

Survey Paper

Fundamental Strategies for Solving Low-Level Vision Problems

MARSHALL F. TAPPEN^{†1}

Low-level vision encompasses a wide variety of problems and solutions. Solutions to low-level problems can be broadly group according to how they propagate local information to global representations. Understanding these categorizations is useful because they offer guidance on how tools like machine learning can be implemented in these systems.

1. Introduction

Computer vision tasks can be roughly divided into the categories of low-level, mid-level, and high-level vision. These categories can be understood in the context of a model of visual processing shown in **Fig. 1**, which is based on Fig. 1 in Ref. 46). In this model, the processing culminates in a high-level, object-centered representation, with the first step starting with low-level processing that extracts scene and surface properties, such as depth, orientation, texture, and shading, that will be used to identify the objects.

Given the breadth of problems that could be considered low-level vision problems, it is difficult to survey the entire field. Instead, this paper will focus on identifying three general strategies that can be used to solve a wide-variety of problems. The discussion will also include the incorporation of machine learning into these systems.

The three general strategies are:

- (1) Systems utilizing only local information,
- (2) Systems that perform global processing via hand-specified local relationships,
- (3) Low-level vision systems that perform global processing, again using lo-

cal relationships, but using machine-learning to extract relationship from training examples.

These strategies do not represent vastly different approaches, but instead build upon each other. The key differences in these strategies lie in how the inter-relationships between the values being estimated are modeled.

A benefit of viewing low-level vision problems from the perspective of these three categories is that it helps clarify the design of the system. This is particularly useful when machine learning techniques are part of the solution. Machine learning offers a number of powerful tools for building systems that can cope with the complexity of real-world imagery. Each of these tools is designed for modeling particular types of relationships. Understanding the type of behavior that is desired in the low-level vision system will help facilitate the choice of mathematical models and algorithms.

The description of these strategies will be motivated by three different problems: lightness recovery, optical flow, and image enhancement. Learning is straightforward to implement in the first two strategies, so the learning discussion for these strategies, described in Sections 4.4 and 5.5, will be brief. However, the topic of learning local relationships in data, covered in Section 6, will be the focus of more discussion. The description of the final two strategies will largely focus on Markov Random Field (MRF) models because they offer a straightforward model for implementing the desired behavior, though Section 7 will discuss alternative approaches. It is useful to focus on the Markov Random Field model because many of the issues raised in learning MRF models can be applied to other models.

2. Low-Level Vision

Before turning to actual models, it is helpful to consider what is meant by the term *low-level* vision. After the discussion in this section, Section 3 will begin the discussion of the three strategies described above.

As pointed out in Ref. 46), low-level visual processing can be thought of as extracting intrinsic images of the scene. In an intrinsic image representation, first proposed in Ref. 4), intrinsic properties of the scene are represented using an image for each characteristic. Useful characteristics suggested in Ref. 4) in-

^{†1} University of Central Florida

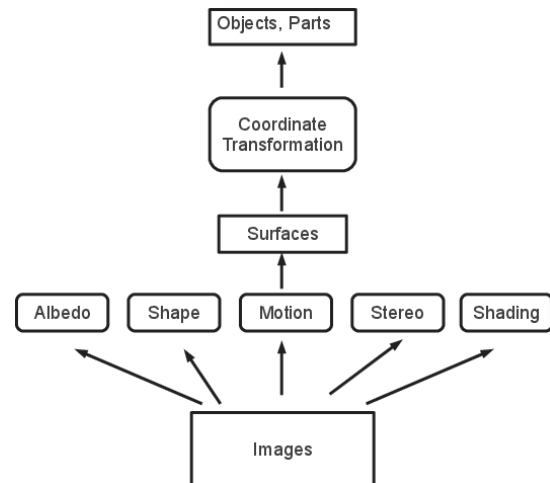


Fig. 1 Low-level vision can be seen as the first step in a hierarchy of tasks shown here, which is similar to Fig. 1 from Ref. 46). However, low-level vision can also be seen as an important problem in itself.

clude distance, reflectance, orientation, and illumination. The pixel values in each intrinsic image encode the value of that characteristic at the location of the corresponding point in the scene. Thus, an intrinsic image representing orientation will contain pixels encoding surface normal orientations, while the pixels in an intrinsic image expressing reflectance could denote the albedo of different points on the surface.

In this framework, low-level processing can be seen as a preliminary step on the path to the ultimate goal of an object-centered description of the scene. In Ref. 46), Szeliski stated that work on representations like intrinsic images, “was motivated by disappointment with feature-based approaches to vision . . .,” but also pointed out contemporary work that successfully matched features directly to object models, such as Ref. 37).

In recent years, progress in feature-based recognition has built on work such as Ref. 37) and most current systems have taken a decidedly feature-based approach. Successful bag-of-words systems, such as the system in Ref. 55) that was a co-winner in the 2009 PASCAL Object Detection Challenge, can be seen as using

no intermediate representations. In these systems, dense features are identified in the image and objects are found by analyzing the spatial distribution of these features in the image.

2.1 Defining of Low-Level Vision in Terms of Operations

Even though low-level vision has not proven vital to object recognition, strong low-level visual processing systems are still important. Before considering the importance of low-level vision, which will be discussed below in Section 2.2, it is useful to consider how low-level vision should be defined. Given that object-based representations have not proven to be the most useful application of low-level processing, it is reasonable to consider a definition of low-level vision that is broader than the hierarchy outlined in Fig. 1.

As has been recently suggested by Weiss⁵⁷⁾, a good working definition of low-level vision is those processing operations that output dense pixel-based representations. This is in line with Szeliski’s suggestion that low-level vision can be thought of as extracting intrinsic images. In thinking about the relationship between low-level vision and other processing, this definition of low-level vision moves away from a hierarchy like Fig. 1. Instead of being an intermediate representation on the path to a high-level result, low-level vision can instead be thought of as encompassing a set of tasks where the most natural output is a dense, pixel-based representation.

2.2 Why Pursue Low-Level Vision?

Although intermediate representations have not proven essential to object recognition, low-level visual processing is still important. The following sections will review these examples of important problems in low-level vision, including:

- Computer graphics applications
- The importance of low-level processing for image enhancement and editing
- The estimation of scene characteristics for tasks beyond recognition

2.2.1 Application in Computer Graphics and Image Editing

A significant trend in computer graphics over the last several years has been the development of systems that use photographs to create renderings of the scene pictured in those photographs. Often, this requires the extraction of intrinsic images capturing key characteristics necessary to re-render the scene.

A good example of this type of graphics application is the pop-up photography

work of Hoeim et al.²³⁾. This system creates a 3D representation of the photograph by identifying regions as being vertical, part of the ground, or belonging to the sky. Once each region has been classified, the surfaces can be rendered from different views. The key low-level visual processing step in this system is the classification of every image region, and thus pixel, as belonging to one of three geometric categories of vertical, ground, or sky.

The material-editing work of Khan et al. in Ref. 28) also shows how low-level visual processing can be used to change and edit objects in an existing photograph. In this work, heuristics are used to estimate quantities such as the local orientation and depth of points on the objects to be edited. These estimates make it possible to re-render objects with new material properties or replace objects in the image with rendered models of new objects.

Low-level vision also has application in video-centered graphics applications. In Ref. 60), Zitnick et al. use depth information recovered from multiple cameras to implement a number of video effects. In this system, a depth image is estimated for each camera view. These depth maps can be used to reason about the visibility of pixels in different views.

In some systems, an intrinsic image is actually the desired output. In Ref. 34), Levin et al. show how to estimate a full color image from a gray-scale image. The result can be viewed as an intrinsic image representing the chromaticity at each pixel.

2.2.2 Low-Level Vision and Image Processing

The definition of low-level vision as processing that leads to dense pixel-based outputs makes image processing and enhancement systems a natural problem for low-level vision.

The computer vision research literature has seen significant work on image enhancement problems such as denoising^{15),36),42),43),49),52)}, super-resolution^{12),16),45),52),53)}, dehazing¹⁹⁾, and image matting³⁵⁾.

2.3 Low-Level Vision for General Applications

It is also important to note that many applications lie outside the object-centered view in Fig. 1. In these applications, the intrinsic images returned by a low-level processing system may be the key piece of data that must be extracted from the scene. For example, if a robot is to successfully maneuver through an

area, it must be able to measure the relative location of possible obstacles in the scene – making depth-maps returned by a low-level visual system a crucial input.

3. Paper Organization

As described in the introduction, the discussion of incorporating learning into low-level vision problems will be facilitated by roughly dividing low-level vision systems into three basic types:

- (1) Systems utilizing only local information,
- (2) Systems that perform global processing via hand-specified local relationships, like Markov Random Fields,
- (3) Low-level vision systems that perform global processing, again using local relationships, but using relationships extracted from training example using machine learning algorithms.

The differences between these methods will be motivated by three different problems, which will be introduced in Section 4. The process of incorporating learning into each of these strategies will be discussed after each of these three strategies is introduced. **Table 1** summarizes some of the advantages and disadvantages of these strategies. The remaining sections of this paper will focus on this discussion, focusing on how these different strategies are applied to the problem. Section 8 will conclude by discussing important factors driving progress on low-level vision.

4. Utilizing Local Information

Because low-level vision systems must output results at every pixel, the solutions are typically structured as a set of operations repeated at each pixel. The most basic solutions use only local information as the basis of these operations. The following subsections each describe a significant low-level vision problem and a basic solution that uses only local information.

In many of these problems, there are fundamental ambiguities that make it impossible to solve this problem using only local information. These ambiguities will drive the need for models built that capture relationships in the estimated image.

Table 1 This table summarizes some of the advantages and disadvantages of the three strategies discussed here for designing low-level vision systems.

Strategy	Advantages	Disadvantages
Local Information Only	<ul style="list-style-type: none"> Typically requires relatively little computation. Problems typically reduce to standard machine learning tasks, such, as classification and regression. 	<ul style="list-style-type: none"> Local information is not sufficient in many cases, such as the aperture problem in optical flow.
Manually-Specified MRF Models	<ul style="list-style-type: none"> Global processing makes it possible to overcome ambiguities in local information, Hand-specified models can be designed intuitively. 	<ul style="list-style-type: none"> Using an MRF makes it necessary to perform inference, which can be computationally demanding.
MRF Models Trained on Data	<ul style="list-style-type: none"> Learning models from data makes it possible to use more complicated models with more parameters. 	<ul style="list-style-type: none"> Because learning depends on inference, the learning process can be very slow – and possibly intractable.

4.1 Optical Flow

In the optical flow problem, two frames, I_1 and I_2 , are used to estimate the motion at every pixel in the image. The solutions to this problem are typically based on the brightness constancy equation, which can be expressed in a discrete form as

$$E(\mathbf{u}, \mathbf{v}) = \sum_{i,j} (I_1(i, j) - I_2(i + u_{i,j}, j + v_{i,j}))^2 \quad (1)$$

similar to⁴⁴⁾, where \mathbf{u} and \mathbf{v} encode the horizontal and vertical motion vectors at each pixel in the image. This equation expresses that all points in the scene should have the same brightness and that all changes are due to motion.

One of the most well-known local solutions for the optical flow problem is the Lucas-Kanade approach³⁸⁾, which solves for the motion by solving a series of quadratic approximations to the local brightness constancy equations. The key assumption in these approximations is that motion is constant over a window surrounding each point.

This type of local approximation can be limited because of the well-known *aperture problem*. As shown in **Fig. 2** (a), when only an edge is visible, the motion seen inside the ring could either be caused by the bar moving vertically or diagonally, as shown in Fig. 2 (b) and Fig. 2 (c). Using only local information, the motion in (a) is ambiguous.

4.2 Separating Illumination and Albedo

Section 2 described how low-level vision can be thought of as extracting intrinsic images representing the different intrinsic characteristics of the scene. These characteristics include the albedo and illumination of each point in the scene.

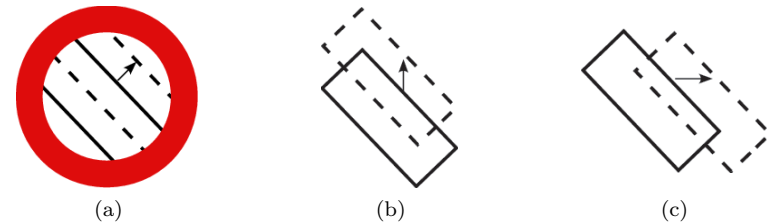


Fig. 2 These figures describe the aperture problem in optical flow. Perceived through the aperture in (a), the motion appears to be diagonal. However, (b) and (c) show that the same both horizontal and vertical motion could cause the same observation. This is a fundamental ambiguity in computing optical flow from only local information.

Assuming that surfaces are approximately Lambertian, albedo refers to the proportion of light reflected from the surface and the illumination of each point expresses the angle between the surface normal and the dominant orientation of the illumination. **Figure 3** shows a synthetic example of a surface and the corresponding shading and reflectance images.

In Ref. 32), Land and McCann propose the Retinex algorithm to solve this problem, which is sometimes referred to as lightness recovery²⁶⁾. The Retinex algorithm is designed with a specific type of image in mind, but has been shown to be remarkably robust on more complicated imagery¹⁸⁾.

The Retinex algorithm is designed for Mondrian images where the albedo is painted from large squares of constant intensity and the illumination varies slowly over the image. **Figure 4** shows an example of this type of image. The Retinex algorithm separates the albedo from the illumination in three steps:

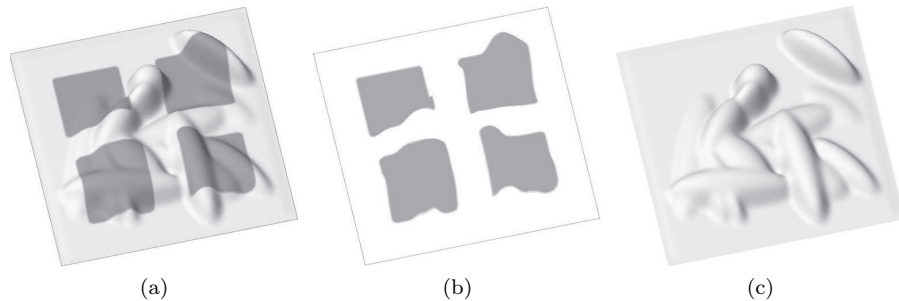


Fig. 3 These images show an example of how the image in (a) can be decomposed into the albedo and shading images shown in (b) and (c), respectively.

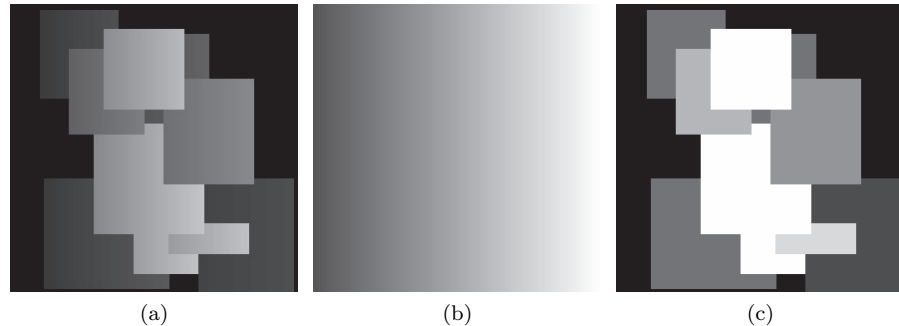


Fig. 4 The images show an example of a Mondrian image. The Retinex algorithm is designed to separate the albedo and illumination in images with a similar appearance. The image in (a) is composed of the smooth illumination in (b) and an albedo pattern like (c), which consists of large patches of constant albedo.

- (1) Compute the derivative along each scan-line in the image.
- (2) Find derivatives with a magnitude below a certain threshold and replace those derivatives with zero.
- (3) Re-integrate along each scan-line to recover the estimated albedo image.

The Retinex algorithm is based on the assumption that the illumination varies slowly, while changes in the albedo image will be caused by derivatives with a high magnitude. Unfortunately, this fundamental assumption is frequently violated in images of complex surfaces. **Figure 5** shows a pair of image patches that appear very similar. However, given the broader images, it is clear that one of these



Fig. 5 These images show how a system based on using only patches of data can encounter difficult ambiguities. The two enlarged patches look very similar, but one patch comes from an edge caused by a change in albedo, while the other comes from a change in shading.

patches is an illumination change and one is an albedo change. A successful system must be able to overcome this local ambiguity.

4.3 Low-Level Image Enhancement

Low-level image enhancement can be thought of as a problem where the goal is to improve corruption of pixel intensities. This includes the problems discussed in Section 2.2.2, like denoising, single-image super-resolution, and in-painting⁵⁾.

With only local information, the solution to most of these problems will primarily involve local filtering or interpolation. Other types of local edge-enhancement has also been considered, such as the method proposed by Greenspan et al., where the edges of an image are boosted with non-linear processing¹⁷⁾.

For tasks where the data is fully observed, like denoising, the primary obstacle is the correct choice of the window size when estimating the result. However, for many tasks, like in-painting or super-resolution, data is missing and local pixels may not provide enough information for a good result.

4.4 Learning Systems That Use Only Local Information

Using only local information, the learning problems in low-level vision directly map onto traditional problems like classification and regression. Fortunately, this also makes it straightforward to apply learning algorithms that have proven successful, such as GentleBoost¹⁴⁾ and the Support Vector Machine⁶⁾. Given the power of current learning algorithms, the challenge in implementing these systems lies in choosing the appropriate features.

5. Global Processing Through Local Relationships

The limitations of local information, discussed in the previous section, make



Fig. 6 Using only local information, the vertical edge at the center of the image could be viewed as either a painted line or a sharp fold in the surface. However, if information is propagated from the regions highlighted with circles along the edges, then the classification can be disambiguated.

it necessary to perform global processing. Formulating this global processing is one of the key problems in low-level vision.

Just as it is natural to express low-level processing in terms of operations that are repeated at each pixel, it is natural to express global processing in terms of local relationships. In Ref. 58), Weiss discusses how these local relationships can be used to implement global behavior.

Figure 6 shows an example of how local relationships can induce global behavior for a shading and illumination problem. In the center of the edge in this image, it is difficult to differentiate whether that edge is caused by a painted line or crease. However, at the top or bottom of the edge, it is clear that this is a painted line. To disambiguate the center of the edge, local relationships can be induced between vertical neighbors along the edge. If these local relationships constrain the neighbors along the edge to have the same classification as either an illumination change or an albedo change, then the pixels at the left and right edges can propagate their knowledge into the center pixels.

5.1 Markov Random Fields

These local relationships are conveniently implemented using Markov Random Fields. A distribution that is a Markov Random Field can be expressed as

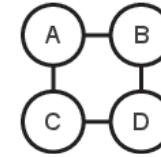


Fig. 7 This figure shows the graph for a pairwise lattice, which is a common model used for pixel relationships in a low-level vision model.

$$p(\mathbf{x}) = \frac{1}{Z} \prod_c \psi(\mathbf{x}_c) \quad (2)$$

where \mathbf{x} is the vector of random variables being estimated. The vector \mathbf{x}_c represents cliques, which can be thought of as subsets of pixels. The functions $\psi(\mathbf{x}_c)$ are functions on the values in each clique. These functions express the relationships between pixels in the clique and can be referred to as potential functions⁶⁾. These functions can be thought of as expressing the compatibility between the different possible states of the pixels in a clique.

As an example, a commonly used model is the pairwise lattice, where relationships are expressed between vertical and horizontal neighbors. For a four-pixel model, with labels shown in **Fig. 7**, the model could be expressed as

$$p(\mathbf{x}) = \psi(x_a, x_b)\psi(x_c, x_d)\psi(x_b, x_d)\psi(x_a, x_c) \quad (3)$$

What is notable about the structure of Eq. (3) is that the model is limited to the pairwise interactions. An important aspect of designing low-level vision systems is keeping the local relationships as simple as possible. This reduces the complexity of both the design of the potentials and the inference process, which will be discussed in Section 5.2. Pairwise models similar to Eq. (3) are particularly popular because they can implement the propagation behavior discussed earlier, in Section 5, or can be used to smooth out noisy, inconsistent decisions based on local information.

This type of model is also known as a graphical model because each element of \mathbf{x} can be represented as a node in the graph with edges representing the structure of the clique potentials. The graph representation of the pairwise model in Eq. (3) is shown in Fig. 7. The graphical model is a flexible, powerful tool for designing probabilistic models. A full discussion of these models is not possible here, but

excellent references include Refs. 6), 13), 29).

5.2 Inference in Markov Random Field Models

Given the distribution in the form of Eq. (2), the key problem is to use this distribution to estimate values for \mathbf{x} . In low-level vision applications, this is most often accomplished by finding a vector, \mathbf{x}^* that maximizes Eq. (2). This is sometimes referred to as *MAP* inference, where MAP is an abbreviation of *maximum a posteriori*.

If \mathbf{x}^* is estimated in this fashion, the probabilistic details can be largely dropped and the estimating \mathbf{x}^* can be viewed as simply minimizing an energy or cost function, which can be found by taking the negative log of Eq. (2). This is a particularly convenient route because it is no longer necessary to consider the normalizing constant, Z , in Eq. (2).

The advent of powerful inference algorithms is one of the reasons that MRFs are such a popular tool in low-level vision. As mentioned in the previous section, an MRF can be represented as a graph. If the graph representing the MRF has loops, then finding \mathbf{x}^* becomes NP-Hard⁸⁾, with some exceptions, including certain binary-valued MRFs and Gaussian MRFs.

Fortunately, a number of algorithms for providing an approximate MAP solution have been introduced, including the well-known Graph Cuts⁸⁾ and Belief Propagation algorithms⁴¹⁾. A recent comparison by Szeliski et al.⁴⁷⁾ presents a good review of several different types of MRF-based low-level problems and compares their performance on several different problems.

5.3 Connections to Other Types of Models

If only MAP inference is performed on the MRF, the probabilistic aspects of the MRF, discussed in Section 5.1 become less important. As mentioned above, with MAP inference in an MRF is equivalent to a model based on minimizing a factorized energy function. From this point of view, the MRF is connected to models based on optimizing energy or penalty functions, such as the KSVD model of images¹⁾.

5.4 Designing MRF Potentials for Low-Level Vision

The clique potentials, $\psi(\cdot)$, in Eq. (2) can be both hand-designed and learned from data. Section 6 will focus on how they can be trained, while we focus on hand-designed models here.

When potentials are hand-designed, they are typically designed to implement smoothing or propagation behaviors. For these applications, the pairwise-lattice is particularly popular, as mentioned above, because the clique potential functions are only functions of two variables. For discrete-valued problems, such as segmentation, the potential function becomes a table where each entry describes the compatibility between a pair of states.

This compatibility is problem dependent. In Freeman et al.'s work on super-resolution, the goal is to estimate a high-resolution image as a mosaic of individual patches¹²⁾. The compatibility between patches at neighboring pixels is measured by examining the similarity of the patches in an overlapping region between the two patches.

Another commonly used type of potential is the Potts model. For an MRF with three states per node, the clique potential has the form

$$\psi(x_a, x_b) = \begin{bmatrix} a & b & b \\ b & a & b \\ b & b & a \end{bmatrix}. \quad (4)$$

The defining characteristic of the Potts model is that one compatibility value is used when the neighboring pixels have the same label and a different compatibility value if the states differ, with the same compatibility value being used for all pairs of differing states. The Potts model has been used for a number of different applications, including stereo⁸⁾ and image enhancement¹⁰⁾. When the MRF is used for smoothing, it is common to modify the difference in a and b based on whether the observations indicate the presence of a discontinuity, as in Ref. 8), where the image is used to find edges in the underlying surface.

5.4.1 Application of MRFs in Optical Flow

It is interesting to note that one of the classic optical flow algorithms, the Horn and Schunck method²⁵⁾, is essentially based around an MRF. The Horn and Schunck method optimizes an energy function that includes both a data term imposing brightness constancy and a smoothness term that penalizes the flow changing from pixel to pixel. These smoothness terms, which are expressed as

$$\sum_{i,j} (u_{(i,j)} - u_{(i+1,j)})^2 + (u_{(i,j)} - u_{(i,j+1)})^2$$

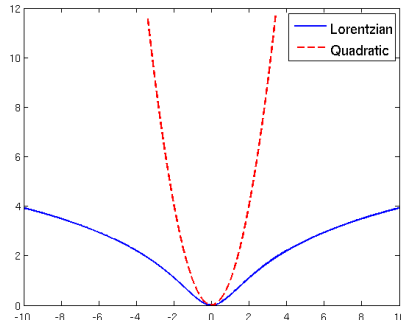


Fig. 8 This figure compares different functions used for enforcing smoothness in applications like optical flow. The advantage of a penalty like the Lorentzian penalty is that its growth slows down as the value of the penalty increases. This has the effect of letting the surface break into piecewise constant sections.

$$+ (v_{(i,j)} - v_{(i+1,j)})^2 + (v_{(i,j)} - v_{(i,j+1)})^2 \quad (5)$$

have the effect of propagating information from areas where the correct motion is clear to areas of the frame where it is ambiguous, such as the example shown in Fig. 2.

In later work, Black and Anandan improved on this formulation by replacing the quadratic terms with a robust penalty. **Figure 8** compares a type of robust penalty, the Lorentzian, with a quadratic penalty. The robust penalty is useful because it stops growing after the magnitude of the error grows beyond a certain level. This has the effect of not enforcing smoothness once the cost becomes too high. In the context of estimating a surface, this can be thought of as enforcing the prior that the surface should be smooth, but is allowed to have discontinuities. In Ref. 7), Black and Rangajaran show how this type of robust potential is equivalent to the line process proposed by Geman and Geman¹⁵⁾, where a separate random variable determines where the discontinuities should lie.

As discussed recently in Ref. 44), the choice of the penalty function can affect the performance of the system. In Ref. 44), Sun et al. discuss how different choices in system design affect the performance of an optical flow system.

5.4.2 MRFs for Illumination and Albedo

In Ref. 51), separating illumination and albedo is posed as a classification problem, similar to the Retinex algorithm. As mentioned in the beginning of this section, this classification can be ambiguous given only local information. Thus, the MRF is configured to propagate information along edges. This is implemented with a Potts model that strongly encourages pixels along an edge to have the same label.

5.4.3 Hand-Designed MRFs for Image Enhancement

In image enhancement tasks, compatibility between overlapping patches is a common approach. As described in Section 5.4, this was used in Freeman et al.'s super-resolution system to choose the patches that form the high-resolution estimate. Similar strategies have also been used for texture synthesis³¹⁾ and demosaicing⁵³⁾.

5.5 Learning in Hand-Designed Models

In a hand-designed model, a simple classifier or regression system can be implemented as a component of the system that is integrated in a hand-specified way. This is related to an approximate learning technique, which will be discussed in Section 6.3.1, where the system is trained on small subsets of the data.

6. Learning Local Relationships from Data

The MRF models discussed in the previous section provide a powerful tool that typically leads to significantly improved results on most low-level vision problems. The systems described in the previous section are similar in that they all rely on hand-designed models.

Because these systems are hand-designed, they tend to be simple, often relying on pairwise lattice models. This opens the question of whether performance can be improved by examining more complex models and how these more complex models should be constructed.

These questions parallel similar issues in the development of classifiers. Using rules like the Neyman-Pearson Hypothesis Test, it is possible to hand-design classifiers based on the distribution of the observations and labels. The development of classifiers like the Support Vector Machine have made it possible to use training data to learn more complex, powerful classifiers than can be created by

hand. These classifiers are more complex because the parameters of the model are optimized directly on the data. This makes it possible to develop non-linear models with much larger numbers of parameters than could be set manually. In the SVM, for example, there is one parameter associated with each support vector in the final classifier.

The power of optimizing classifiers on training data has made these models an important component of most modern computer vision systems. Thus, it is natural to pursue training strategies to improve the performance of MRF-based low-level vision systems.

6.1 Learning MRF Parameters for Low-Level Vision

The following subsections will outline different strategies for learning MRF parameters. Before describing different methods, Section 6.2 will introduce the central problem in learning MRF parameters. Section 6.3 will then review methods for learning parameters. This discussion will start with simple methods that are based on learning using local approximations and continue to newer methods that perform more global optimizations.

6.2 Maximum Likelihood Estimation of MRF Parameters

As mentioned in Section 5.1, a probability distribution based on an MRF can be expressed in the form shown in Eq. (2). With this distribution, the maximum-likelihood method can be used to estimate the parameters of the MRF. Given a set of training examples, $\mathbf{t}_1, \dots, \mathbf{t}_N$, and a set of corresponding observations, $\mathbf{y}_1, \dots, \mathbf{y}_N$, the negative log-likelihood function can be written as

$$L = - \sum_{i=1}^N \sum_c \log \psi(\mathbf{x}_c^i) + \log Z \quad (6)$$

where \mathbf{x}_c^i is clique c in the i th image.

If this model depends on a vector of parameters θ , these parameters can be found by minimizing the negative log-likelihood function. This is expressed formally as

$$\theta^* = \arg \min_{\theta} - \sum_{i=1}^N \sum_c \log \psi(\mathbf{x}_c^i; \theta) + \log Z(\theta). \quad (7)$$

In this formulation, both the potential functions $\psi(\cdot)$ and the normalization

constant, or partition function, are functions of the parameters θ .

The central problem in implementing the minimization in Eq. (7) is that evaluating the partition function is often NP-complete, just as finding the MAP estimate is NP-complete. This makes it impossible to compute the log-likelihood function, much less optimize it.

6.3 Approximate Methods for Learning MRF Parameters

A number of different strategies have been proposed for learning the parameters θ despite the difficulties with computing the partition function. The following subsections outline several strategies that have been successfully applied.

6.3.1 Learning with Local Approximations

One of the simplest strategies is to use approximations based on local data to learn the MRF parameters. For example, in Ref. 51), the potentials are based on a logistic regression classifier that estimates the probability of two pixels having the same label. This classifier is trained directly from pairs of nodes.

The limitation of this approach is that this classifier is trained independently for pairs of nodes. The logistic regression function that is used to create the potentials is simply trained to estimate the probability of two neighboring nodes having the same label. While this is likely related to the overall goal of correctly labeling derivative values, it does not directly optimize the final objective.

The pseudo-likelihood approximation, used in Ref. 30), is a similar approximation where the log-likelihood is replaced with the log of the product of the conditional distributions of each node, conditioned on all other nodes. This has the effect of eliminating the need to compute the partition function²⁹⁾.

6.3.2 Bounding the Partition Function

Another strategy is algorithms that bound the partition function, such as Ref. 56). This approach was successfully used for segmentation in Ref. 33).

6.3.3 Sampling Strategies

If the negative log-likelihood function is optimized with a derivative-based algorithm, the gradient can be expressed in terms of expectations of the clique potentials²⁹⁾. Unfortunately, computing the necessary expected values is also typically intractable when computing the partition function is intractable. An alternative to computing them exactly is to draw a set of samples from the distribution defined by the particular values of θ , then use those samples to ap-

proximate the expectation values. This strategy is used in the work of Zhu and Mumford⁵⁹⁾; where sampling is used to estimate MRF models of texture.

Typically, these samples are drawn using Markov Chain Monte-Carlo sampling methods. Neal's survey in Ref. 40) is an excellent introduction to these methods.

6.3.4 Contrastive Divergence

One of the disadvantages of the sampling strategy is that the computational cost of generating enough samples to compute the expectation accurately can be prohibitive. In Ref. 22), Hinton points out that this expectation can be approximated from just a single sample and lead to good estimates of the parameters. This strategy works in the same way as stochastic gradient descent, where the parameter vector is updated using a gradient calculated from a small number of examples. In stochastic gradient descent, using only a subset of the data to calculate the gradient makes gradient computation fast, which makes it possible to update the parameter vector many times. Updating the parameter vector many times has the effect of averaging out the inaccuracies in the gradient generated using only a subset of the data. In the same way, a single sample can be generated very quickly, which makes it possible to update the parameter vector often, also averaging out inaccuracies in the gradient computation.

In Ref. 22), this is referred to as minimizing a quantity called the contrastive divergence. One of the most well-known applications of this technique in low-level vision is the Field of Experts model proposed by Roth and Black⁴²⁾, which will be discussed in Section 6.4. Contrastive divergence learning has also been applied to segmentation problems by He et al. in Ref. 20).

6.3.5 Discriminative Learning of Parameters

Discriminative methods present an alternative to the maximum likelihood approach to estimating the MRF parameters. To understand how discriminative methods can be applied for learning MRF parameters, it is helpful to review the discriminative approach for training classifiers. When implementing a classifier, the defining characteristic of a discriminative classifier is that it is not constructed by estimating the joint distributions of observations and labels. Instead, a function is directly fit for the sole goal of correctly labeling observations. This function can be fit using different criteria, such as the max-margin criterion used in Support Vector Machines or the log-likelihood criterion used in logistic

regression.

In estimating MRF parameters for low-level vision, discriminative methods also avoid fitting distributions to the data. Instead, discriminative approaches define alternate criteria for estimating parameters. Since distributions will not form the basis for estimation, current discriminative methods base the learning criterion on the MAP solution of the MRF.

As in Section 5.2, the MAP solution is the vector that has the highest probability or

$$\mathbf{x}^* = \arg \max_{\mathbf{x}} - \sum_C \log \psi(\mathbf{x}_i^C) + \log Z. \quad (8)$$

It should be noted that while the MRF in Eq. (8) is expressed in terms of a probability distribution, the partition function, Z , does not depend on \mathbf{x} and can be ignored. In this view, the MRF can be viewed as simply defined by an energy function over possible states of \mathbf{x} .

The training procedure is based on a loss function that measures how similar \mathbf{x}^* is to the ground-truth value. The goal during training is to minimize this loss by optimizing the parameters θ so that the MAP solution of the MRF defined by θ is as similar as possible to the ground-truth.

If the MRF is defined by an energy function $E(\mathbf{x}; \mathbf{y}, \theta)$, based on observations \mathbf{y} , then this training can be expressed as

$$\begin{aligned} \theta^* &= \arg \min_{\theta} L(\mathbf{x}^*, \mathbf{t}) \\ \text{where } \mathbf{x}^* &= \arg \min_{\mathbf{x}} E((\mathbf{x}; \mathbf{y}, \theta) \end{aligned} \quad (9)$$

In this optimization, changing the parameters θ changes the MRF and also changes the location of the MAP estimate \mathbf{x}^* . The goal of the optimization is to find a value of θ that makes \mathbf{x}^* as close to the ground truth \mathbf{t} as possible, as measured by the loss function $L(\cdot)$.

6.3.6 Optimizing Parameters in Discriminative Methods – Gradient Descent

If the optimization of \mathbf{x}^* can be expressed as a closed-form set of differentiable operations, then the chain rule can be used to compute the gradient of $L(\cdot)$ with respect to the parameters θ .

As pointed out in Ref. 52), if the energy function defining the MRF is a

quadratic function, which corresponds to an MRF that is also a Gaussian random vector, then the MAP solution, \mathbf{x}^* , can be found with a set of matrix multiplications and a matrix inversion operation. These are differentiable operations, so it is possible to compute the gradient vector $\partial L/\partial \theta$. While the quadratic model typically does not perform as well as models like the Field of Experts⁵²⁾, and Ref. 49) show how the training process makes it possible to learn to exploit the information in the observations to perform surprisingly well on both image enhancement and segmentation tasks.

For models where \mathbf{x}^* cannot be computed analytically⁴⁹⁾, and Ref. 3) propose using an approximate value of the MAP solution. Both of these systems propose using an approximate value of \mathbf{x}^* that is computed using some form of minimization, that may not lead to a global minimum of the energy function $E(\cdot)$. If the minimization is itself a set of differentiable operations, then the chain rule can be applied to compute how this approximate \mathbf{x}^* changes as the parameter vector θ changes.

These papers differ in the type of optimization used. In Ref. 49), the training is built around an upper-bound minimization strategy that fits a series of quadratic models during the optimization process. Taking a different approach, Barbu proposes using a small number of steps of gradient descent, which leads to a very efficient learning system³⁾.

6.3.7 Optimizing Parameters in Discriminative Methods – Large Margin Methods

Just as the support vector machine uses quadratic programming to optimize the max-margin criterion, quadratic programming can be used to learn parameters in MRF models. In Ref. 54), Taskar et al. describe the M^3N method, which is one of the first margin-based methods for learning MRF parameters. This approach poses learning as a large quadratic program and has been applied to vision problems by Anguelov et al.²⁾.

The most influential approach, across all options for learning MRF parameters, is the cutting plane approach for training²⁷⁾. In this approach, the inference system is used as an oracle to find results that violate constraints in the training criterion. A major advantage of this approach is that it does not require that the inference process be differentiable in some way. This makes it possible to use

a wide variety of structures, including MRF's where inference is intractable¹¹⁾. Recent vision systems based on this training approach include^{9),48)}. In addition, a high-quality implementation of this algorithm is available in the SVMStruct package²⁷⁾.

6.4 Application and Benefits of Learning to Low-Level Problems

These learning systems have made it possible to learn more complicated, better performing models of images than could be fit by hand or with only local evidence. An influential, recent model is the Field of Experts model proposed by Roth and Black⁴²⁾. This model uses robust penalty functions⁷⁾ to penalize the response of a set of higher-order filters. In Ref. 42), this results in a model defined by 245×5 filters with associated weights. These models led to significant improvements on a number of tasks, including denoising, in-painting, and the estimation of optical flow.

These methods have also led to improved capabilities in estimating albedo and shading images. The binary-valued MRF used in Ref. 51) was changed to a continuous-valued, quadratic MRF in Ref. 50). This made it possible to train the MRF discriminatively. The resulting model was able to produce estimates with less noise and artifacts than the classification-based system.

7. Learning-based Low-level Vision Systems without an MRF

Throughout this paper, the discussion on learning in low-level vision has been focused on systems based on Markov Random Fields. The MRF provides a rigorous, well-specified tool for gathering and propagating information across the image. As mentioned in the introduction, an MRF model is also similar to other types of low-level vision models.

The following sections will first review the K-SVD model, which is similar to the MRF model, then describe an alternate approach that is very different from the MRF models presented.

7.1 The K-SVD Model of Images

While the K-SVD model for image enhancement, described in Ref. 39), is not considered an MRF model, it has many similarities. In this model, image enhancement is posed as an optimization, similar to MAP inference in an MRF. Formally, this optimization is described as

$$\{\hat{\alpha}_{ij}, \hat{\mathbf{D}}, \hat{\mathbf{x}}\} = \arg \min_{\mathbf{D}, \alpha_{ij}, \mathbf{x}} \lambda \|\mathbf{x} - \mathbf{y}\|_2^2 + \sum_{i,j} \mu_{i,j} \|\alpha_{i,j}\|_0 + \sum_{i,j} \|\mathbf{D}\alpha_{ij} - \mathbf{R}_{ij}\mathbf{x}\|_2^2, \quad (10)$$

where \mathbf{y} is the observed image, $\hat{\mathbf{x}}$ is the estimated image, α are reconstruction coefficients, and \mathbf{D} is a dictionary that forms a kind of image model. The terms depending on the dictionary are similar in form to potentials in an MRF.

In this optimization, the dictionary \mathbf{D} is optimized during the reconstruction. In Ref. 39), an initial estimate of this dictionary is produced by performing the optimization over a set of training images. Because the minimization is over the reconstruction and the dictionary, this optimization is similar in form to the optimization in Eq. (9). A key difference is that the dictionary is expected to change with each image to which it is applied. In an MRF model, in contrast, the potentials typically do not change.

7.2 Cascading Classification

However, it should be noted that the MRF is not the only tool for accomplishing this behavior. The desired propagation and aggregation can also be implemented using cascading layers of classification steps^{21),23),24)}. This strategy is fundamental to the photo pop-up work of Hoiem et al.²³⁾. As discussed in Section 2.2.1, this system operates by labeling pixels as belonging to one of three categories. This labeling is created by beginning with multiple segmentations then progressively merging segments into larger segments. The decision as to whether to merge a segment is made by a classifier. This makes the labeling process a series of classification steps.

This process is more formalized in Refs. 24) and 21). In these models, the image is processed by layers of off-the-shelf classifiers. These classifiers can examine both the observations and the results of the previous layers. The advantage of this approach is that it can be implemented with standard classification techniques.

8. Learning for the Next Generation of Low-Level Vision Systems

The MRF learning algorithms discussed in the previous section have been particularly valuable for learning models of images because they have made it possible to learn higher order models that better match the low-level statistics of images. The success of these models for image enhancement, and other applica-

tions, raises the question of how low-level vision systems can continue to improve.

With the development of practical learning algorithms, MRF models, and similar classification-based strategies discussed in Section 7, are able to perform well at modeling the local relationships in data. However, system performance will be limited by whether those local relationships are correctly specified. This is a significant limitation in current systems because many systems apply the same local relationships across the whole image or only make modest adjustments based on image information.

However, the advent of strong learning techniques makes it likely that systems are already capturing much of the information available at the local-level. It can be argued that further progress can only be made by involving higher-level recognition. With high-level recognition, the knowledge of the objects and types of surfaces in the image can be used to better determine the local behavior of pixels.

Interestingly, this inverts the hierarchy in Fig. 1. Instead of serving as a step for high-level vision, high-level vision becomes a tool for improving low-level vision. As problems in both areas progress, the major advances in the capability of low-level vision systems will likely come from the combination of low and high-level vision systems.

Acknowledgments The author was supported during the writing of this paper by US National Science Foundation grants IIS-0905387 and IIS-0916868.

References

- 1) Aharon, M., Elad, M., Bruckstein, A. and Katz, Y.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, *IEEE Trans. Signal Processing*, Vol.54, No.11, pp.4311–4322 (2006).
- 2) Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G. and Ng, A.: Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.169–176 (2005).
- 3) Barbu, A.: Learning Real-Time MRF Inference for Image Denoising, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.1574–1581 (2009).
- 4) Barrow, H.G. and Tenenbaum, J.M.: Recovering Intrinsic Scene Characteristics from Images, Hanson, A. and Riseman, E. (Eds.), *Computer Vision Systems*, pp.3–26, Academic Press (1978).

- 5) Bertalmio, M., Sapiro, G., Caselles, V. and Ballester, C.: Image inpainting, *Proc. 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, New York, NY, USA, pp.417–424, ACM Press/Addison-Wesley Publishing Co. (2000).
- 6) Bishop, C.M.: *Pattern Recognition and Machine Learning*, 1st ed. 2006. corr. 2nd printing edition, Springer (2007).
- 7) Black, M.J. and Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision, *International Journal of Computer Vision*, Vol.19, No.1, pp.57–92 (1996).
- 8) Boykov, Y., Veksler, O. and Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.23, No.11, pp.1222–1239 (2001).
- 9) Desai, C., Ramanan, D. and Fowlkes, C.: Discriminative Models for Multi-Class Object Layout, *Proc. IEEE International Conference on Computer Vision*, pp.228–236 (2009).
- 10) Felzenszwalb, P.F. and Huttenlocher, D.P.: Efficient Belief Propagation for Early Vision, *International Journal of Computer Vision*, Vol.70, No.1, pp.41–54 (2006).
- 11) Finley, T. and Joachims, T.: Training Structural SVMs when Exact Inference is Intractable, *International Conference on Machine Learning (ICML)*, pp.304–311 (2008).
- 12) Freeman, W.T., Pasztor, E.C. and Carmichael, O.T.: Learning Low-Level Vision, *International Journal of Computer Vision*, Vol.40, No.1, pp.25–47 (2000).
- 13) Frey, B.J. and Jojic, N.: Advances in Algorithms for Inference and Learning in Complex Probability Models for Vision, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.27, No.9, pp.1392–1416 (2002).
- 14) Friedman, J., Hastie, T. and Tibshirani, R.: Additive Logistic Regression: A Statistical View of Boosting, *The Annals of Statistics*, Vol.38, No.2, pp.337–374 (2000).
- 15) Geman, S. and Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, No.6, pp.721–741 (1984).
- 16) Glasner, D., Bagon, S. and Irani, M.: Super-Resolution From a Single Image, *Proc. IEEE International Conference on Computer Vision*, pp.349–356 (2009).
- 17) Greenspan, H., Anderson, C.H. and Akber, S.: Image enhancement by nonlinear extrapolation in frequency space, *IEEE Trans. Image Processing*, Vol.9, No.6, pp.1035–1048 (2000).
- 18) Grosse, R., Johnson, M.K., Adelson, E.H. and Freeman, W.T.: Ground-truth dataset and baseline evaluations for intrinsic image algorithms, *International Conference on Computer Vision*, pp.2335–2342 (2009).
- 19) He, K., Sun, J. and Tang, X.: Single image haze removal using dark channel prior, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp.1956–1963 (2009).
- 20) He, X., Zemel, R. and Carreira-Perpinan, M.: Multiscale Conditional Random Fields for Image Labelling, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2004).
- 21) Heitz, G., Gould, S., Saxena, A. and Koller, D.: Cascaded Classification Models: Combining Models for Holistic Scene Understanding, *Advances in Neural Information Processing Systems (NIPS 2008)* (2008).
- 22) Hinton, G.: Training products of experts by minimizing contrastive divergence, *Neural Computation*, Vol.14, No.7, pp.1771–1800 (2002).
- 23) Hoiem, D., Efros, A.A. and Hebert, M.: Automatic Photo Pop-up, *ACM Trans. Graphics (SIGGRAPH 2005)*, Vol.24, No.3 (2005).
- 24) Hoiem, D., Efros, A.A. and Hebert, M.: Closing the Loop on Scene Interpretation, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2008).
- 25) Horn, B.K.P. and Schunck, B.G.: Determining Optical Flow, *Artificial Intelligence*, pp.185–203 (1981).
- 26) Horn, B.K.P.: *Robot Vision*, chapter 9, MIT Press (1986).
- 27) Joachims, T., Finley, T. and Yu, C.-N.: Cutting-Plane Training of Structural SVMs, *Machine Learning*, Vol.77, No.1, pp.27–59 (2009).
- 28) Khan, E.A., Reinhard, E., Fleming, R.W. and Bühlhoff, H.H.: Image-based material editing, *ACM Trans. Gr.*, Vol.25, pp.654–663 (2006).
- 29) Koller, D. and Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*, MIT Press (2009).
- 30) Kumar, S. and Hebert, M.: Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification, *Proc. 2003 IEEE International Conference on Computer Vision (ICCV '03)*, Vol.2, pp.1150–1157 (2003).
- 31) Kwatra, V., Essa, I., Bobick, A. and Kwatra, N.: Texture optimization for example-based synthesis, *ACM Trans. Gr.*, Vol.24, pp.795–802 (2005).
- 32) Land, E.H. and McCann, J.J.: Lightness and Retinex Theory, *Journal of the Optical Society of America*, Vol.61, pp.1–11 (1971).
- 33) Levin, A. and Weiss, Y.: Learning to Combine Bottom-Up and Top-Down Segmentation, *European Conference on Computer Vision (ECCV)*, Graz, Austria (2006).
- 34) Levin, A., Lischinski, D. and Weiss, Y.: Colorization using optimization, *ACM Trans. Gr.*, Vol.23, pp.689–694 (2004).
- 35) Levin, A., Lischinski, D. and Weiss, Y.: A Closed-Form Solution to Natural Image Matting, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.30, No.2, pp.228–242 (2008).
- 36) Liu, C., Szeliski, R., Kang, S.B., Zitnick, C.L. and Freeman, W.T.: Automatic Estimation and Removal of Noise from a Single Image, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.30, pp.299–314 (2008).
- 37) Lowe, D.G.: *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, Boston (1985).
- 38) Lucas, B.D. and Kanade, T.: An iterative image registration technique with an

- application in stereo vision, *7th International Joint Conference on Artificial Intelligence (IJCAI-81)*, pp.674–679 (1981).
- 39) Mairal, J., Elad, M. and Sapiro, G.: Sparse representation for color image restoration, *IEEE Trans. Image Processing*, pp.53–69 (2007).
 - 40) Neal, R.M.: Probabilistic Inference Using Markov Chain Monte Carlo Methods, Technical Report CRG-TR-93-1, University of Toronto, Dept. of Computer Science (1993).
 - 41) Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, second edition (1988).
 - 42) Roth, S. and Black, M.: Field of Experts: A Framework for Learning Image Priors, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Vol.2, pp.860–867 (2005).
 - 43) Schmidt, U., Gao, Q. and Roth, S.: A Generative Perspective on MRFs in Low-Level Vision, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.1751–1758 (2010).
 - 44) Sun, D., Roth, S. and Black, M.J.: Secrets of Optical Flow Estimation and Their Principles, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.2432–2439 (2010).
 - 45) Sun, J., Sun, J., Xu, Z.B. and Shum, H.Y.: Image Super-resolution Using Gradient Profile Prior, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2008).
 - 46) Szeliski, R.: Bayesian Modeling of Uncertainty in Low-Level Vision, *International Journal of Computer Vision*, Vol.5, No.3, pp.271–301 (1990).
 - 47) Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M.F. and Rother, C.: A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.30, No.6, pp.1068–1080 (2008).
 - 48) Szummer, M., Kohli, P. and Hoiem, D.: Learning CRFs Using Graph Cuts, *Proc. European Conference on Computer Vision (ECCV)*, pp.582–595 (2008).
 - 49) Tappen, M.F.: Utilizing Variational Optimization to Learn Markov Random Fields, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2007).
 - 50) Tappen, M.F., Adelson, E.H. and Freeman, W.T.: Estimating Intrinsic Component Images using Non-Linear Regression, *Proc. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol.2, pp.1992–1999 (2006).
 - 51) Tappen, M.F., Freeman, W.T. and Adelson, E.H.: Recovering Intrinsic Images from a Single Image, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.27, No.9, pp.1459–1472 (2005).
 - 52) Tappen, M.F., Liu, C., Adelson, E.H. and Freeman, W.T.: Learning Gaussian Conditional Random Fields for Low-Level Vision, *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2007).
 - 53) Tappen, M.F., Russell, B.C. and Freeman, W.T.: Efficient Graphical Models for Processing Images, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Vol.2, pp.673–680 (2004).
 - 54) Taskar, B., Chatalbashev, V., Koller, D. and Guestrin, C.: Learning Structured Prediction Models: A Large Margin Approach, *22nd International Conference on Machine Learning (ICML-2005)* (2005).
 - 55) Vedaldi, A., Gulshan, V., Varma, M. and Zisserman, A.: Multiple Kernels for Object Detection, *Proc. International Conference on Computer Vision (ICCV)*, pp.606–613 (2009).
 - 56) Wainwright, M.J., Jaakkola, T.S. and Willsky, A.S.: A New Class of Upper Bounds on the Log Partition Function, *IEEE Trans. Inf. Theory*, Vol.51, No.7, pp.2313–2335 (2005).
 - 57) Weiss, Y.: Learning and Inference in Low-Level Vision, Invited Talk at NIPS 2009.
 - 58) Weiss, Y.: Interpreting images by propagating Bayesian beliefs, *Advances in Neural Information Processing Systems 9*, pp.908–915 (1996).
 - 59) Zhu, S.C., Wu, Y. and Mumford, D.: Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling, *International Journal of Computer Vision*, Vol.27, No.2, pp.107–126 (1998).
 - 60) Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S. and Szeliski, R.: High-quality video view interpolation using a layered representation, *ACM Trans. Gr. (Proceedings of SIGGRAPH 2004)*, pp.600–608 (2004).

(Received December 16, 2010)

(Accepted September 22, 2011)

(Released December 28, 2011)

(Communicated by David Suter)



Marshall F. Tappen is an Assistant Professor of Computer Science at the University of Central Florida. He joined UCF in 2006 after receiving a Ph.D. from the Massachusetts Institute of Technology. His interests include low-level vision, learning parameters in graphical models, and the combination of machine learning and computer vision.