Learning Random Field Models for Computer Vision

by Yunpeng Li

LEARNING RANDOM FIELD MODELS FOR COMPUTER VISION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Yunpeng Li

August 2010

LEARNING RANDOM FIELD MODELS FOR COMPUTER VISION

Yunpeng Li, Ph.D.

Cornell University 2010

Random fields are among the most popular models in computer vision due to their ability to model statistical interdependence between individual variables. Three key issues in the application of random fields to a given problem are (i) defining appropriate graph structures that represent the underlying task, (ii) finding suitable functions over the graph that encode certain preferences, and (iii) performing inference efficiently on the resulting model to obtain a solution. While a large body of recent research has been devoted to the last issue, this thesis will focus on the first two.

We first study them in the context of three well-known low-level vision problems, namely image denoising, stereo vision, and optical flow, and demonstrate the benefit of using more appropriate graph structures and learning more suitable potential functions. Moreover we extend our study to landmark classification, a problem in the high-level vision domain where random field models have rarely been used. We show that higher classification accuracy can be achieved by considering multiple images jointly as a random field instead of regarding them as separate entities.

BIOGRAPHICAL SKETCH

Yunpeng Li received the B.S. degree in computer science with the highest honor from Middlebury College in 2005, after which he joined the Ph.D. program in computer science at Cornell University.

To its readers.

ACKNOWLEDGEMENTS

The quest for a doctoral degree in computer science can be a lengthy, tiring, and distressful undertaking. Therefore I feel especially lucky that mine is not; instead it is pleasant and cheerful most of the time, in spite of all the challenges and setbacks, all owing to the valuable help from so many people, to whom I am deeply indebted.

First and foremost, I want to thank my advisor, Dan Huttenlocher, without whom this dissertation would simply not exist. Dan's a great teacher and mentor whose guidance and advice are the most precious assets in my graduate life. I also want to thank Thorsten Joachims and David Williamson for kindly agreeing to be my special committee members. I had taken several classes with Thorsten and David, where I learnt a lot from their excellent teaching. I am very grateful to Daniel Scharstein and Amy Briggs, with whom I worked closely as an undergraduate. Daniel and Amy had given me a lot of support and advice, without which I could not have had the opportunity to pursue a graduate study at top institution. I want to thank Sing Bing Kang for giving me the chance to work as an intern at Microsoft Research in Redmond. As my mentor there, Sing Bing's rigorous research style and uncompromising pursuit for knowledge was a great inspiration for me. Needless to say that the help from my other two collaborators at MSR, Neel Joshi and Steve Seitz, was also tremendous and crucial for accomplishing my mission there. I want to thank David Crandall and Noah Snavely, who, as more senior members of my research groups, have given me invaluable help on my work. I appreciate their candor and patience with me, and the knowledge and feedback they provided to me are beyond what I could learn in just the classroom.

I am fortunate enough to know Dexter Kozen, A great teacher and friend to all graduate students. Dexter's versatility in sports, music, and many other aspects convinced me that a researcher's life can be much more than just research. I feel lucky to have the support of Beck Stewart, our assistant director of graduate studies, who has always patiently helped me with all the tedious administrative matters. I also want to thank all my fiends. They are a great company that made my graduate life joyful and memorable.

Finally I want to thank my parents, my dearest ones in the world. I am grateful to them for everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Motivation

Computer vision is a fast growing field in computer science, with areas ranging from low-level vision problems such as segmentation and stereo vision to high-level vision problems such as object recognition and scene recognition. Yet, despite the rapid progress in the last few decades, it remains arguably one of the most "unsolved" field in computer science, at least if the relative performance between machines and humans is used as the judging criterion.

Random fields are graphical models that can be used to capture the statistical interdependence between multiple variables. This is very useful for computer vision, especially low-level vision tasks. In low-level vision, labeling is typically performed on a per-pixel basis. This means that there is a variable for every pixel, whose value needs to be inferred. Since the amount of data associated with each pixel is often very limited, there is usually not enough local evidence to label each pixel independently. Thus the ability of random fields to model the relationship between variables and label them jointly is very important in these situations.

There are three key issues in the successful application of random fields to a given problem: defining appropriate graph structures, finding suitable potential functions over the graph, and performing inference efficiently on the resulting model. While a large body of previous research has been devoted to the last issue, namely the inference problem, the first two have received relatively less

attention. This offers an opportunity to improve the success of random fields – by obtaining better models. If we do not have a good model to start with, we are unlikely to achieve high performance no matter how good the optimization techniques are. Thus this thesis will focus on the first two issues, which we call the "modeling aspect" of random fields.

1.2 Learning for Random Fields

Random fields are a popular choice for many low-level vision problems, such as image denoising, stereo, and optical flow. However, these models usually have parameters that are hand set rather than learnt automatically. Indeed there have been relatively few works on learning random fields for low-level vision until recent years (e.g., [77, 78, 81, 97]). Hand tuning of models not only involves substantial human effort, more importantly it limits our understanding of how well a given method generalizes to unseen data and thus will perform in practice. This is illustrated by the recently developed Middlebury optical flow evaluation database [5], where methods that perform best on the classical benchmark sequence tend not to perform as well on new imagery for which they were not hand tuned.

The scenario for high-level vision is quite different, where learning is ubiquitous. Nevertheless, high-level vision tasks, such as object detection and image classification, are usually performed on an instance-by-instance basis without any effort to model the relationship between the instances (e.g., multiple detection windows or images). Consequently, random fields are rarely used in high-level vision.

In this thesis, we shall investigate the learning of random fields for low-level vision problems and explore the potential of the its application in the high-level vision domain.

1.3 Overview of Random Field Models

Random field models were first proposed in the statistics literature to capture the interdependence between variables that are statistically correlated [11, 34]. As graphical models, they represent each variable that needs to be labeled as a vertex, or node, and encode the dependence between variables as edges of the graph. Originally they were conceived in the Bayesian framework [11] but later was extended to include discriminative, non-Bayesian formulations [54]. The former are called Markov random fields (MRF) while the latter are called conditional random fields (CRF).

1.3.1 Markov Random Field (MRF)

Consider a graph consisting of nodes and edges. Let $\mathcal{V}$ be the set of nodes and $\mathcal{E}$ be the set of edges. Also let I denote the observed data and X denote the labeling over the random field. The Markov random field (MRF) models the posterior probability of the labels of the hidden variables (which correspond to the nodes of the graph) as the product of the probability of the observed data conditioned on the labels and the prior probability of the labels independent of the observed data, using the Bayes rule:

$$\Pr(X|I) = \frac{\Pr(I|X)\Pr(X)}{\Pr(I)}. \tag{1.1}$$

Since the *a priori* probability of the data $\Pr(I)$ is a constant, it is usually omitted. Thus:

$$\Pr(X|I) \propto \Pr(I|X)\Pr(X). \tag{1.2}$$

The MRF formulation hence decomposes the posterior into the *likelihood* $\Pr(I|X)$ and the *prior* $\Pr(X)$. This is similar to the hidden Markov model (HMM) [8] that is widely used in speech recognition [76]. However, unlike the HMM, the MRF is defined on an undirected graph and does not require the graph to be acyclic.

One of the most fundamental results in random fields is the Hammersley-Cliffored Theorem [11], which guarantees that the prior $\Pr(X)$ can always factorized into potentials over the maximal cliques of the graph. Hence if we use C to denote the set of maximal cliques, then

$$\Pr(X) = \frac{1}{Z}\prod_{c\in C}\phi_c(X_c) \tag{1.3}$$

where X_c is the labeling over clique c and the functions $\phi_c(\cdot)$ are often referred to as the clique potentials. The factor $1/Z$ is independent of X and ensures that $\Pr(X)$ is a proper probability function, [1] and the constant Z is called the normalization constant or the partition function.

As a generative model, the MRF has certain restrictions due to the Markovian assumptions. In particular, each node is associated with its own piece of observation (data) that is independent from the data of all other nodes, i.e.,

$$\Pr(I|X) = \prod_{v\in\mathcal{V}}\Pr(I_v|X_v) \tag{1.4}$$

where I_v and X_v denote the observation and label at node v respectively. It is easy to see that this formulation does not allow the data to be shared among nodes

[1] If the label space is continuous, $\Pr(X)$ is a probability density function; it it is discrete, $\Pr(X)$ is a probability mass function.

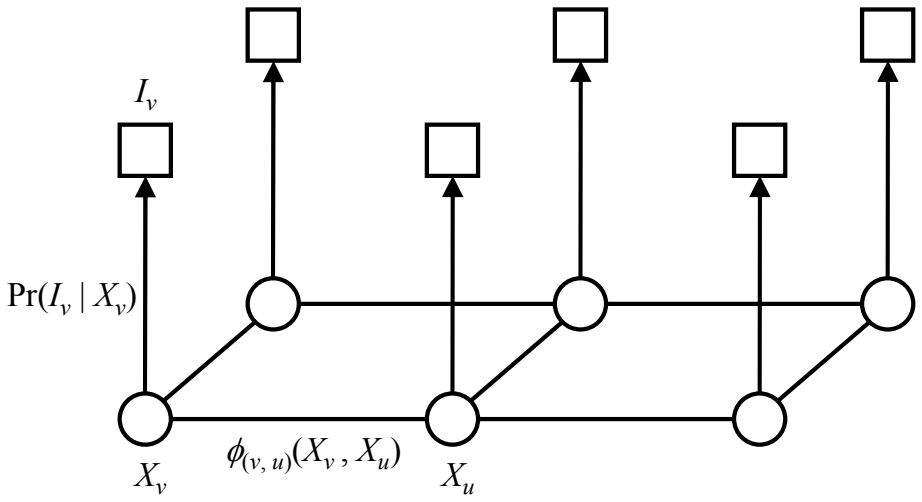


Figure 1.1: An illustrative diagram of an MRF. Circles are variables and squares are observed data. Each variable "generates" its own piece of observation.

(see Figure 1.1). In addition, the prior must depend solely on the labels and must not depend on the data in any way. This is evident from the fact the $\Pr(X)$ does not involve the data I. These restrictions reflect the underlying assumption that the data is generated by the labels of the variables.

1.3.2 Conditional Random Field (CRF)

The conditional random field (CRF), on the other hand, is a discriminative model and does not assume that the data is generated by the labels. Hence it models the posterior $P(X|I)$ directly as a conditional probability without using the Bayes rule. The CRF decomposes the posterior into a *data term* that encourages agreement between the labels and the observed data and a *spatial term* that enforces spatial consistency among the labels (of neighboring nodes). Similar to the likelihood and the prior of an MRF, the data term factorizes over the nodes

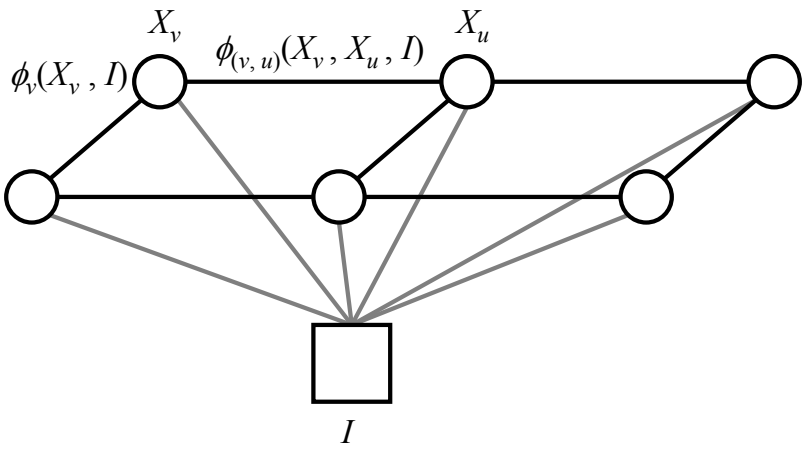


Figure 1.2: An illustrative diagram of a CRF. Circles are variables and the square is observed data, shared among all variables.

and the spatial term factorizes over the cliques of the graph:

$$\Pr(X|I) = \frac{1}{Z} \prod_{v \in \mathcal{V}} \phi_v(X_v, I) \prod_{c \in C} \phi_c(X_c, I). \tag{1.5}$$

In this formulation, however, the observed data can be shared among nodes (for the data term) and the spatial term can depend on data (see Figure 1.2). These properties make CRFs more permissive and more general than MRFs, and a desirable alternative to MRFs in situations where there is not a clear or well-understood generative process and hence the strict MRF formulation may run into difficulties.

The CRF was first proposed in the context of part-of-speech tagging as an alternative to the HMM, in an effect to address the "label bias" suffered by HMM-based approaches [54]. Although originally defined only for acyclic graphs, the CRF has since been extended to loopy graphs and become a popular choice for many computer vision problems.

Due to its similarity to MRF in practice, the terms "MRF" and "CRF" are often used liberally and sometimes interchangeably in the computer vision liter-

ature; and so are "likelihood" and "prior" with "data term" and "spatial term" respectively. We will follow this custom in this thesis.

1.3.3 Relationship to Energy-based Models

Random fields are closely related to energy-based models, when the probability is defined in the exponential family in terms of some energy function E:

$$\Pr(X|I) = \frac{1}{Z} \exp\left(-E(X, I)\right), \tag{1.6}$$

where Z is the normalization constant or partition function. The energy is therefore the negative log-probability, up to a constant due to normalization, i.e.,:

$$E(X, I) = -\log \Pr(X|I) + \log Z. \tag{1.7}$$

Since the normalization constant Z does not depend on the labeling of the random field, finding the labeling that maximizes the probability, namely the maximum *a posteriori* (MAP) solution, is equivalent to finding the one that minimizes the corresponding energy.

Because of the factorization of random fields, the energy also decomposes into a sum of individual energy functions over the nodes and the cliques. Let (again) $\mathcal{V}$ be the set of nodes and $\mathcal{C}$ be the set of cliques of the random field, and let g_v and f_c denote the energy functions associated with node v and clique c respectively. Then the total energy is

$$E(X, I) = \sum_{v \in \mathcal{V}} g_v(X_v, I) \sum_{c \in \mathcal{C}} f_c(X_c, I), \tag{1.8}$$

This is especially useful in computation, since the sum is more convenient to evaluate than the product. Thus in practice MAP inference is almost always

carried out as energy minimization. In computer vision, the "energy" is also commonly referred to as the "cost" and The two words are often used interchangeably.

1.4 Inference

The variables of a random field may belong to either a discrete domain or a continuous domain. For each case, there are different inference techniques available. Below we will review some of these techniques that are commonly used in computer vision.

1.4.1 Discrete Domain

When the variables are discrete, the set of labels is finite and the inference is a combinatorial problem.

Dynamic Programming

In the case where the random field is acyclic, exact inference can be carried out efficiently using dynamic programming based techniques. If the underlying graph has a chain topology, the MAP solution can be found using the Viterbi algorithm [106] and the marginal probability distribution for each node (i.e. variable) can be obtained using the forward-backward algorithm [76]. For both algorithms, the time complexity is linear to the number of nodes (i.e., the length of the chain) and quadratic to the number of possible labels at each node.

Belief Propagation

If the graph is a tree, exact inference can be performed using belief propagation (BP) [72]. There are two versions of BP: max-product BP computes the MAP solution whereas sum-product BP computes the marginal distribution for each node. Both versions have the same time complexity as the Viterbi algorithm and the forward-backward algorithm. Although BP is formulated as a local message passing algorithm, it is in fact a form of implicit dynamic programming when performed in the sequential manner for carrying out exact inference on trees.

For marginal inference (i.e., finding the marginal distribution for each node), the forward-backward algorithm (for chains) and sum-product BP (for trees) essentially employs discrete convolution to compute the forward/backward probabilities and the messages respectively. Hence the time complexity with respect to the number of labels (call it M) can be reduced from $O(M^2)$ to $O(M \log M)$ by using the fast Fourier transform (FFT), which is helpful when M is large. When M is relatively small, however, the computational overhead of FFT could more than offset its benefit and make it slower in practice.

For MAP inference, both the Viterbi algorithm and max-product BP correspond to repeated min-convolutions (in the negative logarithm domain) [27]. For certain subclasses of clique potentials, such as those corresponding to linear or quadratic energy functions (possibly truncated), distance transforms can be employed to reduce time complexity by a factor of $O(M)$ [27]. Unlike FFT, distance transforms have much smaller computational overhead. Although not always applicable in the general, they have been found to be highly useful in many computer vision problems.

Junction Tree

Exact inference becomes much more difficult when the graph contains loops, since dynamic programming techniques do not allow cycles. The junction tree algorithm creates a tree structure on a set of "supernode", where each supernode encapsulates the collective states of a subset of nodes of the original graph. Such a structure is called a junction tree, and the size of the largest supernode (i.e., the number of nodes it contains) minus one is called the width of the junction tree. The minimum width of all possible junction trees of a graph is called the tree width of the graph. Once a junction tree is created, techniques such as BP can again be used for inference. In practice, however, the number of messages that need to be computed at each supernode is exponential to the number of nodes it contains. Hence the complexity of the junction tree algorithm is exponential in the tree width of the graph, which is usually prohibitively large (at least for most computer vision problems). For instance, the m-by-n 4-connected grid has a tree width of $\min(m, n)$. Because of the high complexity, the junction tree algorithm can only be used on very small (and usually artificial) problems that are created to facilitate comparisons between some approximate solution and the optimal solution.

Iterated Conditional Mode

Since exact inference on loopy random fields is usually intractable, much work has focused on effective approximate inference techniques. Once of the earliest of these is the iterated conditional mode (ICM) algorithm, which updates the label of each node to the one with the highest probability conditioned on the labels of its neighbors at each iteration. Although simple and fast, ICM typically

does not produce a high-quality solution and may not converge at all. In fact, it is not guaranteed to increase either the posterior probability of the random field or the marginal probability of any node.

Variational Inference

Another option for approximate inference is the variational method using mean field approximation. In this approach, the interaction between a variable and its neighbors is replaced using the "mean" effect of the its surroundings ("external field"). The "mean field" of each variable is updated according to those of its neighbors, and the whole process iterates. Although the removal of explicit variable interactions makes inference more tractable, the approximation can be too crude sometimes and thus results in poor accuracy.

Loopy Belief Propagation

One of the more promising approaches for approximate inference is loopy belief propagation (LBP), which generalizes BP to loopy graphs [68]. Although BP was originally formulated for exact inference on trees, the forms of its message passing rules does not explicit require this. Hence it is possible to run BP on a loopy graph as an iterative algorithm (i.e., LBP). [2] In practice, LBP works surprisingly well considering the relatively lack of theoretical justification.

As researchers attempted to understand the empirical success of LBP, more sophisticated and arguably better variants with modified message update rules were developed in the process. Generalized belief propagation (GBP) [111] esti-

[2] On a tree, BP needs only to send each message once.

mates the marginals by minimizing the Kikuchi free energy of the random field and has better convergence properties than regular sum-product LBP. GBP is also shown to reduce to regular LBP under specific setting, which can be viewed to minimize the Bethe free energy. Tree-reweighted belief propagation (TRW) [107, 51] was introduced to compute a family of upper bounds on the log partition function based on spanning trees, but is also able to estimate the marginals as well as the MAP solution. For MAP inference, TRW is able to determine the optimality of a solution by examining the strong tree agreement condition [107]. Hence it can provide an optimality guarantee when it is available.

Despite these improved variants, however, the regular LBP remains a useful alternative because of its versatility. When a random field has higher-order clique potentials (i.e., clique potentials of more than two variables), Loopy BP can be easily adapted to operate on the corresponding factor graphs and in this case messages are passed between regular nodes and factor nodes (representing the cliques). On the other hand, the extension of GBP and TRW to such random fields is, though possible in theory, highly non-trivial and very involved in practice. It is worth noting though that the time complexity of message updates grows exponentially with respect to clique size. Thus message passing based algorithms are not feasible for random fields with large cliques in general.

Just like BP on trees, All aforementioned variants of LBP are able to benefit from distance transforms [27] whenever applicable. Moreover message passing can be performed in a hierarchical, coarse-to-fine fashion [27], which leads to faster and more reliable convergence.

Graph Cuts

If a random field consists of binary variables and has only pairwise clique potentials with submodular energies, the optimization can be reformulated as finding the min-cut over a graph [15] and solved using max-flow algorithms. For multi-label problems (where variables have more than two possible values), the α-expansion algorithm [15], which iterates by solving a series of binary sub-problems ("expansion moves"), has proved to be highly effective. However, since the algorithm uses graph cuts as a building block, each binary subproblem must be submodular. This in turn requires the energy functions of the spatial term to be a metric with respect to some ordering of the labels [52], which is not always warranted.

Quadratic pseudo-boolean optimization (QPBO) [14], on the other hand, is capable of optimizing binary energy functions that are not submodular. QPBO is also based on finding a min-cut on a graph, but it creates two nodes for each variable – one for the variable and the other of its complement. This relaxation converts the non-submodular terms of the energy function to submodular terms involving the complements, which can be subsequently solved using graph cuts. Since each variable is represented by two nodes in the constructed graph, however, its value may be inconsistent (i.e., both the variable and its complement may receive the same label after the min cut). If this happens, the variable remains unlabeled in the final output. The proportion of unlabeled variables depends on the extent to which the energy function is non-submodular. If the energy function is in fact submodular, all variables will be labeled by QPBO and the solution is identical in energy to that given by regular graph cuts. For non-submodular energies, QPBO guarantees that the inferred labels of all consistent

variables are part of the optimal solution.

Similar to graph cuts, QPBO can be extended to solve multi-label problems though a series of binary *fusion* moves [57]. Unlike α-expansion, fusion does not impose any explicit restriction on the choice of energy functions; though such choices may affect the quality of the solution.

Recently QPBO have been applied to random fields with higher-order cliques through schematic graph reduction, where each high-order energy is replaced by a set of pairwise energies [45]. Similar to BP, however, the complexity of the reduction process as well as the reduced graph is exponential in clique size. Hence optimizing for random fields with large cliques remains impractical even with these reduction techniques.

Under the special circumstance where the spatial term energy is convex (with respect to some ordering of the labels), a different iterative scheme based on graph cuts can be used to minimize the energy exactly [44]. Unfortunately though, convex spatial terms are not unsuitable for most computer vision problems.

It should be noted that graph cuts based techniques can only be used for MAP-inference, which corresponds to energy minimization, and are not applicable for estimating the marginal probabilities.

LP Relaxation

MAP inference on random fields can also be reformulated into an integer program, and solved (approximately) using linear programming (LP) relaxation

[50]. The algorithm is guaranteed to achieve a 2-approximation of the optimal energy, and the LP dual provides a lower bound on the energy function. However in its formulation it is necessary to create a variable (in the LP) for every possible label of every node and for every possible combination of labels over every clique. This makes it computationally challenging even for pairwise random fields, which has to some extent limited its application in computer vision.

1.4.2 Continuous Domain

Numerical Methods

For random fields with continuous variables, MAP inference can often be performed numerically. Gradient based methods, such as conjugate gradient and iterative reweighted least squares, can be used to search for a local minimum of the energy function. If the energy function is convex, the exact solution can be obtained (up to some small numerical error). Numerical optimization is a field of research in its own right, and a comprehensive review is beyond the scope of this thesis.

Discretization

In addition to numerical methods, one may also have the option to discretize the label space and subsequent use discrete inference techniques. This approach works well for certain problems (e.g., [55, 45]) but may run into difficulty when the number of bins required for adequate discretization is too large to be computationally feasible (e.g., in optical flow).

Semi-discrete Optimization with Fusion

An alternative to discretization for MAP inference is to use some known black box algorithms (not necessarily related to the random field model to be optimized) with varying parameters to generate a collection of "proposal" labelings for the random field. These proposals are then combined through a series of binary fusions using QPBO [57]. This typically produce a final labeling that has lower energy than any of the proposals. The fusion method is especially useful when the energy function is high non-convex with many undesirable local minimums.

Sampling

Final but not the least, sampling remains one of the most important inference techniques, for both discrete and continuous valued random fields. Markov chain Monte Carlo methods, such as Gibbs sampling, can be used to collect a sample from the posterior, and marginal probabilities of each node can be estimated from the sample. However the Markov chain may take a large number of iterations to converge, making the procedure very time consuming. Improving the efficiency of sampling techniques is also an active area of research.

1.5 Learning

The learning task can be regarded as selecting, from a class of (possibly infinitely many) candidate models, a particular model that is best suited to the problem. This usually amounts to estimating the parameters of a (parameterized) model

from a set of labeled training instances.

1.5.1 Maximum Likelihood

Perhaps the most common approach to learning is to seek parameters that maximize the likelihood of the training data. [3] Knowing the value of the likelihood, nevertheless, requires knowing the partition function. In the case where the random field has no cycles (i.e., chains and trees), the log partition function as well as its derivatives with respect to the parameters can be computed efficiently using dynamic programming [54]. Hence a max-likelihood estimate for the parameters is relatively easy to obtain. If the energy function is linear in its parameters, the corresponding learning problem is convex (in the negative log space) and the max-likelihood estimator is exact.

When the random field contains loops, however, the partition function itself is intractable to compute and not even easy to approximate. However the derivatives of the log partition function can be expressed in terms of the marginal probabilities of the cliques, which may be estimated using LBP (on the corresponding factor graph) or sampling [111]. This makes it possible to find an approximate max-likelihood estimate using (approximate) gradient ascent. In practice, however, the gradient obtained though BP tends to be noisy (since the estimated marginals are only approximate) and sampling either requires a very large sample (which can be prohibitively slow) or tends to produce inaccurate results. Although techniques such as contrastive divergence [39] have been developed to improve the efficiency of sampling-based gradient ascent, max-

[3] Here in the context of learning, the word "likelihood" refers to the probability of observing the training data given the model and its parameters. It is not to be confused with the likelihood term (i.e., data term) of a random field.

likelihood learning on loopy random fields remains computational challenging in most situations.

1.5.2 Maximum Margin

An alternative to maximum likelihood is maximum margin. Max-margin classifiers were originally formulated for binary classification problems [105, 47], but were later extended to structured outputs [100, 104]. In the context of random field models, the goal of max-margin learning is to find parameters that make the energy of the ground truth labeling lower than the energy of any other labeling by a margin as large as possible. The margin is usually soft, which means margin violations are allowed at the cost of some penalty terms in the objective function. This is essential for making learning robust against noise (especially outliers) in the training data. The main difficulty in the max-margin formulation of learning is that the number of possible labelings in a structured output space (such as that of a random field) grows exponentially with the number of variables and hence it is impossible to enumerate all of them. The most popular algorithm for learning max-margin classifiers is arguably the structural support vector machine (structural SVM) [104], which addresses this issue by using a cutting plane algorithm to avoid enumerating all possible labelings. The structural SVM proceeds by finding the most violated constraint [104] at each iteration. Finding the most violated constraint, however, is generally at least as hard as inference, which makes it intractable for loopy random fields. Although many approximate inference algorithms (as discussed in Section 1.4) can be used for this purpose, the resulting most violated constraint is not exact hence voiding many of the theoretical guarantees of structural SVM [104]. The

use of approximate most violated constraints for learning random fields was investigated in [29], which shows that the quality of the learnt model depends on the accuracy of the approximate most violated constraint. However, the study focuses on random fields that are complete graphs. Most random fields in computer vision problems, on the other hand, correspond to sparse graphs, and hence may have different characteristics.

Minimum Training Error

Learning can also be cast as simply minimizing the training error (i.e., the error over of training data under some evaluation metric). Assuming that the training data is sufficient and representative for the complexity of model (and hence no overfitting), then lower training error is expected to lead to lower test error (i.e., error on instances unseen during learning). Straightforward as it sounds, this is usually difficult due to the lack of closed form relationship between training error and model parameters.

In certain situations, e.g., when the random field is continuously valued, MAP inference (equivalent to energy minimization) is used, and both the energy and the error functions are continuously differentiable, it may be possible to use implicit differentiation techniques to minimize training error with respect to model parameters via gradient descent [97]. However, the gradient can be noisy due to local minimums of the energy function (unless the it is convex). Moreover such methods need to compute the Hessian of the energy with respect to the variables at some stage, making them computationally challenging for large random fields.

An other option is to use stochastic optimization methods (e.g., [48, 85]) that do not require closed form gradient. Although these class of methods have been popular in engineering, they are relatively unknown in the computer vision literature.

1.5.3 Outline of Thesis

The rest of this thesis is organized as follows. In Chapter 2, we introduce the sparse long-range random field (SLRF) model, a graph structure with long-range connections but only small cliques, and demonstrate its effectiveness in the context of image denoising [62]. We then study the problem of learning the parameters of random field models in Chapter 3, where we formulate a non-parametric CRF model for stereo vision and learn it using the structural support vector machine (structural SVM [104]) [61]. We also revisit the SLRF model and show that long-range connections are beneficial for stereo vision as well. We continue our investigation in Chapter 4, where we present a method for learning a continuous-state MRF for optical flow using stochastic optimization [60]. We show that our approach is highly viable for directly minimizing the training error. In Chapter 5, we apply random field models to high-level vision by applying them to the problem of landmark classification [59]. Our experiments demonstrate that better performance can be obtained by exploiting the hidden relationship between images using random fields. We summarize our work and discuss possible directions of future research in Chapter 6.

CHAPTER 2

SPARSE LONG-RANGE RANDOM FIELD AND ITS APPLICATION TO IMAGE DENOISING

2.1 Introduction

Random fields are among the most common models used in low-level vision problems such as image restoration, segmentation, and stereo. The strength of these models lies in their ability to represent both the interaction between neighboring pixels and the relationship between the observed data values and estimated labels at each pixel. A random field model defines a graph structure with potential functions over the labelings of cliques in this graph. For low-level vision problems the graph generally has a node corresponding to each pixel, edges connecting certain pairs of neighboring pixels, and potentials that encourage neighboring pixels to have similar labels. This chapter focuses on the issue of finding a good graph structure for the problem being modeled.

We propose *sparse long-range random field* (SLRF) models, that represent interactions between distant pixels using sparse edges so as to maintain a fixed clique size. The size of the clique is chosen so as to be appropriate for a particular problem. In image denoising, second-order spatial terms are important for representing intensity change. Thus we use a graph structure that has cliques of size three, as discrete approximations to a second order function require three data values. In this framework the potential functions are defined over fixed-size cliques that have different spatial extents, effectively encoding image structure of a fixed order (defined by the clique size) at multiple scales of observation. This enables such models to produce smooth labelings for highly noisy images

but at the same time allows efficient solution.

In contrast, other recent work using higher-order models and longer-range connections, such as the Field of Experts (FoE) model [77], has large cliques and thus does not support fast optimization. Our main interest is thus in investigating whether simpler models with smaller cliques can produce results comparable to the state-of-the-art achieved with more powerful models, such as FoE, while using much less time. The experiments that we report here, performed on widely used datasets, indicate that this is indeed the case. Not only do we achieve comparable peak signal-to-noise ratio (PSNR) to large-clique methods, our method is also better at avoiding over-smoothing although that is not captured by the PSNR measure. At the same time, our method is over 100 times faster than FoE and at least 10 times faster than other spatial-domain methods that achieve state-of-the-art results.

2.1.1 Related Work

The most widely used graph structure for random field models in low-level vision is a grid where each node is connected to its four immediate neighbors in the horizontal and vertical direction. While this model is simple and convenient to set up and optimize, it suffers from a number of drawbacks. First, as noted above, it can only represent first-order properties of the image, because it is based on pairwise relations between pixels. Second, it can be sensitive to noise. Consider a connected region of n nodes in a 4-connected grid graph. In this case there are only approximately $O(\sqrt{n})$ connections between nodes in the region and those outside the region, because the boundary grows approximately as

the square root of the area. Thus the data term of the n nodes over the region comes to dominate the connections outside the region, especially when robust (i.e. discontinuity preserving) spatial terms are used. For example, in image denoising this can be problematic for high noise levels because good estimates require substantial sized regions. Another way to view this is in terms of the standard deviation of the mean over the region. For concreteness, consider an image with additive Gaussian noise of $\sigma = 25$, and a 5×5 region of the image. The standard deviation of the mean of that region is $\sigma / \sqrt{5 \cdot 5} = 5$. At the same time, the perimeter-to-area ratio of such a neighborhood is only $4 \cdot 5/5^2 = 4/5$, or $1/5$ that of a single pixel. Hence the collective labeling of the group is dominated by its data term, which is subject to a non-trivial standard deviation of 5 in its mean.

The 4-connected grid graph is a special case of graphs that connect a node to all nodes that lie within some distance d. In contrast to our approach, such graphs produce quite dense edges even for moderate values of d. Early work on MRF models in vision, such as [34], used these models but restrict their attention to pairwise clique potentials. However, such pairwise models do not always capture the underlying distribution. For image denoising, in particular, second-order statistics are important, implying a need for cliques of size at least three.

Problems with earlier pairwise random field models have led to higher-order models such the Field of Experts (FoE) [77], where overlapping blocks of pixels are used rather than purely local connections. However, such models are computationally intensive due to their relatively large complete subgraphs. In addition, the learnt priors are also unintuitive, despite recent interpretations as derivative filters [97] and frequency filters [108].

The use of long-range edges have also been studied in the context of texture synthesis [35]. Clique families are chosen using heuristic search based on the strength of interaction, which is evaluated on the training data. However, the model is restricted to pairwise clique potentials. Moreover each model is trained to synthesize a particular type of texture, which usually consists of some characteristic (and often repeating) patterns. Thus it is not well suited to modeling generic structures, such as those of natural scenes.

2.2 Sparse Long-Range Random Field

We now introduce our model. A sparse long-range random field (SLRF) is constructed so as to have a fixed clique size regardless of the spatial extent of the edges in the grid. Consider a set of nodes $\mathcal{V}$ arranged on a grid, where there is a spatial distance defined between each pair of nodes. By choosing edges that increase in length exponentially, we can construct a graph that has a fixed clique size even though there is no bound on the maximum edge length. Consider the case of cliques of size 3, which as noted above (and discussed in more detail below) are important for image restoration because they enable the representation of second-order spatial statistics. A local 3-clique has edges of length 1 that connect each node to its immediate neighbors and edges of length 2 that connect each node to those two-away. Adding edges of length 4 to each node would then create additional 3-cliques composed of edges of length 2, 2 and 4, but does not increase the maximum clique size. Similarly for edges of length 8 and so on, as illustrated for the one-dimensional case in Figure 2.1.

More formally, each node is connected to other nodes at distance 2^k away

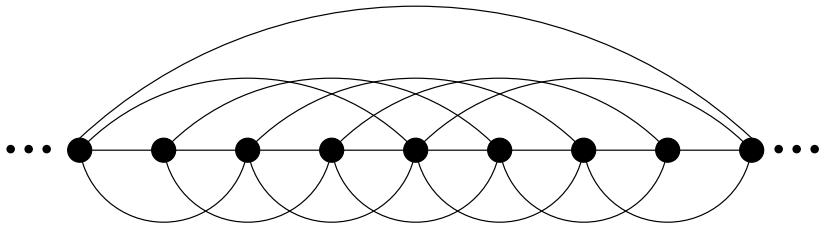


Figure 2.1: Horizontal 3-cliques of $\mathcal{E}_4^2$ with edge lengths $\{1, 2, 4, 8\}$.

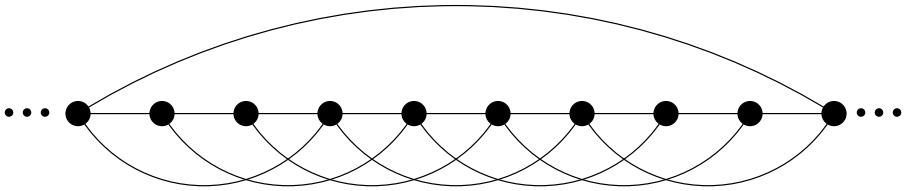


Figure 2.2: Horizontal pairwise cliques of $\mathcal{E}_3^3$ with edge lengths $\{1, 3, 9\}$.

from it, for integer values k such that $0 \leq k < K$. In other words the density of connections decreases exponentially with distance. We let $\mathcal{E}_K^2$ denote this set of edges; for instance, $\mathcal{E}_4^2$ is the set of edges of length $\{1, 2, 4, 8\}$. More generally, one could consider graphs where the edges are of length b^k for some $b > 2$ which yields sparser graphs. However, for $b = 3$ the resulting graphs already have maximum cliques of only size 2 (Figure 2.2), which for image denoising does not allow representing second-order image statistics.

In the case of a two-dimensional image grid, edges may correspond to any underlying orientation. Considering both horizontal and vertical directions using edges in $\mathcal{E}_K^2$ again yields a graph with maximum cliques of size 3. These cliques correspond to spatial neighborhoods at different scales of observation and at different orientations (horizontal or vertical), but in each case capture second-order spatial information based on three nodes.

The inclusion of long-range edges in the SLRF offers the following advantages over a local grid model:

- *Improved information flow*. The graph requires fewer hops to reach one node from another, as is illustrated in Figure 2.1. In the example shown in the figure, the maximum graph distance between any two nodes is 2. Without long range edges, the corresponding numbers would be 8. In general, it can be shown that any two nodes v_1 and v_2 with grid distance d have graph distance $O(d/b^{K-1} + bK)$. The decreased graph distance facilitates the flow of information over the random field.

- *Increased resistance to noise*. Long-range edges address the local bias problem discussed in Section 2.1. For any $n \times n$ neighborhood S with n up to b^{K-1} (i.e. up to the length of the longest edges), each node in S is connected to at least four nodes outside of S. Hence the total amount of interaction between S and the environment is now proportional to the area of S instead of its perimeter as in the 4-connected grid. This makes the strength of the spatial constraints between pixel blocks comparable to that of the data term over the block, suppressing noise-induced local bias without resorting to increasing the weight of the spatial term (which tends to cause over-smoothing).

The sparse nature of the SLRF also has the following computational benefits:

- *Small, fixed clique size*. As previously discussed, the size of the maximal cliques in an SLRF is either 2 or 3 regardless of the span of the longest range interaction being modeled. The low clique size allows arbitrary clique potentials to be optimized globally using efficient approximation algorithms such as belief propagation. In contrast, high-order random fields in general can only be optimized with continuous methods that rely on gradient (e.g. diffusion [77]), which may not exist in problems with

discrete labels. Even when gradient-based methods are applicable, the running time is still super linear in the size of the cliques. [1]

- *Low computational cost*. Since SLRF models have only K different edge lengths in an exponential series, the total number of edges in an SLRF is no more than K times of that in the underlying grid. Hence an SLRF model is at most $\log_b d$ times as costly as one with only short edges, where d is the length of the longest ranged interaction to be modeled. If on the other hand each node is connected to all the nodes near it up to some distance d (such as in [34]), the resulting graph would have $\Theta(d^2)$ edges and hence much higher computational. Although the model can still be called "sparse" from a graph theoretical point of view (as any graph with edge density independent of its size will qualify), it is clearly not so from the aspect of efficient optimization.

2.2.1 Cliques and clique potentials

Let $C = C_K^2$ denote the set of all cliques in an SLRF with edges $\mathcal{E}_K^2$ for a fixed K. There are several distinct types of cliques in this set, which can be characterized by the lengths of their edges. For instance,

$$C_K^2 = C_{1,1,2} \cup C_{2,2,4} \cup ... \cup C_{2^{K-2},2^{K-2},2^{K-1}} \tag{2.1}$$

where $C_{a,b,c}$ is the set of 3-cliques with edge length a, b, and c. Each of these sets of 3-cliques corresponds to observations at a different spatial scale, based on the lengths of the edges. Let $T(c)$ denote the type of clique c, e.g. $T(c) = (1, 1, 2)\ \forall c \in C_{1,1,2}$ and $T(c) = (1)\ \forall c \in C_1$.

[1] The time for computing the gradient is linear in clique size for using linear filters, and quadratic in the general case. At the same time, larger cliques also tend to require more iterations.

We represent the likelihood of the random field as an exponential family of cost functions f and g parameterized by θ, where $f^{\theta}_{T(c)}$ is the spatial term and g^{θ} is the data term. Thus given observation I,

$$p_\theta(X|I) = \frac{1}{Z(\theta)} \exp(-\sum_{c \in C} f^{\theta}_{T(c)}(\mathbf{x}_c; I) - \sum_{v \in \mathcal{V}} g^{\theta}(\mathbf{x}_v; I)) \tag{2.2}$$

where X is the labeling of the random field, and $\mathbf{x}_c$ and $\mathbf{x}_v$ are the configurations of clique c and node v respectively. The configuration of a clique or node includes its labeling, and may also include input-dependent latent variables such as image gradient. This formulation is similar to a CRF except that parametric functions over the clique and node configuration space $\mathcal{X}_f$ and $\mathcal{X}_g$ are used instead of features. The random field becomes Markovian when f is independent of the observed data, i.e. $f^{\theta}_{T(c)}(\mathbf{x}_c; I) = f^{\theta}_{T(c)}(\mathbf{x}_c)$ and g is a function only of the observation at a single node, i.e. $g^{\theta}(\mathbf{x}_v; I) = g^{\theta}(\mathbf{x}_v; I(v))$.

2.3 Parameter Estimation

To learn the parameters θ, it is desirable to find the maximum *a posteriori* (MAP) estimate. By applying Bayes' rule and assuming a uniform prior over the parameter space, this is equivalent to finding the maximum likelihood (ML) estimate. Computing the maximum likelihood estimate is nevertheless hard on loopy graphs due to the intractability of the partition function $Z(\theta)$ in $p_\theta(X|I)$. This makes it impossible to use the standard CRF learning scheme, since it is designed for tree-structured graphs where the partition function can be computed efficiently using dynamical programming [54]. Various approaches have been proposed to address this difficulty. Gradient descent methods [39] have been used to obtain a local minimum in the negative log-likelihood space. The expec-

tation over the model is nonetheless intractable to compute and often has to be estimated by MCMC sampling [77, 39], by loopy belief propagation [68, 111], or approximated using the mode (i.e. MAP labeling) [81]. The last case resembles the perceptron algorithms [19], except that the inference is not exact. As recently proposed in [108], a basis rotation algorithm based on expectation maximization (EM) can be used to learn parameters for filter based image models. This comes from a key observation that the partition function can be kept constant by constraining the parameter vectors to have unit norm. An alternative to maximum likelihood is using discriminative training to optimize for some loss function, typically evaluated on the mode. Such a loss can be minimized by descending along its derivative in the parameter space, when the mode has a closed-form solution [99] (or approximate solution [97]).

Since some approximation must be used, we take the approach of optimizing for the marginal likelihood of the random field cliques, which effectively approximates the global partition function using the product of local partition functions over the cliques. This can be considered as form of piecewise local training [93, 94], which minimizes a family of upper bounds on the log partition function. It can be shown that maximizing the marginal likelihood is equivalent to minimizing the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(p_0 \| p_\theta)$ between the empirical distribution p_0 and the model distribution p_θ for each type of cliques. The minimization can be performed using gradient descent with the standard update rule (as in [77])

$$\delta\theta = \eta \left[\left\langle \frac{\partial f_\theta}{\partial \theta} \right\rangle_{p_\theta} - \left\langle \frac{\partial f_\theta}{\partial \theta} \right\rangle_{p_0} \right], \tag{2.3}$$

where $\langle \cdot \rangle_{p_\theta}$ and $\langle \cdot \rangle_{p_0}$ denote the expectation with respect to the model and the empirical distribution respectively, and η is the learning rate.

Unlike in FoE we do not need to sample, since the model expectation can be computed by summing over all possible values of clique configurations. This computation is inexpensive in our model due to the small clique size. As noted in [77] performance can be improved by learning an additional weight for the data term, which we also use for our model.

2.4 Image Denoising

To test the effectiveness of our model, we apply it to the widely studied problem of image denoising. As is conventional in the literature, the image is assumed to be gray-scale and have been corrupted by additive white Gaussian noise of known standard deviation. Since this is a well-defined generative process, we model the data term using the known Gaussian noise model and only the spatial term needs to be estimated. As described above we use 3-cliques since they capture second-order properties. In order to illustrate the importance of these second-order statistics we considered the marginal statistics of the images in the Berkeley dataset [66] that is commonly used in evaluations of such methods. These images show a strong correlation between the distribution of neighboring pairs, suggesting that simple pairwise models are less appropriate (see Figure 2.3).

We denote clique c of type $C_{s,s,2s}$ as a triplet $(v^c_{-s}, v^c_0, v^c_{+s})$, where v^c_0 is the center node of c, v^c_{-s} is the left node, and v^c_{+s} is the right node. We limit our discussion to horizontal cliques, as the case for vertical ones is essentially the same. Let $d_1(c) = X(v^c_{+s}) - X(v^c_{-s})$ and $d_2(c) = X(v^c_{-s}) + X(v^c_{+s}) - 2X(v^c_0)$, where X is the labeling of the image. Hence d_1 and d_2 are proportional to the discrete first and second

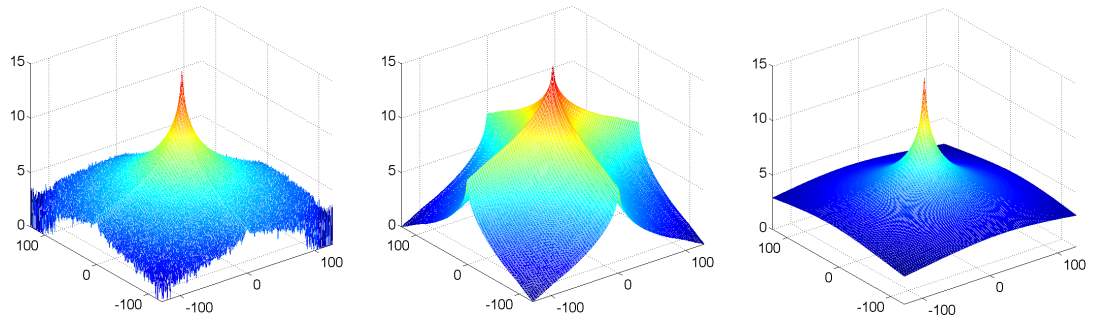


Figure 2.3: Frequency (unnormalized, logarithm scale) plotted against gradients of the two neighboring pairs in a linear 3-clique, from the Berkeley dataset [66]. (a) The empirical marginal distribution. (b) The would-be distribution if gradients of the neighboring pairs were independent. (c) The distribution from a fitted Lorentzian cost function.

derivatives of the image luminance respectively. In other words, the clique potential couples both first and second order spatial information.

The Lorentzian function has been widely used to model the statistics of natural images (e.g., [43, 77, 97]). In our case, we use a family of 2-dimensional Lorentzian functions for the spatial term, i.e.

$$f(\mathbf{x}_c) = \alpha \cdot \log(1 + \frac{1}{2}[(\beta_1 d_1)^2 + (\beta_2 d_2)^2]) \tag{2.4}$$

where $\{\alpha, \beta_1, \beta_2\}$ is the set of parameters for cliques of type $T(c)$. Hence f is intensity-invariant and regulates both the first and the second derivatives of the spatial signal. We choose this family since it not only fits the statistics of natural images (Figure 2.3) but is also able to produce smooth gradient while preserving discontinuities. This form is subtly different from filter based models, such as [77, 108], that use a linear combination of functions over filter responses; in our case the first and second order derivatives are coupled, that is both orders of derivatives are inputs to the same non-linear function rather than using a linear

combination of separate non-linear functions of each spatial filter.

It has been noted that natural images are self-similar over different spatial scales [32, 88]. As a result, cliques with different scales (i.e. edge lengths) all have very similar marginal distributions. This makes the marginals of cliques at different scales highly correlated, which we also observed empirically. Hence using independently collected marginals as the clique potentials is not a good model when dealing with natural scenes. To account for this factor, we reweigh the distribution of smaller-scale cliques according to the marginals of larger-scale ones, so as to make the former learn different trends from what have already been captured by the latter.

2.4.1 Inference

For denoising, inference can be performed using either belief propagation (BP) [55] or gradient based methods such as limited memory BFGS (L-BFGS) [70]. We experimented with both and found that L-BFGS produces the same quality of results as BP while requiring less running time. Hence the results we report in this chapter are based on using L-BFGS. It should be noted, however, that some problems in vision are of a discrete nature and cannot be solved using gradient-based methods. In those cases, discrete optimization techniques such as BP and graph cuts are required.

Table 2.1: Denoising performance of SLRF measured in peak signal-to-noise ratio (PSNR), higher is better. Results from other random field based denoising methods are shown for comparison. (Bold indicates the best performance among the 3-clique MRF models, asterisk denotes the best overall result, and "–" indicates no published data available.)

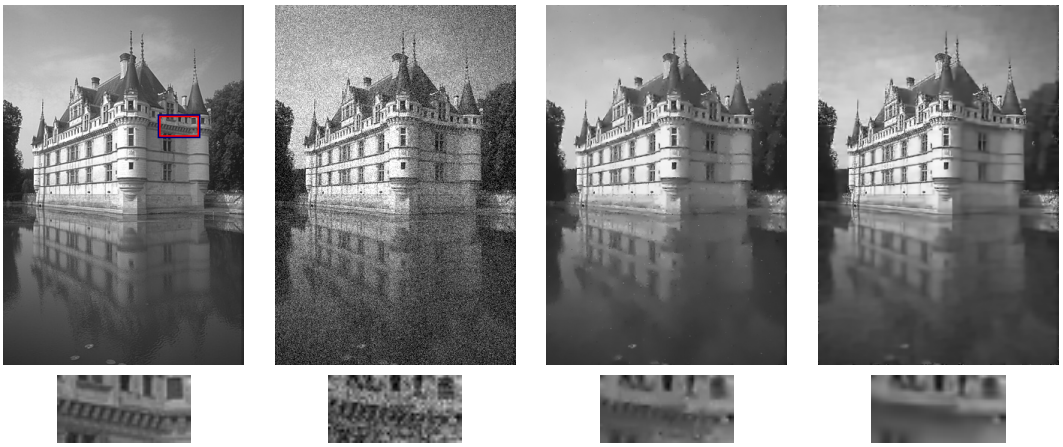
Model \ Noise σ	5	10	15	20	25
SLRF, K=4	**36.90**	**32.71***	**30.39**	**28.86***	**27.73**
Local MRF, K=2	36.51	32.04	29.81	27.89	26.41
FoE [77]	–	32.67	30.47*	28.79	27.59
GCRF [99]	–	–	–	–	28.04*
Var. MRF [97]	–	–	30.25	–	–
SRF [79]	–	–	–	28.32	–

2.5 Experimental Results

To evaluate the model for image denoising we used the Berkeley Segmentation Dataset and Benchmark [66] in order to compare the results with previous methods. The models were trained on the training images in the dataset and performance was measured on a subset of the testing images, which are the same as the ones used in [77, 79, 97, 99]. In all the experiments we ran L-BFGS for 20 iterations, which we found to be sufficient for our model. [2] This is in contrast to large-clique methods, which usually require many hundred iterations to produce results of good quality [77, 99].

Table 2.1 shows the denoising performance of our model along with the results from the FoE model in [77], the steerable random field (SRF) in [79], the Gaussian CRF in [99], and the variational MRF in [97]. This table reports the peak signal-to-noise ratio (PSNR) of each method averaged over the 68 test images (higher is better). These results demonstrate that the performance of our

[2] We also experimented with conjugate gradient as the optimization method, which achieved the same performance but needs a few more iterations (about 30 as opposed to 20 for L-BFGS).



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

Figure 2.4: Denoising output for a medium-texture scene. (a) Original image. (b) Corrupted by Gaussian noise, $\sigma = 25$. (c) Restored using our SLRF model, PSNR = 28.63. (d) Restored using FoE [77], PSNR = 28.72. The magnified view shows that our model, while having comparable PSNR, does a significantly better job at preserving the small and low-contrast structures of the stonework below the windows.

approach is comparable to that of recent top performing random field methods using the standard measure of PSNR. However, as is widely recognized, PSNR does not tell the entire story, thus we also consider some example images in more detail both to show the overall quality and to highlight the extent to which our method removes noise without smearing out the details.

Figures 2.4 and 2.5 display sample outputs from our model (in c) and from FoE (in d), illustrating the comparable quality of our method and FoE. In particular our method is able to reproduce image texture without yielding to the visually unpleasant blockiness that other methods using small cliques tend to produce [27, 55]. The enlarged regions in each of the images illustrate that our method is able to reproduce fine-scale texture better than the FoE. For instance in the castle image (Fig. 2.4), the stonework detail below the windows is smoothed out in the FoE but preserved in our model. The textured surface of the

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

Figure 2.5: Denoising output for a high-texture scene. (a) Original image. (b) Corrupted by Gaussian noise, $\sigma = 25$. (c) Restored using our SLRF model, PSNR = 26.02. (d) Restored using FoE, PSNR = 25.55. Again, the detail illustrates that our model not only achieves good PSNR but also produces less over-smoothing.

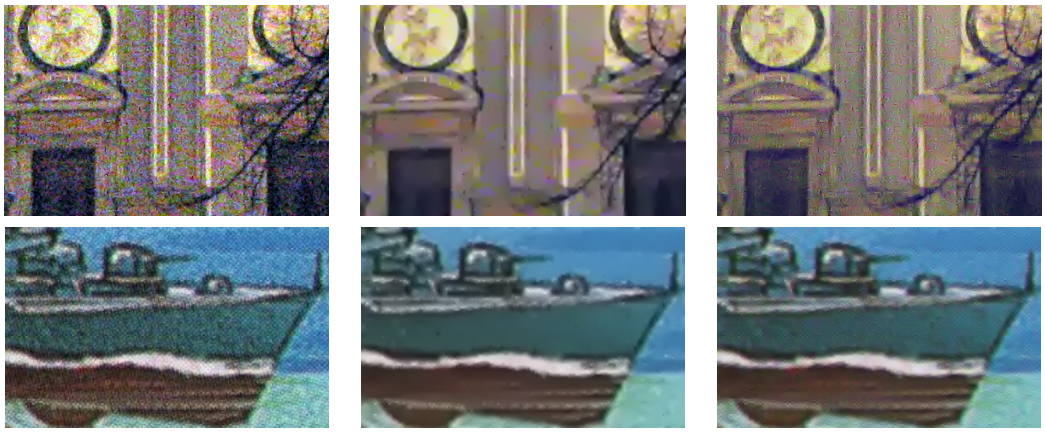
rocks in the sheep image (Fig. 2.5) similarly illustrates the ability of our method to preserve realistic texture while removing noise, rather than over-smoothing. Moreover, our method produces a consistent level of sharpness across the whole image, and, unlike FoE, does not tend to make high-contrast regions very sharp while low-contrast regions very smooth (Fig. 2.4 and 2.5, compare (c) and (d)). This gives the output of our model a more natural look.

Table 2.1 also shows that the model with long-range edges ($K = 4$) performed better than the local model ($K = 2$), in terms PSNR, and that the difference is most pronounced at high noise levels (e.g. $\sigma = 25$) as would be expected. Even at low noise levels (e.g. $\sigma = 5$), where one would not necessarily expect much help from longer-range connections, the long-range model still slightly outperformed the local model. This suggests that long-range interactions increase robustness of the model without sacrificing fine-scale precision. Figure 2.6 shows in side-by-side comparison some sample output of the long-range model and

Figure 2.6: Comparison of denoising outputs of the long-range and the local models. Input images have Gaussian white noise with $\sigma = 25$ (PSNR = 20.17). Left: Results of the long-range ($K = 4$) model. Right: Results of the local ($K = 2$) model. Observe that the outputs of the local model is blocky and appear tainted while those of the long-range model are smooth and clean.

the local model. The difference in visual quality between the two emphasizes that longer-range connections are useful and that our simple second-order models are capturing important properties of images, though these are not completely reflected in the PSNR numbers.

In addition to the experiments with artificial Gaussian noise, we also test our model on real-world noisy images. For color images, we simply transform them



Input SLRF BLS-GSM [74]

Figure 2.7: Results on two real-world noisy images used in [74]. For these two images, our model assumes Gaussian white noise of standard deviation of 50 and 25 respectively. Despite the lack of accurate noise model, the visual quality of the output of our method is comparable to that of [74].

into YCbCr space and apply the model on each channel separately. In all our experiments, Gaussian white noise is assumed. Although this is suboptimal, we obtain qualitatively good results as can be seen in Figure 2.7.

These results illustrate that our model utilizing sparse long-range connections achieves state-of-the-art performance when compared with other random field methods for image denoising. Arguably the better preservation of texture and more natural look compared with FoE, without the blocky effects of other local methods, improves upon previous results. Due to the small clique size and hence low complexity, our model is less prone to artifacts, such as the ringing pattern, which occurs more often with higher-order models. The highest PSNR has been achieved by wavelet based methods (e.g. [75, 74]); nevertheless, such models tend to produce a larger amount of ringing artifacts.

Finally we compare in Table 2.2 the running time of our model with those

Table 2.2: Running time of various image denoising methods.

Method	Image size	Processor	Running time (sec.)
SLRF	481×321	Xeon-3.0GHz	3.2
FoE [77]	481×321	Xeon-3.2GHz	376.9
GCRF [99]	481×321	Xeon-3.2GHz	97.8
GSM [75]	256×256	PentiumIII-1.7GHz	approx. 40

reported for some other methods, including both random field [77, 99] and wavelet-based [75]. These results show that our method is a factor of 30 or more faster than the other random field methods and about 10 times faster than the wavelet-based one (note that while the running time in this last case is for a slower processor, the image is also considerably smaller). The speed of our model makes it a practical denoising method even for high-resolution images.

CHAPTER 3

LEARNING FOR STEREO USING STRUCTURAL SVM

3.1 Introduction

Stereo is among the most widely studied low-level problems in computer vision. It is an especially challenging task due to the inherent ambiguity in pixel matching, which is further complicated by phenomena such as occlusion and untextured regions. Random field models, which address the ambiguity problem by enforcing global consistency using spatial priors, have substantially advanced the state of the art of stereo vision [96, 98]. Despite the progress, however, the parameters of most of these models remain hand-tuned. This not only requires a large amount of human effort but also limits the adaptability of the model, because of the difficulty of optimizing models by hand for new environments.

In this chapter we present a conditional random field [54] based model for stereo vision with non-parametric cost functions, which can be learnt automatically using the structural support vector machine (structural SVM) [104] with linear kernels. We choose the discriminative conditional random field (CRF) over its generative counterpart, the Markov random field (MRF), because the former avoids the necessity to define a generative process, which is somewhat difficult to characterize in stereo. For instance it is common and often desirable to use gradient-adaptive spatial terms, which tend to violate the Markovian independence assumptions of generative models. Deviating from the traditional approach in random field based stereo, we use non-parametric cost functions to model the node and clique potentials of the CRF. In addition to providing flex-

ibility in functional forms, this non-parametric approach allows us to express the total cost of the model as the inner product of a feature vector and a vector of the corresponding costs, where the costs correspond to the parameters of the model. The negated costs are commonly called "feature weights" in the machine learning literature. This linear form of the model enables us to use the structural SVM to learn the model parameters.[1]

The structural SVM is a large-margin method for estimating parameters, and can be an attractive alternative to the commonly used maximum likelihood estimator. A major advantage of the structural SVM, and large-margin methods in general, is that they take the loss function into consideration during training. Therefore, the approach can be used to train different models that specifically target different types of loss. In contrast, the maximum likelihood method is oblivious to loss; in fact it can be regarded as always minimizing the expected aggregate 0/1 loss (which is 0 if the labeling is *completely* correct and 1 otherwise). Such a loss function is clearly not optimal for most low-level vision problems, which usually have pixel-based performance criteria.

In this chapter, we describe a formulation of stereo vision in terms of non-parametric CRF models and a technique for training them using the structural SVM. This approach naturally allows us to learn models using the kinds of evaluation criteria that are normally used to assess stereo, such as the number of pixels whose labels are within 1 unit of the correct disparity level [82]. In our experiments we demonstrate that our method significantly outperforms other pixel-based stereo methods that have parametric (e.g. Potts) potentials trained using maximum likelihood. We also investigate the effect of the underlying graph structure on model performance, and show that the addition of explicit

[1] In this work, we only consider structural SVM with linear kernels.

non-local interactions (as described in Chapter 2) generally improves accuracy on more difficult scenes and especially in the presence of image noise.

3.1.1 Related Work

Random fields are among the most popular models in stereo vision. Common forms of the spatial term for stereo are parametric functions of the disparity difference between neighboring pixels, which usually model the distribution as a mixture of a line process and an outlier process (e.g. in [92, 90, 114]). These include the Potts model and the truncated linear model. The functions are sometimes gradient adaptive (e.g. in [81]) to encourage discontinuity in disparity to coincide with change in image intensity. The data term is typically the value of some dissimilarity measure, such as the absolute intensity difference.

While these functional forms have been successfully used to produce good results, some fundamental issues remain unaddressed. Reasonable and intuitive as they are, parametric spatial terms such as the Potts and line-outlier models make particular assumptions about the form of the disparity distribution, which may not be true for the data. Therefore these models can be over-restrictive and fail to fit the data well. Using any dissimilarity measures directly as the cost function for the data term is also problematic. While a sophisticated metric, such as the sampling-insensitive dissimilarity [12], can provide a faithful measure of image difference and hence a reliable input to the data term, the metric *itself* is not necessarily a good cost function. In our model, the spatial term is a non-parametric function of disparity difference and discretized image gradient, and the data term is a non-parametric function of discretized dissim-

ilarity value. While non-parametric stereo has been studied in many earlier works (e.g. [112, 6]), these approaches are typically based on ordering transforms and formulated as purely local methods rather than the global models that we investigate here.

Learning for stereo vision is a challenging subject. Considerable progress has been made in recent years, largely owing to the increasing availability of ground truth data. The work of [53] learns a probability model for matching errors using the scene structure of the input images. In [114], an expectation maximization (EM) algorithm is used to iteratively estimate disparity and relearn the model parameters based on the estimate. While these methods have shown promising results, they do require some initial model whose parameters still need to be preset. Moreover they are conducted in a manner different from the standard settings of machine learning, where there are separate training and testing data. In these previous works, the model is learned from the same (unlabeled) data that is to be labeled, and the parameters are adjusted in order to improve performance. Our approach, on the other hand, learns the model from labeled training data and tests it on unseen inputs, which is a standard form of supervised training in machine learning.

A recent work that employs supervised learning for stereo vision is [81], where a maximum likelihood estimator for the model parameters is obtained via gradient descent. Computing the likelihood gradient, however, involves the partition function, which is intractable on loopy graphs. Hence the partition function is approximated by the mode of the model distribution (i.e. the MAP solution), which is obtained using graph cuts (GC) [15]. However the gradient tends to be noisy due to the approximation, as is observed in [81], which can

lead to inaccurate estimates.

Large margin methods are an alternative to the maximum-likelihood approach, and were originally introduced in the context of binary classification using optimal hyperplane separation [105]. The idea was first adapted to domains with structured output in the framework of max-margin Markov networks (M^3N) [100], where the required margin is rescaled by the loss of the inferred labeling. Since the set of linear constraints (of the M^3N quadratic program) is exponential in size, it is replaced with a non-linear constraint approximately solvable by linear programming relaxation. The method was subsequently applied to several low-level vision problems, including segmentation and terrain classification, demonstrating improvement [3] over the performance of previous models. Though a remarkable breakthrough, M^3N has its limitation. The linear programming formulation places a restriction on the form of admissible loss functions; more specifically, the per-label (i.e. per-pixel) loss function must be an indicator and must return zero if and only if the the inferred label is the exact same as the ground truth. In particular such a form of loss function is not well suited to stereo, where the performance metric typically allows an error range around the true value (e.g., [82]).

The structural SVM [104] handles the exponential number of linear constraints in the quadratic program by employing a cutting-plane method. The algorithm iteratively finds the most violated constraint, i.e. the labeling with the smallest cost-less-loss value, and recomputes model parameters. The process is repeated until no significantly more violated constraint can be found. Thus the structural SVM places no restrictions on the form of loss functions, as long as the most violated constraint is feasible to compute under such loss.

For random field based stereo, finding the exact most violated constraint is not tractable due the loopy graph structure; nonetheless an approximate one can be obtained using energy minimization techniques. In our work, we use loopy belief propagation (BP) [27, 68, 111] for this purpose. Although loopy BP is known to perform poorly on densely connected graphs [29], the graph structure for stereo vision is sparse enough that this does not become an issue. Our experiments show that the models trained with the approximate most violated constraints obtained in this way perform well in practice.

3.2 CRF Model for Stereo

We model the problem of disparity labeling as a conditional random field on a grid graph. Hence each node, representing a pixel, is connected to its four nearest neighbors in both the horizontal and vertical direction. Later we will also formulate models with longer-range connections and investigate the impact on performance from the modified graph structure. For ease of presentation, however, we will start by describing the model defined on the conventional 4-connected grid.

Let $\mathcal{V}$ be the set of nodes and $\mathcal{E}$ be the set of edges in the graph. As is well known, the likelihood of a labeling X (i.e. the disparity map) given the input I decomposes into the product of maximal clique potentials and node potentials,

$$p(X|I;\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{(u,v)\in\mathcal{E}} \phi^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) \prod_{v\in\mathcal{V}} \phi^{\boldsymbol{\theta}}_{v}(x_v, I), \tag{3.1}$$

where $\boldsymbol{\theta}$ represents the parameters of the model and $Z(\boldsymbol{\theta})$ is the partition function. The notations $\mathbf{x}_{uv}$ and x_v denote the labeling over clique (u, v) and node v respectively. Note that the maximal cliques are simply the edges in the grid,

since the graph is pairwise. As is a common practice, we assume that the distribution is in the general exponential family with $\phi^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) = \exp[-f^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I)]$ and $\phi^{\boldsymbol{\theta}}_{v}(x_v, I) = \exp[-g^{\boldsymbol{\theta}}_{v}(x_v, I)]$, where $f^{\boldsymbol{\theta}}_{uv}$ and $g^{\boldsymbol{\theta}}_{v}$ are cost functions for the spatial and the data terms respectively. Hence the *cost of labeling* X, given input I, can be defined in the negative log-likelihood space as

$$\begin{aligned} E_{\boldsymbol{\theta}}(X, I) &= -\log p(X|I; \boldsymbol{\theta}) - \log Z(\boldsymbol{\theta}) \\ &= \sum_{(u,v)\in\mathcal{E}} f^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) + \sum_{v\in\mathcal{V}} g^{\boldsymbol{\theta}}_{v}(x_v, I). \end{aligned} \tag{3.2}$$

This quantity is also commonly referred to as the *energy* of the random field, and we will use the words "energy" and "cost" interchangeably. Since the input I is a constant and the log partition function $\log Z(\boldsymbol{\theta})$ does not depend on X, finding a labeling that maximizes the likelihood $p(X|I; \boldsymbol{\theta})$ is equivalent to finding one that minimizes the cost $E_{\boldsymbol{\theta}}(X, I)$.

The input for stereo consists of two images $I = (I_L, I_R)$, where I_L is the one taken by the left camera and I_R by the right. We assume without loss of generality that the disparity map is always computed for the left camera scene I_L. Since we model occlusion explicitly, the set of labels include the set of integer disparity levels plus occlusion.

3.2.1 Spatial Term

The spatial cost $f^{\boldsymbol{\theta}}_{uv}$ is a function of disparity levels at neighboring pixels u and v as well as the local image gradient. More specifically,

$$f^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) = f^{\boldsymbol{\theta}}_{uv}(J(x_u, x_v), K(u, v)), \tag{3.3}$$

(recall from Equation 3.1 that $\mathbf{x}_{uv}$ and x_v are the labelings of the clique (u, v) and the node v respectively) with discrete valued functions J and K

$$J(x_u, x_v) = \begin{cases} x_v - x_u & \text{if neither } u \text{ nor } v \text{ is occluded} \\ left\_occl & \text{if } u \text{ is occluded} \\ right\_occl & \text{if } v \text{ is occluded} \\ 0 & \text{if both } u \text{ and } v \text{ are occluded} \end{cases} \tag{3.4}$$

and

$$K(u, v) = \lfloor |I'_L(v) - I'_L(u)| \rfloor. \tag{3.5}$$

For $f^{\boldsymbol{\theta}}_{uv}$ we assume that (u, v) is in the horizontal direction, since the case for vertical direction is entirely analogous. In $K(u, v)$, I'_L is I_L after a small amount of Gaussian smoothing, which is applied to reduce the impact of texture and noise. In the case of color images, $|I'_L(v) - I'_L(u)|$ is averaged over the color channels.

Since the structural SVM requires the model to have a linear discriminative function, the cost function $f^{\boldsymbol{\theta}}_{uv}(J(x_u, x_v), K(u, v))$, abbreviated as $f^{\boldsymbol{\theta}}(J, K)$, has to be linear; in other words, $f^{\boldsymbol{\theta}}(J, K)$ needs to be expressible as the inner product of a parameter vector and some feature vector. This can be achieved using the following form for the cost function

$$f^{\boldsymbol{\theta}}(J, K) = \theta_{f(jk)} \qquad \text{if } J = j \text{ and } K = k \ , \tag{3.6}$$

where $\theta_{f(jk)}$ are real valued model parameters. Let $\psi_{uv}(j, k)$ be the indicator function that $J(x_u, x_v) = j$ and $K(u, v) = k$, i.e. it is one if the condition holds and zero otherwise. Let $\boldsymbol{\psi}_{uv}(\mathbf{x}_{uv}, I)$ denote the vector whose entries are $\psi_{uv}(j, k)$ for each combination of j and k at clique (u, v). Let $\boldsymbol{\theta}_f$ be the vector that contains the corresponding parameters $\theta_{f(jk)}$. Hence $f^{\boldsymbol{\theta}}_{uv}$ can be written as the inner product of $\boldsymbol{\theta}_f$ and $\boldsymbol{\psi}_{uv}(\mathbf{x}_{uv}, I)$, i.e.

$$f^{\boldsymbol{\theta}}_{uv}(\mathbf{x}_{uv}, I) = \left\langle \boldsymbol{\theta}_f, \boldsymbol{\psi}_{uv}(\mathbf{x}_{uv}, I) \right\rangle. \tag{3.7}$$

For notational convenience we assume that horizontal and vertical cliques (i.e. edges in pairwise models) share the same spatial parameters. Below we will discuss the extension to anisotropic clique potentials, which is straightforward. We define the *spatial feature vector*

$$\mathbf{\Psi}_f(X, I) = \sum_{(u,v)\in\mathcal{E}} \boldsymbol{\psi}_{uv}(\mathbf{x}_{uv}, I). \tag{3.8}$$

Hence the total spatial cost (i.e. the first term of Equation 3.2) is

$$E_{\boldsymbol{\theta},f}(X, I) = \left\langle \boldsymbol{\theta}_f, \mathbf{\Psi}_f(X, I) \right\rangle. \tag{3.9}$$

When horizontal and vertical cliques have different potentials, we simply have separate parameter and feature vectors for each type of cliques. The overall vectors are just the concatenations over the different clique types, and hence the cost is still the inner product of the parameter vector and the feature vector as in Equation 3.9. The same extension also applies directly to the scenario where there are multiple classes of edges in the graph that may correspond to various lengths and orientations.

3.2.2 Data Term

Similar to the spatial cost, the data cost $g_v^{\boldsymbol{\theta}}$ is also defined as a non-parametric function

$$g_v^{\boldsymbol{\theta}}(x_v, I) = \begin{cases} c_v \cdot \theta_{g(k)} & \text{if } v \text{ is not occluded and} \\ & \lfloor \delta(v, I_L, v - x_v, I_R) \rfloor = k \\ c_v \cdot \theta_{g(occl)} & \text{if } v \text{ is occluded} \end{cases} \tag{3.10}$$

where c_v is some constant scalar and $\delta(v, I_L, v - x_v, I_R)$ is the sampling-insensitive dissimilarity [12] between pixel v in image I_L and its match in I_R.

As before, let $\boldsymbol{\theta}_g$ be the vector containing all data term parameters $\theta_{g(k)}$ (including $\theta_{g(occl)}$) and let $\boldsymbol{\psi}_v(x_v, I)$ be the corresponding vector of indicators for the conditions in Equation 3.10. Hence $g_v^{\boldsymbol{\theta}}$ is the inner product of $\boldsymbol{\theta}_g$ and $c_v\boldsymbol{\psi}_v(x_v, I)$

$$g_v^{\boldsymbol{\theta}}(x_v, I) = \left\langle \boldsymbol{\theta}_g, c_v\boldsymbol{\psi}_v(x_v, I) \right\rangle. \tag{3.11}$$

Analogous to spatial features, the *data feature vector* is defined as

$$\boldsymbol{\Psi}_g(X, I) = \sum_{v \in \mathcal{V}} c_v\boldsymbol{\psi}_v(x_v, I). \tag{3.12}$$

We let c_v equal the degree of node v, so that the ratio between the total counts of spatial features and data features is constant with respect to the number edges. This prevents potential imbalance between the norms of the spatial and the data feature vectors when the model has multiple families edges and hence a higher edge-to-node ratio. Thus it ensures that SVM never places too much attention on one type of features and not enough on the other.

The total data cost (i.e. the second term of Equation 3.9) is also the inner product of the data parameter vector and the data feature vector,

$$E_{\boldsymbol{\theta},g}(X, I) = \left\langle \boldsymbol{\theta}_g, \boldsymbol{\Psi}_g(X, I) \right\rangle. \tag{3.13}$$

Therefore, the total cost of labeling X given input I is

$$E_{\boldsymbol{\theta}}(X, I) = \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(X, I) \rangle \tag{3.14}$$

where parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_f^T, \boldsymbol{\theta}_g^T)^T$ and feature vector $\boldsymbol{\Psi}(X, I) = (\boldsymbol{\Psi}_f(X, I)^T, \boldsymbol{\Psi}_g(X, I)^T)^T$ are both concatenated over the spatial and the data terms. The desired labeling under the model is simply the one with minimum cost.

3.2.3 Graph Structure with Long-range Edges

In addition to the grid graph, we also explore structures with long-range edges. This corresponds to the SLRF model introduced in Chapter 2. In particular, we consider horizontal and vertical edges that have length 3^k for $k = 0, 1, 2, \cdots, K-1$. The larger K, the greater the maximum range of explicit interaction is modeled. Thus the grid graph is a special case where $K = 1$. We choose 3 as the base, since it is the smallest integer for which the random field remains strictly pairwise (i.e. the maximal cliques are still of size two and thus the same formalization applies). The exponentially increasing edge length also enables us to model longer range of interaction at relatively lower computational expense, compared with earlier models with denser edges (e.g. [34]).

The cost function in this more general setting is still the inner product between parameters and features, where the spatial term vectors are concatenated over each type of edges. Hence the form of Equation 3.14 remains valid under this extension.

3.3 Parameter Learning

The model parameters are learnt using the structural SVM [104]. Let $((I^{(1)}, X^{(1)}), \cdots, (I^{(n)}, X^{(n)}))$ be the training examples, each of which is an input-output pair. The structural SVM optimizes for parameters $\boldsymbol{\theta}$ by minimizing a quadratic objective function subject to a set of linear soft margin constraints

$$\begin{aligned} &\min_{\boldsymbol{\theta}, \boldsymbol{\xi}} \frac{1}{2} \|\boldsymbol{\theta}\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i \\ &\text{s.t. } \forall i, \forall X \in \mathcal{X} : \langle \boldsymbol{\theta}, \boldsymbol{\delta\Psi}_i(X) \rangle \geq \Delta(X^{(i)}, X) - \xi_i \end{aligned} \tag{3.15}$$

where $\mathcal{X}$ is the set of all possible labelings, ξ_i are the slack variables associated with each example, and $\Delta(X^{(i)}, X)$ is the loss function, which we will define later in this section. Also $\boldsymbol{\delta}\boldsymbol{\Psi}_i(X)$ denotes $\boldsymbol{\Psi}_i(X) - \boldsymbol{\Psi}_i(X^{(i)})$ with $\boldsymbol{\Psi}_i(X)$ being shorthand for $\boldsymbol{\Psi}(X, I^{(i)})$, and $C > 0$ is a constant that controls the trade-off between margin and training error. Rearranging the terms of Equation 3.16 shows that the SVM objective function is an upper bound on average training loss (up to a constant factor C), as long as a labeling with cost no higher than that of the ground truth can be found for every training example. While this condition is not guaranteed due to the intractability of exact energy minimization on loopy graphs, it is often true in many real-world low-level vision problems and especially stereo [96].

The apparent difficulty in this formulation is the exponential sized labeling set $\mathcal{X}$. The structural SVM addresses this problem by replacing it with a collection of finite constraint sets S_i. Initially all the constraint sets S_i are empty and the parameter vector $\boldsymbol{\theta}$ is set to some arbitrary value, typically all-zeros. At each iteration and for each example i, the algorithm computes the most violated constraint, i.e. one with the largest slack ξ_i, and adds it to the constraint set S_i if it is more violated than those already in the set. The solution to the quadratic program is then recomputed and hence $\boldsymbol{\theta}$ updated. The algorithm iterates until no new constraints are added.

Since maximizing ξ_i is equivalent to minimizing $\langle \boldsymbol{\theta}, \boldsymbol{\delta}\boldsymbol{\Psi}_i(X) \rangle - \Delta(X^{(i)}, X)$ and $E_{\boldsymbol{\theta}}(X^{(i)}, I^{(i)}) = \left\langle \boldsymbol{\theta}, \boldsymbol{\Psi}_i(X^{(i)}) \right\rangle$ is a constant, the most violated constraint for example i is just the one with the smallest cost-less-loss value

$$\hat{X} = \arg\min_{X \in \mathcal{X}} \left\{ E_{\boldsymbol{\theta}}(X, I^{(i)}) - \Delta(X^{(i)}, X) \right\}. \tag{3.16}$$

For any per-pixel loss function, approximate solutions for $\hat{X}$ can be obtained efficiently using energy minimization techniques. It is worth noting that the struc-

tural SVM also provides several other formulations of the quadratic program [104]. However, the version with linear slack penalties and margin rescaling (Equation 3.16) is the only one under which there are known efficient approximation algorithms for $\hat{X}$ in stereo.

A challenge for learning non-parametric functions using the structural SVM is that the parameters, namely the discrete cost function outputs, are treated as independent variables by the learning algorithm, and hence the learnt cost functions may have certain characteristics that are unnatural for the underlying problem. In stereo, this is mainly manifested as fluctuations in the shape of the data cost function. Though the learnt function does have the expected overall trend of increasing with dissimilarity, it is not strictly monotone as it should be for stereo. We address this problem by imposing a monotonicity constraint on the data cost function after training. This is done by setting $\theta_{g(k)}$ to $\min(\theta_{g(k)}, \theta_{g(k+1)})$ in decreasing order of k ($k \neq occl$, i.e. occlusion cost is unchanged). In this way, we capture the domain-specific knowledge without further restricting the form of the cost function.

We also noticed that the lowest training error is usually achieved not by the final output of the SVM, but by some $\boldsymbol{\theta}$ produced after one of the intermediate training iterations. This is not surprising since the SVM objective function is not the same as training error, even though it is an upper bound on the loss. One reason to formulate learning as constrained optimization of such a bound is that directly minimizing training error is usually not feasible. Also in SVM theory, minimizing the norm of the learnt parameter vector (i.e. the first term of the objective function) is equivalent to increasing the margin [105] and hence guards against overfitting. For our stereo learning problem, however, we found

that overfitting hardly occurs and that generalization error is much more closely correlated with training error than with the value of the SVM objection function. Therefore, we choose from all learnt $\boldsymbol{\theta}$ vectors (produced after each iteration) the one with the lowest training error as the model parameter. Parameters learnt in this way are still large margin estimates since they are obtained through SVM optimization. This modified training procedure can be considered as exploring a subset of the parameter space that has the large margin properties, and choosing the best instance based on training performance.

3.3.1 Loss Functions

The most natural choice of loss function is simply the error function under which model performance is evaluated. For stereo this is usually the number of bad pixels in non-occluded regions (determined by the ground truth), where a pixel is bad if the disparity estimated by the model differs from the true disparity by an amount greater than some threshold r. The conventional choice in stereo is $r = 1$, which we use in our work. Hence the loss function is

$$\Delta(l) = \sum_{v \in \mathcal{V}} l(v) \tag{3.17}$$

where l is the pixel-wise loss and in the case of standard stereo evaluation metric it is

$$l_{std}(v) = \begin{cases} 1 & \text{if } v \text{ is bad and not in occluded regions} \\ 0 & \text{otherwise.} \end{cases} \tag{3.18}$$

Both l and $\Delta(l)$ take as arguments the ground truth and the proposed labeling, which are omitted from the notation above for conciseness. It is easy to see that l_{std} discourages labeling of occlusion, since every non-occluded pixel mislabeled as occlusion encounters a loss while there is no penalty for occluded

pixels mislabeled as non-occlusion. Such a loss function tends to produce models that label very little or no occlusion, though this is consistent with the goal of achieving the best performance in non-occluded regions.

We can extend the definition of bad (i.e. mislabeled) pixel to occluded region. In particular, we consider a pixel a *false negative* if it is occluded in the ground truth but not labeled so by the model; similarly it is a *false positive* if the opposite happens. In either case, the pixel is regarded as mislabeled. We can define a new pixel-wise loss that is aimed at achieving lower overall error rates by correctly identifying occluded regions

$$l_{occl}(v) = \begin{cases} q & \text{if } v \text{ is a false positive} \\ 1 & \text{if } v \text{ is otherwise mislabeled} \\ 0 & \text{if } v \text{ is correctly labeled} \end{cases} \tag{3.19}$$

where constant q adjusts the extent to which occlusion labeling shall be encouraged. If $q = 1$ then $\Delta(l_{occl})$ would measure the model performance as the number of bad pixels over the whole scene. For use as an SVM loss function, we find that a smaller value of 0.06 proves to be a better choice for improving overall accuracy.

3.4 Experimental Results

For performance evaluation we train and test our model mainly on the Middlebury-2005 stereo data set release in [81], which contains scenes that are more complex and challenging than the older ones on the Middlebury Stereo Evaluation page [82]. Since the stereo benchmark does not label occlusions, we simply fill in the occluded region by replacing the occluded pixel (inferred

Table 3.1: Performance of models on the Middlebury-2005 data set [81] as well as the "Teddy" and "Cones" scenes from the Middlebury Stereo Evaluation page [82]. Here the learnt models are trained on all the 6 scenes in Middlebury-2005. The table shows error rates measured as the percentage of bad pixels, lower is better, calculated in non-occluded regions (as is common). Bold fonts indicate the lowest error rates among the models being compared, and "–" indicates result not available.

† Extracted from the plots in Figure 6 – 8 of [81].

Model \ Scene	Art	Books	Dolls	Laun.	Moeb.	Rein.	**Avg.**	Teddy	Cones
- Grid ($K = 1$), l_{std} loss	14.66	19.12	12.70	19.16	10.88	**11.72**	14.71	11.34	4.68
- Grid, l_{occl}	15.24	21.13	**12.11**	17.14	11.28	16.47	15.56	10.92	4.27
- Long-range ($K = 3$), l_{std}	**12.11**	**15.68**	12.14	15.82	**10.80**	15.26	**13.64**	8.89	3.94
- Long-range, l_{occl}	12.69	16.29	12.57	**15.79**	11.30	15.70	14.06	8.15	**3.77**
- [81] w/ 2 gradient bins	–	–	–	–	–	–	18†	11.3	10.7
- [81] w/ 6 gradient bins	–	–	–	–	–	–	20	14.5	16.8
- [82] w/ GC (non-learning)	–	–	–	–	–	–	–	16.5	7.70
- [91] (non-learning)	–	–	–	–	–	–	–	**6.47**	4.79

by the model) with the disparity of the first non-occluded pixel to its left (or to its right if it is near the left boundary) when evaluating performance. This is obviously suboptimal and fails to exploit the full benefit of occlusion labeling; nonetheless, finding a good extrapolation scheme for occluded regions is beyond the scope of this work. For training of all models, we use Joachims's *SVM-struct* [104] with a C value (see equation 3.16) of 10^{-3} that is empirically chosen based on training error. The outcomes are nevertheless rather insensitive to the choice of C, and in fact values from 10^{-4} to 10^{-1} produce models that are indistinguishable in performance.

We compare our results with several other pixel-based stereo algorithms [81, 82, 91], and show that our model achieves a high level of performance. The error rates of our models are compared with those of [81] whenever possible (i.e. when the corresponding data is available in [81]). The comparison with [82] and [91], both non-learning based, is limited to the "Teddy" and "Cones"

Table 3.2: Performance of learnt models in leave-one-out cross validation on the Middlebury-2005 data set (top) and performance of models trained on Middlebury-2006 and tested on Middlebury-2005 (bottom). The error rates are measured in the same way as in Table 3.1.

Model \ Scene	Art	Books	Dolls	Laundry	Moebius	Reindeer	**Average**
Leave-one-out							
- Grid, l_{std}	15.54	20.81	12.83	18.21	11.69	**13.04**	15.35
- Grid, l_{occl}	15.11	21.97	12.88	18.10	11.13	14.09	15.55
- Long-range, l_{std}	**12.77**	17.56	**12.40**	**16.75**	11.25	15.41	**14.36**
- Long-range, l_{occl}	13.49	**15.98**	12.89	17.06	**10.64**	18.94	14.83
- [81], 2 grad. bins	–	–	–	–	17	14	–
- [81], 6 grad. bins	–	–	–	–	13	18	–
Train on Midd. 2006							
- Long-range, l_{std}	**13.60**	**16.13**	13.86	20.65	**12.21**	17.90	15.73
- Long-range, l_{occl}	14.37	16.79	**12.87**	**17.14**	12.84	**15.76**	**14.96**

Figure 3.1: Sample disparity maps for stereo scenes Art and Cones produced by long-range CRF models ($K = 3$) learnt with l_{occl} loss function. For "Art" (top), the model is trained on the rest 5 scenes in the same data set; for "Cones" (bottom), the model is trained on all 6 scenes in the Middlebury-2005 data set (which contains "Art" but not "Cones"). Occluded regions inferred by the model are masked in full black.

scenes, since these algorithms predate [81] and hence no results are reported on the Middlebury-2005 data. It should be noted that our method is raw-pixel based and treats stereo as a generic random field labeling problem, and does not use techniques such as weighted support windows, segmentation, or plane fitting (e.g. [49, 110, 116]). However, many of these more involved methods use MRF or CRF models at some stage, and thus our learning technique should prove useful to further work on such approaches to stereo.

The results in Table 3.1 show that our method achieves performance superior to that of [81], which also uses machine learning. Comparing with the non-learning based methods, the performance of our learnt models by far surpasses [82] and is comparable with [91], one of the top-performing stereo algorithms. This is despite the fact that [91] generates a second disparity map using the other image of the stereo pair in order to exploit visibility constraints, while our models are generic random fields and do not make use of this property.

Table 3.2 shows the error rates of leave-one-out cross validation, where for each scene the model is trained on all the other scenes in the data set. In addition we also train our model on the 2006 data set from the Middlebury Stereo website, which has very different characteristics, and test it on the 2005 data set. As one can see, the performance of leave-one-out cross validation is close to that of training on the whole data set; moreover, training on a very different data set yields only slightly higher error rates. This indicates that model and the training method generalize well to unseen data.

Another observation from the two tables is that when the models are trained directly using loss functions that encourage occlusion labeling (i.e. l_{occl}) they have nearly the same level of performance as those trained using the evaluation

Table 3.3: Model performance on noisy stereo input (Middlebury-2005 data set). The images from the testing scenes are corrupted by additive Gaussian noise with standard deviation σ. Models are trained on the original (non-noisy) images with l_{std} loss. (The scenario for using l_{occl} is essentially the same.) Evaluation is based on leave-one-out cross validation and the error rates are averaged over the whole data set.

Model \ Noise σ	3	5	7	10
Grid	18.84	24.18	32.25	46.44
Long-range	15.55	18.23	21.20	24.20

metric itself as the loss function (i.e. l_{std}). This demonstrates that our method is able to handle occlusion without sacrificing much accuracy in the non-occluded regions. Figure 3.1 displays some sample output disparity maps produced by models trained with l_{occl} loss. One can see that most of the occluded regions are correctly identified.

Moreover, the figures in both Table 3.1 and 3.2 suggest that models with explicit long-range interactions generally perform better than those with only local connections, namely the grid model (e.g. compare row 1, 2 with row 3, 4). This indicates that the inclusion of sparse long-range edges does yield some benefit. To investigate this further, we study the trends in which model performance degenerates with increasing image noise. This bears practical concern for stereo, since in real-world situations the input images are unlikely to be as noise-free as those taken in the lab. In fact, noise in stereo has been a subject of study in several recent works, e.g. [40, 38].

Table 3.3 shows the percentage error rates of grid and long-range models on stereo inputs with Gaussian noise. Here the difference is much more pronounced. The performance of the 4-connected grid model rapidly declines as the noise level increases, whereas the one with long range connections under-

goes a much more graceful degradation. Note that the goal of this comparison is not to develop a new method for noisy stereo, which is itself a separate research topic; it is simply shows the advantage of increased robustness of long-range models over the grid under equal conditions.

CHAPTER 4

LEARNING FOR OPTICAL FLOW USING STOCHASTIC OPTIMIZATION

4.1 Introduction

Optical flow is among the most widely studied problems in low-level vision. While the development of better matching and regularization criteria as well as more effective optimization techniques has significantly advanced the state of the art [16, 69, 2, 9, 17, 13, 1], the parameters of these methods are generally set by hand on the same data that is used for evaluation. Although there has recently been some work on learning for optical flow, such as [78], the practice of hand tuning parameters remains prevalent.

In this chapter, we describe a continuous-state Markov random field (MRF) [11, 33, 95] based model for optical flow and earn the parameters of this model using simultaneous perturbation stochastic approximation (SPSA) [85] to minimize the *training loss* – that is the error on the training data under some error function. In particular, we measure training loss using the average end-point error (AEPE) [71] which is one of the error metrics commonly used to assess the quality of optical flow estimation.

Directly optimizing for training loss as opposed to a maximum likelihood approach (e.g. as pursued in [78]) has a number of advantages. First, the likelihood of the data under models with loopy spatial dependency is intractable to compute. In order to obtain the maximum likelihood estimate, one thus has to resort to approximation techniques such as estimating the mode (i.e. the max-

imum *a posteriori* estimate) [81], sampling [78], or some type of local training [93]. These approximations, however, tend to be imprecise and may lead to noisy and unreliable estimates, as noted in [81]. Moreover, from a statistical learning point of view, the maximum likelihood estimate is not well suited for problems such as optical flow that have *structured* outputs, such as a label at each pixel, as opposed to a single overall right-or-wrong answer. Thus an attractive alternative approach is to minimize some specific loss function, or error metric, on the training data (e.g., [56]) as we discuss further in Section 4.3.

Learning model parameters that minimize the loss on the training data is a challenging optimization problem, since the relationship between the target function (i.e. the training loss) and the model parameters cannot be determined analytically except in some special cases (e.g. [97]). SPSA [85] is a convenient choice in this situation, since it only requires the target function to be smooth and to have non-vanishing gradient (with respect to the parameters being optimized), a rather generous condition that is usually satisfied, but does not require the analytical form or the true gradient to be known. Hence it can be used to optimize for a wide range of loss functions, including the commonly used error metrics on which optical flow quality is judged such as AEPE. Given the large number of problems in computer vision that have structured outputs and the breadth of applicability of SPSA [87], the approach that we develop here is likely to be of broader interest for other problems in computer vision.

We evaluate our method in the standard setup of supervised learning, namely by training the model on a set of sequences with ground truth and testing it on a different set that does not include any of the training sequences. This allows one to assess the generalization power of the learnt model. We compare

our results to those of previous methods and show that our model both generalizes well to unseen data and achieves state-of-the-art performance for optical flow.

4.1.1 Related Work

Optical flow is a highly challenging low-level vision problem due to inherent ambiguity in local regions of the image, known as the aperture problem [10]. This is further complicated by phenomena such as motion discontinuity, untextured regions, and sensor noise. To address these issues, many early methods utilize local support windows over which some matching cost is aggregated (e.g. [65]). Window-based approaches approaches, however, suffer from the generalized aperture problem [46], namely that they are either too small to provide sufficient support or too big that they span over motion boundaries. Although this can be alleviated by using parametric and mixtures models (e.g. [46]), support windows have not proven to be good for accurately estimating the motion of non-rigid bodies undergoing deformation. Moreover, purely local methods are susceptible to erroneous matches in poorly-textured regions even with the aid of support windows. Global models for optical flow, first proposed in [42], compute the flow field by minimizing a global energy function. The energy function is usually composed of a data term that encourages agreement between frames and a spatial term (i.e. regularization) that enforces consistency of the flow field. Markov random fields (MRF) are closely related to energy-based models [56] in that the node and clique potentials of an MRF are often defined in terms of energy or cost functions in general exponential families. Thus equivalence can be drawn between the negative log-posterior of MRF

models and global energy functions. MRF models, however, have explicitly-defined topology and are convenient to model higher-order and non-local (i.e. long-range) interactions (e.g. in [34, 78]), which would otherwise be more difficult to express.

Learning for optical flow is a challenging subject, which has not been studied extensively. A major difficulty for learning optical flow models is the scarcity of ground-truth data. Despite the challenges, considerable progress has been made over the past decades that has improved our understanding of the problem. The robust estimation framework introduced in [13] makes a key observation that the brightness constancy and spatial smoothness assumptions are often violated near motion boundaries, and hence robust energy functions such as the Lorentzian should be used instead of quadratics to account for these violations. Although that work proposes a variety of robust function forms, it does not attempt to automatically estimate their parameters. Probability distributions of optical flow are studied in [84], where they are used to represent uncertainties and account for errors. Nevertheless, the parameters of the models used to compute optical flow are still set by hand. In [30], linear bases of parameterized models are learnt from examples using principal component analysis. Pioneering for its time, these models mainly target certain specific motion types and are not designed for general motion estimation. In a more recent work, field-of-expert (FoE) models [77] are learnt from ground-truth flow fields inferred from range-scan data [78]. This appears to be the first work to employ supervised learning technique for general optical flow estimation. However, their model differs from ours in several important respects, including the use of decoupled large-clique (5×5) filters and learning with approximate maximum likelihood rather than considering training loss.

Simultaneous perturbation stochastic approximation (SPSA) is a stochastic optimization method that iteratively minimizes a given target function. At each iteration all model parameters are simultaneously perturbed at random, and the loss function is evaluated at the perturbed positions in the parameter space to estimate its pseudo-gradient with respect to the parameter vector. This information is then used to determine the direction of descent and to subsequently update the model parameters. Since exact convergence is difficult to determine, the algorithm is usually run for either a fixed number iterations or until the reduction in loss becomes insignificant.

The SPSA algorithm was first proposed in [85] as a gradient-free stochastic optimization method, and it is closely related to the classical finite-difference stochastic approximation (FDSA) algorithm [48] since both methods estimate the gradient of the loss function by measuring its values only and hence avoid the necessity to know its closed-form derivatives. However, by simultaneously perturbing all model parameters, SPSA requires substantially fewer measurements of the loss function and hence achieves faster convergence rates [85, 87]. The gradient-free SPSA is well-suited for problems where the input-output relationship of the system is difficult to determine. Since its introduction, SPSA has been applied to optimize for a variety of engineering systems ranging from traffic control, weapon targeting, to buried object localization [87]. Nevertheless, the method has received less attention in the vision community. To our knowledge, it has not previously been applied to learning parameters for low-level vision problems.

4.2 An MRF Model for Optical Flow

We model the optical flow computation as a labeling problem on a continuous-state Markov random field, where each node p, representing a pixel, receives a 2-dimensional vector label $\mathbf{w}_p \in \mathbb{R}^2$ indicating its flow (i.e. apparent motion). [1] In our MRF model, each node is connected to nodes that are either adjacent or two pixels away in both the horizontal and vertical directions. Hence the resulting MRF consists of linear 3-cliques (i.e. 3-node complete subgraphs) that are either horizontally or vertically oriented. The 3-clique MRF topology allows us to model both the first derivative (i.e. gradient) and the second derivative (i.e. curvature) of the flow field, which are both important motion statistics. On the other hand, the clique size is small enough that it doesn't pose a severe computational burden.

Let $\mathcal{V}$ be the set of nodes and C be the set of cliques of the graph. The posterior of a labeling $\mathbf{w}$ given data I decomposes into the product of maximal clique potentials and node potentials,

$$p(\mathbf{w}|I;\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{c \in C} \phi_c^{\boldsymbol{\theta}}(\mathbf{w}_c) \prod_{p \in \mathcal{V}} \phi_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I), \tag{4.1}$$

where $\boldsymbol{\theta}$ represents the parameters of the model and $Z(\boldsymbol{\theta})$ is the partition function. The notations $\mathbf{w}_c$ and $\mathbf{w}_p$ denote the labeling over clique c and node p respectively. Recall that the MRF is continuous, and thus $\mathbf{w}$ is also in a continuous vector space. As is a common practice, we represent the distribution in the general exponential family so that $\phi_c^{\boldsymbol{\theta}}(\mathbf{w}_c) = \exp(-f_c^{\boldsymbol{\theta}}(\mathbf{w}_c))$ and $\phi_p^{\boldsymbol{\theta}}(\mathbf{w}_c, I) = \exp(-g_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I))$. That is, $f_c^{\boldsymbol{\theta}}$ and $g_p^{\boldsymbol{\theta}}$ are the energy functions for the spatial term and the data term respectively, and the total energy of a labeling

[1] It is convention to use the notation $\mathbf{w}$ for optical flow variables.

(i.e. flow field) $\mathbf{w}$ given input I can be written as

$$\begin{aligned} E(\mathbf{w};\boldsymbol{\theta},I) &= -\log p(\mathbf{w}|I;\boldsymbol{\theta}) - \log Z(\boldsymbol{\theta}) \\ &= \sum_{c\in C} f_c^{\boldsymbol{\theta}}(\mathbf{w}_c) + \sum_{p\in\mathcal{V}} g_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I). \end{aligned} \tag{4.2}$$

Therefore minimizing the energy $E(\mathbf{w};\boldsymbol{\theta},I)$ is equivalent to finding the maximum *a posteriori* (MAP) labeling over the MRF. Although exact minimization of the energy is generally intractable due to the loopy graph structure, methods based on gradient descent can be used to obtain an approximate solution (which we shall return to in Section 4.2.2).

The input for our model is a pair of images, i.e. $I = (I_0, I_1)$. Without loss of generality, we assume that the flow is computed for I_0.

4.2.1 Energy Functions

We use robust energy functions as proposed in [13]; in particular, we choose the family of Lorentzian functions for their ability to maintain spatial consistency and brightness agreement while being tolerant to motion discontinuity in the spatial term and outliers in the data term. The energy function for the spatial term is defined as

$$f_c^{\boldsymbol{\theta}}(\mathbf{w}_c) = \lambda_S \cdot \rho(\sqrt{||\beta_1\mathbf{d_1}||^2 + ||\beta_2\mathbf{d_2}||^2}) \tag{4.3}$$

where λ_S, β_1, and β_2 are model parameters, the function

$$\rho(\cdot) = \log(1 + \frac{1}{2}|\cdot|^2) \tag{4.4}$$

is the standard Lorentzian function, and

$$\begin{aligned} \mathbf{d_1} &= \mathbf{w}_r - \mathbf{w}_p \\ \mathbf{d_2} &= \mathbf{w}_p - 2\mathbf{w}_q + \mathbf{w}_r, \end{aligned} \tag{4.5}$$

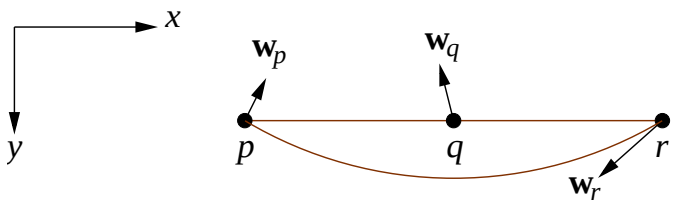


Figure 4.1: A horizontal 3-clique over pixels (nodes) p, q, and r. Each pixel has a 2-D vector label indicating the flow value (i.e. the apparent motion).

with $\mathbf{w}_p = (u_p, v_p)^T$ denoting the flow vector at pixel p. Here p, q, and r are the three pixels (i.e. nodes) belonging to the linear 3-clique c. If the clique c is horizontally oriented, they are the left, middle, and right pixel of the clique respectively and hence $\mathbf{d_1}$ and $\mathbf{d_2}$ are the discrete first and second partial derivatives (up to a constant factor) of the flow field along the horizontal direction; the case for clique c being vertically oriented is analogous.

Figure 4.1 illustrates the layout of a horizontal clique and the vector labels of its nodes. Notice that the spatial energy function (and hence the clique potential of the MRF) is symmetric with respect to x and y directions, and therefore its value is unchanged if the coordinates are rotated by 90° or have the two axes switched. The spatial energy function is also isotropic (i.e. has perfect rotational symmetry) in the motion domain, since only the *magnitude* of relative motion is computed. Thus rotating the flow vectors by the same angle everywhere would result in no change of the spatial energy.

While the form of our energy function is similar to filter based models such as [77, 78, 108], it differs in a subtle but important way. Those models use a linear combination of functions over individual filter responses, which implicitly assumes that the filters are independent of each other. In our case, both the first and second derivative filters are inputs to the same non-linear robust function.

Hence the influence of one derivative is reduced if the other is already large (due to the robust spatial term), thereby avoiding double-penalties at motion boundaries.

The energy function for the data term is defined in terms of the difference between a pixel in I_0 and its matching position in I_1 under the flow field $\mathbf{w}$,

$$g_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I) = \lambda_D \cdot \rho(\beta_D \| I_1(p + \mathbf{w}_p) - I_0(p) \|), \tag{4.6}$$

where p is used synonymously with its 2-vector coordinates $(x_p, y_p)^T$ on the image grid, ρ is the same Lorentzian function as defined in Equation 4.4, and λ_D and β_D are model parameters. For color images, $I(p)$ is simply a 3-vector of the RGB values at position p in image I. Although using more psychophysically motivated color spaces such as Lab or XYZ may yield better matching models, it is beyond the scope of this work. Since in general $p + \mathbf{w}_p$ does not fall on integer grid positions, its value in I_1 is sampled using bilinear interpolation.

4.2.2 Optical Flow Estimation

To estimate optical flow, we perform approximate MAP inference on the MRF by minimizing the energy function in Equation 4.2 using gradient descent. Computing the gradient of the spatial term energy $E_S = \sum_{c \in C} f_c^{\boldsymbol{\theta}}(\mathbf{w}_c)$ is straightforward since it has an analytical form in $\mathbf{w}$, and the gradient at each pixel is given by

$$(\nabla_{\mathbf{w}} E_S)_p = \sum_{c \in C : p \in c} \nabla_{\mathbf{w}_p} f_c^{\boldsymbol{\theta}}(\mathbf{w}_c). \tag{4.7}$$

Since the data term (Equation 4.6) involves the image input, it is not a closed-form expression with respect to $\mathbf{w}$. Nevertheless, by using the chain rule, the

gradient of the data term energy $E_D = \sum_{p \in \mathcal{V}} g_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I)$ can be written as

$$(\nabla_{\mathbf{w}} E_D)_p = \nabla_{I_1(p+\mathbf{w}_p)} g_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I) \nabla_{\mathbf{w}_p} I_1(p + \mathbf{w}_p). \tag{4.8}$$

The value of $\nabla_{I_1(p+\mathbf{w}_p)} g_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I)$ is readily available, since it is analytical in $I_1(p+\mathbf{w}_p)$ (cf. Equation 4.6). Moreover, $\nabla_{\mathbf{w}_p} I_1(p + \mathbf{w}_p)$ is simply the image gradient of I_1 at position $p + \mathbf{w}_p$. (To see this, let $\mathbf{z} = (x, y)^T = p + \mathbf{w}_p$, i.e. $\mathbf{z}$ is the coordinates of the matching position of p. Since $\nabla_{\mathbf{w}_p} \mathbf{z} = \mathbf{1}$ according to the definition of $\mathbf{z}$, it follows that $\nabla_{\mathbf{w}_p} I_1(p + \mathbf{w}_p) = \nabla_{\mathbf{z}} I_1(p + \mathbf{w}_p) \nabla_{\mathbf{w}_p} \mathbf{z} = \nabla_{\mathbf{z}} I_1(p + \mathbf{w}_p)$.) We approximate the image gradient using the $\frac{1}{2}(-1, 0, 1)$ derivative filters and bilinear interpolation.

As is standard in the literature, the input images are preprocessed with a low-pass filter. In our case, we use a small Gaussian kernel with $\sigma = 0.25$. For performing gradient descent we use limited memory BFGS [70, 63], which has faster converge speed than steepest descent. We also employ hierarchical coarse-to-fine strategy for optical flow computation [2, 9], since it is well known to produce globally more consistent and hence more accurate flow estimations. As is commonly done, all flow values are initialized to zero at the beginning of the optimization.

4.3 Learning the Parameters

As noted above, we take the approach of learning models that yield low training loss (e.g., [56]) rather than those that maximize the likelihood of the training data. There are several advantages to this approach. First, as discussed in Section 4.1 maximum likelihood estimation is intractable for labeling problems on large loopy graphs leading to the use of a number of approximation techniques. Second, maximum likelihood estimation may not be consistent with the error

measure that one would like to optimize, which we discuss further here. Finally, directly minimizing training loss is well suited to ground truth from real scenes, as opposed to synthetic data, which generally contain pixels at which the data is unknown.

We now turn to the second of these issues, training models that optimize for an appropriate error measure. If we regard the training data as a sample drawn from some unknown distribution characterizing the domain of the problem, then lower training loss implies lower expected generalization loss (i.e. error rate on unseen testing data) for a given class of models. The maximum likelihood estimate, other other hand, does not take the specific error metric into account. Thus even if the correct (zero error) output is assigned a high likelihood, it does not necessarily discriminate between bad outputs (i.e. those with high loss) and reasonably good ones (i.e. those that are not completely correct but nevertheless have low loss). For instance, suppose there are two different loss functions for optical flow, each designed to suit a different need. One of them heavily penalizes non-smooth flows in the uniform region but does not mind having blurred motion boundaries, whereas the other does just the opposite. Given this scenario, there is little reason to believe that the same model should be optimal for both loss functions. By minimizing the training loss one finds model parameters best suited to the particular loss function. Although in principle one could instead learn the parameters using maximum likelihood and then take the error metric into account during the inference stage, this would pose additional challenges of what optimization problem to solve that took both the model and the metric into account, and whether such a problem could be solved efficiently.

The third issue that is naturally handled by minimizing the training loss is that of incomplete ground-truth. Currently available ground-truth data of real, as opposed to synthetic, motion sequences contains a non-trivial number of pixels with unknown flow values due to phenomena such as occlusion. While this may be improved to some degree with the gathering of additional data, it is an inherent problem that some pixels will have unknown values. Handling pixels with missing ground-truth values is easy with pixel-based loss functions such as AEPE, as such pixels can simply be excluded from consideration. Computing the likelihood of data with missing values, on the other hand, is not so straightforward because of spatial interdependencies.

4.3.1 Training Loss Minimization Using SPSA

Thus we seek parameters $\boldsymbol{\theta} = (\beta_1, \beta_2, \beta_D, \lambda_S, \lambda_D)^T$ that minimize the average training loss $L(\boldsymbol{\theta})$. [2] As noted above we do this using SPSA [85]. SPSA is an iterative pseudo-gradient descent algorithm that updates its solution $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_i, \ldots, \hat{\theta}_m)^T$ at each step by

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k \hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k), \tag{4.9}$$

where a_k is the step size at iteration k and $\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ is the pseudo-gradient of the loss function L. The pseudo-gradient is obtained using two-sided simultaneous perturbation,

$$\left(\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)\right)_i = \frac{l(\hat{\boldsymbol{\theta}}_k + c_k\boldsymbol{\Delta}_k) - l(\hat{\boldsymbol{\theta}}_k - c_k\boldsymbol{\Delta}_k)}{2c_k\,(\boldsymbol{\Delta}_k)_i}, \tag{4.10}$$

[2] One could reduce the dimensionality of $\boldsymbol{\theta}$ by 1 by observing that only the ratio between λ_S and λ_D is relevant under MAP inference. This, however, has little effect on the learning process, since the dimensionality of the space of optimal solutions is also reduced. Thus we do not carry out this explicit reduction in our learning formulation.

where $l(\cdot)$ is some noisy measurement of the true loss $L(\cdot)$ ($l = L$ if the measurement is noise-free), $\mathbf{\Delta}_k$ is a user-defined m-dimensional random perturbation vector satisfying certain conditions [85], and c_k is a scalar factor. The gain sequences a_k and c_k both decrease over time, and are given by $a_k = a/(A + k)^\alpha$ and $c_k = c/k^\gamma$ [85]. In our case, we used the recommended values $(0.602, 0.101)$ for (α, γ) and set (A, a, c) to $(50, 5, 0.001)$ following the guidelines given in [86].

Since all the parameters in our model are some type of scale parameters and have a natural domain of $(0, \infty)$, we transform them into the logarithm space during learning so that $\hat{\boldsymbol{\theta}} = \log \boldsymbol{\theta} \in \mathbb{R}^m$. Although the most commonly used distribution for the random perturbation vector $\mathbf{\Delta}$ is the Bernoulli ± 1 for each component, we find that the Bernoulli distribution is somewhat overly restrictive on the possible directions of descent. Thus we instead sample each component of $\mathbf{\Delta}$ uniformly at random from the union of intervals $[-1 - \delta, -1 + \delta] \cup [1 - \delta, 1 + \delta]$ with $\delta = 0.99$. Note that the distribution has no probability mass at around zero, which is a condition that the perturbation vector is required to satisfy [85]. Since measuring training loss given parameters is deterministic and can be considered essentially noise-free, we require that the loss function (i.e. training error) decreases monotonically with time. Hence a solution $\hat{\boldsymbol{\theta}}_{k+1}$ is rejected, i.e. remains the same as $\hat{\boldsymbol{\theta}}_k$, if the loss $L(\hat{\boldsymbol{\theta}}_{k+1})$ is greater than $L(\hat{\boldsymbol{\theta}}_k)$. We also observe that most common types of parametric motions, such as affine transformation and divergence, result in large first derivative, but much smaller second derivative, of the optical flow field. Thus a large magnitude of the second derivative should reasonably produce more energy (hence lower probability) than that of the first derivative. To this end, we impose the constraint $\beta_2 \geq \beta_1$ to reflect this prior knowledge. If the constraint becomes violated during learning, we simply swap β_1 with β_2 and resume. This to some extent resembles a restart, a common

mechanism used by stochastic methods to depart from undesired local optima. Finally, we run the SPSA algorithm multiple times and choose from the solutions the one with the lowest training loss. This helps to reduce the variance in the performance of learnt parameters.

4.4 Experimental Results

To evaluate our method, we trained our model on the "other" data set [3] from the Middlebury optical flow web site [5] and tested its performance on the "eval" data set from the same web site. For learning, we use the average end-point error (AEPE) [71, 5] as the loss function. We initialized all parameters of the model to one and ran the SPSA algorithm for 300 iterations. The procedure was repeated 5 times (i.e. five models trained), and the model with the lowest training loss was chosen and used for testing on unseen data. [4] Among the multiple runs, the losses (i.e. training error) of the three best models are within 5% of each other while the other two have losses about 15% higher than the best model. This shows that the results obtained by SPSA is quite reliable, especially with multiple trials, given that the initial loss is many times higher. Figure 4.2 shows a plot of the training errors against the number of iterations for all five trials.

For evaluation, we report error rates in both average angular error (AAE) [7] and average end-point error (AEPE). Table 4.1 shows the performance of our learnt model on the eight sequences for flow evaluation from the Middle-

[3] Only sequences with ground-truth flow are used. We excluded the stereo sequence "Venus" from the training set.

[4] Note that the choice of the model is part of the learning procedure and is completely based on the training data, without any knowledge of the data on which the model is evaluated.

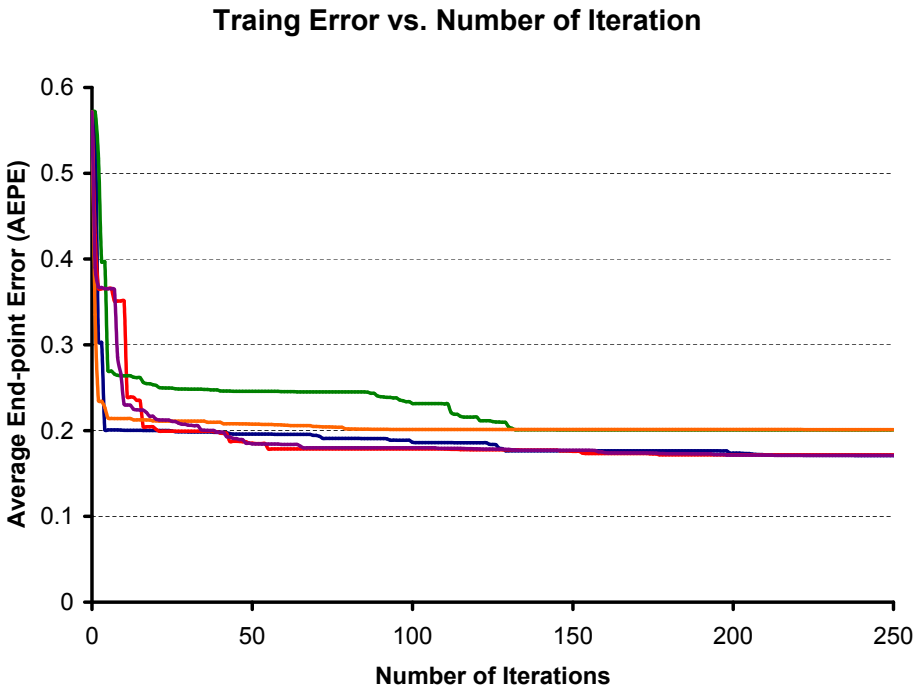


Figure 4.2: Average training error in terms of AEPE plotted against the number of iterations over multiple trials of running the SPSA algorithm. The error rate generally decreases rapidly at the beginning and much more slowly afterward, indicating a reasonably good solution can be obtained with relatively few iterations.

bury optical flow web page [5]. The error rates of some of the well-known methods are also shown for comparison. These results demonstrate that our model achieves state-of-the-art performance, surpassing the previous methods on most of the benchmark sequences. We want to emphasize that the performance of our model is achieved using parameters trained on a different set of sequences, which includes none of those used for evaluation. In other words, the parameters are learnt completely without any knowledge of the testing data. Thus the results demonstrate that our model has good generalization power. In addition we also trained our model using the approximate maximum likelihood scheme of [81] (second method in Table 4.1), so as to compare it with SPSA (first

Table 4.1: Performance on the eight evaluation sequences from the Middlebury optical flow page [5], measured in terms of both average angular error (upper row) and average end-point error (lower row). The lowest error rates for each sequence are shown in bold fonts.

Method\Sequence	Arm.	Meq.	Sch.	Woo.	Gro.	Urb.	Yos.	Ted.	**Avg.**
Our model	**6.84**	**8.47**	**12.5**	**8.40**	3.88	**6.32**	2.56	7.29	**7.03**
	0.18	**0.57**	**0.84**	**0.52**	**1.12**	1.75	0.13	**1.32**	**0.804**
- max-likelihood	6.86	9.11	15.1	8.60	**3.84**	10.1	2.15	10.3	8.26
	0.18	0.65	0.99	0.62	1.28	1.98	0.11	1.81	0.953
Bruhn *et al.* [16]	10.1	9.84	16.9	14.1	3.93	6.77	**1.76**	**6.29**	8.71
	0.28	0.69	1.12	1.07	1.24	1.56	**0.10**	1.38	0.930
Black/Anandan [13]	7.83	9.70	13.7	10.9	4.67	8.00	2.61	8.58	8.25
	0.21	0.65	0.93	0.76	1.40	2.04	0.15	1.68	0.978
Horn/Schunk [42]	8.01	9.13	14.2	12.4	4.69	8.35	4.01	9.16	8.74
	0.22	0.61	1.01	0.78	1.27	**1.42**	0.16	1.51	0.873
Lucas/Kanade [65]	13.9	24.1	20.9	22.2	18.9	22.0	6.41	25.6	19.25
	0.39	1.67	1.50	1.57	2.95	3.30	0.30	3.80	1.94

method in Table 4.1). The results show that the model learnt with SPSA has better overall performance, demonstrating the effectiveness of SPSA learning for optical flow.

For completeness, we show in Table 4.2 the error rates on the training sequences. Results from other methods are quoted from [5] whenever available. One can see that our training data includes only three sequences, which is in contrast with the several hundred used in [78]. The available training sequences are also significantly less comprehensive in terms of the variation of appearance and motion than are the test sequences. For instance the training data does not have any synthetic sequences, which do appear in the evaluation set. Thus learning with this limited training data is especially challenging. Nonetheless, our model obtained under such adverse circumstances performs well on unseen data.

Table 4.2: Error rates on the three training sequences, given in the form of AAE/AEPE, with "–" indicating result not available.

Method\Sequence	Dimetrodon	RubberWhale	Hydrangea
Our model	2.92/0.152	5.22/0.149	2.43/0.198
Bruhn *et al.*[16]	10.99/0.43	–	–
Black/Anandon [13]	9.26/0.35	–	–
Lucas/Kanade [65]	10.27/0.37	–	–

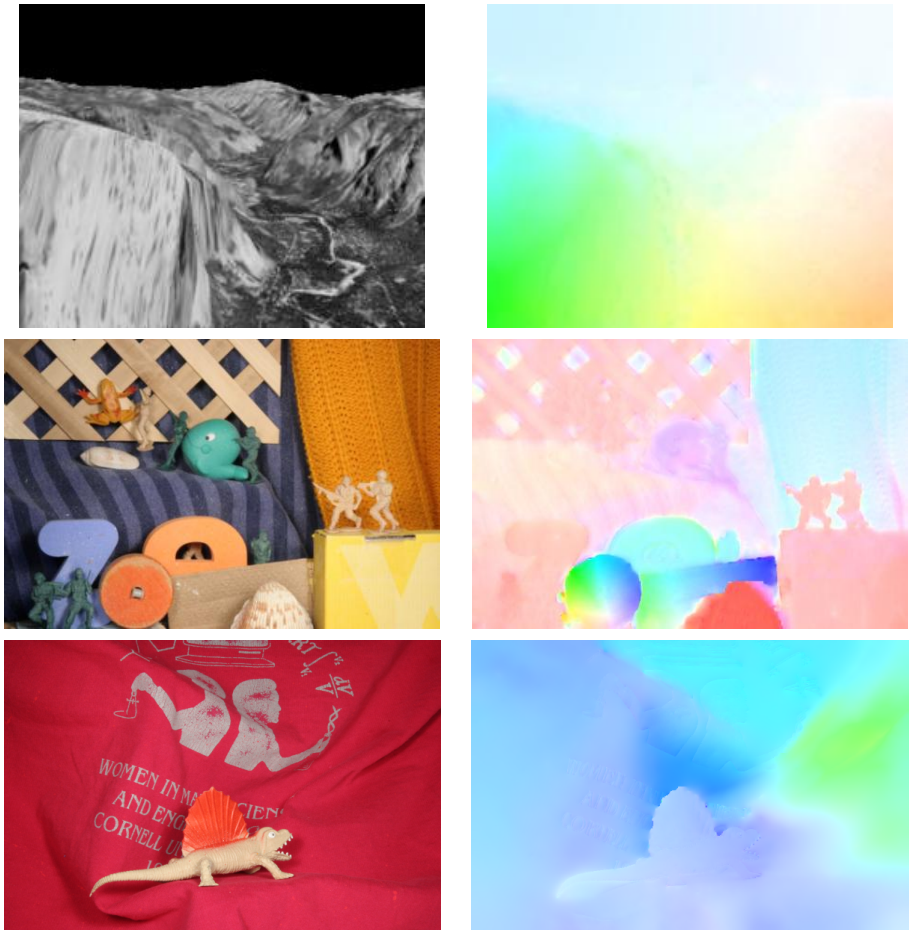


Figure 4.3: Output of our model for the sequences "Yosemite" (top), "Army" (middle), and "Dimetrodon" (bottom). Left: A frame of the image sequence. Right: Estimated flow. Observe the predominantly smooth flow field of "Yosemite" and "Dimetrodon" in contrast to the large amount of motion discontinuity in "Army".

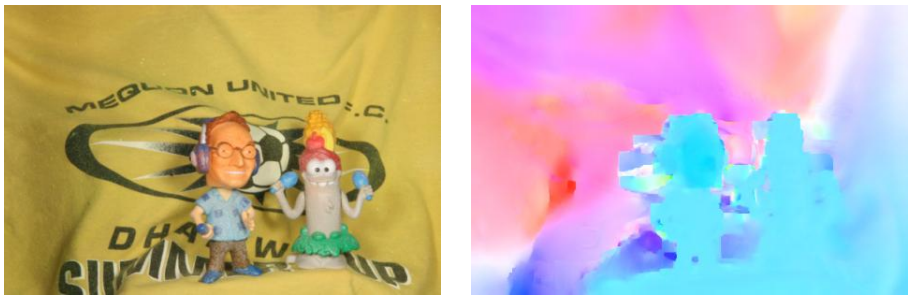


Figure 4.4: Output of our model for the sequence "Mequon". Left: A frame of the image sequence. Right: Estimated flow. Most of the errors occur in the shadows around the cartoon models, due to their high relative motion with respect to the background.

Figure 4.3 displays some sample flow fields produced by our model, color coded using the scheme described in [5]. It can be seen that our model is capable of producing smooth flows (as in "Yosemite") while preserving motion boundaries (as in "Army"). Figure 4.4 shows the result of a sequence on which our model did not perform particularly well. Most of the errors lie inside the shadows (cast by the two cartoon models), which have rather high motion relative to the background on which they lie. Since optical flow is generally defined as the *apparent* motion, the flow inside a shadowed region can be interpreted as either the motion of the background or that of the shadow itself. Thus this is an inherent ambiguity in optical flow, which also occurs with transparent and specular surfaces. A principled approach to dealing with these phenomena is to estimate the multiple motions in such areas. This has attracted a fair amount of investigation from researchers (e.g. [31, 13]), and remains an interesting topic for future work.

CHAPTER 5

LANDMARK CLASSIFICATION IN INTERNET IMAGE COLLECTIONS

5.1 Introduction

The billions of photographs in Internet-scale photo collections offer both exciting opportunities and significant challenges for computer vision, and for the area of object recognition in particular. Achieving Internet-scale object recognition and image classification is currently limited by the relatively small-scale datasets for which ground truth information is available. For instance, the widely-used PASCAL VOC 2008 dataset [26] has about 10,000 images and 20 categories, while the LabelMe dataset [80] is of similar size, with a larger hierarchically-organized label set. Bigger datasets such as Tiny Images [101] have millions of images but do not include category labels, whereas other datasets make use of visual features during image selection which may bias towards certain methods (e.g., [18, 83]). Recent work on scaling classification algorithms to Internet-sized datasets with millions of images (such as [102]) has thus been limited to evaluating classification performance on relatively small datasets such as LabelMe.

Here we consider image classification on much larger datasets featuring millions of images and hundreds of categories. First we develop a collection of over 30 million photos with ground-truth category labels for nearly 2 million of those images. The ground-truth labeling is done automatically based on geolocation information that is separate from the image content and the text tags that we use for classification. The key observations underlying our approach is that photos taken very near one another are likely to be of similar things. Moreover, if many

people have taken photos at a given location, there is a high likelihood that they are photographing some common area of interest, or what we call a *landmark*. Thus we use a mean shift [20] procedure to find peaks in the spatial distribution of geotagged photos, and then use large peaks to define the category labels. The photographs taken at these landmarks are typically quite diverse (see Figure 5.1 for some examples), so that the labeled test datasets are challenging, with significant amounts of visual variation and a large fraction of outliers. In most cases, a landmark does not consist of any one prominent object; for example, many of the landmarks are museums, in which the photos are distributed among hundreds of exhibits. Our landmark classification problem can thus be thought of as more similar to an object category recognition problem than to a specific object recognition problem. In Section 5.2 we discuss the details of our dataset collection approach and compare it to some alternative techniques.

We use multiclass support vector machines [21] to learn models for various classification tasks on this labeled dataset of nearly two million images. We use visual features based on clustering local interest point descriptors [64] into a visual vocabulary that is used to characterize the descriptors found in each image. We also explore using the textual tags that Flickr users assign to photos as additional features. The learning and classification methods and the feature extraction are discussed in more detail in Section 5.3.

Internet photo collections also include rich sources of relational information that can be helpful for classification. For instance, social ties have been found to improve face recognition performance on Facebook [89]. In this chapter, we consider the *photo stream* of a given photographer. In particular we model this sequence of category labels of a photo stream as a conditional random field

(CRF) [54], which is learnt using the structural support vector machine [104]. Feature extraction, learning, and classification methods using temporal context are discussed further in Section 5.4.

In Section 5.5 we present a set of large-scale classification experiments involving between 10 and 500 categories and tens to hundreds of thousands of photos (in contrast to other recent image recognition work which use large datasets but small test subsets). We find that the combination of image and text features performs better than either alone, even when we remove untagged photos from the dataset. We also describe a small study of human performance on landmark classification which suggests that a multiclass SVM using both image and text features performs nearly as well as people can. Finally we show that CRF models that expoit temporal context from photos taken by the same photographer nearby in time yields a striking improvement compared to using visual features alone — around 10 percentage points in most cases. On the other hand, the improvement using the textual tags from those same nearby photos is small.

5.1.1 Related Work

Image classification using bag-of-features models has been studied extensively (see [24] or [113] for recent surveys), however such previous work has been carried out only at much smaller scales. The work we report here uses two orders of magnitude more labeled photos – nearly two million photos as opposed to a few thousand in previous work – and one to two orders of magnitude more categories – up to 500 compared to tens in most previous work. This larger scale

allows us to study how performance is affected by the number of categories and the number of training images available. Our investigation also evaluates text tags versus image features, and considers the use of temporal context which has not received much attention in the literature.

Some recent work has used large datasets, but the number of *labeled* photos available for evaluating performance has usually been quite small. For instance [73] uses one million photos but only 5,000 of them have ground truth labels. The recent work of [102] considers a dataset with tens of millions of images, but only at thumbnail resolutions and again without labels for assessing classification accuracy. Another line of research uses small training sets to automatically label larger image sets (e.g., [18, 83, 115]), however such approaches generally make use of image features and machine learning techniques, and thus the resulting datasets are not independent of the kinds of features and methods that one wants to test. This raises the possibility that methods related to the ones used to create the dataset might be at an unfair advantage.

We also investigate how the visual vocabulary size affects classification performance. Although [109] presents a technique for finding the optimal visual vocabulary size for their task, it is not clear that their method can scale to large datasets because the running time is linear in the number of images and quadratic in the number of categories.

The work of [36] is related to our work in that it studies geolocating photographs, but their goal is quite different from ours, as we do not try to predict location but rather just use location to derive category labels. (For instance, in our problem formulation a misclassification with a geographically proximate category is just as bad as with one that is far away.) Our experiments use a stan-

dard classification paradigm and thus are comparable with many other studies. Moreover, the test set in [36] contains only 237 images that were partially selected by hand, making it difficult to generalize the results beyond that set. In contrast we use automatically-generated test sets that contain tens or hundreds of thousands of photos, providing highly reliable estimates of performance accuracy.

Some very recent studies have considered landmark classification tasks similar to the one we study here, but again have done so at a much smaller scale. For example, [58] studies how to build a model of a landmark by extracting a small set of iconic views from a large set of photographs. However it was tested on just three hand-chosen categories, making it unclear how well the method would scale to more realistic classification tasks. The very recent work of [115] is similar to our approach in that it finds highly-photographed landmarks automatically from a large collection of geotagged photos. However the test set they use is hand-selected and very small — 728 total images for a 124-category problem, or fewer than 6 test images per category — and their approach is based on nearest-neighbor search, which is unlikely to scale to the millions of test images we consider here. The recent of [22] on organizing large photo collections uses a dataset of geotagged photos similar to the one we describe here, however the focus of that work is on geographic embedding and organization of photos instead of image classification.

5.2 Building Internet-Scale Datasets

Our long-term goal is to create large publicly-available, labeled datasets that are representative of photos found on photo-sharing sites on the web. In constructing such datasets, it is critical to avoid potential biases either in selecting the images to include in the dataset or in assigning ground-truth labels. For instance, methods based on searching for photos tagged with hand-selected keywords (e.g., [36, 73]) are prone to bias because one might inadvertently choose keywords corresponding to objects that are amenable to a particular image classification algorithm. A number of previous collection efforts also use unspecified criteria to discard certain photos from the dataset, again introducing the potential for bias towards a particular algorithm. Also problematic is using the same kinds of features to produce ground-truth labels as are used by the classification algorithm (e.g., as in [18, 83, 115]). We thus advocate automatic techniques for creating datasets based on features that are independent from those used by the algorithms being tested. In our case, we avoid using textual tags or visual features to label or select images, instead using a completely separate source of information: geotags.

Our dataset was formed by using the Flickr API to retrieve metadata for over 60 million publicly-accessible geotagged photos. We eliminate photos for which the precision of the geotags (as reported in Flickr metadata) is worse than about a city block. For each of the remaining 30 million photos we consider the latitude-longitude coordinates as a point in the plane, and then perform a mean shift clustering procedure [20] on the resulting set of points to identify local peaks in the photo density distribution, as in [22]. The radius of the disc used in mean shift is about 100m. Since our goal is to identify locations where many

different people took pictures, we count at most 5 photos from any given Flickr user towards any given peak. We currently use the top 500 such peaks as categories; the number of photos becomes small for lower-ranked categories (e.g. the 500th largest peak has 585 photos whereas the 1000th largest peak has 284 photos). Figure 5.1 illustrates the top 10 categories in our dataset, corresponding to the ten most photographed landmarks.

We downloaded the image data for all 1.9 million photos known to our crawler that were geotagged within one of these 500 landmarks. For the experiments on classifying temporal photo streams, we also downloaded all images taken within 48 hours of any photo taken in a landmark, bringing the total number of images to about 6.5 million. The images were downloaded at Flickr's medium resolution level, which is about 1/4-megapixel. The total size of the dataset is just over one terabyte.

5.3 Single Image Classification

To perform image classification we adopt the bag-of-features model of [24]. We build a visual vocabulary by clustering SIFT descriptors from photos in the training set using the *k*-means algorithm. To make *k*-means clustering tractable on this quantity of data we use the approximate nearest neighbor (ANN) technique of [4] to efficiently assign points to cluster centers. The advantage of this technique is that it guarantees an upper bound on the approximation error, unlike other techniques that have recently been used for clustering such as randomized k-d trees [73]. In our implementation we set the bound such that the cluster center found by ANN is no further away than 110% of the distance be-

Landmark Most distinctive tag (bold) and Random tags	Random images
eiffeltower eiffel 1. city travel night	
trafalgarsquare london 2. summer july trafalgar	
bigben westminster 3. london ben night	
londoneye stone 4. cross london day2	
notredame 2000 5. portrait iglesia france	
tatemodern england 6. greatbritian thames streetart	
empirestatebuilding manhattan 7. newyork travel scanned	
venice tourists 8. slide venecia vacation	
colosseum roma 9. england stadium building	
louvre places 10. muséedulouvre eau paris	

Figure 5.1: The world's most photographed landmarks, and the first 10 categories of our dataset. We show the highest-frequency tag and 4 random tags, and 5 random images. The landmark tagged "venice" is Piazza San Marco (St. Mark Square).

tween the point and the optimal cluster center.

Once a visual vocabulary of size k has been generated, a k-dimensional feature vector is constructed for each image by using SIFT to find local interest points and assigning each interest point to the visual word with the closest descriptor. We then form a frequency vector which counts the number of occurrences of each visual word in the image. For textual features we use a similar vector space model in which any tag used by at least three different users is a dimension in the feature space, so that the feature vector for a photo is a binary vector indicating presence or absence of each text tag. Both types of feature vectors are normalized to have L2-norm of 1. We also study combinations of image and textual features, in which case the image and text feature vectors are simply concatenated.

We learn a linear model that scores a given photo for each category and assigns it to the class with the highest score. More formally, let m be the number of classes and $\mathbf{x}$ be the feature vector of a photo. Then the predicted label is

$$\hat{y} = \arg\max_{y \in \{1,\cdots,m\}} s(\mathbf{x}, y; \mathbf{w}), \tag{5.1}$$

where $\mathbf{w} = (\mathbf{w}_1^T, \cdots, \mathbf{w}_m^T)^T$ is the model and $s(\mathbf{x}, y; \mathbf{w}) = \left\langle \mathbf{w}_y, \mathbf{x} \right\rangle$ is the score for class y under the model. Note that in our settings, the photo is always assumed to belong to one of the m categories. Since this is by nature a multi-way (as opposed to binary) classification problem, we utilize the multiclass SVM [21] to learn the model $\mathbf{w}$, using the SVM^{multiclass} software package [47]. For a set of training examples $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$ the multiclass SVM optimizes the objective function

$$\begin{aligned} &\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i \\ &\text{s.t. } \forall i, y \neq y_i : \left\langle \mathbf{w}_{y_i}, \mathbf{x}_i \right\rangle - \left\langle \mathbf{w}_y, \mathbf{x}_i \right\rangle \geq 1 - \xi_i \end{aligned} \tag{5.2}$$

where C is the trade-off between training performance and margin in SVM formulations.[1] Hence for each training example, the learned model is encouraged to give higher score to the correct class label than to the incorrect ones. In fact, by simply rearranging terms it can be shown that the objective function is an upper bound on the training error.

In contrast, many previous approaches to object recognition using bag-of-parts models (such as [24]) train a set of binary SVMs (one for each category) and classify an image by comparing scores from the individual SVMs. Such approaches are problematic for n-way forced-choice problems, however, because the scores produced by a collection of independently-trained binary SVMs may not be comparable, and thus such approaches lack any performance guarantee. It is possible to alleviate this problem by using a different C value for each binary SVM (as is done in [24]), but this introduces additional parameters that need to be tuned, either manually or via a process such as cross validation.

Note that while the categories in this single-photo classification problem correspond to geographic locations, there is no geographical information used in the learning or classification. For example, unlike [36] we are not concerned with pinpointing a photo on a map, but rather with classifying images into discrete categories.

[1] For all our experiments, we simply set C to $1/\bar{x}^2$ where $\bar{x}$ is the average L2-norm of the training feature vectors.

5.4 Temporal Model for Joint Classification

Modern photo-sharing sites collect a rich set of metadata which is potentially useful for image classification tasks. For example, photos taken by the same photographer at nearly the same time are quite likely to be related. In the specific case of classifying landmarks, practical and physical constraints on human movement mean that certain sequences of category labels are much more likely than others. To learn the patterns created by such constraints, we view temporal sequences of photos taken by the same user as a single entity and label them jointly as a structured output.

We model a temporal sequence of photos as a CRF with a chain topology, where the nodes represent photos and edges connect nodes that are consecutive in time. The set of possible labels for each node is simply the set of m landmarks, indexed from 1 to m. The task is to label the entire sequence of photos with category labels, however we evaluate correctness only for a single selected photo in the middle of the sequence, with the remaining photos serving as temporal context for that photo. Denote an input sequence of length n as $X = ((\mathbf{x}_1, t_1), \cdots, (\mathbf{x}_n, t_n))$, where $\mathbf{x}_v$ is a feature vector for node v (encoding evidence about the photo such as textual tags or visual information) and t_v is the corresponding timestamp. Let $Y = (y_1, \cdots, y_n)$ be a labeling of the sequence. We would like to express the scoring function $S(X, Y; \mathbf{w})$ as the inner product of some *feature map* $\Psi(X, Y)$ and the model parameters $\mathbf{w}$, so that the model can be learned efficiently using the structural SVM.

5.4.1 Node Features

To this end, we define the feature map for a single node v under the labeling as,

$$\Psi_V(\mathbf{x}_v, y_v) = (I(y_v = 1)\mathbf{x}^T, \cdots, I(y_v = m)\mathbf{x}^T)^T, \tag{5.3}$$

where $I(\cdot)$ is an indicator function. Let $\mathbf{w}_V = (\mathbf{w}_1^T, \cdots, \mathbf{w}_m^T)$ be the corresponding model parameters with $\mathbf{w}_y$ being the weight vector for class y. Then the node score $s_V(\mathbf{x}_v, y_v; \mathbf{w}_V)$ is the inner product of the $\Psi_V(\mathbf{x}_v, y_v)$ and $\mathbf{w}_V$,

$$s_V(\mathbf{x}_v, y_v; \mathbf{w}_V) = \langle \mathbf{w}_V, \Psi_V(\mathbf{x}_v, y_v) \rangle . \tag{5.4}$$

5.4.2 Edge Features

The feature map for an edge (u, v) under labeling Y is defined in terms of the labels y_u and y_v, the time elapsed between the two photos $\delta t = |t_u - t_v|$, and the speed required to travel from landmark y_u to landmark y_v within that amount of time, $speed(\delta t, y_u, y_v) = distance(y_u, y_v)/\delta t$. Since the strength of the relation between two photos decreases with the elapsed time between them, we divide the full range of δt into M intervals $\Omega_1, \cdots, \Omega_M$. For δt in interval Ω_τ, we define feature vector

$$\psi_\tau(\delta t, y_u, y_v) = (I(y_u = y_v), I(speed(\delta t, y_u, y_v) > \lambda_\tau))^T, \tag{5.5}$$

where λ_τ is a speed threshold. This feature vector encodes whether the two consecutive photos are assigned the same label and, if not, whether the transition requires a person to travel at an unreasonably high speed (i.e. greater than λ_τ). The exact choices of the time intervals and the speed thresholds are not crucial, so long as they are sensible. We also take into consideration the fact that some

photos have invalid timestamps (e.g. a date in the 22nd century) and define the feature vector for edges involving such photos as,

$$\psi_0(t_u, t_v, y_u, y_v) = I(y_u = y_v)(I(z = 1), I(z = 2))^T, \tag{5.6}$$

where z is 1 if exactly one of t_u and t_v is invalid and 2 if both are. Here we no longer consider the speed, since it is not meaningful due to invalid timestamps. The complete feature map for an edge is thus,

$$\begin{aligned} \Psi_E(t_u, t_v, y_u, y_v) &= (I(\delta t \in \Omega_1)\psi_1(\delta t, y_u, y_v)^T, \cdots, \\ &\quad I(\delta t \in \Omega_M)\psi_M(\delta t, y_u, y_v)^T, \\ &\quad \psi_0(t_u, t_v, y_u, y_v)^T)^T \end{aligned} \tag{5.7}$$

and the edge score is,

$$s_E(t_u, t_v, y_u, y_v; \mathbf{w}_E) = \langle \mathbf{w}_E, \Psi_E(t_u, t_v, y_u, y_v) \rangle, \tag{5.8}$$

where $\mathbf{w}_E$ is the vector of edge parameters.

5.4.3 Overall Feature Map

The total score of the CRF with input sequence X under labeling Y and model $\mathbf{w} = (\mathbf{w}_V^T, \mathbf{w}_E^T)^T$ is simply the sum of individual scores over all the nodes and edges. Therefore, by defining the overall feature map as,

$$\begin{aligned} \Psi(X, Y) &= (\sum_{v=1}^{n} \Psi_V(\mathbf{x}_v, y_v)^T, \\ &\quad \sum_{v=1}^{n-1} \Psi_E(t_v, t_{v+1}, y_v, y_{v+1})^T)^T, \end{aligned}$$

the total score becomes an inner product with $\mathbf{w}$,

$$S(X, Y; \mathbf{w}) = \langle \mathbf{w}, \Psi(X, Y) \rangle. \tag{5.9}$$

The predicted labeling for sequence X by model $\mathbf{w}$ is one that maximizes the score,

$$\hat{Y} = \arg\max_{Y \in \mathcal{Y}_X} S(X, Y; \mathbf{w}), \tag{5.10}$$

where $\mathcal{Y}_X = \{1, \cdots, m\}^n$ is the the label space for sequence X of length n. This can be obtained efficiently using Viterbi decoding because the graph is acyclic.

Here we follow the convention in high-level vision to formulate the labeling problem as score maximization as opposed to energy minimization in low-level vision. Nevertheless, the two formulations are equivalent and only differs by a flip of sign.

5.4.4 Parameter Learning

The model parameters are learned using structural SVMs [104]. Let $((X_1, Y_1), \cdots, (X_N, Y_N))$ be the training examples. The structural SVM optimizes for parameters $\mathbf{w}$ by minimizing a quadratic objective function subject to a set of linear soft margin constraints,

$$\begin{aligned} &\min_{\mathbf{w}, \boldsymbol{\xi}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i \\ &\text{s.t. } \forall i, Y \in \mathcal{Y}_{X_i} : \langle \mathbf{w}, \delta\Psi_i(Y) \rangle \geq \Delta(Y_i, Y) - \xi_i, \end{aligned} \tag{5.11}$$

where $\delta\Psi_i(Y)$ denotes $\Psi(X_i, Y_i) - \Psi(X_i, Y)$ (thus $\langle \mathbf{w}, \delta\Psi_i(Y) \rangle = S(X_i, Y_i; \mathbf{w}) - S(X_i, Y; \mathbf{w})$) and the loss function $\Delta(Y_i, Y)$ in this case is simply the number of mislabeled nodes (photos) in the sequence. It is easy to see that the structural SVM degenerates into a multiclass SVM if every example has only a single node.

The difficulty of this formulation is that the label space $\mathcal{Y}_{X_i}$ grows exponentially with the length of the sequence X_i. Structural SVMs address this prob-

lem by iteratively minimizing the objective function using a cutting-plane algorithm, which requires finding the *most violated constraint* for every training exemplar at each iteration. Since the loss function $\Delta(Y_i, Y)$ decomposes into a sum over individual nodes, the most violated constraint,

$$\hat{Y}_i = \arg\max_{Y \in \mathcal{Y}_{X_i}} S(X_i, Y; \mathbf{w}) + \Delta(Y_i, Y), \tag{5.12}$$

can be obtained efficiently via Viterbi decoding in the same manner as making a prediction using the model.

5.5 Experiments

Figure 5.2 presents results for various classification experiments on our dataset of nearly 2 million images. For each of these experiments we evenly divided the dataset into test and training image sets that are disjoint by photographer, so that duplicate photos taken by the same user could not appear during both training and testing. To make classification results easier to interpret across different categories with differing numbers of images, we constructed the test and training datasets by sampling the same number of images from each category. In practice this means that the number of images used in an m-way classification experiment is equal to m times the number of photos in the least popular of the m landmarks, and the baseline probability of a correct random guess is $1/m$.

We see from Figure 5.2 that in classifying single images (as described in Section 5.3), the visual features are less accurate than textual tags but nevertheless significantly better than random baseline — four to six times higher for the 10 category problems and nearly 50 times better for the 500-way classification. The combination of textual tags and visual tags performs significantly higher

Categories	Baseline	Single images			Photo streams		
		visual	textual	combined	visual	textual	combined
Top 10 landmarks	10.00	57.55	69.25	80.91	68.82	70.67	82.54
Landmarks 200-209	10.00	51.39	79.47	86.53	60.83	79.49	87.60
Landmarks 400-409	10.00	41.97	78.37	82.78	50.28	78.68	82.83
Top 20 landmarks	5.00	48.51	57.36	70.47	62.22	58.84	72.91
Landmarks 200-219	5.00	40.48	71.13	78.34	52.59	72.10	79.59
Landmarks 400-419	5.00	29.43	71.56	75.71	38.73	72.70	75.87
Top 50 landmarks	2.00	39.71	52.65	64.82	54.34	53.77	65.60
Landmarks 200-249	2.00	27.45	65.62	72.63	37.22	67.26	74.09
Landmarks 400-449	2.00	21.70	64.91	69.77	29.65	66.90	71.62
Top 100 landmarks	1.00	29.35	50.44	61.41	41.28	51.32	62.93
Top 200 landmarks	0.50	18.48	47.02	55.12	25.81	47.73	55.67
Top 500 landmarks	0.20	9.55	40.58	45.13	13.87	41.02	45.34

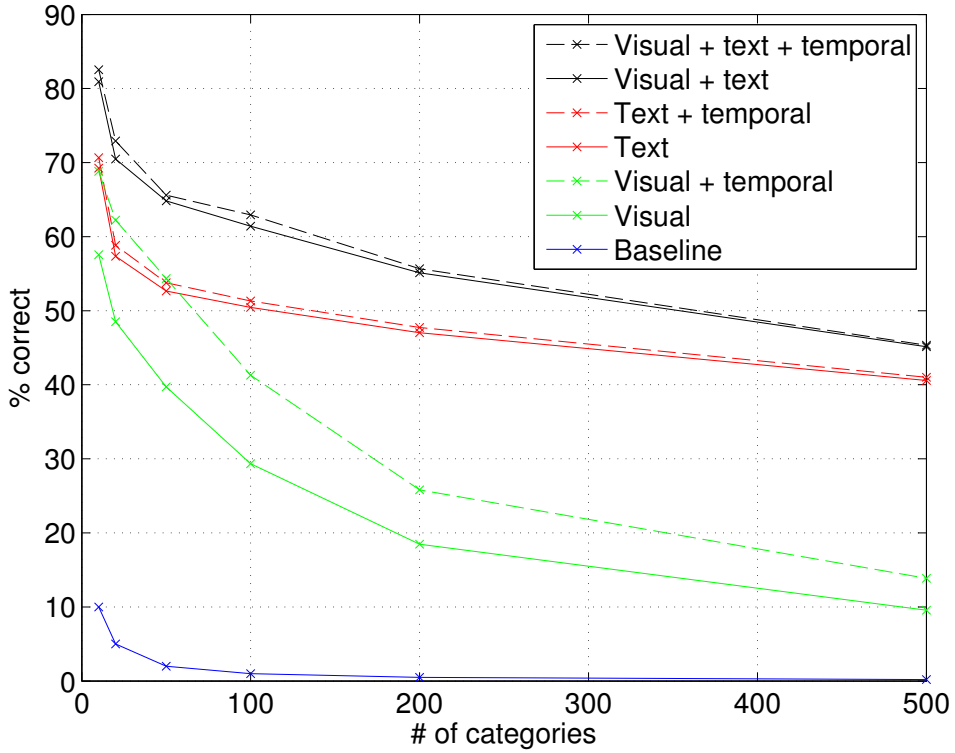


Figure 5.2: Percentage of images correctly classified for varying numbers of categories and combinations of features.

than either alone, increasing performance by about 10 percentage points in most cases. This performance improvement is partially because about 15% of photos in the dataset do not have any textual tags. However even when such photos are excluded from the evaluation, adding visual features still gives a significant improvement over using text tags alone, increasing accuracy from 79.2% to 85.47% in the top-10 category case, for example.

The figure also shows a dramatic improvement in visual classification performance when photo streams are classified jointly using a structural SVM (as described in Section 5.4) — nearly 12 percentage points for the top-10 category problem, for example. In contrast, the temporal information provides little improvement for the textual tags, suggesting that tags from contemporaneous images contain largely redundant information. In fact, the classification performance using temporal and visual features is actually slightly higher than using temporal and textual features for the top-20 and top-50 classification problems. For all of the experiments, the best performance is achieved using the full combination of visual, textual and temporal features, which gives for example 82.54% correct classification for the 10-way problem and 45.34% for the 500-way problem — more than 220 times better than the baseline! For these experiments, the maximum length of a photo stream was limited to 11, or five photos before and after a photo of interest.

Figure 5.2 shows classification experiments for different numbers of categories and also for categories of different rank. For the textual features, problems involving higher-ranked categories are more difficult; for example, the performance on classifying landmarks ranked 1 through 10 is about 10 percentage points worse than for landmarks 200 through 209. This is because the top land-

marks are mostly located in a small set of cities including Paris, London, and New York, so that textual tags like "london" are relatively uninformative. On the other hand, classification using visual cues is significantly better for higher-ranked landmarks, probably because higher-ranked categories have more training images (e.g., 1,829 per category for the top 20 categories vs. 542 per category for 400-419).

A substantial number of Flickr photos are mislabeled or inherently ambiguous — a close-up photo of a dog or a sidewalk could have been taken at almost any landmark. To try to gauge the frequency of such difficult images, we conducted a small-scale human subject study. We asked 20 well-traveled people to each label 50 photos taken at the world's top ten landmarks. Textual tags were also shown for a random subset of the photos. We found that the average human classification accuracy was 68.0% without textual tags and 76.4% when both the image and tags were shown (with standard deviations of 11.61 and 11.91, respectively). Thus the humans performed better than the automatic classifier when using visual features alone (68.0% versus 57.55%) but about the same when both text and visual features were available (76.4% versus 80.91%).

For most of the experiments shown in Figure 5.2, the visual vocabulary size was set to 20,000. This size was computationally prohibitive for our (single-threaded) structural SVM learning code for the 200- and 500-class problems, so for those tasks we instead used 10,000 and 5,000, respectively. An interesting question is how the vocabulary size impacts classification performance on large-scale image sets. To study this we repeated a subset of the experiments for several different vocabulary sizes. As Table 5.1 shows, classification performance improves as the vocabulary size increases, but the relative effect is more

Table 5.1: Visual classification rates for different vocabulary sizes.

# of categories	Single images				
	1,000	**2,000**	**5,000**	**10,000**	**20,000**
10	47.51	50.78	52.81	55.32	57.55
20	39.88	41.65	45.02	46.22	48.51
50	29.19	32.58	36.01	38.24	39.71
100	19.77	24.05	27.53	29.35	30.42

pronounced as the number of categories increases. For example, when the vocabulary size is increased from 1,000 to 20,000, the relative performance of the 10-way classifier improves by about 20% (10.05 percentage points, or about one baseline) while the accuracy of the 100-way classifier increases by more than 50% (10.65 percentage points, or nearly 11 times the baseline). We found that performance on the 10-way problem asymptotes by about 80,000 clusters at about 59.3%. Unfortunately we could not try such large numbers of clusters for the other tasks because the learning becomes computationally challenging; studying how to efficiently learn structural SVMs with such large feature vectors would be an interesting area for future work.

In the experiments presented so far we sampled from the test and training sets to produce equal numbers of photos for each category, in order to make the empirical results easier to interpret. However our approach and results do not depend on this property of the experimental setup; when we sample from the actual photo distribution our techniques still perform dramatically better than the baseline (which is to guess the most frequent category). For example, in the top-10 category classification problem using the actual photo distribution we achieve 53.58% accuracy with visual features and 79.40% when tags are also used, versus a baseline of 14.86%; the 20-way classifier produces 44.78% and

69.28% respectively, versus a baseline of 8.72%.

The experimental results we report here are highly precise because of the large size of our test dataset. Even the smallest of the experiments, the top-10 classification, involves about 35,000 test images. To give a sense of the variation across runs due to differences in sampling, we ran 10 trials of the top-10 classification task with different samples of photos and found the standard deviation to be about 0.15 percentage points. Due to computational constraints we did not run multiple trials for the experiments with large numbers of categories, but the variation is likely even less due to the larger numbers of images involved.

Image classification on a single 2.66 GHz processor takes about 2.4 seconds, most of which is consumed by SIFT interest point detection. Once the SIFT features are extracted, classification requires only approximately 3.06 ms for 200 categories and 0.15 ms for 20 categories. SVM training times varied by the number of categories and the number of features, ranging from less than a minute on the 10-way problems to about 72 hours for the 500-way structural SVM on a single CPU. We conducted our experiments on a small cluster of 60 nodes running the Hadoop open source map-reduce framework.

CHAPTER 6

SUMMARY AND DISCUSSION

In this dissertation, we have investigated several aspects of learning random field models ranging from graph topology to parameter estimation. These studies were done in the context of real-world problems in both low-level and high-level vision.

We first presented in Chapter 2 a model which explicitly represents long-range interactions but only uses low-order cliques, thereby enabling much faster optimization than other approaches that rely on high-order cliques. For image denoising this model achieves state-of-the-art PSNR results among random field methods, is better at preserving fine-scale detail, and runs at least an order of magnitude faster. The low complexity nature of the model not only reduces artifacts such as ringing, but also makes it readily interpretable and easy to understand. The small clique size enables the use of efficient approximate global inference algorithms for arbitrary clique potentials, whilst the explicit long-range interactions effectively counters noise. The combination of speed and expressiveness makes it an efficient and robust approach for low-level vision problems in noisy domains.

In Chapter 3, we described a technique for learning random field based non-parametric models for stereo using the structural support vector machine. Experiments illustrate that our method achieves significantly better performance than previous learning approaches and moreover is capable of explicitly labeling occlusion. We also found that models with long-range interactions generally outperform the grid model, which has only local connections; the performance gap becomes more evident as the noise level increases, reaffirming the findings

in the previous chapter. Though only applied to stereo, the model is formulated as a generic random field labeling problem and the learning algorithm makes few assumptions specific to stereo. As such, it can be adapted to other low-level vision problems as well

In Chapter 4, we described a Markov random field based model for estimating optical flow and a technique for learning its parameters using simultaneous perturbation stochastic approximation. Experiments on publicly available benchmark data sets show that our results compare favorably with previous methods and achieve the state-of-the-art performance. This demonstrates that our learnt model generalizes well to unseen data. Since many low-level vision problems involve parameters that are difficult to optimize deterministically, the learning approach that we employed here may well prove useful in other research areas.

In Chapter 5, we studied the image classification problem for internet photo collections. Our experiments demonstrate that multiclass SVM classifiers using SIFT-based bag-of-word features achieve good classification rates for large-scale problems, with accuracy that in some cases is comparable to that of humans on the same task. Moreover the stream of photos taken by the same photographer can be modeled as a conditional random field, which can be learning using the structural SVM. We show that classifying photos jointly in this way, rather than classifying them independently, yields dramatic improvement in the classification rate.

6.1 Future Work

While the work described in Chapter 5 is a step towards utilizing random field models for high-level vision, potentially much more could still be done in this direction. At present, features, objects, and images are too often treated as independent entities, while their spatial relationships are ignored. Fresh opportunities are there to advance the state of the art, if we can harness the power of such spatial priors.

6.1.1 Features

The histogram of orientated gradient (HOG) [25] features have become one of the most successful methods in object recognition in the recent years. In this approach, the image is transformed into a multi-dimensional grid of feature values based on local image gradient information. These features are often used in conjunction with the support vector machine (SVM) [105, 47], where an object model is learnt to classify each subwindow of the HOG grid of the input image (e.g. [25, 28]). In the case of a linear classifier, the model has the same grid structure and dimensions as the subwindows it is trained to classify. Thus it can be regarded as an object template and the classification score reflects how well a given subwindow matches the object template.

In the typical settings of a linear SVM, the object template is simply treated a vector (with a quadratic regularization on its norm), which effectively ignores any spatial structure. This has worked well in practice and has become more or less a standard approach. However if we go back one step and think about it

closely, we should realize that a lot of information is lost here. Objects of interest typically have geometric shapes, which emerge as consistent contours in the image gradient space. Therefore, all object templates are not equally probable *a priori*. Templates that capture such intrinsic characteristics of objects should be judged as more plausible than those that don't. For instance if the model template has a vertical edge at location (x, y), its desirability should increase if it also has a vertical edge at $(x, y-1)$ or $(x, y+1)$ since this extends the contour. Therefore we can treat the template as a random field with just a spatial term, whose energy function reflects such prior belief. This can be incorporated into the learning framework as a hyperprior. For a linear SVM the energy function of the random field is simply added as an additional regularization term to the object function, which can still be minimized with gradient descent in the primal space. Moreover, the global optimum is still attainable if the energy function is convex.

We have performed some preliminary experiments with this framework on the INRIA pedestrian data set [25], observed some encouraging results. The use of random field regularization significant improved classification accuracy when we used 4×4 HOG cells. Although such improvement was not observed with 8×8 cells, we believe this is due to the low resolution nature of the images and hence the reduced significance of the contour consistency assumption in a coarse grid. We plan to carry out further investigation on this subject in the future, which we hope will verify our hypothesis and provide insight into this class of problems.

6.1.2 Object Locations

Similar to the values in a feature template, object appearances in an image are not independence either. Objects of the same categories (e.g. cars) as well as some objects of certain different categories (e.g. persons and bikes) tend to co-occur with each other. Moreover, they often co-occur with certain relative spatial locations (e.g. cars next to each other). However most object detection methods, include many of the best performing ones (e.g., [28]), use a sliding window approach, which scores each detection window (i.e. a subwindow of the image corresponding to a certain location and scale) independently of each other. In the recent years, researchers have become increasingly aware of the limitation of treating object detection at each location and scale as independent tasks and models that utilize contextual information have been proposed in a number of papers (e.g. [23, 41, 67, 103, 37]). In addition to using image context, it is also possible to use random fields to model the presence of objects at every location and scale jointly. This will increase the expressiveness of the model by encoding direct interactions between objects that are not captured in the existing contextual models. Before we can build this powerful model, of course, there are quite a few challenges we must face: What is a suitable graph topology, given the overlapping and multi-scale nature of object locations and sizes? What is a appropriate spatial prior, and how to learn it? How to keep inference fast yet accurate? These are the open research problems that we are yet to study, and we believe that solving them will have a major impact in the field of object detection.

BIBLIOGRAPHY

[1] Luis Alvarez, Joachim Weickert, and Javier Sanchez. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56, 2000.

[2] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *IJCV*, 2(3):283–310, January 1989.

[3] Dragomir Anguelov, Ben Taskar, Vassil Chatalbashev, Daphne Koller, Dinkar Gupta, Geremy Heitz, and Andrew Ng. Discriminative learning of Markov random fields for segmentation of 3D scan data. In *CVPR*, 2005.

[4] Sunil Arya and David M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *ACM-SIAM Symposium on Discrete Algorithms*, 1993.

[5] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007.

[6] Jasmine Banks, M. Bennamoun, and Peter Corke. Non-parametric techniques for fast and robust stereo matching. In *TENCON*, 1997.

[7] John L. Barron, David J. Fleet, and Steven S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994.

[8] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, 37:1554–1563, 1966.

[9] James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. In *ECCV*, pages 237–252, London, UK, 1992. Springer-Verlag.

[10] Mario Bertero, Tomaso Poggio, and Vincent Torre. Ill-posed problems in early vision. In *Proceedings of IEEE*, 1988.

[11] Julian E. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Stat. Soc., B*, 36(2), 1974.

[12] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *ICCV*, 1998.

[13] Michael J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *CVIU*, 63(1):75–104, 1996.

[14] Endre Boros and Peter L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123:155–225, 2002.

[15] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *ICCV*, 1999.

[16] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *IJCV*, 61(3):211–231, 2005.

[17] Isaac Cohen. Nonlinear variational method for optical flow computation. In *Scandinavian Conference on Image Analysis*, 1993.

[18] Brendan Collins, Jia Deng, Kai Li, and Li Fei-Fei. Towards scalable dataset construction: An active learning approach. In *ECCV*, 2008.

[19] Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *EMNLP*, 2002.

[20] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[21] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2, 2001.

[22] David Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. Mapping the world's photos. In *WWW*, 2009.

[23] David J. Crandall and Daniel P. Huttenlocher. Composite models of objects and scenes for category recognition. In *CVPR*, 2007.

[24] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cedric Bray. Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[25] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[26] Mark Everingham, Luc Van Gool, Chris K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL voc 2008. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[27] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1), 2006.

[28] Pedro F. Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[29] Thomas Finley and Thorsten Joachims. Parameter learning for loopy Markov random fields with structural support vector machines. In *ICML Workshop on Constrained Optimization and Structured Output Spaces*, 2007.

[30] David J. Fleet, Michael J. Black, Yaser Yacoob, and Allan D. Jepson. Design and use of linear models for image motion analysis. *IJCV*, 36(3):171–193, 2000.

[31] David J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *IJCV*, 5(1):77–104, 1990.

[32] William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. Learning low-level vision. *IJCV*, 40(1), 2000.

[33] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *PAMI*, 6(6):721–741, November 1984.

[34] Stuart Geman and Christine Graffigne. Markov random field image models and their applications to computer vision. In *Intl. Congress of Mathematicians*, 1986.

[35] Georgy L. Gimel'farb. Texture modeling by multiple pairwise pixel interactions. *PAMI*, 18(11), 1996.

[36] James Hays and Alexei A. Efros. IM2GPS: Estimating geographic information from a single image. In *CVPR*, 2008.

[37] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.

[38] Yong Seok Heo, Kyoung Mu Lee, and Sang Uk Lee. Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution. In *CVPR*, 2007.

[39] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8), 2002.

[40] Heiko Hirschmüller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, 2007.

[41] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. In *CVPR*, 2006.

[42] Berthold K. P. Horn and Brian G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[43] Jinggang Huang and David Mumford. Statistics of natural images and models. In *CVPR*, 1999.

[44] Hiroshi Ishikawa. Exact optimization for Markov random fields with convex priors. *PAMI*, 25(10):1333–1336, 2003.

[45] Hiroshi Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, 2009.

[46] Allan Jepson and Michael J. Black. Mixture models for optical flow computation. In *CVPR*, pages 760–761, 1993.

[47] Thorsten Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, 1999.

[48] Jack Kiefer and Jocob Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annuals of Mathematical Statistics*, 23(3):462–466, 1952.

[49] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, 2006.

[50] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In *Symposium on Foundations of Computer Science*, 1999.

[51] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10), 2006.

[52] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? In *ECCV*, 2002.

[53] Dan Kong and Hai Tao. A method for learning matching errors in stereo computation. In *BMCV*, 2004.

[54] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[55] Xiangyang Lan, Stefan Roth, Daniel P. Huttenlocher, and Michael J. Black. Efficient belief propagation with learned higher-order Markov random fields. In *ECCV*, 2006.

[56] Yann LeCun and Fu Jie Huang. Loss functions for discriminative training of energy-based models. In *AIStats*, 2005.

[57] Victor Lempitsky, Carsten Rother, Stefan Roth, and Andrew Blake. Fusion moves for Markov random field optimization. Technical Report MSR-TR-2009-60, Microsoft Research, 1999.

[58] Xiaowei Li, Changchang Wu, Christopher Zach, Svetlana Lazebnik, and Jan-Michael Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, 2008.

[59] Yunpeng Li, David J. Crandall, and Daniel P. Huttenlocher. Landmark classification in large-scale image collections. In *ICCV*, Kyoto, Japan, 2009.

[60] Yunpeng Li and Daniel P. Huttenlocher. Learning for optical flow using stochastic optimization. In *ECCV*, Marseille, France, 2008.

[61] Yunpeng Li and Daniel P. Huttenlocher. Learning for stereo vision using the structured support vector machine. In *CVPR*, Anchorage, AK, USA, 2008.

[62] Yunpeng Li and Daniel P. Huttenlocher. Sparse long-range random field and its application to image denoising. In *ECCV*, Marseille, France, 2008.

[63] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(3, (Ser. B)):503–528, 1989.

[64] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[65] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[66] David R. Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[67] Kevin Murphy, Antonio Torralba, and William T. Freeman. Using the forest to see the trees: A graphical model relating features, objects, and scenes. In *NIPS*, 2003.

[68] Kevin Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, 1999.

[69] Hans-Hellmut Nagel and Wilfried Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *PAMI*, 8(5):565–593, 1986.

[70] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980.

[71] Michael Otte and Hans-Hellmut Nagel. Optical flow estimation: Advances and comparisons. In *ECCV*, pages 51–60, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[72] Judea Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *AAAI*, 1982.

[73] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2008.

[74] Javier Portilla. Blind non-white noise removal in images using gaussian scale mixtures in the wavelet domain. In *Benelux Signal Processing Symposium*, 2004.

[75] Javier Portilla, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Imag. Proc.*, 12(11), 2003.

[76] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[77] Stefan Roth and Michael J. Black. Fields of experts: A Framework for Learning Image Priors. In *CVPR*, 2005.

[78] Stefan Roth and Michael J. Black. On the spatial statistics of optical flow. *IJCV*, 74(1):33–50, 2007.

[79] Stefan Roth and Michael J. Black. Steerable random fields. In *ICCV*, 2007.

[80] Bryan C. Russel, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[81] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *CVPR*, 2007.

[82] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3), 2002.

[83] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Harvesting image databases from the web. In *ICCV*, 2007.

[84] Eero P. Simoncelli, Edward H. Adelson, and David J. Heeger. Probability distributions of optical flow. In *CVPR*, pages 310–315, Mauii, Hawaii, 1991. IEEE Computer Society.

[85] James C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37:332–341, 1992.

[86] James C. Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans. Aerosp. Electon. Syst.*, 34(3):817–823, 1998.

[87] James C. Spall. Overview of the simultaneous perturbation method for efficient optimization. *Hopkins APL Technical Digest*, 19:482–492, 1998.

[88] A. Srivastava, A. Lee, E. Simoncelli, and S. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 2003.

[89] Zak Stone, Todd Zickler, and Trevor Darrell. Autotagging Facebook: Social network context improves photo annotation. In *1st IEEE Workshop on Internet Vision*, 2008.

[90] Christoph Strecha, Rik Fransens, and Luc Van Gool. Combined depth and outlier estimation in multi-view stereo. In *CVPR*, pages 2394–2401, Washington, DC, USA, 2006. IEEE Computer Society.

[91] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005.

[92] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *PAMI*, 25(7), 2003.

[93] Charles Sutton and Andrew McCallum. Piecewise training of undirected models. In *UAI*, 2005.

[94] Charles Sutton and Tom Minka. Local training and belief propagation. Technical Report TR-2006-121, Microsoft Research, 2006.

[95] Richard Szeliski. Bayesian modeling of uncertainty in low-level vision. *IJCV*, 5(3), 1990.

[96] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall F. Tappen, and Carsten Rother. A comparative study of energy minimization methods for Markov random fields. In *ECCV*, 2006.

[97] Marshall F. Tappen. Utilizing variational optimization to learn Markov random fields. In *CVPR*, 2007.

[98] Marshall F. Tappen and William T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *ICCV*, 2003.

[99] Marshall. F. Tappen, Ce. Liu, Edward. H. Adelson, and William. T. Freeman. Learning gaussian conditional random fields for low-level vision. In *CVPR*, 2007.

[100] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin Markov networks. In *NIPS*, 2003.

[101] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large dataset for non-parametric object and scene recognition. *PAMI*, 30(11):1958–1970, 2008.

[102] Antonio Torralba, Rob Fergus, and Yair Weiss. Small codes and large databases for recognition. In *CVPR*, 2008.

[103] Antonio Torralba, Kevin Murphy, William T. Freeman, and Mark A. Rubin. Context-based vision system for place and object recognition. In *ICCV*, 2003.

[104] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[105] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[106] Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[107] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching. In *AIStats*, 2003.

[108] Yair Weiss and William F. Freeman. What makes a good model of natural images? In *CVPR*, 2007.

[109] John Winn, Antonio Criminisi, and Tom Minka. Object categorization by learned universal visual dictionary. In *ICCV*, 2005.

[110] Qingxiong Yang, Liang Wang, Ruigang Yang, Henrik Stewénius, and David Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In *CVPR*, 2006.

[111] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized belief propagation. In *NIPS*, 2000.

[112] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, 1994.

[113] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2):213–238, June 2007.

[114] Li Zhang and Steven M. Seitz. Parameter estimation for MRF stereo. In *CVPR*, 2005.

[115] Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the world: Building a web-scale landmark recognition engine. In *CVPR*, 2009.

[116] C. Lawrence Zitnick and Sing Bing Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 75(1):49–65, 2007.