Learning for Optical Flow Using Stochastic Optimization

Yunpeng Li and Daniel P. Huttenlocher

Department of Computer Science, Cornell University, Ithaca, NY 14853
{yuli,dph}@cs.cornell.edu

Abstract. We present a technique for learning the parameters of a continuous-state Markov random field (MRF) model of optical flow, by minimizing the *training loss* for a set of ground-truth images using simultaneous perturbation stochastic approximation (SPSA). The use of SPSA to directly minimize the training loss offers several advantages over most previous work on learning MRF models for low-level vision, which instead seek to maximize the likelihood of the data given the model parameters. In particular, our approach explicitly optimizes the error criterion used to evaluate the quality of the flow field, naturally handles missing data values in the ground truth, and does not require the kinds of approximations that current methods use to address the intractable nature of maximum-likelihood estimation for such problems. We show that our method achieves state-of-the-art results and requires only a very small number of training images. We also find that our method generalizes well to unseen data, including data with quite different characteristics than the training set.

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 [1,2,3,4,5,6,7,8,9], 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 [10], the relative lack of investigation of learning techniques stands in sharp contrast with higher-level vision such as object recognition, where learning techniques are ubiquitous. Hand tuning of models not only involves substantial human effort, more importantly it limits our understanding of how well optical flow methods generalize to unseen data and thus will perform in practice. This is illustrated by the recently developed Middlebury optical flow evaluation database [11], where methods that perform best on the classical "Yosemite" sequence tend not to perform as well on new imagery for which they were not hand tuned.

In this paper, we present a continuous-state Markov random field (MRF) [12,13,14] based model for optical flow. This is in contrast to most recent applications of MRF methods in computer vision, for problems such as stereo, where relatively small sets of discrete states are used. We learn the parameters of this model using simultaneous perturbation stochastic approximation (SPSA) [15] to minimize the *training loss* – that is the error on the training data under some error function. In particular, we measure

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part II, LNCS 5303, pp. 379–391, 2008.

training loss using the average end-point error (AEPE) [16] which is one of the error metrics employed in the Middlebury evaluation [11].

Directly optimizing for training loss as opposed to a maximum likelihood approach (e.g. as pursued in [10]) 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 maximum *a posteriori* estimate) [17], sampling [10], or some type of local training [18]. These approximations, however, tend to be imprecise and may lead to noisy and unreliable estimates, as noted in [17]. Moreover, from a statistical learning point of view, the maximum likelihood estimate is not well justified 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., [19]) as we discuss further in Section 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. [20]). SPSA [15] 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 [21], 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.

The rest of the paper is organized as follows. We provide some background and discuss related work in Section 1.1, and then define our model for optical flow in Section 2. The learning method is described in more detail in Section 3, and the experimental results are presented in Section 4. We conclude in Section 5.

1.1 Background and 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 [22]. 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. [23]). Window-based approaches approaches, however, suffer from the generalized aperture problem [24], 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. [24]), 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 [25], 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 [19] 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 [26,10]), 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 [5] 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 [27], 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 [28], 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. The work most closely related to ours is that of [10], where field-of-expert (FoE) models [29] are learnt from ground-truth flow fields inferred from range-scan data. To our knowledge, it is 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 an approximation to maximum likelihood estimation 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 [15] as a gradient-free stochastic optimization method, and it is closely related to the classical finite-difference stochastic approximation (FDSA) algorithm [30] 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 [15,21]. 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 [21]. 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.

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). 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 $\mathcal{C}$ be the set of cliques of the graph. As is well known, the posterior of a labeling $\mathbf{w}$ given data I decomposes into the product of maximal clique potentials and node potentials,

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

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

$$\begin{aligned} E(\mathbf{w};\boldsymbol{\theta},I) &= -\log p(\mathbf{w}|I;\boldsymbol{\theta}) - \log Z(\boldsymbol{\theta}) \\ &= \sum_{c\in\mathcal{C}} f_c^{\boldsymbol{\theta}}(\mathbf{w}_c) + \sum_{p\in\mathcal{V}} g_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I). \end{aligned} \tag{2}$$

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

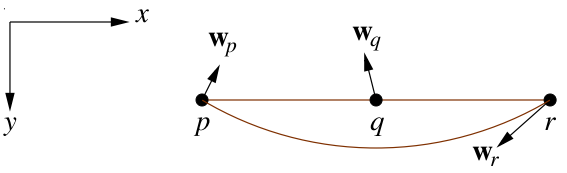

Fig. 1. A horizontal 3-clique over pixels (nodes) p, q, and r. Each pixel has a 2-D vector label indicating the flow value (i.e. the apparent motion).

descent can be used to obtain an approximate solution (which we shall return to in Section 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.

2.1 Energy Functions

We use robust energy functions as proposed in [5]; 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

$$f_c^{\boldsymbol{\theta}}(\mathbf{w}_c) = \lambda_S \cdot \rho(\sqrt{||\beta_1 \mathbf{d_1}||^2 + ||\beta_2 \mathbf{d_2}||^2}) \tag{3}$$

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

$$\rho(\cdot) = \log(1 + \frac{1}{2}|\cdot|^2) \tag{4}$$

is the standard Lorentzian function, and

$$\begin{aligned} \mathbf{d_1} &= \mathbf{w}_r - \mathbf{w}_p \\ \mathbf{d_2} &= \mathbf{w}_p - 2\mathbf{w}_q + \mathbf{w}_r, \end{aligned} \tag{5}$$

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 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 [29,10,31], 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 $\mathbf{w}$,

$$g_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I) = \lambda_D \cdot \rho(\beta_D ||I_1(p + \mathbf{w}_p) - I_0(p)||), \tag{6}$$

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, 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.

2.2 Optical Flow Estimation

To estimate optical flow, we perform approximate MAP inference on the MRF by minimizing the energy function in Equation 2 using gradient descent. Computing the gradient of the spatial term energy $E_S = \sum_{c \in \mathcal{C}} f_c^{\boldsymbol{\theta}}(\mathbf{w}_c)$ is straightforward since it has an analytical form in $\mathbf{w}$, and the gradient at each pixel is given by

$$(\nabla_{\mathbf{w}} E_S)_p = \sum_{c \in \mathcal{C}: p \in c} \nabla_{\mathbf{w}_p} f_c^{\boldsymbol{\theta}}(\mathbf{w}_c). \tag{7}$$

Since the data term (Equation 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

$$(\nabla_{\mathbf{w}} E_D)_p = \nabla_{I_1(p+\mathbf{w}_p)} g_p^{\boldsymbol{\theta}}(\mathbf{w}_p, I) \nabla_{\mathbf{w}_p} I_1(p + \mathbf{w}_p). \tag{8}$$

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 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 [32,33], which has faster converge speed

than steepest descent. We also employ hierarchical coarse-to-fine strategy for optical flow computation [2,3], 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.

3 Learning the Parameters

As noted above, we take the approach of learning models that yield low training loss (e.g., [19]) rather than those that maximize the likelihood of the training data. There are several advantages to this approach. First, as discussed in Section 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.

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 $L(\boldsymbol{\theta})$. [1] As noted above we do this using SPSA [15]. 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

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k \hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k), \tag{9}$$

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,

$$\left(\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)\right)_i = \frac{l(\hat{\boldsymbol{\theta}}_k + c_k \boldsymbol{\Delta}_k) - l(\hat{\boldsymbol{\theta}}_k - c_k \boldsymbol{\Delta}_k)}{2c_k \left(\boldsymbol{\Delta}_k\right)_i}, \tag{10}$$

where $l(\cdot)$ is some noisy measurement of the true loss $L(\cdot)$ ($l = L$ if the measurement is noise-free), $\boldsymbol{\Delta}_k$ is a user-defined m-dimensional random perturbation vector satisfying certain conditions [15], 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 $c_k = c/k^\gamma$ [15]. 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 [34].

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 $\boldsymbol{\Delta}$ 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 $\boldsymbol{\Delta}$ 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 [15]. 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.

[1] 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.

4 Experimental Results

To evaluate our method, we trained our model on the "other" data set [2] from the Middlebury optical flow web site [11] and tested its performance on the "eval" data set from the same web site. For learning, we use the average end-point error (AEPE) [16,11] 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. [3] 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 2 shows a plot of the training errors against the number of iterations for all five trials.

Fig. 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.

For evaluation, we report error rates in both average angular error (AAE) [35] and average end-point error (AEPE). Table 1 shows the performance of our learnt model on the eight sequences for flow evaluation from the Middlebury optical flow web page [11]. 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

[2] Only sequences with ground-truth flow are used. We excluded "Venus" from the training set, since it is a stereo sequence.

[3] 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.

Table 1. Performance on the eight evaluation sequences from the Middlebury optical flow page [11], 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	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy	**Average**
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.* [9]	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 [5]	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 [25]	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 [23]	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

Table 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.*[9]	10.99/0.43	–	–
Black/Anandon [5]	9.26/0.35	–	–
Lucas/Kanade [23]	10.27/0.37	–	–

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 [17] (second method in Table 1), so as to compare it with SPSA (first method in Table 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 2 the error rates on the training sequences. Results from other methods are quoted from [11] whenever available. One can see that our training data includes only three sequences, which is in contrast with the several hundred used in [10]. 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.

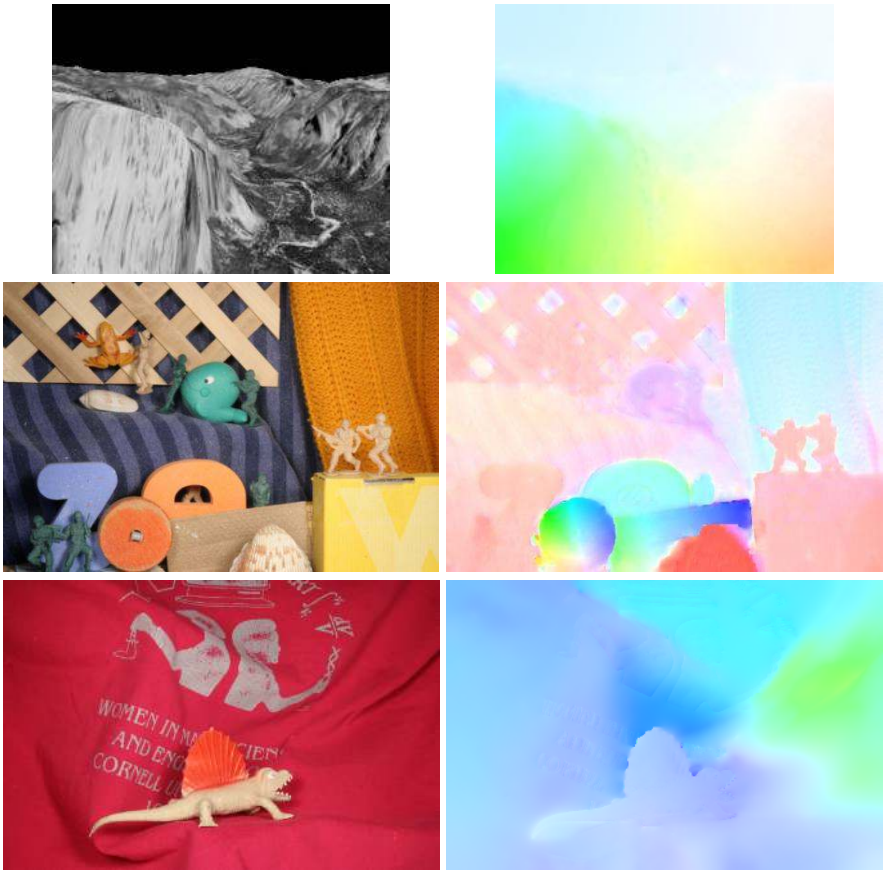

Fig. 3. Output of our model for the sequences "Yosemite" (top), "Army" (middle), and "Dimetrodon" (bottom). Left: A frame of the image sequence. Right: Estimated flow. Observe the predominantly smooth flow field of "Yosemite" and "Dimetrodon" in contrast to the large amount of motion discontinuity in "Army".

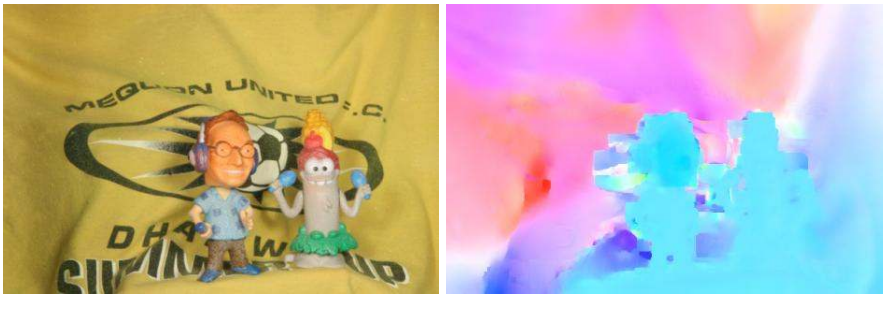

Fig. 4. Output of our model for the sequence "Mequon". Left: A frame of the image sequence. Right: Estimated flow. Most of the errors occur in the shadows around the cartoon models, due to their high relative motion with respect to the background.

Figure 3 displays some sample flow fields produced by our model, color coded using the scheme described in [11]. 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 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. [36,5]), and remains an interesting topic for future work.

5 Conclusion

We have presented 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. Moreover, our model is learnt from a separate set of training sequences that does not contain any of those used for evaluation. This demonstrates that our model generalizes well to unseen data. Since many low-level vision problems involve parameters that are difficult to optimize deterministically, the learning technique that we employed here may well prove useful in other research areas.

Acknowledgments

This work was supported in part by NSF grant IIS-0713185.

References

1. Nagel, H.H., Enkelmann, W.: An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. PAMI 8(5), 565–593 (1986)
2. Anandan, P.: A computational framework and an algorithm for the measurement of visual motion. IJCV 2(3), 283–310 (1989)
3. Bergen, J.R., Anandan, P., Hanna, K.J., Hingorani, R.: Hierarchical model-based motion estimation. In: ECCV, London, UK, pp. 237–252. Springer, Heidelberg (1992)
4. Cohen, I.: Nonlinear variational method for optical flow computation. In: Scandinavian Conference on Image Analysis (1993)
5. Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. CVIU 63(1), 75–104 (1996)
6. Szeliski, R., Coughlan, J.: Spline-based image registration. IJCV 22(3) (1997)
7. Alvarez, L., Weickert, J., Sanchez, J.: Reliable estimation of dense optical flow fields with large displacements. International Journal of Computer Vision 39(1), 41–56 (2000)
8. Ye, M., Haralick, R.M., Shapiro, L.G.: Estimating piecewise-smooth optical flow with global matching and graduated optimization. PAMI 25(12), 1625–1630 (2003)

9. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: combining local and global optic flow methods. IJCV 61(3), 211–231 (2005)
10. Roth, S., Black, M.J.: On the spatial statistics of optical flow. IJCV 74(1), 33–50 (2007)
11. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV (2007)
12. Besag, J.E.: Spatial interaction and the statistical analysis of lattice systems. J. Royal Stat. Soc., B 36(2) (1974)
13. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. PAMI 6(6), 721–741 (1984)
14. Szeliski, R.: Bayesian modeling of uncertainty in low-level vision. IJCV 5(3) (1990)
15. Spall, J.C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control 37, 332–341 (1992)
16. Otte, M., Nagel, H.H.: Optical flow estimation: Advances and comparisons. In: ECCV, Secaucus, NJ, USA, pp. 51–60. Springer, New York (1994)
17. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. CVPR (2007)
18. Sutton, C., McCallum, A.: Piecewise training of undirected models. In: UAI (2005)
19. LeCun, Y., Huang, F.J.: Loss functions for discriminative training of energy-based models. In: AIStats (2005)
20. Tappen, M.F.: Utilizing variational optimization to learn Markov random fields. CVPR (2007)
21. Spall, J.C.: Overview of the simultaneous perturbation method for efficient optimization. Hopkins APL Technical Digest 19, 482–492 (1998)
22. Bertero, M., Poggio, T., Torre, V.: Ill-posed problems in early vision. In: Proceedings of IEEE (1988)
23. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI, pp. 674–679 (1981)
24. Jepson, A., Black, M.J.: Mixture models for optical flow computation. CVPR, 760–761 (1993)
25. Horn, B.K.P., Schunk, B.G.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
26. Geman, S., Graffigne, C.: Markov random field image models and their applications to computer vision. In: Intl. Congress of Mathematicians (1986)
27. Simoncelli, E.P., Adelson, E.H., Heeger, D.J.: Probability distributions of optical flow. In: CVPR, Mauii, Hawaii, pp. 310–315. IEEE Computer Society, Los Alamitos (1991)
28. Fleet, D.J., Black, M.J., Yacoob, Y., Jepson, A.D.: Design and use of linear models for image motion analysis. IJCV 36(3), 171–193 (2000)
29. Roth, S., Black, M.J.: Fields of experts: A Framework for Learning Image Priors. CVPR (2005)
30. Kiefer, J., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. Annuals of Mathematical Statistics 23(3), 462–466 (1952)
31. Weiss, Y., Freeman, W.F.: What makes a good model of natural images? CVPR (2007)
32. Nocedal, J.: Updating quasi-newton matrices with limited storage. Mathematics of Computation 35, 773–782 (1980)
33. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Mathematical Programming 45(3, (Ser. B)), 503–528 (1989)
34. Spall, J.C.: Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Trans. Aerosp. Electon. Syst. 34(3), 817–823 (1998)
35. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. IJCV 12(1), 43–77 (1994)
36. Fleet, D.J., Jepson, A.D.: Computation of component image velocity from local phase information. IJCV 5(1), 77–104 (1990)