$$

For discrete distributions, the same result holds for $s = 1$ *as well.*

Proof. Consider the auxiliary wavelet domain problem :

$$\begin{aligned} &\text{Maximize } \mathbf{p}^T\mathbf{f} = \sum_k p_k f_k + \sum_\lambda p_\lambda f_\lambda \\ \text{subject to} \quad &|f_k| \leq C_0 \quad \text{and} \quad |f_\lambda| \leq C_1 2^{-j(s+n/2)} \end{aligned} \tag{2.19}$$

$\mathbf{p}$ and $\mathbf{f}$ are coefficient vectors of p_λ and f_μ. It is easy to see that $\hat{\mu}_s$ in (2.17) is the solution of this problem. We need to show that the ratio

of the optimal values of the KR problem and auxiliary wavelet problem are bounded by two constants C_L and C_U. Since we use orthonormal wavelets that preserve inner products, the wavelet problem (2.19) has the same objective function as the KR problem dual (2.5).

Note that changing the KR dual problem constraint $C_H(f) < 1$ to $C_H(f) < K$ for any $K > 0$ will simply have the effect of scaling the optimal value by K, since for every function f allowed by the original constraint, there is a corresponding function Kf allowed by the new constraint. Further, the constraints in the auxiliary problem (2.19) will allow functions with $C_H(f) < C$, where C is bounded by the limits in (2.16). So, all functions with $C_H(f)$ less than the lower bound in (2.16) are included by the constraint, and no function with $C_H(f)$ greater than the upper bound are included. Consequently, the optimal value is scaled by a factor C that obeys the bounds in (2.16). This is equivalent to (2.18) with

$$C_L = a_{12}(\psi; s)C_1 \quad \text{and}$$
$$C_U = a_{21}(\psi; s)C_0 + a_{22}(\psi; s)C_1. \tag{2.20}$$

The wavelet EMD metric is thus equivalent to EMD. Note that C_1 must be strictly positive to ensure that the lower bound is non-zero.

Since our lemma is valid for discrete distributions, i.e. distributions defined on a lattice, for $s = 1$, this result is valid as well. A similar but more complex result holds for biorthogonal wavelets as well. See 2.B for details. □

We set $C_0 = 0$ because this gives us the tightest bounds in (2.16). Setting the constant C_1 to 1, we get the simple distance measure :

$$d(p)_{wemd} := \hat{\mu}_s\Big|_{C_0=0,C_1=1} = \sum_{\lambda} |p_\lambda| 2^{-j(s+n/2)} \tag{2.21}$$

$$\text{The bounds ratio } \frac{C_U}{C_L} = \frac{a_{22}(\psi; s)}{a_{12}(\psi; s)} \tag{2.22}$$

measures the maximum possible error. After scaling wavelet EMD suitably, the ratios WEMD/EMD and EMD/WEMD will always be less than the bounds ratio. We will use this fact to estimate the bounds ratio experimentally.

This formula also specifies an embedding into a space equipped with the l_1 norm, since wavelet EMD can be computed as the l_1 norm between the weighted wavelet coefficients of the two histograms. This fact is very useful for information retrieval applications. We can embed all histogram features in our database into this weighted wavelet coefficient space. Processing a query consists of embedding it in the same space and finding the nearest l_1 neighbours. It is easy to apply sub-linear time retrieval techniques like Locality Sensitive Hashing (LSH) [Andoni and Indyk, 2008] since this space is equipped with the l_1 norm.

2.3.3 Why not the Fourier transform ?

At this point, it is clear that a wavelet representation can enable us to approximate EMD because of its effective characterization of the Hölder

continuity of a function. Wavelets provide a tight (*if and only if*) characterization that enables us to construct an equivalent wavelet domain norm.

We know that a Fourier transform also characterizes Hölder or Lipschitz continuity. Let $\hat{f}_k$ be the kth Fourier series coefficient of the function f. The two principal results concerning Fourier series and Lipschitz functions are ([Mallat, 1998] Theorem 6.1 and [Zygmund, 2002] Chapter 2 Section 4) :

$$\begin{aligned} \text{For } s > 0, \sum |\hat{f}_k|(1+|k|^s) < \infty &\implies f \in C^s(\mathbb{R}) \\ &\implies |\hat{f}_k| \leq \frac{K}{1+|k|^s} \text{ for some } K > 0 \end{aligned} \tag{2.23}$$

Neither of these two conditions gives a complete characterization of the Hölder space, the first includes extra functions that are not Hölder continuous while the second excludes some Hölder continuous functions. Hence, we cannot use the Fourier characterization for approximating EMD. Another orthonormal representation can be used instead of wavelets if it provides a tight characterization of Hölder continuity.

Intuitively speaking, the Fourier transform is not very good at localizing features or judging distances between them. So it cannot be used to measure distances between places of excess and deficit mass that computing the EMD requires.

The Fourier transform does give a tight characterization if Lipschitz continuity is defined as a global average using an integral. See ([Titch-

marsh, 1948] theorem 85) for details. We cannot use this since we need a stricter maximal characterization of Lipschitz continuity.

2.4 EMD for partial histograms

For many applications, it may not be possible to gather enough data samples to construct a complete histogram. Colour histograms for images of different sizes will have different number of samples. We are still required to compare two histograms constructed from a different number of data samples. The trivial method of renormalizing the two histograms to the same number of samples is correct only if we can be sure that the measured data samples were picked uniformly from the histogram domain. This is rarely the case. For example, while constructing the colour histogram of an image patch, the colour values are sampled according to image content and will often be clustered together for nearby pixels. If part of the image patch is occluded, part of the histogram is likely to be missing.

We will first look at Kantorovich and Rubinstein's original extension to deal with partial histograms. This is not suitable in many cases. We will then look at how Rubner's signature EMD and examine its relation with the KR extension. Although this allows for a dual representation, it cannot be directly converted into the wavelet domain. Next we will look at Hanin's extension of the KR metric that surprisingly preserves our

current wavelet domain algorithm.

2.4.1 Kantorovich-Rubinstein extension

Lets first take a look at how different quantities change when we have partial histograms. Our probability distributions p_1 and p_2 are now unnormalized non–negative *Borel* measures (i.e. they belong to the space $M_+(S)$). The difference distribution $p = p_1 - p_2$ now belongs to the space of general signed Borel measures $M(S)$ on S. We no longer have $\int_S p(x)dx = 0$, i.e. p need not belong to $M_0(S)$ anymore. As a result, the joint density $q(x, y)$ representing flows is also no longer normalized.

Kantorovich and Rubinstein's extension, referred to as the K-norm in [Guittet, 2002], assigns a constrained *waste* function $w(x)$ for unmatched mass left at the point x. Wasting extra mass is costlier than transporting it anywhere else

$$w(x) > \sup_{y \in S} d(x, y) \quad \forall x \in S \tag{2.24}$$

and it is not worthwhile to transfer mass to another position just to waste it there.

$$|w(x) - w(y)| \leq d(x, y) \quad \forall x, y \in S \tag{2.25}$$

The extended KR metric is a combination of the cost of transporting matching mass and wasting the rest.

$$\|p\|_{w,d} = \inf_{p_0 \in M_0(S)} \left\{ \|p_0\|_d + \int w(x)|p(x) - p_0(x)|dx \right\} \tag{2.26}$$

This extension reduces to the original KR norm in the case that $p \in M_0(S)$, since there is no waste.

The KR extension may not be physically realistic in many cases since the waste cost depends on the size of the domain (2.24). This extension cannot be used for unbounded domains at all. The next two extensions get around this limitation by ignoring the constraint (2.24).

2.4.2 Rubner's EMD for signatures

Rubner's original EMD formulation [Rubner et al., 2000] for signatures deals with partial histograms by minimizing the ratio of movement work to total flow, i.e.

$$EMD_{\text{Rubner}} := \min_{q} \frac{\iint d(x,y)q(x,y)dxdy}{\iint q(x,y)dxdy} \tag{2.27}$$

$$\text{subject to} \quad \int q(x,y)dy \leq p_1(x), \quad \int q(x,y)dx \leq p_2(y) \tag{2.28}$$

$$\text{and} \quad \iint q(x,y)dxdy = \min\left\{\int p_1(x)dx, \int p_2(y)dy\right\} \tag{2.29}$$

The flow is constrained so that all mass is transferred from the smaller distribution. Note that this imposes no penalty for the unmatched part of the larger histogram.

Rubner's EMD for signatures is no longer a metric since it ignores extra positive mass. The EMD between two different histograms can be zero if their difference is non-negative so there is no need to move any mass. It is not a semi-norm either as it does not obey the triangle inequality. The EMD between any two histograms p_1 and p_2 and the zero

histogram are both zero, while the EMD between them can be positive.

This can be simplified by normalizing p_1 and p_2 so that the smaller distribution has unit measure again. This scales the EMD by the normalizing value. Without loss of generality, we can assume that $\int p_2(x)dx = 1$ and $\int p_1(x)dx \geq 1$. This implies $\int p(x)dx \geq 0$. Since all mass from p_2 is to be transferred, we have $\iint q(x,y)dxdy = 1$, i.e. q is a proper joint probability density as before. Finally, the constraint $\int q(u,y)dy - \int q(x,u)du \leq p(u)$ means that overall, we cannot remove more mass from a bin at point u than the excess $p(u)$. The optimization will not allow transferring mass from a bin with mass deficit ($p(x) < 0$) to a bin with mass excess ($p(x) > 0$) since this increases the EMD without affecting the constraints. So, this single inequality actually implies these two constraints :

$$\begin{aligned} \int q(u,y)dy - \int q(x,u)du &= p_1(u) - p_2(u) \quad \text{if} \quad p_1(u) \leq p_2(u) \\ \int q(u,y)dy - \int q(x,u)du &\leq p_1(u) - p_2(u) \quad \text{otherwise} \end{aligned} \tag{2.30}$$

For metric costs, these two inequalities are equivalent to the two inequalities (2.28). (2.28) can be violated while satisfying (2.30) if q removes more mass from bin u than present and then puts it back again from other bins. Such a procedure can only increase EMD if we use a metric ground distance.

We can write the simplified problem as :

$$\begin{aligned} EMD_{\text{simple}} &:= \inf_q \iint d(x,y)q(x,y)dxdy \\ \text{subject to} \quad \int q(u,y)dy - \int q(x,u)du &< p(u) \end{aligned} \tag{2.31}$$

Note that this is the same as the KR extension with a zero waste function.

This is rather similar to our original EMD problem, and the only difference is that we have an inequality constraint instead of equality constraint. We can show that this problem also admits a strong dual given by :

$$EMD_{simple,dual} := \sup_f \int f(x)p(x)dx \tag{2.32}$$
$$\text{subject to} \quad f(x) - f(y) \leq d(x,y) \quad \text{and} \quad f(x) \leq 0$$

As before, we can easily convert the objective function and the Hölder continuity constraint into the wavelet domain. However, the extra constraint $f(x) \leq 0$ poses a serious problem. It does not have any direct wavelet domain conversion. Although there are indirect methods of ensuring negativity in the wavelet domain (for example, using the wavelet representation of convolution operators), they will not be able to give us a simple linear time algorithm. Now we will look at a different partial EMD formulation that allows us to continue using our current simple linear time algorithm.

2.4.3 Hanin's partial EMD formulation

Hanin [Hanin, 1997] proposed a different extension to the Kantorovich–Rubinstein metric for partial histograms. Hanin's extension retains almost all the properties of the original KR metric. Although it is defined for any metric cost function, we will concentrate on the metric cost

$d(x, y) := \|x - y\|^s, \quad 0 < s \leq 1$. It is defined as

$$\mu_s(p) := \inf_{p_0 \in M_0(S)} \{\dot{\mu}_s(p_0) + \mathrm{Var}(p - p_0)\} \tag{2.33}$$

Here $\mathrm{Var}(p) := \int |p(x)| dx$ is the total variation or L_1 norm. Note that the term total variation norm has different meanings in functional analysis and probability theory. Here we are using the probability theory meaning of the term. We can get Hanin's extension by setting the waste function $w(x) = 1$ in the KR extension. This is again a norm provided the cost function is a metric. If $\int p(x)dx = 0$ and D is the diameter of the support of p,

$$\mu_s(p) \leq \dot{\mu}_s(p) \leq \frac{1}{2} \max\{D, 2\} \mu_s(p) \tag{2.34}$$

So, Hanin's extension is in general equivalent to the KR metric. They are identical if $D \leq 2$. In fact, Hanin's extension behaves as if the distance metric was saturated at the value 2. The total variation cost of wasting positive and negative histogram masses of size δp would be $2\delta p$ while the transportation cost would be $d\delta p$. So, it is cheaper to waste histogram mass than move it a distance greater than 2 units. We can make sure that this does not happen and make Hanin's extension identical to the KR metric by scaling the domain to a diameter 2 before computing EMD.

For our purposes, the most important property of Hanin's extension

is that it preserves KR duality (2.10) in essentially the same form.

$$\left|\int f(x)p(x)dx\right| \leq \left|\int f(x)p_0(x)dx\right| + \left|\int f(x)(p(x)-p_0(x))dx\right| \tag{2.35a}$$

$$\leq C_H(f)\dot{\mu}_s(p_0) + \max_x |f(x)| \int |p(x)-p_0(x)|dx \tag{2.35b}$$

$$\leq ||f||_{C^s}\left[\dot{\mu}_s(p_0) + \mathrm{Var}(p-p_0)\right] \tag{2.35c}$$

Using the definition (2.33),

$$|\langle f,p\rangle| \leq ||f||_{C^s}\,\mu_s(p) \tag{2.35d}$$

Hanin [Hanin, 1997] also shows that for any p, there exists an f such that equality is attained. This is identical to the original KR duality except that the potential function f now belongs to the corresponding *inhomogeneous Hölder space*, i.e. constants are now important. We can rephrase this duality relation as

$$\mu_s(p) = \sup_f \int f(x)p(x)dx \quad \text{subject to} \quad C_H(f) \leq 1 \quad \text{and} \quad \max|f(x)| \leq 1 \tag{2.36}$$

2.4.4 WEMD for partial histograms

Hanin's extension clears our path for constructing a wavelet domain approximation. In fact, it is clear that both lemma (1) and theorem (2) are still valid. The WEMD approximation is still given by (2.17) as

$$\hat{\mu}_s = C_0 \sum_k |p_k| + C_1 \sum_\lambda 2^{-j(s+n/2)}|p_\lambda| \tag{2.37}$$

and the same error bounds (2.20) hold. Very roughly, the ratio $\frac{C_0}{C_1}$ determines the relative weight given to the extra histogram mass in p. A higher ratio will give more weight to the total variation part of the norm. Again, we set $C_0 = 0$ since this gives us the tightest error bounds and makes the metric most similar to Rubner's version. Since the domain is finite, addition of a constant affects the boundary wavelets. So even with $C_0 = 0$, we will still have some component of the total variation norm through the wavelet coefficients.

Domain scaling : To ensure that the total variation part of the norm is not activated, we can scale the domain so that its diameter is less than 2. This will scale the computed EMD by the same amount. We will restrict the scale to a power of 2, since otherwise the different wavelet coefficients will get mixed together. In this case, domain scaling simply scales the whole formula by a constant and so can be ignored.

Our scale factor is $2^{-J_0} = 2^{\lceil -\log(D/2) \rceil}$. To preserve signal energy, each wavelet coefficient magnitude will get scaled by $2^{-nJ_0/2}$, while its scale will get changed to $j \leftarrow j + J_0$. Multiplying by the factor 2^{J_0} will give us the EMD for the unscaled domain. The net scale factor is thus :

$$K(J_0) = 2^{-nJ_0/2} \times 2^{-J_0(s+n/2)} \times 2^{J_0} = 2^{-J_0(n+s-1)} \tag{2.38}$$

This scale factor applies to both the detail and approximation parts of the formula. The effect of change in scale is not apparent on the approximation part since the scale of the approximation coefficients is assumed to

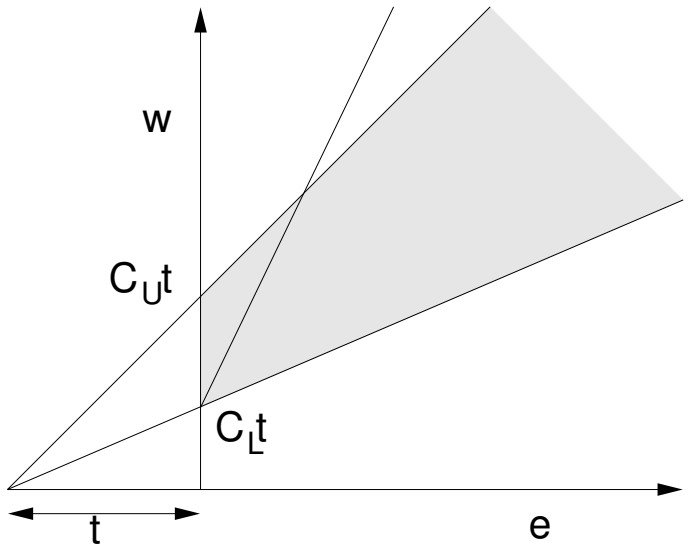


Figure 2.3: Constructing an equivalent measure for Rubner's EMD extension is not possible because we can't enclose the shaded region in a 2D cone.

be zero in the WEMD formula (2.37).

Approximating Rubner's extension : Hanin's extension differs from Rubner's extension only in the presence of the easy to calculate total variation norm term. Although this indicates that we should be able to compute a measure equivalent to Rubner's EMD, this is not possible. Let us see why.

Let $p_e := p - p_0$ be the extra mass. Since p_1 has more mass than p_2 and all the negative mass has been matched up, we have $p_e(x) \geq 0$ for all $x \in S$. In this case,

$$t(p) := \mathrm{Var}(p_e) := \int p(x)dx = \sum_k p_k, \tag{2.39}$$

since the approximation coefficients also add up to the net extra mass. Let e be Rubner's EMD and w the wavelet EMD for Hanin's extension. Then the metric w is equivalent to Hanin's extension $t + e$, so we have

$$C_U \leq \frac{w}{t+e} \leq C_L \tag{2.40}$$

The possible values of w according to this relation are represented by the 2D cone in figure (2.3), for a given value of t. The slopes of its sides are the two bounds C_L and C_U and the intercepts on the w axis are $C_L t$ and $C_U t$. To construct an equivalent measure for e, we need a similar 2D cone with the vertex on the w axis, since w must have a unique value of 0 when $e = 0$. This new cone should include the shaded area for $e \geq 0$ and cannot have sides parallel to an axis. Clearly, it is not possible to construct such a cone since it will always miss part of the shaded region. Further, we cannot transform the shaded area into a cone without knowing the e coordinate of its points. So we cannot construct a measure equivalent to Rubner's EMD using this technique.

2.4.5 Best partial match

The *best partial match* version of EMD [Holmes et al., 2002a], [Holmes et al., 2002b] takes the view that since the larger histogram is constructed from more samples, only a fraction of it should be matched to the smaller histogram. The remaining mass is ignored. This can be useful for matching parts of images to full images or images at difference resolutions.

Our difference density is now $p(\alpha) := \alpha p_1 - p_2$, and the best partial match distance is defined as :

$$\text{EMD}_{\text{BPM}} := \min_{\alpha} \mu_s(\alpha p_1 - p_2) \tag{2.41}$$

Correspondingly, we can now define the wavelet EMD approximation to the best partial match distance as

$$\hat{\mu}_s(p_1, p_2) := \min_{\alpha} \mu_s(\alpha p_1 - p_2) \tag{2.42}$$

We can take advantage of the fact that WEMD is the l_1 norm of the weighted wavelet coefficients of the two histograms to construct a fast $O(N \log N)$ algorithm to compute the wavelet best partial match (WBPM). Let $\mathbf{u}$ and $\mathbf{v}$ be the vectors of weighted wavelet coefficients of p_1 and p_2 respectively, i.e.

$$\mathbf{u} = [C_0 p_{1k}, \quad C_1 2^{-j(s+n/2)} p_{1\lambda}] \tag{2.43a}$$

$$\mathbf{v} = [C_0 p_{2k}, \quad C_1 2^{-j(s+n/2)} p_{2\lambda}] \tag{2.43b}$$

So the WEMD between p_1 and p_2 is given by $\hat{\mu}_s = \|\mathbf{u} - \mathbf{v}\|_1$ while the wavelet best partial match is given by

$$\hat{\mu}_s(p_1, p_2) = \min_{\alpha \in [0,1]} \|\alpha \mathbf{u} - \mathbf{v}\|_1 = \min_{\alpha \in [0,1]} \sum_i |\alpha u_i - v_i| \tag{2.44}$$

In this optimization problem, $\alpha \mathbf{u}$ represents a line through the origin and $\mathbf{v}$ is a point. So this is equivalent to finding the point on a line closest in the l_1 distance to a given point in $\mathbb{R}^N$, where N is the length of the vectors U and v.

This is a convex minimization problem with a piecewise linear objective function. The pieces are joined at the points where one of $|\alpha u_i - v_i|$ changes sign, i.e. at $\alpha_i = \frac{v_i}{u_i}$ $(u_i \neq 0)$. Since our objective function "turns" at only these points, the minimum must occur at one (or two consecutive) of these points. Finding the minimum is then a simple matter of evaluating the objective at each point and selecting the minimum.

Lemma 2. *Given a line through the origin* $\alpha\mathbf{u}$ *and a point* $\mathbf{v}$*, where* $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$*, the point on the line closest to it in the* l_1 *norm is one of* $\alpha_i\mathbf{u}$*,* α_i *defined as above. If two such points are nearest, then they correspond to consecutive* α_i *and all points in between are also equidistant.*

To prove this lemma rigorously, we will need the concepts of *subderivative* and *subdifferential* from non-smooth convex optimization [Bertsekas, 2003]. A subderivative of a function $f : \mathbb{R} \to \mathbb{R}$ at the point x_0 where $f(x_0)$ is finite is any number ξ such that $f(x) - f(x_0) \leq \xi(x - x_0)$ for all $x \in \mathbb{R}$. The set of all such ξ at x_0 is called the subdifferential $\partial f(x_0)$. If f is differentiable at x_0, then the subdifferential is a singleton set with the derivative as its only element. If the function is not differentiable because its left and right hand derivatives exist but are different, then the subdifferential consists of all numbers between and including the two. The figure (2.4.5) shows the subdifferential of the function $|x|$. *Subgradient* is the generalization of this concept in higher dimensions.

Subdifferentials are important in convex optimization because they

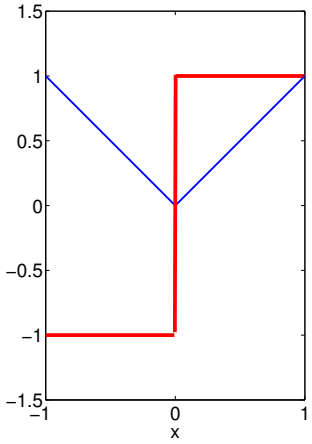


Figure 2.4: The function $|x|$ and its subdifferential. 0 belongs to the subdifferential at the minimum ($x = 0$).

allow the extension of the Karush-Kuhn-Tucker theorem to non-smooth functions. An unconstrained version that suffices for our purpose is as follows :

Theorem 3. *Let $f : \mathbb{R} \to \mathbb{R}$ be a convex function and let $M \subset \mathbb{R}$ be a nonempty convex set. Then the optimization problem $\inf_{x \in M} f(x)$ has a solution $\bar{x}$ if and only if there exists a subderivative $\bar{\xi} \in \partial f(\bar{x})$ such that $\bar{\xi}(x - \bar{x}) \geq 0$ for all $x \in M$.*

If $\bar{x}$ is in the interior of M, this can only happen if $\bar{\xi} = 0$. Now we are ready to prove Lemma 2.

Proof of Lemma 2. Let $f(\alpha) := \|\alpha \mathbf{u} - \mathbf{v}\|_1$. In the light of the above theorem, it is clear that the l_1 distance between the point $\mathbf{v}$ and the line $\alpha \mathbf{u}$ will be minimized when $0 \in \partial f(\bar{\alpha})$. $\partial f(\alpha)$ is non-decreasing piecewise constant

where $f(\alpha)$ is differentiable. If none of these constant values is zero, $0 \in \partial f(\bar{\alpha})$ where $f(\bar{\alpha})$ is not differentiable, i.e. at $\bar{\alpha} = \frac{v_i}{u_i}$ for some i. Thus $f(\alpha)$ is minimized at a point where it is not differentiable. On the other hand, if one of these constant values is zero, then $f(\alpha)$ is minimized on the whole interval $[\alpha_i, \alpha_k]$ for some consecutive α_i, α_k. □

For our problem, we can restrict the search space to $\alpha \in [0, 1]$. Computing the l_1 distance at a point takes $O(N)$ time, so computing the distance at all N α_i will require $O(N^2)$ time. Instead, we will first sort the α_i in $O(N \log N)$ time. Since the objective is convex, we can use a modified binary search to find the minimum. Each time the interval is split in half, discard the subinterval where the distance increases away from the point of splitting. If the distance does not change at the point of splitting, we have found our minimizing interval. Finally, this will yield either the point or interval with minimum distance. This will require $O(\log N)$ distance evaluations for a total time of $O(N \log N)$. The full algorithm thus runs in $O(N \log N)$ time. Figure (2.5) describes the overall algorithm.

2.5 Improving WEMD consistency

We can analyze sources of error in the WEMD algorithm by looking at cases that have the same EMD but different WEMD. The simplest transformation that preserves EMD but changes WEMD is shifting. EMD does not depend on the absolute position of the difference histogram and shift-

Figure 2.5: Wavelet best partial match EMD

Data: Histograms p_1 and p_2

Result: Best partial match distance

$$\mu_s(p_1, p_2) := \min_\alpha f(\alpha) = \min_\alpha \|\alpha \mathbf{u} - \mathbf{v}\|_1$$

1 Compute weighted wavelet transforms :

$$\mathbf{u} = [C_0 p_{1k}, \quad C_1 2^{-j(s+n/2)} p_{1\lambda}]$$

$$\mathbf{v} = [C_0 p_{2k}, \quad C_1 2^{-j(s+n/2)} p_{2\lambda}]$$

2 Compute $\alpha_i = \frac{v_i}{u_i}$. Ignore α_i if $u_i = 0$ or $\alpha_i < 0$ or $\alpha_i > 1$.

3 Add {0,1} to the set of α_i

4 Sort α_i

`/* Binary search */`

5 $i \leftarrow 0$, $j \leftarrow N$

6 **while** $\mathtt{f}(\alpha_i) \neq \mathtt{f}(\alpha_j)$ **do**

7 $\quad k \leftarrow \lfloor \frac{i+j}{2} \rfloor$

8 $\quad$ **if** $f(\alpha_{k+1}) == f(\alpha_k)$ **then**

9 $\quad\quad$ **return** $\mathtt{f}(\alpha_k)$

10 $\quad$ **end**

11 $\quad$ **if** $f(\alpha_{k+1}) < f(\alpha_k)$ **then**

12 $\quad\quad$ $i \leftarrow k + 1$

13 $\quad$ **else**

14 $\quad\quad$ $j \leftarrow k$

15 $\quad$ **end**

16 **end**

17 **return** $\mathtt{f}(\alpha_i)$

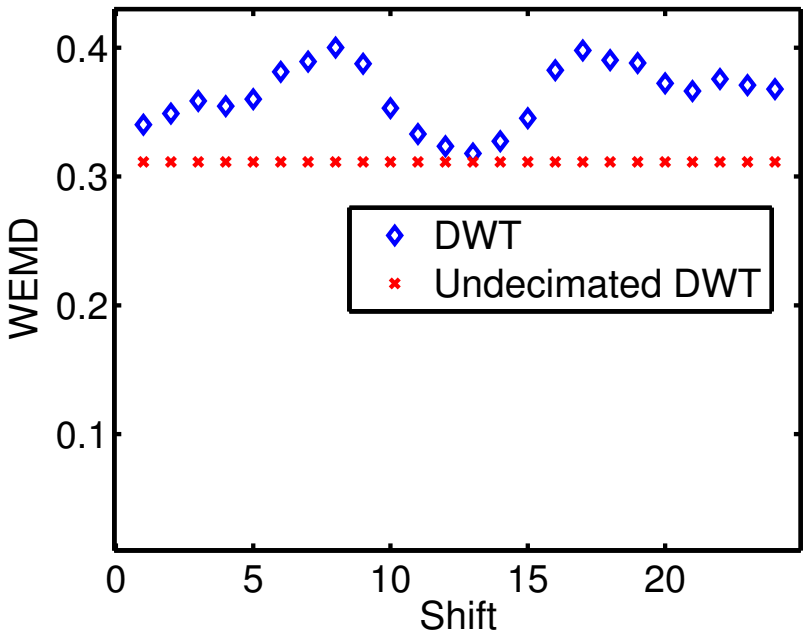


Figure 2.6: WEMD for a pair of delta functions (EMD = distance = 8 units) changes as the pair is shifted. This can be avoided by using the undecimated wavelet transform.

ing it does not have any effect. Unlike the discrete Fourier transform (DFT), where signal translations do not affect the coefficient magnitudes, discrete wavelet transform (DWT) coefficients change in a hard to predict way. The net result is that the weighted L_1 norm (WEMD) changes too.

An obvious way of dealing with these errors is to compute the WEMD for all possible shifts and take the average. This will not change the error bounds, i.e. the maximum normalized error, but will decrease the average error. The *cycle spinning* [Coifman and Donoho, 1995] procedure for signal denoising using wavelet coefficient thresholding takes a similar

approach. The wavelet transform is computed for all shifts of the signal, de-noised separately and shifted back. The final result is the mean over all shifts. This procedure reduces *ringing* artifacts produced during de-noising.

Computing the wavelet transform over all shifts is not actually necessary since some of them are redundant. For example, at the finest scale J, we have wavelet coefficients for all even shifts, so we only need the extra coefficients for odd shifts. We can retain all coefficients by not downsampling the data after filtering. At the next scale, we need to upsample the filters by 2 instead. By repeating this at each scale we get the *undecimated* or *stationary* wavelet transform [Coifman and Donoho, 1995]. This is *covariant to translations*, i.e. a shift in the signal leads to the same shift in the wavelet coefficients. At each scale $O(N)$ operations are required and the transform can be calculated to $O(\log N)$ scales. The total complexity as well as the number of coefficients produced are $O(N \log N)$. This can be a disadvantage, especially for large or high dimensional histograms. As we will describe later, we can use thresholding to remove coefficients with small magnitudes.

At scale j, we need to take the mean of $2^{n(J-j+1)}$ different shifts. So

the adjusted formula for undecimated or stationary WEMD is :

$$d(p)_{uwemd} := \sum_{\lambda} |p_\lambda| 2^{-j(s+n/2)} 2^{-n(J-j+1)} \tag{2.45}$$

$$= \sum_{\lambda} |p_\lambda| 2^{-j(s-n/2)-n(J+1)} \tag{2.46}$$

Note that the coarsest scale is 0 and finest scale is J. Figure (2.6) shows the variation in WEMD when a pair of delta functions are shifted. WEMD computed using the undecimated wavelet transform is constant.

2.6 Experiments

First, in section (2.6.1), we will discuss some implementation issues that affect the accuracy and other aspects of wavelet EMD. In section (2.6.2), we will describe how to choose appropriate wavelets. Finally, in section (2.6.3), We will describe experiments that demonstrate that the wavelet EMD behaves very similar to EMD, but can be computed much faster.

2.6.1 Some implementation notes

For applications that store computed histogram descriptors, we split the wavelet EMD computation into two parts. First, the histogram descriptor is converted into the wavelet domain and its coefficients are scaled according to equation (2.2). The wavelet EMD distance between two descriptors is now the L_1 distance between these coefficients. We should note the following points while computing wavelet EMD :

1. Initialization : The standard Mallat filter bank algorithm ([Mallat, 1998] section 7.3.1) for computing the wavelet transform starts with fine level wavelet coefficients as input. We can use signal values as input if we want to reconstruct the signal again, as in compression or denoising. This does not work if we want to use wavelet coefficients to represent signal properties like Hölder continuity. We can approximate fine scale wavelet coefficients with signal values if we use *coiflets* ([Mallat, 1998] section 7.2.3). Unfortunately, this is not accurate enough for our application. So, we use the wavelet transform initialization method (algorithm 2) of Zhang, Tian and Peng [Zhang et al., 1996]. We assume that the histogram bin values are obtained from a block sampler. The initialization consists of projecting the data onto the subspace of the finest level wavelet coefficients.

2. Periodic and non-periodic histograms : For data like distance and intensity values, there are no samples outside the histogram limits and we use zero padding extension while computing the wavelet transform. Since angles are measured modulo 2π, angle dimensions are extended periodically. For example, SIFT descriptors are 3D histograms of gradient orientation with respect to location around the feature point. So, we should use periodic extension along the gradient orientation dimension and zero padding along the location dimensions.

3. Wavelet transform size : Zero padding increases the size of the wavelet transform. For each decomposition level, the histogram is padded with a vector of zeros about as long as the wavelet filter length. This is significant for multi-dimensional histograms that only have a few bins along each dimension. However, most of these coefficients are close to zero because the wavelet transform is a sparse representation. This is more of a concern for sparse high dimensional histograms since their wavelet transform can be non-sparse.

We can store the coefficients compactly as a sparse vector if we set small coefficients to zero. After weighting the coefficients, we keep the largest coefficients that contribute 95% to the total L_1 norm. The remaining are set to zero. The coefficients are then stacked to form a 1D sparse vector: the final descriptor representation. Descriptor comparison takes time linear in the number of non–zero coefficients. Although there may be about 1–5 times as many elements as in the original histogram, depending on its size and dimensionality, the required time is similar to that for χ^2 or Euclidean distance on similarly enlarged histograms.

4. Wavelet Transform algorithm : Wavelet transform computation time increases exponentially ($O(2^n)$) with dimension using Mallat's fast wavelet transform (FWT) algorithm. On the other hand, the computation time for Swelden's lifting wavelet transform (LWT) algorithm does not depend significantly on the dimensionality. LWT reduces unnecessary computation

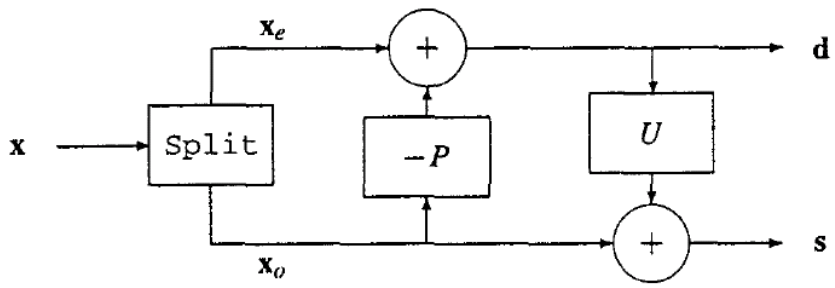


Figure 2.7: Split, predict and update operations that make up one step of the lifting wavelet transform. Additional predict and update steps may be used. The outputs are the detail (d) and approximation (s) coefficients of one coarser scale. Figure from [Daubechies and Sweldens, 1998]

by subsampling before filtering, unlike FWT. For long wavelet filters, LWT requires half as much time as FWT for each dimension, and hence the $O(2^n)$ factor is absent. So LWT is a far better choice for high dimensional histograms.

The lifting algorithm splits the data into even and odd components. A *predict* or *dual lifting* step ($-P$) adds a filtered version of the odd component to the even in an attempt to predict its values. The complementary step U is the *update* or *primal lifting* step and tries to do the opposite. We can choose appropriate filters so that the even components are transformed into the detail coefficients (d) and the odd components into the approximation coefficients (s). This can be repeated to compute higher level coefficients. Figure (2.7) explains the process. For details, please see [Daubechies and Sweldens, 1998], from which the figure is taken.

At each scale, this first subsamples the data and then filters it. This crucial exchange of the order of these two operations, compared to Mallat's fast wavelet transform algorithm, reduces the computation by about half for long filters for each dimension. As a result the number of computation steps does not depend significantly on the dimension.

Next we will look at how to choose wavelets that approximate EMD well.

2.6.2 Which wavelets ?

The conditions of theorem (2) put some restrictions on the wavelets for which this works. We need wavelets with at least one derivative. This rules out the simple Haar wavelet.

We can try choosing the best possible wavelets by computing the bounds ratio C_U/C_L for $C_0 = 0, C_1 = 1$. Table (2.2) lists maximum error estimates (C_U/C_L) for some common wavelets in 1D. These estimates (see 2.A) are computed through combinatorial optimization and are hard to compute for higher dimensions. Without explicit calculation, we cannot say how the bounds will change for a wavelet as the dimension increases. The estimate formulas do indicate that wavelets with small support and fast decay will have a high C_L. C_U will be low if the wavelet has a small absolute value maximum.

In higher dimensions, it is easier to choose wavelets empirically. We

Daubechies	C_U/C_L	Daub. symmetric	C_U/C_L
db3	6.33	sym3	6.33
db4	7.29	sym4	4.64
db5	9.92	sym5	6.01
db6	12.59	sym6	5.58
Coiflets	C_U/C_L	Ojanen	C_U/C_L
coif1	4.38	oj8	7.46
coif2	4.75	oj10	10.56
coif3	5.85	oj12	13.79

Table 2.2: Theoretical (loose) estimates for maximum error for various 1D wavelets. Ojanen wavelets have maximum smoothness for a given filter length. Coiflets have low error despite large support.

measured the error of wavelet EMD with respect to actual EMD for a set of 100 random 16×16 histogram pairs. Since uniform random histogram pairs tend to have EMD concentrated in a small range, we instead generated only one histogram randomly. The second histogram was obtained by changing this at random locations by random amounts. The number of locations as well as maximum allowed change at a location was gradually increased. These random histogram pairs have well distributed EMDs. Wavelet EMD was scaled to make its mean ratio with EMD 1. Table (2.3) shows the normalized RMS error and the observed bounds ratio C_U/C_L. The bounds ratio is the maximum of all the ratios WEMD/EMD and EMD/WEMD, while the normalized RMS error is the RMS deviation of the ratio WEMD/EMD from 1. The table also notes the time needed to compute wavelet EMD in MATLAB R2007a on an Intel Xeon HT 3GHz PC. However, the wavelet transform that is the most time consuming operation, was computed using a C++ lifting wavelet transform implementation. See 2.C for details. We observed that Coiflets of order 3 and symmetric Daubechies wavelets of order 5 produced good results. We use order 3 coiflets in our experiments.

Next we repeated the same experiment for histograms of unequal mass using Hanin's EMD extension as the reference. When the smaller mass histogram is normalized, this is Rubner's EMD plus the excess mass in the larger histogram. Its exact value can be computed using the linear program for Rubner's EMD. We used the same method for generat-

Wavelet	Normalized RMS error	Bounds ratio C_U/C_L	Time (ms)
db3	16%	1.91	2.8
db4	20%	2.45	4.5
db5	17%	1.98	5.6
db6	18%	1.93	6.2
sym3	16%	1.91	2.8
sym4	17%	2.18	3.6
sym5	**13%**	**1.50**	**5.4**
sym6	16%	2.00	6.0
coif1	16%	1.88	3.0
coif2	15%	1.85	8.3
coif3	**14%**	**1.87**	**11.0**
oj8	20%	2.44	3.7
oj10	18%	2.07	5.2
oj12	17%	1.82	8.1

Table 2.3: EMD approximation error and times for random 16×16 histograms for various wavelets. Computing EMD takes **181ms on average.**

ing random histograms as before and then scaled the changed locations of the second histogram by a random value to increase its mass. The same algorithm as before computes the wavelet EMD approximation. Table (2.4) shows the normalized RMS error and the bounds ratio obtained. In general, the results are more accurate in this case. Coiflets of order 1 and 2 and symmetric Daubechies wavelets of order 5 perform better than the others, although the difference between various wavelets is less significant. The computation times are the same as before.

2.6.3 Image retrieval: colour histograms

We tested wavelet EMD on content based image retrieval using colour histograms. We used the SIMPLIcity test database [Wang et al., 2001] that consists of 10 image classes with 100 images each. Figure (2.8) shows a sample image of each class. We will show that wavelet EMD provides a better approximation to EMD than other EMD approximation methods in terms of distance values as well as performance for colour histograms. We computed $16 \times 16 \times 16$ colour histograms in Lab colour space since Euclidean (ground) distances in this colour space are proportional to perceived colour differences. The histograms were clustered into 64 clusters each before computing EMD, but not for computing approximations.

The scatter plots in figure (2.9) compare the wavelet EMD approximation with that of Indyk and Thaper [Indyk and Thaper, 2003] for dis-

Wavelet	Normalized RMS error	Bounds ratio C_U/C_L
db3	14%	1.61
db4	14%	1.71
db5	14%	1.86
db6	16%	1.61
sym3	14%	1.61
sym4	15%	1.72
sym5	**13%**	**1.69**
sym6	14%	1.61
coif1	**15%**	**1.48**
coif2	**13%**	**1.50**
coif3	14%	1.54
oj8	14%	1.69
oj10	14%	1.94
oj12	16%	1.64

Table 2.4: Hanin EMD approximation error for random unequal mass 16×16 histograms for various wavelets

People Beach Architecture Bus Dinosaur

Elephant Rose Horse Mountain Food

Figure 2.8: Sample images of the different classes in from the SIMPLIcity dataset.

Method	Bounds ratio	Normalized RMS error	Preproc. time	Compare time
EMD	–	–	0.92 s	63 ms
Wavelet EMD	7.03	18%	0.66 s	0.11 ms
Indyk-Thaper	11.00	43%	0.51 s	22 ms

Table 2.5: Error and time requirements for 16x16x16 colour histograms. Preprocessing time includes colour space conversion, binning, clustering (EMD only) and weighted wavelet transform (WEMD). Indyk-Thaper random embedding is repeated 5 times.

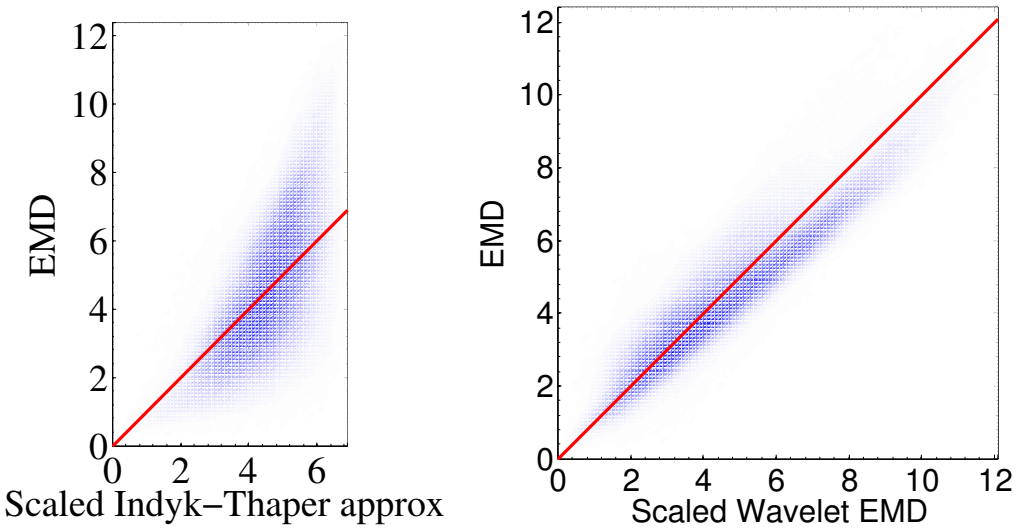


Figure 2.9: EMD approximations with Wavelet EMD using order 3 Coiflets is better than with Indyk and Thaper's [Indyk and Thaper, 2003] method. The red (dark) line indicates points of zero error.

tances computed between these colour histograms. Both approximations are scaled to have a mean ratio with EMD of 1. The plot indicates that Wavelet EMD distances correlate better with EMD than Indyk and Thaper. Note that EMD and its approximations have a maximum value depending on the histogram size. The Indyk-Thaper scatter plot appears cut–off because its greater spread causes it to reach this limit faster. Table (2.5) shows the approximation errors and time requirements for EMD, wavelet EMD and Indyk and Thaper's method. Wavelet EMD needs slightly more preprocessing time than Indyk-Thaper to compute the wavelet transform. The actual comparison is very fast since it is simply an l_1 norm (Manhattan distance). The Indyk-Thaper algorithm is implemented in Matlab,

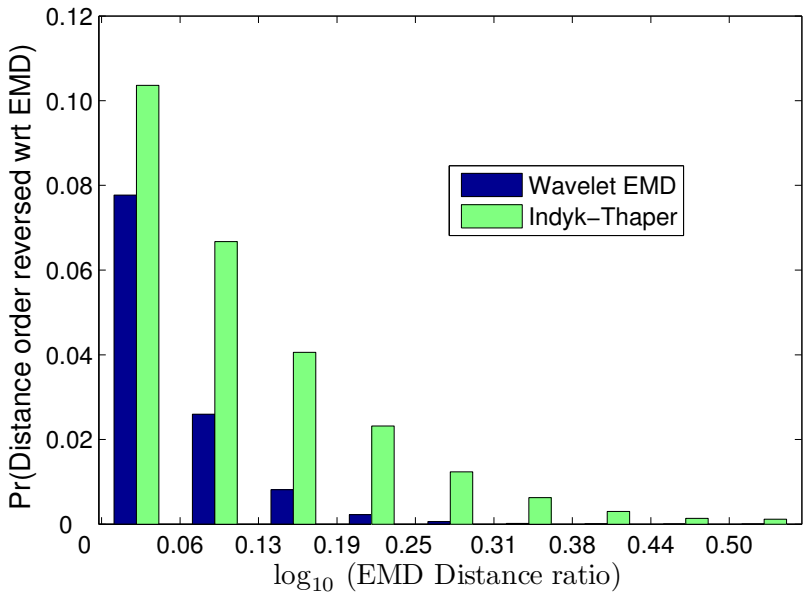


Figure 2.10: Wavelet EMD is less likely to disagree with EMD about ordering of histogram distances than Indyk-Thaper.

while WEMD is implemented in C++. Still, it is clear that WEMD does not add much time to the preprocessing stage while making the comparison extremely fast.

Another method to measure approximation error, in the context of feature matching, is to measure the probability of distance order reversal, i.e. the probability that histogram p_1 is closer to histogram p_2 than to histogram p_3 according to EMD, but not according to an approximation. We expect this probability to decrease as p_3 moves farther away from p_1, compared to p_2, i.e. the ratio $EMD(p_1, p_3)/EMD(p_1, p_2)$ increases. Figure (2.10) shows that this probability starts lower and falls off faster for wavelet EMD than for Indyk and Thaper's approximation. We do not

include EMD–L_1 in these comparisons because it uses a different ground distance.

Figure (2.11) shows ROC curves for EMD and its different approximation methods obtained from leave one out image retrieval experiments on this dataset. Wavelet EMD and EMD have almost the same performance, and this is better than EMD–L_1 and Indyk and Thaper's method.

2.7 Conclusion and future work

We have introduced a new method to approximate the earth mover's distance between two histograms using weighted wavelet transform coefficients of the difference histogram. We provide theoretical bounds to the maximum approximation error. Our experiments with colour histograms demonstrate that the wavelet EMD approximation preserves the performance of EMD while significantly reducing computation time.

We want to use the wavelet EMD approximation for different applications that need to compare probability distributions, and particularly for histogram descriptors for keypoint matching in images. We would also like to explore the use of different ground distances (different powers s). But first, we should keep this limitation for high dimensional sparse histograms in mind. Computing the wavelet transform for high dimensional sparse histograms needs algorithms whose complexity depends on the number of non-zeros rather than the total histogram size. For ex-

ample, since each data point in a dataset of size N only effects $O(\log N)$ wavelet coefficients, there is a trivial algorithm to compute the wavelet transform of a sparse dataset with K non-zeros in $O(K \log N)$ time. Further, the wavelet transform of sparse arrays can be non-sparse, so we should expect a significant increase in storage.

We can also use dimensionality reduction techniques like PCA as long as the pairwise Euclidean distances between the nonzero points are preserved. This could work as follows :

1. Compute the high dimensional difference histogram $p := p_1 - p_2$. Find the coordinates of the non-zero entries of this sparse data.

2. Perform (quasi)isometric dimensionality reduction on this set of coordinates. We can use weighted PCA if the data is known to lie close to a lower dimensional subspace. In weighted PCA, each data point is weighted with its histogram mass when computing the mean and covariance matrix. This stage will introduce some additional error into the EMD computation. The relative magnitude of the retained eigenvalues should give us an idea of this error.

3. Compute the wavelet EMD on the low dimensional data.

2.7.1 Image registration

Image registration is a difficult problem that involves aligning two images so that their structures match. A common approach to image registration

is to formulate an objective function that reflects a goodness of match between one image and the deformed second image. This objective function is then optimized over different possible deformations using a gradient descent procedure. The mutual information between intensity distributions of the two images is a good objective function for registering images of different modalities (like MRI and CAT scan images). Mutual information works well because it tends to model the dependence of the two intensity distributions. [Chefd'hotel and Bousquet, 2007] replace mutual information with the EMD between the joint distribution of the two images and the product of their distributions. This is another measure of the dependence of the distributions. We can directly use wavelet EMD here to speedup image registration.

For images of the same modality, i.e. both optical images or both MRI images, registration can be performed by treating some suitable function of the image as a probability distribution. Solving the continuous EMD, i.e. the Monge-Kantorovich (MK) problem, gives us the optimal map that transforms the first distribution into the second. This also gives the deformation function between the two images. This approach has been used by [Haker et al., 2004]. They solve the MK problem as a variational problem with an iterative method. We may be able to solve this directly using the wavelet EMD technique. It should be possible to compute the optimal deformation map from the potential function f (see theorem 2) that is computed in the wavelet domain. The important work

of [Gangbo and McCann, 1996] where they relate the potential function with the optimal transportation map is a suitable starting point for this research.

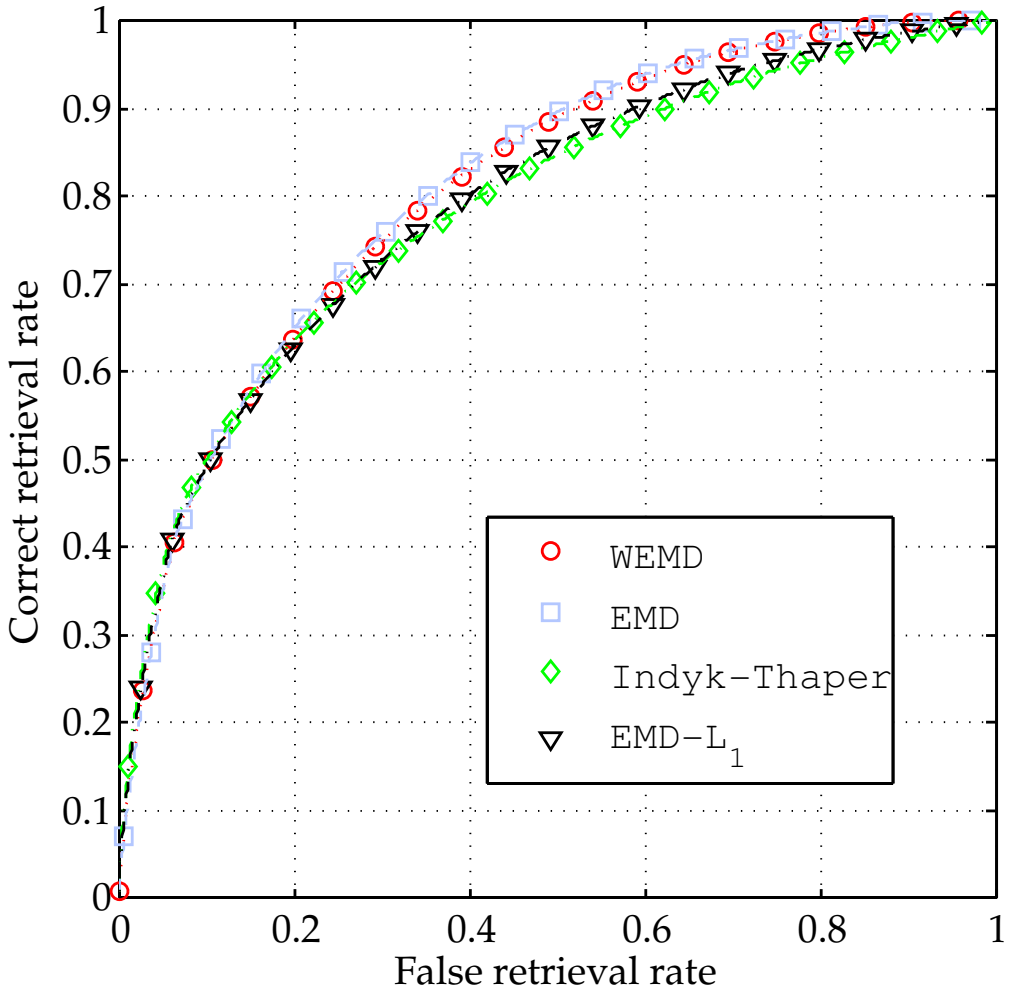


Figure 2.11: Colour histograms for content based image retrieval: wavelet EMD performance compared to other EMD methods

Appendix

2.A Proof of Lemma (1)

Parts of this proof are adapted from Meyer ([Meyer, 1992] section 6.4). We will start with the first inequality $a_{12}(\psi; s)C_1 \leq C$ in (2.16). The proof of this inequality corresponds to the proof of the *only if* part of theorem (1). For all functions $f \in C^s(\mathbb{R}^n), 0 < s \leq 1$ with the seminorm $C_H(f)$, we will compute bounds on their wavelet series coefficients. We will omit the dependence of C_H on f to simplify notation. Suppose that the wavelet coefficient bounds are actually attained. Using the definition of C_H, we can bound the values of $f(x)$ as :

$$|f(x) - f(k+r)| \leq C_H \|x - k - r\|^s \text{ for any } r \in \mathbb{R}^n, k \in \mathbb{Z}^n$$

Since the bounds are attained, we have

$$\begin{aligned}
C_0 &= \sup_k |f_k| = \sup_k \left| \int f(x)\phi(x-k)dx \right| \\
&= \sup_k \left| f(k+r) + \int (f(x) - f(k+r))\phi(x-k)dx \right| \qquad \left(\text{since} \int \phi(x-k)dx = 1 \right) \\
&\leq \sup_k |f(k+r)| + \int |f(x) - f(k+r)|\,|\phi(x-k)|\,dx \\
&\leq \sup_k |f(k+r)| + \int C_H \|x - k - r\|^s\,|\phi(x-k)|\,dx \\
&\leq ||f||_\infty + C_H \int \|x - r\|^s |\phi(x)| dx
\end{aligned}$$

Since this is true for all $r \in \mathbb{R}^n$,

$$C_0 \leq ||f||_\infty + C_H \inf_r \int \|x - r\|^s |\phi(x)| dx \tag{2.47}$$

If we define

$$a_{11}(\psi; s) := \frac{1}{\inf_r \int \|x - r\|^s |\phi(x)| dx}, \tag{2.48}$$

we can write this as

$$C_H \geq a_{11}(C_0 - ||f||_\infty) \tag{2.49}$$

Note that this constant depends only on the wavelet and s.

To compute a bound on C_1, we will first bound f_λ using the definition of C_H.

$$f(y) \leq f(2^{-j}(k+r)) + C_H \|y - 2^{-j}(k+r)\|^s$$

$$\begin{aligned}
|f_\lambda| &= \left| \int f(y)\psi_\lambda(y)dy \right| \\
&= \left| \int (f(y) - f(2^{-j}(k+r)))\psi_\lambda(y)dy \right| \\
&\leq \int \left| f(y) - f(2^{-j}(k+r)) \right| |\psi_\lambda(y)| \, dy \\
&\leq \int C_H \|y - 2^{-j}(k+r)\|^s \left| 2^{nj/2}\psi^\epsilon(2^j y - k) \right| dy
\end{aligned}$$

(using the n dimensional change of variables $x = 2^j y - k$, so $dx = 2^{nj} dy$)

$$\begin{aligned}
&= C_H \int \|x - r\|^s 2^{-js} 2^{-nj/2} |\psi^\epsilon(x)| \, dx \\
&= C_H 2^{-j(s+n/2)} \int \|x - r\|^s |\psi^\epsilon(x)| \, dx
\end{aligned}$$

$$\begin{aligned}
\text{So } C_1 &= \sup_\lambda 2^{j(s+n/2)} |f_\lambda| \\
&\leq \sup_{j,\epsilon} 2^{j(s+n/2)} C_H 2^{-j(s+n/2)} \int \|x - r\|^s |\psi^\epsilon(x)| \, dx
\end{aligned} \tag{2.50}$$

Since this is true for all $r \in \mathbb{R}^n$,

$$C_1 \leq C_H \max_\epsilon \inf_r \int \|x - r\|^s |\psi^\epsilon(x)| \, dx \tag{2.51}$$

If we define

$$a_{12} := \frac{1}{\max_\epsilon \inf_r \int \|x-r\|^s \, |\psi^\epsilon(x)| \, dx}, \tag{2.52}$$

we can write this as

$$C_H \geq a_{12} C_1 \tag{2.53}$$

This constant too depends only on the wavelet and s.

From equations (2.49) and (2.53), we have

$$C_H \geq \max\left\{a_{11}(C_0 - ||f||_\infty), a_{12}C_1\right\} \tag{2.54}$$

If the bounds on the wavelet coefficients of f are not attained, we can instead say that

$$C_H \leq C \text{ such that } C \geq \max\left\{a_{11}(C_0 - ||f||_\infty), a_{12}C_1\right\} \tag{2.55}$$

Since its hard to know $||f||_\infty$ beforehand, we can simply use the looser bound (2.53),

$$C_H \leq C \text{ and } C \geq a_{12}C_1 \tag{2.56}$$

This is our first inequality.

Proving the second inequality is a bit more involved. This corresponds to the proof of the *if* part of theorem (1). We need to look at the converse problem: given a function defined by a wavelet series with approximation and detail coefficients bounded by C_0 and C_1 respectively, what is the corresponding bound on C_H ?

We start with the wavelet series of f

$$f(x) = \sum_k f_k \phi(x-k) + \sum_{\lambda \in \Lambda_j, j \geq 0} f_\lambda \psi_\lambda(x)$$

and split this into a Littlewood-Paley type series as

$$f(x) = \sum_{j>=-1} f_j(x) \tag{2.57}$$

$$\text{with } f_{-1}(x) = \sum_k f_k \phi(x-k) \tag{2.58}$$

$$\text{and } f_j(x) = \sum_{\lambda \in \Lambda_j} f_\lambda \psi_\lambda(x) \qquad \text{for } j \geq 0 \tag{2.59}$$

To begin with, we will establish some properties of the functions f_j. Consider the wavelet series $\Sigma\psi(x;\eta) := \sum_{k,\epsilon} \eta_{k,\epsilon} \psi^{(\epsilon)}(x-k)$ with $-1 \leq \eta_{k,\epsilon} \leq 1$. This is a convergent series because of the fast decay properties of wavelets. So,

$$||\Sigma\psi||_\infty := \sup_{x,\eta} |\Sigma\psi(x;\eta)| \tag{2.60}$$

is finite. This quantity can be computed for wavelets with compact support using combinatorial optimization if we note that the supremum will occur at $\eta_{k,\epsilon} \in \{-1, +1\}$. Otherwise, if the supremum occurs at the point x_0 with $\Sigma\psi(x_0;\eta) > 0$ where the contribution of $\psi^\epsilon(x-k)$ is positive, we can increase the supremum by increasing $\eta_{k,\epsilon}$ up to a maximum of 1. A similar argument holds if the series sum is negative and if the contribution of $\psi^\epsilon(x-k)$ is negative.

If we have $|f_\lambda| \leq C_1 2^{-j(s+n/2)}$, then

$$|f_j(x)| \leq C_1 2^{-j(s+n/2)} 2^{nj/2} ||\Sigma\psi||_\infty \qquad \text{for all } x$$

$$\text{So, } ||f_j||_\infty \leq C_1 ||\Sigma\psi||_\infty 2^{-js} \tag{2.61}$$

With a similar argument, we get

$$||f_{-1}||_\infty \leq C_0 ||\Sigma\phi||_\infty \tag{2.62}$$

where $||\Sigma\phi||_\infty$ is defined similar to $||\Sigma\psi||_\infty$.

Now we can immediately bound $\|f\|_\infty$ as

$$\|f\|_\infty \leq C_0 \, ||\Sigma\phi||_\infty + \sum_{j\geq 0} C_1 \, ||\Sigma\psi||_\infty \, 2^{-js}$$

$$\|f\|_\infty \leq C_0 \, ||\Sigma\phi||_\infty + \frac{C_1}{1-2^{-s}} \, ||\Sigma\psi||_\infty \tag{2.63}$$

Next, we will look at the first derivatives of the functions f_j. Since the wavelets have at least one derivative, we have for first derivatives with respect to all the components x_i $(i = 1, \ldots, n)$ of x :

$$\partial_{x_i} f_{-1}(x) = \sum_k f_k \partial_{x_i} \phi(x-k) \tag{2.64}$$

$$\begin{aligned} \partial_{x_i} f_j(x) &= \sum_{\lambda\in\Lambda_j} f_\lambda \partial_{x_i} 2^{nj/2} \psi^\epsilon(2^j x - k) \\ &= \sum_{\lambda\in\Lambda_j} f_\lambda 2^{j(n/2+1)} (\partial_{x_i}\psi^\epsilon)(2^j x - k) \end{aligned} \tag{2.65}$$

Again using the fast decay properties of wavelet derivatives, we can define the following convergent series and their absolute suprema :

$$\Sigma\phi^{(i)}(x;\eta) := \sum_k \eta_k \partial_{x_i}\phi(x-k) \qquad \|\Sigma\phi^{(i)}\|_\infty := \sup_{x,\eta} |\Sigma\phi^{(i)}(x;\eta)| \tag{2.66}$$

$$\Sigma\psi^{(i)}(x;\eta) := \sum_{k,\epsilon} \eta_{k,\epsilon} \partial_{x_i}\psi^{(\epsilon)}(x-k) \qquad \|\Sigma\psi^{(i)}\|_\infty := \sup_{x,\eta} |\Sigma\psi^{(i)}(x;\eta)| \tag{2.67}$$

Also, the Hölder space embedding $C^1 \subset C^s$ (every continuously differentiable function is Hölder continuous) for $s < 1$ implies that the series $\sum \phi(x;\eta) \in C^s$. We define

$$\left|\left|\sum \phi^s(x)\right|\right|_\infty := \sup_{x\neq y} \frac{\sum \phi(x;\eta) - \sum \phi(y;\eta)}{||x-y||^s} \tag{2.68}$$

Now we can bound the derivatives of f_j as :

$$|\partial_{x_i} f_j(x)| \leq C_1 2^{-j(s+n/2)} 2^{j(n/2+1)} \|\Sigma\psi^{(i)}\|_\infty \qquad \text{for all } x$$

$$\text{So, } ||\partial_{x_i} f_j||_\infty \leq C_1 \|\Sigma\psi^{(i)}\|_\infty 2^{-j(s-1)} \tag{2.69}$$

Similarly, we also get

$$||\partial_{x_i} f_{-1}||_\infty \leq C_0 \|\Sigma\phi^{(i)}\|_\infty \tag{2.70}$$

Finally, we have everything we need to estimate C_H. Define $r_j(x; x_0) := f_j(x) - f_j(x_0)$ and $r(x; x_0) := f(x) - f(x_0) = \sum_j r_j(x; x_0)$, for any $x_0 \in \mathbb{R}^n$. Then, we need to find C_H s.t $|r(x; x_0)| \leq C_H \|x - x_0\|$. Let $m \in \mathbb{Z}$ be defined by $2^{-m} \leq \|x - x_0\| < 2.2^{-m}$. We can split the series for $r(x; x_0)$ as

$$r(x; x_0) = r_{-1}(x) + \sum_{j=0}^{m-1} r_j(x; x_0) + \sum_{j \geq m} r_j(x; x_0) \tag{2.71}$$

We have the following two cases :

Case 1: $\|x - x_0\| < 1$ so that $m > 0$

Starting with the last term of equation (2.71), we have :

$$\begin{aligned}
\left| \sum_{j \geq m} r_j(x; x_0) \right| &\leq \sum_{j \geq m} |f_j(x)| + |f_j(x_0)| \\
&\leq \sum_{j \geq m} 2C_1 \|\Sigma\psi\|_\infty 2^{-js} \qquad \text{(from equation (2.61))} \\
&= 2C_1 \|\Sigma\psi\|_\infty \frac{2^{-ms}}{1 - 2^{-s}} \\
&\leq \frac{2C_1 \|\Sigma\psi\|_\infty}{1 - 2^{-s}} \|x - x_0\|^s
\end{aligned} \tag{2.72}$$

This holds for $s = 1$ as well. To deal with the middle term of equation (2.71), we use the mean value theorem to bound each r_j.

$$\begin{aligned}
|r_j(x; x_0)| &= \left| \sum_{k=1}^{n} (x_k - x_{0k}) \frac{\partial f_j}{\partial x_k}(x') \right| \qquad \text{(for some } x' \text{ between } x \text{ and } x_0\text{)} \\
&\leq \sum_i \|x - x_0\| \cdot \|\partial_{x_i} f_j\|_\infty \\
&\leq C_1 \sum_i \|\Sigma \psi^{(i)}\|_\infty 2^{j(1-s)} \|x - x_0\| \qquad \text{(from equation (2.69))}
\end{aligned}$$

$$\begin{aligned}
\text{So, } \left| \sum_{j=0}^{m-1} r_j(x; x_0) \right| &\leq C_1 \sum_i \|\Sigma \psi^{(i)}\|_\infty \sum_{j=0}^{m-1} 2^{j(1-s)} \|x - x_0\| \\
&= C_1 \sum_i \|\Sigma \psi^{(i)}\|_\infty \frac{2^{m(1-s)} - 1}{2^{1-s} - 1} \|x - x_0\|
\end{aligned}$$

Now $2^{m-1} < \|x - x_0\|^{-1}$ implies $2^{m(1-s)} \|x - x_0\| < 2^{s-1} \|x - x_0\|^s$. So we get

$$\left| \sum_{j=0}^{m-1} r_j(x; x_0) \right| \leq \frac{C_1 \sum_i \|\Sigma \psi^{(i)}\|_\infty}{2^{(1-s)} - 1} (2^{s-1} \|x - x_0\|^s - \|x - x_0\|) \tag{2.73}$$

We cannot use this bound for $s = 1$. In that case, since we are adding up m terms with the same bound for each, we get

$$\left| \sum_{j=0}^{m-1} r_j(x; x_0) \right| \leq C_1 \sum_i \|\Sigma \psi^{(i)}\|_\infty m \|x - x_0\| \tag{2.74}$$

Now we can use the fact that $m \leq 1 - \log_2 \|x - x_0\|$ to get

$$\left| \sum_{j=0}^{m-1} r_j(x; x_0) \right| \leq C_1 \sum_i \|\Sigma \psi^{(i)}\|_\infty (1 - \log_2 \|x - x_0\|) \|x - x_0\| \tag{2.75}$$

We can bound the first term of equation (2.71) using the Hölder norm bound from equation (2.68) as :

$$|r_{-1}(x)| \leq C_0 \left|\left| \sum \phi^s(x) \right|\right|_\infty \|x - x_0\|^s \tag{2.76}$$

Now we add the three terms from equations (2.76), (2.73), (2.72) to get

$$|r(x;x_0)| \leq \left(C_0 \left|\left| \sum \phi^s(x) \right|\right|_\infty + C_1 \frac{\sum_i \|\Sigma\psi^{(i)}\|_\infty}{2^{1-s}(2^{1-s}-1)} + C_1 \frac{2\|\Sigma\psi\|_\infty}{1-2^{-s}} \right) \|x-x_0\|^s \tag{2.77}$$

For $s = 1$, we can add up everything to get

$$\begin{aligned} |r(x;x_0)| \leq C_0 \sum_i \|\Sigma\phi^{(i)}\|_\infty \|x-x_0\| \\ + \|\Sigma\psi^{(i)}\|_\infty (1 - \log_2 \|x-x_0\|) \|x-x_0\| \\ + 4C_1 \|\Sigma\psi\|_\infty \|x-x_0\| \end{aligned} \tag{2.78}$$

The log term indicates that the wavelet coefficient decaying at the rate of $2^{-j(1+n/2)}$ is insufficient to restrict functions to the space C^1. Instead, this condition restricts functions to the Zygmund class $\Lambda_\star$, which includes some extra functions.

Case 2: $\|x - x_0\| \geq 1$ so that $m \leq 0$

The only change here is that the middle term disappears in equations (2.77) and (2.78).

Combining these two cases, for $s < 1$, we get the bound :

$$C_H \leq C_0 \left|\left| \sum \phi^s(x) \right|\right|_\infty + C_1 \frac{\sum_i \|\Sigma\psi^{(i)}\|_\infty}{2^{1-s}(2^{1-s}-1)} + C_1 \frac{2\|\Sigma\psi\|_\infty}{1-2^{-s}} \tag{2.79}$$

If we define

$$a_{21}(\psi;s) := \left|\left| \sum \phi^s(x) \right|\right|_\infty \text{ and} \tag{2.80}$$

$$a_{22}(\psi;s) := \frac{\sum_i \|\Sigma\psi^{(i)}\|_\infty}{2^{1-s}(2^{1-s}-1)} + \frac{2\|\Sigma\psi\|_\infty}{1-2^{-s}}, \tag{2.81}$$

we have the second inequality for $0 < s < 1$:

$$C_H \leq C \text{ and } C \leq a_{21}(\psi; s)C_0 + a_{22}(\psi; s)C_1 \tag{2.82}$$

If our problem domain is a *lattice* with $2^l := \min\|x - x_0\|$, we can show that this inequality is still valid for $s = 1$. From equation (2.74) using the fact that $m \leq -l$, we get the middle term as

$$\left|\sum_{j=0}^{m-1} r_j(x; x_0)\right| \leq C_1 \sum_i \|\Sigma\psi^{(i)}\|_\infty \max\{0, -l\}\|x - x_0\| \tag{2.83}$$

So a_{22} has now changed to :

$$a_{22}(\psi; s = 1) = \sum_i \|\Sigma\psi^{(i)}\|_\infty \max\{0, -l\} + \frac{2\|\Sigma\psi\|_\infty}{1 - 2^{-1}} = \sum_i \|\Sigma\psi^{(i)}\|_\infty \max\{0, -l\} + 4\|\Sigma\psi\|_\infty \tag{2.84}$$

The bounds ratios in table (2.2) were calculated for 1D discrete distributions using this formula with $s = 1$ and $l = 0$.

From equations (2.82) and (2.56), we have the bounds in the lemma :

$$C_H \leq C \text{ and } a_{12}(\psi; s)C_1 \leq C \leq a_{21}(\psi; s)C_0 + a_{22}(\psi; s)C_1 \tag{2.85}$$

□

2.B WEMD with biorthogonal wavelets

Theorem (2) holds in a slightly changed form for biorthogonal wavelets as well. In the auxiliary wavelet domain problem (2.19), we can keep the constraint, but we have to change the objective function since biorthogonal wavelets don't preserve inner products. Since these wavelets are

not orthonormal, the analysis (ϕ, ψ) and synthesis $(\tilde{\phi}, \tilde{\psi})$ scaling function and wavelet are different. They are related by the following biorthogonal relationship :

$$\int \phi(x-k)\tilde{\phi}(x-l) = \delta_{kl} \qquad \int \psi_\lambda(x)\tilde{\phi}(x-l) = 0$$

$$\int \psi_\lambda(x)\tilde{\psi}_\mu(x) = \delta_{\mu\lambda} \qquad \int \phi(x-k)\tilde{\psi}_\mu(x) = 0$$

The wavelet coefficients of a function in a biorthogonal wavelet series expansion are given by :

$$f_k = \int f(x)\phi(x-k)dx \qquad f_\lambda = \int f(x)\psi_\lambda(x)dx \tag{2.86}$$

and the function can be reconstructed as :

$$f(x) = \sum_k f_k\tilde{\phi}(x-k) + \sum_\lambda f_\lambda\tilde{\psi}_\lambda(x) \tag{2.87}$$

We can use equation (2.87) to compute the inner product of two functions.

$$\int f(x)p(x)dx = \int \left(\sum_k f_k\tilde{\phi}(x-k) + \sum_\lambda f_\lambda\tilde{\psi}_\lambda(x)\right)\left(\sum_l p_l\tilde{\phi}(x-l) + \sum_\mu p_\mu\tilde{\psi}_\mu(x)\right)dx$$

Let $\tilde{\theta}_\omega(x) := \tilde{\phi}(x-k)$ or $\tilde{\psi}_\lambda(x)$, i.e. the function $\tilde{\theta}$ represents either $\tilde{\phi}$ or $\tilde{\psi}$ and the index ω first runs over all k and then over all λ.

$$\begin{aligned} &= \sum_{\omega,\sigma} f_\omega p_\sigma \int \tilde{\theta}_\omega(x)\tilde{\theta}_\sigma(x)dx \\ &= \mathbf{f}^T\mathbf{U}\mathbf{p} \end{aligned} \tag{2.88}$$

where f and p are vectors of wavelet coefficients as before and

$$U_{\omega\sigma} := \int \tilde{\theta}_\omega(x)\tilde{\theta}_\sigma(x)dx \tag{2.89}$$

Thus the auxiliary wavelet domain problem now becomes :

$$\begin{aligned} &\text{Maximize } \mathbf{f}^T\mathbf{U}\mathbf{p} \\ &\text{subject to } |f_k| \le C_0 \quad \text{and} \quad |f_\lambda| \le C_1 2^{-j(s+n/2)} \end{aligned} \tag{2.90}$$

This is the same problem as before, except that we must change p to $\tilde{\mathbf{p}} := \mathbf{U}\mathbf{p}$. The solution is :

$$\hat{\mu}_s = C_0 \sum_k |\tilde{p}_k| + C_1 \sum_\lambda 2^{-j(s+n/2)} |\tilde{p}_\lambda| \tag{2.91}$$

If we set $C_0 = 0$ and $C_1 = 1$, we get the simplified formula :

$$d(p)_{wemd} := \sum_\lambda 2^{-j(s+n/2)} |\tilde{p}_\lambda| \tag{2.92}$$

Computing WEMD with biorthogonal wavelets will take a bit longer because we need to compute Up. This raises the overall complexity to $O(n^2)$, though we do not expect it to increase computation time significantly since matrix multiplication has much lower complexity constants than the fast wavelet transform. Although the matrix U is not sparse ($O(n)$ non-zeros), a lot of its elements are still zeros, and the rest can be precomputed and stored.

An advantage of using biorthogonal wavelets is that we can have wavelets with tighter bounds. The constant a_{12} depends on the analysis wavelet while a_{21} and a_{22} depend on the synthesis wavelet. Since

biorthogonal wavelets offer more freedom in choosing these two, we can expect wavelets with lower bounds ratios

$$\frac{C_U}{C_L} = \frac{a_{22}(\tilde{\psi}; s)}{a_{12}(\psi; s)}. \tag{2.93}$$

2.C Wavelet transform implementation

The speed of our fast EMD approximation primarily depends on using a good implementation of the wavelet transform. Initially, we experimented with Matlab's wavelet toolbox, but we found this to be inadequate for three reasons :

1. Matlab can only compute wavelet transforms for 1 and 2 dimensions. Practical histogram feature descriptors often have 3 or more dimensions.

2. Although Matlab programs are compiled at runtime, they are not optimized as well as a C/C++/Fortran program can be optimized. Matlab is still not very good at speeding up loops, which are the major computation in a wavelet transform. There is also some overhead because Matlab controls the running code, somewhat similar to running a C program in a debugger.

3. Matlab's lifting wavelet transform does not allow for boundary extension. This results in coefficients that allow perfect reconstruction, but do not preserve the L_2 norm.

Actually, we were unable to find an implementation of the wavelet transform that satisfies our requirements :

1. Fast implementation (C/C++/Fortran).

2. Lifting or another technique that does not pay a penalty for high data dimensions.

3. Preserves the L_2 norm.

4. Boundaries are handled using zero padding or periodic boundary extension. The signal domain (the real line or the circle) indicates which is appropriate.

5. Finally, the ability to handle *arbitrary* dimensional data.

We have implemented a wavelet transform algorithm that satisfies all these requirements. Some of the important design choices we made were as follows :

1. Using the lifting wavelet transform (LWT) algorithm.

2. The LWT can be computed in place without using extra memory, if no boundary extension is applied. Since we want to preserve the signal energy, we use zero padding or periodic padding.

3. We chose *Blitz++* [Veldhuizen, 1998] as our underlying library for handling multi-dimensional arrays. This is a C++ numerical computing template library that provides an expressiveness almost on par with Matlab code while maintaining speed comparable to optimized Fortran code. Blitz++ uses a C++ feature called *expression templates* to avoid creation of large temporary variables during computation.

Our wavelet transform library can use most common orthogonal (Daubechies, symmetric Daubechies, Coiflets) as well as biorthogonal (Cohen-Daubechies-Feauveau) wavelets. Although we do not use them, in-place computation without boundary extension and inverse transforms are also provided. We also implement a method to initialize [Zhang et al., 1996] the fine scale (scale = 0) wavelet coefficients from the data, before starting the LWT algorithm.

Our wavelet transform library can compute a scale 3 wavelet decomposition using order 2 Coiflets of $16 \times 16 \times 16$ sized data in 7ms, about 14 times faster than Matlab. Both computations were performed on a computer with an Intel Xeon HT 3GHz processor with 3GB RAM.

Chapter 3

General deformation invariant matching

3.1 Introduction

Image matching is the process of finding corresponding points or regions in two images that were produced by the same or similar objects. This is a fundamental subproblem of many important vision tasks like mosaicking, stereo, registration, pose determination, recognition and classification. Matching is hard because the same 3D object produces very different 2D images under different conditions. Most 3D objects or scenes appear very different after a change in viewpoint. Non-rigid and articulating objects can change their appearance even without a change in viewpoint. Assuming diffuse reflection, one way to describe these changes in appearance is by a deformation. A *deformation* of an image is a continuous one to one mapping of its domain, that may expand, contract or leave the domain unchanged, but does not change the image intensities.

For objects with almost planar faces, an affine or piecewise affine deformation can be a good approximation to the complex deformation produced by a change in viewing direction. However, this is insufficient for a 3D object close to the camera with a curved structure, since its image undergoes a complex non-linear deformation. This deformation is

Figure 3.1: Images can deform because objects deform (the flag) or because changes in viewpoint distort their images (the cookie tin). Images from [Ling and Jacobs, 2005]

not completely arbitrary, because it is parallel to the epipolar line and the ordering constraint is obeyed. Often, we also have to include occlusion to completely describe the effect of a viewpoint change, since 3D objects and scenes produce varying self-occlusion during a viewpoint change. A good object recognition system should be able to disregard these effects. Another important area in which we need to deal with complex nonlinear deformations is image registration. If non-rigid objects are involved, the deformation may be almost arbitrary. Matching these non-rigid ob-

jects in a deformation invariant way can be a good initialization step for registration.

We propose using the *contour tree* as a novel framework invariant to arbitrary (smooth) deformations for representing and comparing images. Consider the graph of the image, i.e. the image embedded as a 2D surface in 3D Euclidean space. This surface can be divided into a set of iso-intensity contours, marked off at intervals of a unit intensity level. We can describe the enclosure relationships of this set of contours as a tree, called a contour tree [Morse, 1968] or a topological change tree [Kweon and Kanade, 1994]. Each contour that represents a local extremum or saddle point is a node. Two nodes are connected if one of their contours directly encloses the other. We label the nodes in the contour tree with their intensity values.

Two images related by exactly a deformation have isomorphic trees. For generic images, i.e. images without constant regions, this descriptor is a complete representation modulo deformations. We can reconstruct a deformed version of the image from this descriptor. At the same time, this is a sparse representation with no redundant information. If a deformation is the only possible difference between two images, we can compare two descriptors very efficiently (in time linearly related to the complexity of the two images) to decide if the corresponding images are deformed versions of each other. If the images are subject to noise or occlusion, we can compute the constrained edit distance [Zhang, 1996] between two

trees as a measure of the deformation invariant distance between the images. The edit distance can be thought of as the minimum number of image contours that need to be added or removed to change one image into another. This takes time approximately cubic in the image complexity. In practice, image complexity is represented by a number around $100 - 200$ for a natural image of size 200×300.

3.2 Related work

Mikolajczyk and Schmid [Mikolajczyk and Schmid, 2005] give a good review and evaluation of various local descriptors. Many of these are invariant to affine deformations and show good results in locating corresponding points. Our work has some similarity to the maximally stable extremal region (MSER) descriptor of Matas *et. al* [Matas et al., 2002], used to obtain region correspondences between two images. An MSER is a connected component obtained by a thresholding of the image, with the property that the relative region size changes the least on changing the threshold. MSERs have been successfully used to match corresponding points in applications like stereo. However, MSERS do not encode the local topology, nor are the recorded thresholds deformation invariant. The GIH descriptor [Ling and Jacobs, 2005] is invariant to arbitrary deformations, but is computationally expensive. It also does not capture all the deformation invariant information in the image. Moreover, all of

these approaches are local and do not attempt to take into account the relations between neighbouring interest points.

Vedaldi and Soatto [Vedaldi and Soatto, 2005] show that although non-trivial viewpoint invariant descriptors exist, they must lose some shape information. Our approach loses shape information for a 3D object, which gives us viewpoint invariance. We also lose shape information for the albedo pattern, and this gives us invariance to non-rigid deformations.

The contour tree was introduced by Morse [Morse, 1968] to describe the topological properties of a surface. It is a simplification of the general *Reeb graph* [Reeb, 1946] for describing the topology of the contours of functions defined on a manifold. The Reeb graph becomes the contour tree when the manifold has genus zero (i.e. does not have any holes or handles). Both the Reeb graph and the contour tree have many applications in computer graphics, visualization, geographic information systems and computer aided geometric design. Multiresolution Reeb graphs have been used for searching in a database of 3D models by Hilaga *et al.* [Hilaga et al., 2001]. They represent a 3D shape by a multiresolution Reeb graph with extra contour characteristics stored at the nodes. This representation is invariant to rotation, scale and articulations, and resistant to deformations as well. The multiresolution nature of the Reeb graph enables fast matching with automatic part correspondence. Contour trees are used for speeding up the extraction of contours from meshes [van

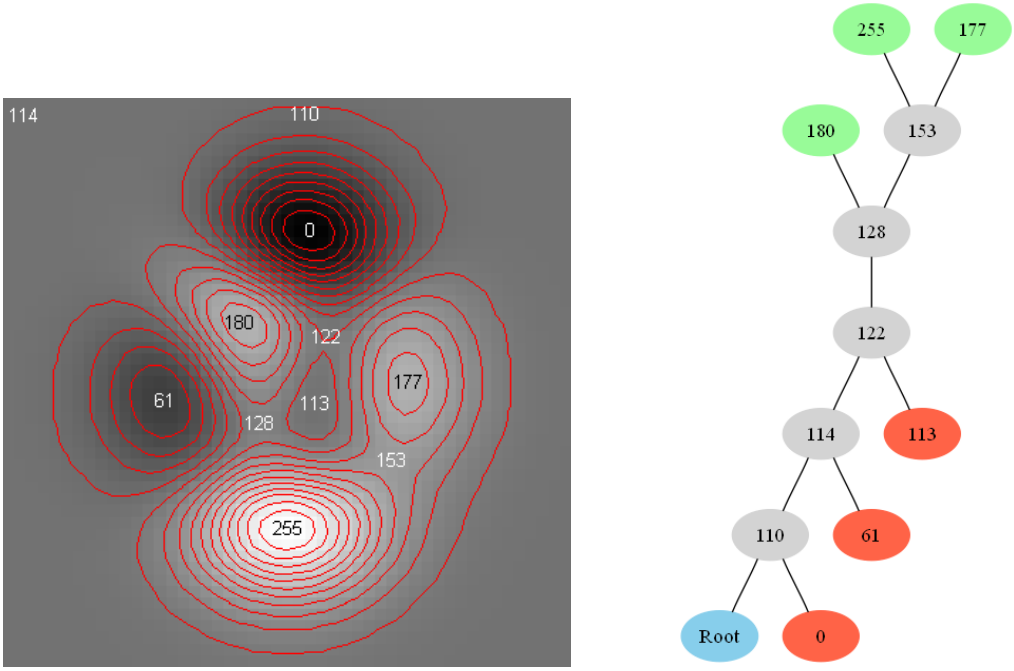


Figure 3.2: A simple synthetic image and its contour tree. Green nodes represent local maxima, red nodes local minima and gray nodes are saddle points. The root node at the bottom corresponds to the added enclosing minimum contour.

Kreveld et al., 1997]. They are also used to represent and extract topographic features in digital elevation models (DEM) [Kweon and Kanade, 1994].

3.3 The contour tree

Suppose that our images are 2D functions defined on a continuous domain. A deformation is a continuous one to one mapping of the image domain that leaves the intensity values unchanged. We can think about

this as a sort of stretching or compression of the image surface. Since deformations change the image domain, we must look at the range (intensity values) of the image for deformation invariant features. Deformations change the shape of the iso-intensity contours of an image, but leave their relative arrangement and values unchanged. The contour tree captures this arrangement of the iso-intensity contours. To begin with, each contour is assigned a node in the contour tree. Two nodes are joined by an edge if the two contours *touch*, i.e. there is no other contour between them, and one of the contours encloses the other. For smooth images, touching contours will always have an intensity difference of 1. Each contour divides the image into two distinct regions. If we want a path from an inside contour (or point) to one outside, it must pass through this contour. This means that in this graph, there is a unique path between any two nodes, i.e. it is a contour *tree*.

An alternate way to define a contour tree that clearly states its deformation invariant nature is given by [Pascucci, 2001]. He defines the contour tree of an image as the graph obtained by contracting each contour to a single point. Thus two images are related by a deformation if and only if they have isomorphic trees. Since all image information can be represented by the contour values, contour arrangement and contour shape, the contour tree contains all the deformation invariant information in an image. Further, we can recreate any deformed version of an image from its contour tree by expanding each node till it becomes a con-

tour. The expansion is only constrained by the fact that no two contours can intersect.

This contour tree consists of long chains of nodes joined at saddle point contours, and terminated at contours corresponding to extrema. This is a redundant representation since the chains can be inferred from their terminating saddle points and extrema. Actually, a contour tree is normally defined only with nodes that represent level set topology changes, i.e. extrema and saddle points. Removing all other nodes gives us a compact contour tree representation. Note that now the edges between nodes represent the regions between extrema and saddle points.

A third way of looking at the contour tree is by considering the topology of level sets of the image surface. A level set is a horizontal cut of the image surface at a particular intensity. As we increase the intensity, the level set's topology changes. It may split into components or some components may merge together or new components may appear or some components may disappear. The contour tree represents these topological changes in the level sets. Each edge corresponds to a connected level set component while a node marks a topological change. This definition of a contour tree helps us to reason out some of its properties. It is clear that at leaves in the contour tree, where level set components appear or disappear, we must have local extrema. When two components split or merge, the plane containing the level set is tangent to the image surface at one point. In a small neighbourhood of this point of contact, the image

surface lies on both sides of the tangent plane. Hence, this point is a saddle point and a node with more than 2 neighbours represents a saddle point in the contour tree.

Figure 3.2 shows a simple synthetic image and its corresponding contour tree. Since tree comparison is easier with rooted trees, we add a contour of value lower than any image value enclosing the whole image. This minimum node is designated as the root of the tree.

3.3.1 Construction

We will only briefly describe the construction of contour trees from a discrete image here. For details, please refer to Carr, Snoeyink and Axen [Carr et al., 2000]. The construction of the contour tree is complicated by several factors like ambiguity in interpolation of contours between grid points and existence of degenerate saddle points. The interpolation ambiguity arises because there is no unique way to represent an arbitrary point in $\mathbb{R}^2$ as a linear combination of its 4 nearest grid points. Consequently, we cannot assign a unique intensity value to that point through interpolation. This makes tracking contours difficult. This is usually solved by converting the square grid into a triangular (simplicial) mesh by joining every alternate grid point to its 4 diagonal neighbours (3.3). Now we can uniquely assign an intensity value to any non-grid point in the image based on the intensity values of its 3 grid point neighbours in the

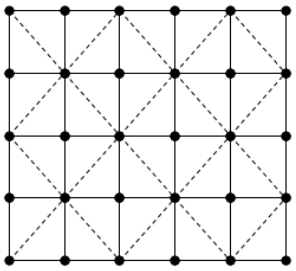


Figure 3.3: Converting a 2D square grid to a simplicial mesh by joining alternate grid points to 4 diagonal neighbours.

simplicial mesh. Next, we use the algorithm of Carr, Snoeyink and Axen [Carr et al., 2000] to compute the contour tree of this simplicial mesh. Their algorithm essentially consists of tracking the topological changes in the levels sets as the intensity is increased. This is complicated since the level set components may appear, disappear, split or merge. These operations may take place simultaneously at degenerate saddle points. It is simpler to look at the topological changes in the components of the sublevel and superlevel sets. A sublevel set, as the name implies, is the set of all image domain points with intensity less than or equal to a specified value. Similarly, a superlevel set is the set of all image domain points with intensity greater than or equal to a specified value. As the intensity level increases, sublevel set components can only be created or merge together. These topological changes are captured in the *join tree*. Its nodes are the iso-intensity contours at which these changes happen. The opposite is true for the superlevel set components – they can only split up or disappear. This gives us the analogous *split tree*. The two trees are augmented to represent the union of the two sets of contours and merged

together to produce the contour tree.

Since an image is finite, many contours are cut off at the boundary. This is remedied by enclosing the whole image in a contour with intensity equal to the minimum intensity value of the image. This minimum contour also gives us a unique node in the tree to designate as the root.

3.3.2 Relation to the GIH

The geodesic intensity histogram [Ling and Jacobs, 2005] is a deformation invariant descriptor for the region around interest points. It describes the distribution of intensity values around the interest point with respect to the geodesic distance from the interest point. The geodesic distance is the distance travelled along the surface, with intensity scaled by $\alpha \in [0, 1]$ and the spatial (x and y) dimensions scaled by $1-\alpha$. The geodesic distance and consequently the GIH is invariant to arbitrary deformations when $\alpha = 1$.

For $\alpha = 1$, we can easily compute the GIH around a point from the complete contour tree, i.e. when all contours have representative nodes. We start from the contour on which the point lies and traverse the tree in a depth first manner. Jumping from a contour to its adjacent contour increases the geodesic distance by 1, and the intensity increases or decreases by 1. At the first visit to each contour node, we increment the corresponding GIH bin by 1. The depth first traversal continues till

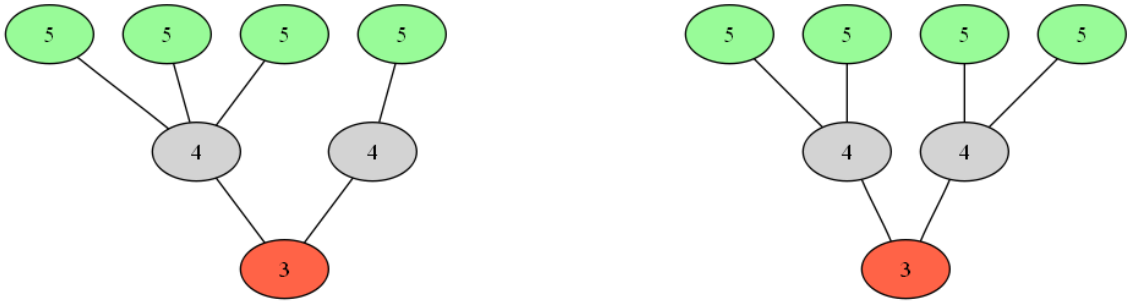


Figure 3.4: Images with these two (non-isomorphic) contour trees have the same GIH for an interest point on the minimum (bottom) contour.

a maximum depth equal to the maximum geodesic distance required is attained. Finally, we can bin the histogram as needed.

The GIH is not a complete deformation invariant representation because there exist images not related by a deformation that have the same GIH. The figure (3.4) shows the contour trees of two such images. They are not related by a deformation since their contour trees are not isomorphic. Points lying on the minimum (bottom) contour have the same GIH. Aggregation during histogram computation leads to the loss of some topological information.

3.3.3 Factors affecting the contour tree

We now have a complete deformation invariant description of an image. Comparing two images in a deformation invariant way now boils down to the comparison of labelled unordered trees. To decide how to compare contour trees, we must know how various image transformations affect

it. We will now look at some important factors.

Cropping and occlusion: For the complete contour tree (with a node for each contour), these cause the removal of some subtrees. For a normal contour tree, extrema values can change and some saddle points may change to extrema or normal contours. Local minima values can only increase, while local maxima values can only decrease. Contours interrupted by cropping will get completed through the new image boundary.

Lighting: Different types of lighting changes have different effects on the contour tree.

1. Simple additive brightness changes simply add a constant to all the node labels.
2. Contrast changes will scale all intensity values and hence the node labels.
3. Monotonic changes in image intensity change all node labels. The tree structure is left unchanged since the order of intensities is preserved. Monotonic intensity changes are equivalent to those that leave the image gradient direction unchanged. [Chen et al., 2000] have shown that most lighting changes do not significantly change the image gradient direction for Lambertian objects. Thus, we can say that the contour tree structure will not change significantly because of lighting changes.

4. General non-uniform lighting changes will have a complicated effect on the structure as well as the values.

We can compensate for the brightness and contrast changes by normalizing the image intensities to have a fixed mean and variance before computing the contour tree. Our algorithm does not provide invariance to monotonic or more complicated intensity variations since this would require throwing away too much information.

Noise: Gaussian noise will create a lot of saddle points with very short branches on top of the contour tree. Thus, a distance measure between two trees should be effective at ignoring short branches. Salt and pepper noise, on the other hand, can be difficult to handle because it will create saddle points with long branches. These cannot be distinguished from important image features. We can use median filtering to remove this noise.

Discretization: The contour tree is constructed from image values sampled from a rectangular grid. It will be accurate only if all critical points (extrema and saddle points) are preserved by the discretization process. In general, discretization can change critical point values or even make whole contour subtrees disappear. Similar to occlusion, it will always increase minima and decrease maxima. The effect of discretization is greatest when a part of the image is heavily compressed during deformation, for example when a viewpoint

change makes an object surface nearly parallel to the viewing direction. This usually leaves a *depth discontinuity* in the image. If the images are smooth, we can assume that discretization has not had this effect and has at most changed the values of the critical points.

3.4 Comparing contour trees: edit distance

There are various algorithms for comparing trees, each with a different set of assumptions about acceptable tree transformations. The factors affecting contour tree described above preclude using exact tree comparison methods like tree isomorphism [Hopcroft and Wong, 1974], even though this can be done in time linear in the number of tree nodes. There are many algorithms for approximate tree comparison like tree edit distance, tree alignment distance, tree inclusion, etc. [Bille, 2005]. Most are unsuitable either because they have very high computational complexity (many unordered tree matching algorithms are NP-hard) or make unsuitable assumptions (*e.g.* only leaves are labelled or the tree is binary). We will now look at a polynomial time algorithms for comparing unordered labeled trees.

The tree edit distance problem is to convert a tree T_1 into tree T_2 by a sequence of node insertions, deletions and substitutions (relabelings). This sequence is called an *edit script*. An edit operation is denoted by $v \to w$. Insertions and deletions are represented as $\lambda \to v$ and $v \to \lambda$

respectively, where λ denotes a blank node. Each operation has an associated non-negative cost, denoted by $\gamma(v \to w)$. Dynamic programming is used to compute the *optimal edit script*, i.e. the edit script with minimum total cost. This is the tree edit distance. T_2 can be converted into T_1 by using the reverse edit script, i.e. the edit script obtained by reversing each operation as well as the operation order. If node insertion and deletion have the same cost and the substitution cost is symmetric, then the tree edit distance is symmetric as well. The edit script defines a one-one mapping between the nodes of the two trees. Computing the tree edit distance for unordered trees is NP-complete but there are polynomial time algorithms if node matches are constrained.

The *constrained edit distance* of [Zhang, 1996] for rooted unordered labelled trees places constraints on which nodes can be mapped to ensure that the mapped nodes in the two trees form a similar tree structure. More accurately, if the nodes v_1, v_2 and v_3 from tree T_1 are mapped to the nodes w_1, w_2 and w_3 from tree T_2 respectively, then the nearest common ancestor (nca) of v_1 and v_2 is an ancestor of v_3 if and only if w_1 and w_2 is an ancestor of w_3. An equivalent condition is that if none of v_1, v_2 and v_3 is an ancestor of any of the others, then v_1 and v_2 have the same nca as v_1 and v_3 if and only if w_1 and w_2 have the same nca as w_1 and w_3 [Bille, 2005]. The constrained tree edit distance algorithm compares two trees with n_1 and n_2 nodes each in $O(n_1 n_2 (n_1 + n_2) \log(n_1 + n_2))$ time.

We describe the algorithm in brief here; please see [Zhang, 1996] for

details. In the following description, we abuse notation by using the same symbol for a node and its label. $p(v)$ represents the parent of the node v. To begin with, the cost (denoted by γ) of deleting subtrees rooted at each node (corresponding to unmatched image contours) of the two trees is calculated. We set the cost of deleting a leaf node as

$$\gamma(l \to \lambda) := |p(l) - l| \qquad (l \text{ is a leaf}) \tag{3.1}$$

This is the number of contours that would be lost in the image if an extremum disappears. The cost of deleting a saddle point (non-leaf) is

$$\gamma(v \to \lambda) := \sum_{w \text{ is a child of } v} \gamma(w \to \lambda) + |p(v) - v| \tag{3.2}$$

The second term is absent when we delete the root. Each pair of nodes (v, w) from the trees T_1 and T_2 respectively are tested for possible matches. Each match has an associated cost or distance. A distance is calculated between the subtrees rooted at these nodes by calculating the minimum distance, from the 3 possible ways of matching :

Substitution: A direct match. Let (m, t, s) be the label triple of merge time, region type and saddle point merge time (if any) of a node. Then, the distance between two leaves v and w is :

$$\gamma(v \to w) := |(p(v) - v) - (p(w) - w)| \qquad (v, w \text{ are leaves}) \tag{3.3}$$

This distance can be interpreted as number of contours that would be added or deleted to stretch the contours between v and its parent

saddle point into w and its parent saddle point. For non-leaves, we add the cost of matching sub-forests rooted at v and w. See [Zhang, 1996] for details.

Insertion: Node v is matched to a child w' of w. Subtrees rooted at all other children of w are deleted. The cost is calculated accordingly.

Deletion: Node w is matched to a child v' of v. v and its other children are deleted. The cost is calculated accordingly.

A dynamic programming algorithm gives us the cost of matching subtrees as well as subforests rooted at all pairs of nodes of the two trees using these recursive rules. The distance between the root nodes gives the cost of matching whole trees. However, since the root nodes are artificially introduced, the actual image matching cost is the cost of matching the subforests obtained by removing the root nodes. We can match one image (T_1) to only a part of the other (T_2) by choosing the lowest subforest match cost among subforests rooted at all the nodes of T_2. It is useful to normalize the distance between two trees by the geometric mean of the number of their nodes. This helps to ensure that distances between simple images can be compared with that between complex images. For partial matching, we normalize by the size of the matched subforest. We have described this algorithm using the L_1 metric to combine distances between different nodes, but we can use any other metric as well.

To find the mapping between the contours in the two images, we

need to keep track of the result (which of substitution, insertion or deletion had the least cost ?) of each node comparison. Contour matches are found by traversing the trees starting from their roots, according to the result of the comparison operations. Note that this gives us only one possible match out of all those that have the same minimum cost.

3.5 Future work

3.5.1 Experiments

We have presented a framework for deformation invariant image matching. We want to test this framework on synthetically deformed images as well as on images showing natural deformations. We expect that discretization effects, lighting and noise will affect matching performance. We think that this framework will work well on smooth low noise images with the same lighting. The smoothness is necessary to reduce discretization effects. We also want to test which of these conditions can be relaxed (and by how much) so that this framework is more generally applicable.

3.5.2 Multiresolution contour tree

A multiscale image representation is very useful in matching. Coarser scales contain less representation and can be matched faster than the whole descriptor. This can be used to quickly eliminate very different

images. Further, information at finer scales is more susceptible to noise. Since the amount of noise in the image is unknown, it is advantageous to have a multiscale descriptor and try matching at a few different scales. If we assume that the two images to be compared are deformed and noisy versions of each other, the least distance among all scale pairs is a good estimate of the actual image distance. We assume that this scale pair was produced by the correct amount of denoising in both images.

A multiscale contour tree can be simply produced by quantizing the contour intensities. If two connected nodes have the same quantized intensity, they are joined. This is similar to the multi–resolution Reeb graph of [Hilaga et al., 2001].

Chapter 4

Recognition of specular objects

4.1 Introduction

The appearance of an object varies significantly with a change in incident lighting. Model based recognition approaches simulate this change to reduce sensitivity to lighting variations. This approach has been successful for objects with diffuse or Lambertian reflectance. However, recognizing shiny or specular objects is still difficult because their appearance changes dramatically with even a minor change in lighting. We will explain why it is important to enforce a non-negative lighting constraint when solving this problem for specular objects. We will then describe a new exact and fast method to enforce it.

Model based object recognition is performed by comparing the image to an object model. A model includes a structural description (e.g.: regularly sampled surface normals) and an optical description (surface albedo, BRDF, etc.). The comparison is an optimization over all possible lighting conditions and produces an image from the model that is as close as possible to the query image. The object whose model produces the closest image is the most likely one to have produced the query image.

Since lighting intensity is a function of direction, and reflected light

is a function of the surface normal, both can be represented as functions on the surface of a sphere. Spherical harmonics provide a basis for these functions that is analogous to a Fourier series expansion for 1D functions (e.g. [Basri and Jacobs, 2003]). With this representation, the set of images that an object can produce lie in a linear subspace, with a dimension that depends on the number of harmonics we use. [Basri and Jacobs, 2003] show that only nine harmonics are needed to recognize convex Lambertian objects, because Lambertian reflectance acts as a low-pass filter. However, specular objects reflect higher frequency light (Thornber and Jacobs[Thornber and Jacobs]), so modeling their appearance requires many more harmonics.

Lighting is everywhere non-negative. With this constraint, a model's images form a convex subset of a linear subspace, making the matching problem more complex. When we use a low-dimensional subspace to represent Lambertian objects, ignoring the non-negative lighting constraint is not too serious. However, as the number of harmonics we use grows, the difference between the images produced by non-negative lighting and linear lighting models grows exponentially.

For example, suppose we try to incorrectly match a uniform albedo sphere to an image of a sphere that has a black dot on it. With low frequency harmonics, which suffice for Lambertian objects, we can never approximate this black dot; it has too many high frequencies. With a high-dimensional representation of light, which we need for specular ob-

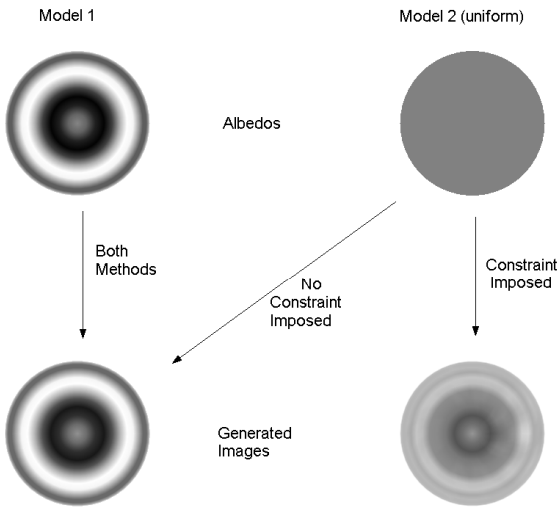


Figure 4.1: Two different albedo models (both are 4% mirror and 96% Lambertian) and images generated from them while trying to best match the image in the lower left figure. If negative light is allowed, we can get the lower left image from the top right albedo exactly.

jects, low frequency lighting can produce smooth shading on a sphere, and high frequencies can create a negative specular highlight that darkens the image in a small spot. To prevent this we must ensure that our optimization does not allow negative light (see Figure 4.1).

Ramamoorthi [Ramamoorthi, Aug 2002] points out that the non-negativity constraint also helps reduce high frequency noise, since by limiting the value of a function to be non-negative, we indirectly limit the value of its high frequency components.

To enforce non-negative light we want a constraint on the first spherical harmonic coefficients of light that will ensure that the light is non-negative everywhere. The lower order coefficients need not correspond

to a non-negative function, but there should exist a way to add higher order harmonics that will make the function non-negative. Looking at the problem more generally, we want to control the range of the function using only the first few coefficients.

In the analogous 1D case, i.e. for a Fourier series, an interesting theorem, due to Gabor Szego [Grenander and Szego, 1958], addresses this problem. It describes a Toeplitz matrix (see section 3.1) of the first few Fourier coefficients whose eigenvalues are contained in the range of the function. Also, the Szego Eigenvalue distribution theorem states that as we use more harmonics, the eigenvalues mimic the values taken by the function itself. As we make the Toeplitz matrix larger, the smallest and largest eigenvalues converge to the minimum and maximum values of the function respectively. Negative eigenvalues mean that the coefficients can not be extended to correspond to a non-negative function. So, if we constrain the eigenvalues to be non-negative (i.e. the matrix to be positive semidefinite), we can exclude all those low frequency functions that can not be extended by adding higher frequencies to become non-negative everywhere.

We extend this theorem to spherical harmonics. In this case, we obtain a much more complicated matrix whose eigenvalues are similar to the function values. To constrain this matrix to be positive semidefinite while minimizing the error between the query and generated images, we use semidefinite programming.

Next, we perform experiments on both synthetic and real data to explore the usefulness of imposing this constraint. We observe that imposing this constraint results in a significantly greater mismatch between the query and incorrect models, for most specular objects. This can improve recognition since now it is harder for the algorithm to get confused by noise in the model or query image.

This chapter is divided as follows. First, in section 4.2, we review some earlier work that has used the non-negativity constraint. In section 4.3, we present the extension of Szego's eigenvalue distribution theorem to spherical harmonics: the key ingredient in our algorithm. Next, in section 4.4, we review recovering lighting from an image given an object model, and formulate the problem as a semidefinite program. Finally, section 4.5 describes some experiments on synthetic as well as real data, which demonstrate the usefulness of the constraint.

4.2 Past Work

Various approaches to object recognition have used low dimensional linear subspace representations of the set of images produced by an object. For example, Hallinan [Hallinan, 1994], Murase and Nayar [Murase and Nayar, 1995] and Yuille et al. [Yuille et al., 1999] have used PCA to model lighting variation and Basri and Jacobs [Basri and Jacobs, 2003] and Ramamoorthi and Hanrahan [Ramamoorthi and Hanrahan, 2001] have

used a spherical harmonic representation for an analytic computation of the linear subspace of images. We also use a spherical harmonic representation for images and lighting.

Belhumeur and Kriegman [Belhumeur and Kriegman, 1998] have shown that the set of all possible images of an object under arbitrary lighting is a convex cone, the *illumination cone*. Lighting is represented as a convex combination (to ensure non-negativity) of the extreme rays of the convex cone. Computation and memory requirements can be reduced by projecting the image, the illumination cone and the extreme rays into a low dimensional subspace, although this makes the representation approximate. Calculation of the extremal rays can be avoided by further approximating lighting as a convex combination of rays uniformly sampled from the illumination sphere. They use a non-negative least squares routine to perform the convex optimization.

For Lambertian objects, Basri and Jacobs [Basri and Jacobs, 2003] build on this by expressing the uniformly sampled rays in terms of spherical harmonics. This approach works well for Lambertian objects since they only reflect the diffused (low frequency) components of the incident lighting which are well approximated by a few delta functions. However, since specular objects reflect many more components of light, a very large number of delta functions are needed to represent lighting accurately for them. This method is also approximate since the delta functions are approximated by a few low frequency harmonics and are no longer

just positive peaks. Non-negativity of lighting was also enforced by Ramamoorthi et al. [Ramamoorthi, Aug 2002] using a regularization term during optimization. This clearly cannot guarantee that the solution will be non-negative.

There have been many other attempts at recognizing specular objects. [Osadchy et al., 2003] have used specular reflection in recognition by decoupling Lambertian reflection and highlights and using them as separate cues. [Sato et al., 1991] use a physics-based simulator to predict specular features and analyze their detectability and reliability for recognition. Specularity detection is performed using a set of aspects generated from the model by deformable template matching. [Gremban and Ikeuchi, May 1993] use multiple views of an object to remove ambiguities due to specularities.

We will now describe a new, exact method for enforcing non-negativity, as a direct constraint on the spherical harmonic coefficients of light.

4.3 The non-negativity constraint

We need a condition on the first few spherical harmonic coefficients f_{lm} of a function $f(u) : S^2 \rightarrow \mathbb{R}$ that will imply that we can complete the spherical harmonic expansion of f such that $f(u) \geq 0$ for all u. Here, $u := (\theta, \phi)$ is a point on the surface of the unit sphere, denoted as S^2. This problem is easier to deal with in 1D, when we need a condition on the Fourier series

coefficients f_m of a function $f(\theta) : S^1 \to \mathbb{R}$ (θ is a point on the unit circle S^1). The condition for non-negativity that we obtain in these two cases is completely analogous; but the expressions are simpler for S^1 and the more familiar Fourier series will help us to understand the method better.

4.3.1 The Fourier case

Let $\mathcal{Q}_n$ denote the space of functions on S^1 spanned by $\{e^{im\theta} : 0 \leq m \leq n\}$, i.e. functions that only have low frequency components. The process of low pass filtering a function, so that the output belongs to $\mathcal{Q}_n$ is the same as an orthogonal projection from the space of all (integrable or L^1) functions to $\mathcal{Q}_n$. We will represent this operation as Q_n. Let f_m denote the m^{th} Fourier coefficient of the function $f \in L^1(S^1)$.

We will now develop some intuitive ideas about the non-negativity condition. First, let's represent the time domain product of two functions f and g, using only their Fourier series coefficients, as a product of a matrix (composed of the coefficients of f) and the vector of coefficients of g, denoted as $\hat{g}$. We can do this using the convolution theorem, if we consider only the first n coefficients of f and g. Then, the result will be the first n coefficients of the time domain product fg.

Let $[f]$ denote the operator *multiplication by* f. Then, denote the matrix of coefficients of f by $Q_n[f]Q_n$. In this notation, the first Q_n indicates that we are considering only the first n coefficients of f, which is equiv-

alent to applying an ideal low pass filter to f or projecting f into a low dimensional subspace spanned by the first n Fourier basis functions. The resulting time domain function is $f^{(n)}$. The second Q_n indicates the same for the function g. $\hat{g}_n = Q_n\hat{g}$ is a vector of the first n Fourier coefficients of g. Thus, we have,

$$Q_n[f]Q_n\hat{g} = Q_n\widehat{fg} \tag{4.1}$$

Using the convolution theorem, we arrive at the following form for the matrix $Q_n[f]Q_n$, called a Toeplitz matrix.

$$Q_n[f]Q_n = T_n(f) = \begin{bmatrix} f_0 & f_1 & \cdots & f_n \\ f_{-1} & f_0 & \ddots & \\ \vdots & \ddots & \ddots & f_1 \\ f_{-n} & & f_{-1} & f_0 \end{bmatrix} \tag{4.2}$$

The $(ij)^{th}$ element of this matrix is f_{j-i}. Now if $\hat{g}_n$ is an eigenvector of the matrix $Q_n[f]Q_n$, with the eigenvalue λ.

$$T_n(f)\hat{g} = \lambda\hat{g} \tag{4.3}$$

In the time domain, we have

$$f^{(n)}g^{(n)} = \lambda g^{(n)} \tag{4.4}$$

It is clear from this equation that λ lies in the range of values taken by $f^{(n)}$. Actually, we can show that λ lies in the range of f too. Also, although this is not obvious from our crude treatment, the eigenvalues λ

are representative of the values taken by the function f itself. These ideas are made concrete in Szego's eigenvalue distribution theorem [Grenander and Szego, 1958]. This theorem states that the mean value of any continuous function is the same whether it is applied to the eigenvalues of $Q_n[f]Q_n$ or to the values of the function f, i.e. the eigenvalues are "distributed" in the same way as the values of f. Hence, we can constrain the range of f by constraining the eigenvalues.

Before stating the theorem, we need this definition: the *essential lower bound* (or *essential infimum*, denoted by $\operatorname{ess\,inf}$) of a function $f(x)$ is the largest number m for which the inequality $f(x) \geq m$ holds everywhere, except perhaps in a set of measure zero. The *essential upper bound* (or *essential supremum*, denoted by $\operatorname{ess\,sup}$) is defined similarly. First, we give another result that states that the eigenvalues of $T_n(f)$ lie in the range of f.

Theorem 4. *Let $f(\theta) \in L^1(S^1)$ be a real valued function and $T_n(f)$ be the Toeplitz matrix of its Fourier series coefficients. $\lambda_i^{(n)},\ i = 1, \ldots, n+1$ are the eigenvalues of $T_n(f)$ arranged in non-decreasing order. Let m and M be the essential lower and upper bounds of $f(\theta)$ respectively. Then,*

$$m \leq \lambda_1^{(n)} \leq \lambda_2^{(n)} \leq \cdots \leq \lambda_{n+1}^{(n)} \leq M \tag{4.5}$$

The next step is the Szego Limit theorem, a fundamental result in the theory of Toeplitz forms proved by Gabor Szego in 1917, and extended in 1955. Here we only need the original limit theorem, and not its strong

form. This can be thought of as a particular case of the main theorem, and is the primary result used in its proof.

Theorem 5 (Szego Limit Theorem)**.** *Let $f(\theta) \in L^1(S^1)$ be a real valued function and $T_n(f)$ be its Toeplitz matrix as defined above. Then*

$$\lim_{n\to\infty} \frac{\operatorname{tr} \log T_n(f)}{n+1} = \frac{1}{2\pi} \int_{S^1} \log f(\theta) d\theta$$

Note that $\operatorname{tr} \log T_n(f) = \log \det T_n(f) = \sum_{i=1}^{n+1} \log \lambda_i^{(n)}$. Now, we state the main theorem.

Theorem 6 (Szego: Eigenvalue Distribution Theorem)**.** *With notation and conditions as above, and with m and M finite, let $F(\lambda)$ be any continuous function defined in the interval $\lambda \in [m, M]$, then*

$$\lim_{n\to\infty} \frac{F(\lambda_1^{(n)}) + \cdots + F(\lambda_{n+1}^{(n)})}{n+1} = \frac{1}{2\pi} \int_{S^1} F(f(\theta)) d\theta \tag{4.6}$$

Corollary 1. *With notation and conditions as above,*

$$\lim_{n\to\infty} \lambda_1^{(n)} = m, \quad \lim_{n\to\infty} \lambda_{n+1}^{(n)} = M \tag{4.7}$$

The proofs of these theorems (except that of the Szego Limit theorem) and the corollary are very similar to the proofs for the spherical harmonics casegiven in the next section. Proofs can also be found in Grenander and Szego [Grenander and Szego, 1958]. The first theorem and the corollary imply that if the function $f(\theta)$ is non-negative everywhere, then the matrix $T_n(f)$ is positive semidefinite [1] for all n. Also, as we will see from

[1]The symmetric matrix A is positive semidefinite if $x^t A x \geq 0$ for all vectors x, or equivalently, if all its eigenvalues are non-negative.

Theorem 10, if $T_n(f)$ is positive semidefinite, there exists a function f with the projection $Q_n f$ that is non-negative everywhere (except perhaps on a set of measure zero). Thus, even though the projection $Q_n f$ that we obtain may not be non-negative everywhere, we are guaranteed that it is the projection of a function that is non-negative everywhere. Also, if $T_n(f)$ is not positive semidefinite, we are guaranteed that the projection $Q_n f$ cannot be extended into a non-negative function f. Thus, using this constraint rules out all those lighting function projections and only those projections that do not correspond to a physical lighting function. Note that although the Dirac delta function is not an element of L^1, the theorem is valid for it too, since all the eigenvalues of $T_n(\delta)$ are zero, except for $\lambda_{n+1}^{(n)} = n$ which goes to infinity as n increases.

4.3.2 Spherical Harmonics

Next, we extend the theorem to the case of spherical harmonics, i.e. to functions on S^2. Let $\mathcal{P}_L$ be the space of functions that only have spherical harmonic components of order up to L. Correspondingly, P_L denotes the process of ideal low pass filtering the function f, so that we only retain spherical harmonic components of order at most L. As we go from S^1 to S^2, we have to restrict the set of functions on which the corresponding theorems are valid to $H^{\frac{1}{2}}(S^2)$, a Sobolev space of functions defined on S^2. A function is said to belong to a Sobolev space H^k if it has finite norm

(the L^2 norm in this case) and also if the norm of all derivatives of the function up to order k is finite. The derivatives need to exist only in a 'weak' sense. (see [Evans, 1998] for more on Sobolev spaces and [Okikiolu, 1996] for details on this theorem). $H^{\frac{1}{2}}(S^2)$ is the space of functions such that the norm $\|(I-\Delta)^{\frac{1}{4}}f\|_{L^2}$ is finite. Here, Δ is the Laplace-Beltrami second derivative operator, a generalization of the normal Laplacian operator for manifolds. I is the identity operator. For more details, see [Kreyszig, 1968]. $C(S^2)$ is the space of continuous functions defined on S^2. Most well-behaved and smooth functions belong to these spaces. First we show that the eigenvalues of $P_L[f]P_L$ are contained in the range of f.

Theorem 7. *Let $f(u) \in L^1(S^2)$ be a real valued function. Let m and M be the essential lower and upper bounds of $f(u)$, respectively, $\lambda_i,\ i = 1, \ldots, (L+1)^2$ are the eigenvalues of the matrix $P_L[f]P_L$. Then,*

$$m \leq \lambda_1^{(L)} \leq \lambda_2^{(L)} \leq \cdots \leq \lambda_{(L+1)^2}^{(L)} \leq M \tag{4.8}$$

Proof. This proof is similar to the proof for the Fourier case given in [Grenander and Szego, 1958]. Let $\hat{g}$ be any vector of length $(L+1)^2$ with unit l^2 norm and let $g(u) = \sum_{lm} \hat{g}_{lm} Y_{lm}(u)$. Thus, $\int_{S^2} |g(u)|^2 d\sigma(u) = \sum_{lm} |\hat{g}_{lm}|^2 = 1$. By definition of $P_L[f]P_L$, we have,

$$P_L[f]P_L\hat{g} = \widehat{fg}$$

Consider the following quadratic form :

$$\begin{aligned}
\hat{g}^* P_L[f] P_L \hat{g} &= \sum_{lm} \hat{g}^*_{lm} \widehat{fg}_{lm} \\
&= \sum_{lm} \hat{g}^*_{lm} \int_{S^2} f(u) g(u) \overline{Y}_{lm}(u) d\sigma(u) \\
&= \int_{S^2} f(u) g(u) \left(\sum_{lm} \hat{g}^*_{lm} \overline{Y}_{lm}(u) \right) d\sigma(u) \\
&= \int_{S^2} f(u) g(u) \overline{g(u)} d\sigma(u) \\
&= \int_{S^2} f(u) |g(u)|^2 d\sigma(u)
\end{aligned}$$

Since, g is normalized, the last expression is simply a weighted mean of the function f and hence it lies in the essential range of f. Hence, we have,

$$m \leq \hat{g}^* P_L[f] P_L \hat{g} \leq M$$

for all vectors $\hat{g}$ with unit norm. If we choose $\hat{g}$ to be an eigenvector corresponding to any eigenvalue λ of $P_L[f]P_L$, we get

$$m \leq \lambda \leq M$$

Thus, all the eigenvalues of $P_L[f]P_L$ are contained in the range of f. □

We will now state the Szego limit theorem which is the main result needed in the proof of the eigenvalue distribution theorem. A more general form of this theorem is stated and proved by Okikiolu in [Okikiolu, 1996].

Theorem 8 (Szego Limit Theorem in S^2)**.** *Let $f \in C(S^2) \cap H^{1/2}(S^2)$ be such*

that the closed convex hull of the image of f does not contain the origin, [2] *then*

$$\lim_{n\to\infty} \frac{\mathrm{tr}\log P_n[f]P_n}{(n+1)^2} = \frac{1}{4\pi}\int_{S^2} \log f(u) d\sigma(u)$$

Before stating and proving the eigenvalue distribution theorem, we need the formal definition of the concept of *equal distributions*, as given by H. Weyl [Grenander and Szego, 1958]. Two sequences of numbers $\{a_i^{(n)}\}_{i=1,\ldots,n+1}$ and $\{b_i^{(n)}\}_{i=1,\ldots,n+1}$ such that $|a_i^{(n)}| < K$ and $|b_i^{(n)}| < K$ for all i and n are equally distributed in the interval $[-K, K]$ as $n \to \infty$ if for any continuous function $F(t)$ defined in the interval $[-K, K]$, we have

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n+1}[F(a_i^{(n)}) - F(b_i^{(n)})]}{n+1} = 0 \tag{4.9}$$

Here we use a slightly modified definition in which for each n, the sequences consist of $(n+1)^2$ instead of $(n+1)$ numbers. Roughly, we can say that two sequences are equally distributed if they take on similar values. We also need this test for equally distributed sequences: Two sequences obey the equation (4.9) for arbitrary continuous functions F, i.e. they are equally distributed, if the equation (4.9) is satisfied for certain special classes of functions. Two such classes of these functions are $F(t) = log(1 + zt)$ where z is real and $|z| < K^{-1}$ and $F(t) = t^s$ where $s = 0, 1, 2, \ldots$

[2] Since are dealing with real valued functions only, this means that f takes either only positive or only negative values, but not both.

Theorem 9 (Eigenvalue Distribution Theorem in S^2)**.** *Let $f(u) \in C(S^2) \cap H^{1/2}(S^2)$ be a real valued function. Let m and M be the essential lower and upper bounds of $f(u)$, respectively and assume that m and M are finite. $\lambda_i^{(L)},\ i = 1, \ldots, (L+1)^2$ are the eigenvalues of the matrix $P_L[f]P_L$. If $F(\lambda)$ is any continuous function defined in the finite interval $\lambda \in [m, M]$, then*

$$\lim_{L\to\infty} \frac{F(\lambda_1^{(L)}) + \cdots + F(\lambda_{L+1}^{(L)})}{(L+1)^2} = \frac{1}{4\pi} \int_{S^2} F(f(u)) d\sigma(u)$$

This is a novel result. The proof of this theorem closely follows the proof of Szego's original theorem and uses Okikiolu's [Okikiolu, 1996] extension of a key lemma used in the theorem's proof - the Szego limit theorem.

Proof. The proof of this result follows Szego's original proof. Using the definition of Riemann integration as a limit, the theorem is equivalent to :

$$\lim_{L\to\infty} \frac{\sum_{m=1}^{(L+1)^2} F(\lambda_m^{(L)}) - F(f(u_m^{(L)})}{(L+1)^2} = 0$$

where $u_m^{(L)} = (\frac{2a\pi}{L+2} - \pi, \frac{b\pi}{L+2} - \frac{\pi}{2});\ a, b = 1, 2, \ldots, L+1;\ a + (b-1)L = m$. Using the definition of equal distributions we can restate the limit relation as follows:

The sequences $\lambda_m^{(L)}$ and $f(u_m^{(L)})$ are equally distributed.

Proving this for the special class of functions $F(t) = log(1 + zt)$ is sufficient. We will now apply the Szego limit theorem to the function $1 + zf(u)$, where $z \in \mathbb{R}$ is such that $|zf(u)| < 1$ for all $u \in S^2$. We can do this

since the function $f(u)$ is bounded. This transformation ensures that the closed convex hull of the image of $1 + zf(u)$ does not contain the origin. We have

$$P_L[1 + zf]P_L = I + zP_L[f]P_L$$

since the vector space $\mathcal{P}_L$ is closed with respect to scaling and shifting. The eigenvalues of $I + zP_L[f]P_L$ are $1 + z\lambda_m^{(L)}$. Hence, using the fact that for any nonsingular matrix A, $\operatorname{tr}\log A = \log\det(A) = \sum_i \log(\lambda_i(A))$, we can write

$$\lim_{L\to\infty} \frac{\sum_{m=1}^{(L+1)^2} \log(1 + z\lambda_m^{(L)})}{(L+1)^2} = \frac{1}{4\pi}\int_{S^2} \log(1 + zf(u))d\sigma(u)$$

Again using the definition of Riemann integration, we have

$$\lim_{L\to\infty} \frac{\sum_{m=1}^{(L+1)^2} \log(1 + z\lambda_m^{(L)}) - \log(1 + zf(u_m^{(L)}))}{(L+1)^2} = 0$$

Thus, the theorem is valid for the set of functions $\log(1 + zt)$ and hence is valid for all continuous functions. □

We also have the corresponding corollary :

Corollary 2. *With notation and conditions as above,*

$$\lim_{L\to\infty} \lambda_1^{(L)} = m, \quad \lim_{L\to\infty} \lambda_{(L+1)^2}^{(L)} = M \tag{4.10}$$

Proof. In the Fourier case, this is proved using a specific property of Toeplitz matrices (the fact that T_L is a principal submatrix of T_{L+1}) that does not hold in the case of spherical harmonics. Here we give a different proof of this result. From the proof of theorem 7, we have

$$\frac{\hat{g}_L^* T_L(f)\hat{g}_L}{\hat{g_L}^*\hat{g}_L} = \frac{\int_{S^2} f(u)|g_L(u)|^2 d\sigma(u)}{\int_{S^2}|g_L(u)|^2 d\sigma(u)}$$

The subscript L indicates that $g_L \in \mathcal{P}_L$. So, the minimum eigenvalue of $T_L(f)$ is given by

$$\lambda_{min}^{(L)} = \min_{g_L \in \mathcal{P}_L} \frac{\int_{S^2} f(u)|g_L(u)|^2 d\sigma(u)}{\int_{S^2} |g_L(u)|^2 d\sigma(u)}$$

Since f is defined on the closed and bounded interval S^2 and is continuous on it, it attains its essential lower bound. Let u_m be a point where $f(u)$ attains its essential lower bound. Now, choose g_L as the real valued function:

$$g_L = P_L[C_L e^{-L^2\|u-u_m\|^2}]$$

i.e., g_L is the projection of a Gaussian into $\mathcal{P}_L$. The variance of the Gaussian decreases as $1/L^2$, but the normalization constant C_L is chosen to keep its area constant. Clearly, as $L \to \infty$, $g_L \to C_g \delta(u - u_m)$ and $|g_L|^2 \to C_{g^2}\delta(u - u_m)$ as well, for some suitable constants C_g and C_{g^2}. C_L can be chosen to make $\int_{S^2} |g_L|^2 d\sigma(u) = C_{g^2} = 1$. Note that we can take g_L as any function in $\mathcal{P}_L$ whose square converges to a delta function. So, we have

$$\begin{aligned}
\lim_{L\to\infty} \lambda_{min}^{(L)} &\leq \lim_{L\to\infty} \frac{\int_{S^2} f(u)|g_L(u)|^2 d\sigma(u)}{\int_{S^2} |g_L(u)|^2 d\sigma(u)} \\
&= \frac{\int_{S^2} f(u) \left(\lim_{L\to\infty} |g_L(u)|^2\right) d\sigma(u)}{\int_{S^2} \left(\lim_{L\to\infty} |g_L(u)|^2\right) d\sigma(u)} \\
&= \frac{\int_{S^2} f(u)\delta(u - u_m) d\sigma(u)}{\int_{S^2} \delta(u - u_m) d\sigma(u)} \\
&= f(u_m) \\
&= m
\end{aligned}$$

Since we must have $\lambda_{min}^{(L)} \geq m$ for all L, we conclude that

$$\lim_{L\to\infty} \lambda_{min}^{(L)} = m$$

The second limit result can be proved similarly. □

All we need to do now is calculate the matrix $T_L(f) := P_L[f]P_L$. We will use something similar to the convolution theorem and calculate the $(l_1, m_1)th$ coefficient of the time domain product fg, where $g \in L^2(S^2)$ is any real valued function.

$$\begin{aligned}
(P_L[f]P_L\hat{g})_{l_1m_1} &= (\widehat{fg})_{l_1m_1} \\
&= \int_{S^2} fg\overline{Y}_{lm} d\sigma(u) \\
&= \sum_{l_2m_2}\sum_{lm} f_{lm}g_{l_2m_2}\int_{S^2} Y_{lm}Y_{l_2m_2}\overline{Y}_{l_1m_1} d\sigma(u) \\
&= \sum_{l_2m_2}\sum_{lm} f_{lm}g_{l_2m_2}G(ll_2l_1; mm_2m_1) \\
&= \sum_{l_2m_2}\sum_{l=|l_1-l_2|}^{l_1+l_2} f_{l,m_1-m_2}G(ll_2l_1; m_1-m_2, m_2, m_1)g_{l_2m_2}
\end{aligned}$$

Here, $G(l_1l_2l_3; m_1m_2m_3) = \sqrt{\frac{(2l_1+1)(2l_2+1)}{4\pi(2l_3+1)}}C(l_1l_2l_3; m_1m_2m_3)C(l_1l_2l_3; 000)$ is the Gaunt coefficient. $C(l_1l_2l_3; m_1m_2m_3)$ are the Clebsch-Gordan (CG) coefficients. Both these coefficients are real constants that arise naturally during the evaluation of integrals of products of spherical harmonics. For more details, please see Appendix 4.A.1 and [Rose, 1957; Homeier and Steinborn, 1996]. Thus the term in position $(l_1m_1, l_2m_2) := (l_1(l_1+1)+m_1, l_2(l_2+1)+m_2)$ in the matrix $P_L[f]P_L$ is

$$T_L(f)_{l_1m_1;l_2m_2} = \sum_{l=|l_1-l_2|}^{l_1+l_2} f_{l,m_1-m_2}G(ll_2l_1; m_1-m_2, m_2, m_1) \tag{4.11}$$

The choice of the subscripts was made so that 1 corresponds to the row number and 2 to the column number in the matrix. The size of the matrix is $(L+1)^2 \times (L+1)^2$, since there are $(L+1)^2$ spherical harmonics of degree less than or equal to L. We can show that this matrix is Hermitian, which is important because optimization software usually needs symmetric matrices as inputs.

$$\begin{aligned}
\overline{(P_n[f]P_n)}_{l_1m_1;l_2m_2} &= \sum_{l=|l_1-l_2|}^{l_1+l_2} \overline{f}_{l,m_1-m_2} \sqrt{\frac{(2l+1)(2l_2+1)}{4\pi(2l_1+1)}} C(ll_2l_1; m_1-m_2, m_2, m_1) C(ll_2l_1; 000) \\
&= \sum_{l=|l_1-l_2|}^{l_1+l_2} (-1)^{m_1-m_2} f_{l,m_2-m_1} \sqrt{\frac{(2l+1)(2l_2+1)}{4\pi(2l_1+1)}} \\
&\times (-1)^{m_2-m_1} \left(\frac{2l_1+1}{2l_2+1}\right) C(ll_1l_2; m_2-m_1, m_1, m_2) C(ll_1l_2; 000)
\end{aligned}$$

(using properties (4.21) and (4.23) of the CG coefficients as listed in 4.A.1)

$$\begin{aligned}
&= \sum_{l=|l_1-l_2|}^{l_1+l_2} f_{l,m_2-m_1} \sqrt{\frac{(2l+1)(2l_1+1)}{4\pi(2l_2+1)}} C(ll_1l_2; m_2-m_1, m_1, m_2) C(ll_1l_2; 000) \\
&= (P_n[f]P_n)_{l_2m_2;l_1m_1}
\end{aligned}$$

If the function f is non-negative everywhere, $T_L(f)$ is positive semi-definite (denoted as $T_L(f) \succeq 0$). We also need to look at the converse problem. Does the positive semidefiniteness of $T_L(f)$ imply that f is non-negative everywhere ? Since we deal only with the first few harmonic components of f, we can arbitrarily add higher order harmonics to f. This gives an infinite number of functions f corresponding to the same matrix $T_L(f)$, all of which are obviously not non-negative. However what we are interested in is the existence of at least one function with the given matrix $T_L(f)$, i.e.

with the given lower order harmonics, that is non-negative everywhere. This will ensure that the set of lower order harmonics obtained from the optimization corresponds to a non-negative lighting condition. For the Fourier case, we have the following theorem that answers this question.

Theorem 10. *Given the first n Fourier coefficients of a real valued function $f(\theta),\ \theta \in S^1$, if the Toeplitz matrix $T_n(f)$ (defined by equation (4.2)) is positive semidefinite, there exists a unique function with the given lower order Fourier coefficients that is non-negative everywhere. This is the sum of delta functions given by :*

$$f(\theta) = K_0 + \sum_{p=1}^{n} K_p \delta(\theta - \theta_p) \tag{4.12}$$

$$(K_0 = 0,\ K_p \geq 0)$$

Furthermore, if the matrix $T_n(f)$ is strictly positive definite, then there exist infinite functions with the given lower order frequency components that are non-negative everywhere.

Proof. First consider the case when $T_n(f)$ is positive semidefinite. We need to express the given Fourier coefficients as the sum of the Fourier series coefficients of a set of n non-negative delta functions. This can be done by using the theorem of Carathéodory, (Appendix 4.C and [Grenander and Szego, 1958] section 4.1). Corresponding to the complex constants $f_1, f_2, \ldots, f_n$, we have the unique real numbers $K_p \geq 0$ and $\theta_p \in S^1;\ p =$

$1, 2, \ldots, n$ such that

$$f_\nu = \sum_{p=1}^{n} K_p \, e^{i\nu\theta_p}$$

(note that some of the K_p may be zero). Since we already have $f_{-\nu} = \overline{f}_\nu$ and f_0 is such that the Toeplitz matrix formed from them is positive semidefinite, this equation is valid for all the given Fourier series coefficients of f. One possible function that has these initial Fourier series coefficients and is non-negative everywhere is :

$$f(\theta) = \sum_{p=1}^{n} K_p \delta(\theta - \theta_p)$$

Now suppose there is another function f' that is non-negative everywhere and has the same low frequency components. Then $f' - f$ is composed only of high frequency components and hence must be negative on a set of finite non-zero measure. Since f is zero everywhere except at finitely many points, the sum of f and $f' - f$ cannot be non-negative everywhere. We thus have a contradiction.

Now, if $T_n(f)$ is positive definite, we can write the following form for $f(\theta)$:

$$f(\theta) = K_0 + \sum_{p=1}^{n} K_p \delta(\theta - \theta_p)$$

To obtain this representation, note that we can write $f(\theta) = K_0 + \tilde{f}(\theta)$ such that $T_n(\tilde{f})$ is positive semidefinite and $K_0 > 0$. Now we can obtain a representation for $\tilde{f}(\theta)$ as before. To see that there are infinitely many other non-negative functions with the same $T_n(f)$, add any frequency component to $f(\theta)$ of order greater than n and magnitude less than K_0. The

resulting function has the same Toeplitz matrix and is non-negative everywhere. □

In the case of spherical harmonics, we have the corresponding conjecture:

Conjecture 1. *Given the frequency components of a real valued function* $f(u),\ u \in S^2$*, up to order* L*, if the matrix* $T_L(f)$ *(defined by equation* (4.11)*) is positive semidefinite, there exists a unique function* $f(u)$ *with the given lower order frequency components that is non-negative everywhere. This is the sum of* δ *function given by :*

$$f(u) = K_0 + \sum_{p=1}^{N(L)-1} K_p\ \delta(u - u_p) \tag{4.13}$$

$$(K_0 = 0,\ K_p \geq 0,\ N(L) = (L+1)^2)$$

Furthermore, if the matrix $T_L(f)$ *is strictly positive definite, then there exist infinite functions with the given lower order frequency components that are non-negative everywhere.*

We are unable to prove this because, as far as we know, Carathéodory's theorem does not have a spherical harmonics analog. If the existence part is proved, then uniqueness in the positive semidefinite case and the existence of infinite functions in the strictly positive definite case can be proved exactly as in the Fourier case.

4.4 Recovering Lighting from an Image: Semidefinite Programming

The problem of recovering lighting from the image of an object using its model, can be treated as an optimization problem. We assume a geometric model in the form of surface normals at the various pixel locations and a reflectance model. If we have several models from several different objects, the model that gives the least error corresponds to the object that created the image.

We represent lighting in terms of spherical harmonics and analytically compute the image when the object is illuminated by each individual harmonic. If these images are treated as vectors and stacked as columns of a matrix, we obtain the model matrix M. In rendering the images, we can use any reflectance model, or even a mixture of models. Now if the lighting is described by the spherical harmonic coefficient vector a, the resulting image (as a vector I) is given by $I = Ma$. If the image has N pixels and we use spherical harmonics up to order L to describe the image, M has size $N \times (L+1)^2$ and a and I are column vectors of lengths $(L+1)^2$ and N respectively. Then, given the query image $r = I +$ noise, a can be found by minimizing $\|Ma - r\|$ subject to $T_L(a) \succeq 0$. Since we model frequencies only up to L, the error will usually be non-zero even in the absence of noise. The problem size here is the number of pixels in the image, which can be pretty large. We can reduce this by transforming the

problem from the image space to the space of spherical harmonics basis images (see [Basri and Jacobs, 2003]). First, compute the QR decomposition of the matrix M, i.e. $M = QR$ where Q is an $N \times (L+1)^2$ matrix with orthonormal columns ($Q^TQ = I$) and R is an $(L+1)^2 \times (L+1)^2$ upper triangular matrix. Next we project both Ma and f into the low ($(L+1)^2$) dimensional subspace by multiplying with Q^T. We now need to solve the size $(L+1)^2$ problem:

$$\begin{aligned} \min_{a} \quad & \|Ra - Q^T r\|^2 \\ \text{subject to} \quad & T_L(a) \succeq 0 \end{aligned}$$

This is an optimization problem with a quadratic objective function and a matrix positive semi-definiteness constraint. Such problems are called semidefinite programming (SDP) problems. The matrix constraint itself is considered linear in SDP since each element of the matrix depends linearly on the vector a (see section 4.5.1) and is a type of a Linear Matrix Inequality. We can convert this into a linear problem (see [Todd, 2001]) by introducing a slack variable q.

$$\begin{aligned} \min_{a} \quad & q \\ \text{subject to} \quad & q > \|Ra - Q^T r\|^2 \ \text{ and } \ T_L(a) \succeq 0 \end{aligned}$$

Now, the Schur complement property (see [Todd, 2001]) is used to convert the quadratic constraint into the following equivalent linear constraint.

$$\begin{bmatrix} (1+q)I & \binom{1-q}{Ra-Q^Tr} \\ \binom{1-q}{Ra-Q^Tr}^T & (1+q)I \end{bmatrix} \succeq 0$$

This can be readily converted to a second order cone program (SOCP), which can be solved faster. In an SOCP, the constraint is of the form $\binom{\alpha}{v} \in K_2 := \left\{\binom{\alpha}{v} | \alpha \geq \|v\|_2\right\}$. K_2 is called the second order cone or Lorentz cone.

$$1 + q \geq \left\| \begin{array}{c} 1 - q \\ Ra - Q^T r \end{array} \right\|$$

So the final problem to be solved is :

$$\begin{aligned} \min_{a} \quad & q \\ \text{subject to} \quad & 1 + q \geq \left\| \begin{array}{c} 1 - q \\ Ra - Q^T r \end{array} \right\| \text{ and } T_L(a) \succeq 0 \end{aligned} \tag{4.14}$$

This is a mixed SOCP-SDP problem and can be solved using standard SDP software.

4.5 Experiments

4.5.1 Implementation

The entries of the matrix $T_L(a)$ are linear combinations of the entries of the vector a and are given by:

$$T_L(l_1(l_1+1)+m_1, l_2(l_2+1)+m_2) = \sum_{l=|l_1-l_2|}^{l_1+l_2} a_{l,m_1-m_2} G(ll_2l_1; m_1-m_2, m_2, m_1) \tag{4.15}$$

where $G(ll_2l_1; m_1 - m_2, m_2, m_1)$ are the Gaunt coefficients. Since each element of T_L is a linear combination of Gaunt coefficients, we can write it as a linear combination of matrices of Gaunt coefficients, with the elements

of a as coefficients.

$$T_L = \sum_{l=0}^{L} \sum_{m=-l}^{l} a_{lm} G_{lm} \tag{4.16}$$

where $G_{lm}(l_1(l_1{+}1){+}m_1, l_2(l_2{+}1){+}m_2) = G(ll_2l_1; mm_2m_1)$. Note that $G(ll_2l_1; mm_2m_1)$ is zero unless $m = m_1 - m_2$. To speed up computation, the matrices G_{lm} are precomputed and stored. Since SDP solvers usually deal only with real valued problems, we use real versions of spherical harmonics instead of the normal complex versions, as described in [Homeier and Steinborn, 1996]. The flowchart 4.2 describes the whole object recognition algorithm.

The SDP is solved in MATLAB 6.5 using the SDPT3 [Tutuncu et al., 2003] package. Since implementing it directly in SDPT3 is difficult, YALMIP [Löfberg, 2004] is used for formulating the problem. This is a problem translator that can convert the problem description in its format to that of a wide variety of SDP solvers available for MATLAB. SDPT3 uses a polynomial time predictor-corrector primal-dual infeasible path following algorithm to solve SDP and SOCP problems, and is one of the fastest solvers available for small to medium scale problems. Table 4.1 compares the times for recovering lighting from an image, using our algorithm (SDP) and the Basri and Jacobs [Basri and Jacobs, 2003] algorithm (Delta), as the number of harmonics used increases. The computer used was a 2.66GHz Pentium 4 with 512MB RAM. [Basri and Jacobs, 2003] use a non-negative combination of delta functions to represent lighting.

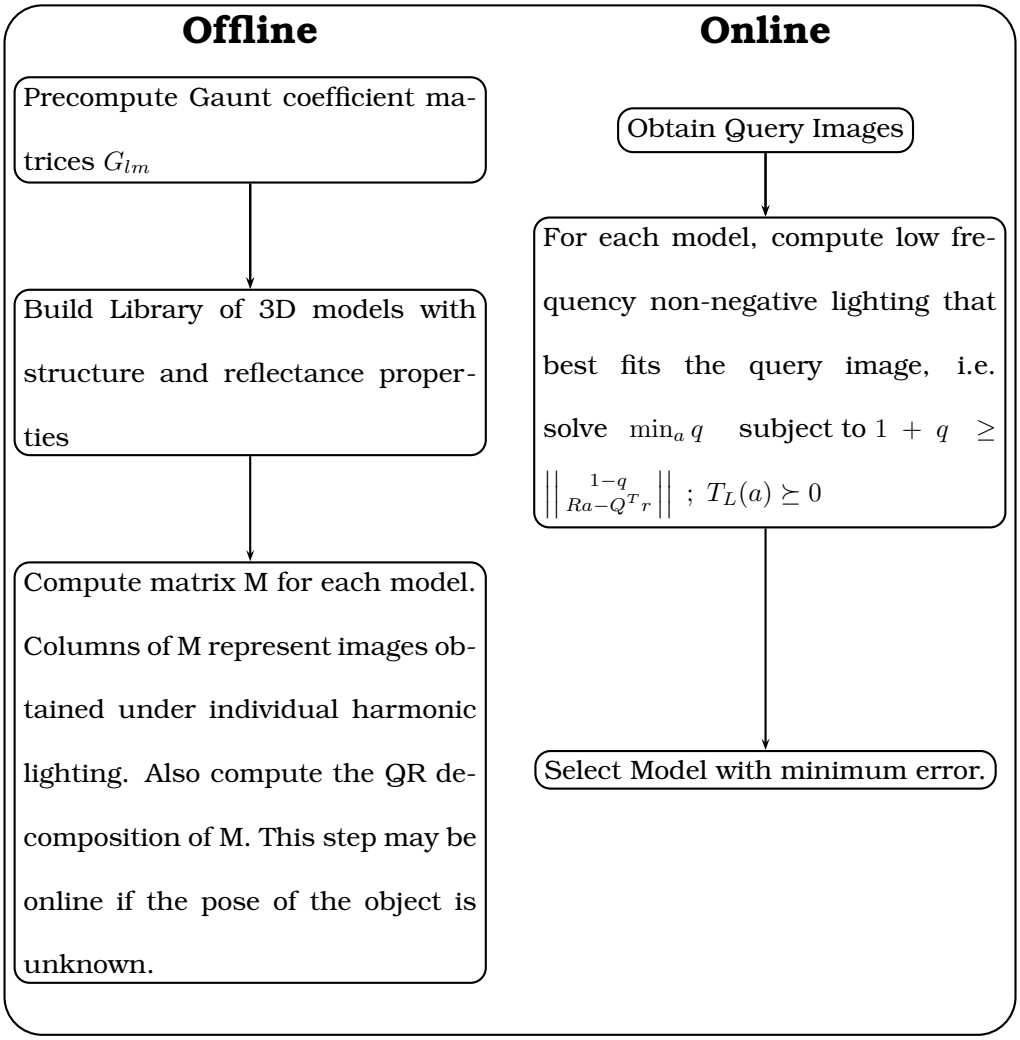


Figure 4.2: Specular Object Recognition Algorithm

As the number of delta functions increases, we find that this method produces the same solution as SDP. In this comparison, the number of delta functions is set to obtain less than 1% error. The image size used was 26×51. Increasing image size just adds the same small time (for the QR decomposition) to both methods. From the table, we see that the time for SDP increases more slowly than that of Delta, and SDP overtakes Delta at around $L = 6$. In our experiments, we have found using $L = 10$ to be sufficiently accurate for a lot of common specular objects. In this case, our method is 35 times faster than Delta, while being exact as well.

Max L	Number of Harmonics $(L+1)^2$	SDP time (s)	Number of Delta functions	Delta time (s)
2	9	0.38	32	0.03
4	25	0.61	1922	0.52
6	49	1.41	1922	1.41
8	81	1.92	1922	4.28
10	121	3.55	30722	126

Table 4.1: Speed comparison of SDP and Delta function method [Basri and Jacobs, 2003].

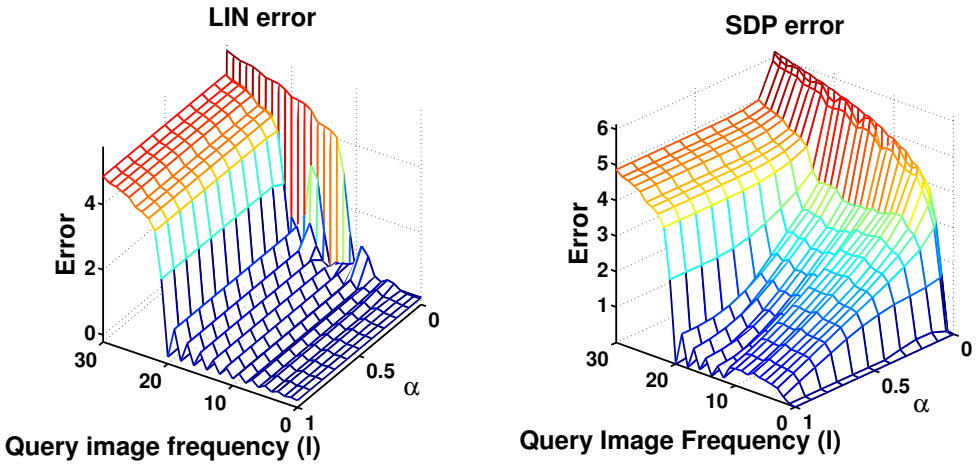
4.5.2 Experiments on Synthetic Images

In these experiments, we evaluate the effect of using the non-negativity constraint (SDP). If the constraint is not imposed, the problem is reduced to simply solving a system of linear equations (LIN).

4.5.2.1 Variation of error with model specularity and query image frequency

In this experiment, we investigate the effect of the frequency content of the query image and the specularity of the object on the importance of the non-negativity constraint. The model is a varying linear combination of a mirror and a uniform Lambertian albedo (α is the proportion of mirror). The query images are composed of individual harmonics (Y_{l0} for $l = 1, \ldots, 30$). These elementary query images will enable us to predict SDP's behavior qualitatively on normal images, which are a linear combination of individual harmonics. The optimization procedure uses spherical harmonics up to degree $L = 10$. Mirror reflection causes image harmonics of order up to double that of incident light harmonics (i.e. up to order 20).

Since the images are not produced by any object, we don't expect any model to have zero error. The magnitude of error will give us an idea of how well the two methods can avoid choosing the wrong model : a larger error means that it is more difficult to fool. The results are shown



(a) (b)

Figure 4.3: Error vs query image frequency and model specularity: (a) Non-negativity not imposed (LIN), (b) Non-negativity imposed (SDP)

in figure 4.3. Firstly, note that SDP has higher error than LIN (which is almost zero) for $L \leq 20$. This is the range of image frequencies that are modeled by the algorithm and the use of SDP should reduce recognition errors here. However, SDP error decreases as the model becomes more specular, and hence the advantage of using SDP decreases. For mirrors and almost-mirrors, using SDP is not likely to help significantly in recognition. Also, we can use the spherical harmonic content of the image to guide us in choosing the number of harmonics required to represent lighting. For example, if most of the harmonic content of the image is of order less than 20, $L = 10$ in the recognition algorithm should suffice.

4.5.2.2 Fooling LIN

We can use the conclusions drawn from the previous experiment to construct synthetic examples that will clearly show that it is possible for LIN to make a mistake between two objects. Take a sphere with uniform albedo. Obtain a non-negative image from this sphere using low frequency lighting that is negative at some places. Use this image as the albedo of a second sphere. Under low frequency lighting, LIN cannot distinguish between these two objects but SDP can. The example is shown in figure 4.1.

4.5.3 Experiments on Real Images

Experiments were performed on two real objects to support the results of the synthetic experiments. The first object was a shiny rubber ball, chosen because it was easy to construct its structural model. The second object was a painted ceramic salt shaker. In both these experiments, it is assumed that the object can be fairly well represented by the mirror + Lambertian model, and that α is constant for the whole surface. These assumptions are not necessary for our method, but they make model construction easier. The first experiment shows that SDP is more robust to noisy models than LIN or a method that simply ignores specularity. First we describe the procedure used for obtaining the surface normals, albedo and α of the surface.

4.5.3.1 Reflectance Model Construction

The objects used in the experiment were either spherical (ball) or cylindrically symmetric (salt shaker) to enable obtaining surface normals from silhouette images. The full procedure for model construction was as follows :

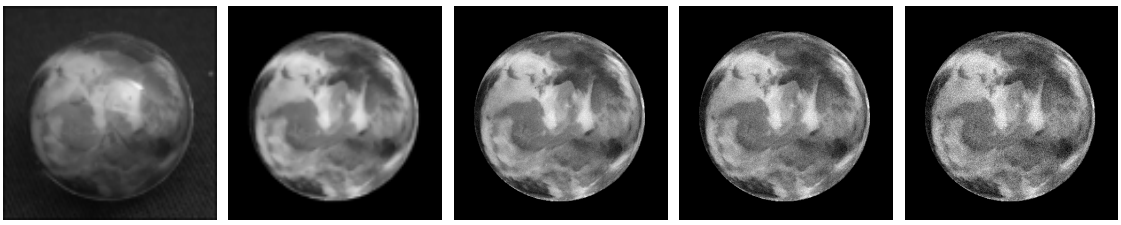
Surface Normals : First, a silhouette image of the object was taken, by strongly illuminating the background of the object, while keeping the light incident on the object itself to a minimum. On appropriate thresholding, we get a silhouette image. Erroneous pixels in the silhouette image were reduced using a morphological closing operation. The object outline was obtained using the image gradient. This was further smoothed using a Gaussian. For the ball, the center and radius were estimated by fitting a circle to the object outline. These were used to obtain a 3D model and hence surface normals of the sphere. For cylindrically symmetric objects, the axis of symmetry of the object was estimated by fitting a straight line through the outline points. The object outline and the axis of symmetry together gave the 3D structure and hence surface normals of the cylindrically symmetric object.

Albedo and α : The object was illuminated by a point source of light and its image was captured. To enable reflectance measurement at specular points, 3 images were taken with different exposure times. The

first exposure time was set very small (around $\frac{1}{100}$th–$\frac{1}{20}$th of a second) so that the specularity caused very little saturation. The second exposure time was set to the largest value so that the Lambertian reflection did not cause saturation (around 10–30 seconds). Finally, the third exposure was set to a suitable intermediate value (around 1 second). 6 such image sets using different directions of the point light source were obtained, taking care to keep the source intensity constant. Next, a single high dynamic range image was constructed from each image set by combining unsaturated pixels from each image, appropriately scaled by the exposure ratio. The region that was not dark in the shortest exposure image was marked as specular. The direction of incident light was calculated using the position of the center of the specular region. The Lambertian region reflectance, corrected for the $\cos(\theta)$ (θ = angle between surface normal and incident light direction) factor, is proportional to the albedo. For each image set, this is obtained only in the non-specular regions that was illuminated by the points source. To obtain it everywhere on the surface, we calculate the median of the values obtained from the 6 image sets. Since the constant of proportionality doesn't affect our computations, and we simply normalize the obtained albedo by the maximum value. Next, we need an estimate of α. This is the ratio of the specular to total (specular + Lambertian reflection, if the Lambertian albedo was 1) reflection at a point. An estimate of the

specular reflection at a point is the sum of grayscale values in the specular area (since lighting is by a point source). Now we need the total Lambertian reflection from this point in all directions, assuming albedo 1. For a sphere, this is simply the sum of the Lambertian reflectance at all points of the illuminated hemisphere, normalized by the albedo. For a cylindrically symmetric object, this has to be estimated using the range of directions of the surface normals that are present. The median of the ratio of specular to total reflection from the 6 image sets gives an estimate of α.

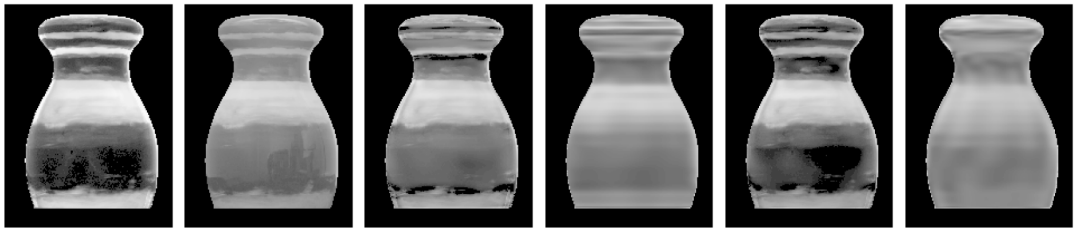
4.5.3.2 Shiny rubber ball



(a) (b) (c) (d) (e)

Figure 4.4: Shiny Rubber Ball: Left to Right: (a) Query Image (b) Measured Albedo (c),(d),(e) Albedo with 8.3%, 15.4% and 24.5% noise levels that fooled LIN, LAMB and SDP respectively.

The experiment consisted of comparing the error difference when lighting is recovered by the correct model, and when it is recovered by a uniform albedo model. The albedo and α for the ball were measured in a separate experiment. A value of $\alpha = 0.04$ was obtained. Next, we



(a) (b) (c) (d) (e) (f)

Figure 4.5: Ceramic shaker (LIN vs SDP): Left to Right: (a) Measured Albedo. (b) Query Image. (c) Best image using correct model and LIN (Error = 8.0%). (d) Best image using uniform model and LIN (Error = 8.7%). (e) Best image using correct model and SDP (Error = 10.3%). (f) Best image using uniform model and SDP (Error = 11.8%). SDP has a higher error difference than LIN between correct and wrong models, and so should be harder to fool.

repeated the experiment using noisy versions of the albedo, to find out which method gets confused first. For comparison, the same experiments were also done assuming a Lambertian model (LAMB), not using the non-negativity constraint and only using a 9D subspace ($L = 2$). The results are shown in table 4.2 and the corresponding images are shown in figure 4.4. The error difference with SDP is larger than that with LIN or LAMB. Since it is a specular object, LAMB has a much higher error than LIN or SDP, even with the correct model. A more noisy model is needed to confuse SDP, as compared to LIN or LAMB.

Method	LIN	SDP	LAMB
Correct Model Error	6.99	10.34	19.48
Uniform Model Error	9.12	19.18	22.26
Noise needed to fool method	8.3%	24.5%	15.4%

Table 4.2: RMS error obtained when matching the query image to the correct model and a uniform albedo model, using various methods. Gaussian noise (with σ as a percentage of correct albedo standard deviation) is added to the model albedo till the method gives the same error as that for the uniform model. LIN and LAMB are fooled more easily. Also, since it is a specular object, LAMB has a large error even for the correct albedo.

4.5.3.3 Ceramic shaker

The albedo pattern and α of the shaker were obtained exactly as that of the ball. The measured value of α was 0.0031. Although this does not seem much, the shaker was specular enough so that the entire room could be seen in it under normal room lighting. The 3D model of the shaker was also obtained using its cylindrical symmetry. A query image was obtained by using almost uniform lighting (to give a low frequency image). The errors obtained when lighting recovery was attempted using LIN and SDP for a uniform model, as well as the measured model are shown in figure 4.5, along with the generated images. We can see that SDP produces an error difference between the correct and uniform (incorrect) models that

is larger than that produced by LIN. We can expect that in this case too, a noisy model will fool LIN more easily than SDP.

4.6 Conclusion and Future Work

We have introduced a new method for enforcing the non-negativity constraint of light in lighting recovery and object recognition. The method is based on the extension of Szego's eigenvalue distribution theorem to spherical harmonics. It is exact and faster than the previous method. From the experiments on synthetic as well as real data, it is clear that the non-negativity constraint is indeed helpful in recognition. The SDP method enables better discrimination between the correct and incorrect models, especially in the presence of noise.

The non-negativity constraint enables better recognition by reducing the space of lighting conditions that are possible. We would like to theoretically quantify this reduction in the space of images. Also, we would like to apply this method to various other problems, like environment map creation in computer graphics.

Appendix

4.A Spherical Harmonics

We will now describe spherical harmonics and state some of their useful properties; first with respect to the more common complex case, and then for their real version.

4.A.1 Complex Spherical Harmonics

The *Surface Spherical Harmonics* [Groemer, 1996] are a set of orthonormal basis functions for the set of all functions $f(u)$ defined on the surface of the sphere S^2, similar to Fourier basis functions on the Circle. They are denoted by Y_{lm}, with $l = 0, 1, 2, \ldots$ and $-l \leq m \leq l$.

$$Y_{lm}(u) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} P_{lm}(\cos\theta) e^{im\phi}$$
$$\theta \in [0, \pi], \quad \phi \in [-pi, pi], \quad u = (\theta, \phi) \tag{4.17}$$

where P_{lm} are the *Associated Legendre Functions*, defined by Rodriguez Formula as:

$$P_{lm}(z) = \frac{(1-z^2)^{m/2}}{2^l l!}\frac{d^{l+m}}{dz^{l+m}}(z^2-1)^l$$

A useful relation of spherical harmonics is :

$$\overline{Y}_{lm} = (-1)^m Y_{l,-m} \tag{4.18}$$

A function $f(u)$ can be expanded in terms of spherical harmonic basis functions, analogous to a Fourier Series expansion of $f(\theta)$ on the circle S.

$$f(u) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_{lm} Y_{lm}(u) \tag{4.19}$$

The spherical harmonic coefficients f_{lm} can be computed as

$$f_{lm} = \int_{S^2} f(u) \overline{Y}_{lm}(u) d\sigma(u)$$
$$d\sigma(u) = \sin\theta d\theta d\phi \tag{4.20}$$

The *Clebsch-Gordon (CG) coefficients* $C(l_1 l_2 l_3; m_1 m_2 m_3)$ [Rose, 1957] are real numbers which appear in many relations involving spherical harmonics. They are zero unless all of these conditions are satisfied :

1. $m_1 + m_2 = m_3$
2. l_1, l_2 and l_3 satisfy a triangle condition $\Delta(l_1 l_2 l_3)$: $\quad l_i \leq l_j + l_k \quad \forall\, i, j, k = 1, 2, 3$
3. $|m_1| \leq l_1, \quad |m_2| \leq l_2, \quad |m_3| \leq l_3$

CG coefficients satisfy the following symmetry properties [Rose, 1957]:

$$C(l_1 l_2 l_3; m_1 m_2 m_3) = (-1)^{l_1+l_2-l_3} C(l_1 l_2 l_3; -m_1, -m_2, -m_3) \tag{4.21}$$
$$= (-1)^{l_1+l_2-l_3} C(l_2 l_1 l_3; m_2 m_1 m_3) \tag{4.22}$$
$$= (-1)^{l_1-m_1} \left(\frac{2l_3+1}{2l_2+1}\right)^{\frac{1}{2}} C(l_1 l_3 l_2; m1, -m3, -m2) \tag{4.23}$$

The integral of the product of three spherical harmonics, called the *Gaunt Coefficient* or coupling coefficient, appears in a lot of applications.

$$\begin{aligned} G(l_1 l_2 l_3; m_1 m_2 m_3) &:= \int_{S^2} Y_{l_1 m_1} Y_{l_2 m_2} \overline{Y}_{l_3 m_3} d\sigma(u) \\ &= \sqrt{\frac{(2l_1+1)(2l_2+1)}{4\pi(2l_3+1)}} C(l_1 l_2 l_3; m_1 m_2 m_3) C(l_1 l_2 l_3; 000) \end{aligned} \tag{4.24}$$

Note that both CG Coefficients and Gaunt Coefficients are real numbers.

4.A.2 Real Spherical Harmonics

For a real function, representation in terms of complex spherical harmonics is redundant due to the relation (4.18). A more efficient representation is in terms of real spherical harmonics [Homeier and Steinborn, 1996], defined as :

$$X_{lm}(u) = \begin{cases} \sqrt{2}\Re(Y_{l|m|}(u)) & \text{for } m > 0 \\ Y_{l0}(u) & \text{for } m = 0 \\ \sqrt{2}\Im(Y_{l|m|}(u)) & \text{for } m < 0 \end{cases} \tag{4.25}$$

This choice of real spherical harmonics conserves the total energy in a complete set of harmonics of a particular order (l). We can also treat this as an orthonormal transformation :

$$X_{l\mu}(u) = \sum_m U^{\mu}_{lm} Y_{lm}(u)$$

The matrix U_l of size $(2l+1) \times (2l+1)$ has the following form :

$$U_l = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & & & & & & (-1)^l \\ & \ddots & & & & & \\ & & 1 & & -1 & & \\ & & & \sqrt{2} & & & \\ & & -i & & -i & & \\ & & & & & \ddots & \\ -i & & & & & & (-1)^l i \end{bmatrix}$$

Next, we can calculate the real Gaunt (R-Gaunt) coefficients [Homeier and Steinborn, 1996], or the coupling coefficients for the real spherical harmonics. These can be defined as :

$$G_R(l_1 l_2 l_3; m_1 m_2 m_3) = \int_{S^2} X_{l_1 m_1}(u) X_{l_2 m_2}(u) X_{l_3 m_3}(u) du \tag{4.26}$$

It is immediately clear that the R-Gaunt coefficients are invariant to any permutation of the $(l_i m_i)$ pairs. Hence, without loss of generality, we can assume that $|m_1| \geq |m_2| \geq |m_3|$. This choice makes their calculation from Gaunt coefficients much simpler. Using the definitions of real spherical

harmonics, we can calculate R-Gaunt coefficients as in equation (4.27).

$$G_R(l_1l_2l_3;m_1m_2m_3) = \begin{cases} 2G(l_1l_2l_3;m_2+m_3m_2m_3)\Re[\overline{U}^{m_1}_{l_1m_2+m_3}U^{m_2}_{l_2m_2}U^{m_3}_{l_3m_3}] + & \\ 2G(l_1l_2l_3;m_2-m_3m_2-m_3)\Re[\overline{U}^{m_1}_{l_1m_2-m_3}U^{m_2}_{l_2m_2}U^{m_3}_{l_3-m_3}] & \text{all } m_i \neq 0 \\ 2G(l_1l_2l_3;m_2m_20)\Re[\overline{U}^{m_1}_{l_1m_2}U^{m_2}_{l_2m_2}] & m_3 = 0 \\ \delta_{m_10}G(l_1l_2l_3;000) & m_2 = m_3 = 0 \end{cases} \tag{4.27}$$

4.B Semidefinite Programming

The basic problem of semidefinite programming [Todd, 2001] in *primal standard form* can be stated as :

$$\begin{aligned} \min_X \quad & C \bullet X \\ \text{subject to} \quad & A_i \bullet X = b_i, \quad i = 1,\ldots,m. \\ \text{and} \quad & X \succeq 0 \end{aligned} \tag{4.28}$$

Here $X \in S\mathbb{R}^{n\times n}$ (the space of real symmetric matrices of size $n \times n$) is the variable and $A_i \in S\mathbb{R}^{n\times n}, C \in SR\mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^m$ are given. $A \succ (\succeq) B$ means that $A - B$ is a positive (semi)definite matrix.

The $\bullet$ operator is the inner product operator of two matrices, defined by $A \bullet B = \mathrm{tr}\,(A^T B)$. The associated norm is the Frobenius norm $\|A\|_F = (A \bullet A)^{\frac{1}{2}}$.

A more convenient formulation of SDP is the *dual standard form* :

$$\begin{aligned} \max_{y,S} \quad & b^T y \\ \text{subject to} \quad & \sum_{i=1}^{m} y_i A_i + S = C \\ \text{and} \quad & S \succeq 0 \end{aligned} \tag{4.29}$$

where $y \in \mathbb{R}^m$ and $S \in S\mathbb{R}^{n \times n}$ are the variables and. This is a convenient formulation because we can get rid of the slack variable S and write the problem as :

$$\begin{aligned} \max_{y,S} \quad & b^T y \\ \text{subject to} \quad & \sum_{i=1}^{m} y_i A_i \preceq C \end{aligned} \tag{4.30}$$

This is the Linear Matrix Inequality form, which occurs commonly in applications. Note that our constraint $T_L(a) \succeq 0$ is also of this form, with $C = 0$.

A closely related problem is *Second Order Cone Programming* (SOCP). Here, the constraint, instead of positive definiteness of a matrix, is $t \geq \|y\|$, where t is a scalar and $y \in \mathbb{R}^n$. This makes the vector $\binom{t}{y}$ lie inside a second order or Lorentz cone in the space $\mathbb{R}^{n+1}$. This is a useful formulation because algorithms for solving SOCP problems are faster than those for SDP problems. The following interesting property enables the conversion of some SDP problems into SOCP ones.

Theorem 11 (Schur Complement)**.** *Suppose* $U = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$ *with* A *and* C *symmetric and* $A \succ 0$*. Then,*

$$U \succ 0 \ (\succeq 0) \quad \textit{iff} \quad S = C - B^T A^{-1} B \succ 0 \ (\succeq 0)$$

The matrix S is called the *Schur complement* of A. For a proof, see [Todd, 2001]. Using this theorem, we can see that

$$\begin{bmatrix} tI & y \\ y^T & t \end{bmatrix} \succeq 0 \text{ is equivalent to } t \geq \|y\|$$

4.C Theorem of Carathéodory

This is an important theorem in the study of the Trigonometric Moment Problem.

Theorem 12. *[Grenander and Szego, 1958](Section 4.1) Let* $c_1, c_2, \ldots, c_n$ *be given complex constants not all zero,* $n > 1$*. There exists an integer* m*,* $1 \leq m \leq n$*, and certain real constants* $\rho_p, \theta_p;\ p = 1, 2, \ldots, m$*, such that* $\rho_p > 0$*,* $e^{i\theta_p} \neq e^{i\theta_q}$ *if* $p \neq q$*, and*

$$c_\nu = \sum_{p=1}^{m} \rho_p \, e^{i\nu\theta_p} \tag{4.31}$$

The integer m *and the constants* ρ_p *and* $e^{i\theta_p}$ *are uniquely determined.*

If we define $c_{-\nu} := \bar{c}_\nu$ and c_0 is such that the Toeplitz matrix formed from c_ν $\{\nu = -n, -n+1, \ldots, n\}$ is positive semidefinite, then the equation (4.31) is valid for all $\nu = -n, -n+1, \ldots, n$.

Bibliography

Alexandr Andoni and Piotr Indyk. Near–optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008. 30

R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2): 218–233, 2003. vii, 1, 9, 103, 106, 107, 126, 128, 130

Peter N. Belhumeur and David J. Kriegman. What is the set of images of an object under all possible illumination conditions ? *International Journal of Computer Vision*, 28:245–260, 1998. 107

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape context. *IEEE Transactions on PAMI*, 24(4):509–522, Apr 2002. 4, 11, 13

Dimitri P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, 2003. 43

Philip Bille. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1-3):217–239, 2005. ISSN 0304-3975. 96, 97

Hamish Carr, Jack Snoeyink, and Ulrike Axen. Computing contour trees in all dimensions. In *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms*, pages 918–926, 2000. 7, 90, 91

C. Chefd'hotel and G. Bousquet. Intensity-based image registration using EMD. In *Medical Imaging 2007: Image Proc. Proc. of the SPIE*, volume 6512, Mar. 2007. 4, 5, 14, 64

Hansen F. Chen, Peter N. Belhumeur, and David W. Jacobs. In search of illumination invariants. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 01, page 1254, 2000. 94

R. R Coifman and D. L Donoho. Translation-invariant de-noising. *Lecture notes in statistics*, 103:125–150, 1995. 22, 47, 48

Ingrid Daubechies and Wim Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4:247–269, May 1998. 52

L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 1998. 114

W. Gangbo and R. McCann. The geometry of optimal transportation. *Acta Mathematica*, 177:113–161, 1996. 65

K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *IEEE ICCV*, pages 1458–1465, 2005. 20

Kristen Grauman and Trevor Darrell. Fast contour matching using approximate earth mover's distance. In *IEEE Conference on CVPR*, volume 01, pages 220–227, 2004. 5, 14

K.D. Gremban and K. Ikeuchi. Planning multiple observations for specular object recognition. In *IEEE Conference on Robotics and Automation*, volume 2, pages 599–604, May 1993. 108

Ulf Grenander and Gabor Szego. *Toeplitz Forms and their Applications*. University of California Press, 1958. 10, 105, 111, 112, 114, 116, 122, 146

H. Groemer. *Geometric Applications of Fourier Series and Spherical Harmonics*. Cambridge University Press, 1996. 140

Kevin Guittet. Extended Kantorovich norms : a tool for optimization. Technical Report 4402, INRIA, March 2002. 33

Steven Haker, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent. Optimal mass transport for registration and warping. *International Journal of Computer Vision*, 60:225–240, December 2004. 3, 5, 14, 64

P. Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 995–999, 1994. 106

L.G. Hanin. An extension of the kantorovich norm. *Contemporary Mathematics*, 226:113–130, 1997. 17, 21, 36, 38

Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212, 2001. 86, 101

A.S. Holmes, C.J. Rose, and C.J. Taylor. Transforming pixel signatures into an improved metric space. *Image and Vision Computing*, 20(9): 701–707(7), August 2002a. 5, 14, 21, 41

A.S. Holmes, C.J. Rose, and C.J. Taylor. Measuring similarity between pixel signatures. *Image and Vision Computing*, 20(4):239–248, April 2002b. 5, 7, 14, 21, 41

Herbert H.H. Homeier and E. Otto Steinborn. Some properties of the coupling coefficients of real spherical harmonics and their relation to gaunt coefficients. *Journal of Molecular Structure (Theochem)*, 368:31–37, 1996. 120, 128, 142, 143

J. E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 172–184, 1974. 96

Piotr Indyk and Nitin Thaper. Fast image retrieval via embeddings. In *3rd International Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003. 20, 57, 60

Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on PAMI*, 21(5):433–449, 1999. 4, 11

Bernhard Korte and Jens Vygen. *Combinatorial optimization: Theory and Algorithms*. Springer, 2000. 18

Erwin Kreyszig. *Introduction to Differential Geometry and Riemannian Geometry*. University of Toronto Press, 1968. 114

In So Kweon and Takeo Kanade. Extracting topographic terrain features from elevation maps. *Journal of Computer Vision, Graphics and Image Processing: Image Understanding*, 59(2):171–182, March 1994. 84, 87

S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. *IEEE Transactions on PAMI*, 27(8):1265–1278, August 2005. 4, 11

Elizaveta Levina and Peter Bickel. The earth movers distance is the mallows distance: Some insights from statistics. In *IEEE ICCV*, pages 251–256, 2001. 21

Haibin Ling and D.W. Jacobs. Deformation invariant image matching. In *Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1466–1473 Vol. 2, 2005. 7, 83, 85, 92

Haibin Ling and Kazunori Okada. Diffusion distance for histogram comparison. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–253, 2006a. 20

Haibin Ling and Kazunori Okada. An efficient earth movers distance algorithm for robust histogram comparison. *IEEE Transactions on PAMI*, 29(5):840–853, May 2006b. 18

J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL http://control.ee.ethz.ch/~joloef/yalmip.php. 128

David Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 4, 11, 12

Ste'phane Mallat. *A wavelet tour of signal processing*. Academic Press, second edition, 1998. 31, 50

J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393, 2002. 1, 85

Yves Meyer. *Wavelets and Operators, Vol 1*. Cambridge university press, 1992. 17, 22, 26, 27, 67

Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 27(10):1615–1630, 2005. ISSN 0162-8828. 2, 7, 85

Stephen P. Morse. A mathematical model for the analysis of contour-line data. *Journal of the ACM*, 15(2):205–220, 1968. ISSN 0004-5411. 84, 86

H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14(1):5–25, 1995. 106

Kate Okikiolu. The analogue of the strong szego limit theorem on the 2 and 3-dimensional spheres. *Journal of the American Mathematical Society*, 9(2):345–372, April 1996. 114, 115, 117

Margarita Osadchy, David Jacobs, and Ravi Ramamoorthi. Using specularities for recognition. In *Proceedings of IEEE International Conference on Computer Vision*, 2003. 108

Valerio Pascucci. On the topology of the level sets of a scalar field. In *Abstracts of the 13th Canadian Conference on Computational Geometry*, pages 141–144, 2001. 88

S. T. Rachev and L. Rüschendorf. *Mass Transportation Problems, Vol 1: Theory*. Springer, 1998. 15, 21, 23

Ravi Ramamoorthi. *A Signal-Processing Framework for Forward and Inverse Rendering*. PhD thesis, Stanford University, Aug 2002. 104, 108

Ravi Ramamoorthi and Pat Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *Journal of Optical Society of America*, 18(10): 2448–2459, 2001. 106

G. Reeb. Sur les points singuliers d'une forme de pfaff completement intergrable ou d'une fonction numerique [on the singular points of a complete integral pfaff form or of a numerical function]. *Comptes Rendus Acad. Science Paris*, 222:847–849, 1946. 86

M. E. Rose. *Elementary Theory of Angular Momentum*. John Wiley & Sons, 1957. 120, 141

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, Nov 2000. 4, 5, 11, 14, 18, 19, 34

K. Sato, K. Ikeuchi, and T. Kanade. Model based recognition of specular objects using sensor models. In *Automated CAD-Based Vision, 1991., Workshop on Directions in*, pages 2–10, 2–3 June 1991. 108

K. K. Thornber and D. W. Jacobs. roadened, specular reflection and linear subspaces. Technical report, NEC Research Institute Inc., Princeton, NJ. 103

E. C. Titchmarsh. *Introduction to the theory of Fourier Integrals*. Oxford University press, 2nd edition, 1948. 31

M.J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001. 126, 144, 146

R.H. Tutuncu, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming*, 95(2):189–217, Feb. 2003. 128

Marc J. van Kreveld, Rene van Oostrum, Chandrajit L. Bajaj, Valerio Pascucci, and Daniel Schikore. Contour trees and small seed sets for isosurface traversal. In *Proceedings of the 13th ACM Symposium on Computational Geometry*, pages 212–220, 1997. 86

A. Vedaldi and S. Soatto. Features for recognition: Viewpoint invariance for non-planar scenes. In *Proceedings of the International Conference on Computer Vision, Beijing, China*, 2005. 86

Todd L. Veldhuizen. Arrays in blitz++. In *Proceedings of the 2nd International Scientific Computing in Object-Oriented Parallel Environments (ISCOPE'98)*, Lecture Notes in Computer Science. Springer-Verlag, 1998. URL http://www.oonumerics.org/blitz. 80

James Z. Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on PAMI*, 23(9):947–963, 2001. 13, 57

M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histograms,. *Computer Vision, Graphics and Image Processing*, 32:328–336, December 1985. 18

A. Yuille, D. Snow, R. Epstein, and P. Belhumeur. Determining generative models of objects under varying illumination: Shape and albedo from multiple images using svd and integrability. *International Journal of Computer Vision*, 35(3):203–222, 1999. 106

K. Zhang. A constrained edit distance between unordered labeled trees. *Algorithmica*, 15(6):205–222, 1996. 84, 97, 99

Lei Zhang and D. Samaras. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):351–363, March 2006. 1

Xiao-Ping Zhang, Li-Sheng Tian, and Ying-Ning Peng. From the wavelet series to the discrete wavelet transform – The initialization. *IEEE Trans. on signal proc.*, 44(1):129–133, 1996. 50, 81

Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003. 4

Zoran Zivkovic and Ben Kröse. An em-like algorithm for color-histogram-based object tracking. In *Proc. IEEE Conference Computer Vision Pattern Recognition*, 2004. 11, 13

A. Zygmund. *Trigonometric Series*, volume 1. Cambridge University Press, 2002. 31