Adaptive Matching based Optical Flow Estimation with Discrete Optimization

Ph.D. Dissertation

Kyong Joon Lee

Department of Electrical Engineering and Computer Science

Seoul National University

ABSTRACT

Optical flow estimation aims to find dense visual correspondences between a reference and a target images. Obtaining such dense correspondences may benefit various computer vision algorithms. For decades, many researches have been dedicated to resolve the problem, but it still remains challenging problem and is actively studied these days. Specifically, various appearance of objects may weaken implicit segmentation of flow and degrade estimation on motion boundaries. Complex and large displacement of an object may also degenerate the performance. In addition, individual movements of adjacent objects inherently produce occlusion in the target image; and may increase the estimation error as the corresponding point for the occlusion is actually undefined.

In this work, we propose several methods to address these problems. Our methods construct discrete energy models for the problems and obtain solutions with discrete optimization techniques. First, to reduce errors of estimated flow around motion boundaries, we propose a novel adaptive window matching approach utilizing statistical information in the window. The proposed approach is based on using large correlation windows with adaptive support-weights. We present three new types of weighting constraints derived from image gradient, color statistics and occlusion information. Each of the proposed constraints appreciably elevates the

quality of estimations, and that they jointly yield results that compare favorably to current techniques, especially on the motion boundaries.

Second, to handle complex non-transitional motion with large displacement, we present a new energy model presenting discrete analog to the diffusion tensor-based regularizer. Inspired from the fact that the regularization process works as a convolution kernel filtering, we formulate the difference between original flow and filtered flow as a smoothness prior. Experiments demonstrate the proposed method yields plausible results on the various data sets including large displacement and complex motion boundaries.

Third, we address occlusion by simultaneously estimate flow and detect occlusion in a single framework using a novel support-weight based window matching. The proposed support-weight provides a very effective clue to detect occlusion based on the assumption that occlusion is sparse; and also presents reasonable estimation for the flow of the occluded pixels. Our method improves the flow accuracy as well as detection performance, compared to the approach alternatively finding solutions in individual frameworks; and also yields highly competitive results outperforming the previous state-of-the-art methods.

Keywords: Optical flow estimation, occlusion detection, window matching, support-weight, bilateral filtering, discrete optimization, high-order MRF, message-passing.

Student Number: `2006-23181`

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Optical flow is, by Horn's definition [4], "the distribution of apparent velocities of movement of brightness patterns in an image, caused by the relative motion between an observer (an eye or a camera) and the scene." Optical flow estimation aims to find dense visual correspondences between a reference and a target images.

Obtaining such dense correspondences is indeed a very fundamental task in image processing and computer vision problems, and so it can benefit a number of algorithms in the literature; e.g., motion segmentation, motion compensated coding, frame interpolation, medical image registration, super-resolution, 3D scene reconstruction, video denoising, high dynamic range image and so on.

For decades, many researches have been dedicated to resolve the problem, and have presented plausible solutions. However, some issues still remain challenging problem and is actively studied these days, and this work proposes novel approaches, addressing three issues in general. First, most of recent methods employ implicit segmentation of flow field for improve estimation on motion boundaries, but various appearance of objects may weaken the implicit segmentation and degenerate the

performance. Second, complex and large movements of an object are exceptional but frequently shown in the images; and it is cumbersome to appropriately model them by the previous approaches. Finally, separate motions of neighboring objects generate occlusion, and since the corresponding point in the target image is missing by definition, it may increase the estimation error around the occlusion.

1.1 Outline of this work

The remainder of this thesis is structured as follows. We briefly introduce some fundamental approaches to the problem in Chapter 2, with explanation for choosing the discrete energy model, which has been recently introduced in the flow estimation literature, and presented several advantages compared to the previous frameworks based on the continuous optimization.

In Chapter 3, we propose a novel adaptive window matching approach utilizing statistical information in the window, reducing errors of estimated flow around motion boundaries. The proposed approach is based on using large correlation windows with adaptive support-weights. We present three new types of weighting constraints derived from image gradient, color statistics and occlusion information. The first type provides gradient structure constraints that favor flow consistency across strong image gradients. The second type imposes perceptual color constraints that reinforce relationship among pixels in a window according to their color statistics. The third type yields occlusion constraints that reject pixels that are seen in one window but not seen in the other. All these constraints contribute to suppress the effect of cluttered background, which is unavoidably included in the large correlation windows. Experimental results demonstrate that each of the proposed constraints appreciably

elevates the quality of estimations, and that they jointly yield results that compare favorably to current techniques, especially on the motion boundaries.

Addressing the issue for complex non-transitional motion with large displacement, Chapter 4 presents a new energy model presenting discrete analog to the diffusion tensor-based regularizer. Inspired from the fact that the regularization process works as a convolution kernel filtering, we formulate the difference between original flow and filtered flow as a smoothness prior. Then the discrete framework enables us to employ a robust penalizer less concerning convexity and differentiability of the energy function. In addition, we provide a new kernel design based on the bilateral filter, adaptively controlling intensity variance according to the local statistics. The proposed kernel simultaneously addresses over-segmentation and over-smoothing problems, which is hard to achieve by tuning parameters. Involving a complex graph structure with large label sets, this work also presents a strategy to efficiently reduce memory requirement and computational time to a tolerable state. Experiments demonstrate the proposed method yields plausible results on the various data sets including large displacement and complex motion boundaries.

In Chapter 5, we also manage the occlusion issue, by simultaneously estimate flow and detect occlusion in a single framework using a novel support-weight based window matching. The proposed support-weight provides a very effective clue to detect occlusion based on the assumption that occlusion is sparse; and also presents reasonable estimation for the flow of the occluded pixels. Applying coarse-to-fine approach, our method successfully detects non-sparse occlusion as well. The energy model with the matching cost and flow smoothing cost is optimized by efficient discrete optimization method. Our method improves the flow accuracy as well as detection performance, compared to the approach alternatively finding solutions in

individual frameworks; and also yields highly competitive results outperforming the previous state-of-the-art methods.

We finalize this work by providing conclusion and discussion for the future work in Chapter 6.

Chapter 2

Background

A basic clue to the optical estimation problem is the brightness constancy constraint, assuming the intensity value of a pixel in the reference image may not change in the target image. With this constraint only, however, the problem is highly under-constrained with ambiguity that one pixel in the reference image may correspond to multiple pixels in the target image.

Various approaches have been introduced. In [4], a global formulation is combined with the linearized brightness constancy assumption, to enforce spatial coherence between locally adjacent flows, using a quadratic function. Although this seminal approach has been employed as a baseline algorithm in numerous follow-up researches, but it suffers from over-penalized outliers included in the motion boundaries. To address this challenge, several works proposed more robust penalizing functions. Black et al. [5] replace the quadratic error function with the Lorentzian function, which is robust to outliers, but is very difficult to find the optimum due to its non-convexity. The functions based on L1-norm [6–8] have been shown to be a good substitute for the non-convex robust function, utilizing variational methods

for optimization.

Apart from this limitation, variational approaches may also suffer from restricted form of data term. The data term for brightness constancy should be linearized based on the Taylor series approximation to make the functional differentiable. The estimated motion fields are assumed to be in small displacements and can not catch up with large deformations with non-linear movement. Traditional approach [9] uses successive pyramids in a coarse-to-fine fashion, arising another problem losing detail structures in images. A recent work [3] proposed a solution by discretizing the data cost functional and decoupling the energy for minimization; not guaranteed to find the global minimum.

While most of these works employ the variational method for optimization, several works reported promising results using discrete framework. These approaches consider the flow estimation as a labelling problem and constructs a discrete energy model to be optimized by the discrete methods [10–12]. In [13], proposal solutions using continuous optimization are first computed and then, combined with discrete optimization. To reduce the high complexity of discrete methods, Glocker et al. [14] incrementally update flow vectors, only within highly probable regions. In [15], an input image is represented as a tree of over-segmented regions, which defines an energy function to be optimized by dynamic programming. A non-local smoothness prior on the discrete framework [16] is also proposed, showing competitive results to the variational models. In contrast, Rhemann et al. [17] presents plausible estimations, using cost-volume filtering without using any prior information, which reduces much of the computational complexity.

In this work, we also employed the discrete framework which presents several advantages compared to the previous frameworks based on the continuous optimiza-

tion. On this framework, more options are available for the robust penalizer less concerning convexity and differentiability. The data term in our model inherently covers large displacement flow since the brightness constancy assumption does not have to be linearized. It is also compatible to various data matching functions, which may not be the case for the variational approach. The main drawback of the discrete method is that the computational complexity is proportional to the number of labels. Addressing this challenge, we introduced and developed various efficient techniques to reduce the complexity for our framework.

Chapter 3

Adaptive Window Correlation with Local Statistics

3.1 Introduction

One of big issues in optical flow estimation problem is enhancing performance on motion boundaries. Regardless of optimization framework, whether it is the variational or the discrete method, the estimation error in *discontinuous* regions is almost always bigger than the error in *untextured* regions. For example, the average endpoint error computed with the method of Brox et al. [3] for the Army sequence in the Middlebury flow site is 0.11/0.32/0.11, for overall/discontinuous/untextured regions respectively, while the error computed with the method of Glocker et al. [14] is 0.12/0.34/0.11. Both methods yields relatively high estimation error on the discontinuous motion boundaries.

3.1.1 Previous work

Various approaches have been introduced to address the boundary issue. Many of them employ filtering-based implicit segmentation of flow. In [18], a multi-cue driven bilateral filter is employed to discard background clutter in the smoothing kernel. Sun et al. [1] reveal that applying the median filter to intermediate flow estimations [19] is a very effective approach to the issue. They incorporate this heuristic scheme in their energy function as non-local L1 smoothness prior and present state-of-the-art results.

While most of these works employ the variational method for optimization, several works reported promising results using discrete optimization methods [10–12]. A non-local smoothness prior on the discrete framework [16] is proposed, showing competitive results to the variational models. Rhemann et al. [17] presents plausible estimations, using cost-volume filtering without using any prior information, which reduces much of the computational complexity.

3.1.2 Our approach

We address the estimation based on the discrete MAP-MRF framework [14, 20], comprising the data matching cost and the spatial smoothness cost. In comparing brightness of pixel for the data cost calculation, we employ local neighborhoods of the pixel [7, 21], instead of the single pixel of interest. This pixel set of local neighborhoods is referred to as a *correlation window*. We aim to improve the estimation, by enhancing the quality of the correlation window matching.

One critical factor for the quality of the matching is the size of the window. A large correlation window can address the aperture phenomenon, and other robust-

ness issues, such as illumination change and/or random noise. On the other hand, the large window may also include cluttered background motion segments, which may cause incorrect window matching on motion boundaries. To address this issue, the support-weight based approach [22, 23] has been widely employed in the field of stereo matching. It imposes different weights on each pixel t in the window according to geometric and photometric constraints, e.g., the pixel's proximity and the color difference to the central pixel s, defined as follows:

$$w_s^{prox}(t) = \exp\left(-\frac{||\mathbf{x}_s - \mathbf{x}_t||^2}{2\sigma_g^2}\right), \tag{3.1}$$

$$w_s^{color}(t) = \exp\left(-\frac{||I(s) - I(t)||^2}{2\sigma_p^2}\right), \tag{3.2}$$

where $\mathbf{x}_s$, $\mathbf{x}_t$ indicate 2D coordinates, and $I(s)$, $I(t)$ mean color values of the points s and t, respectively.

This strategy gives an effect accentuating the foreground object, and outperforms previous works, such as adaptively changing window size [24] and using multiple windows [25]. However, the fixed variances (σ_g, σ_p) that are applied to all image regions can degrade the performance on certain image regions, particularly for very large correlation windows. Learning the parameters on test images can be a solution, but it requires extra time complexity, and may not cover the variety of real world scenes.

We propose to employ three new types of weight constraints, which are adaptively adjusted, based on contents in the correlation window.

Gradient structure constraint: The previous proximity constraint applies the homogeneous weight to every correlation window regardless of the image contents in each window. This may help coherent estimation on a single large object, but

may degrade results on objects with detailed geometric structures, such as small branches of a tree. Since such objects form strong image gradients in general, we propose to use the structure tensor of the window, to adaptively apply the weight distribution according to the geometric structure. We assign strong weights normal to the predominant gradient directions in the window. As a result, the shape of weighted region in the window appears to be a sharp ellipse along with the objects, reducing the effect of background regions outside the objects.

Perceptual color constraint: In the previous color constraint in Eq. (3.2), σ_p needs to be small enough, to clearly distinguish a foreground object from background objects by their color difference. However, the small variance may result in over-segmented estimation on a single large object containing various colors inside, which could stem from object texture, image noise or specular illumination. Addressing the issue, we propose to use perceptual color distance that takes account of color distribution and accordingly calculate the weight. By applying a new perceptual color distance, instead of the Euclidean color distance, we could obtain relatively high coherence inside the single object, while using the small σ_p for the boundary distinction.

Occlusion constraint: Occlusion indicates the phenomenon that a certain area of the reference image is not seen in the target image due to various reasons, such as object movement and/or view change. Since the pixels in the occlusion do not correspond to any pixel in the target image, we propose to exclude this region in computing widow correlation.

We compare our results with results using the proximity and color constraints in [22, 23], and show the proposed constraints outperform the previous constraints

in our experiments. To demonstrate the effect of the occlusion constraint, we show that adding the occlusion constraints to the geometric and photometric constraints leads to improvements in quantitative evaluation. We also show the proposed constraints jointly yield highly competitive performance on various data sets, especially on motion boundaries.

The rest of this chapter is organized as follows. Section 2 briefly defines our energy formulation for the discrete framework. In Section 3 we propose the new adaptive correlation window design and show its advantages. Section 4 introduces a method to enhance the efficiency of the discrete optimization, and Section 5 presents experimental results evaluating the proposed model. We finalize this chapter by providing the conclusion and the future work in Section 6.

3.2 Background

Let $\mathcal{G}$ be an undirected graph with a node set $\mathcal{V}$ and an edge set $\mathcal{E}$. A node in $\mathcal{V}$ corresponds to a pixel in the reference image. Let l_s be a label, i.e., a random variable for a node s in some discrete sample space $\mathcal{L}_s = \{1, \ldots, L^2\}$, representing the quantized displacement vector set $\mathcal{T}_s = \{\mathbf{u}_s(1), \ldots, \mathbf{u}_s(L^2)\}$. Note the displacement vector is two dimensional, i.e., $\mathbf{u}_s = (u_s, v_s)$, and each dimension is homogeneously quantized by L labels. Optical flow estimation can be expressed as finding the labels for each pixel, which minimizes an energy function such as:

$$\sum_{s \in \mathcal{V}} \Phi_s(l_s) + \sum_{(s,t) \in \mathcal{E}} \Psi_{st}(\mathbf{u}_s(l_s) - \mathbf{u}_t(l_t)) \,, \tag{3.3}$$

where $\Phi_s(\cdot)$ imposes the cost for matching the correlation window for s, and $\Psi_{st}(\cdot)$ denotes the spatial smoothness term between s and t.

3.2.1 Coarse-to-fine approach

The discrete sample space $\mathcal{L}$ is a finite set. The size of the space $|\mathcal{L}|$ $(= L^2)$ is proportional to the maximum displacement over the desired flow precision μ, such that, $|\mathcal{L}| \propto \max(\mathcal{T})/\mu$. If the given scenes contain very large displacement, or if we desire very accurate flow estimation, using a big sample space with sufficient number of labels would be a simple solution; however, it may drastically increase computational complexity and memory requirement. To yield similar estimation results using fewer labels, we employ the following coarse-to-fine approaches.

Image pyramid: We build Gaussian image pyramids for the input images, and find the rough solution from the top level of the pyramids. Down to the next level, the dense flow field is estimated by interpolating the coarse solution, and is provided as the initial flow field for further estimation. The number of pyramid level is determined by $\log_d(\max(\mathcal{T})/|\mathcal{L}|)$ where d^{-1} is the downsampling factor building the image pyramid. We use $d = 2$ in our experiments.

Incremental flow update: To produce high precision flow using the limited number of labels, we iteratively find the incremental flow based on the current flow field. For the i_{th} iteration, the flow precision is set to $f^{(i)} = 0.5f^{(i-1)}$, so that the discrete algorithm employs smaller quantization unit for the incremental flow. We note, in practice, the flow accuracy at a higher level pyramid strongly influences the performance of the next level pyramid; thus we run this process to obtain sufficiently high precision at every pyramid level.

3.2.2 Data matching criteria

A virtue of the discrete approach is that we can test various data matching cost $\Phi_s(l_s)$ in Eq. (5.1), without changing the optimization scheme according to the cost function. We tested SAD (Sum of Absolute Difference), SSD (Sum of Squared Difference), and NCC (Normalized Cross Correlation), which are combined with GIP (Gradient Inner Product) [14]. GIP is a measure for geometric constancy [26], computing the angle between the gradients of two input images. Except NCC, weighted versions of these matching criteria are defined using the support-weight on each pixel in the window, such that:

$$\Phi_s(l_s) = \frac{\sum_{t \in W(s)} w_s(t) w_{s'}(t') \rho(t, t')}{\sum_{t \in W(s)} w_s(t) w_{s'}(t')}, \tag{3.4}$$

where $W(s)$ is a node set in the window supporting s, and w_s means a weight function for s. s' and t' are points in the target image, where s, t in the reference image are mapped to by displacement vector $\mathbf{u}(l_s)$. $\rho(t, t')$ denotes a similarity measure between pixels at t and t', for example of GIP:

$$\rho(s, s') = \left| \frac{\nabla I_1(s)}{|\nabla I_1(s)|} \cdot \frac{\nabla I_2(s')}{|\nabla I_2(s')|} \right| , \tag{3.5}$$

where the subscripts 1 and 2 for I denote the reference and the target images respectively. The weighted NCC [23] is defined as follows:

$$\Phi_s^{WNCC}(l_s) = \frac{\sum_{t \in W(s)} w_s(t) w_{s'}(t') J_1(s, t) J_2(s', t')}{\sqrt{\sum_{t \in W(s)} |w_s(t) J_1(s, t)|^2} \sqrt{\sum_{t \in W(s)} |w_{s'}(t') J_2(s', t')|^2}} , \tag{3.6}$$

where $J(s, t) = I(t) - \bar{I}(s)$ and $\bar{I}(s)$ is the mean intensity of pixels in $W(s)$.

The combination of GIP and each criterion is implemented by summing two terms with a balancing parameter λ; e.g., GIP+NCC is defined as follows:

$$\Phi_s(l_s) = \lambda\Phi_s^{WGIP}(l_s) + (1-\lambda)\Phi_s^{WNCC}(l_s).$$

We empirically found GIP+NCC with $\lambda = 0.7$ showed the best performance among other combinations, and used it through experiments in the work.

3.3 Proposed Constraints for Adaptive Window Correlation

The weight function in Eq. (5.2) consists of three types of constraints. We assume all constraints are independent events and that they can be measured individually. Then the function is determined by multiplying each factor as follows:

$$w_s(t) = w_s^{grad}(t)w_s^{perc}(t)w_s^{occ}(t), \tag{3.7}$$

where w_s^{grad}, w_s^{perc} and w_s^{occ} are functions related to the three proposed constraints.

3.3.1 Gradient structure constraint

At the point where strong gradient is found, we may expect distinct object boundary in the direction normal to the gradient. By adaptively imposing weak weight along this gradient, the pixels across the boundary have less effect on the matching cost calculation. This constraint can be implemented using the structure tensor (i.e., second moment matrix) of a correlation window. The eigenvectors of the structure tensor indicate the predominant directions of the gradient in the window. We define an anisotropic tensor $\mathbf{T}_p(s)$ with the structure tensor of the window around s as follows:

$$\mathbf{T}_g(s) = \frac{1}{D_g}\left(\frac{1}{|W(s)|}\sum_{k\in W(s)} \nabla I_g(k)\nabla I_g(k)^T + \nu^2\mathbf{1}\right), \tag{3.8}$$

where $\mathbf{1}$ means the 2×2 identity matrix, and ∇I_g means the gradient of the gray-scaled input image. D_g represents a denominator for normalization, defined as the trace of the matrix in the parenthesis. ν is the parameter controlling the degree of isotropy. If ν is large enough, $\mathbf{T}_g(s)$ becomes close to the identity matrix, and if ν is close to zero, $\mathbf{T}_g(s)$ is almost identical to the second moment matrix.

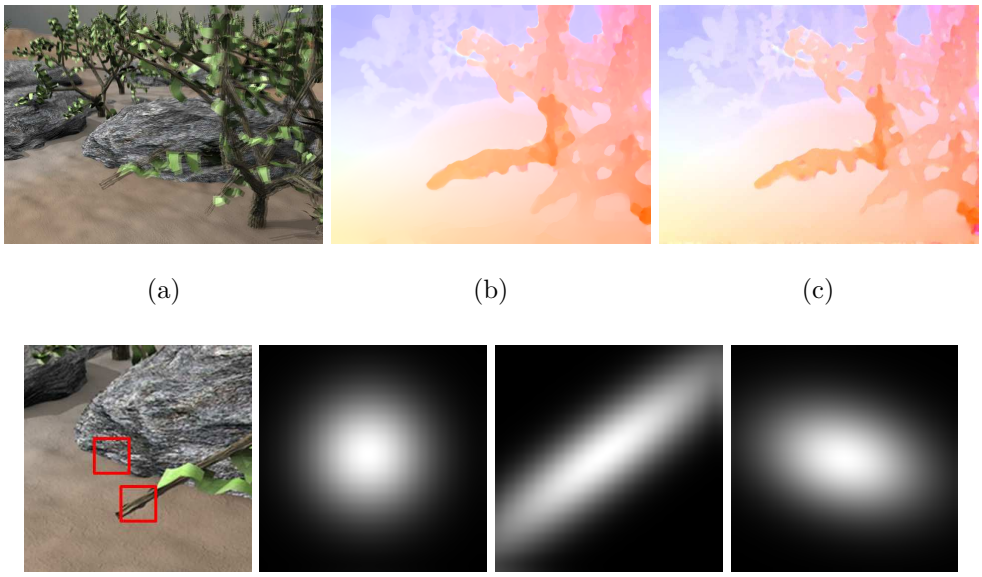
With this tensor, we define the weight function as follows:

$$w_s^{grad}(t) = \exp\left(-\frac{(\mathbf{x}_s - \mathbf{x}_t)^\mathsf{T}\,\mathbf{T}_g(s)\,(\mathbf{x}_s - \mathbf{x}_t)}{2\sigma_g^2}\right). \tag{3.9}$$

The effect of the proposed constraint is demonstrated in Figure 3.1. We sample two points (shown in red boxes in (d)) and present their weight distributions. Compared to the previous proximity constraint shown in (e), the proposed constraint can impose low weight on background regions; e.g., around the branch as shown in (f). As a result, the proposed constraint presents improvements, with less foreground-fattening effect around fine structures of the scene, as seen in (b) compared to (c). While the weight distributions for the previous constraint are homogeneous regardless of the sample points, the proposed constraint controls degree of isotropy according to the image content, as presented in (g).

3.3.2 Perceptual color constraint

To define a new color distance between two pixels, proposed perceptual color constraint makes use of the Mahalanobis distance style, instead of the Euclidean distance. Assuming the color distance in a correlation window share the same distribu-



(a) (b) (c)

(d) (e) (f) (g)

Figure 3.1: Weight calculation around fine details of objects in images. The estimation results are represented in HSI color space (direction: hue, magnitude: saturation) (a) The Grove3 sequence. (b) Estimation using the proximity constraint. (c) Estimation using the gradient structure constraint. (d) Magnified view of the reference image, containing a thin branch and a stone highlighted by red boxes. (e) Weight distribution using the proximity constraint. The distribution is homogeneous regardless of image contents. (f) Weight distribution for the branch using the proposed gradient structure constraint. (g) Weight distribution for the stone using the proposed constraint.

tion, we define the new color distance measure using the covariance matrix of color difference of adjacent pixels in the window. For a three-channel color space, the covariance matrix in the window around a node s is defined as follows:

Table 3.1: Quantitative evaluation for γ

γ	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
EPE	0.353	0.342	0.325	0.314	0.310	0.312	0.316	0.319	0.318	0.321

$$\mathbf{\Sigma}_c(s) = \frac{1}{D_p} \begin{bmatrix} \sigma_{C_0,C_0} & \sigma_{C_0,C_1} & \sigma_{C_0,C_2} \\ \sigma_{C_1,C_0} & \sigma_{C_1,C_1} & \sigma_{C_1,C_2} \\ \sigma_{C_2,C_0} & \sigma_{C_2,C_1} & \sigma_{C_2,C_2} \end{bmatrix}, \tag{3.10}$$

where $\sigma_{X,X}$, $\sigma_{X,Y}$ means the variance and covariance of channel $X, Y \in \{C_0, C_1, C_2\}$ for the gradient image ∇I. D_p represents a denominator for normalization, defined as the trace of the matrix. In experiments, we use the CIELab color space.

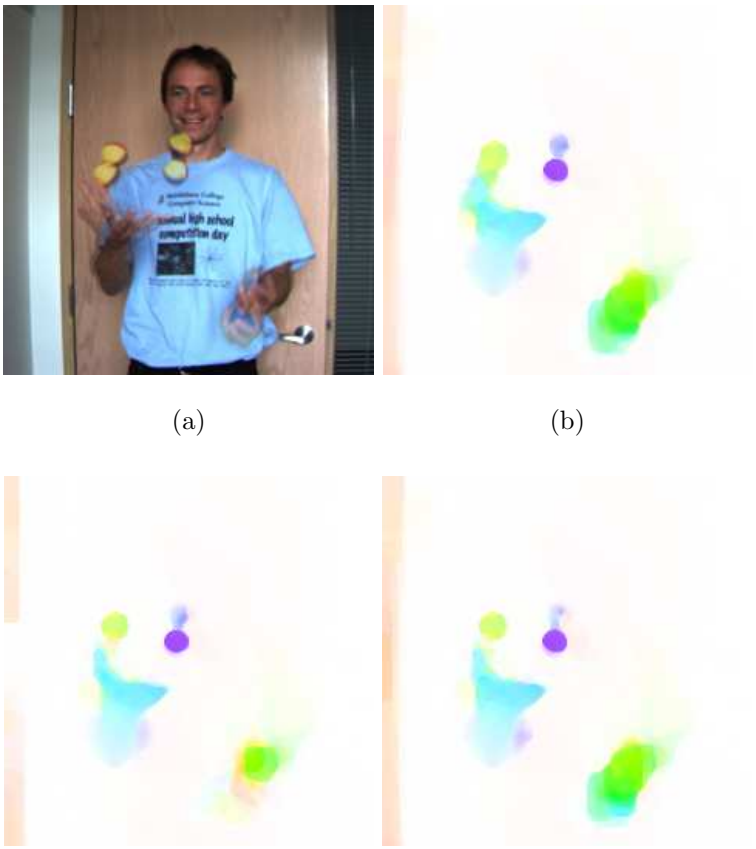
We define the weight function to adaptively compute the difference of color:

$$w_s^{perc}(t) = \exp\left(-\frac{(I(s) - I(t))^{\mathsf{T}} \mathbf{\Sigma}_c^{-1}(s) (I(s) - I(t))}{2\sigma_p^2}\right). \tag{3.11}$$

Figure 3.2 demonstrates the effect of the perceptual color constraint. The estimation in (b) employs $\sigma_p = 2.4$, and the flow around the right hand shows foreground-fattening effect. Applying a smaller variance (e.g., $\sigma_p = 1.8$) addresses the problem as seen in (c), but it causes over-segmentation artifact for the flow around the left hand. In contrast, the proposed constraint using the same variance $\sigma_p = 1.8$ addresses the problem without the artifact, as shown in (d).

3.3.3 Occlusion constraint

In the reference image, a certain area of a scene can disappear in the target image for some reasons, e.g., object movement and/or view change. Since the pixels in the area do not play any role in correlating the windows, we may ignore those pixels by



(a) (b)

(c) (d)

Figure 3.2: Effect of the perceptual color constraint. (a) The Beanbags sequence. (b) Estimation using the previous color constraint with $\sigma_p = 2.4$. (c) Estimation using the previous color constraint with $\sigma_p = 1.8$. (d) Estimation using the proposed constraint with $\sigma_p = 1.8$.

setting the weight to zero. Given an occlusion map $\mathcal{O} = \{v(s)|s \in \mathcal{V}\}$, where the binary variable $v(s)$ indicates if the pixel s is occluded or not, we define the weight function as follows:

$$w_s^{occ}(t) = 1 - v(t). \tag{3.12}$$

Although the occlusion map is not initially given, we can generate it by checking consistency, in the iterations for the incremental flow update. In each iteration, we additionally estimate backward flow from the target to the reference, and check consistency of a pixel using the following equation:

$$v(s) = \begin{cases} 0 & if \quad \|\mathbf{u}_s + \mathbf{u}'_{s'}\| < \gamma \\ 1 & else \end{cases} \tag{3.13}$$

where $\mathbf{u}_s$ means the forward flow at $\mathbf{x}_s$, $\mathbf{u}'_{s'}$ means the backward flow at $\mathbf{x}_{s'} = \mathbf{x}_s + \mathbf{u}_s$, and γ is a threshold parameter. In experiments, we found the optimal $\gamma = 2.5$ by quantitative evaluation of the Middlebury *test* dataset, which produces the minimum average end-point error, (i.e., EPE=0.310,) as shown in Table 3.1.

Figure 3.3 demonstrates the qualitative effect of the occlusion constraints. The example image contains complex motion boundaries with non-rigid movement and similar color distribution. The result with a non-weighted window presents the obvious foreground-fattening effect around object boundaries. Although utilizing the proximity and the color constraints can reduce such an effect, the occlusion constraints present much better results with more structured mesh around motion boundaries.

3.4 Efficient Optimization

To find the optimal solution for the MRF formulation in (5.1), we employ the TRW-S [11], which has shown state-of-the-art results [27] in many discrete framework applications. The asymptotic computational complexity of the TRW-S, in general, is $O(|\mathcal{V}||\mathcal{L}|^2)$. In our current framework, we may rewrite it as $O(|\mathcal{V}|L^4)$. Since

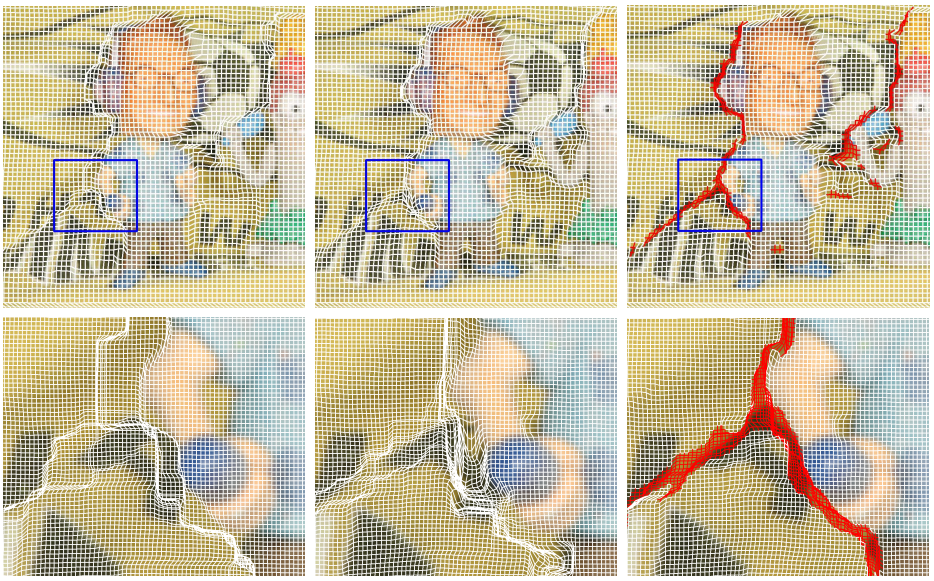


Figure 3.3: Flow estimation for the Mequon sequence (in part). Results are illustrated using mesh deformation with the occluded region shown in red. The bottom row shows magnified views of the top row (in the blue boxes) **Left:** Result using the non-weighted window. **Middle:** Result using the proximity and color constraints. **Right:** Result using the proposed constraints.

the complexity is dominated by the number of labels, and our method requires an adequate number of labels to yield plausible estimation results, we introduce a technique to address the complexity issue.

3.4.1 Node decomposition

We apply the node decomposition scheme [28], reducing the complexity to $O(|\mathcal{V}|L^2)$. The scheme decomposes the node $s \in \mathcal{V}$ into two nodes $s_x \in \mathcal{V}_x$ and $s_y \in \mathcal{V}_y$. We may define l_{s_i} as a random variable for a node s_i in some discrete sample space $\mathcal{L}_{s_i} = \{1, \ldots, L\}$, representing the quantized 1D displacement vector set

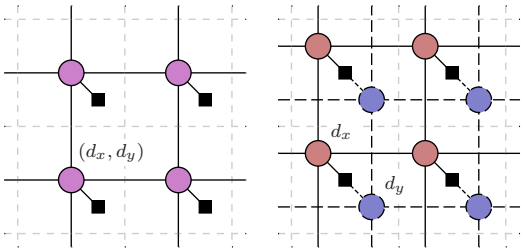


Figure 3.4: Conceptual illustration for the node decomposition. **Left:** The original MRF model. A node represents a label for 2D displacement vector: (d_x,d_y). **Right:** The original node is decomposed into two nodes representing labels for 1D vectors: d_x and d_y respectively. The unary term (shown in a black square) in the original MRF model becomes a pairwise term between the decomposed nodes.

$\mathcal{T}_{s_i} = \{u_{s_i}(1), \ldots, u_{s_i}(L)\}$ where $i \in \{x, y\}$. The original displacement vector $\mathbf{u}_s(l_s)$ corresponds to $(u_{s_x}(l_{s_x}), u_{s_y}(l_{s_y}))$. The original edge set $\mathcal{E}$ is decomposed into $\mathcal{E}_x$ and $\mathcal{E}_y$, and the new edge set $\mathcal{E}_{xy}$ is introduced, to account for the pairwise potential between the decomposed nodes. Figure 5.4 shows a conceptual illustration of the decomposition scheme. The original MRF formulation in (5.1) is updated as follows:

$$\sum_{(s_x,s_y)\in\mathcal{E}^{xy}} \Phi_{xy}(l_{s_x}, l_{s_y}) + \sum_{(s_x,t_x)\in\mathcal{E}^{x}} \Psi_{st}(u_{s_x}(l_{s_x}) - u_{t_x}(l_{t_x})) + \sum_{(s_y,t_y)\in\mathcal{E}^{y}} \Psi_{st}(u_{s_y}(l_{s_y}) - u_{t_y}(l_{t_y})). \quad (3.14)$$

We note the original unary potential Φ_s is updated to the pairwise potential Φ_{xy}, defined by the decomposed nodes. Unary potentials for these nodes are undefined, imposing no cost on any configuration. As the number of labels for a node reduces to L, the complexity of the TRW-S also reduces to $O(|\mathcal{V}|L^2)$.

In addition, the decomposition enables defining the pairwise potential Ψ_{st} as

linear to the label difference; that is, we may rewrite $\Psi_{st}(u_s(l_s) - u_t(l_t))$ as $\Psi'_{st}(l_s - l_t)$. Then we can apply the min-convolution algorithm [29] for the TRW-S, reducing the time complexity to $O(|\mathcal{V}|L)$. In experiments, we set $\Psi'_{st}(l_s - l_t) = \alpha|l_s - l_t|$, which is a parametric and robust convex penalizer.

3.5 Experimental Results

We validate our flow estimation method on the Middlebury flow dataset [30]. The dataset contains several image sequences of indoor and outdoor scenes, containing various real or synthetic objects.

We assumed the maximum deformation for each direction to be 32, and quantized each direction by 4 with the target precision $\mu = 0.05$. The size of correlation windows was fixed to 35×35. The parameters affecting the relative influence and strictness of the different constraints, were evaluated in Section 3.5.2, and fixed to optimal values for other experiments: $\sigma_g = 7.2$, $\sigma_p = 1.8$. By varying σ_g, we can control the effective area with non-negligible weight in the window, and so we may think σ_g implicates the actual window size. Other parameters were empirically tuned to $\alpha = 0.05$ and $\nu = 10^{-6}$.

3.5.1 Effect of individual constraints

To show the effects of each proposed constraint, Figure 3.5 provides quantitative analysis comparing estimation errors obtained with our full algorithm ("*full*") to errors computed with various constraints removed. Removing the perceptual constraint ("*w/o perc*") causes significant degradation compared to the full algorithm, particularly when using the large windows (e.g., $\sigma_p > 5$.) Not using the gradient

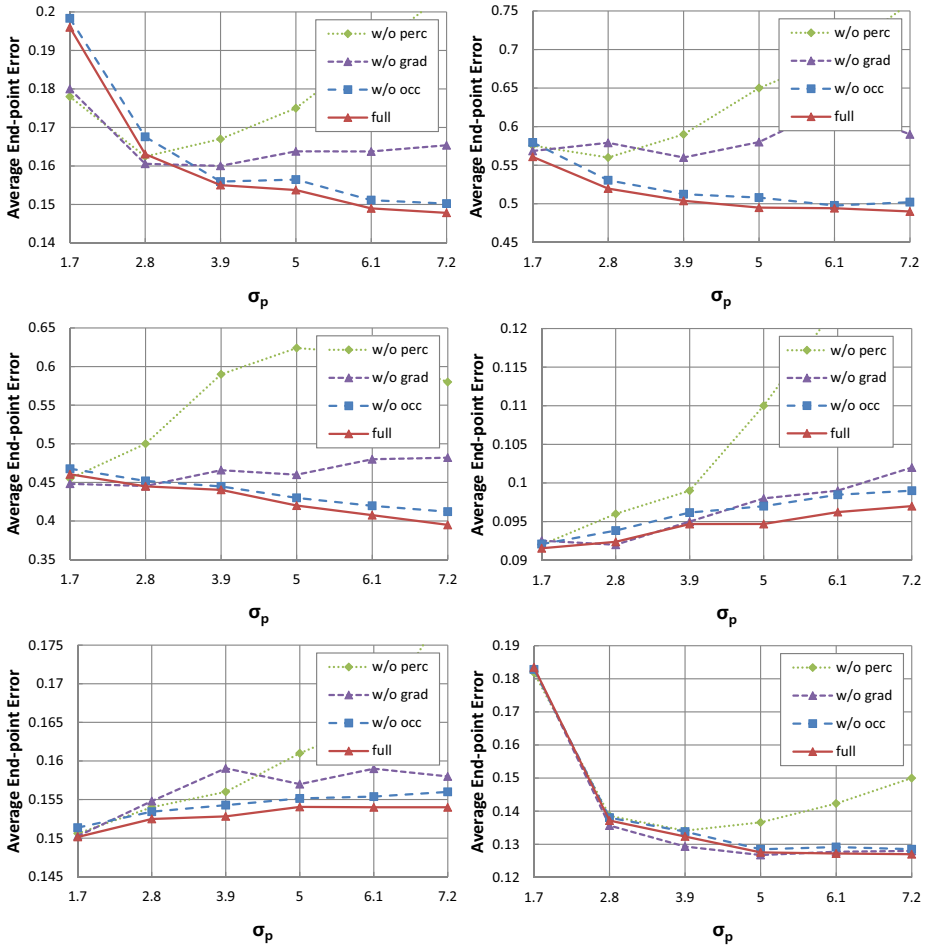


Figure 3.5: Quantitative evaluation for the effect of individual constraints. From top left to bottom right, we present average end-point errors of the Grove2, Grove3, Urban2, RubberWhale, Hydrangea, and Dimetrodon sequences, for varying window size.

structure constraint ("*w/o grad*") also leads to an amount of increase in errors, as the window size increases. Leaving out the occlusion constraint ("*w/o occ*") also shows worse results than the full algorithm; although the difference is not significant, due to the fact that the occluded region is sparse in general. Figure 3.6 shows

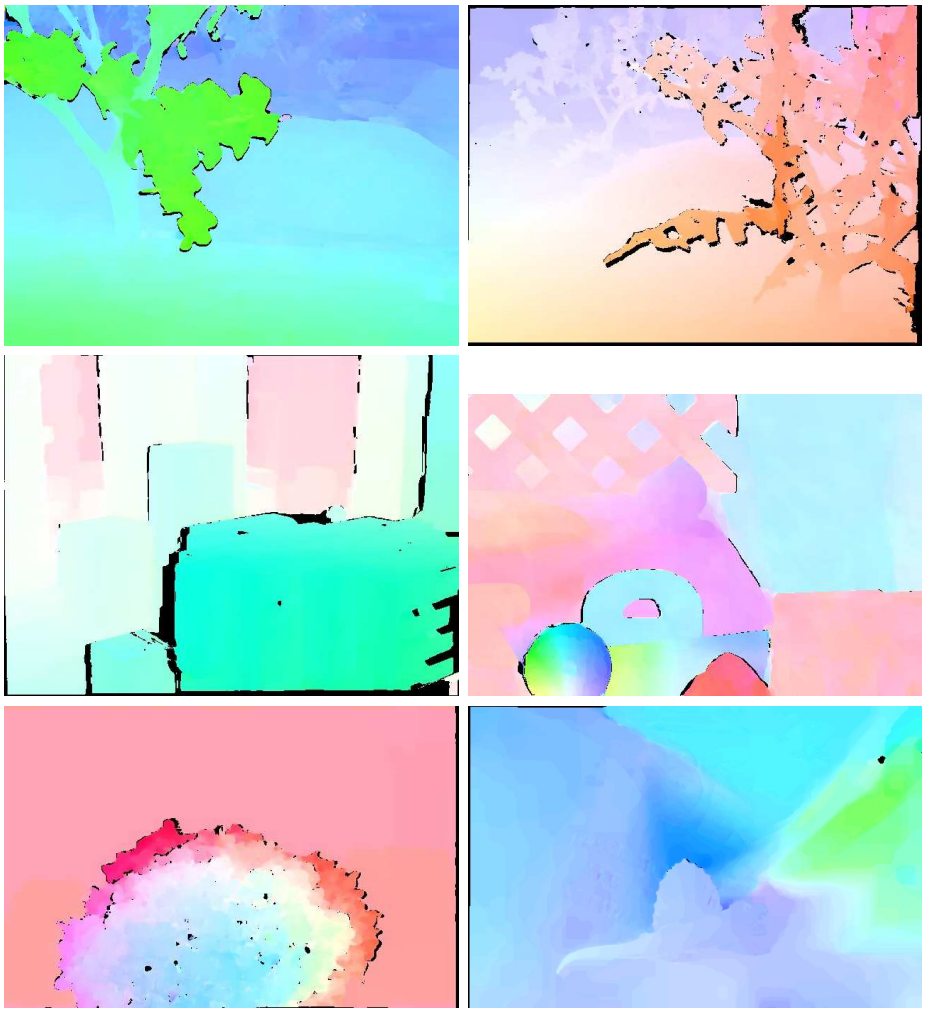


Figure 3.6: From top left to bottom right: Estimation results of the Grove2, Grove3, Urban2, RubberWhale, Hydrangea, and Dimetrodon sequences. The flow vectors are represented in HSI color space (direction: hue, magnitude: saturation, occlusion: black)

the visualization of the flow results, obtained with our full algorithm.

3.5.2 Comparison to previous constraints

We additionally validate our technique through comparisons with the previous support-weight constraints, employing the bilateral filtering-based weights [22,23]; i.e., proximity and color constraints which correspond to the gradient structure and perceptual color constraints. The occlusion constraint is excluded for fair comparison.

Figure 3.7 (a) compares the geometric constraints without the photometric constraints. Although the geometric constraints generally degrade performance as we apply larger windows, the proposed constraint ("*grad*") presents lower EPE than the previous constraint ("*prox*") for all varied parameters. Next, the photometric constraints is compared in Figure 3.7 (b). We chose a mid-size window (e.g., 15×15) and varied σ_p, from 0.6 to 4.8. The perceptual color ("*perc*") yields better estimations with relatively small variances, presenting the lowest EPE at $\sigma_p = 1.8$. Finally, we compare the combinations of the geometric and photometric constraints in Figure 3.7 (c). With the optimal color variances obtained from the previous analysis, we varied the window size by σ_g. Although the proposed constraints ("*grad+perc*") yielded slightly worse estimations for small windows, it outperforms the previous constraints ("*prox+color*") for large windows $\sigma_g > 5$.

3.5.3 Comparison to other methods

We also provide comparison to other top-performing methods, to validate overall performance of the full algorithm. Table 4.1 lists the seven best methods in the Middlebury Flow site [30] for the average end-point error, and the average angular error measured on different image part: whole image (*all*), motion boundary (*disc*,) and untextured region (*untext*). We also present the results from the previous con-

Table 3.2: Quantitative comparison with top-performing methods for the Middlebury evaluation data set

Average end-point error	Army			Mequon			Schefflera			Wooden			Grove			Urban			Yosemite			Teddy		
	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
MDP-Flow2	0.08	0.21	0.07	0.15	0.48	0.11	0.20	0.40	0.14	0.15	0.80	0.08	0.63	0.93	0.43	0.26	0.76	0.23	0.11	0.12	0.17	0.38	0.79	0.44
nLayers	0.07	0.19	0.06	0.22	0.59	0.19	0.25	0.54	0.20	0.15	0.84	0.08	0.53	0.78	0.34	0.44	0.84	0.30	0.13	0.13	0.20	0.47	0.97	0.67
Sparse-NonSparse	0.08	0.23	0.07	0.22	0.73	0.18	0.28	0.64	0.19	0.14	0.71	0.08	0.67	0.99	0.48	0.49	1.06	0.32	0.14	0.11	0.28	0.49	0.98	0.73
LSM	0.08	0.23	0.07	0.22	0.73	0.18	0.28	0.64	0.19	0.14	0.70	0.09	0.66	0.97	0.48	0.50	1.06	0.33	0.15	0.12	0.29	0.50	0.99	0.73
Classic+NL	0.08	0.23	0.07	0.22	0.74	0.18	0.29	0.65	0.19	0.15	0.73	0.09	0.64	0.93	0.47	0.52	1.12	0.33	0.16	0.13	0.29	0.49	0.98	0.74
TV-L1-MCT	0.08	0.23	0.07	0.24	0.77	0.19	0.32	0.76	0.19	0.14	0.69	0.09	0.72	1.03	0.60	0.54	1.10	0.35	0.11	0.12	0.20	0.54	1.04	0.84
IROF-TV	0.09	0.25	0.08	0.22	0.77	0.19	0.30	0.70	0.19	0.18	0.93	0.11	0.73	1.04	0.56	0.44	1.69	0.31	0.09	0.11	0.12	0.50	1.08	0.73
Adapt-Window	0.10	0.24	0.09	0.19	0.59	0.15	0.27	0.64	0.17	0.18	0.82	0.11	0.74	1.07	0.56	1.78	1.73	0.95	0.22	0.16	0.45	0.70	1.28	0.88
Bilateral-Window	0.11	0.27	0.10	0.20	0.62	0.16	0.28	0.65	0.19	0.24	1.16	0.15	1.00	1.38	0.89	1.81	2.06	0.96	0.25	0.18	0.57	0.90	1.71	1.25
Average average error	**Army**			**Mequon**			**Schefflera**			**Wooden**			**Grove**			**Urban**			**Yosemite**			**Teddy**		
	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
nLayers	2.80	7.42	2.20	2.71	7.2	2.55	2.61	6.24	2.45	2.30	12.7	1.16	2.30	3.02	1.70	2.62	6.95	2.09	2.29	3.46	1.89	1.38	3.06	1.29
MDP-Flow2	3.23	7.93	2.60	1.92	6.64	1.52	2.46	5.91	1.56	3.05	15.8	1.51	2.77	3.50	2.16	2.86	8.6	2.70	2.00	3.50	1.59	1.28	2.67	0.89
IROF++	3.17	8.7	2.61	2.79	9.6	2.33	3.43	8.9	2.38	2.87	14.8	1.52	2.74	3.57	2.19	3.20	9.7	2.71	1.96	3.45	1.22	1.80	4.06	2.50
Sparse-NonSparse	3.14	8.75	2.76	3.02	10.6	2.43	3.45	8.96	2.36	2.66	13.7	1.42	2.85	3.75	2.33	3.28	9.4	2.73	2.42	3.31	2.69	1.47	3.07	1.66
LSM	3.12	8.62	2.75	3.00	10.5	2.44	3.43	8.9	2.35	2.66	13.6	1.44	2.82	3.68	2.36	3.38	9.4	2.81	2.69	3.52	2.84	1.59	3.38	1.80
Classic+NL	3.20	8.7	2.81	3.02	10.6	2.44	3.46	8.8	2.38	2.78	14.3	1.46	2.83	3.68	2.31	3.40	9.1	2.76	2.87	3.82	2.86	1.67	3.53	2.26
TV-L1-MCT	3.16	8.5	2.71	3.28	10.8	2.60	3.95	10.5	2.38	2.69	13.9	1.45	2.94	3.79	2.63	3.50	9.8	3.06	2.08	3.35	2.29	1.95	3.89	2.71
Adapt-Window	4.07	9.3	3.54	2.42	8.0	1.99	3.47	9.0	2.05	3.55	17.0	1.97	3.34	4.21	2.82	5.93	14.8	4.83	4.32	4.61	5.39	3.27	5.89	3.16
Bilateral-Window	4.36	10.2	3.62	2.63	8.6	2.31	3.61	9.2	2.31	4.68	21.5	2.80	3.84	4.69	3.65	7.09	18.7	5.80	4.85	5.15	7.40	5.25	10.6	6.50

straints, shown as *Bilateral-Window*.

For the first four sequences (i.e., the Army, Mequon, Schefflera and Wooden,) our method presents very competitive results on the overall regions. We note these sequences are based on real scenes and provide accurate ground truth occlusion maps, by which the occluded regions are excluded in calculating the end-point/angular errors. We also note our method outperforms *Bilateral-Window*, especially on motion boundaries. The performance gain increases for the sequences containing an amount of occlusions, e.g., the Urban and Teddy.

In Figure 3.8, we also provide qualitative results for various real-world scenes. The first column of the figure demonstrates layered two input frames for each scene, showing various motions of objects. While the top-performing method [2] in the Middlebury Flow evaluation (shown in the third column) generally presents the best estimations, our method also shows competitive results. Compared to the method of Sun et al. [1] (shown in the second column), which is also famous for its state-of-

the-art performance, the proposed method yields better estimations particularly on regions with large displacements. (e.g., the beak of the duck and the right foot of the football player.)

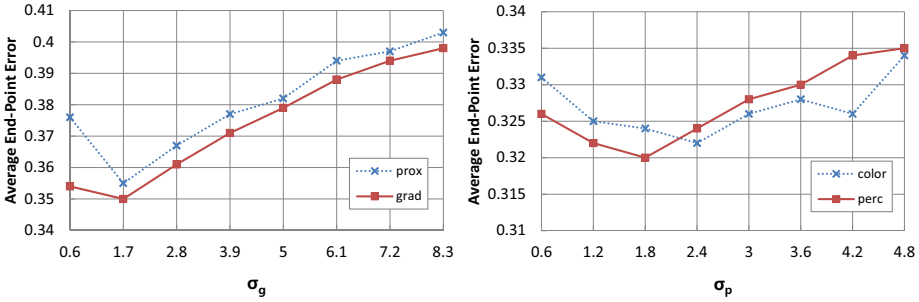
3.6 Discussion

In this chapter, we presented a new adaptive window correlation for optical flow estimation on the discrete MRF framework. A novel data cost design incorporating various constraints efficiently ignores inhomogeneous motion in correlation windows on object boundaries, helping to enlarge the window size to cover the aperture phenomenon. The effect of each constraint compared to the previous constraints has been shown with quantitative analysis. In order to reduce computational complexity and fully utilize image resolution, we utilized the decomposed scheme combined with the course-to-fine approach.

In future work, we plan to enhance the occlusion detection algorithm. The current algorithm adopts a symmetric approach, obtaining an occlusion map given a flow field and vice versa, implying that the field and map are not generated at once. Using discrete optimization, an integrated framework combining both energy models can provide a better solution, which is closer to the global minimum. More sophisticated occlusion reasoning may result in further improvement in flow estimation.

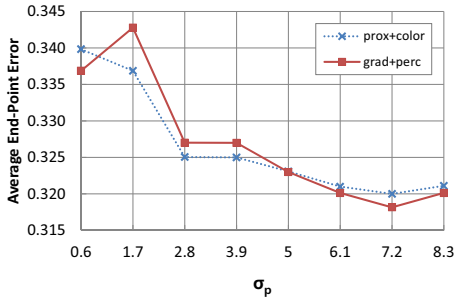
Our current implementation takes 935.4 seconds, on average, to find the estimation of a 640×480 image, with a 35×35 correlation window. We employ graphic hardware for parallel calculation of the data matching cost, which takes only 182.4 seconds. The rest of the time is taken for optimization on the CPU; and we believe significantly faster processing can be obtained with a full implementation that

computes message-passing based optimization on parallel graphics hardware [31].



(a)

(b)



(c)

Figure 3.7: Quantitative analysis using the Middlebury flow test dataset. (a) Comparison of the proposed gradient structure constraint and the previous proximity constraint. (b) Comparison of the proposed perceptual color constraint and the previous color constraint. (c) Comparison of combination of the proposed constraints and that of the previous constraints. The occlusion constraint is excluded for fair comparison.

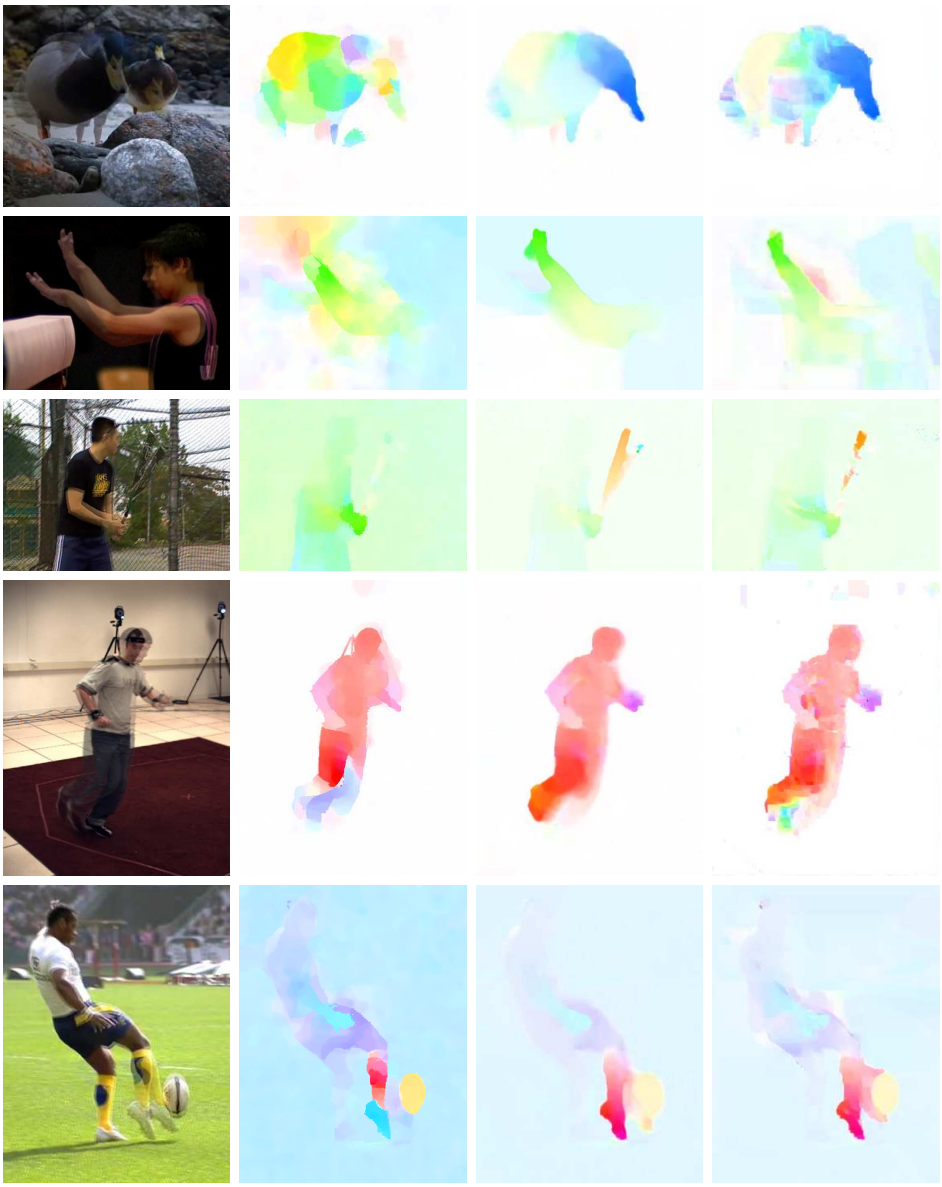


Figure 3.8: Qualtitative comparison with other methods. **Column 1** input frames, **Column 2** Sun et al. [1], **Column 3** Xu et al. [2]. **Column 4** Ours.

Chapter 4

Convolution Kernel Prior

4.1 Introduction

The linearized brightness constancy is frequently employed data matching criterion in the methods using the variational approaches. It assumes infinitesimal displacement of a pixel and may produce inaccurate estimation for images including objects with large motions. The window matching based approaches (e.g., our method in Chapter 3) inherently addresses the large displacement issue. However, it may suffer from images containing complex motions, such as rotational motion; since it generally assumes transitional model due to limited number of labels.

4.1.1 Previous work

Trying to preserve discontinuities, various regularization terms have been introduced. Many of them are based on diffusion tensors considering a global regularization process can be seen as a local diffusion of flow. Alvarez *et al* [32] proposed to adaptively control the degree of diffusivity according to the gradient magnitude of an image.

Earlier than this approach, Nagel [33] presented a novel regularizer using anisotropic diffusion along object boundaries. Showing plausible edge-preserving property, this regularizer has been adopted in numerous methods [34, 35] until nowadays. While these approaches employ smoothness priors reflecting local *image* structure, other researches [36, 37] focused on the *flow* structure to avoid over-segmentation artifact on textured area. Recently a new diffusion tensor combining both image and flow structure [38] is introduced and its modification [39] shows state-of-the-art performance on the Middlebury evaluation site [30].

In contrast, Xiao *et al* [40] presented a rather different type of regularizer. Assuming the diffusion tensor-based energy functional can be minimized through two individual updating processes, they convert the diffusion process to the corresponding convolution kernel filtering [41] applied to the intermediate flow estimation. The kernel is further replaced by the bilateral filter [42], known as an excellent edge-preserving smoother.

Meanwhile, Black *et al* [43] proposed to refrain from a quadratic penalizer for the cost calculation, which is claimed to excessively penalize outliers from a statistical viewpoint. Employing a linear penalizer, e.g., total variation for smoothness cost or L^1-norm for data cost [9, 35, 44] presented enhanced performances. In addition, statistical analysis for derivatives and brightness constancy errors in natural images showed the distribution is highly kurtotic and heavy tailed [45]. As the robust penalization becomes non-differentiable and non-convex, optimizing energy function with variational approach becomes more complicated; and prone to be trapped in local minima.

4.1.2 Proposed approach

Apart from this limitation, variational approaches also suffer from restricted form of data term. The data term for brightness constancy should be linearized based on the Taylor series approximation to make the functional differentiable. The estimated motion fields are assumed to be in small displacements and can not catch up with large deformations with non-linear movement. Traditional approach [9] uses successive pyramids in a coarse-to-fine fashion, arising another problem losing detail structures in images. A recent work [3] proposed a solution by discretizing the data cost functional and decoupling the energy for minimization; not guaranteed to find the global minimum.

Addressing these challenges, we propose a new energy model defined on the *discrete* MRF framework. On this framework, more options are available for the robust penalizer less concerning convexity and differentiability. The data term in our model inherently covers large displacement flow since the brightness constancy assumption does not have to be linearized. We also manage the issues of the image-based regularizers as well, proposing a new convolution kernel based on the bilateral filter. Using perceptual information, this kernel adapts to the local statistics avoiding over-segmentation as well as over-smoothing without parameter tuning.

The rest of this paper is organized as follows. Section 2 defines a new regularizer on the discrete framework, which we named as *convolution kernel prior*. In Section 3 we introduce the new adaptive kernel design and show its advantages. Section 4 gives a strategy to enhance the efficiency of the algorithm and Section 5 presents experimental results evaluating the proposed model. We finalize this work by providing the conclusion and the future work in Section 6.

4.2 Convolution Kernel Prior

We briefly introduce the two-step updating procedure [40] and derive the new regularizer. The derivation is also possible using the definitions from [46] where discrete regularization term for p-Dirichlet energy is theoretically defined. Note the proposed model can be extended to other types of neighborhood smoothness prior with diffusion tensor as well as non-local prior with arbitrary graph structure.

An energy functional is defined to find appropriate dense flow vector $\mathbf{u}$ on image domain Ω

$$E(\mathbf{u}) = E_d(\mathbf{u}) + E_s(\nabla\mathbf{u}) = \int_\Omega (e_d(\mathbf{u}) + e_s(\nabla\mathbf{u})d\mathbf{x}, \tag{4.1}$$

where e_d and e_s represent data matching cost and smoothness cost respectively. Minimizing (4.1), we apply the Euler-Lagrange equation to iteratively find the answer yielding the updating process as follows:

$$\frac{\partial\mathbf{u}}{\partial\tau} = \mathbf{u}^\tau - \mathbf{u}^{\tau-1} = -\left(\frac{\partial e_d(\mathbf{u})}{\partial\mathbf{u}} - \text{div}\left(\frac{\partial e_s(\nabla\mathbf{u})}{\partial\nabla\mathbf{u}}\right)\right).$$

where τ indicates a time step in the iteration. As proposed in [40], we decouple this updating process into a two-step procedure; such that,

$$\mathbf{u}^{\tau'} - \mathbf{u}^{\tau-1} = -\frac{\partial e_d(\mathbf{u})}{\partial\mathbf{u}}, \tag{4.2}$$

$$\mathbf{u}^\tau - \mathbf{u}^{\tau'} = \text{div}\left(\frac{\partial e_s(\nabla\mathbf{u})}{\partial\nabla\mathbf{u}}\right), \tag{4.3}$$

where τ' means the intermediate time step. Employing a certain diffusion tensor D to define $e_s(\nabla\mathbf{u})$, we can rewrite (4.3) as

$$\mathbf{u}^\tau - \mathbf{u}^{\tau'} = \text{div}\left(\frac{\partial(\nabla\mathbf{u}^T D\nabla\mathbf{u})}{\partial\nabla\mathbf{u}}\right) = \text{div}\left(D\nabla\mathbf{u}^{\tau'}\right).$$

We develop this divergence form into corresponding oriented Laplacian formulations, whose solution can be shown in terms of convolution kernel filtering [41] as follows:

$$\mathbf{u}^{\tau} = G * \mathbf{u}^{\tau'},$$

where the kernel G represents a 2-D oriented Gaussian kernel defined by eigenvectors of the diffusion tensor D. This equation gives an inspiration for a new smoothness prior: the prior can be formulated with the difference between the original flow and the filtered flow with the convolution kernel G corresponding to the diffusion scheme. We replace the smoothness term in (4.1) with

$$E_s(\mathbf{u}) = \int_{\Omega} \Psi(\mathbf{u} - G * \mathbf{u}) d\mathbf{x},$$

where $\Psi(\cdot)$ represents a certain penalizing function.

For the corresponding discrete MRF model, let $\mathcal{G}$ be an undirected graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$. A node in $\mathcal{V}$ corresponds to a pixel in an input image. Let l_s be a random variable for a node s in some discrete sample space $\mathcal{L}_s = \{1, \ldots, L\}$, representing quantized displacement vector set $\mathcal{T}_s = \{\mathbf{u}(1), \ldots, \mathbf{u}(L)\}$. Discrete analog to (4.1) can be given as,

$$E(\mathbf{l}) = \sum_{s \in \mathcal{V}} \left(\Phi_s(l_s) + \Psi(\mathbf{u}(l_s) - G_s * \mathbf{u}(l_s)) \right), \tag{4.4}$$

with $\Phi_s(\cdot)$ defining the data cost for a node s. Transforming the convolution into weighted sum, we obtain

$$E(\mathbf{l}) = \sum_{s\in\mathcal{V}} \left(\Phi_s(l_s) + \Psi \left(\mathbf{u}(l_s) - \frac{\sum_{t\sim s} w_{st}\mathbf{u}(l_t)}{\sum_{t\sim s} w_{st}} \right) \right)$$
$$= \sum_{s\in\mathcal{V}} \left(\Phi_s(l_s) + \Psi \left(\sum_{t\sim s} \bar{w}_{st}(\mathbf{u}(l_s) - \mathbf{u}(l_t)) \right) \right), \tag{4.5}$$

where $\bar{w}_{st}$ means the normalized weight.

Unfortunately, this energy formulation involves higher order clique potentials inducing practically intractable complexity for current discrete optimization techniques. Managing this challenge, we suggest to find the solution for an upper bound equation of (4.5) which guarantees to make lower energy for the original equation. Without loss of generality, we assume the penalizing function $\Psi(\cdot)$ as convex, yielding the upper bound equation defined by,

$$E^{UB}(\mathbf{l}) = \sum_{s\in\mathcal{V}} \left(\Phi_s(l_s) + \sum_{t\sim s} \bar{w}_{st}\Psi\left(\mathbf{u}(l_s) - \mathbf{u}(l_t)\right) \right)$$
$$= \sum_{s\in\mathcal{V}} \Phi_s(l_s) + \sum_{(s,t)\in\mathcal{E}} \bar{w}_{st}\Psi\left(\mathbf{u}(l_s) - \mathbf{u}(l_t)\right). \tag{4.6}$$

This equation eventually converts the higher order clique potentials into a highly connected graph structure consisting only of unary and pairwise terms. Note the proposed model does not need to depend on the support window-based data matching [47], the widespread method on the discrete framework to address the aperture problem. In fact, since the window-based matching scheme generally assumes only transitional movement of objects, it often generates false matching on the regions with severe non-rigid motion or rotation. Figure 4.1 gives an example of this limitation where a synthetic basketball is rotated on smoothly varying background. The window-based matching (with simple neighborhood prior) moderately works on the background region but fails to find accurate flow on the rotated object.

Figure 4.1: Flow estimation involving rotation. **Left top and bottom:** Input images with a rotating basketball synthesized on smoothly varying background (part of the Schefflera sequence.) **Center and right top:** Flow and deformed image from 11×11 support window-based matching with 8-neighborhood prior. **Center and right bottom:** Flow and deformed image from the proposed model. The flow images are illustrated by the HSI color space (direction: hue, magnitude: saturation)

4.3 Adaptive Regularizer

The main issue for the regularizer is how to preserve accurate motion boundaries. Numerous edge-preserving regularizers have been proposed so far; particularly, image-driven regularizers (e.g., the anisotropic diffusion tensor) have shown plausible performances. They adaptively give weak smoothing effect on areas where a certain intensity change is detected. This strategy works very fine on a region where the intensity-based segments are identical to the actual motion segments, but shows some over-segmentation artifact on textured regions sharing an identical motion. Several works [38, 39] presented flow-driven smoothers based on intermediate flow estimation; however, this approach requires an accurate initial estimation. Otherwise, it may recursively worsen the following estimation, commonly resulting in over-smoothing artifact.

This work adopts a regularizing kernel based on the bilateral filtering. We define the weight in (4.5) using the product of two factors; i.e., proximity and color, such that $w_{st} = w^p_{st} * w^c_{st}$ where

$$w^p_{st} = \exp\left(-\frac{||(\mathbf{x_s} - \mathbf{x_t})||^2}{\sigma_p}\right), \tag{4.7}$$

$$w^c_{st} = \exp\left(-\frac{||I(\mathbf{x_s}) - I(\mathbf{x_t})||^2}{\sigma_c}\right). \tag{4.8}$$

The problem is, this regularizer is also image-driven and suffer from the similar drawback. In specific, the performance is much influenced by the parameter σ_c in the image-driven factor w^c_{st}. If it is set too small, the result undergoes over-segmentation on textured regions; while in opposite case, the motion boundaries are often overly smoothed.

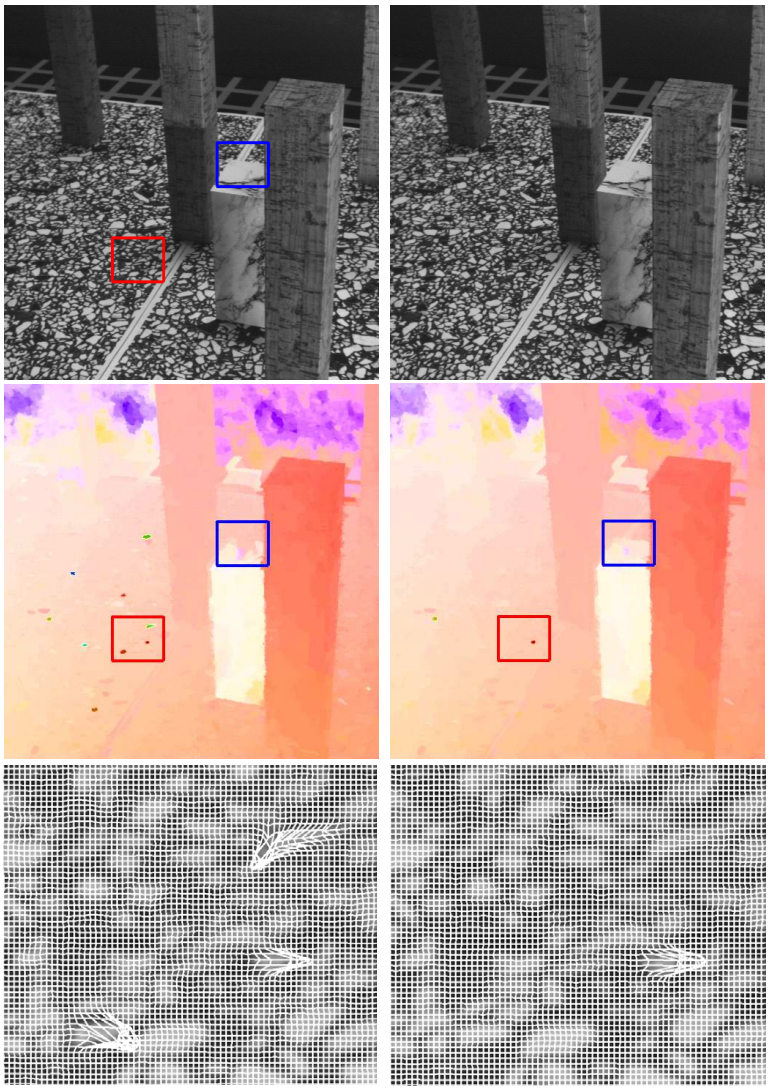


Figure 4.2: Comparison between the bilateral kernel and the proposed kernel. **Top row:** Input images from the Marble sequence. **Middle row:** Flow estimation resulting from both kernels. **Bottom row:** Detailed views in the red box with the flow illustrated using mesh. The proposed kernel shows better regularizing performance on the textured area without over-smoothing artifact (in the blue box.)

To this end, we propose to replace the Euclidean color difference in (4.8) with perceptual difference defined by the Mahalanobis distance. The new weight factor gives an effect adaptively changing the intensity (color) variance parameter σ_c based on local statistics. We rewrite the color weighting factor as follows:

$$w_{st}^{c} = \exp\left(-\frac{\left(I(\mathbf{x_s}) - I(\mathbf{x_t})\right)^T \mathbf{T}_c \left(I(\mathbf{x_s}) - I(\mathbf{x_t})\right)}{\sigma_c}\right),$$

with a certain anisotropic tensor $\mathbf{T}_c$ on color space defined by

$$\mathbf{T}_c = \frac{1}{1+\nu^2}(e_1 e_1^T + \nu^2 \mathbf{1}),$$

where $\mathbf{1}$ is a 3×3 identity matrix and ν is a parameter controlling the degree of isotropy. e_1 is the largest eigenvector of $\mathbf{\Sigma}_c^{-1}$, the inverse of the color variance-covariance matrix defined by

$$\mathbf{\Sigma}_c = \begin{bmatrix} \sigma_{R,R} & \sigma_{R,G} & \sigma_{R,B} \\ \sigma_{G,R} & \sigma_{G,G} & \sigma_{G,B} \\ \sigma_{B,R} & \sigma_{B,G} & \sigma_{B,B} \end{bmatrix}.$$

where $\sigma_{X,X}$ means the intensity variance of adjacent pixels around $\mathbf{x_s}$, for each color attribute X and $\sigma_{X,Y}$ is the covariance between attribute X and Y.

Figure 4.2 illustrates an example comparing the original bilateral kernel and the proposed kernel. Both kernels share the same parameter setting. In the result from the bilateral kernel, distinctive outliers are detected on textured area as marked with the red box; while the blue box points out over-smoothed flow examples due to the similar intensity distribution as shown in the input images. Note one can not simultaneously address both artifacts by tuning the parameter σ_c because they are in trade-off relation. In contrast, the proposed kernel is less affected by this

limitation since it conceptually employs larger σ_c on textured areas; at the same time, employs smaller σ_c on homogeneous areas enhancing discrimination.

4.4 Optimization

Although the upper bound energy (4.6) successfully converts higher order terms into pairwise edge terms, it still yields acute computational complexity due to the highly connected graph structure. In this section we present two approaches to efficiently reduce the complexity of the proposed model.

4.4.1 Coarse-to-fine approach

In order for the discrete approach to follow up the natural flow field with subpixel accuracy, the quantized unit needs to be small enough. However, *dense* quantization of the vector space will cause radical increase in computational complexity as well as memory requirement. We employ a hierarchical approach in a coarse-to-fine manner [48] acquiring plausible accuracy with limited number of labels.

The algorithm starts with sparse quantization for the maximum deformation range. After the first optimization is completed, we obtain a solution with rough motion field. For further iterations, we set a smaller deformation range starting from the moved position by the current motion field. In this fashion, the algorithm reduces the quantization unit and incrementally forms accurate motion field.

When the deformation is over the maximum range of the algorithm, we use a pyramidal approach to coarsen the image and find rough solution with scaled motion. Backing to the original scale, dense flow field is estimated by interpolating the coarse solution and provided as initial flow field for the further subpixel estimation.

4.4.2 Node decomposition

Executing a discrete optimization procedure, we generally define a pairwise cost table to be directly referred during the procedure. Online calculation of the cost is not preferred due to serious degradation of convergence speed. In our case, the coarse-to-fine approach may define slightly different pairwise interaction on each edge in the graph; that is, using a pre-defined and fixed pairwise cost table is not available. In consequence, the algorithm requires vast amount of memory space amounting to $O(|\mathcal{E}||\mathcal{L}|^2)$ to keep the cost table for each edge.

One possible solution is employing the min-convolution algorithm [49] with $\Psi(\cdot)$ defining a parametric penalizer; e.g., linear or quadratic to the difference of *labels*. Then the algorithm allows each edge to store only a few parameters instead of the whole cost table. However, $\Psi(\mathbf{u}(l_s)-\mathbf{u}(l_t))$ in (4.6) will hardly define such a penalizer since a quantized *2-D* vector $\mathbf{u}(l_s)$ can not be defined as linear to the corresponding label l_s.

We address this issue through the node decomposition [50]. A node $s \in \mathcal{V}$ is decomposed into two nodes s_x and s_y representing one dimensional displacement vectors, as described in Figure 5.4. In consequence, the quantized *1-D* vector $\mathbf{u}_i(l_{s_i})$ can be defined as linear to the corresponding label l_{s_i}; e.g., $\mathbf{u}_x(l_{s_x}) - \mathbf{u}_x(l_{t_x}) = \beta(l_{s_x} - l_{t_x})$ where β indicates a scale parameter. Note that the actual virtue of the node decomposition lies in the fact that the number of labels for a node is decreased from $|\mathcal{L}|$ to $\sqrt{|\mathcal{L}|}$, considerably reducing computational complexity for the discrete optimization. The original unary potential Φ_s is determined by the two labels; i.e., transformed to pairwise potential. The MRF formulation in (4.6) subsequently becomes

Table 4.1: Quantitative analysis on the Middlebury evaluation data set. Only five top-performing results are listed for the average angular and end-point error. The least errors are written in bold and underlined for each column.

Average angle error	Army			Mequon			Schefflera			Wooden			Grove			Urban			Yosemite			Teddy		
	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Complementary OF	4.44	11.20	4.04	2.51	9.77	1.74	3.93	10.60	2.04	3.87	18.80	2.19	**3.17**	**4.00**	2.92	4.64	13.80	3.64	**2.17**	**3.36**	2.51	3.08	7.04	3.65
Adaptive	**3.29**	9.43	**2.28**	3.10	11.40	2.46	6.58	15.70	2.52	3.14	15.60	1.56	3.67	4.46	3.48	**3.32**	13.00	**2.38**	2.76	4.39	**1.93**	3.58	8.18	2.88
Proposed	4.19	**9.27**	3.60	**2.40**	**8.21**	**1.65**	3.40	**8.96**	**1.84**	**2.87**	14.40	**1.44**	3.36	4.15	3.07	6.35	16.10	4.90	4.21	4.80	6.03	3.29	5.99	2.82
Aniso. Huber-L1	3.71	10.10	3.08	4.36	13.00	3.77	6.92	15.30	3.60	3.54	15.90	2.04	3.38	4.45	**2.47**	3.88	**12.90**	2.74	3.37	4.36	2.85	3.16	7.52	2.90
DPOF	5.12	12.90	3.49	3.07	10.30	2.44	**3.09**	7.47	2.43	3.42	**12.90**	2.41	3.55	4.56	3.35	4.69	14.20	5.14	3.59	4.67	3.83	**2.00**	**4.93**	**1.65**
Average end-point error	**Army**			**Mequon**			**Schefflera**			**Wooden**			**Grove**			**Urban**			**Yosemite**			**Teddy**		
	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Adaptive	**0.09**	0.26	**0.06**	0.23	0.78	0.18	0.54	1.19	0.21	0.18	0.91	0.10	0.88	1.25	0.73	0.50	1.28	0.31	0.14	0.16	**0.22**	0.65	1.37	0.79
Complementary OF	0.11	0.28	0.10	**0.18**	0.63	**0.12**	0.31	0.75	0.18	0.19	0.97	0.12	0.97	1.31	1.00	1.78	1.73	0.87	**0.11**	**0.12**	**0.22**	0.68	1.48	0.95
DPOF	0.13	0.35	0.09	0.25	0.79	0.19	**0.24**	**0.49**	0.21	0.19	**0.62**	0.15	**0.74**	**1.09**	**0.49**	0.66	1.80	0.63	0.19	0.17	0.35	**0.50**	**1.08**	**0.55**
Aniso. Huber-L1	0.10	0.28	0.08	0.31	0.88	0.28	0.56	1.13	0.29	0.20	0.92	0.13	0.84	1.20	0.70	**0.39**	**1.23**	**0.28**	0.17	0.15	0.27	0.64	1.36	0.79
Proposed	0.11	**0.25**	0.09	**0.18**	**0.59**	0.13	0.27	0.64	**0.16**	**0.15**	0.78	**0.09**	0.82	1.14	0.71	1.90	1.90	0.99	0.23	0.17	0.49	0.77	1.44	0.91

$$E^{UB}(\mathbf{l}) = \sum_{(s_x,s_y)\in\mathcal{E}^{xy}} \Phi_s(l_{s_x}, l_{s_y}) + \sum_{(s,t)\in\mathcal{E}^x\cup\mathcal{E}^y} \bar{w}_{st}\Psi(\beta(l_{s_i} - l_{t_i}) + k_{s_i}), \tag{4.9}$$

where k represents some offset value stemming from the coarse-to-fine approach. Despite the offset value, Ψ holds convex to the label difference and thus the min-convolution algorithm is still applicable for optimizing (4.9). In experiments we set $\Psi(\cdot) = \alpha|\cdot|$, which is a parametric and convex penalizer.

In order to utilize the node decomposition, the optimization method should be based on the message-passing algorithm. We employ the TRW-S [51] which is proven to give state-of-the-art results [52] in many intensive test cases; moreover, use of the min-convolution algorithm enable the TRW-S to run in $O(|\mathcal{V}|\sqrt{|\mathcal{L}|})$ time, as fast as the Graph-cuts [53].

4.5 Experimental Results

Experiments consist of three parts. First, the overall performance is evaluated through the evaluation data set provided by the Middlebury Flow site [30, 54]. Second, we compare the proposed method with the control group replacing two key

contribution factors; i.e., the convolution kernel prior and the adaptive kernel. Finally, we show the proposed method is also applicable to images containing large displacements. We start with some details of our implementation followed by experimental environments and actual parameters for the replication of our work.

Data matching criteria Defining the data cost functional in (4.6), we employ the NCC (Normalized Cross Correlation) [55] combined with the gradient constancy measure as follows:

$$\Phi_s(l_s) = \lambda\left(1 - \frac{\sum_W J_1(\mathbf{x}_u)J_2(\mathbf{x}'_u)}{\sqrt{\sum_W |J_1(\mathbf{x}_u)|^2}\sqrt{\sum_W |J_2(\mathbf{x}'_u)|^2}}\right) + (1-\lambda)\bar{\arccos}\left(\frac{\nabla I_1(\mathbf{x}_u)^T \nabla I_2(\mathbf{x}'_u)}{|\nabla I_1(\mathbf{x}_u)||\nabla I_2(\mathbf{x}'_u)|}\right), \quad (4.10)$$

where $J(\mathbf{x}_u) = I(\mathbf{x}_u) - \bar{I}(\mathbf{x}_s)$, $\mathbf{x}'_u = \mathbf{x}_u + \mathbf{u}(l_u)$ while u is an element in a node set W supporting the node s. The balancing parameter λ is set to 0.3 and the support window size is set to 3×3 squared pixel, which is the minimum size implementing the NCC.

Experimental environments All the experiments are performed on a 2.60GHz Intel QuadCore CPU. We use a two-level pyramid with the higher level search range covering ten pixels for each direction while lower one is set to 2.5 pixels. Conceptual coverage is up to 20 pixels and corresponding search range becomes $(2 \times 20 + 1)^2$ squared pixels for the maximum deformation. The number of quantization for each direction is set to 20 generating $41(= 2 \times 20 + 1)$ labels for a decomposed node, which would be $1684(= 41 \times 41)$ labels without the decomposition. Coarse-to-fine precision

level is set to three, enabling the flow accuracy up to 0.125 pixel for each direction. The edges with weight factor less than $\exp(-4\sigma_p)$ are ignored at the optimization process. This strategy generates 72.24 connected neighbors per node on average for the Urban sequence. In the meantime, we have few parameters to be empirically tuned. The linear penalizing term α (= 2.0) also works as a balancing term between data and smoothness cost. We set $\sigma_p = 3.5$, $\sigma_c = 10.0$ and $\nu = 0.1$ respectively. The scale parameter β in (4.9) does not require tuning; it is initialized to 1 at the highest pyramid level and further recalculated according to the precision level.

4.5.1 Overall performance

The evaluation data set contains various tough situations such as non-rigid motion, rotation, textured region and large displacement. Table 4.1 shows the average angular error and end-point error measured on different criteria: overall image (all), motion boundary (disc) and homogeneous region (untext). We list five top-performing results, with the least error emphasized for each item. The analysis suggests our method is highly competitive with other state-of-the-art methods on discontinuous areas as well as textureless regions.

4.5.2 The control group

Figure 4.1 have qualitatively demonstrated a simple neighborhood system cannot properly handle non-linear motion as the proposed prior. Qualitative analysis in Figure 4.2 and Figure 4.3 suggest the proposed adaptive kernel provides better performance than the bilateral kernel, on textured area as well as motion boundaries.

Table 4.2: Quantitative analysis on the Middlebury test data set. We compare the proposed method with two control group; replacing the convolution kernel prior with the 8-neighborhood prior on the support window, and the adaptive kernel with the bilateral kernel.

Average angle error	RubberWhale	Hydrangea	Dimetrodon	Grove2	Grove3	Urban2	Urban3	Venus
8-neighborhood	**3.09**	2.37	**2.92**	3.15	7.24	4.47	9.46	7.50
Bilateral kernel	3.27	1.98	3.65	2.18	5.94	2.76	3.72	**6.49**
Proposed	3.12	**1.89**	3.51	**2.00**	**5.38**	**2.63**	**3.89**	6.69
Average end-point error	**RubberWhale**	**Hydrangea**	**Dimetrodon**	**Grove2**	**Grove3**	**Urban2**	**Urban3**	**Venus**
8-neighborhood	**0.09**	0.19	**0.15**	0.24	0.70	0.50	1.34	0.49
Bilateral kernel	0.10	0.16	0.17	0.16	0.61	0.43	0.59	**0.41**
Proposed	**0.09**	**0.15**	0.17	**0.14**	**0.54**	**0.37**	**0.57**	0.47

Figure 4.3: Flow estimation on the Middlebury test data set without the groundtruth flow (high-speed camera sequences.) **Left column:** The 10th frames in the Beanbag, the Dogdance **Center column:** Results from the bilateral kernel. **Right column:** Results from the proposed kernel. The proposed model shows better smoothing result inside motion segments while keeping sharp boundaries. The HSI color code is changed for better visualization of the difference.

Figure 4.4: Flow estimation on the Middlebury test data set without the groundtruth flow (high-speed camera sequences.) **Left column:** The 10th frames in the Mini-cooper and the Walking sequence. **Center column:** Results from the bilateral kernel. **Right column:** Results from the proposed kernel. The proposed model shows better smoothing result inside motion segments while keeping sharp boundaries. The HSI color code is changed for better visualization of the difference.

We also present the quantitative comparison with these control group, using the Middlebury *test* data set of which the groundtruth is available. For the fair comparison, the 8-neighborhood system uses 11×11 support window with adaptive weighting [47] and the bilateral kernel shares the same parameter setting with the proposed one. As shown in Table 4.2, the proposed model generally shows smaller errors (in 11 out of 16 items) than these two control group.

4.5.3 Large displacements

We present qualitative comparison with the work of Brox *et al* [3] addressing large displacement problems. The video sequences provided in the website contain various fast motions occurring in real scenes. Figure 4.5 presents results for the several key frames in the Tennis sequence, including extremely large displacements. As can be seen, the proposed method not only catch up with large deformations but find more accurate motion boundaries.

4.6 Discussion

This work provides a discrete analog to the historically well-developed variational methods and benefit from both approaches. The data cost in our model does not require linearization to be differentiable; hence, inherently covers large displacement problems. The convolution prior model is shown to be more powerful and flexible than the simple neighborhood system with the support window, which has been frequently used in the literature of the discrete framework. The new adaptive kernel generalize the bilateral kernel and presents competitive results on motion boundaries

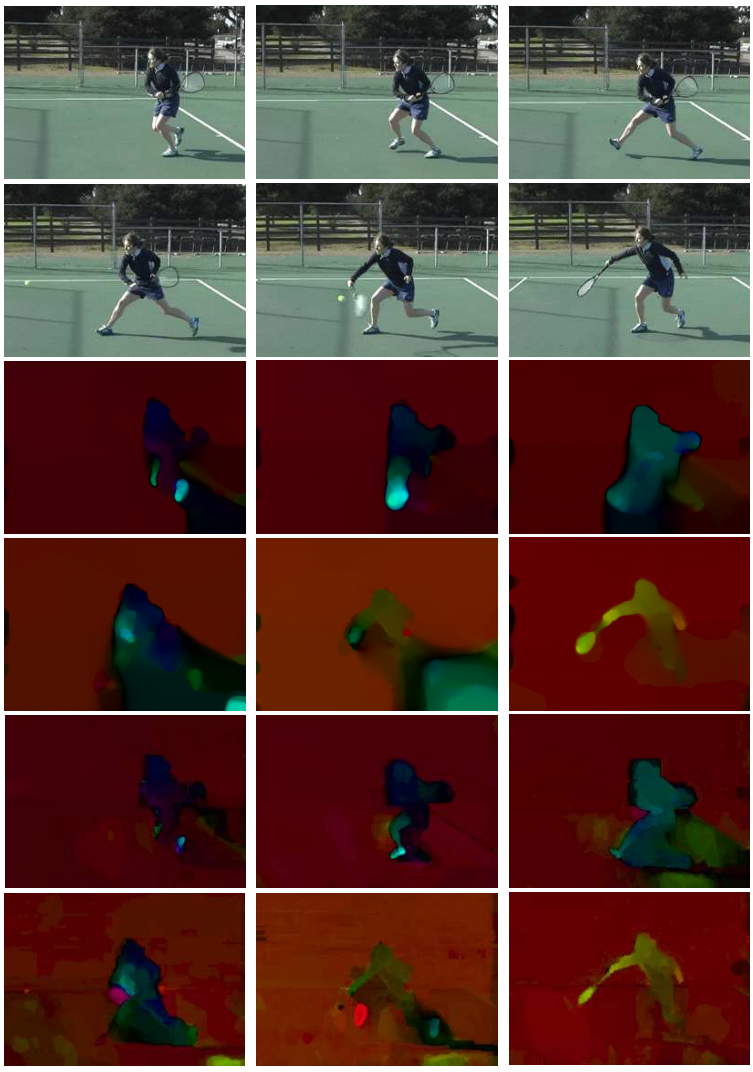


Figure 4.5: Flow estimation on large displacements. **Row 1,2:** Frame 34, 37, 40, 43, 46 and 49 in the Tennis sequence. **Row 3,4:** Results from the work of Brox *et al* [3]. **Row 5,6:** Results from the proposed model. All the results are estimation for the frame and the right next frame e.g., Frame 34 and 35.

as well as textured motion segments without cumbersome parameter tuning.

For the future work we consider following issues. Despite the reduced complexity, the running time for highly connected graph still requires certain amount of time; e.g., 5872 seconds for the 640×480 Urban sequence. As the data cost calculation time is easily shortened by parallelization techniques using GPU, we expect a faster algorithm will be possible in the near future by parallelizing BP [56].

Next, since the penalizer $\Psi(\cdot)$ in (4.5) should be convex to refrain from the higher order clique potential, use of non-convex robust function (e.g., truncated linear) for the regularizer is rather limited. A verification is left for the future work if the result can be seriously enhanced by directly using higher order cliques with a robust penalizer.

Chapter 5

Sparse Occlusion Detection via Window Matching

5.1 Introduction

In estimating optical flow between a reference and a target image frames, *occlusion* refers to a certain region of the reference image that does not correspond to any region in the target image due to, e.g., movement of objects and/or view change. Unless properly defined, occlusion may degrade the quality of estimation, particularly on object boundaries, and may lead to severe performance degeneration in many applications of optical flow estimation; for example, frame interpolation [57], motion segmentation, motion layer ordering [58], and motion compensated coding.

A convincing method to find exact occlusion is grasping exact motion of all objects in the images; inversely, if we obtain exact occlusion in advance, the accuracy of flow estimation will be much improved. Unfortunately, none of the exact motion or the exact occlusion is given in advance in most of cases, and so it is very difficult

to obtain highly accurate optical flow and occlusion at the same time.

5.1.1 Related work

Various approaches have been presented for optical flow estimation and occlusion detection. Many works individually estimate optical flow, and then detect occlusion based on the estimated flow, and iterate these steps until convergence. One simple approach to detect occlusion given flow estimation is thresholding the residual of subtracting warped target image from the reference image [18]. Strecha et al. [59] introduced a probabilistic criterion employing histogram of image contents. Other researches [60] define occlusion by checking symmetric consistency of forward and backward flows. Xu et al. [2] employed an observation that if a certain point in the target image is accessible by multiple pixels in the reference through forward warping, the point may probably be occluded. They refined the estimated flow using this probability map of accessibility. These approaches may suffer from the fact that they depend on initial flow results which could be incorrectly estimated in occluded area, and so the subsequent iteration may also yield erroneous results accordingly. Moreover, they require additional computational cost, e.g., for obtaining backward flow or the occlusion probability map.

Occlusion has also been a big issue in stereo matching problems. Zitnick et al. [61] proposed to iteratively update a 3D disparity array using the uniqueness and smoothness constraint, detecting occlusion by thresholding. The uniqueness constraint implies that each pixel in the target should have at most one correspondence to the reference. In [62], Kolmogorov et al.showed promising results by applying Graph-cuts algorithm [10] to efficiently enforce the uniqueness constraint. Other method [63] used backward disparity and visibility maps to obtain symmetric occlu-

sion detection using iterative optimization with Belief Propagation [64]. These methods generally find solutions in discrete sample spaces, and for the *two-dimensional* flow estimation problem, the size of space as well as the computational complexity may exponentially increase.

Recently, Ballester et al. [65] utilized an assumption that an occluded pixel may be visible in the previous frame to the reference frame, but their approach is limited to the case that multiple frames are provided, and motion across the frames is relatively simple. In [66], Ayvaci et al. showed a new model incorporating a cost for occlusion which is supposed to be very sparse with infinitesimal time interval. While this method presents state-of-the-art performance in detecting occlusion, it degenerates the performance of flow estimation as the process iterates. Also the performance can be very sensitive to the threshold value controlling the penalty of sparseness.

5.1.2 Proposed approach

In this chapter, we aim to simultaneously estimate optical flow and detect occlusion within a single optimization framework. Our method does not iterate through flow estimation and occlusion detection. Compared to the previous state-of-the-art method [66], the proposed approach does not degrade the performance of optical flow estimation, indeed it does not require sensitive threshold parameter tuning.

Our method employs support-weight based window matching [16, 17, 22] with discrete MRF optimization. Each pixel is related to a node representing 3D vector, (i.e., 2D flow vector and occlusion status,) with well defined matching cost for every possible vector values. The window refers to local neighbourhoods of the pixel to be matched; and we fix it large enough to address the aperture phenomenon and other

robustness issues, such as random noise. The support-weight is for accentuating pixels in the window, if the pixels belong to the object that the central pixel belongs to. For example, the range difference is commonly employed to decide if two pixels belong to the homogeneous object:

$$w = \exp\left(-\frac{||I(s) - I(t)||^2}{2\sigma_p^2}\right),$$

where $||I(s) - I(t)||$ indicates range difference between a pixel t in the window and the central pixel s.

Denoting w^{ref} as the weight for a window in the reference and w^{tar} for a window in the target, we propose to define *normal weight* as $w^{ref}w^{tar}$, and *occlusion weight* as $w^{ref}(1-w^{tar})$. For the matching cost assuming the pixel is occluded, our method employs the occlusion weight, otherwise it uses the normal weight. Our observation shows these weights can be utilized as a constraint for occlusion detection and can improve the flow estimation for the occluded pixels.

Constraint for occlusion detection Figure 5.1 presents an example of the support-weights for an occluded pixel. The bright region represents a background object while the dark region represents a foreground object. The foreground in the reference (a) moves to left in the target (b); and the pixel indicated by red in (a) is occluded in (b). The second and third rows of Figure 5.1 illustrate the normal weight and the occlusion weight respectively, in case of the window in the reference is matched to background, occlusion and foreground regions in the target. The accentuated part of each window is shown in high intensity, which we refer as *effective area*.

When a window in the reference image is matched to occlusion, the effective area of the normal weight exactly represents occluded area, while that of the occlusion

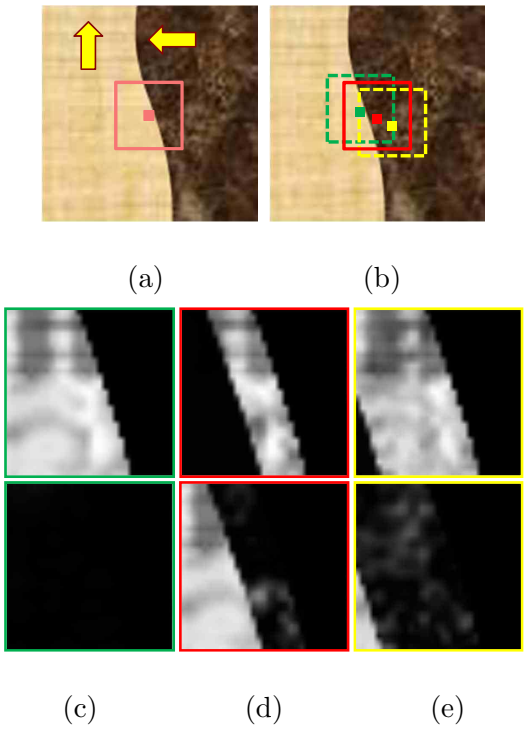


Figure 5.1: Support-weight for an occluded pixel. (a) The reference frame. The bright region (upper layer) moves to top, while the dark region (lower layer) moves to left. The middle pixel shown in pink is occluded in the target frame. (b) The target frame. Three target points (with windows) for the upper layer, occlusion, and the lower layer are shown in green, red and yellow, respectively. (c,d,e) Illustration of support-weights for the three target points computed with the normal weight (top) and with the occlusion weight (bottom.)

weight accentuate background object only. When matched to other regions, i.e., foreground or background, the effective area of the occlusion weight is almost none. Without loss of generality, the occlusion is supposed to be very sparse assuming the time difference between the target and the reference frame is infinitesimal; and we propose a simple (yet powerful) constraint for occlusion detection: if sum of the

normal weight is smaller than that of the occlusion weight, then it is very likely that the window is matched to occlusion, and we assign some penalty on the cost using the normal weight.

Matching cost for an occluded pixel When a pixel in the reference image is occluded, the pixel does not match to any pixel in the target, and its actual flow is undefined. Instead of defining the matching cost for the occluded pixel, previous approaches [18, 63, 66] generally assigned a constant cost for the occlusion. Not only the constant cost is hard to fix due to its sensitiveness, but the resulting flow estimation for the occluded pixel fully depends on regularization by neighbouring flows. In contrast, our method assigns reasonable cost for the occluded pixel. We observe that the matching cost for the occluded pixel may be defined using the non-occluded neighbors which are in the homogeneous object that the occluded pixel belongs to. As shown in Figure 5.1, the effective area of the occlusion weight exactly represents non-occluded *background* area in the window, thus it can be a good clue for estimating undefined flow for the occluded pixel.

The rest of this paper is organized as follows. Section 2 briefly defines our energy formulation for the discrete framework. In Section 3 we propose the new support-weight design and show its advantages. Section 4 introduces a method to enhance the efficiency of the discrete optimization, and Section 5 presents experimental results evaluating the proposed model. We finalize this work by providing the conclusion and the future work in Section 6.

5.2 Background

Let $\mathcal{G}$ be an undirected graph with a node set $\mathcal{V}$ and an edge set $\mathcal{E}$. A node in $\mathcal{V}$ corresponds to a pixel in the reference image. Let l_s be a label, i.e., a random variable for a node s in some discrete sample space $\mathcal{L}_s = \{1, \ldots, 2L^2\}$, representing the quantized vector set $\mathcal{T}_s = \{\mathbf{u}_s(1), \ldots, \mathbf{u}_s(2L^2)\}$. A vector in $\mathcal{T}_s$ is three dimensional, i.e., $\mathbf{u}_s = (u_s, v_s, o_s)$. First two dimensions represent displacement vector for x and y directions; and are homogeneously quantized by L labels. The last dimension, denoted by $o_s \in \{0, 1\}$ indicates occlusion status of the node. The flow estimation and occlusion detection problem can be expressed as finding the labels for each pixel, which minimizes an energy function such that:

$$\sum_{s \in \mathcal{V}} \Phi_s(l_s) + \sum_{(s,t) \in \mathcal{E}} \Psi_{st}(\mathbf{u}_s(l_s) - \mathbf{u}_t(l_t)) \,, \tag{5.1}$$

where $\Phi_s(\cdot)$ imposes the cost for matching the correlation window for s, and $\Psi_{st}(\cdot)$ denotes the spatial smoothness term between s and t.

The discrete sample space $\mathcal{L}$ is a finite set. The size of the space $|\mathcal{L}|$ $(= 2L^2)$ is proportional to the maximum displacement over the desired flow precision μ, such that, $|\mathcal{L}| \propto \max(\mathcal{T})/\mu$. To cover large displacement with limited number of labels, we build Gaussian image pyramids for the input images, and find the rough solution from the top level of the pyramids. Down to the next level, the dense flow field is estimated by interpolating the coarse solution, and is provided as the initial flow field for further estimation. The number of pyramid level is determined by $\log_d(\max(\mathcal{T})/|\mathcal{L}|)$ where d^{-1} is the downsampling factor building the image pyramid. We use $d = 2$ in our experiments.

5.3 Proposed Data Matching

The conventional support-weight based window matching cost can be defined as follows:

$$\Phi_s(l_s) = \frac{\sum_{t \in W(s)} w_s^{ref}(t) w_{s'}^{tar}(t') \rho(t, t')}{\sum_{t \in W(s)} w_s^{ref}(t) w_{s'}^{tar}(t')}, \tag{5.2}$$

where $W(s)$ is a neighboring node set in the window supporting s. w_s^{ref} means the support-weight function for s in the reference, and $w_{s'}^{tar}$ indicates the function for s' in the target. s and t are mapped to s' and t' by displacement vector of $\mathbf{u}_s(l_s)$. $\rho(t, t')$ denotes a similarity measure between pixels at t and t', e.g., the absolute difference, the squared difference, or the gradient inner product.

We propose to use different support-weight according to the occlusion status. The modified definition of the matching cost is shown as follows:

$$\Phi_s(l_s) = (1 - o_s) \frac{\sum_t w_s^{ref}(t) \rho(t, t')}{Z} + o_s \left(\frac{\sum_t w_s^{ref}(t)(1 - w_{s'}^{tar}(t')) \rho(t, t')}{Z_o} + \hat{\beta} \right), \tag{5.3}$$

where $Z = \sum_t w_s^{ref}(t)$ and $Z_o = \sum_t w_s^{ref}(t)(1 - w_{s'}^{tar}(t'))$. $\hat{\beta}$ is a conditional parameter that implements the constraint for sparse occlusion detection, such as:

$$\hat{\beta} = \begin{cases} 0 & \text{if } Z > Z_o \\ \beta & \text{otherwise} \end{cases} \tag{5.4}$$

Defining the support-weight for the reference image, we employ the conventional bilateral filtering based approach [22], shown as follows:

$$w_s^{ref}(t) = \exp\left(-\frac{||\mathbf{x}_s - \mathbf{x}_t||^2}{2\sigma_g^2} - \frac{||I(s) - I(t)||^2}{2\sigma_r^2} \right), \tag{5.5}$$

where $\mathbf{x}_s$, $\mathbf{x}_t$ indicate 2D coordinates, and $I(s)$, $I(t)$ mean color values of the points s and t, respectively.

For w^{tar}, we modified the geometric constraint in Eq. (5.5) such that,

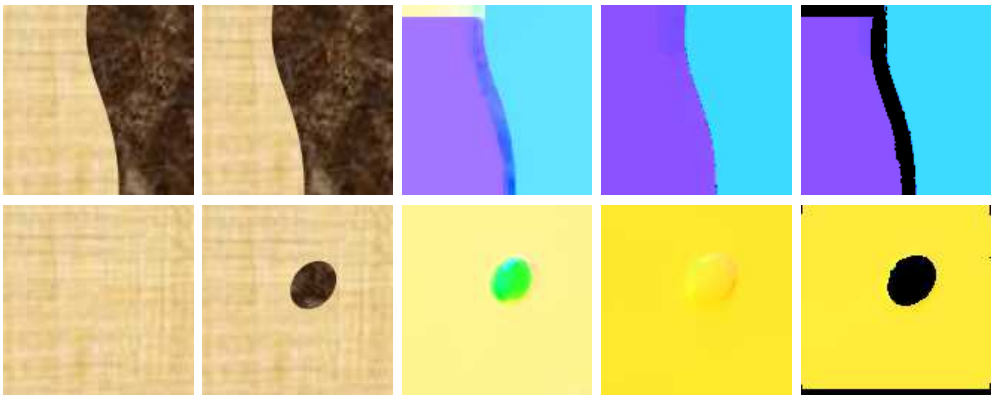
$$w_s^{tar}(t) = \exp\left(-\frac{(\mathbf{x}_s - \mathbf{x}_t)^{\mathsf{T}} \mathbf{T}_g(s) (\mathbf{x}_s - \mathbf{x}_t)}{2\sigma_g^2} - \frac{||I(s) - I(t)||^2}{2\sigma_t^2}\right). \tag{5.6}$$

$\mathbf{T}_g(s)$ is an anisotropic tensor produced by the structure tensor (i.e., second moment matrix) of the supporting window $W(s)$, defined as follows:

$$\mathbf{T}_g(s) = \frac{1}{D_g}\left(\frac{1}{|W(s)|}\sum_{k\in W(s)} \nabla I_g(k) \nabla I_g(k)^T + \nu^2 \mathbf{1}\right), \tag{5.7}$$

where $\mathbf{1}$ means the 2×2 identity matrix, and ∇I_g means the gradient of the gray-scaled input image. D_g represents a denominator for normalization, defined as the trace of the matrix in the parenthesis. ν is the parameter controlling the degree of isotropy. The eigenvectors of the structure tensor indicate the predominant directions of the gradient in the window. Thus, use of this tensor is more suitable to preserves non-occluded region along the motion boundary in the window, compared to the geometric constraint in Eq. (5.5) with fixed distribution regardless of image contents.

In Figure 5.2, we present estimation results for some example problems. We compare the proposed weight in Eq. (5.2) to the conventional weight in Eq. (5.3). As seen, the proposed approach not only detect very accurate occlusion but also find plausible flow estimation even for the occluded region.



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

Figure 5.2: Estimation results for toy problems. (a) Reference frames. (b) Target frames. (c) Flow estimation with the conventional weight. (d,e) Flow estimation and occlusion detection with the proposed weight

5.3.1 Coarse-to-fine occlusion update

Our method assumes sparse occlusion in subsequent frames to detect occlusion by comparing the normal weight to the occlusion weight. However, the input images occasionally contain large occlusion, and as shown in the middle row of Figure 5.3, the proposed method may yield a poor result in middle of the large occlusion.

As we build Gaussian image pyramids for the input images, the occluded region is also scaled down in the upper level of pyramids; and can be considered as *sparse*. Down to the next level, we interpolate the found occlusion and apply rank filtering. We use rank 4 for 5×5 filtering mask. The resulting occlusion map is combined to a new occlusion map generated in the current level.

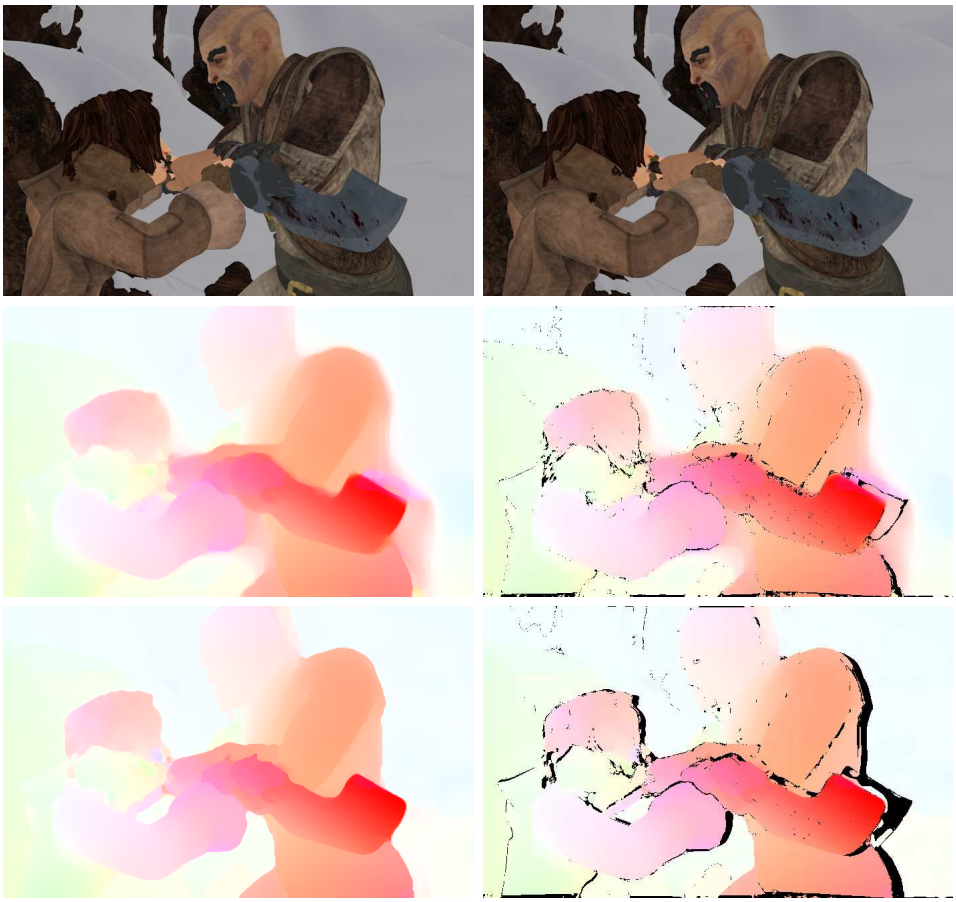


Figure 5.3: Coarse-to-fine occlusion update. **Top row:** The Ambush 5 sequence. **Middle row:** Estimation result without the update. **Bottom row:** Estimation result with the update. Detected occlusion is shown in black.

5.4 Optimization

To find the optimal solution for the MRF formulation in Eq. (5.1), we employ the TRW-S [11], which has shown state-of-the-art results [27] in many discrete framework applications. The asymptotic computational complexity of the TRW-S, in general, is $O(|\mathcal{V}||\mathcal{L}|^2)$. In our current framework, we may rewrite it as $O(|\mathcal{V}|L^4)$. Since

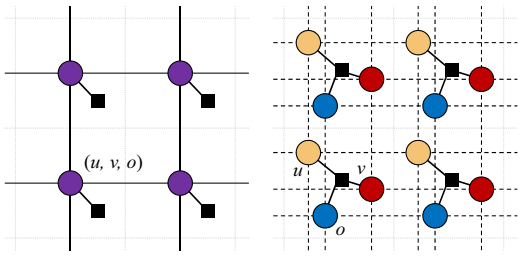


Figure 5.4: Conceptual illustration for the node decomposition. **Left:** The original MRF model. A node represents a label for 3D displacement vector: (u,v, o). **Right:** The original node is decomposed into three nodes representing labels for 1D vectors: u, v, and o respectively. The unary term (shown in a black square) in the original MRF model becomes a high-order potential term between the decomposed nodes.

the complexity is dominated by the number of labels, and our method requires an adequate number of labels to yield plausible estimation results, we introduce techniques to address the complexity issue.

5.4.1 Node decomposition

We apply the node decomposition scheme [28,67], reducing the complexity to $O(|\mathcal{V}|L^2)$. The scheme decomposes the node $s \in \mathcal{V}$ into three nodes $x \in \mathcal{V}_x$, $y \in \mathcal{V}_y$, and $o \in \mathcal{V}_o$. We may define l_i as a random variable for a node i in some discrete sample space $\mathcal{L}_i = \{1, \ldots, L\}$, representing the quantized 1D displacement vector set $\mathcal{T}_i = \{u_i(1), \ldots, u_i(L)\}$ where $i \in \{x, y\}$; and l_o as a random variable in $\mathcal{L}_o = \{0, 1\}$. The original displacement vector $\mathbf{u}_s(l_s)$ corresponds to $(u_x(l_x), u_y(l_y), l_o)$. The original edge set $\mathcal{E}$ is decomposed into $\mathcal{E}_x$, $\mathcal{E}_y$, $\mathcal{E}_o$, and the new hyper-edge set $\mathcal{E}_{xyo}$ is introduced, to account for the high-order potential between the decomposed nodes. Figure 5.4 shows a conceptual illustration of the decomposition scheme. The original

MRF formulation in (5.1) is updated as follows:

$$\sum_{(x,y,o)\in\mathcal{E}^{xyo}} \Psi_{xyo}(l_x, l_y, l_o) \; + \sum_{(i,i')\in\mathcal{E}^{i},i\in\{x,y,o\}} \Psi_{ii'}(u_i(l_i) - u_{i'}(l_{i'})). \tag{5.8}$$

We note the original unary potential Φ_s is updated to the factor node, an element of a hyper edge set $\mathcal{E}^{\mathcal{F}}$, represented by the ternary potential Ψ_{xyo}. Unary potentials for these nodes are undefined, imposing no cost on any configuration. As the number of labels for a node reduces to L, the complexity of message-passing for pairwise interactions reduces to $O(|\mathcal{V}|L^2)$.

Conversion from high-order potential to pairwise interactions The factor node induced by the decomposition is not easy to control in message-passing in the TRW-S algorithm. We apply the conversion proposed in [68, 69] and introduce its efficient implementation. Let a be an auxiliary variable node. a is associated to a new label $z \in \mathcal{A}$, where $\mathcal{A}$ is the Cartesian product of label spaces of connected three nodes, i.e., $\mathcal{A} = \mathcal{L}_x \times \mathcal{L}_y \times \mathcal{L}_o$. We replace the factor node with the auxiliary node a. Unary and pairwise potentials for the converted energy model are described as follows:

$$\Phi_a(z) = \Psi_{xyo}(l_x, l_y, l_o) \; , \tag{5.9}$$

$$\Psi_{ai}(z, l_i) = \begin{cases} 0 & \text{if } z_i = l_i \\ \infty & \text{otherwise} \end{cases} \quad \forall i \in \{s, t, u\} \; . \tag{5.10}$$

where any possible value z has one-to-one correspondence to the triplets (l_x, l_y, l_o). z_i $(i \in \{x, y, o\})$ denotes the associated component of the triplet. The pairwise potential Ψ_{ai} enforces z_i to be consistent with the labels of the neighboring decomposed nodes. After the conversion, energy model (5.8) changes to following,

$$\sum_{z\in\mathcal{A}} \Phi_a(z) \; + \sum_{(a,i)\in\mathcal{E}_{\mathcal{F}}} \Psi_{ai}(z, l_i) + \sum_{(i,i')\in\mathcal{E}^i, i\in\{x,y,o\}} \Psi_{ii'}(u_i(l_i) - u_{i'}(l_{i'})). \tag{5.11}$$

The auxiliary node may induce $O(L^3)$ of time complexity as well as $O(L^2)$ of memory space to store messages. However, a message a to $i \in \{x, y, o\}$ does not require any storage since the pairwise potential Psi_{ai} never adds a value on the message; in addition we may ignore all updating operations if $z_i \neq l_i$. In sum, by just modifying message-update logics in the implementation, the time complexity remains $O(L^2)$.

5.4.2 Regularization for occlusion

The pairwise potential for $\mathcal{E}^o$ can impose regularization for occlusion status between adjacent pixels. We have applied the Potts model with various λ values, and found the best results with $\lambda = 0$, i.e., no regularization.

5.4.3 Min convolution

The decomposition enables defining the pairwise potential Ψ_{st} as linear to the label difference; that is, we may rewrite $\Psi_{st}(u_s(l_s) - u_t(l_t))$ as $\Psi'_{st}(l_s - l_t)$. Then we can apply the min-convolution algorithm [29] for the TRW-S, reducing the time complexity to $O(|\mathcal{V}|L)$. In experiments, we set $\Psi'_{st}(l_s - l_t) = \alpha|l_s - l_t|$, which is a parametric and robust convex penalizer.

5.4.4 Incremental flow update

To obtain very high precision of sub-pixel accuracy, we iteratively find the incremental flow based on the current flow field. For the i_{th} iteration, the flow precision is

set to $\mu^{(i)} = 0.5\mu^{(i-1)}$, so that the discrete algorithm employs smaller quantization unit for the incremental flow. We note, in practice, the flow accuracy at a higher level pyramid strongly influences the performance of the next level pyramid; thus we run this process to obtain sufficiently high precision at every pyramid level.

5.5 Experiments

We validate our flow estimation and occlusion detection method on various datasets, e.g., the Sintel dataset [70, 71] and the Middlebury flow dataset [30, 72]. The Sintel dataset contains several image sequences generated by movements of synthetic objects, thus providing the exact ground-truth optical flow and occlusion map. The Middlebury dataset includes image sequences of indoor and outdoor scenes, containing various real or synthetic objects. To assess the accuracy of estimated flow, we compare average end-point error (AEPE) and average angular error (AAE); and for the performance of occlusion detection, we calculate the F_1 scores using the ground-truth occlusion maps.

We assumed the maximum deformation for each direction to be 64, and quantized each direction by 8 with the target precision $\mu = 0.05$. The size of correlation windows was fixed to 30×30. The parameters affecting the relative influence and strictness of the different constraints are fixed to optimal values for other experiments: $\sigma_g = 7.2$, $\sigma_r = 3.8$, $\sigma_d = 3.8$, and $\alpha = 0.05$.

To show the effect of occlusion detection, Figure 5.5 presents qualitative analysis comparing estimation obtained with the proposed algorithm to the result computed with our algorithm without occlusion detection, referred as *ours w/o occ.* We simply impose very large penalty on the matching cost using the occlusion weight. As seen,

Figure 5.5: Estimation results for the Alley 1 sequence. **Top left:** The reference frame. **Top right:** Flow estimation with our method, which does not use occlusion detection. **Bottom left/right:** Flow estimation with our method. Detected occlusion is shown in black in the left image.

flow estimation on homogeneous regions is not clearly different. However, on the region with occlusion, (as shown in black in the left-bottom image, e.g., upper region of the left arm and the fruit,) the proposed method obviously improves the quality of flow estimation.

Figure 5.6 additionally presents illustrative comparison of our algorithm to various related methods. We employed the source codes provided in their website. The method of Xu et al. [2] (the bottom row of the first column,) one of top-performing methods in the Middlebury flow site [30], demonstrates state-of-the-art estimation overall, (AEPE=0.44,) but the lack of occlusion detection causes degenerated es-

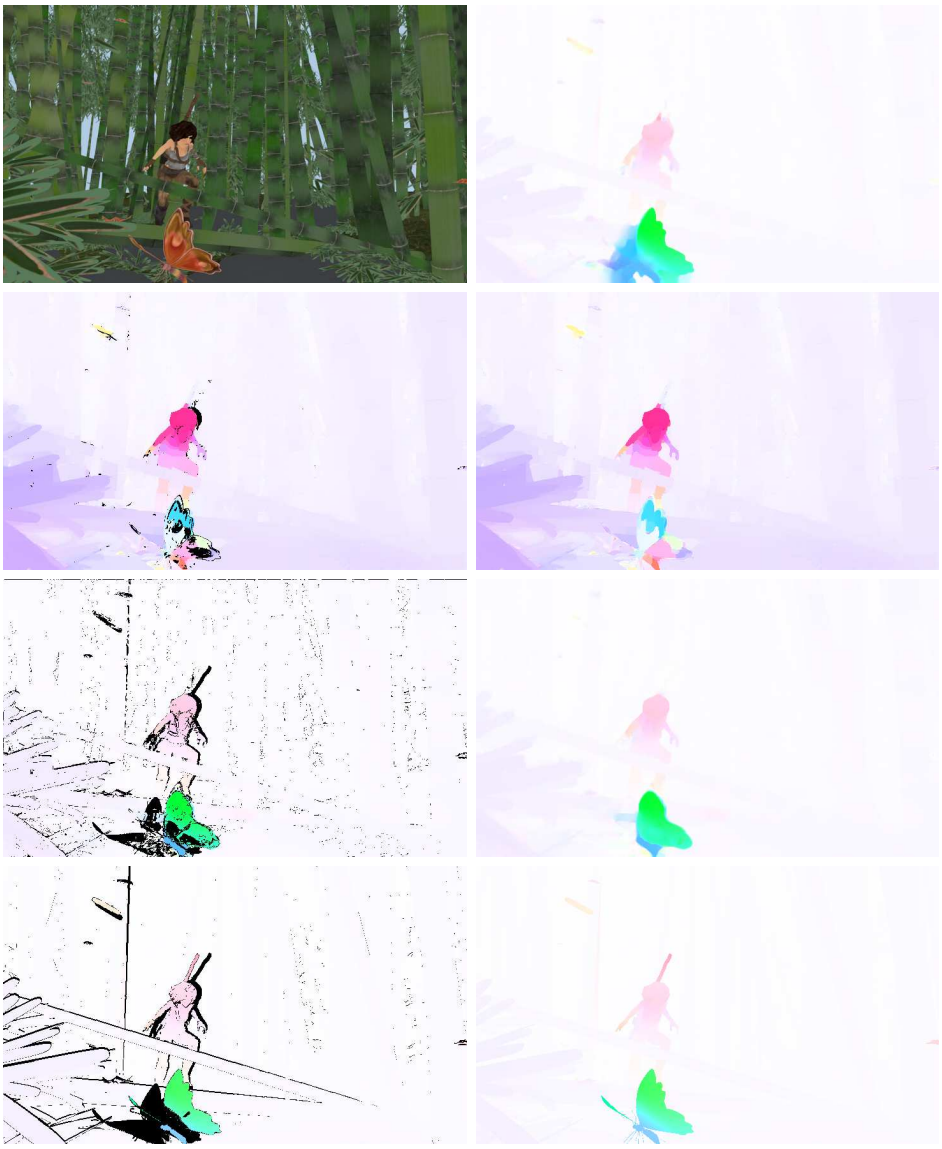


Figure 5.6: Estimation results for the Bamboo 2 sequence using various algorithms. **Row 1:** The referenc image and estimation with Xu et al. **Row 2:** Estimation with Ayvaci et al. **Row 3:** Estimation with ours **Row 4:** Ground-truth flow. Detected (or ground-truth) occlusion is shown in black in the top row.

Table 5.1: Flow estimation error.

	alley1	alley2	ambush5	bamboo1	bamboo2
Ayvaci et al.	0.36/3.09	0.47/5.63	1.37/28.26	0.34/5.46	0.67/7.39
Xu et al.	0.23/2.44	**0.16**/2.28	0.70/9.37	0.27/4.11	0.44/5.63
Ours w/o occ.	0.25/3.01	0.17/2.34	0.56/8.32	0.25/**3.94**	0.33/6.48
Ours	**0.16**/**1.75**	**0.16**/**1.74**	**0.44**/**7.17**	**0.24**/4.00	**0.30**/**5.42**

bandage1	bandage2	market2	temple2	temple3	**average**
1.31/8.59	0.63/7.01	1.44/10.03	1.64/8.54	**0.51**/2.17	0.87/8.62
0.48/3.57	0.34/3.87	0.96/6.27	0.78/4.06	0.57/2.11	0.49/4.37
0.53/4.24	0.32/4.07	1.07/7.66	0.75/4.27	0.65/2.15	0.49/4.65
0.49/**3.45**	**0.26**/**3.68**	**0.81**/**6.05**	**0.65**/**3.68**	0.57/**1.90**	**0.41**/**3.88**

Table 5.2: Occlusion detection evaluation.

	alley1	alley2	ambush5	bamboo1	bamboo2
Ayvaci et al.	**.94**/.17/.28	**.84**/.23/.36	**.97**/.34/.50	**.62**/.02/.04	.33/.05/.09
Ours	.77/**.49**/**.60**	.35/**.63**/**.45**	.57/**.56**/**.57**	.49/**.40**/**.44**	**.58**/**.50**/**.54**

bandage1	bandage2	market2	temple2	temple3	**average**
.45/.11/.17	.65/.08/.15	.58/**.33**/**.42**	.48/.06/.11	**.93**/.45/**.61**	**.68**/.18/.27
.69/**.52**/**.59**	**.73**/**.52**/**.61**	**.59**/.31/.41	**.60**/**.43**/**.50**	.49/**.47**/.48	.59/**.48**/**.52**

timation around the region with large occlusion. (e.g., the right wing of the butterfly.) Estimation with Ayvaci et al. [66] finds very delicate motion boundaries, but both of flow estimation and occlusion detection are also severely deteriorated (AEPE=0.67, F1=0.09), particularly on the region with large displacement. In contrast, our method presents plausible flow estimation (AEPE=0.29) and occlusion detection (F1=0.54) even on the problematic region.

In Table 5.1, we show quantitative analysis comparing AEPE/AAE with these methods to estimation errors with ours, for several sequences in the Sintel dataset.

The reference frame is the tenth frame for each sequence. We excluded the results from the sequences containing too large displacement, that no algorithm found meaningful estimation with AEPE less than 10. For reference, we also add estimation results with *ours w/o occ.* Compared to the method of Xu et al., ours shows lower AEPE in average, probably due to better estimation on the occluded region, as *ours w/o occ.* also shows similar performance with the method of Xu et al. Table 5.2 also provides analysis comparing precision/recall/F1-score measuring occlusion detection performance with the method of Ayvaci., the state-of-the-art simultaneously estimating flow and detect occlusion, to detection performance of ours. While the method of Ayvaci et al. shows better precision in some sequences, but the proposed method outperforms Ayvaci et al. for recall, and yields higher F1-score for most of sequences. We note the method of Ayvaci degenerates the accuracy of flow estimation for occlusion detection while our method even improve the accuracy, as shown in table 5.1.

5.6 Discussion

In this work, we presented a novel support-weight based window matching method for simultaneously estimate optical flow and detect occlusion. Our method works on a unified optimization framework, which does not require any explicit estimation of flow for occlusion detection, nor additional computation for occlusion. The proposed support-weight provides an effective clue to detect occlusion, and improve estimation of flow in occluded area. Experiments showed our method yields highly competitive results, for optical flow estimation as well as occlusion detection. Compared to the previous state-of-the-art method, our method does not degrade performance of

optical flow to enhance detection.

We currently assume foreground and background objects are distinguishable by their color, and so our algorithm may not present plausible estimation and detection, for the region that assumption is not valid. More effective approach to distinguish those objects can much improve the results in the future. Also, we plan to use multi-labels for occlusion to specify the class of occlusion, e.g., order of motion layer, rotation, viewpoint change, or severe illumination change.

Our current implementation takes 2012.4 seconds, on average, to find the estimation of a 640×480 image, with a 30×30 correlation window. Graphic hardware is already employed for parallel computing of the data matching cost, but we believe significantly faster processing can be obtained with a full implementation that computes message-passing based optimization on parallel graphics hardware [31].

Chapter 6

Conclusion

In this work, we proposed novel methods that address several current issues in optical flow estimation.

To reduce errors around motion boundaries, we presented a new adaptive window correlation based on the discrete MRF framework. A novel data cost design incorporating various constraints efficiently ignores inhomogeneous motion in correlation windows on object boundaries, helping to enlarge the window size to cover the aperture phenomenon. The effect of each constraint compared to the previous constraints has been shown with quantitative analysis. In order to reduce computational complexity and fully utilize image resolution, we utilized the decomposed scheme combined with the course-to-fine approach.

Addressing complex non-transitional motion with large displacement, we propose a discrete analog to the historically well-developed variational methods and benefit from both approaches. The data cost in our model does not require linearization to be differentiable; hence, inherently covers large displacement problems. The convolution prior model is shown to be more powerful and flexible than the simple

neighborhood system with the support window, which has been frequently used in the literature of the discrete framework. The new adaptive kernel generalize the bilateral kernel and presents competitive results on motion boundaries as well as textured motion segments without cumbersome parameter tuning.

For the occlusion issue, we presented a novel support-weight based window matching method for simultaneously estimate optical flow and detect occlusion. Our method works on a unified optimization framework, which does not require any explicit estimation of flow for occlusion detection, nor additional computation for occlusion. The proposed support-weight provides an effective clue to detect occlusion, and improve estimation of flow in occluded area. Experiments showed our method yields highly competitive results, for optical flow estimation as well as occlusion detection. Compared to the previous state-of-the-art method, our method does not degrade performance of optical flow to enhance detection.

Bibliography

[1] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2432–2439.

[2] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 9, pp. 1744–1757, 2012.

[3] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," in *CVPR*, 2009.

[4] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[5] M. J. Black, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.

[6] J. Weickert and C. Schnörr, "A theoretical framework for convex regularizers in pde-based computation of image motion," *International Journal of Computer Vision*, vol. 45, no. 3, pp. 245–264, 2001.

[7] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[8] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, "Anisotropic huber-l1 optical flow," in *Proceedings of the British machine vision conference*, vol. 34, 2009, pp. 1–11.

[9] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, "Highly accurate optic flow computation with theoretically justified warping," *Int. J. Computer Vision*, vol. 67, no. 2, pp. 141–158, 2006.

[10] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.

[11] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 10, pp. 1568–1583, 2006.

[12] N. Komodakis, G. Tziritas, and N. Paragios, "Fast, approximately optimal solutions for single and dynamic mrfs," in *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*. IEEE, 2007, pp. 1–8.

[13] V. Lempitsky, C. Rother, S. Roth, and A. Blake, "Fusion moves for markov random field optimization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 8, pp. 1392–1405, 2010.

[14] B. Glocker, N. Paragios, N. Komodakis, G. Tziritas, and N. Navab, "Optical flow estimation with uncertainties through dynamic mrfs," in *Computer Vision*

and Pattern Recognition (CVPR), 2008 IEEE Conference on. IEEE, 2008, pp. 1–8.

[15] C. Lei and Y.-H. Yang, "Optical flow estimation on coarse-to-fine region-trees using discrete optimization," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1562–1569.

[16] K. J. Lee, D. Kwon, I. D. Yun, and S. U. Lee, "Optical flow estimation with adaptive convolution kernel prior on discrete framework," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2504–2511.

[17] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3017–3024.

[18] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi, "Bilateral filtering-based optical flow estimation with occlusion detection," *Computer Vision–ECCV 2006*, pp. 211–224, 2006.

[19] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for tv-l1 optical flow," *Statistical and Geometrical Approaches to Visual Motion Analysis*, pp. 23–45, 2009.

[20] T. Cooke, "Two applications of graph-cuts to image processing," in *Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2008, pp. 498–504.

[21] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981.

[22] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 4, pp. 650–656, 2006.

[23] Y. S. Heo, K. M. Lee, and S. U. Lee, "Robust stereo matching using adaptive normalized cross-correlation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 4, pp. 807–822, 2011.

[24] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 9, pp. 920–932, 1994.

[25] A. Fusiello, V. Roberto, and E. Trucco, "Efficient stereo with multiple windowing," in *Computer Vision and Pattern Recognition (CVPR), 1997 IEEE Conference on*. IEEE, 1997, pp. 858–863.

[26] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," *Computer Vision-ECCV 2004*, pp. 25–36, 2004.

[27] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 6, pp. 1068–1080, 2008.

[28] A. Shekhovtsov, I. Kovtun, and V. Hlaváč, "Efficient mrf deformation model for non-rigid image matching," *Computer Vision and Image Understanding*, vol. 112, no. 1, pp. 91–99, 2008.

[29] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 41–54, 2006.

[30] "Middlebury flow website." [Online]. Available: http://vision.middlebury.edu/flow

[31] C.-K. Liang, C.-C. Cheng, Y.-C. Lai, L.-G. Chen, and H. H. Chen, "Hardware-efficient belief propagation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 5, pp. 525–537, 2011.

[32] L. Alvarez, J. Esclarin, M. Lefébure, and J. Sánchez, "A pde model for computing the optical flow," in *Proc. XVI Congreso de Ecuaciones Diferenciales Aplicaciones*, 1999.

[33] H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 565–593, 1986.

[34] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *Image Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 421–432, 1998.

[35] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, "Anisotropic huber-l1 optical flow," in *BMVC*, 2009.

[36] C. Schnörr, "Segmentation of visual motion by minimizaing convex non-quadratic functionals," in *ICPR*, 1994.

[37] J. Weickert and C. Schnörr, "A theoretical framework for convex regularizers in pde-based computation of image motion," *Int. J. Computer Vision*, vol. 45, no. 3, pp. 245–264, 2001.

[38] S. Roth and M. J. Black, "Steerable random fields," in *CVPR*, 2007.

[39] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, and H. P. Seidel, "Complementary optic flow," in *EMMCVPR*, 2009.

[40] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi, "Bilateral filtering-based optical flow estimation with occlusion detection," in *Proc. 9th European Conference on Computer Vision*, 2006.

[41] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with pde's : A common framework for different applications," in *CVPR*, 2003.

[42] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, 1998.

[43] M. J. Black and P. Anandan, "Robust dynamic motion estimation over time," in *CVPR*, 1991.

[44] A. Wedel, D. Cremers, T. Pock, and H. Bischof, "Structure- and Motion-adaptive Regularization for High Accuracy Optic Flow," in *Proc. IEEE Int'l Conf. on Computer Vision*, 2009.

[45] D. Sun, S. Roth, J. P. Lewis, and M. J. Black, "Learning optical flow," in *ECCV*, 2008.

[46] S. Bougleux, A. Elmoataz, and M. Melkemi, "Local and nonlocal discrete regularization on weighted graphs for image and mesh processing," *Int. J. Computer Vision*, vol. 84, no. 8, pp. 220–236, 2009.

[47] K. J. Yoon and S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 4, p. 650, 2006.

[48] B. Glocker, N. Paragios, N. Komodakis, G. Tziritas, and N. Navab, "Optical flow estimation with uncertainties through dynamic mrfs," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[49] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Computer Vision*, vol. 70, no. 1, pp. 41–54, 2006.

[50] A. Shekhovtsov, I. Kovtun, and V. Hlavác, "Efficient MRF Deformation Model for Non-Rigid Image Matching," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[51] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, 2006.

[52] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields," in *Proc. 9th European Conference on Computer Vision*, 2006.

[53] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.

[54] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Proc. IEEE Int'l Conf. on Computer Vision*, 2007.

[55] O. Faugeras, B. Hotz, H. Mathieu, T. Viéville, Z. Zhang, P. Fua, E. Théron, L. Molli, G. Berry, J. Vuillemin, P. Bertin, and C. Proy, "Real time correlation based stereo: algorithm implementation and applicaton," in *INRIA Technical Report*, 1993.

[56] C. K. Liang, C. C. Cheng, Y. C. Lai, L. G. Chen, and H. H. Chen, "Hardware-efficient belief propagation," in *Proc. IEEE Int'l Conf. on Computer Vision*, 2009.

[57] D. Mahajan, F.-C. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur, "Moving gradients: a path-based method for plausible image interpolation," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, p. 42, 2009.

[58] J. Xiao and M. Shah, "Motion layer extraction in the presence of occlusion using graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1644–1659, 2005.

[59] C. Strecha, R. Fransens, and L. Van Gool, "A probabilistic approach to large displacement optical flow and occlusion detection," in *Statistical methods in video processing*. Springer, 2004, pp. 71–82.

[60] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez, "Symmetrical dense optical flow estimation with occlusions detection," *International Journal of Computer Vision*, vol. 75, no. 3, pp. 371–385, 2007.

[61] C. L. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 7, pp. 675–684, 2000.

[62] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 508–515.

[63] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum, "Symmetric stereo matching for occlusion handling," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 399–406.

[64] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 7, pp. 787–800, 2003.

[65] C. Ballester, L. Garrido, V. Lazcano, and V. Caselles, "A tv-l1 optical flow method with occlusion detection," in *Pattern Recognition*. Springer, 2012, pp. 31–40.

[66] A. Ayvaci, M. Raptis, and S. Soatto, "Sparse occlusion detection with optical flow," *International journal of computer vision*, vol. 97, no. 3, pp. 322–338, 2012.

[67] K. J. Lee, D. Kwon, I. D. Yun, and S. U. Lee, "Deformable 3d volume registration using efficient mrfs model with decomposed nodes," in *British Machine Vision Conference*, 2008, pp. 1–10.

[68] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 736–744, 2001.

[69] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Map estimation via agreement on trees: message-passing and linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 11, pp. 3697–3717, 2005.

[70] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 611–625.

[71] "Sintel database website." [Online]. Available: http://sintel.is.tue.mpg.de

[72] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.