From Pixels to Layers: Joint Motion Estimation and Segmentation

by
Deqing Sun
B. Eng., Harbin Institute of Technology, 2003
M. Phil., The Chinese University of Hong Kong, 2007
Sc. M., Brown University, 2009

A Dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2013

This dissertation by Deqing Sun is accepted in its present form
by the Department of Computer Science as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date ________________ ____

Michael J. Black, Director

Recommended to the Graduate Council

Date ________________ ____

Erik B. Sudderth, Reader

Date ________________ ____

Yair Weiss, Reader
(The Hebrew University of Jerusalem)

Approved by the Graduate Council

Date ________________ ____

Peter M. Weber
Dean of the Graduate School

Vitæ

Deqing Sun was born on 10 March 1981 in Lanxi, Heilongjiang province, China.

Education

- *Ph.D. in Computer Science*, Brown University, Providence, Rhode Island, USA, May 2013.
- *Sc.M. in Computer Science*, Brown University, Providence, Rhode Island, USA, May 2009.
- *M.Phil. in Electronic Engineering*, The Chinese University of Hong Kong, Hong Kong, China, July 2007.
- *B.Eng. Electronic and Information Engineering*, Harbin Institute of Technology, Harbin, China, July 2003.

Professional Experience

- Intern at Microsoft Research New England, Cambridge, Massachusetts, USA, fall 2010.
- Research assistant at the department of Electronic Engineering, the Chinese University of Hong Kong, Hong Kong, China, Sep. 2004 - Aug. 2005.

Refereed Journal Articles

- Deqing Sun and Wai-Kuen Cham. "Postprocessing of Low Bit Rate Block DCT Coded Images Based on a Fields of Experts Prior". IEEE Transactions on Image Processing (TIP), vol. 16(11), pp. 2743-2751, Nov. 2007.

Refereed Conference Papers

- Deqing Sun and Ce Liu. "Non-causal Temporal Prior for Video Deblocking". European Conference on Computer Vision (ECCV), 2012.
- Deqing Sun, Erik B. Sudderth, and Michael J. Black. "Layered Segmentation and Optical Flow Estimation over Time". IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
- Ce Liu and Deqing Sun. "A Bayesian Approach to Adaptive Video Super Resolution". IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
- Deqing Sun, Erik B. Sudderth, and Michael J. Black. "Layered Image Motion with Explicit Occlusions, Temporal Consistency, and Depth Ordering". Neural Information Processing Systems (NIPS), 2010.

- Deqing Sun, Stefan Roth, and Michael J. Black. "Secrets of Optical Flow Estimation and Their Principles". IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2010.
- Deqing Sun, Stefan Roth, J.P. Lewis, and Michael J. Black. "Learning Optical Flow". European Conference on Computer Vision (ECCV), 2008.
- Deqing Sun and Wai-Kuen Cham. "An Effective Postprocessing Method for Low Bit Rate Block DCT Coded Images". IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2007.
- Yifeng Jiang, Jun Xie, Deqing Sun, and Hung-Tat Tsui. "Shape Registration by Simultaneously Optimizing Representation and Transformation". International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), 2007.
- Deqing Sun, Wai-Kuen Cham, and Junhao Xie. "A New Postprocessing Algorithm of Low Bit Rate JPEG Coded Images". SPIE Conference on Visual Communication and Image Processing (VCIP), 2005.

Dedicated to my parents.

Acknowledgements

This dissertation would not have been possible without the guidance, help, and support of so many people.

First, I want to thank my advisor, Prof. Michael J. Black. Michael has made great efforts to guide me during my Ph.D. study, from teaching me how to approach a problem to advising me on how to give a presentation. He has been a terrific teacher, scholar, role model, and friend. His curiosity into the unknown and his love for new ideas have motivated me greatly in my research. His passion for research and his belief in my abilities have encouraged me to overcome many obstacles. His saying, "If you get the math wrong, you are a toast," always reminded me to pay attention to the math, an effective way to debug my programs. Much of what I have achieved in the past five years I owe to him. Thank you, Michael. It has been a great intellectual journey.

I am also grateful to my other thesis committee members, Prof. Erik B. Sudderth and Prof. Yair Weiss. I was Erik's TA for his first course at Brown and benefited significantly from his deep mathematical insight. During our collaboration, I have always been enlightened by his way of understanding complicated ideas from simple examples. Yair's pioneering work on the layered model has inspired our work. His critical questions before and during my thesis proposal made me think more about the fundamental problems at the core of our projects before going into the details.

Before coming to Brown, I already knew Prof. Stefan Roth from his "Fields-of-Experts" work with Michael. It was a splendid beginning for me as a new PhD student to collaborate with a mature academic colleague and friend. He taught me a tremendous amount on programming, math, writing, and giving presentations. I was particularly motivated by his own experience to be persistent in one's academic pursuit. Thanks for the sharing, Stefan.

I met Dr. Ce Liu at ECCV 2008 and liked his "SIFT Flow" work very much. I had a great time working with Ce as an intern at Microsoft Research New England. Ce is a great mentor and friend to me personally, treating me like a younger brother. His anecdotes have been inspiring me to be more self-critical and open-minded. I learned to think more about why a given algorithm works or doesn't, in particular theoretically. Thanks, Ce, for your teaching, support, and help.

I would also like to thank Prof. Wai-Kuen Cham for his patient guidance and kind help when I studied at the Chinese University of Hong Kong. From him I learned to be a "detective" in research and to enjoy the process of discovering new knowledge. Thanks to Prof. Junhao Xie from the Harbin Institute of Technology, for helping me cultivate a serious attitude toward research.

I have been fortunate to have very interesting academic conversations with Brown faculty, including Professors Tom Doeppner, James Hays, John Hughes (Spike), Sorin Istrail, Chad Jenkins, Philip Klein, David Laidlaw, John Savage, and Gabriel Taubin. Their critical questions and open attitudes have offered me important insights into their fields of expertise. Special thanks to James for taking over and supporting the computer vision reading group, where we learn by critiquing research papers from various angles.

A number of people have contributed to this work, either through discussions and suggestions, providing details on their work, or commenting on my papers and code. I would like to thank Simon Baker, Andrew Blake, Thomas Brox, Lo-Bin Chang, Jianshu Chen, Rob Fergus, David Fleet, Bill Freeman, Geoffrey Hinton, Berthold K.P. Horn, Wei Hu, Erik Learned-Miller, J.P. Lewis, Dahua Lin, Jitendra Malik, Andrew Owens, Xiaofeng Ren, Michael Rubinstein, Stan Sclaroff, Josef Sivic, Richard Szeliski, Antonio Torralba, Lorenzo Torresani, Andreas Wedel, Donglai Wei, Manuel Werlberger, Jianxiong Xiao, Jiejie Zhu, and Andrew Zisserman. I would like to thank Daniel Scharstein for maintaining the online optical flow benchmark.

I would like to thank the members of Brown SCAPE and LIV groups, including Alexandru Balan, Oren Fredfeld, Soumya Ghosh, Peng Guan, David Abraham Hirshberg, Michael Hughes, Dae Il Kim, Matthew Loper, Teodor Mihai Moldovan, Jason Pacheco, Laura Sevilla-Lara, Leonid Sigal, Aggeliki Tsoli, Alex Weiss, Payman Yadollahpour, and Silvia Zuffi. Thanks for your invaluable comments to my research, papers, presentations, and dissertation. I will definitely miss the light academic conversations we had at the Grad Center Bar (GCB).

I had a great spring in the pretty town of Tübingen. Thanks to members of the PS group at Max Planck Insitute for their help and support and the amusing lunch talks: Jon Anning, Eric Rachlin, Melanie Feldhofer, Juergen Gall, Peter Gehler, Soren Hauberg, Hueihan Jhuang, Martin Kiefel, Andreas Lehrmann, Naureen Mahmood, Jessica Purmort, Eric Rachlin, Javier Romero, Elena Tretyak, Dimitris Tzionas, and Jonas Wulff.

The graduate students make the computer science department at Brown an enjoyable place to study and work in. I feel grateful to these colleagues: Alexandra Sasha Berkoff, Jesse Butterfield, Lijuan Cai, Carleton Coffrin, Tingjian Ge, Steven Gomez, Hua Guo, Serdar Kadioglu, Suman Karumuri, Hideaki Kimura, Alptekin Küpçü, Jie Mao, Jadrian Miles, Greg Nicholas, Olya Ohrimenko, Genevieve Patterson, Andrew Pavlo, Eric Rachlin, Anna Ritz, Deepak Santhanam, Warren Schudy, Eric Sodomka, Libin Sun, Aggeliki Tsoli, Marek Vondrak, Dongbo Wang, Zhe Zhang, Zhenyuan Zhao, and Wenjin Zhou. Both the acedemic and technical staff have been very helpful. Thanks also to my friends, Daosheng Deng, Jie Dong, Jian Gong, Huan Lei, Dov Sax, Jean Keith, Xiang Li, Yifeng Jiang, Ali Osman Ulusoy, Wanchun Wei, Hui Zhao, and Yong Zhao.

I am forever grateful to my parents and my sister for their long-time encouragement and love. My gratitude to them is beyond words. Finally, thanks goes to my fiancée, Jie Ren, for her love and support.

Contents

List of Tables

List of Figures

Abstract of "From Pixels to Layers: Joint Motion Estimation and Segmentation"

by Deqing Sun, Brown University, May 2013

Estimating image motion, or optical flow, in scenes with multiple moving objects and segmenting the individual moving objects are two fundamental problems in computer vision and have applications in many fields, including medical imaging, image processing, graphics, and robotics. Motion estimation and scene segmentation are particularly challenging because of lighting changes, motion boundaries, occlusions, and indiscriminative appearances. Despite decades of extensive research effort, current methods still tend to produce large optical flow errors near motion boundaries and in occlusion regions and falsely merge foreground objects with the background.

A key feature of optical flow methods is an energy term, or prior, that prefers spatially smooth flow fields. In this dissertation, we show that image-dependent and non-local prior models can better preserve motion boundaries than the widely used pairwise Markov Random Field (MRF) models. We also demonstrate that joint motion estimation and segmentation can achieve more accurate results than the separate treatment of each problem.

First, we formulate fully learnable low-level models of optical flow and learn the models from training data. Our results show that image-dependent, steerable models outperform standard MRF models, especially in recovering motion boundaries. Second, to understand what makes optical flow accurate, we perform a quantitative analysis of recent practices in optical flow estimation. Median filtering of the flow field is one of the key features of the most accurate methods and we formalize this as a non-local smoothness term that integrates information over a large spatial neighborhood. We further define a weighted non-local smoothness term that uses both image and motion cues to preserve motion boundaries. Third, inspired by recent successes in static image segmentation we develop a layered model to segment moving objects (layers) using image-dependent, continuous support functions. The method orders each layer in depth and explicitly models the occlusions between layers and the temporal consistency of layers. In an attempt to avoid being trapped in poor local optima, we define a discrete formulation of our objective function and extend graph cuts optimization methods to obtain good initial values for the continuous formulation. The mixed continuous-discrete optimizer can automatically infer the number of layers and their depth ordering for a given scene. Experimental results on benchmark datasets demonstrate the benefits of joint motion estimation and segmentation: the layered approach achieves more accurate motion estimates in motion boundary and occlusion regions and better segments the foreground from the background when compared with solving each problem separately.

CHAPTER 1

Introduction

We live in a dynamic world and constantly perceive visual motion between us and our surroundings. The perceived motion provides a rich source of information for us to understand the world and make decisions. For example, humans use motion cues to extract the three-dimensional structure of their environment, locate themselves, and control walking towards a target object [**182**]. There is also evidence that motion plays a fundamental role in the early stage of visual learning [**122**].

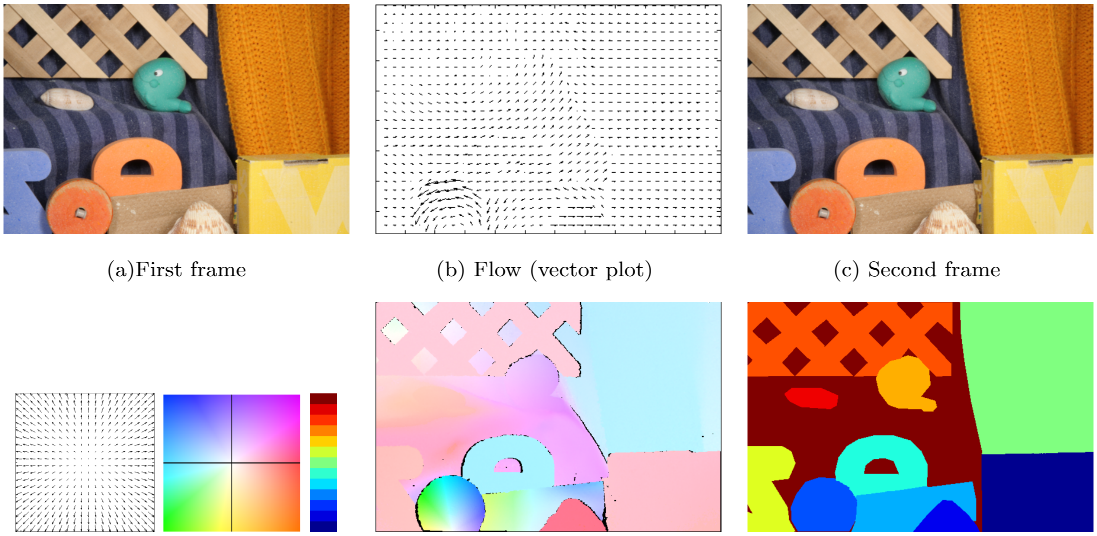
Computer vision aims at extracting useful descriptions from input images for understanding the visual world [**107**]. Significant progress has been achieved in static image analysis and understanding, such as the development of real-time face detection system [**173**]. Nevertheless, understanding the dynamic world is still challenging to current computer vision systems. One missing capability is to perceive and understand motion from image sequences. This dissertation aims to improve automatic estimation of pixel-level motion between related images of the same scene.

Motion estimation and segmentation. We now define the "motion estimation and segmentation" problem of interest. Given a sequence of images taken by a camera over scenes with multiple moving objects, we want to estimate the motion of each pixel between subsequent time steps (frames), segment the scene into different moving objects, and infer the depth ordering of the moving objects. The per-pixel motion field is usually called the optical flow field. Figure 1.1 shows an input image pair, the ground truth optical flow field, and the human labeled segmentation.

1.1. Motivations

Motion provides the temporal correspondence between neighboring frames and is a fundamental tool for video processing and analysis. We will use several examples in computer vision and graphics to explain applications of motion.

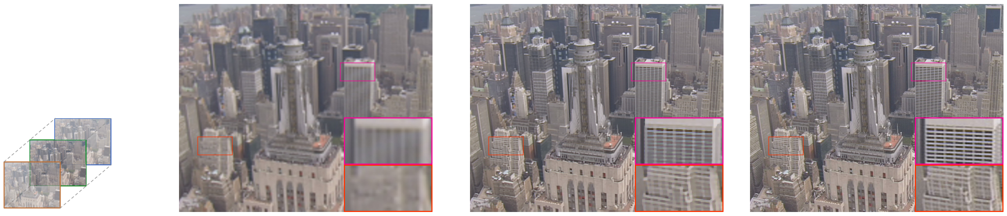
1.1.1. Video Super Resolution. High definition TVs (HDTVs) have greatly enhanced our visual experiences. However a significant amount of the video data have been captured in the standard definition format. Playing standard definition TV programs on the HDTVs does not necessarily result in pleasant visual experiences. Video super resolution aims at reconstructing a high resolution video sequence with more details than an input low resolution sequence. Every frame of the low resolution sequence provides a slightly different observation of the same scene. If we know where every pixel goes from one frame to another, we can accumulate the information over time to enhance the spatial details. To do so requires accurate and detailed alignment of the pixels (fine-level motion estimates). Using fine-level motion estimates for video super resolution was first proposed in the late 1990s [**15**], but the motion estimation techniques were not accurate enough. Only recently have reliable motion estimation techniques made such an approach promising [**103**] for scenes with slow and smooth motion fields, as shown in Figure 1.2.



(a)First frame (b) Flow (vector plot) (c) Second frame

(d) Color key (e) Flow (color) (f) Segmentation

Figure 1.1. The Middlebury "RubberWhale" sequence and its ground truth optical flow field and human segmentation. First row, left to right: (a) first image, (b) vector plot of the ground truth optical flow, and (c) second image. Second row, left to right: (d) color key for the flow field, color key for the depth ordering (blue is close and red is far), (e) color display of the ground truth optical flow (black means occlusion), and (f) the human segmentation.



(a) Input low-res (b) Bicubic up-sampling ×4 (c) Output from [**103**] (d) Original frame

Figure 1.2. Optical flow-based video super resolution system is able to recover image details after ×4 up-sampling for sequences with smooth flow fields [**103**] © 2011 IEEE.

1.1.2. Pedestrian Detection and Action Recognition. Motion is one of the most important low-level cues for visual grouping and provides useful information for many high-level tasks. Walk *et al.* [**176**] show that motion features yield significant performance improvement on pedestrian detection (the detection rate increases from about 30% to more than 40%). Wang *et al.* [**178**] find that the motion boundary cues [**45**] can suppress most camera motion in the background and highlight the foreground objects for action recognition tasks.

1.1.3. Content-aware Video Resizing. More and more videos are captured everyday using different devices, such as smart phones and high definition cameras. To play video sequences in different display devices, we need to change (usually reduce) the size of the video. The most popular interpolation methods treat every part of the images equally and are not effective at preserving the important content of the image [**10**]. Instead we can resize the images according to their content.

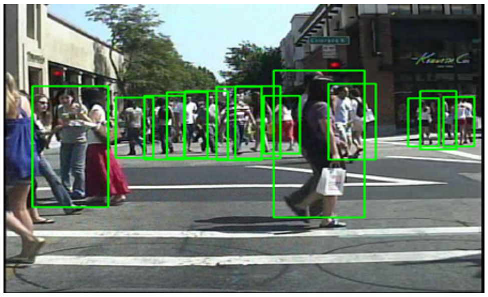


Figure 1.3. Accurate motion estimation methods help pedestrian detection [**176**] © 2010 IEEE.

For example, removing background pixels is more desirable than removing foreground objects. One key challenge is to make sure that the resized video is temporally coherent. Wang *et al.* [**181**] use motion to define the temporal persistence of the video content and their motion-based method can achieve high-quality video resizing results for videos with complex motions.

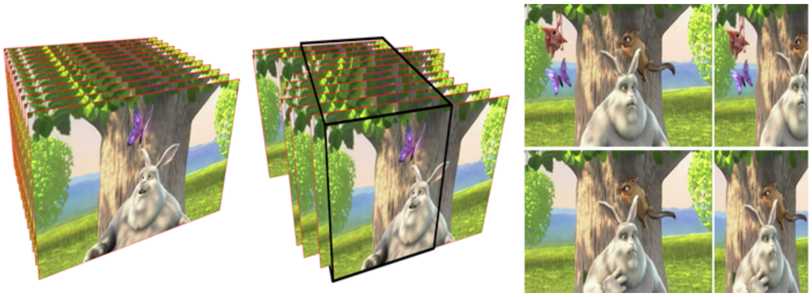


Figure 1.4. Accurate motion estimation methods enable temporally coherent video resizing [**181**] © 2010 ACM.

1.2. Challenges

1.2.1. Aperture Problem. The underlying assumption behind optical flow is *brightness constancy*. A pixel is assumed to retain its brightness or certain image property despite its position change over time. However the brightness constancy constraint is locally inadequate to recover motion. Every pixel has only one measurement[1], while we want to solve for the two unknown components of the motion vector. For a given pixel in the first image, we can find many pixels with similar color in the second image, as shown in Figure 1.5. Hence optical flow is an inherently ill-posed problem.

1.2.2. Motion Boundaries. One common assumption to make the problem well-posed is the *smoothness* of the optical flow. Neighboring pixels usually come from the same surface in the 3D world and should have similar motion. The smoothness assumption tends to fail across motion boundaries, because the neighboring pixels are from different surfaces and may have different motion, as shown in Figure 1.6. Imposing the smoothness at these regions inevitably blurs the motion boundaries.

[1] Color seems to provide multiple observations but the three color components give nearly the same constraint.

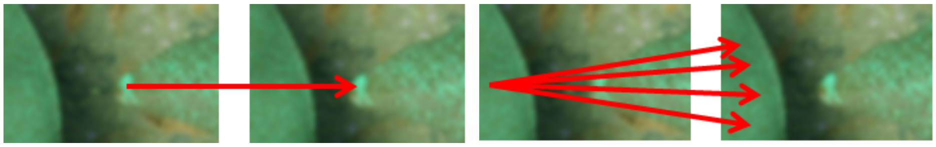


Figure 1.5. Brightness constancy is commonly assumed for optical flow estimation. It is possible to locally track salient image features (left) but not homogeneous regions (right).

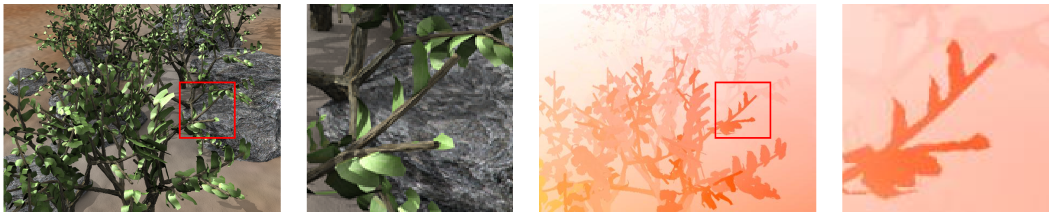


Figure 1.6. Motion boundaries: imposing smoothness across the motion boundaries will destroys the fine motion structures.

1.2.3. Occlusions. Occlusions make the recovery of motion boundaries even harder. When two moving objects meet, the front object occludes the one behind. The occluded pixels do not have corresponding pixels in the next image and usually violate the brightness constancy assumption, as shown in Figure 1.7.

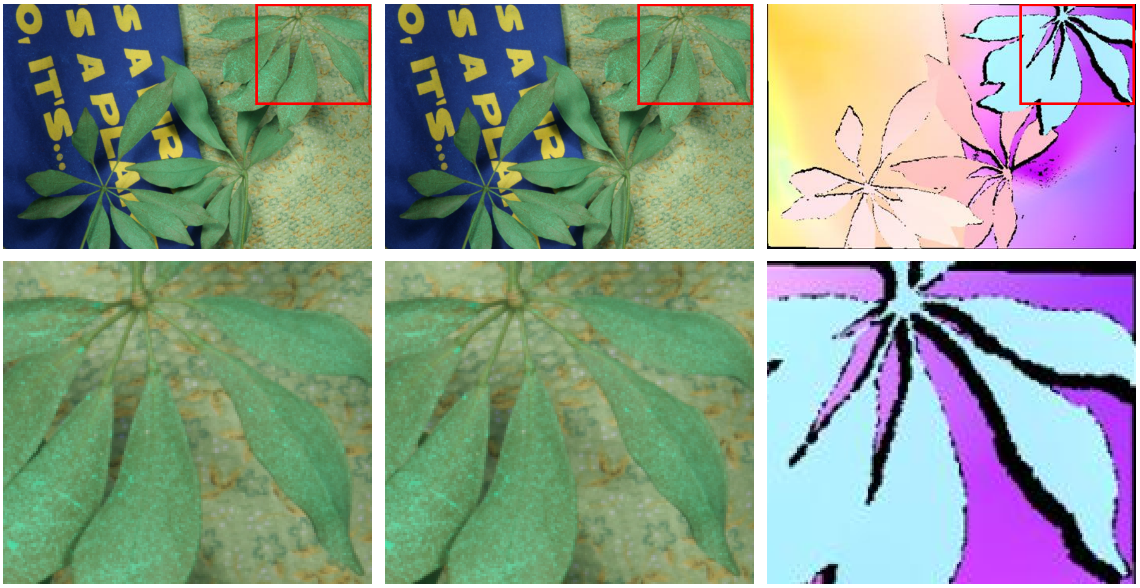


Figure 1.7. Dark pixels in the ground truth are occluded and brightness constancy does not hold for these occluded pixels. From left to right: first image, second image, and the ground truth flow field.

1.3. Approach

Let us summarize our discussions up to now. Optical flow estimation is an inherently ill-posed problem. We cannot determine the image motion locally by the data (brightness) constancy assumption and need smoothness prior models of the flow field. Motion boundaries and occlusion regions violate the smoothness assumption and make the problem challenging.

Motion estimation is usually regarded as a typical low-level vision problem. The ill-posedness of the problem requires prior models. Pairwise Markov Random Field (MRF) models enforce neighboring pixels to have similar values (motion) and have been popular in low-level vision tasks. We can allow some motion discontinuities by using robust potential functions, but the robust MRF models are still insufficient to recover sharp motion boundaries and occlusion regions [**16**]. We need more advanced models.

1.3.1. Image-dependent, Non-local Flow Prior Model. The idea has been around for some time that static image cues provide useful information for detecting and preserving motion boundaries. Motion boundaries usually occur across the boundaries of different moving objects. These objects tend to have different appearances. Consequently motion boundaries and appearance (color) boundaries are likely to coincide. Figure 1.8 shows a cropped motion detail as well as the corresponding image. The static image edges can well predict the location of the motion boundaries. We can make the spatial model dependent on the input images to take advantage of such knowledge.

We study two image-dependent prior models for motion estimation. The first one exploits the so called oriented smoothness introduced by Nagel and Enkelmann [**117**]. Locally the flow field is more likely to be smooth in the direction parallel to the image edges and to have discontinuities in the orthogonal direction. We use training data to analyze and model the statistical relationship between image and flow boundaries by a Steerable Random Field (SRF). The second one uses the observation that in a large spatial neighborhood, pixels with similar colors are more likely to come from the same surface and so share similar motion. The commonly used pairwise MRF only allows interaction between a pixel and its four nearest pixels. We show that direct interaction between long-range spatial neighbors enables better integration of the spatial information to preserve motion boundaries.

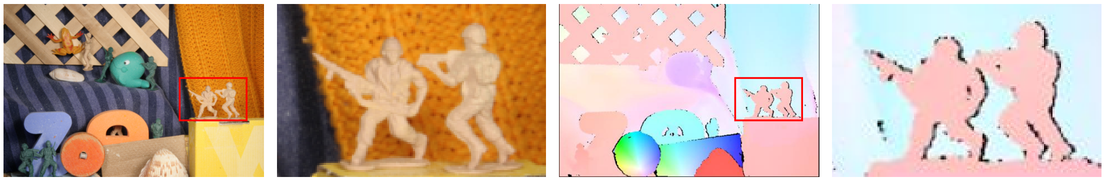


Figure 1.8. Static image information, such as color, provides strong cues to detect and preserve motion boundaries. Left pair: first frame and its enlarged detail; right pair: ground truth flow field and its enlarged detail (dark pixels correspond to occlusion regions). The motion boundaries coincide with the image edges. In addition, a pixel tends to have similar motion with pixels coming from the same surface, though the two may not be directly adjacent to each other.

1.3.2. The Layered Approach. Motion estimation is inherently related to segmentation. If we can separate each moving objects (segmentation), we can integrate information only within a particular object to preserve motion boundaries. Segmentation also allows occlusion reasoning over time. On the other hand, different objects tend to have different motion; knowledge of motion enables us to perform segmentation more confidently.

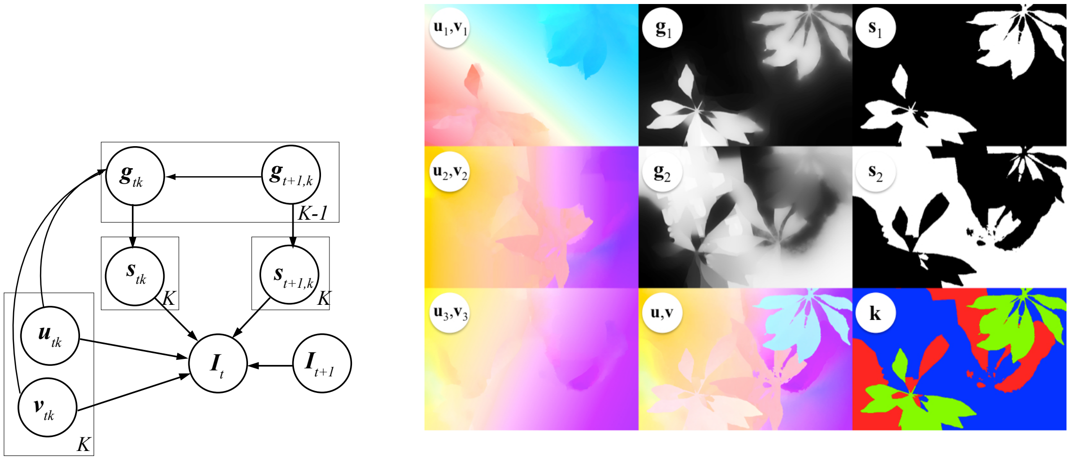


Figure 1.9. *Left:* Graphical representation for the proposed layered model. *Right:* Illustration of variables from the graphical model for the "Schefflera" sequence. Labeled sub-images correspond to nodes in the graph. The left column shows the flow fields for three layers, color coded as in [**16**]. The **g** and **s** images illustrate the reasoning about layer ownership (see text in Chapter 5). The composite flow field $(\mathbf{u}, \mathbf{v})$ and layer labels $(\mathbf{k})$ are also shown.

However, existing methods tend to solve either motion or segmentation separately, with the assumption that a satisfactory solution to other problem is available.

We treat both the motion and the segmentation as unknowns and solve for them simultaneously using a layered approach. The layered approach has been popularized by Wang and Adelson [**180**] and has been regarded as an elegant approach to motion. However, previous layered methods have not reported state-of-the-art motion estimation results on the widely used Middlebury optical flow benchmark [**16**]. We develop a probabilistic layered model that fixes several key issues of previous approaches. Our model explicitly models the depth ordering of layers, occlusions between layers, and the temporal consistency of the layered structures, as shown in Figure 1.9.

1.4. Contributions and Overview

In this dissertation, we make two major contributions. First, we show that image-dependent, non-local prior models can better integrate spatial information and preserve motion boundaries than the widely used pairwise Markov Random Field (MRF) models. Second, we show that joint motion estimation and segmentation by the layered approach produces more accurate results than the separate treatment of each problem, in particular in motion boundary and occlusion regions.

In Chapter 2, we review the basic concepts behind optical flow estimation, in particular the energy minimization approach introduced by Horn and Schunck [**73**] and the layered approach popularized by Wang and Adelson [**180**].

In Chapter 3, we study the statistics of optical flow based on a training dataset with ground truth flow fields. We learn standard models of optical flow and extend the standard model to deal with lighting changes and oriented smoothness of the flow field.

In Chapter 4, we perform a quantitative evaluation of recent practices in optical flow estimation starting from the classical formulation by Horn and Schunck. We find that the key is to perform a median filtering of the flow field during iterative flow estimation. We show that the median filtering is equivalent to using a non-local term to integrate information in a large spatial neighborhood. Based on this insight, we modify the non-local term to use static images cues to detect and preserve motion boundaries.

In Chapter 5, we develop a probabilistic layered model based on thresholded continuous support functions. Using spatio-temporal coherent support functions explicitly models depth ordering, occlusions, and temporal consistency of the scene structure. We also use semi-parametric motion models to capture globally coherent but locally flexible flow fields.

In Chapter 6, we propose a discrete-continuous optimization scheme to avoid local minima of our layered model. We formulate a discrete layered model using a sequence of depth ordered Ising MRFs and introduce non-standard "cooperative" moves to optimize our model. The proposed layered method achieves state-of-the-art performance on the Middlebury optical flow benchmark [**16**].

In Chapter 7, we summarize findings of this dissertation and describe limitations of our approach. We also discuss directions that may be worth investigating in the future.

1.5. List of Related Papers

This dissertation is based on the materials in the following papers [**158, 159, 160, 161**][2].

- Deqing Sun, Erik B. Sudderth, and Michael J. Black. "Layered Segmentation and Optical Flow Estimation over Time." *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1768–1775, 2012.
- Deqing Sun, Erik B. Sudderth, and Michael J. Black. "Layered Image Motion with Explicit Occlusions, Temporal Consistency, and Depth Ordering." *Advances in Neural Information Processing Systems (NIPS)*, pages 2226–2234, 2010.
- Deqing Sun, Stefan Roth, and Michael J. Black. "Secrets of Optical Flow Estimation and Their Principles." *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2439, 2010.
- Deqing Sun, Stefan Roth, J.P. Lewis, and Michael J. Black. "Learning Optical Flow." *European Conference on Computer Vision (ECCV)*, Part III, pages 83–97, 2008.

[2] Joint work with Prof. Michael J. Black, Dr. J.P. Lewis, Prof. Erik B. Sudderth, and Prof. Stefan Roth.

CHAPTER 2

A Review of the State of the Art

In this chapter, we first review dense optical flow estimation methods descended from the classical approach by Horn and Schunck, because nearly all the top-performing methods take this global approach. We then discuss the relationship between motion and segmentation and review related work. We then review the layered approach to jointly solve for the motion and the scene segmentation. Finally, we address how to learn the model parameters and how training and testing datasets can be used for both training and evaluation. Later chapters provide additional discussions of methods related to the particular method of each chapter.

2.1. Optical Flow Estimation

In this section, we review previous approaches in terms of three key properties: the model (objective function), the optimization method, and the implementation details.

2.1.1. Models. The fundamental assumption behind optical flow is the constancy of brightness, or some image property, despite the position change of the pixels. Mathematically,

$$I_t(i,j) \approx I_{t+1}(i+u_t^{i,j}, j+v_t^{i,j}), \tag{1}$$

where I_t and I_{t+1} are the current and the next input frames, i and j are the horizontal and vertical pixel indices, and $u_t^{i,j}$ and $v_t^{i,j}$ is the horizontal and vertical motion for pixel (i,j) at frame t.

The brightness constancy assumption is often violated in natural scenes with non-Lambertian reflectance, complex or changing lighting, cast shadows, or occlusion. For small motions it is surprising however how often the assumption is valid (or at least useful). Nevertheless, the constancy constraint is only a necessary condition and usually insufficient to recover the underlying motion. For every pixel in the first frame, we can find many pixels in the next frame with similar brightness, making the problem ill-posed. We need to pool information from spatial neighbors to overcome the ill-posed problem, either locally or globally.

Local methods assume the motion of all the pixels within a local window follows a parametric form [**105, 164**], such as translational motion in the simplest case. This approach works at well-textured regions but becomes ill-posed in homogeneous regions. Defining the right window size is a challenge to the local approach. A large-sized window allows the integration of information over many pixels but may include pixels in other motion groups. The dilemma in choosing the right window size is usually referred to as the generalized aperture problem [**26**].

Global methods assign an individual motion vector for every pixel and solve for the motion field using the information from the whole image. This approach makes assumptions about the local behavior of the motion fields. For example, neighboring pixels usually come from the same surface and therefore tend to have the same motion, i.e.,

$$u_t^{i,j} \approx u_t^{i+1,j}, \quad u_t^{i,j} \approx u_t^{i,j+1}, \tag{2}$$

$$v_t^{i,j} \approx v_t^{i+1,j}, \quad v_t^{i,j} \approx v_t^{i,j+1}. \tag{3}$$

The global approach defines an energy (objective) function to combine conflicting constraints on the unknown flow fields. Horn and Schunck (HS) [**73**] propose to combine a brightness (data) constancy term that assumes constancy of some image property with a spatial smoothness term that models how the flow is expected to vary across the image.

$$\begin{aligned} E(\mathbf{u}_t, \mathbf{v}_t) =& E_{\text{data}}(\mathbf{u}_t, \mathbf{v}_t) + \lambda E_{\text{space}}(\mathbf{u}_t, \mathbf{v}_t) \\ =& \sum_{(i,j)} \Big\{ \big(I_t(i,j) - I_{t+1}(i + u_t^{i,j}, j + v_t^{i,j})\big)^2 + \lambda\big[(u_t^{i,j} - u_t^{i+1,j})^2 \\ & + (u_t^{i,j} - u_t^{i,j+1})^2 + (v_t^{i,j} - v_t^{i+1,j})^2 + (v_t^{i,j} - v_t^{i,j+1})^2\big]\Big\}. \end{aligned} \tag{4}$$

The minimum of the energy function is the solution.

One key component of the energy minimization approach is the definition of how to penalize the violations of the model assumptions. Horn and Schunck use a quadratic penalty function which makes the optimization relatively easy. Probabilistically the quadratic penalty corresponds to Gaussian assumptions and is not robust to outliers, such as occlusions and motion boundaries. Shulman and Herve [**149**] use a Huber minimax robust penalty for the spatial term to preserve motion boundaries. Black and Anandan [**25, 26**] introduce a robust framework to deal with outliers in both the data and the spatial terms.

$$\begin{aligned} E(\mathbf{u}_t, \mathbf{v}_t) =& \sum_{(i,j)} \Big\{ \rho_D\big(I_t(i,j) - I_{t+1}(i + u_t^{i,j}, j + v_t^{i,j})\big) + \lambda\big[\rho_S(u_t^{i,j} - u_t^{i+1,j}) + \\ & \rho_S(u_t^{i,j} - u_t^{i,j+1}) + \rho_S(v_t^{i,j} - v_t^{i+1,j}) + \rho_S(v_t^{i,j} - v_t^{i,j+1})\big]\Big\}. \end{aligned} \tag{5}$$

The robust penalty functions give less confidence to the large-norm errors to reduce their influence on the estimates. The robust approach has been widely adopted in later approaches [**38, 185, 197**].

Although allowing outliers to certain extent, the robust approach still fails and does not produce satisfactory results when the outliers dominate in motion boundary and occlusion regions. The dominance of the outliers requires more principled treatment for a satisfactory solution. We can use the prior knowledge to explicitly model these phenomena.

The smoothness term by Horn and Schunck assumes that neighboring pixels tend to have similar motion, and can be probabilistically interpreted as a pairwise Markov Random Field (MRF) model. One observation is that the static image information provides strong cues for detecting motion boundaries. Different surfaces tend to have different material properties, resulting in intensity/color edges in the 2D image. Motion boundaries usually happen at surface boundaries. Hence we can use the static image edges to predict the location of the motion boundaries. Nagel and Enkelmann [**117**] first introduce "oriented smoothness" to preserve motion discontinuities using static image edges. Specifically, the flow fields are enforced to be smooth in the direction parallel to image intensity edges but allowed to be discontinuous in the direction orthogonal to image edges, as shown in Figure 2.1. Probabilistically we can formulate the oriented smoothness as a Steerable Random Field (SRF) [**134, 159**], which allows the flow field to have different behaviors in the directions orthogonal and parallel to image edges. Zimmer *et al.* [**204**] interpret the SRF model as a Joint Image and Flow (JIF) regularizer in the variational setting and extend the SRF/JIF model by calculating the local orientations using the motion tensor from the data constraint term. Several recent methods [**184, 189, 197**] adopt an image-dependent smoothness prior model too.

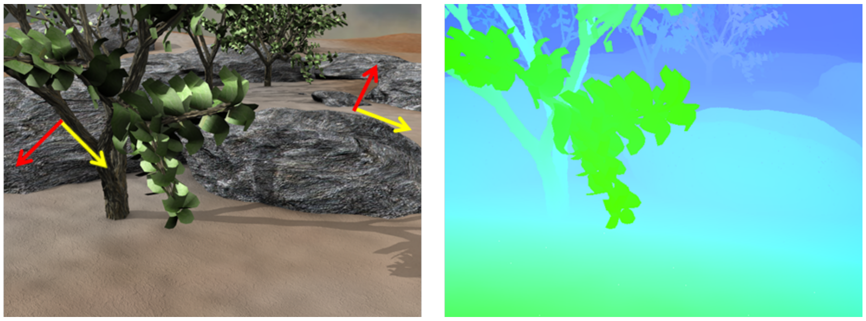


Figure 2.1. The flow field tends to be smooth in the direction parallel to image edges (yellow) but may have discontinuities in the direction orthogonal to image edges (red).

Another observation is that pairwise models are not expressive enough to model the rich structure of natural images and scene motion [**135**]. Previous work on optical flow has considered the second-order smoothness prior model [**8, 171**] and the high-order Fields-of-Expert (FoE) prior model [**133**]. The FoE models have been trained on flow fields generated using depth maps and may not be optimal for general video sequences. There are not enough representative real-world sequences with associated ground truth flow fields for training.

Optical flow assumes that every pixel goes somewhere. However some pixels may disappear, either being occluded by other pixels or moving out of the image, and others may appear. The occluded pixels violate the constancy assumption and produce large errors with standard optical flow formulation. If we can detect these pixels, we can modify the flow formulation to handle the occlusions. The detected occlusion results are also useful for scene understanding and play an important role in our interactions with the world. Therefore, occlusion detection has been studied in various contexts.

In optical flow estimation, previous work has used the symmetric constraint between the forward and the backward flow fields [**7**] to detect occlusions. Without occlusions, a pixel at the current frame should correspond to the same pixel at the next frame according to the both the forward and the backward flow field. Another criterion borrowed from the stereo community is unique mapping [**197**]. Every pixel in the next frame should correspond to at most one pixel in the current frame. If multiple pixels at the current frame correspond to the same pixel in the next frame, at least one of the pixels is occluded. This unique mapping criterion however does not classify the occluding pixels from the occluded pixels.

Kolmogorov and Zabih [**86**] formulate occlusion detection as a binary classification problem and solve the NP hard problem using approximate graph cuts methods. Strecha *et al.* [**156**] use a mixture formulation and detect large data constancy errors as occlusions. Ayvaci *et al.* [**12**] classify the data constancy errors into occluded and unoccluded, apply a sparse prior on the region of occlusions, and solve the convex problem using reweighted-L1 minimization.

Stein and Hebert [**153**] use Adaboost [**49**] to learn a classifier using motion and appearance cues. He and Yullie *et al.* [**68**] achieve similar performance using pseudo depth map obtained from optical flow and local edge maps. Sundberg *et al.* [**163**] use motion differences predicted by each side of static image boundaries and improve over both methods above. Humayun *et al.* [**76**] train a two-frame occlusion detector using a variety of cues, such as appearance, motion, and textures. However, none of the more advanced work has concretely shown the benefits of occlusion detection for optical flow estimation.

2.1.2. Optimization Method. One challenge in optimizing the HS objective is the nonlinear data term. The unknown motion vectors are input variables to a nonlinear image function. Assuming that the motion is small, we can do a first-order Taylor expansion w. r. t. the motion to linearize the data term

$$I_{t+1}(i+u_t^{i,j}, j+v_t^{i,j}) \approx I_{t+1}(i,j) + \frac{\partial I_{t+1}}{\partial x}(i,j)u_t^{i,j} + \frac{\partial I_{t+1}}{\partial y}(i,j)v_t^{i,j}, \tag{6}$$

where $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ are the derivative operators in the horizontal and vertical directions. The so called linearization step assumes that the motion is be small to justify the first-order approximation, as shown in Figure 2.2. The gradient of the data term for the quadratic energy function w. r. t. the horizontal flow field at (i,j) is

$$\frac{\partial E_{\text{data}}(\mathbf{u}_t, \mathbf{v}_t)}{\partial u_t^{i,j}} = 2\big(I_{t+1}(i,j) - I_t(i,j) + \frac{\partial I_{t+1}}{\partial x}(i,j)u_t^{i,j} + \frac{\partial I_{t+1}}{\partial y}(i,j)v_t^{i,j}\big)\frac{\partial I_{t+1}}{\partial x}(i,j). \tag{7}$$

the gradient of the spatial term for the quadratic energy function w. r. t. the horizontal flow field at (i,j) is

$$\frac{\partial E_{\text{space}}(\mathbf{u}_t, \mathbf{v}_t)}{\partial u_t^{i,j}} = 2\big[4u_t^{i,j} - u_t^{i-1,j} - u_t^{i,j-1} - u_t^{i+1,j} - u_t^{i,j+1}\big]. \tag{8}$$

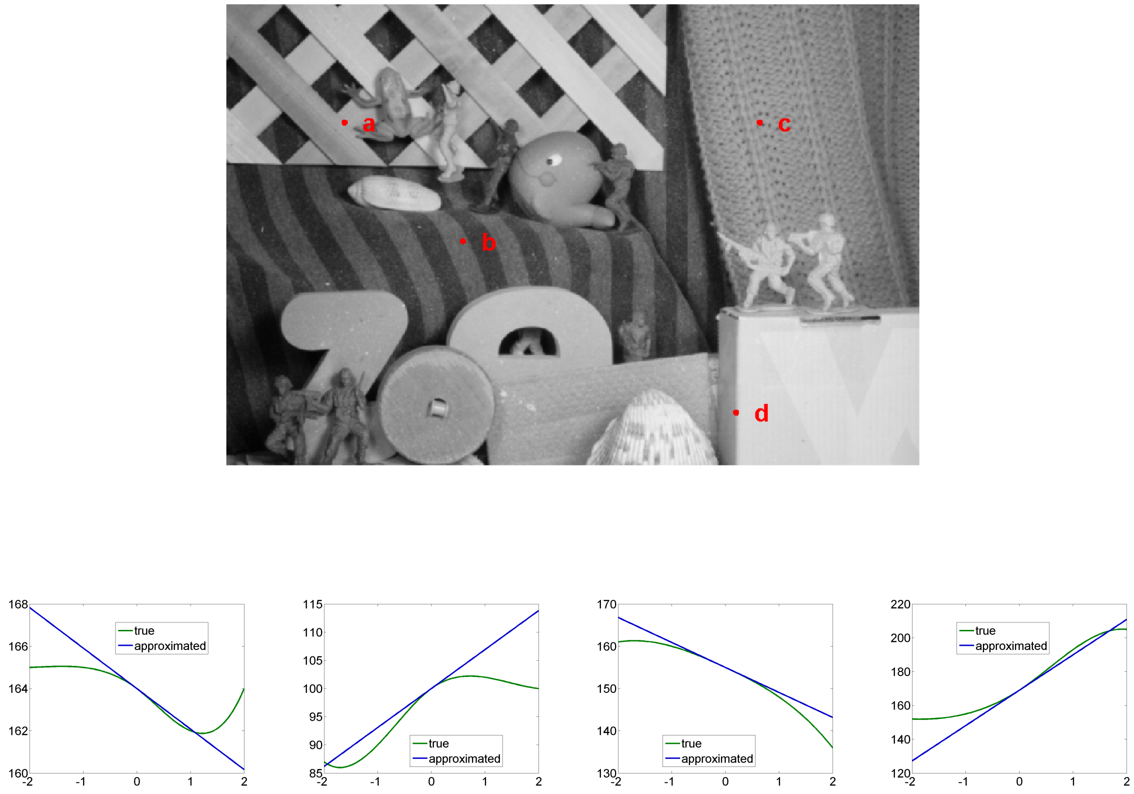
We can then perform gradient-based optimization.

To deal with large motions, a common approach is to adopt a coarse-to-fine, warping-based optimization [**6, 21, 38**], as shown in Figure 2.3. We warp the second image toward the first using the current flow estimate and then estimate a small increment between the first image and the warped image. Because the flow increment is small, the linear approximation holds. This warping method has been theoretically justified as a numerical scheme to optimize the original energy function [**34**].

An additional challenge to optimize the robust version of the HS objective is the robust penalty function. We can approximate the original objective with a quadratic objective around the current motion estimate $\mathring{\mathbf{u}}, \mathring{\mathbf{v}}$ and solve for the small flow increment using the quadratic approximation.

$$\begin{aligned} E(\mathbf{u}_t, \mathbf{v}_t) = \sum_{(i,j)} \Big\{ & W_D\big(I_t(i,j) - I_{t+1}(i+\mathring{u}_t^{i,j}, j+\mathring{v}_t^{i,j})\big) \cdot \big(I_t(i,j) - I_{t+1}(i+u_t^{i,j}, j+v_t^{i,j})\big)^2 \\ & + \lambda\big[W_{Sux}(\mathring{u}_t^{i,j} - \mathring{u}_t^{i+1,j}) \cdot (u_t^{i,j} - u_t^{i+1,j})^2 + W_{Suy}(\mathring{u}_t^{i,j} - \mathring{u}_t^{i,j+1}) \cdot (u_t^{i,j} - u_t^{i,j+1})^2 \\ & + \quad W_{Svx}(\mathring{v}_t^{i,j} - \mathring{v}_t^{i+1,j}) \cdot (v_t^{i,j} - v_t^{i+1,j})^2 + W_{Svy}(\mathring{v}_t^{i,j} - \mathring{v}_t^{i,j+1}) \cdot (v_t^{i,j} - v_t^{i,j+1})^2\big]\Big\}, \end{aligned} \tag{9}$$

where the weights $W_D(\cdot)$, $W_{Sux}(\cdot)$, $W_{Suy}(\cdot)$, $W_{Svx}(\cdot)$, and $W_{Svy}(\cdot)$ depend on the current motion estimates and the selected robust function. For example, the weight for the Lorentzian penalty function $\rho(x) = \log(1 + \frac{x^2}{2\sigma^2})$ is $W(x) = \frac{2}{2\sigma^2 + x^2}$. This can be considered as iterated reweighted least squares [**98**], variational inference [**82**], or fixed-point iteration [**34, 124**] methods.



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

Figure 2.2. Local Taylor expansion approximation in the horizontal direction for each selected pixel; each plot has been shifted so that the select pixel is at the origin position. The approximation is not accurate in regions more than one pixel away form the selected pixels

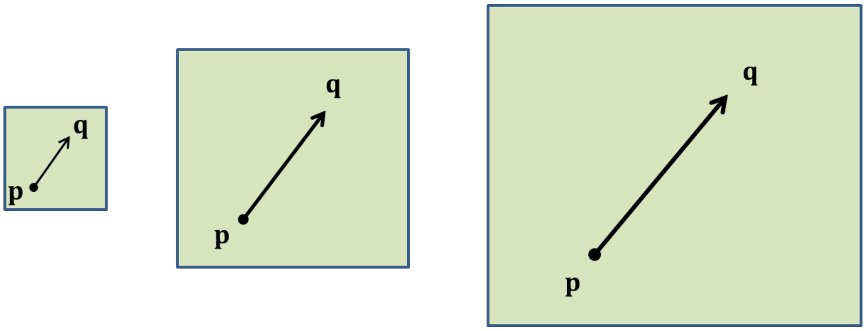


Figure 2.3. Coarse-to-fine, incremental estimation is a standard technique to deal with large motion, which becomes smaller as we downsample the images. **q** is the corresponding pixel at the next frame for pixel **p**.

Large motion and fast-moving objects. Coarse-to-fine, pyramid-based estimation tends to produce errors when the size of the object is much smaller than the motion of the object. When we downsample the images to build the pyramid, the small-sized objects will be merged into the background. Their motion estimate at the coarse level is wrongly assigned to be that of the background. Because the warping-based approach can only change the flow field locally, the solution gets stuck at the wrong estimate from the coarse-level estimation.

Wills *et al.* [**193, 192**] propose a two-stage, feature-based process to deal with large motion. At the first stage, feature matching provides the initial correspondence and the gross motion. The graph cuts method is used to segment the scene into different moving objects (layers). At the second stage, a regularized thin plate spline fit is used to account for non-planar motion. However, the thin plate spline model prefers a smooth motion field without discontinuities. Berg *et al.* [**20**] use Integer Quadratic Programming (IQP) to regularize descriptor matching, but the IQP method is computationally expensive and only allows sparse correspondence.

Brox *et al.* [**33, 37**] perform descriptor matching to find correspondence for fast-moving objects and add a feature matching term to the original objective function [**34**]. The Large Displacement Optical Flow (LDOF) method has a slight decrease in the accuracy of the estimated motion as measured by the Middlebury optical flow benchmark [**37, 16**]. Steinbrucker *et al.* [**154**] decouples the original objective into the data and the spatial term. They use exhaustive search to solve the data term to deal with the non-linear data term. The method is computationally expensive because of the exhaustive search, in particular when fine-grid search is adopted for accurate motion estimation.

Another recent development is the FusionFlow work by Lempitsky *et al.* [**92, 93**]. They propose to fuse a set of flow candidates to minimize the nonconvex energy function via discrete optimization. At each move, they solve a binary fusion problem to select the motion of every pixel from two candidate flow fields. The fused solution is guaranteed to have no higher energy than the two candidate flow fields. The original FusionFlow work uses solutions of the HS objective and the Lucas-Kanade methods to generate candidate flow fields. Xu *et al.* [**197**] use SIFT feature matching [**104**] to generate flow candidates. They fuse the SIFT matching candidate flow fields with the initial flow field from coarse-level estimation at every image pyramid level to get rid of local optima and recover fine motion details.

Rhemann *et al.* [**129**] propose a filtering framework to estimate the correspondence. They first compute the data matching cost for every possible motion vector for each pixel and then use filtering guided by the image information to the data matching cost. Experimentally the cost filter approach can estimate fast motion in some Middlebury sequences with large displacement. The explicit objective function for the cost-filtering approach however has not been established.

Shekhovtsov *et al.* [**146**] reformulate the optical flow problem by treating the data matching term as a pairwise interaction term between the horizontal and vertical motion and solve the resultant pairwise MRF model using the Belief Propagation (BP) algorithm. The method is restricted to (discrete) integer motion because of the discrete nature of the formulation. Liu *et al.* [**102**] use the SIFT features [**104**] in the data term and compute the semantic correspondence between different scenes. The "SIFT flow" correspondence enables high-level tasks, such as label transfer [**101**]. Because the SIFT feature does not have good localization ability, the SIFT flow does not perform well in standard optical flow estimation problem.

2.1.3. Implementation Details. Existing methods differ in how to define the objective functions, how to approximate the objective functions to make the computation feasible, and how to optimize them. Some implementation details have been reported to be important in flow estimation, such as the downsampling ratio to construct the pyramid [**124**], the anti-aliasing filter for the pyramid [**38**], and the pre-processing method to deal with lighting change [**124, 185**]. However the conclusions are based on a particular method or a particular image sequence (mostly the "Yosemite" sequence).

The brightness constancy assumption of HS tends to fail for lighting changes, occlusions, transparency, noise etc. Gaussian filtering has been used to reduce the influence of noise [**38, 97**]. High-order (filter response) constancy has been used to deal with lighting changes which usually belong to the low-frequency components [**6, 34**]. We can learn the filters for the filter response constancy [**159**]. Zhou *et al.* [**201**] learn a discriminative similarity function for motion estimation, but only for sparse feature tracking. Wedel *et al.* [**185**] propose to decompose the input images into structure and texture components and combine the two components in a certain proportion as the input to a flow estimation method. Xu *et al.* [**197**] adaptively choose the color constancy assumption and the gradient constancy assumption for the data term.

One interesting implementation detail is to perform a median filtering of the intermediate flow field during the iterative warping step. This step has been used in several recent methods that achieved good performance on the Middlebury benchmark when these methods first appeared [**183, 184, 185, 189**]. The median filtering step has been used as a heuristic step to remove outliers in the estimated flow field

2.1.4. Existing Issues. Despite the progress, the classical approach and its descended versions still have problems in recovering fine motion boundaries that violate the smoothness assumption, and occlusion regions that violate the data constancy assumption.

2.2. Optical Flow and Segmentation

The pooling of information by the spatial term inherently relates optical flow to segmentation. If we can segment the scene into different moving objects, we can integrate spatial information only for each moving object and thereby preserve motion boundaries. If we know the true motion of every pixel, separating different objects becomes easier. Both motion and segmentation are fundamental problems in computer vision but are often treated separately. Next we will classify and review papers according to their emphasis.

2.2.1. Static Image Segmentation. The segmentation of scenes into regions of coherent structure is a fundamental problem in computer vision. It simplifies the representation of images into something that is more meaningful and easier to analyze. The goal of image segmentation is to cluster pixels into salient image regions, i.e., regions corresponding to individual surfaces, objects, or natural parts of objects. Static image segmentation has received intensive research attention and made significant progress in recent years. However, the segmentation results of state-of-the-art methods [**9, 43, 53, 108, 148**] are still far from those obtained by humans. One challenge is that the object appearance is not a strong cue for separation against the cluttered background.

In the graphics community, several interactive segmentation systems have been developed [**14, 95, 136**]. Humans provide direct feedback at places that confuse the computers. These systems have been widely used in the graphics and movie communities, clearly demonstrating the need for accurate video segmentation. However, these methods focus on separating a single foreground object from the background and are not easily applicable to generic video with multiple independently moving foreground objects.

2.2.2. Motion for Segmentation. This approach first estimates motion without segmentation information and then uses the estimated motion to assist segmentation.

One straightforward way to extend the static image segmentation methods to video data is to model the video as a 3D spatio-temporal volume and add one dimension to the 2D methods. The analysis is performed locally and the segmentation results are usually not consistent over time. Grundmann *et al.* [**65**] extend the graph-based method [**53**] to video data and use optical flow as an additional feature. Lezama *et al.* [**94**] further add constraints from long-term point trajectories and reason about the occlusion relationships. The optical flow is precomputed without any segmentation or occlusion information. The errors in motion estimation propagate into the segmentation results. Brox and Malik [**36**] cluster point trajectories into different motion groups. However, their output are sparse points and not coherent segmentation regions. Ochs and Brox [**120**] define a variational model to interpolate the sparse clustering results to dense segmentation. Their method may suffer from the errors made by the trajectory analysis process and does not reason about occlusion relationships.

The limitation of the motion for segmentation approach is that errors in the motion estimation process propagate into segmentation.

2.2.3. Static Segmentation for Motion. This approach [**30, 91, 196, 205**] first segments the static images without motion information and then performs motion estimation with the precomputed segmentation. Errors in the static segmentation may propagate into the estimated motion. Falsely merging two moving objects produces big errors in motion estimation. Consequently these methods perform over-segmentation of the static images to avoid big motion errors. The small segment size often cannot capture the global information within a moving object.

2.2.4. Variational Approach to Joint Segmentation and Estimation. We can treat both the motion and the segmentation as unknowns and solve for both simultaneously. The motion competition framework [**35, 44**] uses level sets to model the scene segmentation in a variational setting. Mémin and Pérez [**111, 112**] use robust techniques to couple the motion and the segmentation by sharing the hidden line processes between the motion and the segmentation. However the depth ordering of the segments is missing and these methods cannot reason about occlusions.

2.3. Layered Models

One promising approach to deal with occlusions is the layered model, as shown in Figure 2.4. A video sequence usually contains very few moving objects (layers). If we can segment each individual layer and know the relative depth ordering, we can reason about occlusions between layers. We can also describe the motion more compactly for each layer. Besides being a compact representation, the layered model allows the integration of information over multiple frames. With the separation of the individual moving objects, the motion boundaries are separated from the smoothness and the pixel appearances near the motion boundaries are more accurately modeled.

We first briefly review the early work on layered models and then discuss how later work improves the early work in both the modeling and the inference.

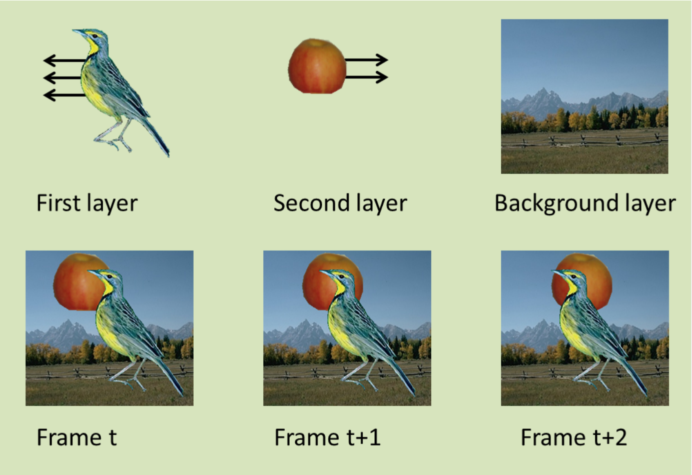


Figure 2.4. The layered decomposition of a synthetic sequence. The bird layer is in front of the apple layer and so occludes the apple layer.

2.3.1. Early Formulations. Darrell and Pentland [**47, 46**] provide the first full approach that incorporates a Bayesian model, support maps for segmentation, and robust statistics. Compared with the robust approach/line process formulation, they show that region-based support maps can better integrate information from disconnected regions separated by foreground objects. They apply the layered approach to solve a variety of segmentation problems, including range image, image sequences, and tracked points. Wang and Adelson [**179**] clearly motivate layered models of image sequences. They illustrate the benefits of separating smoothness from discontinuities by describing motion discontinuities as discontinuities in the object surfaces rather than discontinuities in the actual object motion. They further demonstrate the layered approach as a compact representation of the video data by developing an efficient video encoder using their layered decomposition algorithm. Jepson and Black [**78**] formalize the problem using probabilistic mixture models and derive the corresponding Expectation Maximization (EM) algorithm. Their work assumes translational motion for each layer and does not estimate the number of layers. Ayer and Sawhney [**11**] use the mixture model formulation with affine motion models and determine the number of layers using the Minimum Description Length (MDL) principle.

2.3.2. Motion Models. Early methods use simple parametric models of image motion within layers [**46, 78, 179**]. The parametric models can compactly represent the motion and impose global coherence on the estimated motion fields. However the parametric models are not highly accurate and only apply to restrictive scenes. For example, the commonly used affine motion is a good approximation to planar surfaces far away from the camera. Observing that rigid parametric models are too restrictive for real scenes, Weiss [**186**] proposes using the more flexible nonparametric

Gaussian processes to describe the motion within each layer. Although the Gaussian process model can better handle curved surfaces, the model still cannot capture the fine local behaviors.

Some work considers combing the benefits of both the parametric and the nonparametric models. Similar to the "plane + parallax" work [**77, 90, 141**] in 3D reconstruction, Hsu *et al.* [**75**] first fit a parametric motion for each layer and then estimate the residual optical flow from the parametric motion for each pixel. They describe the process in an algorithmic way but do not have an explicit formulation. For stereo reconstruction, Baker *et al.* [**18**] propose to decompose the scene into a collection of 3D layers. Each layer has a plane equation, and a per-pixel depth offset relative to the plane of the layer. Torr *et al.* [**169**] use a similar motion model for layer extraction in general video sequences.

For standard optical flow estimation, more advanced nonparametric motion models have been developed, such as robust MRF [**26**], image-dependent prior models [**117**], higher-order MRF [**133**], and non-local terms [**158, 188**]. These advanced models can preserve the fine motion details but have not been adopted by previous layered methods.

2.3.3. Depth Ordering. One key advantage of the layered model is the ability to realistically model occlusion boundaries and reason about occlusions. To do this properly, however, one must know the relative depth ordering of the layers. In their pioneering work, Wang and Adelson [**179**] incorporate the depth ordering of layers and layers in the front occlude layers behind. They determine the depth ordering of the layers heuristically by comparing the predicted appearance with the observed image intensity. Critically the motion estimation process does not have the occlusion information and hence the estimated motion tends to have big errors in the occlusion regions. The errors in the motion will further propagate into the segmentation and occlusion determination step.

Later models however tend to ignore the depth ordering in the formulations or do not infer the depth ordering. Performing inference over the combinatorial range of possible occlusion relationships is challenging and, consequently, only a few layered flow models explicitly encode relative depth ordering [**81, 202**]. For example, the widely-used Ising/Potts segmentation models have no notion of depth ordering: switching the ordering of any two layers causes no difference in the energy of the solutions. Xiao and Shah [**194**] observe that over a short video clip, the occlusion region increases with time, i.e., the occlusion region between layers A and B from frame t to $t+1$ is a subset of the occlusion region between A and B from frame t to $t+k$, $k>1$. They formulate the occlusion order constraint within an MRF model and solve the problem using graph cuts. Nevertheless, their model does not order the layers in depth and mainly reasons about occlusions according to the data matching error. Recent work revisits the layered approach to handle occlusions [**63**] but does not explicitly model the depth ordering.

2.3.4. Segmentation Models. Layered models all have some way of making either a hard or soft assignment of pixels to layers. There are numerous ways of assigning pixels to different layers. One extreme case is to assign every pixel an individual layer, while the other is to assign every pixel to be in the same layer. Any assignment will be consistent with the generative process of the layered model and we need a good segmentation model to rule out poor segmentations. Designing a good segmentation model is in fact one of the central problems for the layered approach.

Early work [**46, 78, 179**] assigns every pixel to different layers individually and is not robust to outliers. The noise and the motion estimation error can cause big errors at each isolated pixel. Because neighboring pixels usually come from the same surface (layer), it is more desirable to incorporate spatial coherence into the segmentation. Weiss and Adelson [**187**] introduce spatial coherence to the layer assignment using a spatial MRF model and optimize their objective using local search. They also advocate using static image cues to constrain the motion segmentation. Wills *et al.* [**191**] use the powerful graph cuts algorithm to optimize the layer segmentation. However the Ising/Potts MRF model assigns low probability to typical segmentations of natural scenes [**114**]. In addition the Potts model has no notion of depth ordering, because switching the ordering of any two segments has no influence on the energy of the solutions. Sudderth and Jordan [**157**] propose using thresholded continuous support functions to model the static segmentation of images. The samples from this model have the desired property of being piecewise smooth with continuous and smooth segmentation boundaries. In addition, the segments are ordered in depth and depth ordering is naturally incorporated. The model however is defined only for static images.

2.3.5. Temporal Consistency. While most current optical flow methods use two consecutive frames, layered methods naturally extend to longer sequences [**81, 198, 202**]. It is one of the advantages of the layered approach to integrate information from more than two frames.

Most previous methods use a center mask at the key frame and warp the center mask by parametric motion to each individual frame. This approach may suffer from poor initialization at the key frame and motion estimation becomes harder for frames far away from the center one.

Jepson *et al.* [**79**] define a parametric shape mask for every layer at every frame and enforce the shape parameters at neighboring frames to be similar. The parametric shape model is restrictive for general videos with arbitrarily-shaped objects. In addition, each shape is an isolated region and cannot integrate information from other shapes that belong to the same physical objects but are occluded by some foreground objects. Liu *et al.* [**100**] use postprocessing to ensure the temporal consistency of segmentation for complex videos.

2.3.6. Number of Layers. One of the most difficult problems in grouping is to determine the number of groups. There are numerous possible decompositions. Ideally we expect the number of layers to roughly match the number of independently moving objects in the scene. Hence the algorithm should have the mechanism to reason about the number of layers to use for a particular video sequence.

Most early work assumes a fixed number of layers [**78, 179**]. Darrell and Pentland [**47, 46**] estimate multiple models and select the number of layers according to the Minimum Description Length (MDL) principle. Ayer and Sawhney [**11**] present a mixture model formulation and determine the number of layers by the MDL principle too. Weiss and Adelson [**187**] estimate the number of layers by controlling the noise parameter, which indicates the expected level of model failure. Torr *et al.* [**169**] use a Bayesian decision framework to determine the number of approximately planar layers. Jepson *et al.* [**79**] use stochastic search over solutions with different number of layers and select the one most plausible with their layered model. Common to these methods is the trade off between different requirements, such as the complexity of the motion field versus the complexity of the shape for each layer.

2.3.7. Inference for Layered Models. Inference for the flexible layered model is challenging too. The layered models usually have multiple local optima and it is easy to get trapped. Frey and Jojic [**58**] compare different algorithms, such as Iterated Conditional Mode (ICM), Expectation Maximization (EM), Belief Propagation (BP), and Mean Field (MF) variational inference, for minimizing the free energy of a simplified generative layered model. The model considers the generation of a single image and makes strong independence assumptions to factorize the overall probabilistic model. Even the simplified model has many local minima that can trap these inference schemes.

Iterative optimization of the motion and the segmentation in the EM manner [**48**] has been popular for the layered models. The algorithm usually estimates the motion first and then performs the segmentation, or alternates estimating the motion and the segmentation. Such an optimization scheme is susceptible to local optima where both the motion and the segmentation need to be changed simultaneously. Most early approaches use a local search scheme and tend to get trapped at poor local minima [**147**]. Kumar *et al.* [**89**] and Wills *et al.* [**191**] use discrete optimizers for the layer segmentation tasks to get rid of local minima in segmentation. Thayananthan *et al.* [**166**] combine bottom up and down down cues and optimize their objective by the Expectation Propagation (EP) algorithm [**113**]. Graph cuts have also been used for segmentation and tracking. For example Kumar *et al.* [**89**] alternate the optimization of motion and segmentation and use affine motion models for each layer. Wang *et al.* [**177**] also use graphical models but their method requires the manual segmentation of objects in the first frame and uses parametric motion models.

2.3.8. Existing Issues. Although long regarded as an elegant approach to motion, the layered methods have not achieved competitive motion estimation results when compared with other approaches. As of June 2010, none of the top-performing methods in the Middlebury benchmark took a layered approach. Most previous work focuses on obtaining correct segmentation and uses simple parametric motion models. The parametric models cannot handle the complex motion in the real-world sequences. Furthermore, most models do not incorporate the depth ordering and thus do not take advantage of the layered approach for occlusion reasoning. In addition, inferring the layer segmentation, within layer flow field, the number of layers, and the depth ordering is a big optimization problem. There are many poor local minima that can trap an inference algorithm. We need to carefully define both the model and the inference scheme to overcome these limitations.

2.4. Learning

The statistics of natural images has been widely studied [**137, 135, 203**]. The statistics of optical flow, however, has been less studied, in particular due to the lack of ground truth optical flow fields for real scenes.

Simoncelli *et al.* [**151**] formulate an early probabilistic model of optical flow and model the statistics of the deviation of the estimated flow from the true flow. Their approach however assumes Gaussian noise. Black *et al.* [**28**] learn parametric models for different classes of flow (e.g. edges and bars). Roth and Black [**133**] model the spatial structure of optical flow fields using a high-order MRF, called Field of Experts (FoE), and learn the parameters from training data. They combine their learned prior model with a standard data term [**38**] and find that the FoE model improves the accuracy of optical flow estimates. While their work provides a learned prior model of optical flow, it only models the spatial statistics of the optical flow. Neither the data term nor the relationship between flow boundaries and image edges is studied.

Freeman *et al.* [**57**] also learn an MRF model of image motion but their training is restricted to simplified "blob world" scenes. Ross and Kaelbling [**130, 131**] apply a similar learning method, but use segmentation from moving sequences to learn how to segment static objects. Scharstein and Pal [**142**] learn a full model of stereo, formulated as a Conditional Random Field (CRF), from training images with ground truth disparities. This model also combines spatial smoothness and brightness constancy in a learned model, but uses simple models of brightness constancy and spatially-modulated Potts models for spatial smoothness; these are likely inappropriate for optical flow. Li and Huttenlocher [**97**] learn the unknown parameters by minimizing the training loss using stochastic optimization. Their training data consist of three Middlebury training sequences with associated ground truth flow fields. Jia *et al.* [**80**] learn sparse models of optical flow to patch-wisely regularize the optical flow using the eight Middlebury training sequences.

Learning simple layered models is feasible but the learned models do not generalize to real-world videos [**81, 89**]. Richer hierarchical layered models require more advanced learning methods and more representative training data.

2.5. Benchmark Datasets, Evaluation, and Software

It is essential to have datasets that can quantitatively benchmark the performance of existing algorithms to ensure continued progress. For example, the recent developments in image segmentation [**108**], stereo matching [**143**], object detection and recognition [**50**], have been stimulated by the availability of a large amount of real data with human labeled ground truth. In the following, we review several representative datasets according to the date of their release.

2.5.1. Synthetic ("Yosemite"-like) Dataset.

Optical flow is one of the early subfields in computer vision to have a benchmark dataset. However, obtaining the ground truth flow field is more difficult than other vision tasks, such as detection and recognition. Hence most evaluations in the early days were performed with synthetic datasets, such as the popular "Yosemite" sequence.

Barron *et al.* [**19**] perform an evaluation of optical flow estimation methods in the early 1990's. Nearly all evaluated methods perform poorly on the synthetic "Yosemite" sequence, though it contains relatively simple motion, very few large motion discontinuities, and almost no occlusion, as shown in Figure 2.5. The best dense optical flow method [**152**] at that time had an Average Angular Error (AAE) larger than 10 degrees. Otte and Nagel [**123**] obtain ground truth for real scenes, but the scenes are extremely simple. McCane *et al.* [**109**] generate scenes with polyhedral objects and simple synthetic sequences to evaluate several flow estimation methods.

The work of Barron *et al.* is so influential that nearly all following work is compared on the "Yosemite" sequence. After more than ten years of intensive research efforts, the best dense two-frame methods [**38, 124, 133**] have AAE below 2 degrees, making it necessary to collect a new dataset with more challenging sequences.

2.5.2. Middlebury Dataset.

Baker *et al.* [**17, 16**] develop the Middlebury dataset to better benchmark the optical flow methods than the "Yosemite" sequence and stimulate new developments in the field. There are three types of sequences for evaluating the accuracy of optical flow estimation

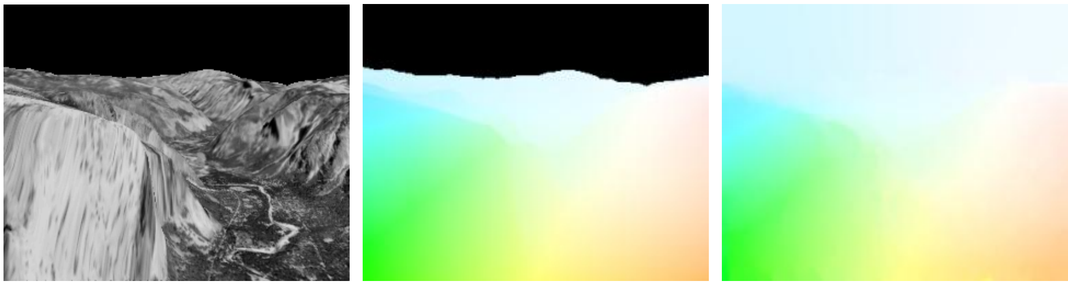


Figure 2.5. First frame of the "Yosemite" sequence, the ground truth flow field (the sky region has been excluded from the evaluation), and the estimated flow field by the combined local and global (CLG) method [**38**]. The result by the CLG method is very close to the ground truth (AAE less than 2 degrees).

methods: real, synthetic, and modified stereo. The real sequences were captured in a controlled lab environment; the scenes' surfaces were painted with a fluorescent paint and captured under both UV and normal lights. The ground truth was obtained by local matching on the UV texture images and then downsampled to have subpixel accuracy. The images taken under normal lights were downsampled to the same resolution. The real sequences tend to have strong lighting changes and shadows. The synthetic sequences were generated using computer graphics techniques and have large motion, big occlusion regions, and strong motion discontinuities. The modified stereo pairs are adapted from the Middlebury stereo benchmark [**143**] to give a comparison between stereo and optical flow methods. To reduce the influence of over-fitting, they further divided the dataset into training and test parts. The training part has ground truth flow fields while the test part only provides the input images.

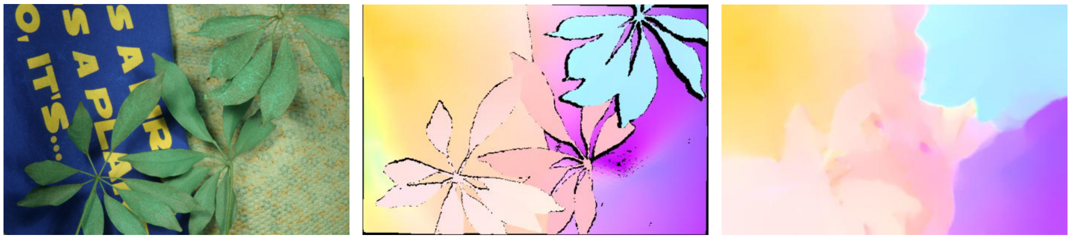


Figure 2.6. First frame of the "Schefflera" sequence from the Middlebury benchmark, the ground truth flow field (the dark regions correspond to occlusions and are excluded from evaluation), and the estimated flow field by the Combined Local and Global (CLG) method [**38**]. Though achieving good result on the "Yosemite" sequence (cf. Figure 2.5), The CLG method fails to recover the sharp motion boundaries and the occlusion regions.

As shown in Figure 2.6, the CLG method produces large errors in near sharp motion boundaries and in occlusion regions on the Middlebury sequences, though this highly-regarded method achieves good results on the "Yosemite" dataset. This means that the new dataset is more challenging. The difficulties with previous methods to deal with the Middlebury sequences have stimulated a rapid growth in optical flow estimation since the publication of the Middlebury dataset in 2007. The online benchmark website [**1**] has recorded the fast development.

2.5.3. MIT Layered Segmentation Dataset. Liu *et al.* [**99**] collect a dataset of outdoor sequences, manually segment the scene into layers, and estimate the motion field using the human segmentation (MIT dataset). As shown in Figure 2.7, the images are more natural than the Middlebury dataset. The reliance on a particular motion estimation method [**38**], however, hinders its applicability for evaluating flow algorithms.

In this thesis, we use the Middlebury training dataset to tune the parameters of the models for motion estimation, while testing the methods on both the Middlebury test and MIT datasets. We mainly use the MIT dataset to evaluate motion segmentation results. To alleviate the influence of over-fitting, we use separate training and test datasets and adopt datasets of different natures. We also generate some synthetic sequences to check the limitations of the proposed approaches in the ideal case.

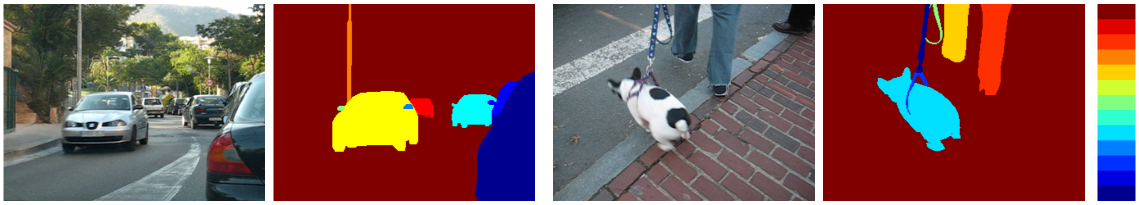


Figure 2.7. MIT dataset. Left/right for each pair: first frame/human labeled ground truth. on the far right is a color key for the depth ordering (blue is close and red is far)

2.5.4. Performance Measures. Two commonly used performance measures for optical flow are the Angular Error (AE) [**56, 19**] and the End-Point Error (EPE) [**123, 16**]. The AE between a flow vector (u, v) and the ground-truth flow $(u_{\mathrm{GT}}, v_{\mathrm{GT}})$ is the angle in 3D space between $(u, v, 1.0)$ and $(u_{\mathrm{GT}}, v_{\mathrm{GT}}, 1.0)$

$$\mathrm{AE} = \cos^{-1}\left(\frac{1.0 + u \times u_{\mathrm{GT}} + v \times v_{\mathrm{GT}}}{\sqrt{1 + u^2 + v^2}\sqrt{1 + u_{\mathrm{GT}}^2 + v_{\mathrm{GT}}^2}}\right). \tag{10}$$

The EP is the Euclidean distance between these two vectors

$$\mathrm{EP} = \sqrt{(u - u_{\mathrm{GT}})^2 + (v - v_{\mathrm{GT}})^2}. \tag{11}$$

One common measure for the segmentation is the RandIndex (RI) [**127**]. Given one segmentation $\mathbf{k}$ and the ground truth segmentation $\mathbf{k}_{\mathrm{GT}}$ of an image $\mathbf{I}$. Define

- a, the number of pairs of pixels in $\mathbf{I}$ that are in the same segment in $\mathbf{k}$ and in the same segment in $\mathbf{k}_{\mathrm{GT}}$
- b, the number of pairs of pixels in $\mathbf{I}$ that are in different segments in $\mathbf{k}$ and in different segments in $\mathbf{k}_{\mathrm{GT}}$
- c, the number of pairs of pixels in $\mathbf{I}$ that are in the same segment in $\mathbf{k}$ and in different segments in $\mathbf{k}_{\mathrm{GT}}$
- d, the number of pairs of pixels in $\mathbf{I}$ that are in different segments in $\mathbf{k}$ and in the same segment in $\mathbf{k}_{\mathrm{GT}}$

Intuitively, $a + b$ can be considered as the number of agreements between $\mathbf{k}$ and $\mathbf{k}_{\mathrm{GT}}$ and $c + d$ as the number of disagreements between $\mathbf{k}$ and $\mathbf{k}_{\mathrm{GT}}$. The RI for the segmentation $\mathbf{k}$ is

$$\mathrm{RI} = \frac{a+b}{a+b+c+d}. \tag{12}$$

2.5.5. Evaluation and Software. Hirschmuller and Scharstein [**70**] perform a quantitative evaluation of different features for stereo matching. Papenberg *et al.* [**124**] use the "Yosemite" sequence to study the effects of different parameters on the accuracy of their flow estimation method, such as the noise variance, the regularization weight, and the downsampling ratio.

The Middlebury website has a ranking for different methods based on the submitted results. However, there are many factors influencing the ranking of a method, such as model, optimization, and implementation details. The key that makes optical flow accurate is somehow confusing. Such difficulty arises because the performance of an algorithm depends on the objective function, the optimization method, and the implementation details. For example, the HS method has been widely regarded as inaccurate. One somewhat surprising result we find is that, the HS objective, when optimized with modern practices, produces competitive results. To precisely evaluate a factor, it is important to isolate other factors and test only the role of the selected one.

Despite recent algorithmic advances there is a lack of publicly available, easy to use, and accurate flow estimation software [**165**]. The GPU4Vision project [**2**] has made a substantial effort to change this and provides executable files for several accurate methods [**184, 185, 189**]. The dependence on the GPU and the lack of source code are limitations. At the time of the publication of our conference paper [**158**], we released our MATLAB codes. The public software has been used by both researchers to develop new flow estimation methods [**5, 80**] and practitioners to apply optical flow in their work [**118**].

CHAPTER 3

Learning Low-level Models of Optical Flow

In this chapter, we analyze the assumptions of standard optical flow formulations using training data and show how to extend the standard models to deal with violations of the assumptions. Assumptions of brightness constancy and spatial smoothness underlie most optical flow estimation methods. In contrast to standard heuristic formulations, we learn a statistical model of both brightness constancy error and the spatial properties of optical flow using image sequences with associated ground truth flow fields. The result is a complete probabilistic model of optical flow. Specifically, the ground truth enables us to model how the assumption of brightness constancy is violated in naturalistic sequences, resulting in a probabilistic model of “brightness inconstancy”. We also generalize previous high-order constancy assumptions, such as gradient constancy, by modeling the constancy of responses to various linear filters in a high-order random field framework. These filters are free variables that can be learned from training data. Additionally we study the spatial structure of the optical flow and how motion boundaries are related to image intensity boundaries. Spatial smoothness is modeled using a Steerable Random Field (SRF), where spatial derivatives of the optical flow are steered by the image brightness structure. These models provide a statistical motivation for previous methods and enable the learning of all parameters from training data. All proposed models are quantitatively compared on the Middlebury flow dataset.

3.1. Introduction

We address the problem of learning models of optical flow from training data. Optical flow estimation has a long history and we argue that most methods have explored some variation of the same theme. Particularly, most techniques exploit two constraints: brightness constancy and spatial smoothness. The brightness constancy constraint (data term) is derived from the observation that surfaces usually persist over time and hence the intensity value of a small region remains the same despite its position change [**26**]. The spatial smoothness constraint (spatial term) comes from the observation that neighboring pixels generally belong to the same surface and so have nearly the same image motion. Despite the long history, there have been very few attempts to *learn* what these terms should be [**133**]. Recent advances [**17**] have made sufficiently realistic image sequences with ground truth optical flow available to finally make this practical. Here we revisit several classic and recent optical flow methods and show how training data and machine learning methods can be used to train these models. We then go beyond previous formulations to define new versions of both the data and spatial terms.

We make two primary contributions. First we exploit image intensity boundaries to improve the accuracy of optical flow near motion boundaries. The idea is based on that of Nagel and Enkelmann [**117**], who introduce oriented smoothness to prevent blurring of flow boundaries across image boundaries; this can be regarded as an anisotropic diffusion approach. Here we go a step further and use training data to analyze and model the statistical relationship between image and flow boundaries. Specifically we use a Steerable Random Field (SRF) [**134**] to model the conditional statistical relationship between the flow and the image sequence. Typically, the spatial smoothness of optical flow is expressed in terms of the image-axis-aligned partial derivatives of the flow field. Instead, we use the local image edge orientation to define a *steered* coordinate system for the flow derivatives and note that the flow derivatives along and across image boundaries are highly kurtotic. We then model the flow field using a Markov Random Field (MRF) and formulate the steered potentials using Gaussian Scale Mixtures (GSM) [**175**]. All parameters of the model are learned from examples thus providing a rigorous statistical formulation of the idea of Nagel and Enkelmann.

Our second key contribution is to learn a statistical model of the data term. Numerous authors have addressed problems with the common brightness constancy assumption. Brox *et al.* [**34**], for example, extend brightness constancy to high-order constancy, such as gradient and Hessian constancy in order to minimize the effects of illumination change. Additionally, Bruhn *et al.* [**38**] show that integrating constraints within a local neighborhood improves the accuracy of dense optical flow. We generalize these two ideas and model the data term as a general high-order random field that allows the principled integration of local information. In particular, we extend the Field-of-Experts formulation (FOE) [**133**] o the spatio-temporal domain to model temporal changes in image features. The data term is formulated as the product of a number of experts, where each expert is a non-linear function (GSM) of a linear filter response. One can view previous methods as taking these filters to be fixed: Gaussians, first derivatives, second derivatives, etc. Rather than assuming known filters, our framework allows us to learn them from training data.

In summary, by using naturalistic training sequences with ground truth flow we are able to learn a complete model of optical flow that not only captures the spatial statistics of the flow field but also the statistics of brightness inconstancy and how the flow boundaries relate to the image intensity structure. The model combines and generalizes ideas from several previous methods and the resulting objective function is at once familiar and novel. We present a quantitative evaluation of the different methods using the Middlebury flow database [**17**] and find that the learned models outperform previous models, particularly at motion boundaries. Our analysis uses a single, simple, optimization method throughout to focus the comparison on the effects of different objective functions. The results suggest the benefit of learning standard models and open the possibility to learn more sophisticated ones.

3.2. Previous Work

Horn and Schunck [**73**] introduce both the brightness constancy and the spatial smoothness constraints for optical flow estimation, however their quadratic formulation assumes Gaussian statistics and is not robust to outliers caused by reflection, occlusion, motion boundaries etc. Black and Anandan [**26**] introduce a robust estimation framework to deal with such outliers, but do not attempt to *model* the true statistics of brightness constancy errors and flow derivatives. Fermüller *et al.* [**54**] analyze the effects of noise on the estimation of flow, but do not attempt to learn flow statistics from examples. Rather than assuming a model of brightness constancy we acknowledge that brightness can change and, instead, attempt to explicitly model the statistics of *brightness inconstancy*.

Many authors have extended the brightness constancy assumption, either by making it more physically plausible [**60, 67**] or by linear or non-linear pre-filtering of the images [**170**]. The idea of assuming constancy of first or second image derivatives to provide some invariance to lighting changes dates back to the early 1980's with the Laplacian pyramid [**6**] and has recently gained renewed popularity [**34**]. Following a related idea, Bruhn *et al.* [**38**] replace the pixelwise brightness constancy model with a spatially smoothed one. They find that a Gaussian-weighted spatial integration of brightness constraints results in significant improvements in flow accuracy. If filtering the image is a good idea, then we ask what filters should we choose? To address this question, we formulate the problem as one of learning the filters from training examples.

Most optical flow estimation methods encounter problems at motion boundaries where the assumption of spatial smoothness is violated. Observing that flow boundaries often coincide with image boundaries, Nagel and Enkelmann [**117**] introduce oriented smoothness to prevent blurring of optical flow across image boundaries. Alvarez *et al.* [**7**] modify the Nagel-Enkelmann approach so that less smoothing is performed close to image boundaries. The amount of smoothing along and across boundaries has been determined heuristically. Fleet *et al.* [**55**] learn a statistical model relating image edge orientation and amplitude to flow boundaries in the context of a patch-based motion discontinuity model. Black [**29**] proposes an MRF model that couples edges in the flow field with edges in the brightness images. This model, however, is hand designed and tuned. We provide a probabilistic framework within which to learn the parameters of a model like that of Nagel and Enkelmann from examples.

Simoncelli *et al.* [**151**] formulate an early probabilistic model of optical flow and modeled the statistics of the deviation of the estimated flow from the true flow. Black *et al.* [**28**] learn parametric models for different classes of flow (e.g. edges and bars). More recently, Roth and Black [**133**] model the spatial structure of optical flow fields using a high-order MRF, called a Field of Experts (FoE), and learn the parameters from training data. They combine their learned prior model with a standard data term [**38**] and find that the FoE model improves the accuracy of optical flow estimates. While their work provides a learned prior model of optical flow, it only models the spatial statistics of the optical flow and not the data term or the relationship between flow and image brightness.

Freeman *et al.* [**57**] also learn an MRF model of image motion but their training is restricted to simplified "blob world" scenes; here we use realistic scenes with more complex image and flow structure. Scharstein and Pal [**142**] learn a full model of stereo, formulated as a conditional random field (CRF), from training images with ground truth disparity. This model also combines spatial smoothness and brightness constancy in a learned model, but uses simple models of brightness constancy and spatially-modulated Potts models for spatial smoothness; these are likely inappropriate for optical flow.

3.3. Statistics of Optical Flow

3.3.1. Spatial Term. Roth and Black [**133**] study the statistics of horizontal and vertical optical flow derivatives and found them to be heavy-tailed, which supports the intuition that optical flow fields are typically smooth, but have occasional motion discontinuities. Figure 3.1 (a, b (solid)) shows the marginal log-histograms of the horizontal and vertical derivatives of horizontal flow, computed from a set of 45 ground truth optical flow fields. These include four from the Middlebury "other" dataset, one from the "Yosemite" sequence, and ten of our own synthetic sequences. These synthetic sequences were generated in the same way as, and are similar to, the other Middlebury synthetic sequences (Urban and Grove); two examples are shown in Figure 3.3. To generate additional training data the sequences were also flipped horizontally and vertically. The histograms are heavy-tailed with high peaks, as characterized by their high kurtosis ($\kappa = E[(x-\mu)^4]/E[(x-\mu)^2]^2$).

We go beyond previous work by also studying the steered derivatives of optical flow where the steering is obtained from the image brightness of the reference (first) frame. To obtain the steered derivatives, we first calculate the local image orientation in the reference frame using the structure tensor as described in [**134**]. Let $(\cos\theta(\mathbf{I}), \sin\theta(\mathbf{I}))^T$ and $(-\sin\theta(\mathbf{I}), \cos\theta(\mathbf{I}))^T$ be the eigenvectors of the structure tensor in the reference frame $\mathbf{I}$, which are respectively orthogonal to and aligned with the local image orientation. Then the orthogonal and aligned derivative operators $\partial_O^{\mathbf{I}}$ and $\partial_A^{\mathbf{I}}$ of the optical flow are given by

$$\partial_O^{\mathbf{I}} = \cos\theta(\mathbf{I}) \cdot \partial_x + \sin\theta(\mathbf{I}) \cdot \partial_y \quad \text{and} \quad \partial_A^{\mathbf{I}} = -\sin\theta(\mathbf{I}) \cdot \partial_x + \cos\theta(\mathbf{I}) \cdot \partial_y, \tag{13}$$

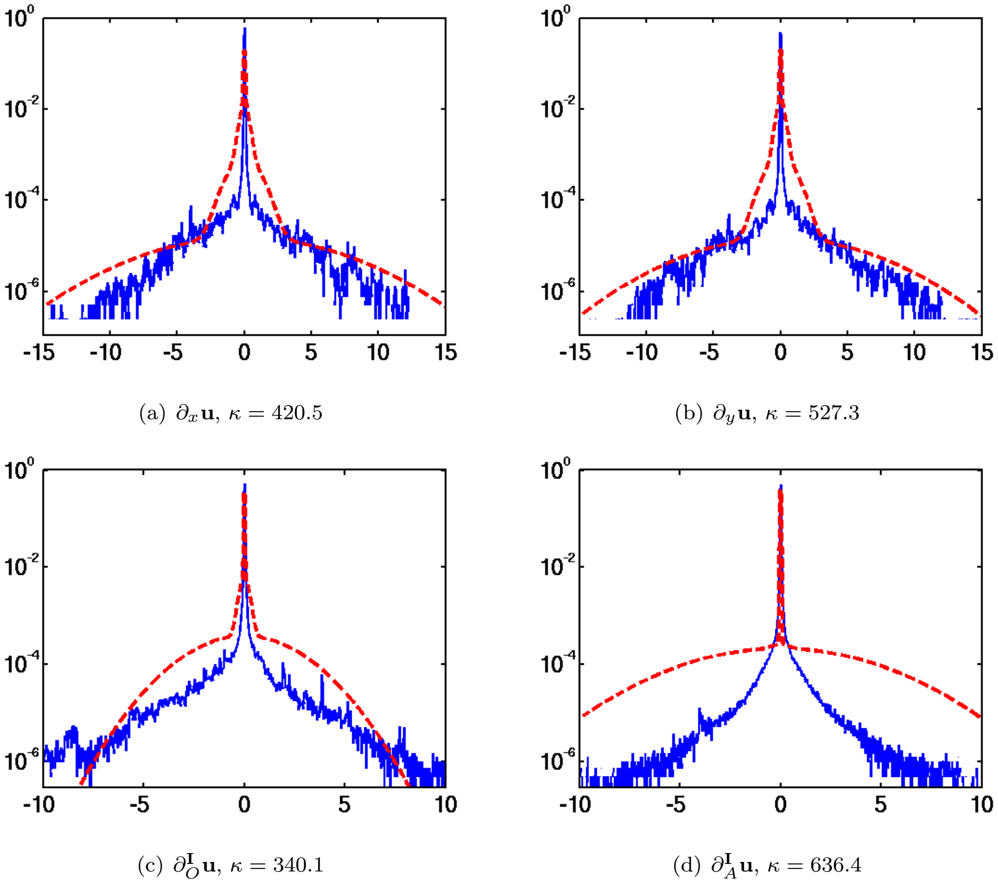
where ∂_x and ∂_y are the horizontal and vertical derivative operators. We approximate these using the 2×3 and 3×2 filters from [**134**].

Figure 3.1 (c, d) shows the marginal log-histograms of the steered derivatives of the horizontal flow (the vertical flow statistics are similar and are omitted here). The log-histogram of the derivative orthogonal to the local structure orientation has much broader tails than the aligned one, which confirms the intuition that large flow changes occur more frequently across the *image* edges.

These findings suggest that the steered marginal statistics provide a statistical motivation for the Nagel-Enkelmann method, which performs stronger smoothing along image edges and less orthogonal to image edges. Furthermore, the non-Gaussian nature of the histograms suggest that non-linear smoothing should be applied orthogonal to *and* aligned with the image edges.

3.3.2. Data Term. To our knowledge, there has been no formal study of the statistics of the brightness constancy error, mainly due to the lack of appropriate training data. Using ground truth optical flow fields we compute the brightness difference between pairs of training images by warping the second image in each pair toward the first using bi-linear interpolation. Figure 3.2 shows the marginal log-histogram of the brightness constancy error for the training set; this has heavier tails and a tighter peak than a Gaussian of the same mean and variance. The tight peak suggests that the value of a pixel in the first image is usually nearly the same as the corresponding value in the second image, while the heavy tails account for violations caused by reflection, occlusion, transparency, etc. This shows that modeling the brightness constancy error with a Gaussian, as has often been done, is inappropriate, and this also provides a statistical explanation for the robust data term used by Black and Anandan [**26**]. The Lorentzian used there has a similar shape as the empirical histogram in Figure 3.2.

We should also note that the shape of the error histogram will depend on the type of training images. For example, if the images have significant camera noise, this will lead to brightness changes even in the absence of any other effects. In such a case, the error histogram will have a more rounded



(a) $\partial_x \mathbf{u}$, $\kappa = 420.5$

(b) $\partial_y \mathbf{u}$, $\kappa = 527.3$

(c) $\partial^{\mathbf{I}}_{O} \mathbf{u}$, $\kappa = 340.1$

(d) $\partial^{\mathbf{I}}_{A} \mathbf{u}$, $\kappa = 636.4$

Figure 3.1. Marginal filter response statistics (log scale) of standard derivatives (left) and derivatives steered to local image structure (right) for the horizontal flow $\mathbf{u}$. The histograms are shown in solid blue; the learned experts in dashed red. κ denotes kurtosis.

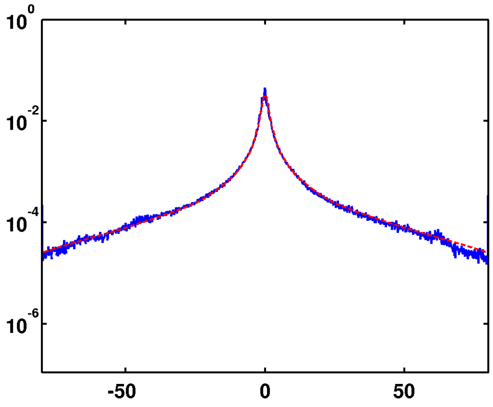


Figure 3.2. Statistics of the brightness constancy error: The log-histogram (solid blue) is fit with a GSM model (dashed red).

peak depending on how much noise is present in the images. Future work should investigate adapting the data term to the statistical properties of individual sequences.

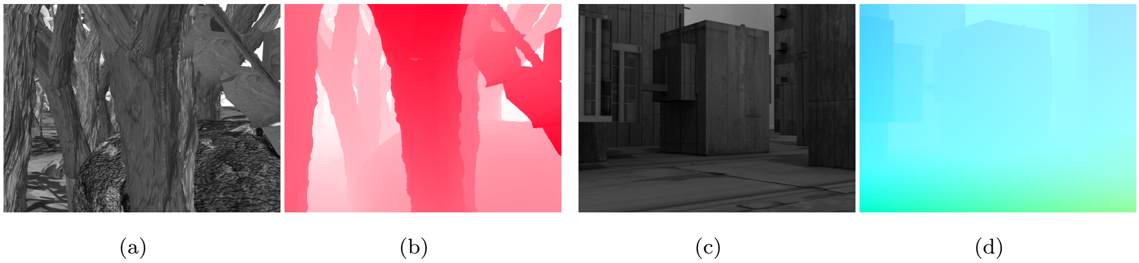

(a) (b) (c) (d)

Figure 3.3. (a)-(d) two reference (first) images and their associated flow fields from our synthetic training set.

3.4. Modeling Optical Flow

We formulate optical flow estimation as a problem of probabilistic inference and decompose the posterior probability density of the flow field $(\mathbf{u}, \mathbf{v})$ given two successive input images $\mathbf{I}_1$ and $\mathbf{I}_2$ as

$$p(\mathbf{u}, \mathbf{v} | \mathbf{I}_1, \mathbf{I}_2; \Omega) \propto p(\mathbf{I}_2 | \mathbf{u}, \mathbf{v}, \mathbf{I}_1; \Omega_{\mathrm{D}}) \cdot p(\mathbf{u}, \mathbf{v} | \mathbf{I}_1; \Omega_{\mathrm{S}}), \tag{14}$$

where Ω_{D} and Ω_{S} are parameters of the model. Here the first (data) term describes how the second image $\mathbf{I}_2$ is generated from the first image $\mathbf{I}_1$ and the flow field, while the second (spatial) term encodes our prior knowledge of the flow fields given the first (reference) image. Note that this decomposition of the posterior is slightly different from the typical one, *e.g.,*in [**151**], in which the spatial term takes the form $p(\mathbf{u}, \mathbf{v}; \Omega_{\mathrm{S}})$. Standard approaches assume conditional independence between the flow field and the image structure, which is typically not made explicit. The advantage our formulation is that the conditional nature of the spatial term allows for more flexible methods of flow regularization.

3.4.1. Spatial Term. For simplicity we assume that horizontal and vertical flow fields are independent; Roth and Black [**133**] experimentally show that this is a reasonable assumption. The spatial model thus becomes

$$p(\mathbf{u}, \mathbf{v} | \mathbf{I}_1; \Omega_{\mathrm{S}}) = p(\mathbf{u} | \mathbf{I}_1; \Omega_{\mathrm{Su}}) \cdot p(\mathbf{v} | \mathbf{I}_1; \Omega_{\mathrm{Sv}}). \tag{15}$$

To obtain our first model of spatial smoothness, we assume that the flow fields are independent of the reference image. Then the spatial term reduces to a classical optical flow prior, which can, for example, be modeled using a pairwise MRF:

$$p_{\mathrm{PW}}(\mathbf{u}; \Omega_{\mathrm{PWu}}) = \frac{1}{Z(\Omega_{\mathrm{PWu}})} \prod_{p} \prod_{q \in \Gamma_p} \phi(u^p - u^q; \Omega_{\mathrm{PWu}}), \tag{16}$$

where $p = (i, j)$ is the pixel index and the set Γ_p contains the four nearest neighbors of p. Note that the difference between the flow at neighboring pixels approximates the horizontal and vertical image derivatives (see *e.g.,*[**26**]). $Z(\Omega_{\mathrm{PWu}})$ here is the partition function that ensures normalization. Note that although such an MRF model is based on products of very local potential functions, it provides a global probabilistic model of the flow. Various parametric forms have been used to model the potential function ϕ (or its negative log): Horn and Schunck [**73**] use Gaussians, the Lorentzian robust error function is used by Black and Anandan [**26**], and Bruhn *et al.* [**38**] assume the Charbonnier error function. In this paper, we use the more expressive Gaussian Scale Mixture (GSM) model [**175**], i.e.,

$$\phi(x;\Omega) = \sum_{l=1}^{L} \omega_l \cdot \mathcal{N}(x;0,\sigma^2/s_l), \tag{17}$$

in which $\Omega = \{\omega_l | l = 1, \ldots, L\}$ are the weights of the GSM model, s_l are the scales of the mixture components, and σ^2 is a global variance parameter. GSMs can model a wide range of distributions ranging from Gaussians to heavy-tailed ones. Here, the scales and σ^2 are chosen so that the empirical marginals of the flow derivatives can be represented well with such a GSM model and are not trained along with the mixture weights ω_l.

The particular decomposition of the posterior used here (14) allows us to model the spatial term for the flow conditioned on the measured image. For example, we can capture the oriented smoothness of the flow fields and generalize the Steerable Random Field (SRF) model [**134**] to a steerable model of optical flow, resulting in our second model of spatial smoothness:

$$p_{\mathrm{SRF}}(\mathbf{u}|\mathbf{I}_1;\Omega_{SRFu}) \propto \prod_p \phi\left((\partial_O^{\mathbf{I}_1} u)^p;\Omega_{SRFu}\right) \cdot \phi\left((\partial_A^{\mathbf{I}_1} u)^p;\Omega_{SRFu}\right). \tag{18}$$

The steered derivatives (orthogonal and aligned) are defined as in (13); the superscript denotes that steering is determined by the reference frame $\mathbf{I}_1$. The potential functions are again modeled using GSMs.

3.4.2. Data Term. Models of the optical flow data term typically embody the brightness constancy assumption, or more specifically model the deviations from brightness constancy. Assuming independence of the brightness error at the pixel sites, we can define a standard data term as

$$p_{\mathrm{BC}}(\mathbf{I}_2|\mathbf{u},\mathbf{v},\mathbf{I}_1;\Omega_{\mathrm{BC}}) \propto \prod_p \prod_{q\in\mathcal{N}_p} \phi(I_1^p - I_2^q);\Omega_{\mathrm{BC}}), \tag{19}$$

where the set $\mathcal{N}_p = \{i + u^p, j + v^p\}$ contains the corresponding pixel of the pixel p at the next frame according to the flow field. As with the spatial term, various functional forms (Gaussian, robust, etc.) have been assumed for the potential ϕ or its negative log. We again employ a GSM representation for the potential, where the scales and global variance are determined empirically before training the model (mixture weights).

Brox *et al.* [**34**] extend the brightness constancy assumption to include high-order constancy assumptions, such as gradient constancy, which may improve accuracy in the presence of changing scene illumination or shadows. We propose a further generalization of these constancy assumptions and model the constancy of responses to several general linear filters:

$$p_{\mathrm{FC}}(\mathbf{I}_2|\mathbf{u},\mathbf{v},\mathbf{I}_1;\Omega_{\mathrm{FC}}) \propto \prod_p \prod_{q\in\mathcal{N}_p} \prod_k \phi_k\{(J_{k1} * I_1)^p - (J_{k2} * I_2)^q;\Omega_{\mathrm{FC}}\}, \tag{20}$$

where the J_{k1} and J_{k2} are linear filters. Practically, this equation implies that the second image is first filtered with J_{k2}, after which the filter responses are warped toward the first filtered image using the flow $(\mathbf{u},\mathbf{v})$[1]. Note that this data term is a generalization of the Fields-of-Experts model (FoE),

[1] It is, in principle, also possible to formulate a similar model that warps the image first and then applies filters to the warped image. We did not pursue this option, as it would require the application of the filters at each iteration of the flow estimation procedure. Filtering before warping ensures that we only have to filter the image once before flow estimation.

which has been used to model prior distributions of images [**132**] and optical flow [**133**]. Here, we generalize it to a spatio-temporal model that describes brightness (in)constancy.

If we choose J_{11} to be the identity filter and define $J_{12} = J_{11}$, this implements brightness constancy. Choosing the J_{k1} to be derivative filters and setting $J_{k2} = J_{k1}$ allows us to model gradient constancy. Thus this model generalizes the approach by Brox *et al.* [**34**] [2]. If we choose J_{k1} to be a Gaussian smoothing filter and define $J_{k2} = J_{k1}$, we essentially perform pre-filtering as, for example, suggested by Bruhn *et al.* [**38**]. Even if we assume fixed filters using a combination of the above, our probabilistic formulation still allows learning the parameters of the GSM experts from data as outlined below. Consequently, we do not need to tune the trade-off weights between the brightness and gradient constancy terms by hand as in [**34**]. Beyond this, the appeal of using a model related to the FoE is that we do not have to fix the filters ahead of time, but instead we can learn these filters alongside the potential functions.

3.4.3. Learning. Our formulation enables us to train the data term and the spatial term separately, which simplifies learning. Note though, that it is also possible to turn the model into a conditional random field (CRF) and employ conditional likelihood maximization (cf. [**155**]); we leave this for future work. To train the pairwise spatial term $p_{\text{PW}}(\mathbf{u}; \Omega_{\text{PWu}})$, we can estimate the weights of the GSM model by either simply fitting the potentials to the empirical marginals using expectation maximization, or by using a more rigorous learning procedure, such as maximum likelihood (ML). To find the ML parameter estimate we aim to maximize the log-likelihood $\mathcal{L}_{\text{PW}}(\mathcal{U}; \Omega_{\text{PWu}})$ of the horizontal flow components $\mathcal{U} = \{\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(t)}\}$ of the training sequences w. r. t. the model parameters Ω_{PWu} (i.e., GSM mixture weights). Analogously, we maximize the log-likelihood of the vertical components $\mathcal{V} = \{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(t)}\}$ w. r. t. Ω_{PWv}. Because ML estimation in loopy graphs is generally intractable, we approximate the learning objective and use the contrastive divergence (CD) algorithm [**69**] to learn the parameters.

To train the steerable flow model $p_{\text{SRF}}(\mathbf{u}|\mathbf{I}_1; \Omega_{\text{SRF}})$ we aim to maximize the conditional log-likelihoods $\mathcal{L}_{\text{SRF}}(\mathcal{U}|\mathcal{I}_1; \Omega_{\text{SRFu}})$ and $\mathcal{L}_{\text{SRF}}(\mathcal{V}|\mathcal{I}_1; \Omega_{\text{SRFv}})$ of the training flow fields given the first (reference) images $\mathcal{I}_1 = \{\mathbf{I}_1^{(1)}, \ldots, \mathbf{I}_1^{(t)}\}$ from the training image pairs w. r. t. the model parameters Ω_{SRFu} and Ω_{SRFv}.

To train the simple data term $p_D(\mathbf{I}_2|\mathbf{u}, \mathbf{v}, \mathbf{I}_1; \Omega_D)$ modeling brightness constancy, we can simply fit the marginals of the brightness violations using the expectation maximization (EM) algorithm [**23, 48**]. This is possible, because the model assumes independence of the brightness error at the pixel sites. For the proposed generalized data term $p_{\text{FC}}(\mathbf{I}_2|\mathbf{u}, \mathbf{v}, \mathbf{I}_1; \Omega_{\text{FC}})$ that models filter response constancy, a more complex training procedure is necessary, since the filter responses are not independent. Ideally, we would maximize the conditional likelihood $\mathcal{L}_{\text{FC}}(\mathcal{I}_2|\mathcal{U}, \mathcal{V}, \mathcal{I}_1; \Omega_{\text{FC}})$ of the training set of the second images $\mathcal{I}_2 = \{\mathbf{I}_2^{(1)}, \ldots, \mathbf{I}_2^{(t)}\}$ given the training flow fields and the first images. Due to the intractability of ML estimation in these models, we use a conditional version of contrastive divergence (see *e.g.,*[**134, 155**]) to learn both the mixture weights of the GSM potentials as well as the filters.

[2] Formally, there is a minor difference: [**34**] penalizes changes in the gradient magnitude, while the proposed model penalizes changes of the flow derivatives. These are, however, equivalent in the case of Gaussian potentials.

3.5. Optical Flow Estimation

Given two input images, we estimate the optical flow between them by maximizing the posterior from (14). Equivalently, we minimize its negative log

$$E(\mathbf{u}, \mathbf{v}) = E_{\mathrm{D}}(\mathbf{u}, \mathbf{v}) + \lambda E_{\mathrm{S}}(\mathbf{u}, \mathbf{v}), \tag{21}$$

where E_{D} is the negative log (i.e., energy) of the data term, E_{S} is the negative log of the spatial term (the normalization constant is omitted in either case), and λ is an optional trade-off weight (or regularization parameter).

Optimizing such energies is generally difficult, because of their non-convexity and many local optima. The non-convexity in our approach stems from the fact that the learned potentials are non-convex and from the warping-based data term used here and in other competitive methods [**34**]. To limit the influence of spurious local optima, we construct a series of energy functions

$$E_{\mathrm{C}}(\mathbf{u}, \mathbf{v}, \alpha) = \alpha E_{\mathrm{Q}}(\mathbf{u}, \mathbf{v}) + (1 - \alpha) E(\mathbf{u}, \mathbf{v}), \tag{22}$$

where E_{Q} is a quadratic, convex, formulation of E that replaces the potential functions of E by a quadratic form and uses a different λ. Note that E_{Q} amounts to a Gaussian MRF formulation. $\alpha \in [0, 1]$ is a control parameter that varies the convexity of the compound objective. As α changes from 1 to 0, the combined energy function in (22) changes from the quadratic formulation to the proposed non-convex one (cf. [**31**]). During the process, the solution at a previous convexification stage serves as the starting point for the current stage. In practice, we find using three stages produces reasonable results.

At each stage, we perform a simple local minimization of the energy. At a local minimum, it holds that

$$\nabla_{\mathbf{u}} E_{\mathrm{C}}(\mathbf{u}, \mathbf{v}, \alpha) = 0, \tag{23}$$

and

$$\nabla_{\mathbf{v}} E_{\mathrm{C}}(\mathbf{u}, \mathbf{v}, \alpha) = 0. \tag{24}$$

Since the energy induced by the proposed MRF formulation is spatially discrete, it is relatively straightforward to derive the gradient expressions, similar to the derivation in Chapter 2. Setting these to zero and linearizing them, we rearrange the results into a system of linear equations, which can be solved by a standard technique. The main difficulty in deriving the linearized gradient expressions is the linearization of the warping step. For this we follow the approach of Brox *et al.* [**34**] while using the derivative filters proposed in [**38**].

To estimate flow fields with large displacements, we adopt an incremental multi-resolution technique (*e.g.* [**26, 38**]). As is quite standard, the optical flow estimated at a coarser level is used to warp the second image toward the first at the next finer level and the flow increment is calculated between the first image and the warped second image. The final result combines all the flow increments. At the first stage where $\alpha = 1$, we use a 4-level pyramid with a downsampling factor of 0.5. At other stages, we only use a 2-level pyramid with a downsampling factor of 0.8 to make full use of the solution at the previous convexification stage.

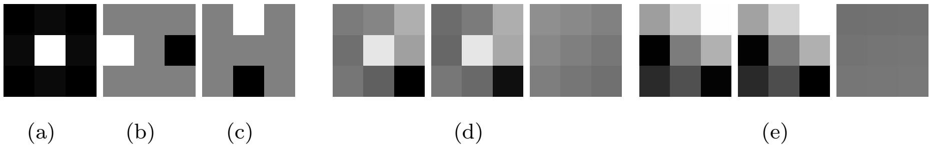


Figure 3.4. Three fixed filters from the FFC model: (a) Gaussian, (b) horizontal derivative, and (c) vertical derivative. (d,e) Two of the six learned filter pairs of the LFC model and the difference between each pair (left: J_{k1}, middle: J_{k2}, right: $J_{k1} - J_{k2}$).

3.6. Experiments and Results

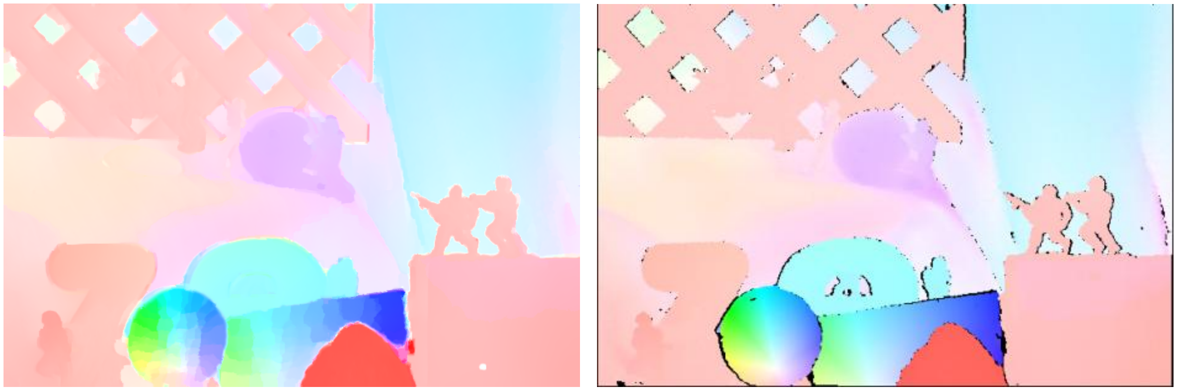
3.6.1. Learned Models. The spatial terms of both the pairwise model (**PW**) and the steerable model (**SRF**) were trained using contrastive divergence on $20,000$ 9×9 flow patches that were randomly cropped from the training flow fields (see above). To train the steerable model, we also supplied the corresponding $20,000$ image patches (of size 15×15 to allow computing the structure tensor) from the reference images. The pairwise model used 5 GSM scales; and the steerable model 4 scales.

The simple brightness constancy data term (**BC**) was trained using expectation-maximization. To train the data term that models the generalized filter response constancy (**FC**), the CD algorithm was run on $20,000$ 15×15 flow patches and corresponding 25×25 image patches, which were randomly cropped from the training data. 6-scale GSM models were used for both data terms. We investigated two different filter constancy models. The first (**FFC**) used 3 fixed 3×3 filters: a small variance Gaussian ($\sigma = 0.4$), and horizontal and vertical derivative filters similar to [**34**]. The other (**LFC**) used 6 3×3 filter pairs that were learned automatically. Note that the GSM potentials were learned in either case. Figure 3.4 shows the fixed filters from the FFC model, as well as two of the learned filters from the LFC model. Interestingly, the learned filters do not look like ordinary derivative filters nor do they resemble the filters learned in an FoE model of natural images [**132**]. It is also noteworthy that even though the J_{k2} are not enforced to be equal to the J_{k1} during learning, they typically exhibit only subtle differences as Figure 3.4 shows.

Given the non-convex nature of the learning objective, contrastive divergence is prone to finding local optima, which means that the learned filters are likely not optimal. Repeated initializations produced different-looking filters, which however performed similarly to the ones shown here. The fact that these "non-standard" filters perform better (see below) than standard ones suggests that more research on better filters for formulating optical flow data terms is warranted.

For the models for which we employed contrastive divergence, we used a hybrid Monte Carlo sampler with 30 leaps, $l = 1$ CD step, and a learning rate of 0.01 as proposed by [**134**]. The CD algorithm was run for 2000 to 10000 iterations, depending on the complexity of the model, after which the model parameters did not change significantly. Figure 3.1 shows the learned potential functions alongside the empirical marginals. We should note that learned potentials and marginals generally differ. This has, for example, been noted by Zhu *et al.* [**203**], and is particularly the case for the SRFs, since the derivative responses are not independent within a flow field (cf. [**134**]).

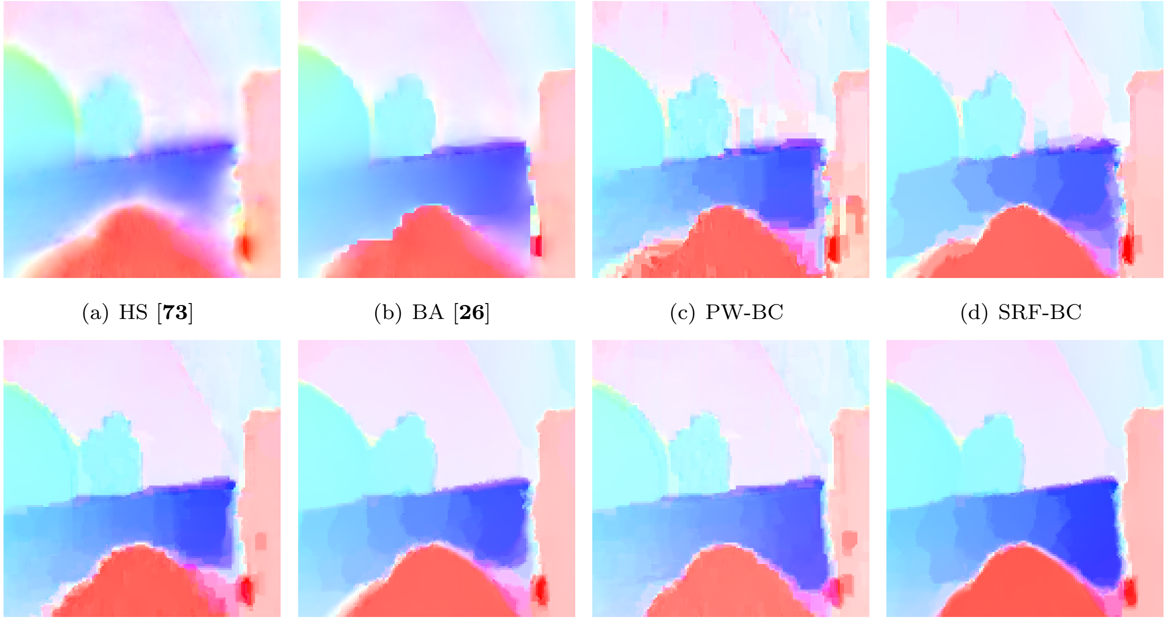
To estimate the flow, we proceeded as described in Section 3.5 and performed 3 iterations of the incremental estimation at each level of the pyramid. The regularization parameter λ was optimized



(a) Estimated flow

(b) Ground truth

Figure 3.5. Results of the SRF-LFC model for the "Army" sequence.



(a) HS [**73**]

(b) BA [**26**]

(c) PW-BC

(d) SRF-BC

(e) PW-FFC

(f) SRF-FFC

(g) PW-LFC

(h) SRF-LFC

Figure 3.6. Details of the flow results for the "Army" sequence. HS=Horn & Schunck; BA=Black & Anandan; PW=pairwise; SRF=steered model; BC=brightness constancy; FFC=fixed filter response constancy; LFC=learned filter response constancy.

for each method using a small set of training sequences. For this stage we added a small amount of noise to the synthetic training sequences, which led to larger λ values and increased robustness to novel test data.

3.6.2. Flow Estimation Results. We evaluated all 6 proposed models using the test portion of the Middlebury optical flow benchmark [**17**][3]. Figure 3.5 shows the results on one of the sequences along with the ground truth flow. Table 3.1 gives the average angular error (AAE) of the models on

[3] Note that the Yosemite frames used for testing as part of the benchmark are not the same as those used for learning.

Table 3.1. Average angular error (AAE) on the Middlebury optical flow benchmark for various combinations of the proposed models. The ranking information was at the writing of our conference paper [**159**] (August 2008); the AAE rank for SRF-LFC is 51.0 at the writing of this dissertation (July 2012).

	Rank	**Average**	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
HS[**73**]	18.2	8.72	8.01	9.13	14.20	12.40	4.64	8.21	4.01	9.16
BA[**26**]	12.1	7.17	6.81	8.77	13.00	8.29	4.18	6.19	3.63	6.45
PW-BC	13.0	7.37	6.75	10.20	14.00	7.31	4.19	6.08	3.05	7.39
SRF-BC	12.4	7.43	6.60	10.70	14.00	7.11	4.15	5.75	3.85	7.27
PW-FFC	12.3	5.96	4.29	4.81	8.95	7.60	5.33	6.45	2.93	7.34
SRF-FFC	10.5	5.79	4.29	5.26	8.53	7.02	5.01	6.12	3.38	6.67
PW-LFC	11.0	5.47	4.29	3.97	6.57	6.78	4.92	6.81	2.90	7.53
SRF-LFC	9.5	5.34	4.23	4.16	6.78	6.41	4.66	6.33	3.09	7.08

Table 3.2. Average angular error (AAE) in motion boundary regions.

	Average	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
PW-BC	15.81	14.00	19.90	23.80	25.60	5.33	19.60	4.35	13.90
SRF-BC	15.47	13.40	19.80	23.20	25.20	5.27	18.20	5.19	13.50
PW-FFC	15.37	12.10	16.70	19.70	26.30	6.73	21.60	4.45	15.40
SRF-FFC	14.74	11.50	16.60	18.90	25.40	6.32	20.60	4.83	13.80
PW-LFC	14.97	12.20	15.40	17.00	26.00	6.43	22.20	4.65	15.90
SRF-LFC	14.47	11.70	15.30	16.90	25.30	6.01	20.70	4.84	15.00

the test sequences, as well as the results of two standard methods [**26, 73**]. Note that the standard objectives from [**26, 73**] were optimized using exactly the same optimization strategy as used for the learned models. This ensures fair comparison and focuses the evaluation on the model rather than the optimization method. The table also shows the average rank from the Middlebury flow benchmark, as well as the average AAE across all 8 test sequences. Table 3.2 shows results of the same experiments, but here the AAE is only measured near motion boundaries. From these results we can see that the steerable flow model (SRF) substantially outperforms a standard pairwise spatial term (PW), particularly also near motion discontinuities. This holds no matter what data term the respective spatial term is combined with. This can also be seen visually in Figure 3.6, where the SRF results exhibit the clearest motion boundaries.

Among the different data terms, the filter response constancy models (FFC & LFC) very clearly outperform the classical brightness constancy model (BC), particularly on the sequences with real images ("Army" through "Schefflera"), which are especially difficult for standard techniques, because the classical brightness constancy assumption does not appear to be as appropriate as for the synthetic sequences, for example because of stronger shadows. Moreover, the model with learned filters (LFC) slightly outperforms the model with fixed, standard filters (FFC), particularly in regions with strong brightness changes. This means that learning the filters seems to be fruitful, particularly for challenging, realistic sequences. Further results, including comparisons to other recent techniques are available at *http://vision.middlebury.edu/flow/*.

3.7. Conclusions and Discussions

Enabled by a database of image sequences with ground truth optical flow fields, we have studied the statistics of both optical flow *and* brightness constancy, and formulated a fully learned probabilistic model for optical flow estimation. We extend our initial formulation by modeling the steered derivatives of optical flow, and generalize the data term to model the constancy of linear filter responses. This provides a statistical grounding for, and extension of, various previous models of optical flow, and at the same time enables us to learn all model parameters automatically from training data. Quantitative experiments show that both the steered model of flow as well as the generalized data term substantially improved performance.

The estimated flow fields by the learning approach are still not satisfactory. Compared with the ground truth, the estimated flow fields have over-blurred motion boundaries and large errors in occlusion regions. There are several reasons. First, we have focused on the models (objective functions) and adopted a simple optimization scheme. Better optimization methods and implementation details are likely to further improve the performance. Second, the low-level formulation does not model occlusions, where large errors tend to occur. Third, the training data may not be representative enough. The majority of the training sequences are synthetic, while the majority of the test sequences are real. More representative training data will make the learned models more robust. Lastly, there is a mismatch between the learning method and the inference scheme. As shown for image denoising [**144**], the learned potentials by the contrastive divergence algorithm for natural images produce inferior results to the hand-designed L1 norm using the MAP estimator. The MMSE estimator with the learned potentials however produces the optimal performance.

We will further study and address first two issues in the rest of the thesis. For example, the next chapter performs a more thorough analysis of the optical flow methods, according to models, optimization methods, and implementation details.

CHAPTER 4

A Quantitative Analysis of Recent Practices in Optical Flow Estimation and Their Principles

The accuracy of optical flow estimation algorithms has been improving steadily as evidenced by results on the Middlebury optical flow benchmark. The typical formulation, however, has changed little since the work of Horn and Schunck. In this chapter, we attempt to uncover what has made recent advances possible through a thorough analysis of how the objective function, the optimization method, and modern implementation practices influence accuracy. We discover that "classical" flow formulations perform surprisingly well when combined with modern optimization and implementation techniques. Moreover, we find that while median filtering of intermediate flow fields during optimization is a key to recent performance gains, it leads to higher energy solutions. To understand the principles behind this phenomenon, we derive a new objective that formalizes the median filtering heuristic. This objective includes a non-local term that robustly integrates flow estimates over large spatial neighborhoods. By modifying this new term to include information about flow and image boundaries we develop a method that can better preserve motion details.

4.1. Introduction

The field of optical flow estimation is making steady progress as evidenced by the increasing accuracy of current methods on the Middlebury optical flow benchmark [**17**]. After nearly 30 years of research, these methods have obtained an impressive level of reliability and accuracy [**184, 185, 189, 204**]. *But what has led to this progress?* The majority of today's methods strongly resemble the original formulation of Horn and Schunck (HS) [**73**]. They combine a data term that assumes constancy of some image property with a spatial term that models how the flow is expected to vary across the image. An objective function combining these two terms is then optimized. Given that this basic structure is unchanged since HS, what has enabled the performance gains of modern approaches?

The chapter has three parts. In the first, we perform an extensive study of current optical flow methods and models. The most accurate methods on the Middlebury flow dataset make different choices about how to model the objective function, how to approximate this model to make it computationally tractable, and how to optimize it. Since most published methods change *all* of these properties at once, it can be difficult to know which choices are most important. To address this, we define a baseline algorithm that is "classical", in that it is a direct descendant of the original HS formulation, and then systematically vary the model and method using different techniques from the art. The results are surprising. We find that only a small number of key choices produce statistically significant improvements and that they can be combined into a very simple method that achieves accuracies near the state of the art. More importantly, our analysis reveals what makes current flow methods work so well.

Part two examines the *principles* behind this success. We find that one algorithmic choice produces the most significant improvements: applying a median filter to intermediate flow values during incremental estimation and warping [**184, 185**]. While this heuristic improves the accuracy of the recovered flow fields, it actually *increases* the energy of the objective function. This suggests that what is being optimized is actually a new and different objective. Using observations about median filtering and L1 energy minimization from Li and Osher [**96**], we formulate a new *non-local term* that is added to the original, classical objective. This new term goes beyond standard local (pairwise) smoothness to robustly integrate information over large spatial neighborhoods. We show that minimizing this new energy approximates the original optimization with the heuristic median filtering step. Note, however, that the new objective falls outside our definition of classical methods.

Finally, once the median filtering heuristic is formulated as a non-local term in the objective, we immediately recognize how to modify and improve it. In part three we show how information about image structure and flow boundaries can be incorporated into a weighted version of the non-local term to prevent over-smoothing across boundaries. By incorporating structure from the image, this weighted version does not suffer from some of the errors produced by median filtering. At the time of writing (March 2010) of our conference paper [**158**], the resulting approach was ranked 1st in both angular and end-point errors in the Middlebury evaluation. At the writing of the dissertation (July 2012), **Classic+NL** ranks 13th in both AAE and EPE. Several recent and high-ranking methods directly build on **Classic+NL**, such as the layered model we will present in the following chapters, methods with more advanced motion prior models [**42, 80**], and efficient optimization schemes for the non-local term [**87**].

4.2. Previous Work

It is important to separately analyze the contributions of the objective function that defines the problem (*the model*) and the optimization algorithm and implementation used to minimize it (*the method*). The HS formulation, for example, has long been thought to be highly inaccurate. Barron *et al.* [**19**] reported an average angular error (AAE) of ~ 30 degrees on the "Yosemite" sequence. This confounds the objective function with the particular optimization method proposed by Horn and Schunck[1]. When optimized with today's methods, the HS objective achieves surprisingly competitive results despite the expected over-smoothing and sensitivity to outliers.

As shown in Figure 4.1, the reported accuracy of a method is jointly determined by the objective function of a paper, the optimization techniques, the implementation details, and the parameter tuning/learning. We will review papers in the first three aspects below.

Models: The global formulation of optical flow introduced by Horn and Schunck [**73**] relies on both brightness constancy and spatial smoothness assumptions, but suffers from the fact that the quadratic formulation is not robust to outliers. Black and Anandan [**26**] addressed this by replacing the quadratic error function with a robust formulation. Subsequently, many different robust functions have been explored [**34, 92, 159**] and it remains unclear which is best. We refer to

[1] They noted that the correct way to optimize their objective is by solving a system of linear equations as is common today. This was impractical on the computers of the day so they used a heuristic method.

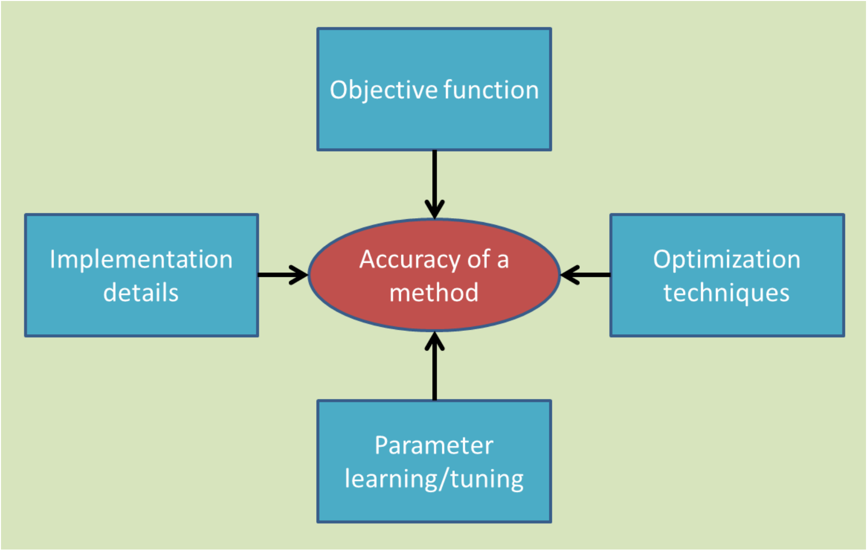


Figure 4.1. The accuracy of a reported method depends on several factors; we need to isolate the influence of other factors when evaluating the effect of a particular factor.

all these spatially-discrete formulations derived from HS as "classical." We systematically explore variations in the formulation and optimization of these approaches. The surprise is that the classical model, appropriately implemented, remains very competitive.

There are many formulations beyond the classical ones that we do not consider here. Significant ones use oriented smoothness [**117, 159, 184, 204**], rigidity constraints [**183, 184**], or image segmentation [**30, 91, 205, 196**]. While they deserve similar careful consideration, we expect many of our conclusions to carry forward. Note that one can select among a set of models for a given sequence [**106**], instead of finding a "best" model for all the sequences.

Methods: Many of the implementation details that are thought to be important date back to the early days of optical flow. Current best practices include coarse-to-fine estimation to deal with large motions [**21, 38**], texture decomposition [**183, 185**] or high-order filter constancy [**6, 34, 62, 92, 204**] to reduce the influence of lighting changes, bicubic interpolation-based warping [**92, 185**], temporal averaging of image derivatives [**72, 185**], graduated non-convexity [**31**] to minimize non-convex energies [**26, 159**], and median filtering after each incremental estimation step to remove outliers [**185**].

This median filtering heuristic is of particular interest as it makes non-robust methods more robust and improves the accuracy of all methods we tested. The effect on the objective function and the underlying reason for its success have not previously been analyzed. Least median squares estimation can be used to robustly reject outliers in flow estimation [**13**], but previous work has focused on the data term.

Related to median filtering, and our new non-local term, is the use of bilateral filtering to prevent smoothing across motion boundaries [**195**]. The approach separates a variational method into two filtering update stages, and replaces the original anisotropic diffusion process with multi-cue driven bilateral filtering. As with median filtering, the bilateral filtering step changes the original energy function.

Models that are formulated with an L1 robust penalty are often coupled with specialized total variation (TV) optimization methods [**200**]. Here we focus on generic optimization methods that can apply to any model and find they perform as well as reported results for specialized methods.

Despite recent algorithmic advances, there is a lack of publicly available, easy to use, and accurate flow estimation software. The GPU4Vision project [**2**] has made a substantial effort to change this and provides executable files for several accurate methods [**183, 184, 185, 189**]. The dependence on the GPU and the lack of source code are limitations. We hope that our public Matlab code [**3**] will not only help in understanding the practices of optical flow, but also let others exploit optical flow as a useful tool in computer vision and related fields.

4.3. Classical Models

We write the "classical" optical flow objective function in its spatially discrete form as

$$E(\mathbf{u},\mathbf{v}) = \sum_{p}\{\sum_{q\in\mathcal{N}_p}\rho_D(I_1^p - I_2^q) + \lambda\sum_{q\in\Gamma_p}[\rho_S(u^p-u^q)+\rho_S(v^p-v^q)]\}, \tag{25}$$

where $\mathbf{u}$ and $\mathbf{v}$ are the horizontal and vertical components of the optical flow field to be estimated from images I_1 and I_2, λ is a regularization parameter, $\mathcal{N}_p = \{(i+u^p, j+v^p)\}$, the set Γ_p contains the four nearest spatial neighbors of p, and ρ_D and ρ_S are the data and spatial penalty functions. We consider three different penalty functions: (1) the quadratic HS penalty $\rho(x) = x^2$; (2) the Charbonnier penalty $\rho(x) = \sqrt{x^2+\epsilon^2}$ [**38**], a differentiable variant of the L1 norm, the most robust convex function; and (3) the Lorentzian $\rho(x) = \log(1+\frac{x^2}{2\sigma^2})$, which is a non-convex robust penalty used in [**26**]. Note that this classical model is related to a standard pairwise Markov Random Field (MRF) based on a 4-neighborhood.

In the remainder of this section we define a baseline method using several techniques from the literature. This is not the "best" method, but includes modern techniques and will be used for comparison. We only briefly describe the main choices, which are explored in more detail in the following section and the cited references.

Quantitative results are presented throughout the remainder of the text. In all cases we report the average end-point error (EPE) on the Middlebury training and test sets, depending on the experiment. Given the extensive nature of the evaluation, only average results are presented in the main body, while the details for each individual sequence are given in Appendix A.

4.3.1. Baseline Methods. To gain robustness against lighting changes, we follow [**185**] and apply the Rudin-Osher-Fatemi (ROF) structure texture decomposition method [**138**] to pre-process the input sequences and linearly combine the texture and structure components (in the proportion 20:1). The parameters are set according to [**185**].

Optimization is performed using a standard incremental multi-resolution technique (*e.g.,* [**26, 38**]) to estimate flow fields with large displacements. The optical flow estimated at a coarse level is used to warp the second image toward the first at the next finer level, and a flow increment is calculated between the first image and the warped second image. The standard deviation of the Gaussian anti-aliasing filter is set to be $\frac{1}{\sqrt{2d}}$, where d denotes the downsampling factor. Each level is recursively downsampled from its nearest lower level. In building the pyramid, the downsampling factor is not critical as pointed out in the next section and here we use the settings in [**159**], which

Table 4.1. Models. Average rank and end-point error (EPE) on the Middlebury *test set* using different penalty functions. Two state-of-the-art methods in December 2010 are included for comparison. The ranking information was at the publication of our conference paper [**158**] (June 2010); the average EPE ranks for **Adaptive** and **Complementary OF** are 26.8 and 29.3 at the writing of the dissertation (July 2012).

	Avg. Rank	**Avg. EPE**
Classic-C	14.9	0.408
HS	24.6	0.501
Classic-L	19.8	0.530
HS [**159**]	35.1	0.872
BA (Classic-L) [**159**]	30.9	0.746
Adaptive [**184**]	11.5	0.401
Complementary OF [**204**]	10.1	0.485

uses a factor of 0.8 in the final stages of the optimization. We adaptively determine the number of pyramid levels so that the top level has a width or height of around 20 to 30 pixels. At each pyramid level, we perform 10 warping steps to compute the flow increment.

At each warping step, we linearize the data term once, which involves computing terms of the type $(\frac{\partial I_2}{\partial x})^q$, where $\partial/\partial x$ denotes the partial derivative in the horizontal direction, and $q = (i + u^p, j + v^p)$. As suggested in [**185**], we compute the derivatives of the second image using the 5-point derivative filter $\frac{1}{12}[-1\ 8\ 0\ \ -8\ 1]$, and warp the second image and its derivatives toward the first using the current flow estimate by bicubic interpolation. We then compute the spatial derivatives of the first image, average with the warped derivatives of the second image (cf. [**72**]), and use this in place of $\frac{\partial I_2}{\partial x}$. For pixels moving out of the image boundaries, we set both their corresponding temporal and spatial derivatives to zero. After each warping step, we apply a 5×5 median filter to the newly computed flow field to remove outliers [**185**].

For the Charbonnier (**Classic-C**) and Lorentzian (**Classic-L**) penalty function, we use a graduated non-convexity (GNC) scheme [**31**] as described in Chapter 3 that linearly combines a quadratic objective with a robust objective in varying proportions, from fully quadratic to fully robust. The standard deviations of the corresponding quadratic penalty function are set to be 1 for the Charbonnier penalty and the same as those of the penalty functions for the Lorentzian. Unlike Chapter 3, a single regularization weight λ is used for both the quadratic and the robust objective functions.

4.3.2. Baseline Results. The regularization parameter λ is selected among a set of candidate values to achieve the best average end-point error (EPE) on the Middlebury training set. For the Charbonnier penalty function, the candidate set is $[1,\ 3,\ 5,\ 8,\ 10]$ and 5 is optimal. The Charbonnier penalty uses $\epsilon = 0.001$ for both the data and the spatial term in Eq. (25). The Lorentzian uses $\sigma = 1.5$ for the data term, and $\sigma = 0.03$ for the spatial term. These parameters are fixed throughout the experiments, except where mentioned.

Table 4.1 summarizes the EPE results of the basic model with three different penalty functions on the Middlebury test set, along with the two top performers at the time of publication of our conference paper (considering only published papers). The classic formulations with two non-quadratic penalty functions (**Classic-C**) and (**Classic-L**) achieve competitive results despite their simplicity. The baseline optimization of **HS** and BA (**Classic-L**) results in significantly better accuracy than

Table 4.2. Pre-Processing. Average end-point error (EPE) on the Middlebury *training set* for the baseline method (**Classic-C**) using different pre-processing techniques. Significance is always with respect to **Classic-C**.

	Avg. EPE	significance	p-value
Classic-C	0.298	—	—
HS	0.384	**1**	**0.0078**
Classic-L	0.319	**1**	**0.0078**
Classic-C-brightness	0.288	0	0.9453
HS-brightness	0.387	**1**	**0.0078**
Classic-L-brightness	0.325	0	0.2969
Gradient	0.305	0	0.4609
Gaussian + Dx + Dy constancy	0.290	0	0.6406
Sobel edge magnitude [**172**]	0.417	**1**	**0.0156**
Laplacian [**92**]	0.430	**1**	**0.0078**
Laplacian1:1	0.301	0	0.6641
Gaussian pre-filtering ($\sigma = 0.5$)	0.281	0	0.5469
Texture4:1	0.286	0	0.5312

previously reported for these models [**159**]. Note that the analysis also holds for the training set (Table 4.2).

Because of the good performance of **Classic-C** despite its simplicity, we set it as the baseline below. It is worth noting that the spatially discrete MRF formulation taken here is competitive with variational methods such as [**184**]. Moreover, our baseline implementation of **HS** has a lower average EPE than many more sophisticated methods.

4.4. Practices Explored

We evaluate a range of variations from the baseline approach that have appeared in the literature, in order to illuminate which may be of importance. This analysis is performed on the Middlebury training set by changing only *one property at a time*. Statistical significance is determined using a Wilcoxon signed rank test [**190**] between each modified method and the baseline **Classic-C**; a p value less than 0.05 indicates a significant difference.

Pre-Processing. For each method, we optimize the regularization parameter λ for the training sequences and report the results in Table 4.2. The baseline uses a non-linear pre-filtering of the images to reduce the influence of illumination changes [**185**]. Table 4.2 shows the effect of removing this and using a standard brightness constancy model (***-brightness**). **Classic-C-brightness** actually achieves lower EPE on the training set than **Classic-C** but significantly lower accuracy on the test set: **Classic-C-brightness** = 0.726, **HS-brightness** = 0.759, and **Classic-L-brightness** = 0.603 – see Table 4.1 for comparison. This disparity suggests overfitting is more severe for the brightness constancy assumption.

Simpler alternatives, such as filter response (or high-order) constancy [**34, 159**] can serve the same function as ROF texture decomposition. A variety of pre-filters have been used in the literature, including derivative filters, Laplacians [**40, 92**], and Gaussians. Edges have also been emphasized using the Sobel edge magnitude [**172**].

Table 4.3. Model and Methods. Average end-point error (EPE) on the Middlebury *training set* for the baseline method (**Classic-C**) using different algorithm and modeling choices.

	Avg. EPE	significance	p-value
Classic-C	0.298	—	—
3 **warping steps**	0.304	0	0.9688
Down-0.5	0.298	0	1.0000
w/o GNC	0.354	0	0.1094
Bilinear	0.302	0	0.1016
w/o TAVG	0.306	0	0.1562
Central derivative filter	0.300	0	0.7266
7-point derivative filter [**38**]	0.302	0	0.3125
Deriv-warp	0.297	0	0.9531
Bicubic-II	0.290	**1**	**0.0391**
Deriv-warp-II	0.287	**1**	**0.0156**
Warp-deriv-II	0.288	**1**	**0.0391**
C-L ($\lambda = 0.6$)	0.303	0	0.1562
L-C ($\lambda = 2$)	0.306	0	0.1562
GC-0.45 ($\lambda = 3$)	0.292	**1**	**0.0156**
GC-0.25 ($\lambda = 0.7$)	0.298	0	1.0000
MF 3×3	0.305	0	0.1016
MF 7×7	0.305	0	0.5625
2× **MF**	0.300	0	1.0000
5× **MF**	0.305	0	0.6875
w/o MF	0.352	**1**	**0.0078**
Classic++	0.285	**1**	**0.0078**

Gradient only imposes constancy of the gradient vector at each pixel as proposed in [**34**] (i.e., it robustly penalizes Euclidean distance between image gradients). We use central difference filters ($Dx = [-0.5\ 0\ 0.5]$ and $Dy = Dx^T$). **Gaussian+Dx+Dy** assumes separate brightness, horizontal derivative, and vertical derivative constancy. A weighted combination of robust functions applied to each term is used as in [**159**]. Neither of these methods differ significantly from the baseline texture decomposition (**Classic-C**). Two methods are significantly worse: the **Sobel edge magnitude** [**172**] and **Laplacian** pre-filtering (5×5) as used in [**92**]. **Gaussian pre-filtering** ($\sigma = 0.5$) performed well on the synthetic sequences, but poorly on real ones. Finally, the texture-structure blending ratio is 20:1 in [**185**] but 4:1 in [**189**]. We find that (**Texture**4**:**1) performs better (but not significantly) on the synthetic sequences with a little degradation on the real ones.

For the Laplacian pre-filtering, we find combining the filtered image with the original image, in the proportion 1:1, improves accuracy significantly (**Laplacian**1**:**1). Similar to the ROF texture decomposition, such approach boosts certain frequency while suppressing the low frequency components that contain the lighting change.

Good practices: Some form of image filtering is useful but simple derivative constancy is nearly as good as the more sophisticated texture decomposition method.

Coarse-to-fine estimation and GNC. We vary the number of warping steps per pyramid level and find that **3 warping steps** gives similar results as using 10 (Table 4.3). For the GNC scheme, [**159**] uses a downsampling factor of 0.8 for non-convex optimization. A downsampling factor of 0.5 (**Down-**0.5), however, has nearly identical performance

Removing the GNC step for the Charbonnier penalty function (**w/o GNC**) results in higher EPE on most sequences and higher energy on all sequences (Table 4.5). This suggests that the GNC method is helpful even for the convex Charbonnier penalty function due to the nonlinearity of the data term.

Good practices: The downsampling factor does not matter when using a convex penalty; a standard factor of 0.5 is fine. Some form of GNC is useful even for a convex robust penalty like Charbonnier because of the nonlinear data term.

Interpolation method and derivatives. We find that bicubic interpolation is more accurate than bilinear (Table 4.3, **Bilinear**), as already reported in previous work [**185**]. Removing temporal averaging of the gradients (**w/o TAVG**), using **Central difference filter**, or using a **7-point derivative filter** $[-1\ 9\ -45\ 0\ 45\ -9\ 1]/60$ [**38**] all reduce accuracy compared to the baseline, but not significantly.

Another way to obtain the image derivative is to compute the derivatives of the warped image and then perform the temporal averaging with the spatial derivatives of the first image [**38**]. We find it produces similar results (**Deriv-warp**). However, the derivatives computed in either way are inconsistent with those implicitly interpolated by the bicubic interpolation. Bicubic interpolation interpolates not only the image but also the derivatives [**126**]. Because the MATLAB built-in function *interp2* is based on cubic convolution [**84**] and does not provide the derivatives used in interpolation, we use the spline-based implementation in [**126**]. With the new implementation, the three different ways to compute the derivatives give very similar EPE results, all better than the MATLAB built-in function. However, the one with consistent derivatives (**Bicubic-II**) gives the lowest energy solution, as shown in Table 4.4.

Table 4.4. Eq. (25) energy ($\times 10^6$) for the optical flow fields computed on the Middlebury *training set*, evaluated using spline-based bicubic interpolation [**126**]. Note the derivatives consistent with the interpolation method (**Bicubic-II**) produces the lowerest energy solution.

	Venus	Dimetr- odon	Hydr- angea	Rubber- Whale	Grove2	Grove3	Urban2	Urban3
Bicubic-II	0.552	0.734	0.835	0.481	1.656	2.167	1.061	1.275
Deriv-warp	0.559	0.745	0.840	0.484	1.682	2.201	1.073	1.333
Warp-deriv	0.563	0.745	0.845	0.486	1.694	2.238	1.117	1.347

Good practices: Use spline-based bicubic interpolation with a 5-point filter. Compute the derivative during the interpolation to obtain the lowest energy solutions. Temporal averaging of the derivatives is probably worthwhile for a small computational expense.

Penalty functions. We find that the convex Charbonnier penalty performs better than the more robust, non-convex Lorentzian on both the training and test sets. We use Charbonnier for the data term and Lorentzian for the spatial term (**C-L**) and vic versa (**L-C**). Both perform better than the Lorentzian for both terms but worse than the Charbonnier for both terms.

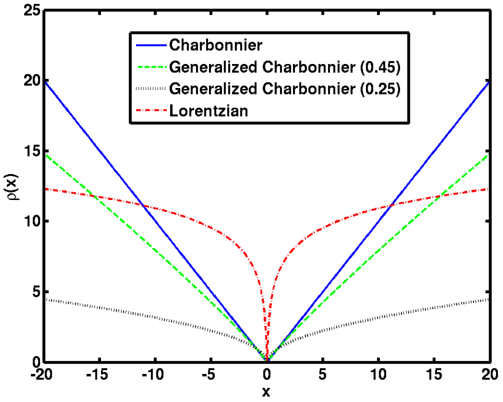


Figure 4.2. Different penalty functions for the spatial terms: Charbonnier ($\epsilon = 0.001$), generalized Charbonnier ($a = 0.45$ and $a = 0.25$), and Lorentzian ($\sigma = 0.03$).

One reason might be that non-convex functions are more difficult to optimize, causing the optimization scheme to find a poor local optimum. Another reason might be the MAP estimation actually favors the "wrong" penalty functions [**144**].

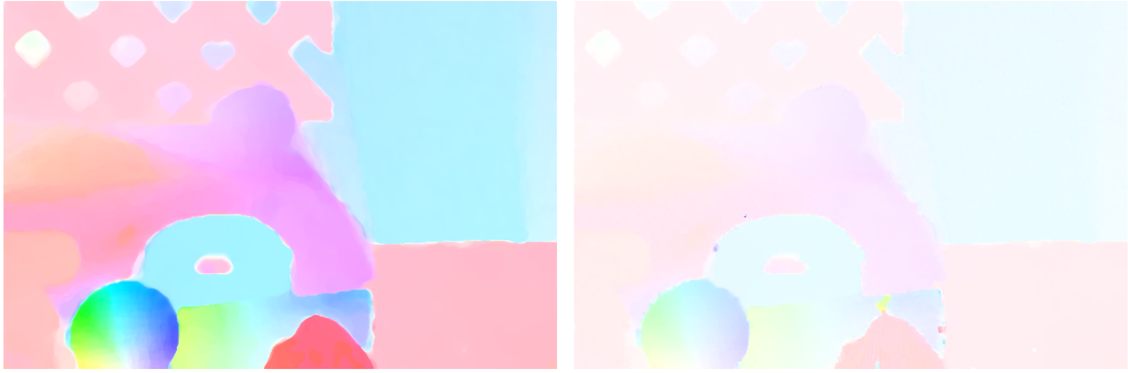
We investigate a generalized Charbonnier penalty function $\rho(x) = (x^2 + \epsilon^2)^a$ that is equal to the Charbonnier penalty when $a = 0.5$, and non-convex when $a < 0.5$ (see Figure 4.2). We optimize the regularization parameter λ again. We find a slightly non-convex penalty with $a = 0.45$ (**GC-**0.45) performs consistently better than the Charbonnier penalty, whereas more non-convex penalties (**GC-**0.25 with $a = 0.25$) show no improvement.

Good practices: The less-robust Charbonnier is preferable to the Lorentzian and a slightly non-convex penalty function (**GC-**0.45) is better still.

Median filtering. The baseline 5 × 5 median filter (**MF** 5 × 5) is better than both **MF** 3 × 3 [**185**] and **MF** 7 × 7 but the difference is not significant (Table 4.3). When we perform 5 × 5 median filtering twice (2× **MF**) or five times (5× **MF**) per warping step, the results are worse. Finally, removing the median filtering step (**w/o MF**) makes the computed flow significantly less accurate with larger outliers as shown in Table 4.3 and Figure 4.3.

Good practices: Median filtering the intermediate flow results once after every warping iteration is the single most important secret; 5 × 5 is a good filter size.

4.4.1. Best Practices. Combining the analysis above into a single approach means modifying the baseline to use the slightly non-convex generalized Charbonnier and the spline-based bicubic interpolation. This leads to a statistically significant improvement over the baseline (Table 4.3, **Classic++**). This method is directly descended from **HS** and **BA**, yet updated with the current best optimization practices known to us. This simple method ranks 32$^{\text{th}}$ out of 73 methods in both EPE and AAE on the Middlebury test set at the writing of the dissertation (July 2012).



(a) **With** median filtering

(b) **Without** median filtering

Figure 4.3. Estimated flow fields on sequence "RubberWhale" using **Classic-C** with and without (**w/o MF**) the median filtering step. Color coding as in [**17**]. (a) (**w/ MF**) energy $502,387$ and (b) (**w/o MF**) energy $449,290$. The median filtering step helps reach a solution free from outliers but with a higher energy.

Table 4.5. Eq. (25) energy ($\times 10^6$) for the optical flow fields computed on the Middlebury *training set*, evaluated using convolution-based bicubic interpolation [**84**]. Note that **Classic-C** uses graduated non-convexity (GNC), which reduces the energy, and median filtering, which increases it.

	Venus	Dimetr-odon	Hydr-angea	Rubber-Whale	Grove2	Grove3	Urban2	Urban3
Classic-C	0.589	0.748	0.866	0.502	1.816	2.317	1.126	1.424
w/o GNC	0.593	0.750	0.870	0.506	1.845	2.518	1.142	1.465
w/o MF	0.517	0.701	0.668	0.449	1.418	1.830	1.066	1.395

4.5. Models Underlying Median Filtering

Our analysis reveals the practical importance of median filtering during optimization to *denoise* the flow field. We ask whether there is a *principle* underlying this heuristic?

One interesting observation is that flow fields obtained with median filtering have substantially *higher* energy than those without (Table 4.5 and Figure 4.3). If the median filter is helping to optimize the objective, it should lead to lower energies. Higher energies and more accurate estimates suggest that incorporating median filtering changes the objective function being optimized.

The insight that follows from this is that the median filtering heuristic is related to the minimization of an objective function that differs from the classical one. In particular the optimization of Eq. (25), with interleaved median filtering, approximately minimizes

$$\begin{aligned} E(\mathbf{u},\mathbf{v}) = &\sum_p \{ \sum_{q\in\mathcal{N}_p} \rho_D(I_1^p - I_2^q) + \lambda \sum_{q\in\Gamma_p} [\rho_S(u^p - u^q) + \rho_S(v^p - v^q)] \} \\ &+ \sum_p \sum_{q\in\Gamma_p^{\mathrm{NL}}} \lambda_N(|u^p - u^q| + |v^p - v^q|), \end{aligned} \tag{26}$$

where N_p is the set of neighbors of pixel p in a possibly large area and λ_N is a scalar weight. The term in braces is the same as the flow energy from Eq. (25), while the last term is new. This *non-local* term [**39, 61**] imposes a particular smoothness assumption within a specified region of the flow

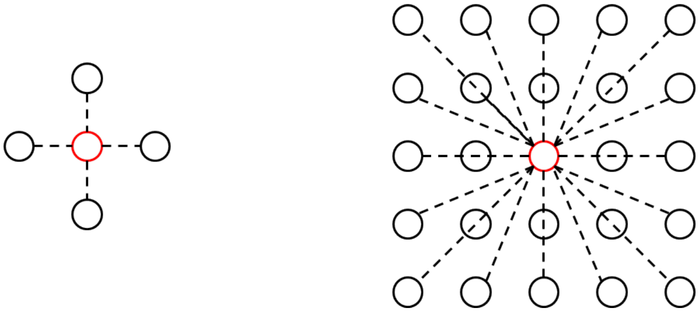


Figure 4.4. Neighbors for the pairwise model (left) and the non-local model (right). The pairwise model connects a pixel with its nearest neighbors, while the non-local term connects a pixel with many pixels in a large spatial neighborhood.

field [2]. Here we take this term to be a 5×5 rectangular region to match the size of the median filter in **Classic-C**. Figure 4.4 shows the neighborhood for the standard pairwise model and the non-local term.

It is usually difficult to directly optimize the objective (26) with a large spatial term. A common practice is to relax the objective with an auxiliary flow field as

$$\begin{aligned} E_A(\mathbf{u}, \mathbf{v}, \hat{\mathbf{u}}, \hat{\mathbf{v}}) = & \sum_p \{ \sum_{q \in \mathcal{N}_p} \rho_D(I_1^p - I_2^q) + \lambda \sum_{q \in \Gamma_p} [\rho_S(u^p - u^q) + \rho_S(v^p - v^q)]\} \\ & + \lambda_C(||\mathbf{u} - \hat{\mathbf{u}}||^2 + ||\mathbf{v} - \hat{\mathbf{v}}||^2) + \sum_p \sum_{q \in \Gamma_p^{\mathrm{NL}}} \lambda_N(|\hat{u}^p - \hat{u}^q| + |\hat{v}^p - \hat{v}^q|), \end{aligned} \tag{27}$$

where $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ denote an auxiliary flow field and λ_C is a scalar weight. A third (coupling) term encourages $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ and $\mathbf{u}, \mathbf{v}$ to be the same (cf. [**184, 200**]).

The connection to median filtering (as a denoising method) derives from the fact that there is a direct relationship between the median and L1 minimization. Consider a simplified version of Eq. (27) with just the coupling and non-local terms,

$$E(\hat{\mathbf{u}}) = \lambda_C||\mathbf{u} - \hat{\mathbf{u}}||^2 + \sum_p \sum_{q \in \Gamma_p^{\mathrm{NL}}} \lambda_N |\hat{u}^p - \hat{u}^q| \tag{28}$$

While minimizing this is similar to median filtering $\mathbf{u}$, there are two differences. First, the non-local term minimizes the L1 distance between the central value and all flow values in its neighborhood except itself. Second, Eq.(28) incorporates information about the data term through the coupling equation; median filtering the flow ignores the data term.

The formal connection between Eq.(28) and median filtering[3] is provided by Li and Osher [**96**] who show that minimizing Eq.(28) is related to a different median computation

$$\hat{u}^p_{(k+1)} = \mathrm{median}(\mathrm{Neighbors}^{(k)} \cup \mathrm{Data}) \tag{29}$$

[2] Bruhn *et al.* [**38**] also integrated information over a local region in a global method but did so for the data term.

[3] Hsiao *et al.* [**74**] established the connection in a slightly different way.

where $\text{Neighbors}^{(k)} = \{\hat{u}^q_{(k)}\}$ for $q \in \Gamma_p^{\text{NL}}$ and $\hat{\mathbf{u}}^{(0)} = \mathbf{u}$ as well as

$$\text{Data} = \{u^p, u^p \pm \tfrac{\lambda_N}{\lambda_C}, u^p \pm \tfrac{2\lambda_N}{\lambda_C} \cdots, u^p \pm \tfrac{|\Gamma_p^{\text{NL}}|\lambda_N}{2\lambda_C}\},$$

where $|\Gamma_p^{\text{NL}}|$ denotes the (even) number of neighbors of (p). Note that the set of "data" values is balanced with an equal number of elements on either side of the value u^p and that information about the data term is included through u^p. Repeated application of Eq. (29) converges rapidly [**96**].

Observe that, as λ_N/λ_C increases, the weighted data values on either side of u^p move away from the values of Neighbors and cancel each other out. As this happens, Eq. (29) approximates the median at the first iteration

$$\hat{u}^p_{(1)} \approx \text{median}(\text{Neighbors}^{(0)} \cup \{u^p\}). \tag{30}$$

The relaxed energy function given by Eq. (27) thus combines the original objective with an approximation to the median, the influence of which is controlled by λ_N/λ_C. Note in practice the weight λ_C on the coupling term is usually small or is steadily increased from small values [**185, 200**]. We optimize the new objective (27) by alternately minimizing

$$E_O(\mathbf{u}, \mathbf{v}) = \sum_p \{ \sum_{q \in \mathcal{N}_p} \rho_D(I_1^p - I_2^q) + \lambda \sum_{q \in \Gamma_p} [\rho_S(u^p - u^q) + \rho_S(v^p - v^q)]\} + \lambda_C(||\mathbf{u} - \hat{\mathbf{u}}||^2 + ||\mathbf{v} - \hat{\mathbf{v}}||^2) \tag{31}$$

and

$$E_M(\hat{\mathbf{u}}, \hat{\mathbf{v}}) = \lambda_C(||\mathbf{u} - \hat{\mathbf{u}}||^2 + ||\mathbf{v} - \hat{\mathbf{v}}||^2) + \sum_p \sum_{q \in \Gamma_p^{\text{NL}}} \lambda_N |\hat{u}^p - \hat{u}^q| \tag{32}$$

We find that optimization of the coupled set of equations is superior in terms of EPE performance.

The alternating optimization strategy first holds $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ fixed and minimizes Eq. (31) w. r. t. $\mathbf{u}, \mathbf{v}$. Then, with $\mathbf{u}, \mathbf{v}$ fixed, we minimize Eq. (32) w. r. t. $\hat{\mathbf{u}}, \hat{\mathbf{v}}$. Note that Eqs. (28) and (32) can be minimized by repeated application of Eq. (29); we use this approach with 5 iterations. We perform 10 steps of alternating optimizations at every pyramid level and change λ_C logarithmically from 10^{-4} to 10^2. During the first and second GNC stages, we set $\mathbf{u}, \mathbf{v}$ to be $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ after every warping step (this step helps reach solutions with lower energy and EPE; cf. **Classic-C-A-noRep** in Tables 4.6 and 4.7). In the end, we take $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ as the final flow field estimate. The other parameters are $\lambda = 5, \lambda_N = 1$.

Alternatingly optimizing this new objective function (**Classic-C-A**) leads to similar results as the baseline **Classic-C** (Table 4.6). We also compare the energy of these solutions using the new objective and find the alternating optimization produces the lowest energy solutions, as shown in Table 4.7. To do so, we set both the flow field $\mathbf{u}, \mathbf{v}$ and the auxiliary flow field $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ to be the same in Eq.(27).

We find that approximately optimizing the new objective by changing λ_C logarithmically from 10^{-4} to 10^{-1} produces better EPE results but higher energy (**Classic-C-A-II**). We also try the conjugate gradient descent method [**4**] to solve Eq. (28) but obtain results with slightly worse EPE performance and higher energy.

In summary, we show that the heuristic median filtering step in **Classic-C** can now be viewed as energy minimization of a new objective with a non-local term. The explicit formulation emphasizes

Table 4.6. Average end-point error (EPE) on the Middlebury *training set* is shown for the new model with alternating optimization (**Classic-C-A**).

	Avg. EPE	significance	p-value
Classic-C	0.298	—	—
Classic-C-A	0.305	0	0.8125
Classic-C-A-noRep	0.309	0	0.5781
Classic-C-A-II	0.296	0	0.7188
Classic-C-A-CGD	0.305	0	0.5625

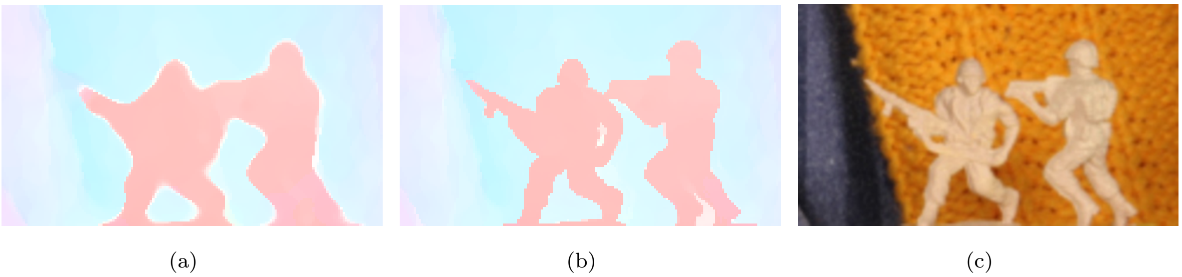

(a) (b) (c)

Figure 4.5. Median filtering over-smoothes the rifle in the "Army" sequence, while the proposed weighted non-local term preserves the detail. (a) **Classic++**, (b) **Classic+NL**, and (c) first frame.

the value of robustly integrating information over large neighborhoods and enables the improved model described below.

Table 4.7. Eq. (27) energy ($\times 10^6$) for the computed flow fields on the Middlebury *training set*. The alternating optimization strategy (**Classic-C-A**) produces the lowest energy solutions.

	Venus	Dimetr- odon	Hydr- angea	Rubber- Whale	Grove2	Grove3	Urban2	Urban3
Classic-C	0.817	0.903	1.202	0.674	2.166	3.144	1.954	2.153
Classic-C w/o MF	0.886	0.945	1.299	0.725	2.315	3.513	2.234	2.712
Classic-C-A	0.784	0.889	1.139	0.666	2.064	2.976	1.922	2.049
Classic-C-A-noRep	0.790	0.894	1.165	0.670	2.092	3.143	2.005	2.317
Classic-C-A-II	0.830	0.915	1.235	0.686	2.223	3.247	1.990	2.182
Classic-C-A-CGD	0.833	0.909	1.224	0.674	2.213	3.357	2.020	2.236

4.6. Improved Model

By formalizing the median filtering heuristic as an explicit objective function, we can find ways to improve it. While median filtering in a large neighborhood has advantages as we have seen, it also has problems. A neighborhood centered on a corner or thin structure is dominated by the surround and computing the median results in oversmoothing as illustrated in Figure 4.5(a).

Examining the non-local term suggests a solution. For a given pixel, if we know which other pixels in the area belong to the same surface, we can weight them more highly. The modification to the objective function is achieved by introducing a weight into the non-local term [**39, 61**]:

$$\sum_{p} \sum_{q \in \Gamma_p^{\mathrm{NL}}} w_q^p (|\hat{u}^p - \hat{u}^q| + |\hat{v}^p - \hat{v}^q|), \tag{33}$$

where w_q^p represents how likely pixel q is to belong to the same surface as p.

Of course, we do not know w_q^p, but can approximate it. We draw ideas from [**140, 195, 199**] to define the weights according to their spatial distance, their color-value distance, and their occlusion state as

$$w_q^p \propto \exp\Big\{ - \tfrac{|p-q|^2}{2\sigma_1^2} - \tfrac{|\mathbf{I}^p - \mathbf{I}^q|^2}{2\sigma_2^2 n_c} \Big\} \frac{o(q)}{o(p)}, \tag{34}$$

where $\mathbf{I}^p$ is the color vector in the Lab space, n_c is the number of color channels and 1 for gray image, $\sigma_1 = 7, \sigma_2 = 7$, and the occlusion variable $o(p)$ is calculated using Eq. (22) in [**140**] as

$$o(p) = \exp\left\{ -\frac{d^2(p)}{2\sigma_d^2} - \frac{(I(p) - I(q))^2}{2\sigma_e^2} \right\}, \tag{35}$$

where $I(p) - I(q)$ is the data term constancy error and $d(p)$ is the one-sided divergence function, defined as

$$d(p) = \begin{cases} \mathrm{div}(p), & \mathrm{div}(p) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{36}$$

in which the flwo divergence $\mathrm{div}(p)$ is

$$\mathrm{div}(p) = \frac{\partial}{\partial x} u(p) + \frac{\partial}{\partial y} v(p), \tag{37}$$

where $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ are respectively the horizontal and vertical flow derivatives. The occlusion variable $o(p)$ is near zero for occluded pixels and near one for non-occluded pixels. We set the parameters in Eq. 35 as $\sigma_d = 0.3$ and $\sigma_e = 20$; this is the same as in [**140**].

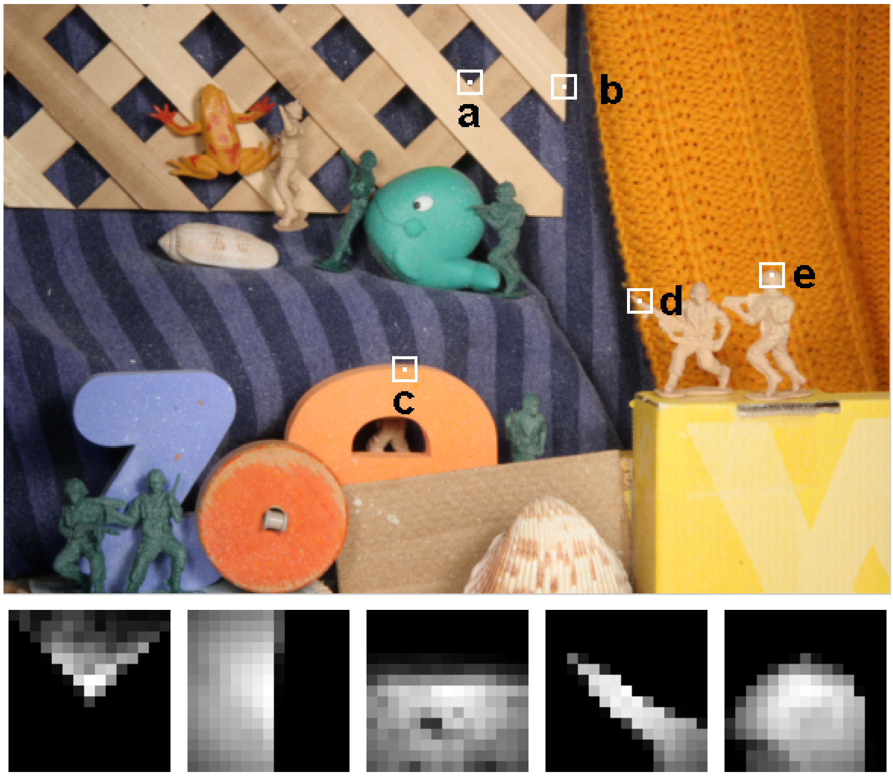
Examples of such weights are shown for several 15 × 15 neighborhoods in Figure 4.6; bright values indicate higher weights. Note the neighborhood labeled **d**, corresponding to the rifle. Since pixels on the rifle are in the minority, an unweighted median would oversmooth. The weighted term instead robustly estimates the motion using values on the rifle. A closely related piece of work is [**128**], which uses the intervening contour to define affinities among neighboring pixels for the local Lucas and Kanade [**105**] method. However it only uses this scheme to estimate motion for sparse points and then interpolates the dense flow field.

We approximately solve for $\hat{\mathbf{u}}$ as the following weighted median problem

$$\min_{\hat{u}^p} \sum_{q \in \Gamma_p^{\mathrm{NL}} \cup \{p\}} w_q^p |\hat{u}^p - u^q|, \tag{38}$$

using the formula (3.13) in [**96**] for all the pixels (**Classic+NL-Full**). Note if all the weights are equal, the solution is just the median. In practice, we can adopt an adaptive version (**Classic+NL**) to save computation without performance loss. Given a current estimate of the flow, we detect motion boundaries using a Sobel edge detector and dilate these edges with a 5 × 5 mask to obtain flow boundary regions. In these regions we use the weighting in Eq.(34) in a 15 × 15 neighborhood. In the non-boundary regions, we use equal weights in a 5×5 neighoborhood to compute the median.

To further reduce the computation, we can adopt a two-stage GNC process and perfrom 3 warping steps per pyramid level. This fast version (**Classic+NL-Fast**) has nearly the same overall performance, with slightly degradation on the "Urban3" sequence that has large motion.



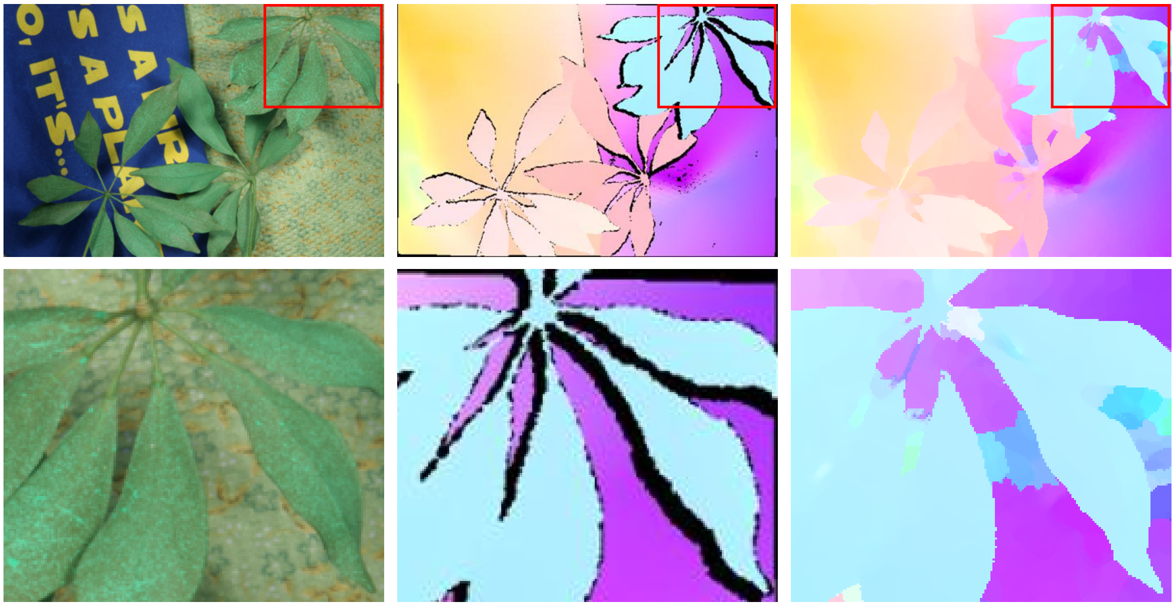
(a) (b) (c) (d) (e)

Figure 4.6. Neighbor weights of the proposed weighted non-local term at different positions in the "Army" sequence.

Tables 4.8 and 4.9 show that the weighted non-local term (**Classic+NL**) improves the accuracy on both the training and the test sets. Note that the fine detail of the "rifle" is preserved in Figure 4.5(b). When our conference paper [**158**] was published in June 2010, **Classic+NL** ranked 1st in both AAE and EPE in the Middlebury evaluation and had the lowest average AAE and EPE among all listed algorithms. At the writing of the dissertation (July 2012), **Classic+NL** ranks 13th in both AAE and EPE. The running time on the test "Urban" sequence is about 70 minutes for **Classic+NL-Full**, about 16 miniutes for **Classic+NL**, and about 2.5 minutes for **Classic+NL-Fast** in MATLAB.

We test some variants of the weighted non-local term (**Classic+NL**). Table 4.8 shows the relative importance of each term in determining the weight and influence of the parameter setting on the final results. Using different color spaces results in some performances loss. Removing the color information from the weighting scheme (**w/o color**) results in significant degradation in performance. Without occlusion (**w/o occ**) or spatial distance (**w/o spa**) cues does not degrade the method significantly. The gray version (**Classic+NL-Gray**) of the non-local term is statistically worse than the color version.

Changing the σ_2 for the color cue to 5 or 10 obtains similar results as the default 7. The default λ is 3, while 1 and 9 produce some loss in performance. We also study the maximum size of the



(a)First frame (b) Ground truth (c) Estimated flow field

Figure 4.7. Occlusions causes serious problem to **Classic+NL**, which does not explicitly model occlusions.

neighborhood for the non-local term. 11×11 gives close performance, while 19×19 is slightly better at a higher computational cost.

Closely-related work: Werlberger *et al.* [**188**] independently propose a non-local term for optical flow estimation and the spatial term is similar to our non-local term. They use the normalized cross correlation as the data term to deal with lighting changes and optimize their objective function by the primal-dual method. Their work is motivated by the success of the non-local regularization [**39**] in image restoration and stereo, while our work is inspired by the success for the heuristic median filtering step in flow estimation and the link between median filtering and non-local regularization term.

Limitations: Classic+NL produces larger errors in occlusion regions on some sequences, such as "Schefflera" shown in Figure 4.7. The classical flow formulation assumes that every pixel at the current frame should have a corresponding pixel at the next frame. However this assumption breaks down when occlusion happens. Pixels that are occluded by some foreground objects do not have corresponding pixels, resulting in large errors to the classical formulation.

4.6.1. Results on the MIT Dataset.

To test the robustness of these models on other data, we applied **HS**, **Classic-C**, and **Classic+NL** to sequences from Liu *et al.* [**99**], and compared the estimated flow fields to the human labeled ground truth. Note only five of the eight test sequences in [**99**] are available on-line; these are tested here.

Tables 4.10 and 4.11 show the results on these sequences, which are very different in nature from the Middlebury set and include an outdoor scene as well as a scene of a fish tank. The results are compared with the CLG method [**38**] used in [**99**]. It is important to point out that the CLG method was tuned to obtain the optimal results on the test sequences. Our method had no such

	Avg. EPE	significance	p-value
Classic+NL	0.221	—	—
Classic+NL-Full	0.222	0	0.8203
Classic+NL-Fast	0.221	0	0.3125
RGB	0.240	**1**	**0.0156**
HSV	0.231	**1**	**0.0312**
LUV	0.226	0	0.5625
Gray	0.253	**1**	**0.0078**
w/o color	0.283	**1**	**0.0156**
w/o occ	0.226	0	0.1250
w/o spa	0.223	0	0.5625
$\sigma_2 = 5$	0.221	0	1.0000
$\sigma_2 = 10$	0.224	0	0.2500
$\lambda = 1$	0.236	0	0.1406
$\lambda = 9$	0.244	0	0.1016
11×11	0.223	0	0.5938
19×19	0.220	0	0.8750

Table 4.8. Average end-point error (EPE) on the Middlebury *training set* is shown for the the improved model and its variants.

Table 4.9. Average end-point error (EPE) on the Middlebury *test set* for the **Classic++** model with two different preprocessing techniques and its improved model. The ranking information was at the publication of our conference paper [**158**] (June 2010); the EPE rank for **Classic+NL** is 16.2 at the writing of the dissertation (July 2012).

	Avg. Rank	**Avg. EPE**
Classic++	13.4	0.406
Classic++Gradient	15.1	0.430
Classic+NL	6.2	0.319
Classic+NL-Full	6.6	0.316

tuning and we used the same parameters as those used in all the other experiments. This suggests that training on the Middlebury data results in a method that generalizes to other sequences. The only place where this fails is on the “fish” sequence where there is transparent motion in a liquid medium; the statistics in this sequence are very different from the Middlebury training data.

Table 4.10. Results on the MIT dataset [**99**]. Average end-point error (EPE). Results of the combined local and global (CLG) method [**38**] are from [**99**], which was tuned for each sequence.

	Average	Table	Hand	Toy	Fish	CameraMotion
CLG [**38, 99**]	1.239	0.976	4.181	0.456	0.196	0.385
HS	2.129	1.740	6.108	0.620	1.309	0.869
Classic-C	1.345	1.064	3.428	0.482	1.061	0.690
Classic+NL	1.106	0.91	2.75	0.487	0.772	0.611

Table 4.11. Results on the MIT dataset [**99**]. Average Angular error (AAE). Results of the combined local and global (CLG) method [**38**] are from [**99**], which was tuned for each sequence.

	Average	Table	Hand	Toy	Fish	CameraMotion
CLG [**38, 99**]	16.281	8.996	58.904	2.573	5.689	5.243
HS	16.769	9.633	32.572	2.931	29.180	9.531
Classic-C	10.266	6.124	13.292	2.046	23.667	6.201
Classic+NL	8.156	5.289	8.469	2.543	18.187	6.291

4.7. Conclusions and Discussions

Implemented using modern practices, classical optical flow formulations produce competitive results on the Middlebury training and test sets. To understand the "secrets" that help such basic formulations work well, we quantitatively studied various aspects of flow approaches from the literature, including their implementation details. Among the good practices, we found that using median filtering to denoise the flow after every warping step is key to improving accuracy, but that it increases the energy of the final result. Exploiting connections between median filtering and L1-based denoising, we showed that algorithms relying on a median filtering step are approximately optimizing a different objective that regularizes flow over a large spatial neighborhood. This principle enables us to design and optimize improved models that weight the neighbors adaptively in an extended image region. The resulting **Classic+NL** method can better preserve fine motion details and achieves consistent improvement than the median filtering heuristic. The MATLAB code is publicly available at our webpages [**3**].

The **Classic+NL** method still cannot deal with large occlusions. Small occlusions may be well handled by adopting a robust data term, but large regions of occlusions still cause problems to the robust formulation. In the next chapter we will investigate how to explicitly model and reason about occlusions using a generative layered approach.

CHAPTER 5

A Generative Layered Model Based on Thresholded Support Functions

In this chapter, we present a probabilistic layered model for image motion. Layered models are a powerful way of describing natural scenes containing smooth surfaces that may overlap and occlude each other. For image motion estimation, such models have a long history but have not achieved the wide use or accuracy of non-layered methods. We present a new probabilistic model of optical flow in layers that addresses many of the shortcomings of previous approaches. In particular, we define a probabilistic graphical model that explicitly captures: 1) occlusions and disocclusions; 2) depth ordering of the layers; 3) temporal consistency of the layer segmentation. Additionally the optical flow in each layer is modeled by a combination of a parametric model and a smooth deviation based on an MRF with a robust spatial prior; the resulting model allows *roughness in layers*. Finally, a key contribution is the formulation of the layers using an image-dependent support function prior based on recent models for static scene segmentation. The method achieves consistent improvement over the non-layered **Classic+NL** method on the Middlebury benchmark and produces meaningful scene segmentations as well as detected occlusion regions.

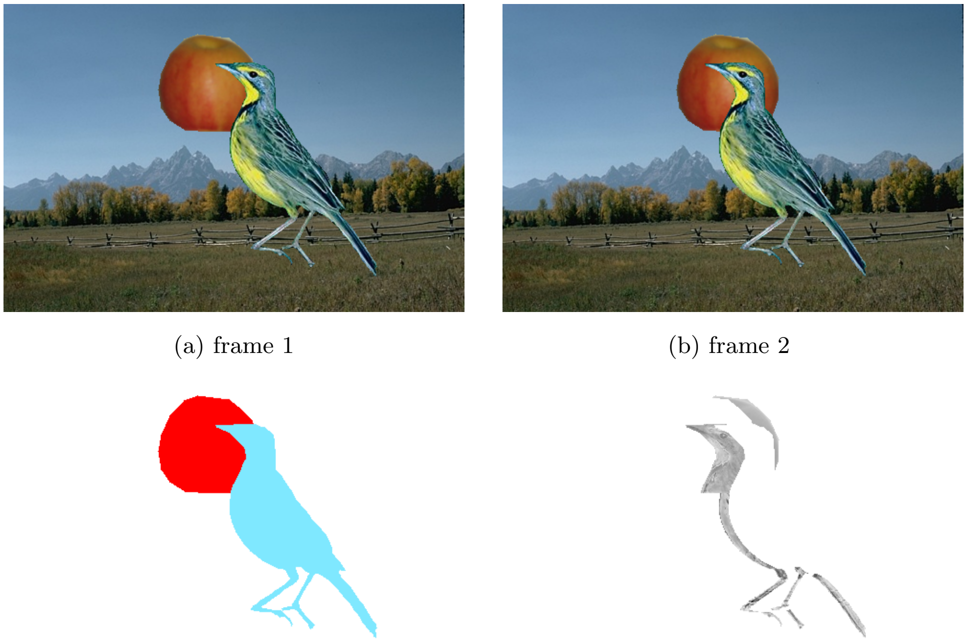
5.1. Introduction

The classical optical flow formulation by Horn and Schunck assumes that every pixel has a corresponding pixel at the next frame. Occlusions violate this assumption because the occluded pixels do not appear at the next frame, as shown in Figure 5.1. Imposing constancy on the occluded pixels causes problems for most methods descended from the classical formulation. We need a better model to describe occlusions.

Typical scenes contains a few moving foreground objects together with a background motion. Instead of assuming that every pixel is an individual unit, we can group them together by common motion. Every moving object can be labeled as an individual layer that has its own motion. This is usually called the layered approach.

Layered models of scenes offer significant benefits for optical flow estimation [**46, 78, 179**]. Splitting the scene into layers enables the motion in each layer to be defined more simply, and the estimation of motion boundaries to be separated from the problem of smooth flow estimation. Layered models also make reasoning about occlusion relationships easier. If we can segment the scene into layers and know the relative depth ordering, we can reason about occlusions. The segmentation or the visibility mask of each layer tells whether a pixel is occluded by another layer over time, as shown in Figure 5.2.

In practice, however, none of the current top performing optical flow methods use a layered approach [**16**]. The most accurate approaches are single-layered, and instead use some form of



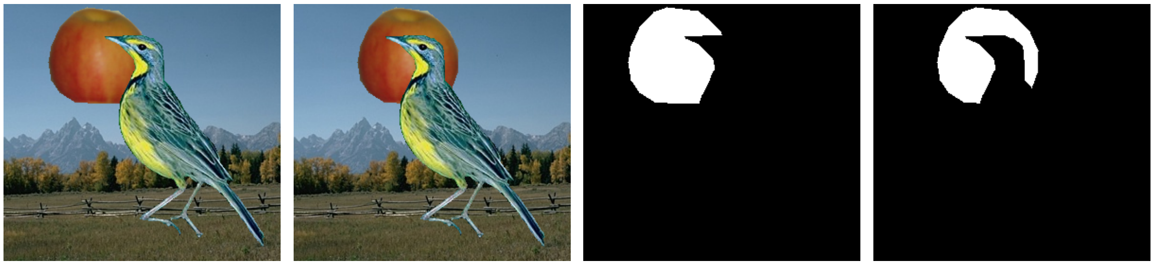
(a) frame 1

(b) frame 2

(c) ground truth motion

(d) inverse absolute brightness constancy error

Figure 5.1. Occlusions are ill-defined for the classical optical flow formulation: even ground truth motion has large brightness constancy error in occlusion regions. Darker indicates larger brightness constancy errors in (d).



(a) frame 1

(b) frame 2

(c) visibility at frame 1

(d) visibility at frame 2

Figure 5.2. Occlusion reasoning for the apple layer.

robust smoothness assumption to cope with flow discontinuities [**26**]. This chapter formulates a new probabilistic, layered motion model that addresses the key problems of previous layered approaches. At the time of writing, it achieves the lowest average error of all tested approaches on the Middlebury optical flow benchmark [**16**]. In particular, the accuracy at occlusion boundaries is significantly better than previous methods. By segmenting the observed scene, our model also identifies occluded and disoccluded regions.

Layered models provide a segmentation of the scene and this segmentation, because it corresponds to scene structure, should persist over time. However, this persistence is not a benefit if one is only computing flow between two frames; this is one reason that multi-layer models have not surpassed their single-layer competitors on two-frame benchmarks. Without loss of generality, here we use three-frame sequences to illustrate our method. In practice, these three frames can be constructed from an image pair by computing both the forward and backward flow. The key is that this gives two segmentations of the scene, one at each time instant, both of which must be consistent with the flow. We formulate this *temporal layer consistency* probabilistically. Note that the assumption of temporal layer consistency is much more realistic than previous assumptions of temporal motion consistency [**25**]; while the scene motion can change rapidly, scene structure persists.

One of the main motivations for layered models is that, conditioned on the segmentation into layers, each layer can employ affine, planar, or other strong models of optical flow. By applying a single smooth motion across the entire layer, these models combine information over long distances and interpolate behind occlusions. Such rigid parametric assumptions, however, are too restrictive for real scenes. Instead one can model the flow within each layer as smoothly varying [**186**]. While the resulting model is more flexible than traditional parametric models, we find that it is still not as accurate as robust single-layer models. Consequently, we formulate a hybrid model that combines a base affine motion with a robust Markov random field (MRF) model of *deformations* from affine [**30**]. This *roughness in layers* model, which is similar in spirit to work on plane+parallax [**77, 90, 141**], encourages smooth flow within layers but allows significant local deviations.

Because layers are temporally persistent, it is also possible to reason about their relative depth ordering. In general, reliable recovery of depth order requires three or more frames. Our probabilistic formulation explicitly orders layers by depth, and we show that the correct order typically produces more probable (lower energy) solutions. This also allows explicit reasoning about occlusions, which our model predicts at locations where the layer segmentations for consecutive frames disagree.

Many previous layered approaches are not truly "layered": while they segment the image into multiple regions with distinct motions, they do not model what is in front of what. For example, widely used MRF models [**187**] encourage neighboring pixels to occupy the same region, but do not capture relationships between regions. In contrast, building on recent state-of-the-art results in static scene segmentation [**157**], our model determines layer support via an ordered sequence of occluding binary masks. These binary masks are generated by thresholding a series of random, continuous support functions. This approach uses image-dependent Gaussian random field priors and favors partitions which accurately match the statistics of real scenes [**157**]. Moreover, the continuous layer support functions play a key role in accurately modeling temporal layer consistency. The resulting model produces accurate layer segmentations that improve flow accuracy at occlusion boundaries, and recover meaningful scene structure.

As summarized in Figure 5.3, our method is based on a principled, probabilistic generative model for image sequences. By combining recent advances in dense flow estimation and natural image segmentation, we develop an algorithm that simultaneously estimates accurate flow fields, detects occlusions and disocclusions, and recovers the layered structure of realistic scenes.

5.2. Previous Work

Layered approaches to motion estimation have long been seen as elegant and promising, since spatial smoothness is separated from the modeling of discontinuities and occlusions. Darrell and Pentland [**47, 46**] provide the first full approach that incorporates a Bayesian model, "support maps" for segmentation, and robust statistics. Wang and Adelson [**179**] clearly motivate layered models of image sequences, while Jepson and Black [**78**] formalize the problem using probabilistic mixture models. Chapter 2 reviews more recent methods [**11, 22, 81, 83, 119, 89, 145, 169, 187, 198**].

Early methods, which use simple parametric models of image motion within layers, are not highly accurate. Observing that rigid parametric models are too restrictive for real scenes, Weiss [**186**] uses a more flexible Gaussian process to describe the motion within each layer. Even using modern implementation methods [**158**] this approach does not achieve state-of-the-art results. Allocating a separate layer for every small surface discontinuity is impractical and fails to capture important global scene structure. Our approach, which allows "roughness" within layers rather than "smoothness," provides a compromise that captures coarse scene structure as well as fine within-layer details.

One key advantage of layered models is their ability to realistically model occlusion boundaries. To do this properly, however, one must know the relative depth order of the surfaces. In their pioneering work, Wang and Adelson [**179**] incorporate the depth ordering of layers and layers in the front occlude layers behind. Performing inference over the combinatorial range of possible occlusion relationships is challenging and, consequently, only a few later models explicitly encode relative depth [**81, 202**]. Recent work revisits the layered model to handle occlusions [**63**], but does not explicitly model the layer ordering or achieve state-of-the-art performance on the Middlebury benchmark. While most current optical flow methods are "two-frame," layered methods naturally extend to longer sequences [**81, 198, 202**].

Layered models all have some way of making either a hard or soft assignment of pixels to layers. Weiss and Adelson [**187**] introduce spatial coherence to these layer assignments using a spatial MRF model. However, the Ising/Potts MRF they employ assigns low probability to typical segmentations of natural scenes [**114**]. Adapting recent work on static image segmentation by Sudderth and Jordan [**157**], we instead generate spatially coherent, ordered layers by thresholding a series of random continuous functions. As in the single-image case, this approach realistically models the size and shape properties of real scenes. For motion estimation there are additional advantages: it allows accurate reasoning about occlusion relationships and modeling of temporal layer consistency. The motion competition framework [**35, 44**] uses level sets to model the scene segmentation in a variational setting, but does not address occlusion reasoning.

5.3. A Layered Motion Model

Building on this long sequence of prior work, our generative model of layered image motion is summarized in Figure 5.3. Below we describe how the generative model captures piecewise smooth deviation of the layer motion from parametric models (Sec. 5.3.1), depth ordering and temporal consistency of layers (Sec. 5.3.2), and regions of occlusion and disocclusion (Sec. 5.3.3).

5.3.1. Roughness in Layers.

Our approach is inspired by the smoothness in layers model by Weiss [**186**]. Given a sequence of images $\mathbf{I}_t, 1 \leq t \leq T$, we model the evolution from the current frame $\mathbf{I}_t$, to the subsequent frame $\mathbf{I}_{t+1}$, via K locally smooth, but potentially globally complex, flow fields. Let $\mathbf{u}_{tk}$ and $\mathbf{v}_{tk}$ denote the horizontal and vertical flow fields, respectively, for layer k at time t. The corresponding flow vector for pixel $p = (i, j)$ is then denoted by (u^p_{tk}, v^p_{tk}).

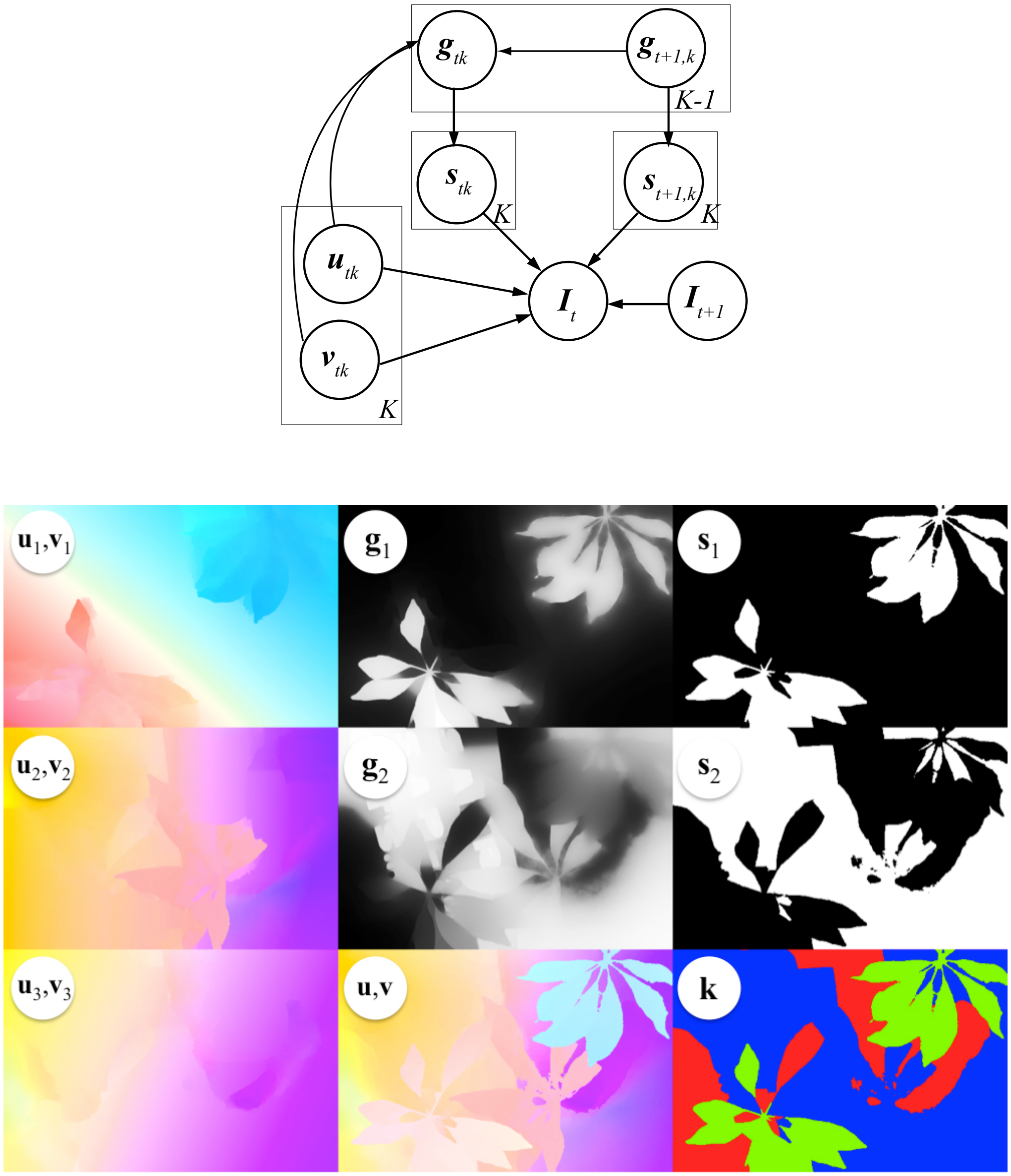


Figure 5.3. *Top:* Graphical representation for the proposed layered model. *Bottom:* Illustration of variables from the graphical model for the "Schefflera" sequence. Labeled sub-images correspond to nodes in the graph. The left column shows the flow fields for three layers, color coded as in [**16**]. The **g** and **s** images illustrate the reasoning about layer ownership (see text). The composite flow field $(\mathbf{u}, \mathbf{v})$ and layer labels $(\mathbf{k})$ are also shown.

Each layer's flow field is drawn from a distribution chosen to encourage piecewise smooth motion. For example, a pairwise Markov random field (MRF) would model the horizontal flow field as

$$\Pr(\mathbf{u}_{tk}) \propto \exp\{-E_{\mathrm{mrf}}(\mathbf{u}_{tk})\} = \exp\left\{ -\frac{1}{2}\sum_{p}\sum_{q\in\Gamma_p} \rho_s(u_{tk}^p - u_{tk}^q)\right\}, \tag{39}$$

where the set Γ_p contains the spatial neighbors of pixel p, often its four nearest neighbors. The potential $\rho_s(\cdot)$ is some robust function [**26**] that encourages smoothness, but allows occasional significant deviations. The vertical flow field $\mathbf{v}_{tk}$ can then be modeled via an independent MRF prior as in Eq. (39), as justified by the statistics of natural flow fields [**133**].

While such MRF priors are flexible, they capture very little dependence between pixels separated by even moderate image distances. In contrast, real scenes exhibit coherent motion over large scales, due to the motion of (partially) rigid objects in the world. To capture this, we associate an affine (or planar) motion model, with parameters θ_{tk}, to each layer k at frame t. We then use an MRF to allow piecewise smooth *deformations* from the globally parametric assumptions of affine motion:

$$E_{\text{aff}}(\mathbf{u}_{tk}, \theta_{tk}) = \frac{1}{2} \sum_p \sum_{q \in \Gamma_p} \rho_s\Big((u^p_{tk} - \bar{u}^p_{\theta_{tk}}) - (u^q_{tk} - \bar{u}^q_{\theta_{tk}})\Big). \tag{40}$$

Here, $\bar{u}^p_{\theta_{tk}}$ denotes the horizontal motion predicted for pixel p by an affine model with parameters θ_{tk}.

$$\bar{u}^p_{\theta_{tk}} = a^1_{tk} + a^2_{tk} \cdot i + a^3_{tk} \cdot j. \tag{41}$$

Unlike classical models that assume layers are globally well fit by a single affine motion [**30, 179**], this prior allows significant, locally smooth deviations from rigidity. Unlike the basic smoothness prior of Eq. (39), this semiparametric construction allows effective global reasoning about non-contiguous segments of partially occluded objects. More sophisticated flow deformation priors may also be used, such as those based on robust non-local terms [**158, 188**].

5.3.2. Layer Support and Spatial Contiguity. We assume that a single, unique layer is observable at each location and that the observed motion of that pixel is determined by its assigned layer. We define the layers to be ordered with respect to the camera, so that the layer k occludes the layers $k' > k$. Analogous to level set representations[**121**], the binary visibility mask $\mathbf{s}_{tk}$ of the layer k is determined by sequentially thresholding $\mathbf{g}_{tk}$

$$s^p_{tk} = \begin{cases} \delta(g^p_{tk} > 0) \prod_{k'=1}^{k-1} \delta(g^p_{tk'} \leq 0), & 1 \leq k < K \\ \prod_{k'=1}^{K-1} \delta(g^p_{tk'} \leq 0), & k = K. \end{cases} \tag{42}$$

Note that only a single layer is visible at each pixel p because $\sum_{k=1}^{K} s^p_{tk} = 1$. The layer K is essentially a background layer that captures all pixels not assigned to the first $K-1$ layers. For this reason, only $K-1$ support functions $\mathbf{g}_{tk}$ are needed (see Figure 5.3). These visibility masks provide a segmentation of the scene. A pixel p at frame t belongs to the k^p_{t*}th layer whose visibility mask is 1 at p.

We capture the spatial coherence via a Gaussian conditional random field in which edge weights are modulated by local differences in color:

$$E_{\text{space}}(\mathbf{g}_{tk}) = \frac{1}{2} \sum_p \sum_{q \in \Gamma_p} w^p_q (g^p_{tk} - g^q_{tk})^2, \tag{43}$$

where

$$w^p_q = \max\Big\{ \exp\Big\{ -\frac{1}{2\sigma_c^2} ||\mathbf{I}^p_t - \mathbf{I}^q_t||^2 \Big\}, \delta_c \Big\}, \tag{44}$$

in which $\mathbf{I}_t^p$ denotes the CIE Lab color vector at pixel p. The threshold $\delta_c > 0$ adds robustness to large color changes in the internal object texture. The temporal coherence of surfaces is then encouraged via a corresponding Gaussian MRF:

$$E_{\text{time}}(\mathbf{g}_{tk}, \mathbf{g}_{t+1,k}, \mathbf{u}_{tk}, \mathbf{v}_{tk}) = \sum_p \sum_{q \in \mathcal{N}_{tk}^p} (g_{tk}^p - g_{t+1,k}^q)^2, \tag{45}$$

where the set $\mathcal{N}_{tk}^p$ is the same as the discrete model and contains the corresponding pixel at frame $t+1$ according to the flow field of the kth layer at frame t. Critically, this energy function uses the corresponding flow field to non-rigidly align the layers at subsequent frames. Compared with the discrete model, the continuous support functions naturally allows the sub-pixel interpolation. By allowing smooth deformation of the support functions $\mathbf{g}_{tk}$, we allow layer support to evolve over time, as opposed to transforming a single rigid template [**81**].

Our model of the layer coherence is inspired by a recent method for image segmentation, based on spatially dependent Pitman-Yor processes [**157**]. That work makes connections between layered occlusion processes and *stick breaking* representations of nonparametric Bayesian models. By assigning appropriate stochastic priors to layer thresholds, the Pitman-Yor model captures the power law statistics of natural scene partitions and infers an appropriate number of segments for each image.

5.3.3. Depth Ordering and Occlusion Reasoning. The preceding generative process defines a set of K ordered layers, with corresponding flow fields $\mathbf{u}_{tk}, \mathbf{v}_{tk}$ and visibility masks $\mathbf{s}_{tk}$. Recall that the layer visibility masks $\mathbf{s}$ are a deterministic function (threshold) of the underlying layer support functions $\mathbf{g}$ (see Eq. (42)). To consistently reason about occlusions, we examine the layer visibilities s_{tk}^p and $s_{t+1,k}^q, q =\in \mathcal{N}_{tk}^p$ at locations corresponded by the underlying flow fields. This leads to a far richer occlusion model than standard spatially independent outlier processes: geometric consistency is enforced via the layered sequence of flow fields.

Let $\mathbf{I}_t^p$ denote an observed image feature for pixel p; we work with a filtered version of the intensity images to provide some invariance to illumination changes. If the visible layer for pixel p at time t remains visible at time $t+1$, i.e., $s_{tk}^p = s_{t+1,k}^q = 1$, the image observations are modeled using a standard brightness (or, here, feature) constancy assumption. Otherwise, that pixel has become occluded, and is instead generated from a uniform distribution. The image likelihood model can then be written as

$$\begin{aligned} &\Pr(\mathbf{I}_t \mid \mathbf{I}_{t+1}, \mathbf{u}_t, \mathbf{v}_t, \mathbf{g}_t, \mathbf{g}_{t+1}) \propto \exp\{-E_{\text{data}}(\mathbf{u}_t, \mathbf{v}_t, \mathbf{g}_t, \mathbf{g}_{t+1})\} \qquad (46)\\ &\qquad = \exp\Big\{ -\sum_{k=1}^K \sum_p \sum_{q \in \mathcal{N}_{tk}^p} \Big(\rho_d(\mathbf{I}_t^p - \mathbf{I}_{t+1}^q) s_{tk}^p s_{t+1,k}^q + \lambda_d s_{tk}^p (1 - s_{t+1,k}^q)\Big)\Big\}, \end{aligned}$$

where $\rho_d(\cdot)$ is a robust potential function and the constant λ_d arises from the difference of the log normalization constants for the robust and uniform distributions. With algebraic simplifications, the data error term can be written as

$$E_{\text{data}}(\mathbf{u}_t, \mathbf{v}_t, \mathbf{g}_t, \mathbf{g}_{t+1}) = \sum_{k=1}^K \sum_p \sum_{q \in \mathcal{N}_{tk}^p} \Big(\rho_d(\mathbf{I}_t^p - \mathbf{I}_{t+1}^q) - \lambda_d\Big) s_{tk}^p s_{t+1,k}^q, \tag{47}$$

up to an additive, constant multiple of λ_d. The shifted potential function $(\rho_d(\cdot) - \lambda_d)$ represents the change in energy when a pixel transitions from an occluded to a visible configuration. Note that occlusions have higher likelihood only for sufficiently large discrepancies in matched image features and can only occur via a corresponding change in layer visibility.

5.4. Posterior Inference from Image Sequences

Considering the full generative model defined in Sec. 5.3, the *maximum a posteriori* (MAP) estimation for a T frame image sequence is equivalent to the minimization of the following energy function:

$$E(\mathbf{u},\mathbf{v},\mathbf{g},\theta) = \sum_{t=1}^{T-1}\Bigg\{E_{\text{data}}(\mathbf{u}_t,\mathbf{v}_t,\mathbf{g}_t,\mathbf{g}_{t+1}) + \sum_{k=1}^{K}\lambda_a(E_{\text{aff}}(\mathbf{u}_{tk},\theta_{tk}) + E_{\text{aff}}(\mathbf{v}_{tk},\theta_{tk})) + \sum_{k=1}^{K-1}\lambda_b E_{\text{space}}(\mathbf{g}_{tk}) + \lambda_c E_{\text{time}}(\mathbf{g}_{tk},\mathbf{g}_{t+1,k},\mathbf{u}_{tk},\mathbf{v}_{tk})\Bigg\} + \sum_{k=1}^{K-1}\lambda_b E_{\text{space}}(\mathbf{g}_{Tk}). \tag{48}$$

Here λ_a, λ_b, and λ_c are weights controlling the relative importance of the affine, spatial, and temporal terms respectively. Simultaneously inferring flow fields, layer support maps, and depth ordering is a challenging process; our approach is summarized below.

5.4.0.1. *Relaxation of the Layer Assignment Process.* Due to the non-differentiability of the threshold process that determines assignments of regions to layers, direct minimization of the overall energy function Eq. (48) is challenging. For a related approach to image segmentation, a mean field variational method has been proposed [**157**]. However, that segmentation model is based on a much simpler, spatially factorized likelihood model for color and texture histogram features. Generalization to the richer flow likelihoods considered here raises significant complications.

Instead, we relax the hard threshold assignment process using the logistic function $\sigma(g) = 1/(1+\exp(-g))$. Applied to the sequential thresholding process by Eq. (42), this induces the following soft layer assignments:

$$\tilde{s}_{tk}^{p} = \begin{cases} \sigma(\lambda_e g_{tk}^{p})\prod_{k'=1}^{k-1}\sigma(-\lambda_e g_{tk'}^{p}), & 1 \le k < K, \\ \prod_{k'=1}^{K-1}\sigma(-\lambda_e g_{tk'}^{p}), & k = K. \end{cases} \tag{49}$$

Note that $\sigma(-g) = 1-\sigma(g)$, and $\sum_{k=1}^{K}\tilde{s}_{tk}^{p} = 1$ for any g_{tk} and constant $\lambda_e > 0$.

Substituting these soft assignments $\tilde{s}_{tk}^{p}$ for s_{tk}^{p} in the data error term Eq.. (47), we obtain a differentiable energy function that can be optimized via gradient-based methods. A related relaxation underlies the classic backpropagation algorithm for neural network training. See *Appendix B* for the gradient formulae of the energy function w. r. t. the flow fields and the support functions.

5.5. Experimental Results

5.5.1. Implementation Details. We pre-process the input images using the structure texture decomposition method developed by [**185**] and described in Sec. 4.3.1. We compute the initial flow field using the **Classic+NL** method presented in Chapter 4 and fit K affine motion fields to the initial forward flow field. The fitting method is similar to K-means, where we cluster the flow vectors and fit the affine parameters of each cluster. A pixel is visible at the layer that best explains its motion and invisible at the other layers. To avoid local minima, we perform 25 independent runs of the fitting method and select the result with the lowest fitting error. Warping the resultant segmentation using the backward flow field produces the initial segmentation of the next frame.

Table 5.1. Average end-point error (EPE) on the Middlebury optical flow benchmark *training* set. * means **3layers** and ** means **3layers w/ WMF** in the table.

	Avg.	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3
Weiss [186]	0.487	0.510	0.179	0.249	0.236	0.221	0.608	0.614	1.276
Classic++	0.285	0.271	0.128	0.153	0.081	0.139	0.614	0.336	0.555
Classic+NL	0.221	0.238	0.131	0.152	0.073	0.103	0.468	0.220	0.384
1layer	0.248	0.243	0.144	0.175	0.095	0.125	0.504	0.279	0.422
2layers	0.212	0.219	0.147	0.169	0.081	0.098	0.376	0.236	0.370
3layers	0.200	0.212	0.149	0.173	0.073	0.090	0.343	0.220	0.338
*** w/ WMF**	0.195	0.211	0.150	0.161	0.067	0.086	0.331	0.210	0.345
*** * 4 frames**	0.190	0.211	0.151	0.157	0.067	0.084	0.330	0.207	0.311
*** * all**	0.194	0.211	0.146	0.162	0.067	0.089	0.331	0.212	0.336
*** * C++Init**	0.203	0.212	0.151	0.161	0.066	0.087	0.339	0.210	0.396
4layers	0.194	0.197	0.148	0.159	0.068	0.088	0.359	0.230	0.300

To convert the hard segmentation into the initial support functions, the kth ($k < K$) support function takes value 1.5 at pixels visible at the kth layer and -1.5 otherwise. Around occlusion/disocclusion regions, the layer assignment tends to change from one frame to the next. We detect pixels where the layer assignments, aligned by the initial flow field, disagree. For these pixels we divide their initial support function values by 10 to represent our uncertainty about the initial layer assignment in these occlusion/disocclusion regions. The initial motion of the visible pixels in each layer is the same as the initial flow field from **Classic+NL**, while the motion of the invisible pixels is interpolated by the fitted affine motion to the flow field.

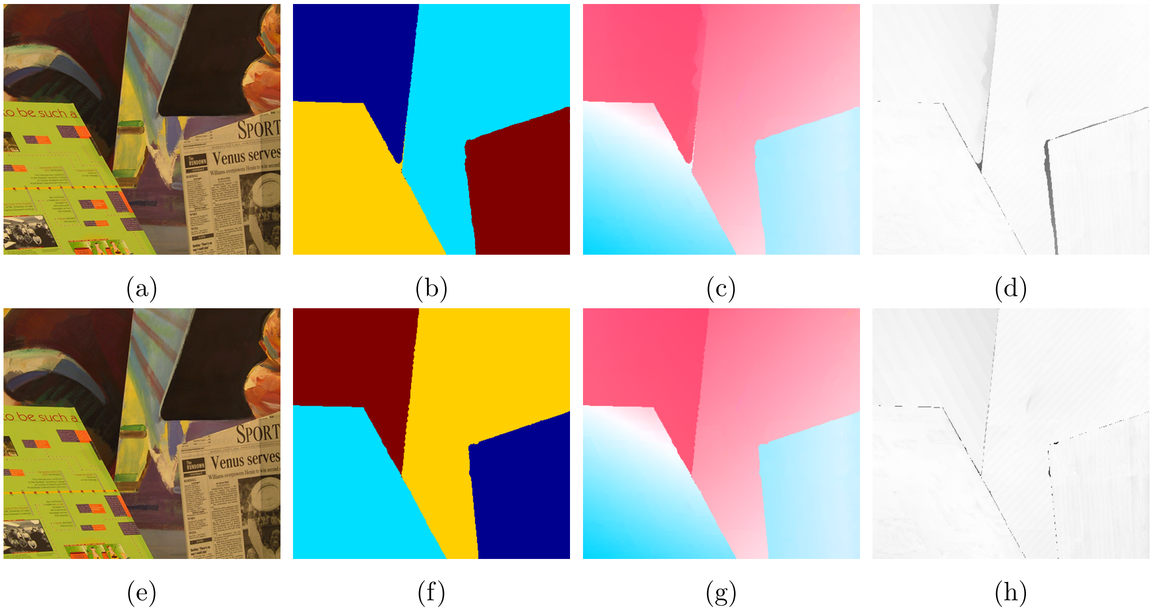
We then use a two-level Gaussian pyramid (downsampling factor 0.8) and perform a fairly standard incremental estimation of the flow fields for each layer. At each level, we perform 20 incremental warping steps and during each step alternately solve for the support functions and the flow estimates. In the end, we threshold the support functions to compute a hard segmentation, and obtain the final flow field by selecting the flow field from the appropriate layers.

Occluded regions are determined by inconsistencies between the hard segmentations at subsequent frames, as matched by the final flow field. We would ideally like to compare layer initializations based on all permutations of the initial flow vector clusters, but this would be computationally intensive for large K. Instead we compare two orders: a fast-to-slow order appropriate for rigid scenes, and an opposite slow-to-fast order (for variety and robustness). We illustrate automatic selection of the preferred order for the "Venus" sequence in Figure 5.4.

The parameters for all experiments are set to $\lambda_a = 3$, $\lambda_b = 30$, $\lambda_c = 4$, $\lambda_d = 9$, $\lambda_e = 2$, $\sigma_i = 12$, and $\delta_c = 0.004$. We use the generalized Charbonnier penalty function $\rho(x) = (x^2 + \epsilon^2)^a$ with $\epsilon = 0.001$ and $a = 0.45$, introduced in Chapter 4, for $\rho_d(\cdot)$ in the data term, E_{data}, and $\rho_s(\cdot)$ in the spatial flow term, E_{aff}. Optimization takes about 5 hours for the two-frame "Urban" sequence using our Matlab implementation.

5.5.2. Results on the Middlebury Benchmark.

5.5.2.1. *Training Set.* As a baseline, we implement the smoothness in layers model [**186**] using modern techniques, and obtain an average training end-point error (EPE) of 0.487. This is reasonable


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

(e) (f) (g) (h)

Figure 5.4. Results on the "Venus" sequence with 4 layers. The two background layers move faster than the two foreground layers, and the solution with the correct depth ordering has lower energy and smaller error. (a) First frame. (b-d) Fast-to-slow ordering: EPE 0.252 and energy -1.786×10^6. Left to right: motion segmentation, estimated flow field, and absolute error of estimated flow field. (e) Second frame. (f-h) Slow-to-fast ordering: EPE 0.195 and energy -1.808×10^6. Darker indicates larger flow field errors in (d) and (h).

Table 5.2. Average end-point error (EPE) on the Middlebury optical flow benchmark *test set*. The ranking information was at the publication of our conference paper [**160**] (October 2010); the EPE ranks for **Classic+NL** and **Layers++** are 16.2 and 9.2 at the writing of the dissertation (July 2012).

	Rank	**Avg.**	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
EPE										
Layers++	4.3	0.270	0.08	0.19	0.20	0.13	0.48	0.47	0.15	0.46
Classic+NL	6.5	0.319	0.08	0.22	0.29	0.15	0.64	0.52	0.16	0.49
EPE in boundary regions										
Layers++		0.560	0.21	0.56	0.40	0.58	0.70	1.01	0.14	0.88
Classic+NL		0.689	0.23	0.74	0.65	0.73	0.93	1.12	0.13	0.98

but not competitive with state-of-the-art methods. The proposed model with 1 to 4 layers produces average EPEs of 0.248, 0.212, 0.200, and 0.194, respectively (see Table 5.1). The one-layer model is similar to **Classic+NL**, but has a less sophisticated (more local) model of the flow within that layer. It thus performs worse than the **Classic+NL** initialization; the performance improvements allowed by additional layers demonstrate the benefits of a layered model.

Accuracy is improved by applying a 15 × 15 *weighted median filter* (WMF) [**158**] to the flow fields of each layer during the iterative warping step (EPE for 1 to 4 layers: 0.231, 0.204, 0.195, and 0.193). As shown in Chapter 4, weighted median filtering can be interpreted as a non-local spatial smoothness term in the energy function that integrates flow field information over a larger spatial neighborhood.

The “correct” number of layers for a real scene is often not well defined (consider the “Grove3” sequence, for example). We use a restricted number of layers, and model the remaining complexity of the flow within each layer via the roughness-in-layers spatial term and the WMF. As the number of layers increases, the complexity of the flow within each layer decreases, and consequently the need for WMF also decreases; note that the difference in EPE for the 4-layer model with and without WMF is insignificant. For the remaining experiments we use the version with WMF.

To test the sensitivity of the result to the initialization, we also initialized with **Classic++** (**C++Init** in Table 5.1), a good, but not top, non-layered method [**158**]. The average EPE for 1 to 4 layers increases to 0.248, 0.206, 0.203, and 0.198, respectively. While the one-layer method gets stuck in poor local minima on the “Grove3” and “Urban3” sequences, models with additional layers are more robust to the initialization.

5.5.2.2. *Test Set.* For evaluation, we focus on a model with 3 layers (denoted **Layers++** in the Middlebury public table). On the Middlebury test set it has an average EPE of 0.270 and average angular error (AAE) of 2.556; this was the lowest among all tested methods [**16**] at the writing of our conference paper [**160**] (October 2010). **Layers++** is ranked 3$^{\text{rd}}$ in both EPE and AAE at the writing of the dissertation (July 2012). Table 5.2 summarizes the results for individual test sequences. The layered model is particularly accurate at motion boundaries, probably due to the use of layer-specific motion models, and the explicit modeling of occlusion in E_{data} (Eq. (47)).

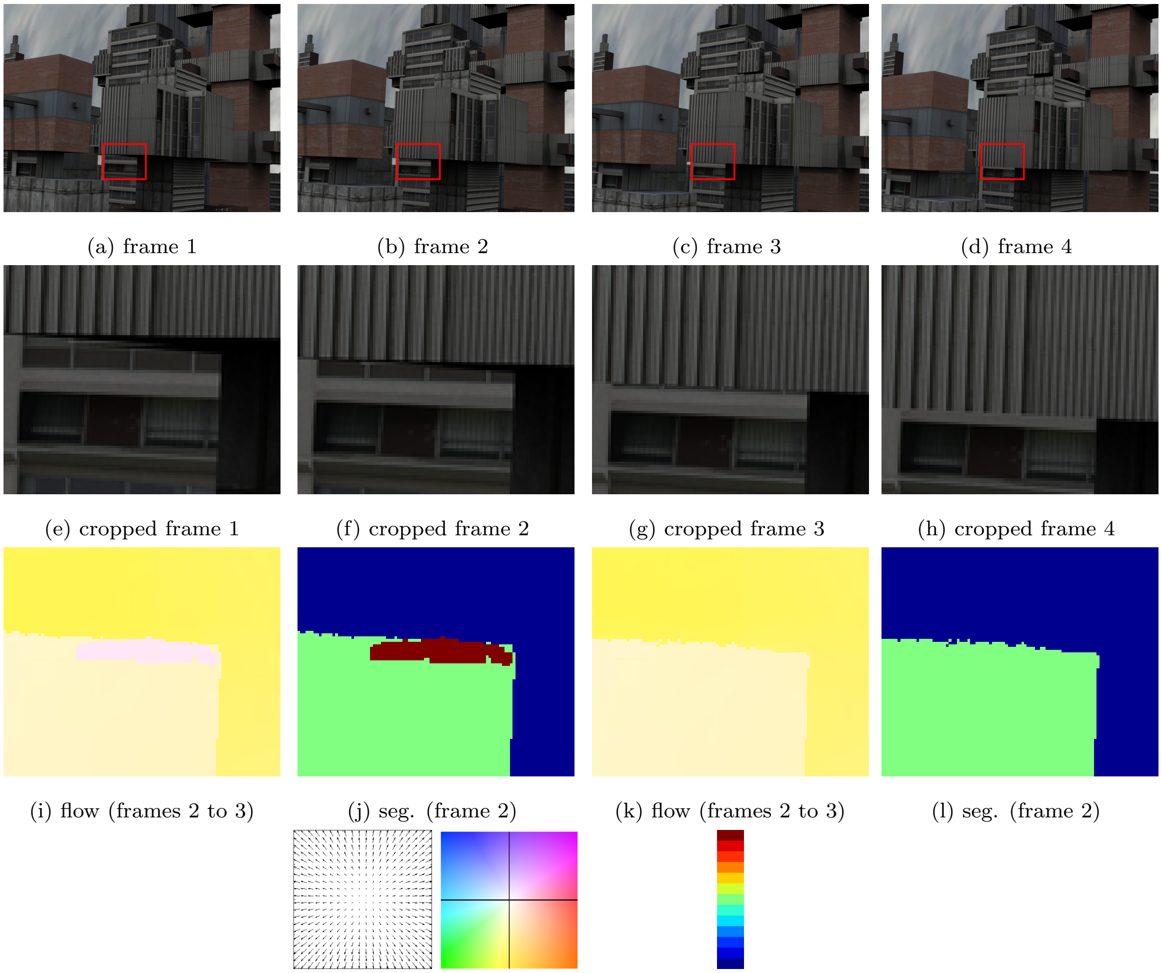
We evaluate several variants of the proposed method. The results above are obtained with 20 warping steps per level, which is computationally expensive. Using 3 warping steps has slightly better overall performance for 1-3 layers, but 20 warping steps produces more accurate results in motion boundary regions.

To evaluate the method’s sensitivity to the initialization, we compare the energy of the final solutions with initial flow fields from the **Classic++** and the **Classic+NL** methods. As shown in Table 5.3, solutions with the **Classic++** initialization have similar energy as those with the **Classic+NL** initialization. Table 5.4 shows that the average EPE obtained from both initializations is also similar. This suggests that the method works as long as the initialization is sensible. We expect that our current inference methods would not be able to recover from a really poor initialization. Finally average EPE and average AAE for all the test sequences are shown in Table 5.2.

Changing the total nubmer of frames to 4 results in a small but consistent improvment over 2. Figure 5.5 shows a case where using 4 frames resolves an ambiuity in layer assignment by only 2 frames.

Table 5.3. Energy ($\times 10^6$) of the solutions obtained by the proposed method with three layers. The energy is shown for all the sequences in the *training* set using two different initializations.

	Venus	Dimetr-odon	Hydr-angea	Rubber-Whale	Grove2	Grove3	Urban2	Urban3
Classic+NL Init	-1.814	-2.609	-2.370	-3.039	-2.679	-1.979	-3.198	-3.044
Classic++ Init	-1.814	-2.613	-2.369	-3.039	-2.680	-1.974	-3.200	-2.998



(a) frame 1 (b) frame 2 (c) frame 3 (d) frame 4

(e) cropped frame 1 (f) cropped frame 2 (g) cropped frame 3 (h) cropped frame 4

(i) flow (frames 2 to 3) (j) seg. (frame 2) (k) flow (frames 2 to 3) (l) seg. (frame 2)

(m) flow color key (n) depth ordering key

Figure 5.5. Occlusion reasoning using frames 2 and 3 (i-j) is hard (detail from Urban3); enforcing temporal coherence of the support functions using 4 frames significantly reduces the errors in both the flow field and the segmentation (k-l). The flow field is from frame 2 to frame 3 and the segmentation is for frame 2.

5.5.2.3. *Visual Comparison.* Figures 5.6 and 5.8 show results for the 3-layer model on the Middlebury training and test sequences. Notice that the layered model produces a motion segmentation that captures the major structure of the scene, and the layer boundaries correspond well to static image edges. It detects most occlusion regions and interpolates their motion reasonably well. Several sequences show significant improvement due to the global reasoning provided by the layered model. On the training "Grove3" sequence, the proposed method correctly identifies many holes between the branches and leaves as background. It also associates the branch at the bottom right corner with branches in the center. As the branch moves beyond the image boundary, the layered model interpolates its motion using long-range correlation with the branches in the center. In contrast, the single-layered approach incorrectly interpolates from local background regions. The "Schefflera" result illustrates how the layered method can separate foreground objects from the background (e.g., the leaves in the top right corner), and thereby reduce errors made by single-layer approaches such as **Classic+NL**.

Table 5.4. Average end-point error (EPE) on the Middlebury *training* set.

	Avg.	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3
Weiss [**186**]	0.487	0.510	0.179	0.249	0.236	0.221	0.608	0.614	1.276
Classic++	0.285	0.271	0.128	0.153	0.081	0.139	0.614	0.336	0.555
Classic+NL	0.221	0.238	0.131	0.152	0.073	0.103	0.468	0.220	0.384
20 warping steps (WS)									
1layer	0.248	0.243	0.144	0.175	0.095	0.125	0.504	0.279	0.422
2layers	0.212	0.219	0.147	0.169	0.081	0.098	0.376	0.236	0.370
3layers	0.200	0.212	0.149	0.173	0.073	0.090	0.343	0.220	0.338
4layers	0.194	0.197	0.148	0.159	0.068	0.088	0.359	0.230	0.300
5layers	0.196	0.195	0.151	0.169	0.063	0.086	0.345	0.211	0.351
20 WS w/ WMF : overall									
1layer	0.231	0.235	0.144	0.155	0.075	0.106	0.462	0.245	0.426
2layers	0.204	0.217	0.149	0.156	0.070	0.090	0.357	0.219	0.373
3layers	0.195	0.211	0.150	0.161	0.067	0.086	0.331	0.210	0.345
4layers	0.193	0.195	0.150	0.155	0.064	0.087	0.351	0.222	0.321
5layers	0.197	0.196	0.149	0.173	0.065	0.087	0.347	0.214	0.346
20 WS w/ WMF: boundary region									
1layer	0.545	0.617	0.222	0.379	0.218	0.295	0.868	0.703	1.061
2layers	0.468	0.456	0.250	0.390	0.206	0.231	0.652	0.670	0.889
3layers	0.451	0.441	0.252	0.409	0.197	0.220	0.596	0.610	0.885
4layers	0.436	0.348	0.250	0.393	0.182	0.230	0.636	0.647	0.801
5layers	0.437	0.345	0.250	0.438	0.182	0.221	0.626	0.602	0.834
3 WS w/ WMF: overall									
1layer	0.219	0.231	0.119	0.152	0.074	0.097	0.454	0.230	0.394
2layers	0.195	0.211	0.122	0.159	0.070	0.084	0.364	0.205	0.346
3layers	0.190	0.212	0.128	0.163	0.066	0.080	0.347	0.206	0.321
4layers	0.194	0.192	0.132	0.158	0.063	0.081	0.365	0.227	0.337
5layers	0.196	0.192	0.136	0.159	0.063	0.080	0.362	0.224	0.349
3 WS w/ WMF: boundary region									
1layer	0.551	0.642	0.218	0.385	0.229	0.291	0.859	0.710	1.074
2layers	0.472	0.464	0.236	0.414	0.218	0.238	0.672	0.662	0.876
3layers	0.463	0.441	0.254	0.428	0.207	0.221	0.630	0.632	0.891
4layers	0.465	0.353	0.264	0.415	0.187	0.229	0.665	0.671	0.934
5layers	0.466	0.354	0.271	0.418	0.191	0.220	0.653	0.662	0.962
20 WS w/ WMF: **Classic++** init									
1layer	0.248	0.232	0.144	0.155	0.079	0.107	0.523	0.261	0.487
2layers	0.206	0.218	0.149	0.156	0.072	0.090	0.373	0.218	0.372
3layers	0.203	0.212	0.151	0.161	0.066	0.087	0.339	0.210	0.396
4layers	0.198	0.195	0.149	0.155	0.064	0.087	0.342	0.229	0.360
5layers	0.192	0.194	0.148	0.161	0.063	0.085	0.326	0.231	0.327

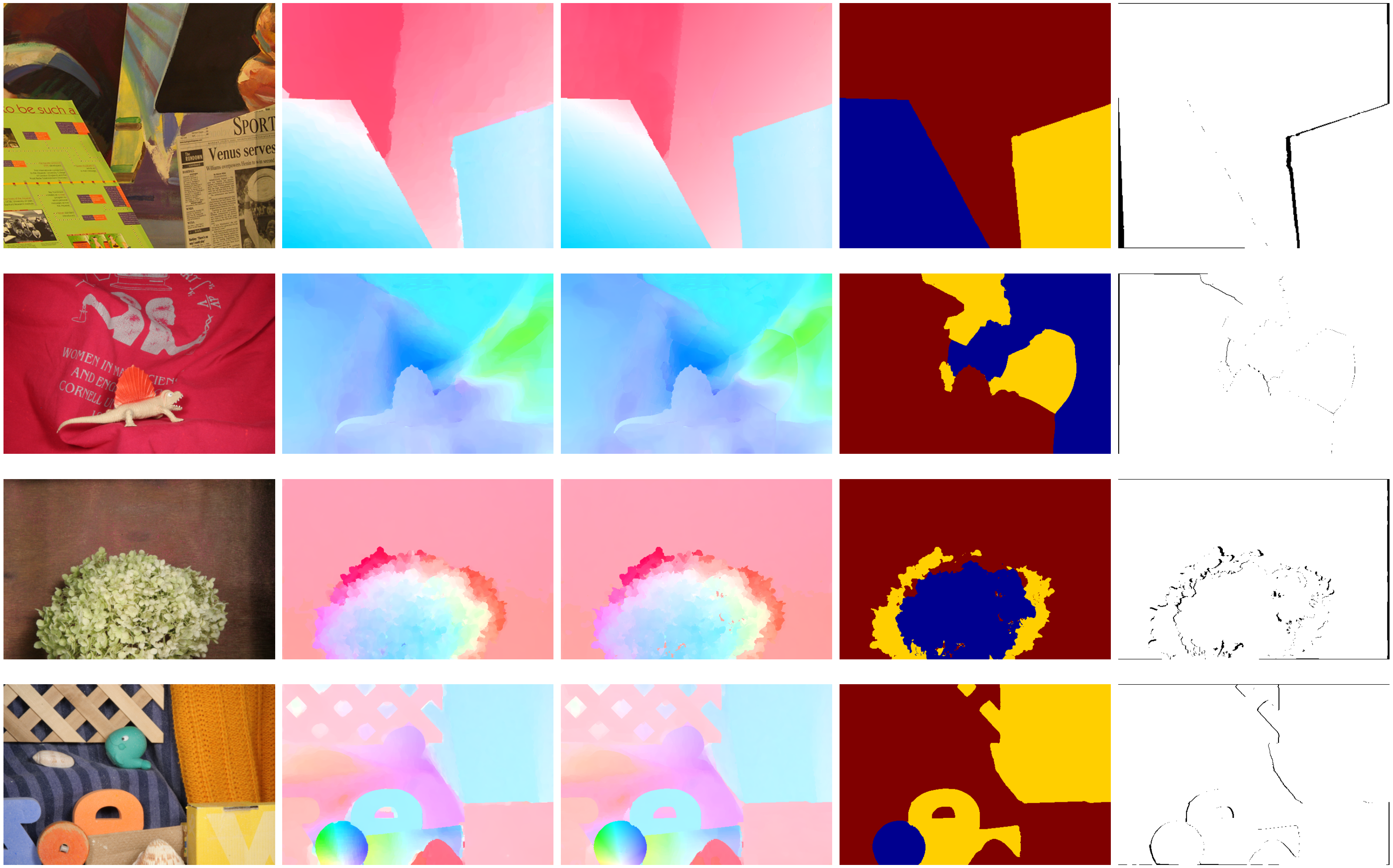


Figure 5.6. Results on the Middlebury *training* set. Left to right: first image, initial flow field given by Classic+NL, final flow field, motion segmentation, and detected occlusions (black). Top to bottom: “Venus”, “Dimetrodon”, “Hydrangea”, and “RubberWhale”. Best viewed in color and better enlarged for comparing the flow fields. Color key for the depth ordering is the same as Figure 5.5 (blue is close and red is far).

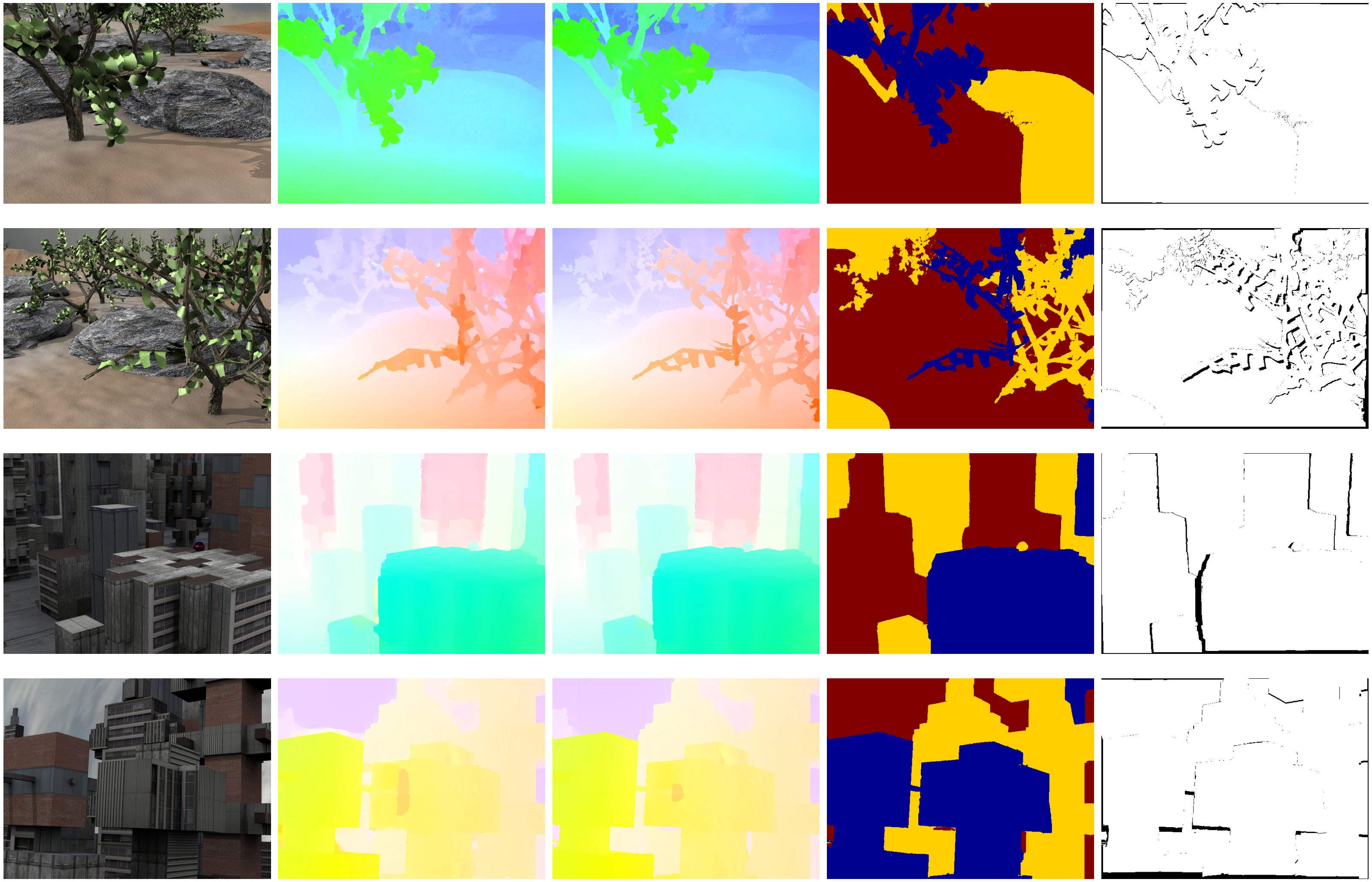


Figure 5.7. Results on the Middlebury *training* set. Left to right: first image, initial flow field given by Classic+NL, final flow field, motion segmentation, and detected occlusions (black). Top to bottom: "Grove2", "Grove3", "Urban2", and "Urban3". Best viewed in color and better enlarged for comparing the flow fields. Color key for the depth ordering is the same as Figure 5.5 (blue is close and red is far).

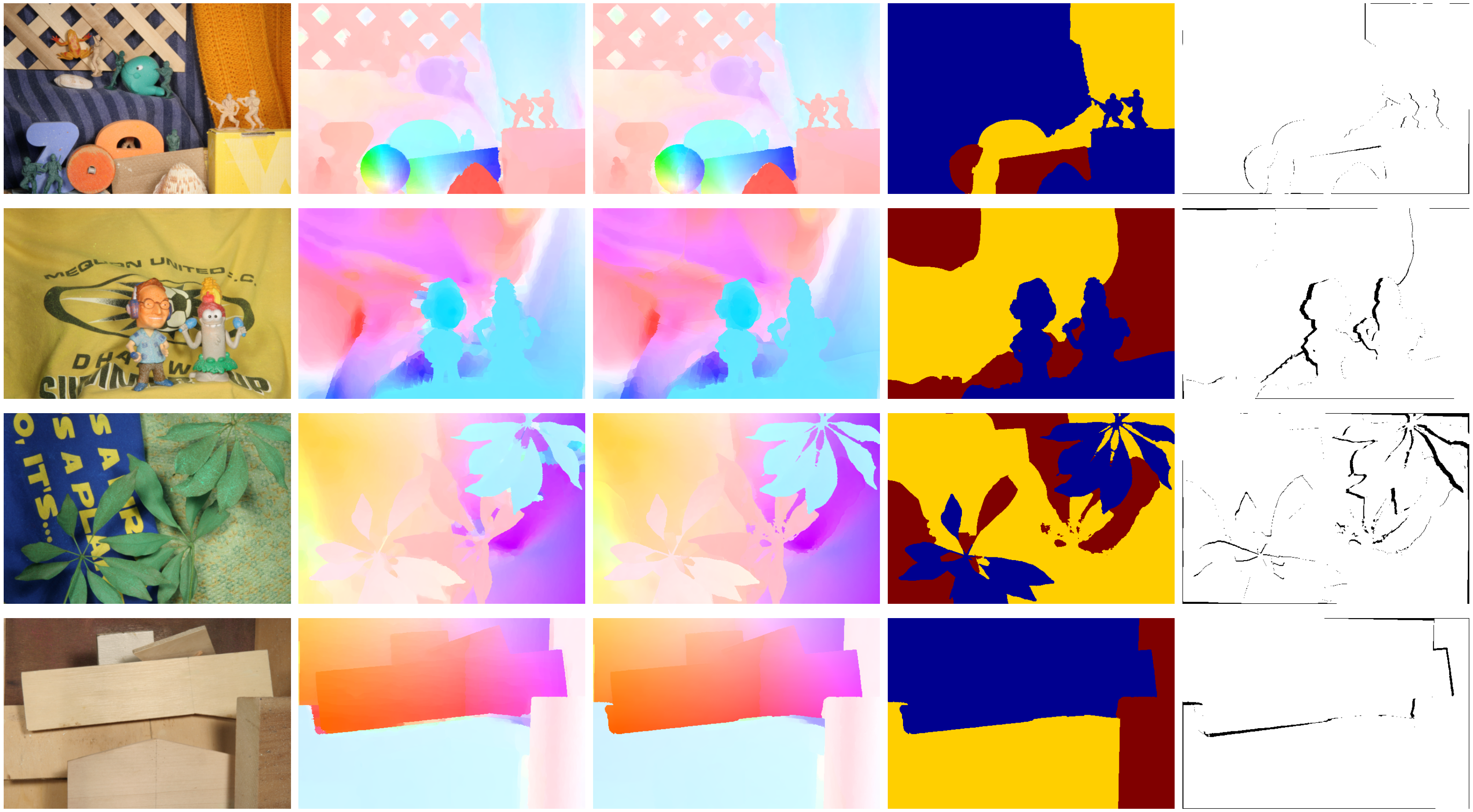


Figure 5.8. Results on the Middlebury *test* set. Left to right: first image, initial flow field given by Classic+NL, final flow field, motion segmentation, and detected occlusions (black). Best viewed in color and better enlarged for comparing the flow fields. Top to bottom: “Army”, “Mequon”, “Schefflera”, and “Wooden”. Color key for the depth ordering is the same as Figure 5.5 (blue is close and red is far).

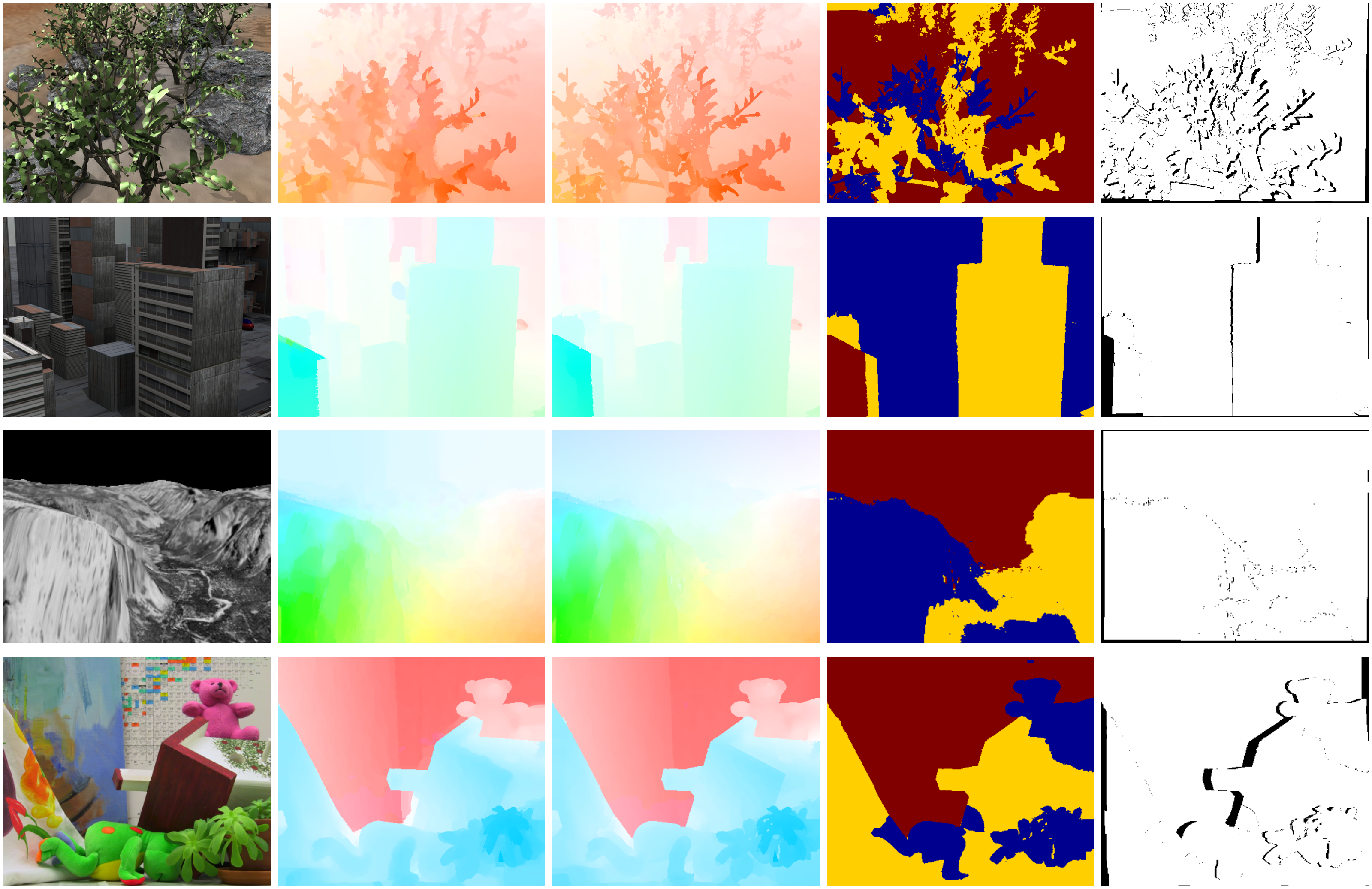


Figure 5.9. Results on the Middlebury *test* set. Left to right: first image, initial flow field given by Classic+NL, final flow field, motion segmentation, and detected occlusions (black). Best viewed in color and better enlarged for comparing the flow fields. Top to bottom: “Grove”, “Urban”, “Yosemite”, and “Teddy”. Color key for the depth ordering is the same as Figure 5.5 (blue is close and red is far).

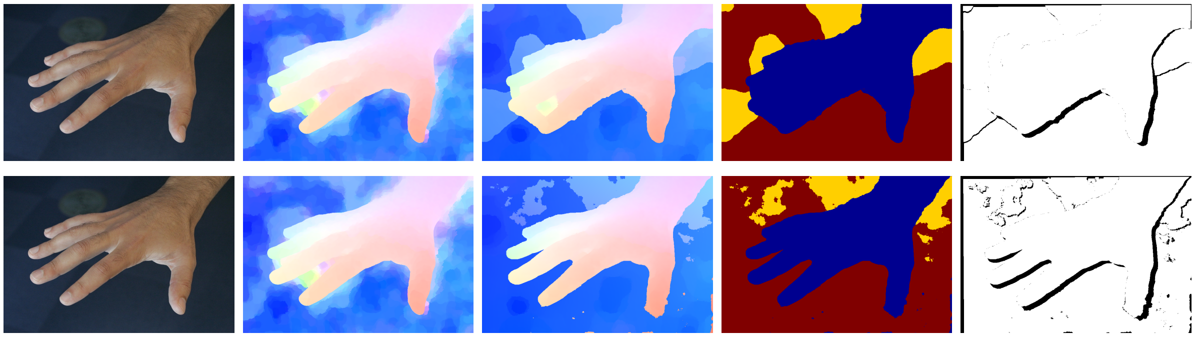


Figure 5.10. Results on the "Hand" sequence. Top: using same parameters as those used for the Middlebury data set (EPE 2.754). Bottom: using parameters tuned for hand (EPE 1.909). Left to right: first image, initial flow field given by **Classic+NL**, final flow field, motion segmentation, and detected occlusions (black) by **Layers++**. Best viewed in color. Color key for the depth ordering is the same as Figure 5.5 (blue is close and red is far).

5.5.3. Results on the "Hand" Sequence. While the method performs well on the Middlebury evaluation, how well do the results generalize to other sequences? To find out, we apply the proposed model with 3 layers to the challenging "Hand" sequence [**99**], as shown in Figure 5.10. With the parameter settings tuned to the Middlebury training sequences, the proposed model does not recover the regions between fingers (Figure 5.10, top row). With a different parameter setting ($\lambda_d = 5$, and $\lambda_b = 90$), the proposed model can successfully recover the regions between fingers. The EPE for this sequence drops from 2.754 to 1.909. Moreover, note that the model successfully recovers from failures of the initialization in the regions between the fingers.

Table 5.5 compares this new parameter settings with the old settings on the Middlebury training sequences. The new settings produces an average training EPE of 0.215 which is about 10% worse than the optimal results.

This suggests that the proposed method may suffer from over fitting to the Middlebury evaluation. Future work should consider learning the parameters using a more representative data set [**159**] and automatically adapting the parameters to a particular sequence.

Table 5.5. Average end-point error (EPE) on the Middlebury *training* set by the proposed model with 3 layers and two different sets of parameters.

	Avg.	Venus	Dimetr-odon	Hydr-angea	Rubber-Whale	Grove2	Grove3	Urban2	Urban3
$\lambda_b = 10$, $\lambda_d = 9$	0.195	0.211	0.150	0.161	0.067	0.086	0.331	0.210	0.345
$\lambda_b = 90$, $\lambda_d = 5$	0.215	0.210	0.155	0.169	0.071	0.090	0.373	0.273	0.379

5.6. Conclusions and Discussions

We have described a new probabilistic formulation for layered image motion that explicitly models occlusion and disocclusion, depth ordering of layers, and the temporal consistency of the layer segmentation. The approach allows the flow field in each layer to have piecewise smooth deformation from a parametric motion model. Layer segmentation is modeled using an image-dependent support function prior that supports a model of temporal layer continuity over time.

The image data error term takes into account layer occlusion relationships, resulting in increased flow accuracy near motion boundaries. Our method achieves consistently better results than the single-layered **Classic+NL** method on the Middlebury optical flow benchmark while producing meaningful segmentation and occlusion detection results.

We have used a fairly simple inference method for the complicated energy function. In particular, we assume that the number of layers is fixed and the depth ordering is only limited to fast to slow or slow to fast.

CHAPTER 6

Optical Flow Estimation and Layered Segmentation over Time

In this chapter, we will study optimization methods for the proposed layered model. The local inference scheme developed in the previous chapter cannot effectively estimate the number of layers in a scene, or robustly determine the depth ordering of the layers. Furthermore, previous methods have focused on optical flow estimation using two frames. We show that image sequences with more than two frames are necessary to resolve ambiguities in depth ordering at occlusion boundaries; temporal layer consistency makes the reasoning feasible. We propose a novel discrete approximation of the continuous objective in terms of a sequence of depth-ordered MRFs and extend graph-cut optimization methods with new "moves" that make joint layer segmentation and motion estimation feasible. Our optimizer, which mixes discrete and continuous optimization, automatically determines the number of layers and reasons about their depth ordering.

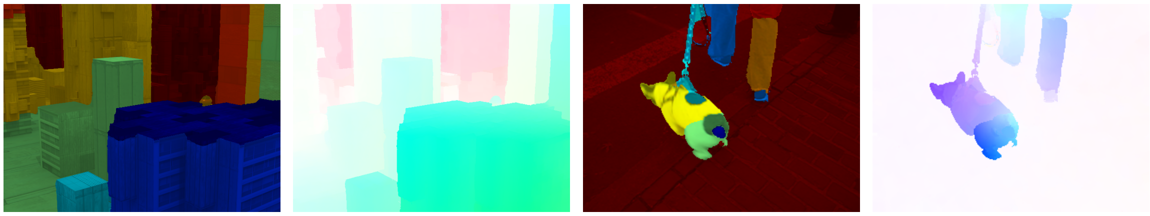


Figure 6.1. The proposed method achieves state-of-the-art optical flow estimation and layer segmentation results.

6.1. Introduction

One key issue is that the layer-structure inference problem is difficult to optimize. Most methods adopt an expectation maximization (EM) style algorithm that is susceptible to local optima. For example, in Chapter 5 we have proposed a generative layered model that combines mixture models with state-of-the-art static image segmentation models [**157**]. This **Layers++** method estimates image motion very accurately as measured by the Middlebury optical flow benchmark [**16**]. However, our gradient-based inference algorithm is susceptible to local optima, resulting in errors in the estimated scene structure and flow field, as illustrated in Figure 6.2.

Overcoming such limitations requires an optimization method that can make large changes to the solution at a single step, a task more suitable for discrete optimization. Hence we propose a discrete layered model based on a sequence of ordered Markov random fields (MRFs). This model, unlike standard Ising/Potts MRFs, cannot be directly solved by "off-the-shelf" optimizers, such as graph cuts. Therefore we develop a sequence of non-standard moves that can simultaneously change

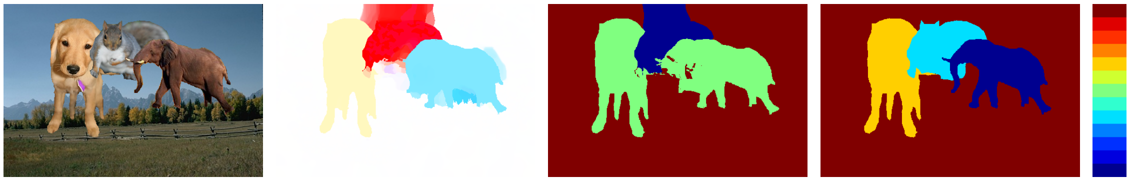


Figure 6.2. A failure case for the **Layers++** method from Chapter 5. Left to right: first image in a pair, initial flow estimate, color coded as in [**16**], segmentation by **Layers++**; segmentation with our proposed **nLayers** method, which automatically determines the number of layers, their depth ordering, and is able to make large changes to the initial flow field to reach a good solution; on the far right is a color key for the ordering of depth layers (blue is close and red is far).

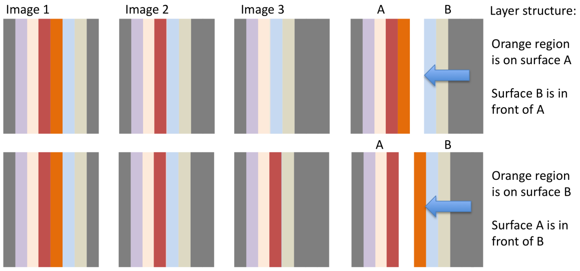


Image 1
Image 2
Image 3
A
B
Layer structure:
Orange region is on surface A
Surface B is in front of A
A
B
Orange region is on surface B
Surface A is in front of B

Figure 6.3. Relative depth ordering can be ambiguous in 2 frames. A third frame enables the motion of the occlusion boundary to be computed. This motion is consistent with the occluding surface, removing the ambiguity. Two cases are shown where image 1 and 2 are the same. In both, surface B moves to the left. With image 3 the ambiguity is resolved because the motion of the occluding contour is known.

the states of several binary MRFs. We also embed continuous flow estimation into the discrete framework to adapt the state space to estimate sub-pixel motion. The resultant discrete-continuous scheme enables us to infer the number of layers and their depth ordering automatically for a sequence.

We evaluate our layer segmentation using the MIT human-assisted motion annotation dataset [**99**]. Our method produces semantically more meaningful segmentations that are also quantitatively more consistent with human labeled ground truth than the continuous-only **Layers++** method. With a reliable layer segmentation and the relative depth ordering obtained with the discrete method, we initialize the more precise **Layers++** continuous model of optical flow. The discrete-continuous approach gives a concrete improvement over a purely continuous optimization that can easily become trapped in local optima.

In summary, our contributions include *a)* formulating a discrete layered model based on a sequence of ordered Ising MRFs and devising a set of non-standard moves to optimize it; *b)* formulating methods for automatically determining the number of layers and their depth ordering for a given sequence; *c)* concretely improving layer segmentation on a set of real-world sequences than the local method in Chapter 5; *d)* demonstrating the benefits of using more frames for optical flow estimation and layered segmentation on the Middlebury optical flow benchmark and the MIT segmentation benchmark.

6.2. Previous Work

Occlusion. Reasoning about occlusion in image sequences dates to the mid 1970's and early 1980's. Early authors (*e.g.,* [**46, 71, 116, 168**]) note that occlusion boundaries move with the occluding surface. We illustrate this in Figure 6.3, similar to the examples in [**167, 168**]. This simple fact is a key reason why two-frame optical flow estimation is fundamentally limited. In a layered model, inferring the wrong depth order results in significant errors at motion boundaries. The idea of using three or more frames has been embodied in recent methods for computing motion boundaries and depth order [**27, 52**] but appears to be missing from recent dense flow estimation methods.

Estimating flow over time. Again, estimation of flow over time has a long history [**24, 115**] yet few methods have demonstrated improved accuracy through temporal consistency of flow. Brox *et al.* [**34**] and Farneback [**51**] demonstrate the benefits of using multiple frames on the "Yosemite" sequence, but the flow field for this sequence is smooth both spatially and temporally. Volz *et al.* [**174**] propose a multi-frame optical flow method that shows improvement using 5 frames over their 2-frame baseline. However their 5-frame method is still less accurate than the top performing 2-frame methods [**160, 197**]. We argue that, while flow fields are not always temporally consistent, the scene structure represented by a layered segmentation is.

Optimization. It is common to alternate optimization between segmentation and motion estimation [**145, 160, 186**]. However, previous methods change the flow and segmentation separately, while we argue that they must be coupled. An object may appear in the wrong layer but with the correct motion. Consequently one must change the motion and layer segmentation simultaneously to avoid local optima.

Discrete optimization techniques, such as belief propagation [**88**] and graph cuts [**32, 85**], have been used in single-layer robust optical flow formulations [**92, 146**]. Particularly, Lempitsky *et al.* [**92**] fuse a large set of candidate flow fields to minimize a robust energy function. These methods can reach good local optima but tend to produce large errors in occlusion regions. This can partly be remedied by explicit occlusion detection and post processing [**197**]. In contrast, we exploit the layered model to explicitly model the occlusion process during continuous flow refinement. Graph cuts have also been used for segmentation and tracking. For example Kumar *et al.* [**89**] alternate the optimization of motion and segmentation and use affine motion models for each layer. Additionally, Wang *et al.* [**177**] require the manual segmentation of objects in the first frame and use parametric motion models.

The power of layered models is as much about segmentation as motion estimation, and we thus compare to a contemporary graph-based [**65**] video segmentation method. Chapter 2 reviews the video segmentation methods.

6.3. Models and Inference

We first define a discrete generative layered model based on an ordered sequence of binary, Ising MRFs. We then introduce a family of "cooperative" discrete optimization moves, as well as methods to determine the number of layers and their depth ordering.

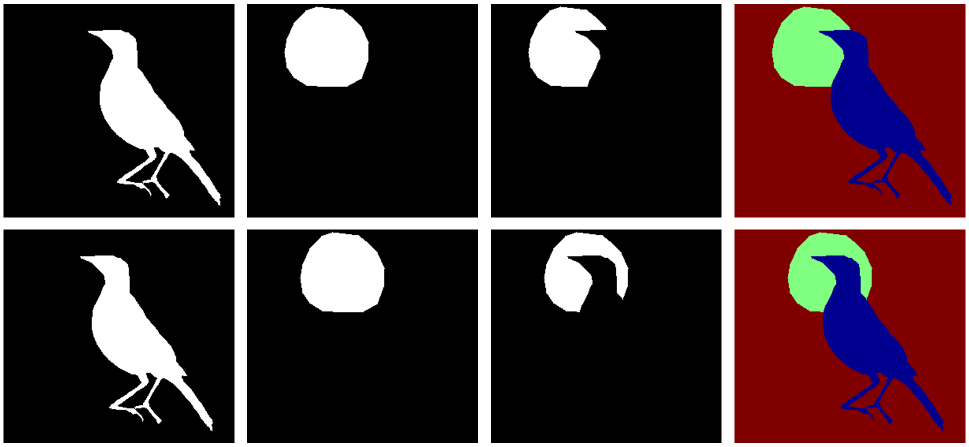


Figure 6.4. Left to right: binary masks for the first (front) and second layers, visibility mask for the second layer, and the layer segmentation. Top: frame t; bottom: frame $t + 1$. The "bird" layer is in front and occludes the second "apple" layer, resulting in the layer segmentation; Bottom row: The binary masks at time $t+1$ are temporally consistent with those at time t according to the flow field for each layer, resulting in temporally consistent layer segmentation. Note that the layer segmentation for a pixel is jointly determined by the two binary masks.

6.3.1. A Discrete Layered Model for Optical Flow. Consider an image sequence presumed to have K depth-ordered motion layers. In Chapter 5, we propose to model the spatial support of these layers by thresholding a sequence of smooth, continuous layer support functions. However, the local inference scheme is susceptible to local optima, especially in determining the scene structures.

In contrast to gradient-based methods, discrete optimization methods like graph cuts [**32, 85**] can substantially change the model configuration in a single move. This inspires us to develop a discrete formulation of the layered flow model using binary masks.

We capture the spatial coherence of the binary masks by a conditional Ising MRF with weights determined by image color differences:

$$E_{\text{space}}(\mathbf{m}_{tk}) = \frac{1}{2}\sum_{p}\sum_{q\in\Gamma_p} w_q^p \delta(m_{tk}^p \neq m_{tk}^q). \tag{50}$$

where $\mathbf{m}$ are binary functions and the weight w_q^p is the same as the continuous model in Chapter 5 and defined by Eq. 44.

The temporal consistency of the binary masks, as aligned by the inferred flow field, is encouraged by an Ising MRF:

$$E_{\text{time}}(\mathbf{m}_{tk}, \mathbf{m}_{t+1,k}, \mathbf{u}_{tk}, \mathbf{v}_{tk}) = \sum_{p}\sum_{q\in\mathcal{N}_{tk}^p} \delta(m_{tk}^p \neq m_{t+1,k}^{[q]}), \tag{51}$$

where the set $\mathcal{N}_{tk}^p = \{(i + u_{tk}^p, j + v_{tk}^p)\}$ contains the corresponding pixel at frame $t+1$ according to the flow field of the kth layer at frame t and $[q]$ means rounding the pixel position q to the nearest integer pixel. For non-integer flow vectors, sub-pixel interpolation introduces high-order temporal terms and may result in non-integer values around boundaries. We therefore round these flow vectors to obtain an approximation with only pairwise terms.

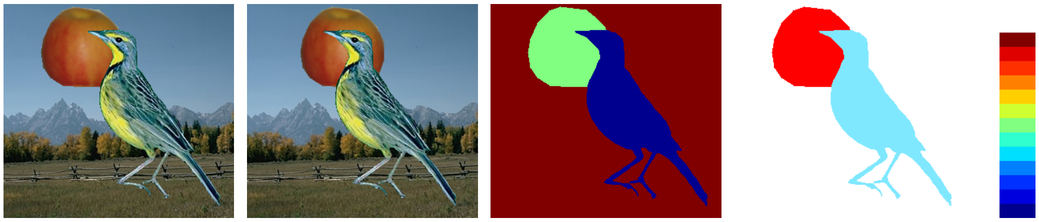
We also make a modification to the motion model for the discrete formulation, which produces more robust results than the original motion model in Eq. (40). We model the motion of each layer by a pairwise MRF with a unary term. The energy term is

$$E_{\text{aff}}(\mathbf{u}_{tk}, \theta_{tk}) = \frac{1}{2} \sum_{p} \sum_{q \in \Gamma_p} \rho_s \left(u_{tk}^{p} - u_{tk}^{q} \right) + \lambda_{\text{aff}} \sum_{p} \rho_{\text{aff}}(u_{tk}^{p} - \bar{u}_{\theta_{tk}}^{p}), \tag{52}$$

where the unary term encourages the flow field of each layer to be close to its affine flow (with weight λ_{aff}). Compared with the continuous model (Eq. 40), the motion of the discrete model is enforced to be similar to the affine motion field more. Note that this semiparametric model still allows deviation from the affine motion and is more flexible than parametric models. In automatically determining the number of layers, there is an important balance between the binary mask prior term (50) and the flow prior term (52): the former penalizes support discontinuities, while the latter favors additional layers so that each layer's flow is closer to affine.

6.3.2. Inference for the Discrete Model. The standard moves of graph cuts are not directly applicable to the discrete model, because of the high-order interaction terms in the data term (42). We therefore define a set of "cooperative" moves that can *a)* change a group of pixels to be visible at a particular layer while also selecting their flow fields; *b)* change a group of pixels to be visible at a particular layer; *c)* select the flow fields of a particular layer from a candidate set. Each move solves a binary problem via the Quadratic Pseudo-Boolean Optimization (QPBO) algorithm [**66, 85**], where the auxiliary binary variable, $\mathbf{b}$, encodes the states of several model variables.

We will use a toy example to explain the effect achieved by each move. Figure 6.5 shows the desired layer segmentation and flow field for the input "Bird-apple" sequence. During optimization we will see that there are several (fairly bad) local optima and we will need to make large changes to the solution to get out of these optima. Note that the binary selection variable $\mathbf{b}$ controls different variables for each move. The potential functions for each move are also defined differently though the functions may share the same name.

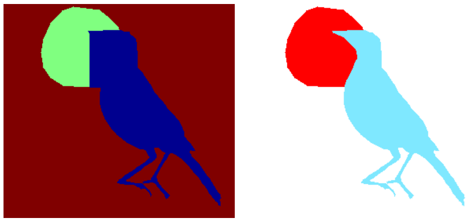


(a) first image (b) second image (c) segmentation (d) flow field (f) key

Figure 6.5. Input "Bird-apple" frames, the ground truth segmentation, the ground truth flow field, and the depth ordering key (blue is close and red is far).

6.3.2.1. *Simultaneous Segmentation and Flow Move.* Sometimes a region may be assigned to a wrong layer with the correct motion, as shown in Figure 6.6. To avoid this local optimum, we need to simultaneously change the segmentation and flow fields.

Consider a pixel p in frame t, for which layer k' is currently visible. We define a binary decision variable b_t^p such that the configuration is unchanged when $b_t^p = 0$, and an alternative layer $\hat{k}$ becomes visible when $b_t^p = 1$. Making this new layer $\hat{k}$ visible may alter all the binary masks for the first $\hat{k}$



(a) segmentation (b) flow field

Figure 6.6. Local minimum of the objective requiring the simultaneous segmentation and flow move. A large region of pixels in the center has been incorrectly assigned to the bird layer though their motion is correct. Changing either the segmentation or the flow field alone will not get out of the local minimum. We need to make these pixels visible in the apple layer and still retain their motion using the simultaneous segmentation and flow move (see text).

layers: when $b_t^p = 1$, $m_{t\hat{k}}^p(1) = 1$ and $m_{tk}^p(1) = 0, k < \hat{k}$. In this case, we also set the motion for layer $\hat{k}$ to that of the formerly visible layer ($u_{t\hat{k}}^p(1) = u_{tk'}^{p,\text{old}}$), and the motion for layer k' to its affine mean ($u_{tk'}^p(1) = \bar{u}_{\theta_{tk'}}^p$).

This segmentation and flow move involves several terms of the overall model, and the energy function can be represented using the binary selection variable as

$$E(\mathbf{b}) = \sum_{t=1}^{T-1}\sum_{k=1}^{K}\sum_{p}\sum_{q\in\mathcal{N}_{tk}^p(b_t^p)} \phi_{\text{time}}(b_t^p, b_{t+1}^q) + \sum_{t=1}^{T}\sum_{p}\left[\sum_{q\in\Gamma_p}\phi_{\text{space}}(b_t^p, b_t^q) + \phi_{\text{affine}}(b_t^p)\right]. \tag{53}$$

The choice of the binary variable b_t^p influences the flow vector at pixel p, and thus determines which of two candidate pixels p is linked to at the next frame. The temporal neighbors are

$$\mathcal{N}_{tk}^p(b_t^p) = \Big\{\big(i + u_{tk}^p(b_t^p), j + v_{tk}^p(b_t^p)\big)\Big\}, 1 \le k \le K \tag{54}$$

and the potential function is

$$\phi_{\text{time}}(b_t^p, b_{t+1}^q) = \Big(\rho_d(\mathbf{I}_t^p - \mathbf{I}_{t+1}^q) - \lambda_d\Big)\cdot s_{tk}^p(b_t^p)\cdot s_{t+1,k}^{[q]}(b_{t+1}^{[q]}) + \lambda_c\delta\Big(m_{tk}^p(b_t^p) \neq m_{t+1,k}^{[q]}(b_{t+1}^{[q]})\Big)\cdot\delta(k<K), \tag{55}$$

which incorporates both the data and the temporal terms for the $K-1$ binary masks. We evaluate the warped image I_{t+1}^q at subpixel positions and the visibility mask $s_{t+1,k}^{[q]}$ and the warped binary mask $m_{t+1,k}^{[q]}$ at integer positions.

For the spatial term the binary selection variable changes the states of several binary masks and flow fields. The effects sum together as

$$\phi_{\text{space}}(b_t^p, b_t^q) = \sum_{k=1}^{K-1}\lambda_b w_q^p\cdot\delta\Big(m_{tk}^p(b_t^p) \neq m_{tk}^q(b_t^q)\Big) + \sum_{k=1}^{K}\lambda_a\rho_{\text{mrf}}\Big(u_{tk}^p(b_t^p) - u_{tk}^q(b_t^q)\Big)\cdot\delta(t<T). \tag{56}$$

The unary term can be obtained from Eq. (52) as

$$\phi_{\text{affine}}(b_t^p) = \sum_{k=1}^{K}\lambda_a\lambda_{\text{aff}}\rho_{\text{aff}}\Big(u_{tk}^p(b_t^p) - \bar{u}_{\theta_{tk}}^p\Big)\cdot\delta(t<T). \tag{57}$$

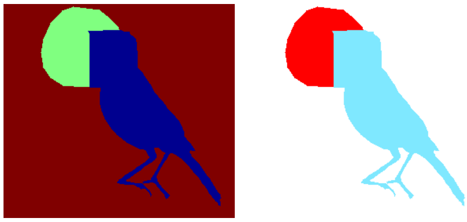

(a) segmentation (b) flow field

Figure 6.7. Local minimum that can be solved with the visibility move. A large number of pixels in the center of the image are assigned incorrectly to the bird layer, while the apple layer has the correct motion field for these pixels. Changing these pixels to be visible at the apple layer will fix this particular solution and get us out of the local minimum.

6.3.2.2. *Visibility Move.* Sometimes the segmentation is wrong, while an unassigned layer has the right motion. Hence we can make big changes to the layer segmentation to correct the errors in the motion field, as shown in Figure 6.7. Given the current flow estimate, we decide whether to make a pixel p visible for some layer $\hat{k}$ by modifying the previous layer support $\mathbf{g}^{\text{old}}$. Because the visibility state of a pixel is jointly determined by several binary masks, the binary selection variable controls the states of all the binary masks involved. When $b_t^p = 0$, all the binary masks retain their previous value at p, i.e., $m_{tk}^p(0) = m_{tk}^{p,\text{old}}$. When $b_t^p = 1$, we need to adjust the binary masks of the first $\hat{k}$ layers so that layer $\hat{k}$ is visible at p, i.e., $m_{tk}^p(1) = 0$ if $k < \hat{k}$, and $m_{tk}^p(1) = 1$ if $k = \hat{k}$. When $\hat{k}$ is the last layer, all the binary masks of the first $K-1$ layers are set to be 0 at p. The energy function for the binary variable is

$$\begin{aligned}
E(\mathbf{b}) = & \sum_{t=1}^{T-1} \Big\{ E_{\text{data}}(\mathbf{u}_t, \mathbf{v}_t, \mathbf{m}_t(\mathbf{b}_t), \mathbf{m}_{t+1}(\mathbf{b}_{t+1})) + \lambda_c E_{\text{time}}(\mathbf{m}_{tk}(\mathbf{b}_t), \mathbf{m}_{t+1,k}(\mathbf{b}_{t+1}), \mathbf{u}_{tk}, \mathbf{v}_{tk}) \Big\} \\
& + \sum_{t=1}^{T} \sum_{k=1}^{K-1} \lambda_b E_{\text{space}}(\mathbf{m}_{tk}(\mathbf{b}_t)) \\
= & \sum_{t=1}^{T-1} \sum_{k=1}^{K-1} \sum_{p} \sum_{q \in \mathcal{N}_{tk}^p} \phi_{\text{time}}(b_t^p, b_{t+1}^q) + \sum_{t=1}^{T} \sum_{p} \sum_{q \in \Gamma_p} \phi_{\text{space}}(b_t^p, b_t^q),
\end{aligned} \tag{58}$$

in which $\mathcal{N}_{tk}^p = \{(i + u_{tk}^p, j + v_{tk}^p)\}, 1 \leq k \leq K$ and the time term incorporates both the data term and the temporal consistency of the first $K-1$ binary masks $\phi_{\text{time}}(b_t^p, b_{t+1}^q) =$

$$\begin{cases} \Big(\rho_d(\mathbf{I}_t^p - \mathbf{I}_{t+1}^{[q]}) - \lambda_d\Big) s_{tk}^p(b_t^p) s_{t+1,k}^{[q]}(b_{t+1}^{[q]}) + \lambda_c \delta\Big(m_{tk}^p(b_t^p) \neq m_{t+1,k}^{[q]}(b_{t+1}^{[q]}))\Big), & k < K, \\ \Big(\rho_d(\mathbf{I}_t^p - \mathbf{I}_{t+1}^{[q]}) - \lambda_d\Big) s_{tk}^p(b_t^p) s_{t+1,k}^{[q]}(b_{t+1}^{[q]}), & k = K, \end{cases} \tag{59}$$

where the visibility mask s depends on the binary mask which in turn depends on the binary variable b, and the corresponding pixel q depends on the flow vector of the kth layer. The visibility move may change several binary masks. The potential function for the spatial term is

$$\phi_{\text{space}}(b_t^p, b_t^q) = \sum_{k=1}^{K-1} \lambda_b w_q^p \delta\Big(m_{tk}^p(b_t^p) \neq m_{tk}^{[q]}(b_t^{[q]})\Big). \tag{60}$$

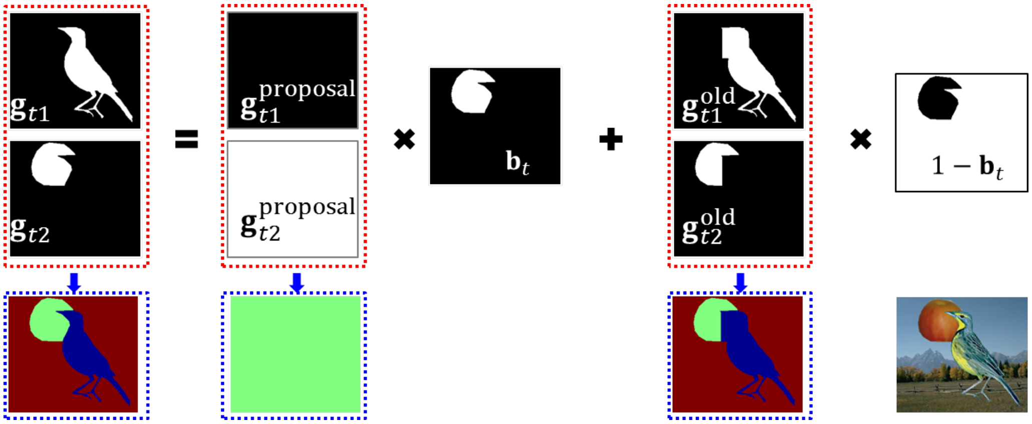

$\mathbf{g}_{t1}$
$\mathbf{g}_{t2}$
=
$\mathbf{g}_{t1}^{\text{proposal}}$
$\mathbf{g}_{t2}^{\text{proposal}}$
×
$\mathbf{b}_t$
+
$\mathbf{g}_{t1}^{\text{old}}$
$\mathbf{g}_{t2}^{\text{old}}$
×
$1-\mathbf{b}_t$

Figure 6.8. Illustration for the visibility move: we need to change the binary masks simultaneously to make desired change to the layer segmentations.

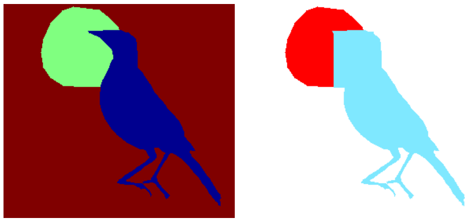
6.3.2.3. *Support Function Move.* Given the current flow estimate, we decide whether to make the binary mask of a pixel p at the selected layer $\hat{k}$ equal to 1 or retain its previous value. The energy function and the potential functions are the same as Eqs. (58)- (60) but the binary variable selects the solutions in a different way. When $b_t^p = 0$, $g_{t\hat{k}}^p(0) = g_{t\hat{k}}^{p,\text{old}}$. When $b_t^p = 1$, $g_{t\hat{k}}^p(1) = 1$.

6.3.2.4. *Occlusion-aware FusionFlow Move.* Here we deal with the case in which the segmentation is correct, but the motion field has a large region of errors, particularly in occlusion regions, as shown in Figure 6.9. We must make large changes to the flow field to reduce the motion errors. Given the current estimate of the binary masks for each layer, we want to select the motion of each pixel from a set of canidate flow fields. Here we use the current flow estimate and the affine mean flow field of the selected layer, $u_{tk}^p(0) = u_{tk}^{p,\text{old}}$ and $u_{tk}^p(1) = \bar{u}_{\theta_{tk}}^p$. Note that for occluded pixels, their data likelihood term does not provide useful information to estimate the motion. Instead we can predict the motion of the occluded pixels using the affine motion field fitted to the visible pixels of the same object (layer). For a particular layer k of frame t ($t < T$), the energy function is

$$\begin{aligned}
E(\mathbf{b}_{tk}) =& E_{\text{data}}(\mathbf{u}_t(\mathbf{b}_{tk}), \mathbf{v}_t(\mathbf{b}_{tk}), \mathbf{m}_t, \mathbf{m}_{t+1}) + \sum_{k=1}^{K} \lambda_a (E_{\text{aff}}(\mathbf{u}_{tk}(\mathbf{b}_{tk}), \theta_{tk}) + E_{\text{aff}}(\mathbf{v}_{tk}(\mathbf{b}_{tk}), \theta_{tk})) \\
&+ \lambda_c \sum_{k=1}^{K-1} E_{\text{time}}(\mathbf{m}_{tk}, \mathbf{m}_{t+1,k}, \mathbf{u}_{tk}(\mathbf{b}_{tk}), \mathbf{v}_{tk}(\mathbf{b}_{tk})) \\
=& \sum_{p} \phi_{\text{unary}}(b_{tk}^p) + \sum_{p} \sum_{q \in \Gamma_p} \phi_{\text{space}}(b_{tk}^p, b_{tk}^q), \qquad (61)
\end{aligned}$$

where the unary term incorporates the data term, the temporal consistency of the binary mask, and the deviation from the affine motion field

$$\begin{aligned}
\phi_{\text{unary}}(b_{tk}^p) = & \sum_{q \in \mathcal{N}_{tk}^p(b_{tk}^p)} \Big(\rho_d(\mathbf{I}_t^p - \mathbf{I}_{t+1}^q) - \lambda_d\Big) s_{tk}^p s_{t+1,k}^{[q]} + \lambda_c \delta(m_{tk}^p \neq m_{t+1,k}^{[q]}) \delta(k < K) \\
& + \lambda_a \lambda_{\text{aff}} \rho_{\text{aff}}(u_{tk}^p(b_t^p) - \bar{u}_{\theta_{tk}}^p), \qquad (62)
\end{aligned}$$



(a) segmentation (b) flow field

Figure 6.9. Local minimum that can be solved using the occlusion-aware fusion flow move. A large number of pixels in the center of the image have the wrong motion vectors. Making large changes to the flow field can reduce the errors. Note that we use the segmentation information to reason about occlusions.

in which the corresponding pixel at the next frame depends on the flow field selection at the current frame. This energy function differs from the FusionFlow method [**92**] in that the segmentation information directly modulates the data likelihood term in Eq. (62) and enables occlusion reasoning.

6.3.2.5. *Continuous Flow Refinement.* Given the current estimate of the binary masks for each layer, we refine the flow field for each layer by minimizing

$$\begin{aligned} E(\mathbf{u}_{tk}, \mathbf{v}_{tk}) =& E_{\text{data}}(\mathbf{u}_{tk}, \mathbf{v}_{tk}, \mathbf{m}_t, \mathbf{m}_{t+1}) + \lambda_a (E_{\text{aff}}(\mathbf{u}_{tk}, \theta_{tk}) + E_{\text{aff}}(\mathbf{v}_{tk}, \theta_{tk})) \\ &+ \lambda_c E_{\text{time}}(\mathbf{m}_{tk}, \mathbf{m}_{t+1,k}, \mathbf{u}_{tk}, \mathbf{v}_{tk}) \delta(k < K). \end{aligned} \tag{63}$$

Compared with standard optical flow formulation, this energy function contains the segmentation information necessary to reason about occlusions and an additional temporal consistency term for the binary masks. We can perform gradient-based optimization and the gradient of the energy function w. r. t. the flow fields can be derived similar to Chapter 2. This refinement step adaptively changes the flow field for the discrete optimization.

6.3.3. Layer Number Determination and Depth Order Reasoning. We initialize with an upper bound on the number of layers. During optimization, when a layer has no visible pixels associated with it, we remove it from the solution. The new solution can equally explain the image data, pays no penalty for the removed layer, and so has lower energy. Inferring the depth ordering of layers requires testing all the possible combinations and is computationally prohibitive. We instead use heuristics to reduce the search space. We first order the layers from fast to slow by their average motion. We then perform the moves above to estimate the binary masks and the flow fields in both the fast-to-slow and the slow-to-fast ordering. The ordering with the lower energy is further refined as follows. For each pair of neighboring layers, we propose to switch their ordering, and optimize their visibility mask and binary masks. If the new solution has a lower energy than its previous one, we accept this new depth ordering and proceed to other pairs. In practice, we find that this local greedy search scheme is fairly robust.

Table 6.1 provides the high-level algorithm. Tables 6.2 and 6.3 gives the detailed algorithms for inferring the depth ordering. For Table 6.2, where *maxIters* is set to be 10 in the current implementation and the algorithm usually stops after 4 to 6 iterations.

Table 6.1. The high-level algorithm for inferring the layered model.

```
Input: frames $\{I_t, 1 \le t \le T\}$, upper layer number, $K_{\max}$
• Estimate flow $(\mathbf{u}_{t*}, \mathbf{v}_{t*})$ between $I_t$ and $I_{t+1}$ using Classic+NL
• Perform affine K-means on $(\mathbf{u}_{t*}, \mathbf{v}_{t*})$ to obtain $K_{\max}$ groups
• Perform the following moves (Sequence A) for fast-to-slow and slow-to-fast depth orderings, including
    - Perform visibility move for each layer, remove redundant layers
    - Perform joint segmentation and flow move, remove redundant layers
    - Perform binary mask move, remove redundant layers
    - Re-estimate affine flow field for each layer
    - Perform flow Fusion move
    - Perform continuous flow refinement
• Pick up the solution with lower energy between $solution^{\text{fast-to-slow}}$ and $solution^{\text{slow-to-fast}}$
• Perform Sequence B (Table 6.2) to decide depth ordering between neighboring layers
• Perform Sequence A to refine segmentation and flow
Output: $solution = \big\{K, \{(\mathbf{u}_{tk}, \mathbf{v}_{tk}), 1 \le t \le T-1, 1 \le k \le K\}, \{\mathbf{m}_{tk}, 1 \le t \le T, 1 \le k \le K-1\}\big\}$
```

6.4. Experimental Results

We evaluate the proposed layered model on both motion estimation and layer segmentation tasks. Throughout this section, the proposed method is called **nLayers**, since it can automatically determine the number of layers. **Layers++** refers to the continuous method developed in Chapter 5 which uses a fixed number of 3 layers. For layer segmentation, we also compare our method to a recent, hierarchical graph-based video segmentation algorithm [**65**], referred to as **HGVS** in the comparison below. Note that **HGVS** uses the output from a recent optical flow estimation method [**189**].

6.4.1. Implementation Details and Parameter Settings. We start with the single-layered output from **Classic+NL** [**158**] and cluster the flow field into 10 layers. We then run the discrete method to obtain an initial estimate of the scene structure and the flow fields, and use them to initialize the more precise continuous layered model. It takes **nLayers** about 10 hours to compute three forward and three backward flow fields from the four-frame 640 × 480 "Urban" sequence in MATLAB with the QPBO solver, complied as MEX. It takes **Layers++** about 5 hours to compute one forward and one backward flow fields from two frames. **HGVS** uses ten frames, or all the frames if a sequence has fewer than ten frames. **HGVS** has three different outputs for the same video. We show the segmentation results produced at 90 percent of highest hierarchy level, because it gave the best visual and numeric results.

Table 6.2. The algorithm for infering the depth ordering between neighboring layers.

```
Input: $solution^{\textbf{input}}$, $maxIters$

For $iter = 1 : maxIters$
  • Select candidate pairs according to the occlusion area
  • $solution^{\textbf{curr}} = solution^{\textbf{input}}$
  • $is\_local\_minimum = true$
  For each selected pair of selected neighboring layers
    - Swap depth ordering
    - Perform visibility move for the two selected layers
    - Perform binary mask move for the two layers
    - If $energy(solution^{\textbf{new}}) < energy(solution^{\textbf{curr}})$
      ∘ $solution^{\textbf{curr}} = solution^{\textbf{new}}$
      ∘ $is\_local\_minimum = false$
    End of If
  End of For
  • Remove redundant layers
  • If $is\_local\_minimum == true$
    - break
  End of If
End of For

Output: $solution^{\textbf{curr}}$
```

Table 6.3. The algorithm for selecting the candidate neighboring layer pairs.

```
Input: $maxPairNumbers$ (max number of layer pairs to compare)

•Compute the number of ($pairNumber$) neighboring pairs using current segmentation
If $pairNumber \leq maxPairNumbers$
  - Output all the neighboring pairs and exit
Else
  - Compute occlusion regions between each neighboring pairs
  - Order the pairs by the occlusion area in descending order
  - Output the top $maxPairNumbers$ neighboring layer pairs
End of If

Output: pairs of neighboring layers to compare
```

6.4.2. Motion Estimation. We use the Middlebury optical flow benchmark to evaluate the motion estimation results.We manually set $\lambda_{\text{aff}} = 0.3, \lambda_{\text{b}} = 80,$ and $\lambda_c = 10$ for the discrete model, use the provided values for the other parameters from Chapter 5, and fix them for all the motion

Table 6.4. Average end-point error (EPE) on the Middlebury *training* set. Using four frames and the new optimization improves accuracy.

	Avg.	Venus	Dimetr- odon	Hydr- angea	Rubber- Whale	Grove2	Grove3	Urban2	Urban3
Classic+NL (2 frames)	0.221	0.238	0.131	0.152	0.073	0.103	0.468	0.220	0.384
Layers++ (2 frames)	0.195	0.211	0.150	0.161	0.067	0.086	0.331	0.210	0.345
Layers++ (4 frames)	0.190	0.211	0.151	0.157	0.067	0.084	0.330	0.207	0.311
nLayers (4 frames)	0.183	0.191	0.126	0.175	0.062	0.080	0.336	0.175	0.316

Table 6.5. Average end-point error (EPE) and angular error (AAE) on the Middlebury optical flow benchmark *test* set. The discrete-continuous optimization (**nLayers**) obtains similar EPE and better AAE than the continuous-only inference method (**Layers++**). The ranking information is at the writing of the dissertation (July 2012).

		Rank	**Avg.**	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
EPE	**Layers++**	9.2	0.27	0.08	0.19	0.20	0.13	0.48	0.47	0.15	0.46
	nLayers	9.8	0.28	0.07	0.22	0.25	0.15	0.53	0.44	0.13	0.47
AAE	**Layers++**	10.9	2.56	3.11	2.43	2.43	2.13	2.35	3.81	2.74	1.45
	nLayers	6.5	2.38	2.80	2.71	2.61	2.30	2.30	2.62	2.29	1.38

Table 6.6. RandIndex measures on the MIT human labeled dataset for the three methods (and variants).

	Avg.	***p*-value**	Car	Car2	Car3	Dog	Phone	Table	Toy	Hand	Person
HGVS [**65**]	0.550	0.008	0.602	0.401	0.689	0.260	0.493	0.766	0.809	0.499	0.430
Layers++ [**160**]	0.775	0.050	0.612	0.512	0.778	0.964	0.567	0.909	0.832	0.814	0.986
Layers++ (8 layers)	0.690	0.021	0.711	0.510	0.802	0.531	0.564	0.842	0.874	0.590	0.790
Layers++ (10 layers)	0.675	0.027	0.661	0.517	0.799	0.613	0.551	0.853	0.846	0.640	0.597
nLayers	0.823	—	0.836	0.589	0.766	0.974	0.578	0.979	0.858	0.881	0.944
nLayers (8 layers)	0.808	0.275	0.611	0.590	0.821	0.975	0.575	0.981	0.852	0.920	0.951
nLayers (12 layers)	0.800	0.204	0.603	0.574	0.810	0.974	0.575	0.944	0.876	0.889	0.952
nLayers (2 frames)	0.793	0.124	0.608	0.535	0.755	0.970	0.578	0.979	0.841	0.923	0.951

estimation experiments. We set all the robust functions to be the generalized Charbonnier penalty function $\rho(x) = (x^2 + \epsilon^2)^a$ with $\epsilon = 0.001$ and $a = 0.45$ as in Chapter 4.

Results on the Middlebury *training* set are shown in Table 6.4. Changing from 2 to 4 frames improves results for the **Layers++** model supporting our hypothesis that longer sequences are important. More improvement comes from using a discrete model to obtain a good segmentation of the scene and then use the inferred structure for flow estimation (**nLayers**, 4 frames).

On the *test* set, **nLayers** obtains EPE similar to **Layers++** but better AAE, as shown in Table 6.5. At the time of writing (July 2012), **nLayers** is ranked 1^{textst} in AAE and 5^{textth} in EPE. (see Appendix A for the screen shot). The results suggest that **nLayers** estimates motion directions more accurately.

Figures 6.10-6.13 show the estimated segmentation and flow fields on both the training and the test sequences. **nLayers** performs well on estimating the major scene structure. Nearly all the major structures of "Urban" in Figure 6.13 are correctly recovered, resulting in the best boundary

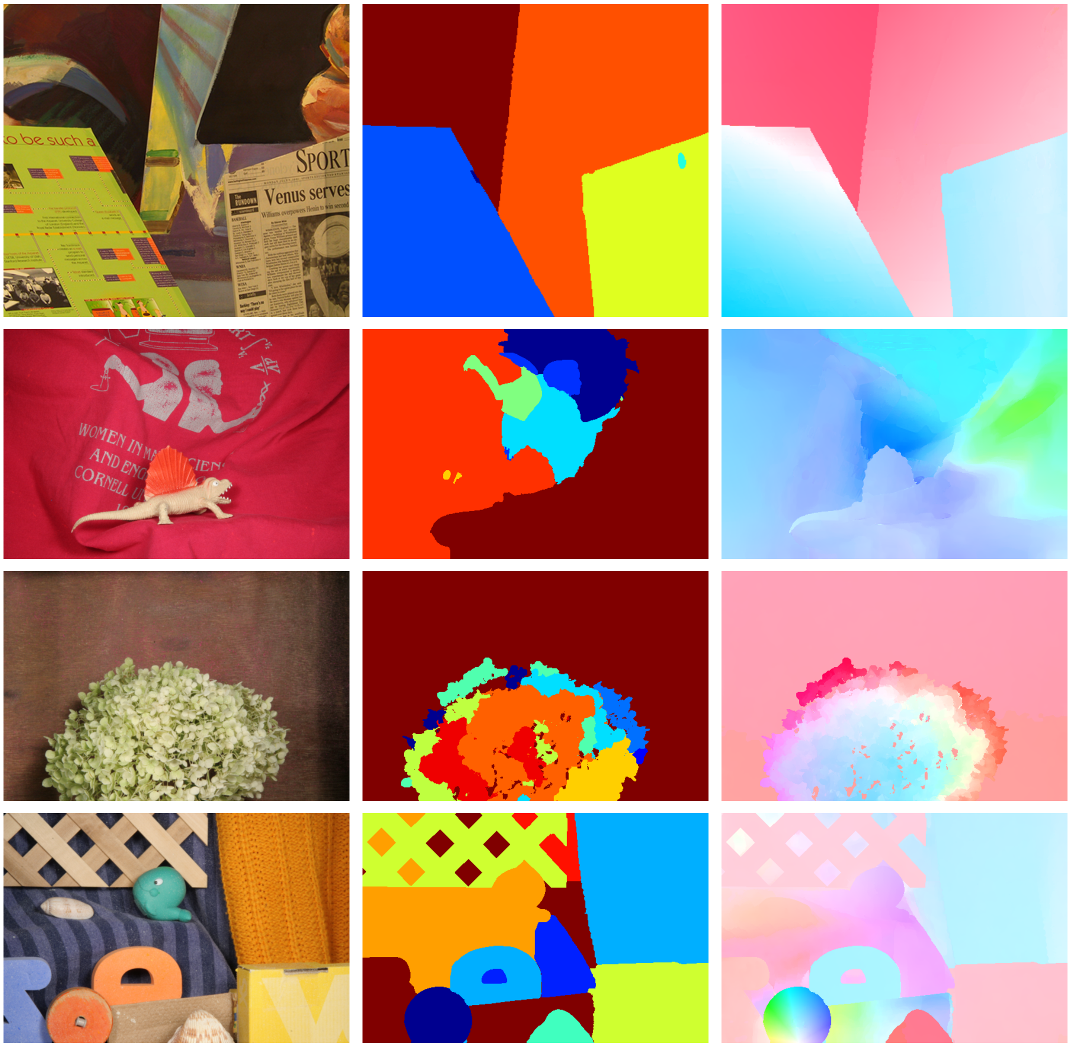
EPE and AAE performance (Figures A.1 and A.2). The higher overall error results from the bottom left building. A major part of the building moves out of the image boundary and has no data term to estimate the motion. **nLayers** uses the affine model to interpolate the motion of the out-of-boundary pixels, but the building's motion violates the affine assumption. Future work will study sequence-adaptive model selection to reduce such errors.

6.4.3. Layer Segmentation. The Middlebury dataset does not have motion segmentation ground truth and so we use the MIT human annotated dataset [**99**] to evaluate segmentation performance. Segmentation accuracy is computed using the RandIndex measure [**127**] (larger is better). Because the MIT dataset is different in nature from the Middlebury dataset and has more rigidly moving, distant objects, we use a larger weight on the affine unary term as $\lambda_{\text{aff}} = 1,$ and $\lambda_c = 3$ for the discrete model while keeping the other parameters unchanged.

Table 6.6 summarizes the RandIndex measure on all the 9 sequences. On several sequences, **nLayers** (default: 10 layers and 4 frames) outperforms **Layers++** by a large margin. A bootstrap significance test is used between the **nLayers** and other methods. Small p values suggest that the improvement over **HGVS** and **Layers++** by **nLayers** is significant. **nLayers** with the maximum number of layers being 8 or 12 produces results similar to the baseline 10-layer model suggesting the method is not highly sensitive to the maximum number of layers. **nLayers** with only 2 frames is more accurate than the 2-frame **Layers++** method, demonstrating the benefits of discrete optimization. The improvement with 4 frames over 2 frames shows the benefits of using more frames to recover scene structure. Also note that the performance of **Layers++** drops with the number of layers used because its local inference scheme has to deal with more local optima as the number of layers increases.

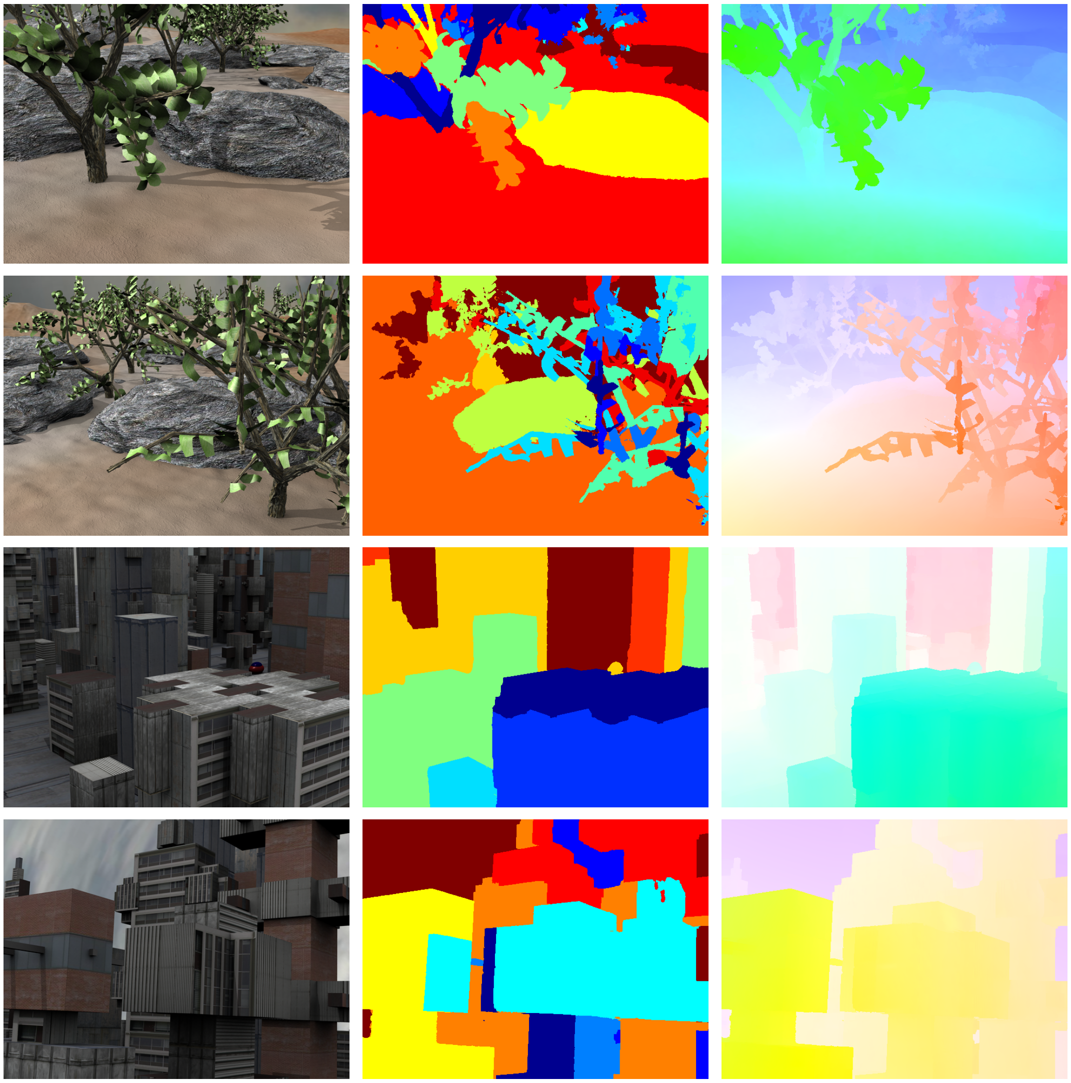
Figures 6.14-6.16 shows the segmentation results. On "Table", the layer segmentation by **nLayers** roughly matches the structure of the scene and is close to the human labeled ground truth. Although **HGVS** uses optical flow, it still merges some foreground objects and background especially when their appearance is similar. **nLayers** tends to fail when motion cues are weak, such as the "Phone" sequence. Adding an appearance model [**89, 179**] is likely to better exploit the static image cues and thereby resolve ambiguity.

We also apply **nLayers** with the same parameter setting to some "old" popular sequences used in the literature, including "Cow" [**89**], "Hedvig" [**150**], and "Jojic & Frey" [**81**]. The segmentation results capture the major scene structures. However, **nLayers** assigns the lower parts of legs for the cow and the person to the background layer because their motion is small in the four frames we process. Using more frames may resolve the ambiguity by propagating information from frames where the legs have large motion.



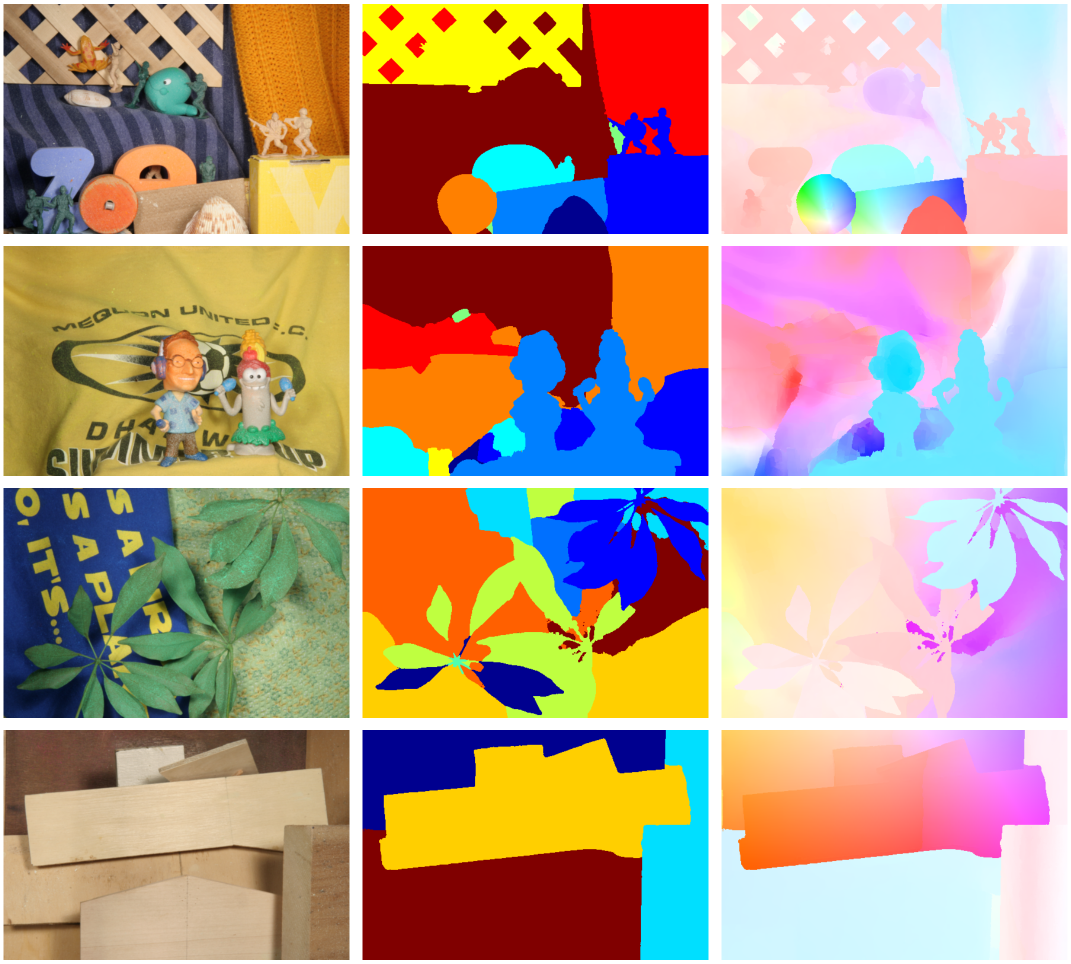
(a)First frame (b) Layer segmentation (c) Estimated flow field

Figure 6.10. Estimated flow fields and scene structure on the Middlebury *training* sequences. Left to right: first frame, layer segmentation, and estimated flow field. Top to bottom: “Venus”, “Dimetrodon”, “Hydrangea”, and “RubberWhale”. Color key for the depth ordering is the same as Figure 6.5 (blue is close and red is far).



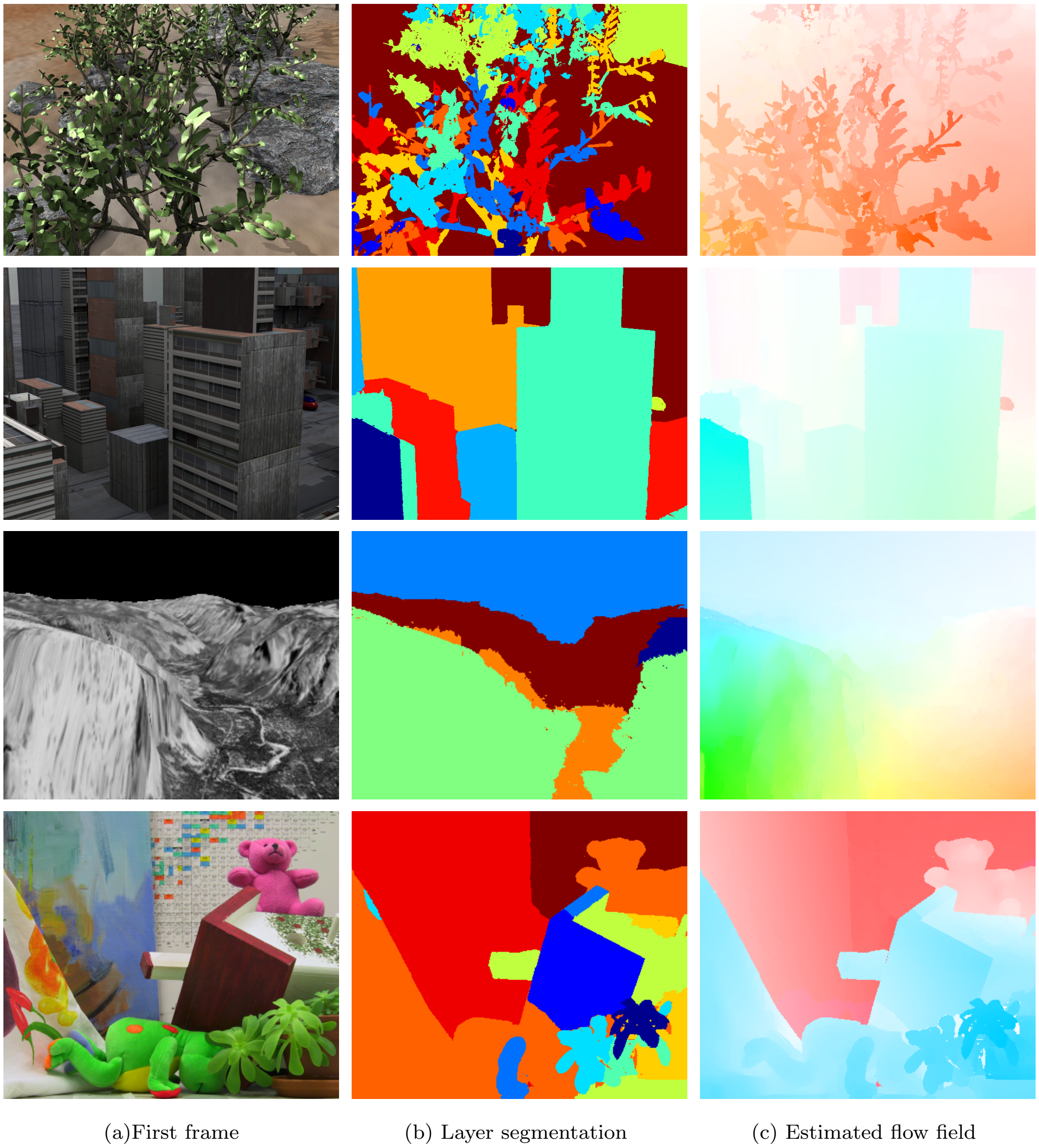
(a)First frame (b) Layer segmentation (c) Estimated flow field

Figure 6.11. Estimated flow fields and scene structure on the Middlebury *training* sequences. Left to right: first frame, layer segmentation, and estimated flow field. Top to bottom: “Grove2”, “Grove3”, “Urban2”, and “Urban3”. Color key for the depth ordering is the same as Figure 6.5 (blue is close and red is far).



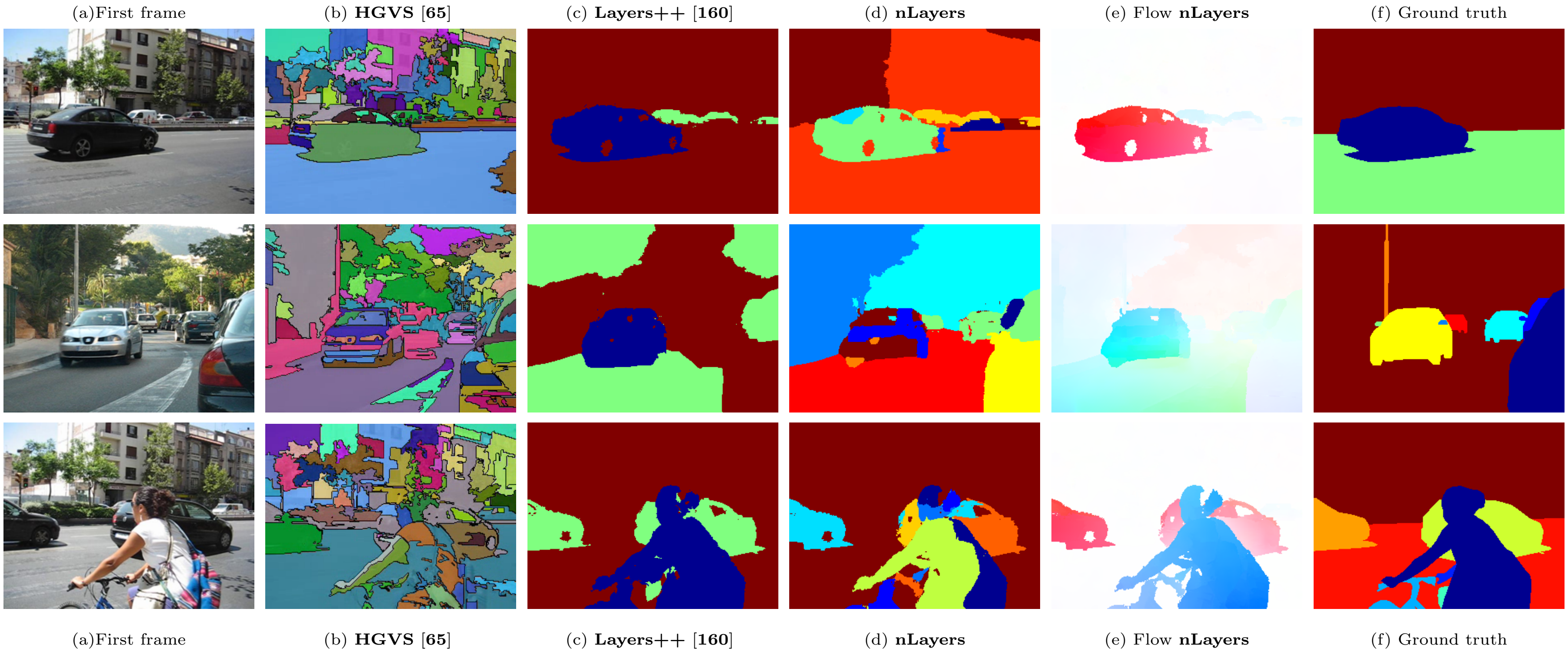
(a)First frame (b) Layer segmentation (c) Estimated flow field

Figure 6.12. Estimated flow fields and scene structure on the Middlebury *test* sequences. Left to right: first frame, layer segmentation, and estimated flow field. Top to bottom: “Army”, “Mequon”, “Schefflera”, and “Wooden”. Color key for the depth ordering is the same as Figure 6.5 (blue is close and red is far).



(a)First frame (b) Layer segmentation (c) Estimated flow field

Figure 6.13. Estimated flow fields and scene structure on the Middlebury *test* sequences. Left to right: first frame, layer segmentation, and estimated flow field. Top to bottom: "Grove", "Urban", "Yosemite", and "Teddy". Color key for the depth ordering is the same as Figure 6.5 (blue is close and red is far).



(a)First frame

(b) **HGVS** [**65**]

(c) **Layers++** [**160**]

(d) **nLayers**

(e) Flow **nLayers**

(f) Ground truth

(a)First frame

(b) **HGVS** [**65**]

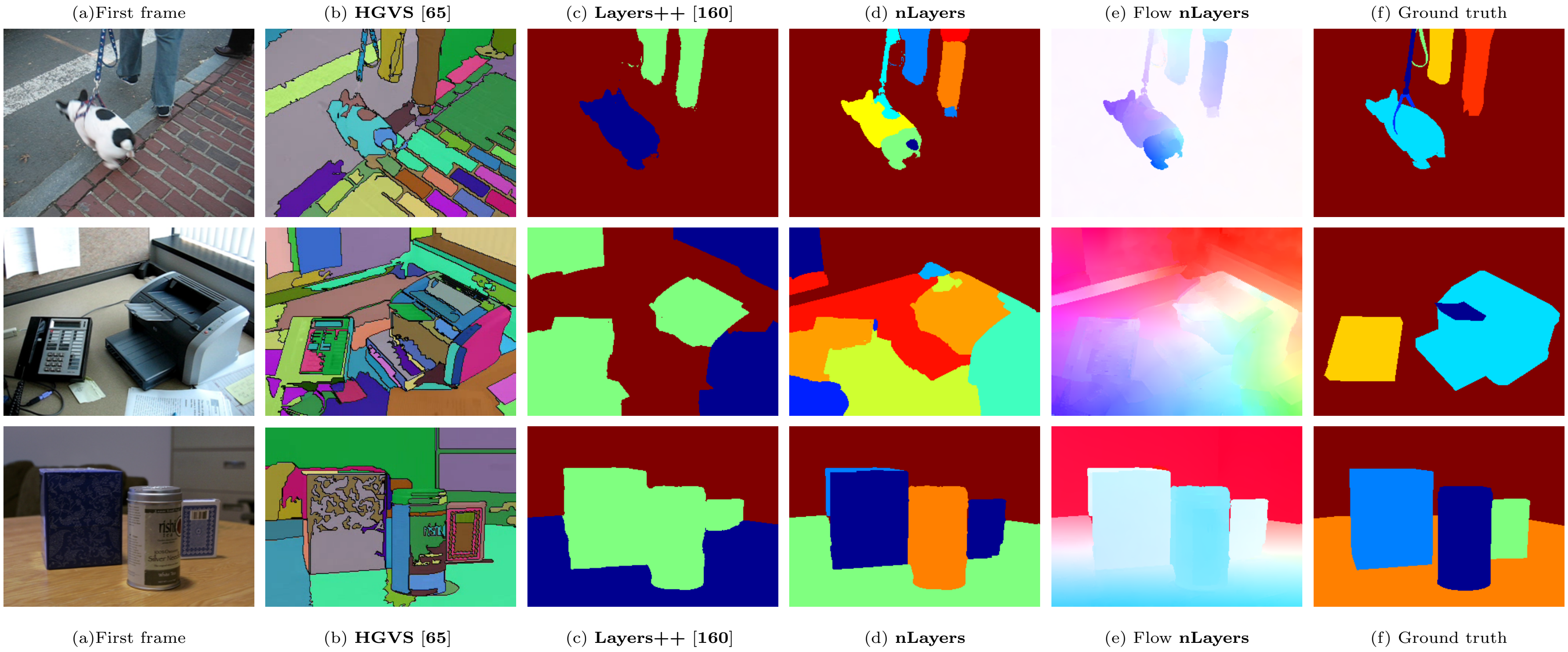
(c) **Layers++** [**160**]

(d) **nLayers**

(e) Flow **nLayers**

(f) Ground truth

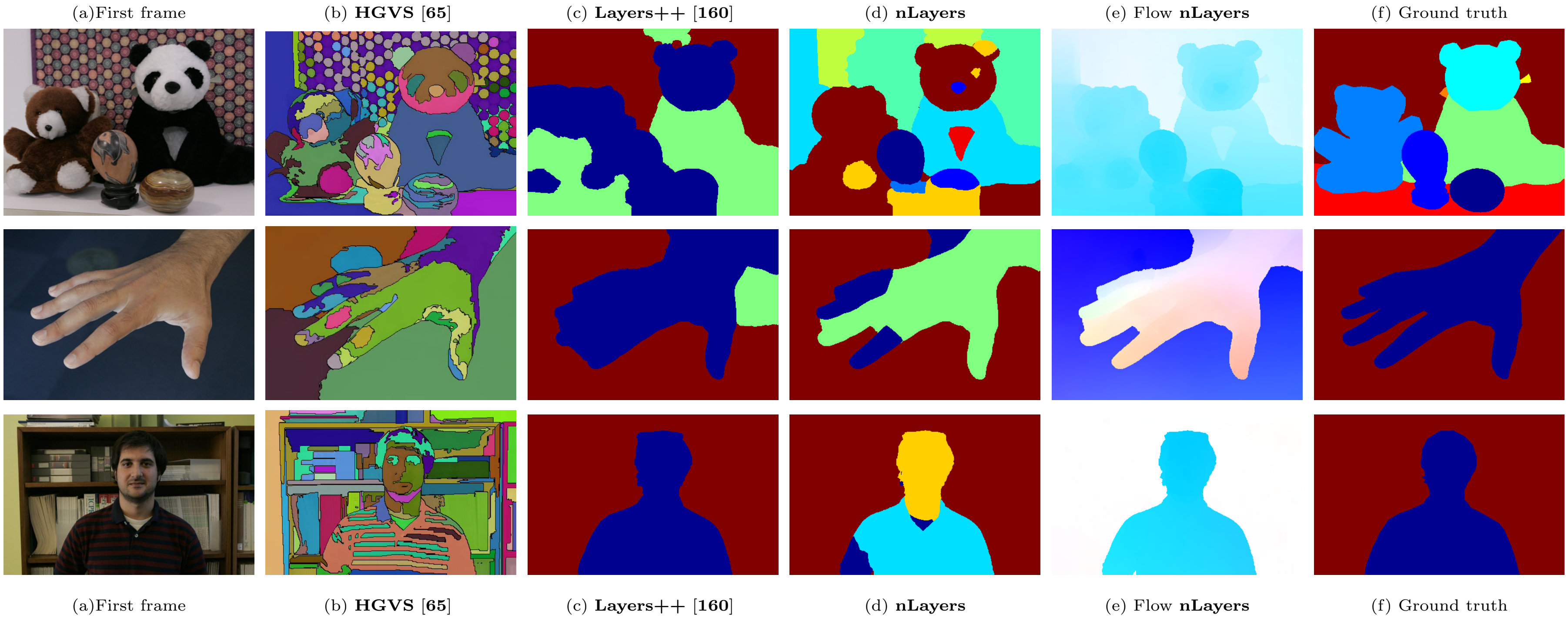
Figure 6.14. Results on the MIT dataset. Top to bottom: first frame, segmentation results by **HGVS** [**65**], **Layers++** [**160**], **nLayers**, estimated flow field by **nLayers**, and human labeled ground truth. Left to right: "Car", "Car2", and "Car2". Color key for the depth ordering is the same as Figure 6.5 (blue is close and red is far).



(a)First frame (b) **HGVS** [**65**] (c) **Layers++** [**160**] (d) **nLayers** (e) Flow **nLayers** (f) Ground truth

(a)First frame (b) **HGVS** [**65**] (c) **Layers++** [**160**] (d) **nLayers** (e) Flow **nLayers** (f) Ground truth

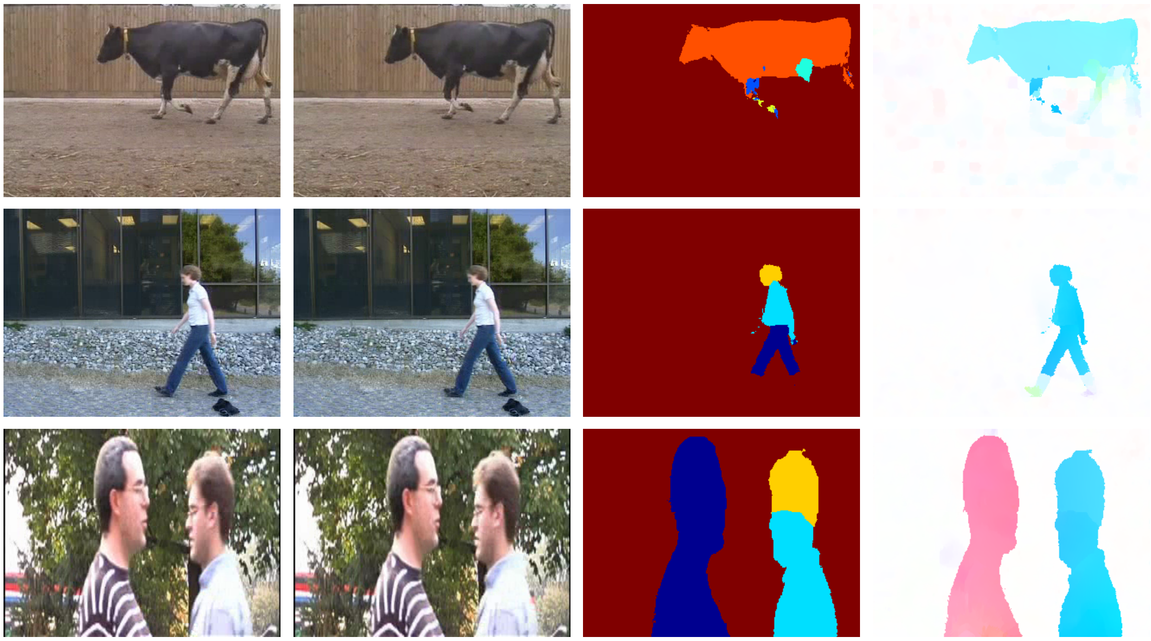
Figure 6.15. Results on the MIT dataset. Top to bottom: first frame, segmentation results by **HGVS** [**65**], **Layers++** [**160**], **nLayers**, estimated flow field by **nLayers**, and human labeled ground truth. Left to right: “Dog”, “Phone”, and “Table”. Color key for the depth ordering is the same as Figure 6.5 (blue is close and red is far).



(a)First frame (b) **HGVS** [**65**] (c) **Layers++** [**160**] (d) **nLayers** (e) Flow **nLayers** (f) Ground truth

(a)First frame (b) **HGVS** [**65**] (c) **Layers++** [**160**] (d) **nLayers** (e) Flow **nLayers** (f) Ground truth

Figure 6.16. Results on the MIT dataset. Top to down: first frame, segmentation results by **HGVS** [**65**], **Layers++** [**160**], **nLayers**, estimated flow field by **nLayers**, and human labeled ground truth. Left to right: “Toy”, “Hand”, and “Person”. Color key for the depth ordering is the same as Figure 6.5 (blue is close and red is far).



(a)Second frame (b)Third frame (c) Layer segmentation (d) Estimated flow field

Figure 6.17. Estimated flow field from the second frame to the third frame and scene structure at the second frame on some "old", popular sequences. Top to bottom: "Cow", "Hedvig", and "Jojic & Frey". Color key for the depth ordering is the same as Figure 6.5 (blue is close and red is far).

6.5. Conclusions and Discussions

We have formulated a discrete layered model based on a sequence of ordered Ising MRFs and developed non-standard moves to optimize the model. In particular, our moves can simultaneously change the layer assignment together with the flow field, which helps avoid local optima common to alternating optimization schemes. The discrete optimizer enables us to adapt the number of layers to each sequence and decide their depth ordering automatically. Our method produces meaningful segmentations on the Middlebury and the MIT datasets, and achieves better quantitative results w. r. t. the human labeled ground truth than the local inference method developed in Chapter 5. Our flow estimation results show the benefits of using more frames and discrete optimization to resolve depth-ordering ambiguities. Our work advances the state of the art in layered motion modeling and suggests that layered models can provide a rich and flexible representation of complex scenes.

CHAPTER 7

Conclusions and Future Work

Preceding chapters have introduced several methods to go beyond the classical optical flow formulation to recover motion boundaries, reason about occlusions, and segment scenes into coherently moving objects. This chapter summarizes our main contributions, limitations, and directions for future research.

7.1. Contributions and Recommendations

Image motion estimation is an inherently ill-posed problem. Similar to other low-level vision tasks, such as denoising, we need good prior models to regularize the unknown flow fields. However, motion estimation differs from denoising in that the data observation model is more complicated. To design a good motion estimation method, we should choose both the prior model and the data observation model carefully.

The pairwise MRF models have been a popular choice for low-level vision problems. We reformulate the classical formulation of optical flow by Horn and Schunck and its descendants probabilistically as pairwise MRF models in Chapter 3. This formulation enables us to learn the parameters from training data, an important step missing in the original work by Horn and Schunck. The heavy-tailed histograms of both the brightness constancy error and the derivatives of the flow fields confirm our intuitions that brightness constancy does not always hold, and motion discontinuities exist. The learned parameters achieve similar performance as the hand-tuned parameters but save the task of tuning the unknown parameters. However, even with learned parameters, the classical approach still has difficulties dealing with motion boundaries, occlusions, and lighting changes etc.

We extend the standard models in both the data and the spatial terms. To deal with lighting changes, we learn high-order constancy models from training data and further generalize the idea to learned filter constancy. To better preserve motion boundaries, we formalize the concept of oriented smoothness probabilistically as a Steerable Random Field (SRF). These advanced models consistently outperform the standard models, in particular, in the motion boundary and shadow regions. Nevertheless, the motion boundary and occlusion regions still cause serious problems to these advanced models.

To understand the recent developments in optical flow estimation, we perform a thorough analysis of how the objective function, the optimization method, and the implementation details influence the performance in Chapter 4. We find that the classical formulation by Horn and Schunck achieves competitive results with modern implementations. The new implementation also produces significantly better results than the initial implementation presented in Chapter 3. The key is to apply median filtering to the flow field during the incremental estimation process. However, the median filtering step increases the energy of the final results, suggesting that a different objective is being minimized. Exploiting connections between median filtering and L1 energy minimization, we show that algorithms relying on a median filtering step are approximately optimizing a different objective that regularizes flow over a large spatial neighborhood. The good performance by the classical formulation, especially the Horn and Schunck method, results from the use of a large robust spatial prior via the median filtering. This observation enables us to design and optimize improved models that weigh the neighbors adaptively in an extended image region. Similar to the SRF models, the weighted non-local term uses the static image information to detect and preserve motion details.

The non-local term still cannot handle occlusions well, because the model does not incorporate occlusions. We revisit the layered approach and develop a probabilistic model based on thresholded support functions in Chapter 5. Our model fixes several limitations of previous approaches and explicitly models the occlusions between layers, the depth ordering of layers, and the temporal consistency of layers. The support functions flexibly evolve over time according to globally coherent and locally flexible motion fields.

Inferring the layered model is challenging because of poor local optima. As a baseline, we develop a local, gradient-based scheme and obtain promising results on motion estimation, as tested on the Middlebury dataset. However, the local inference scheme fails on some worst-case toy examples that violate the assumptions on the number of layers and their depth ordering.

Inspired by the ability of discrete optimization to make big, non-local changes to the solution, we have proposed a discrete-continuous optimization scheme for our layered model in Chapter 6. We formulate a discrete layered model using a sequence of depth-ordered Ising MRFs and develop a series of nonstandard moves to optimize the proposed model. In particular, we develop simultaneous motion and segmentation moves that can solve local minima where typical EM-type algorithms tend to fail. Experimental results show that the proposed discrete-continuous approach is more robust to the initialization and produces better scene segmentation results than the local approach. The layered approach achieves competitive results on both layer segmentation and motion estimation, demonstrating the benefits of jointly solving for the motion and the segmentation.

7.2. Limitations and Future Work

7.2.1. Long-term Video Analysis with the Layered Model. Although we have obtained some benefits by using more than two frames, our layered method still cannot analyze long video sequences, because of several challenges in the modeling, the inference, and the implementation.

Our layered model does not incorporate long-term temporal dynamics. An object may appear in the scene, become fully occluded by another, or exit the scene. Two objects may change their depth ordering at different time instances. All these phenomena violate our current model assumptions.

Another missing part is the appearance of each layer, *e.g.,* the color of every pixel for every layer. We rely on the data constancy term between two neighboring frames as the observation model. Appearance, on the other hand, is more likely to persist over many frames. Appearance will help the segmentation where the motion cues are too weak. The learned appearances will be useful for other video analysis tasks. The segmentation and the learned appearances from many video sequences can be useful inputs for object recognition tasks. The resultant system uses motion as a weak supervised signal. The key challenge in adding the appearance is the increased solution space and more local optima. Similarly to simultaneous segmentation and flow moves, we may need to develop moves to change the segmentation, motion, and appearance simultaneously.

In addition, the depth ordering is ill-defined for mutually occluding layers, such as two shaking hands. Breaking these objects into small layers is one possible solution, but may invoke enormous computational costs or lose the global reasoning ability of the motion model. Local layering [**110**] is one possible solution, but the inference is more difficult. 3D representations, such as depth map, is more suitable for describing the mutual occlusion or the self occlusion relationship and is worth pursuing in the long run.

Instead of using a single flexible motion model, such as the semiparametric ones, we may perform on-line model selection for the given input sequences. We can create a set of motion models, ranging from translational motion all the way to the most flexible robust MRFs. In the middle of the spectrum, we may add depth-based motion representation for rigidly moving objects, and descriptor matching for fast-moving objects [**37**]. We may extract some high-level video descriptors and use them to choose the right motion models.

One practical challenge for processing long video sequences is the high computational cost for the proposed layered model. One possibility is to use larger representation units, such as super pixel/voxel to obtain a rough solution. We can then refine the solution using pixel units if required by the particular applications. It is also worth investigating using signal process techniques [**125**] to efficiently implement the algorithms.

Learning is a more principled way than hand tuning for setting the model parameters. We have demonstrated the benefits of learned models over the standard models in Chapter 3. However, limited training data may cause over-fitting problems, especially for the flexible layered models. We may take a fully Bayesian approach by putting priors on the parameters. We can either marginalize the parameters or re-sample them depending on conjugacy. This is harder to implement but likely to bring higher rewards, in particular with no over-fitting issues.

Despite these challenges, long-term motion estimation is appealing because the output is useful for a variety of applications, such as video super resolution [**103**] and video manipulation [**64**]. Our results suggest that the layered approach is favorable for long-term motion analysis because the scene structure is more likely to persist over time than the optical flow. Even with four frames, we have already observed some benefits by imposing temporal consistency on the scene structures. Extending the proposed layered model to long-term motion analysis can better integrate information over time.

7.2.2. Implementation. We released our MATLAB implementation for **Classic+NL** together with the publication of our paper [**158**]. The public software has been used by both researchers to develop new flow methods [**5, 80**] and practitioners to apply optical flow in their work [**118**]. However, three minutes per 640 × 480 image pair is too slow for real-time video processing and may exceed the total computational time for some high-level video analysis systems [**139**]. Making motion estimation method faster and able to process high definition (HD) images is important for motion to become a basic tool for both low and high level video processing and analysis. Implementing the non-local term in C++ and on a GPU may provide a significant speedup. It is more challenging but also profitable to make the inference faster for the layered model.

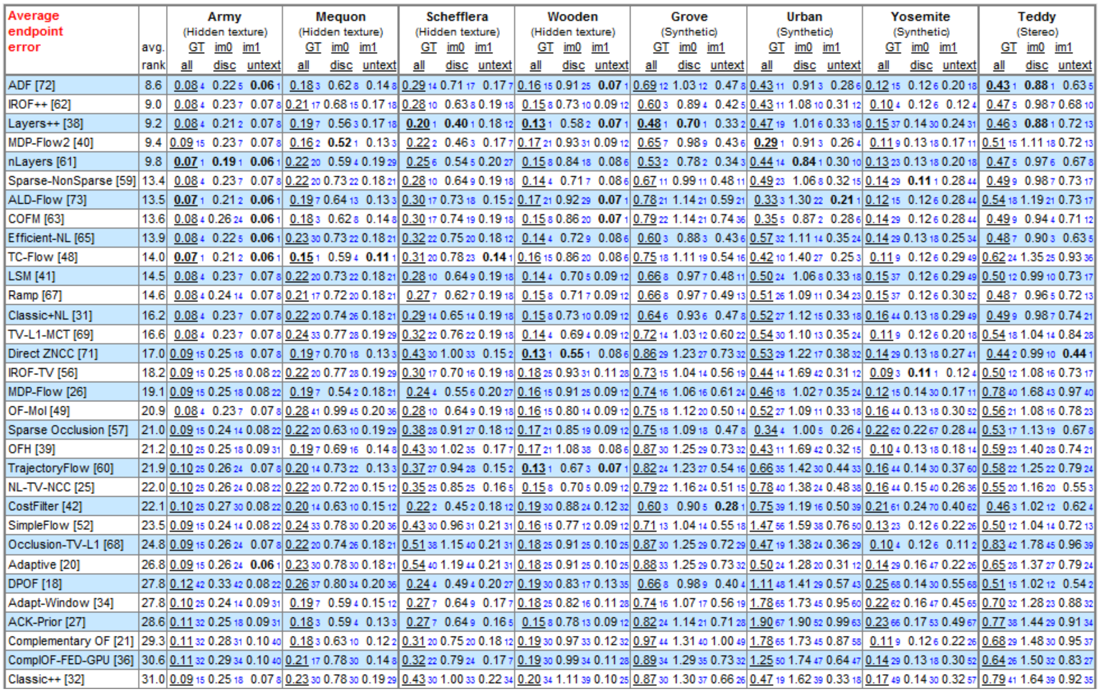
7.2.3. Dataset and Over-fitting Issues. We have mainly used the Middlebury optical flow and the MIT layer segmentation dataset. These two datasets are more challenging than the previously widely used "Yosemite" sequence and have more sequences of different natures. However, both datasets are small in size and over-fitting is likely to exist, which may have biased our conclusions. Although the proposed methods have not addressed all the challenges imposed in these two datasets, it will be more profitable to test and design algorithms on more realistic, representative datasets [**41, 59**].

There are several issues in real-world video sequences that we have not addressed in this dissertation. Real-world sequences may contain motion blur and structured noise and hence have different statistics from those used in this thesis. In addition, real-world video sequences are all stored in compressed format and suffer from compression artifacts to some degree. The compression artifacts are false high frequency signals and may be mistaken as the true signals. We may need to solve for both the motion and the original video together [**162**]. The sizes of the real-world HD sequences are much larger than the sequences used in the thesis and further increases the computational burden. We need to consider all these differences to design accurate motion estimation methods that can serve as a reliable tool for applications in many fields.

APPENDIX A

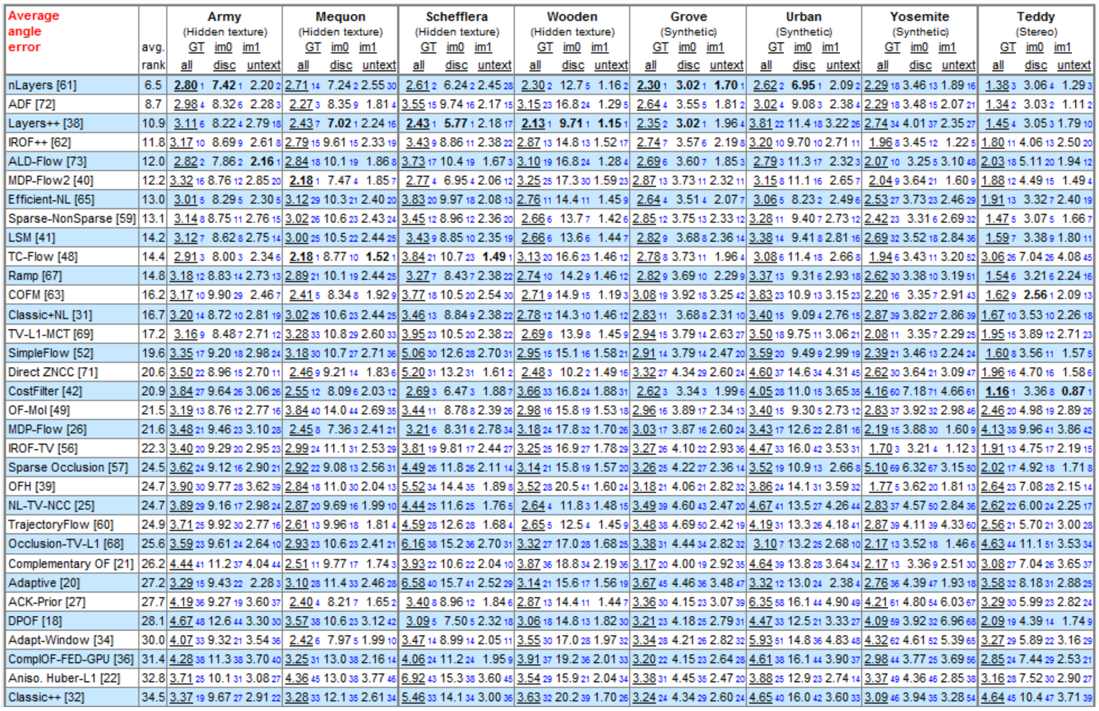
Detailed Tables for Chapter 4

- Figures A.1 and A.2 show the screen shot for the top-performing methods at the time of writing the dissertation (July 2012).
- Tables A.1 and A.2 summarize the EPE and AAE results on the Middlebury optical flow benchmark.
- Tables A.3 and A.4 summarize the EPE and AAE results on the Middlebury *training set* for the classical formulations (***Table 2** in Chapter 4*). Note that **Classic-C-brightness** actually achieves lower EPE and AAE on the training set than **Classic-C** but significantly lower accuracy on the test set, suggesting overfitting to the training data.
- Table A.5 shows the EPE results on the Middlebury *training set* for the baseline model (**Classic-C**) using different pre-processing techniques (***Table 2** Chapter 4*). Two additional preprocessing techniques are provided that are not shown in the main text since neither is significantly different from the baseline **Classic-C**. **Dx+Dy** assumes separate horizontal derivative and vertical derivative constancy. A weighted combination of robust functions applied to each term is used. By default, the blended result from texture decomposition is normalized to $[-1, 1]$ in [**185**] and $[0, 255]$ in our experiment. **Unnormalized texture** tests the effect of not doing this normalization.
- Table A.6 shows some additional results that are not shown in the main paper. They focus on variations of the **HS** optimization and, while interesting, are not central to the main claims of the paper. One interesting result is that repeatedly applying median filtering (20 times) at every warping step improves the **HS** formulation and the improvement is statistically significant (**HS** 20× **MF**). We also find that, for **HS**, the downsampling factor in the pyramid is not significant: a downsampling factor of 0.95 (**HS-Down-0.95**) produces similar results to one of 0.5 (**HS**).
- Table A.7 shows the EPE results on the Middlebury *training set* for the baseline model (**Classic-C**) using different algorithmic and modeling choices (***Table 3** Chapter 4*).
- Tables A.8 and A.9 shows EPE results for the new objective function with alternating optimization (**-A**) and its improved model (**-NL**); this corresponds to ***Table 5** Chapter 4*.



Average endpoint error	avg.	Army (Hidden texture)			Mequon (Hidden texture)			Schefflera (Hidden texture)			Wooden (Hidden texture)			Grove (Synthetic)			Urban (Synthetic)			Yosemite (Synthetic)			Teddy (Stereo)		
		GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
ADF [72]	8.6	0.08 4	0.22 5	0.06 1	0.18 3	0.62 8	0.14 8	0.29 14	0.71 17	0.17 7	0.16 15	0.91 25	0.07 1	0.69 12	1.03 12	0.47 8	0.43 11	0.91 3	0.28 6	0.12 15	0.12 6	0.20 18	0.43 1	0.88 1	0.63 5
IROF++ [62]	9.0	0.08 4	0.23 7	0.07 8	0.21 17	0.68 15	0.17 18	0.28 10	0.63 8	0.19 18	0.15 8	0.73 10	0.09 12	0.60 3	0.89 4	0.42 5	0.43 11	1.08 10	0.31 12	0.10 4	0.12 6	0.12 4	0.47 5	0.98 7	0.68 10
Layers++ [38]	9.2	0.08 4	0.21 2	0.07 8	0.19 7	0.56 3	0.17 18	0.20 1	0.40 1	0.18 12	0.13 1	0.58 2	0.07 1	0.48 1	0.70 1	0.33 2	0.47 19	1.01 6	0.33 18	0.15 37	0.14 30	0.24 31	0.46 3	0.88 1	0.72 13
MDP-Flow2 [40]	9.4	0.09 15	0.23 7	0.07 8	0.16 2	0.52 1	0.13 3	0.22 2	0.46 3	0.17 7	0.17 21	0.93 31	0.09 12	0.65 7	0.98 9	0.43 6	0.29 1	0.91 3	0.26 4	0.11 9	0.13 18	0.17 11	0.51 15	1.11 18	0.72 13
nLayers [61]	9.8	0.07 1	0.19 1	0.06 1	0.22 20	0.59 4	0.19 29	0.25 6	0.54 5	0.20 27	0.15 8	0.84 18	0.08 6	0.53 2	0.78 2	0.34 3	0.44 14	0.84 1	0.30 10	0.13 23	0.13 18	0.20 18	0.47 5	0.97 6	0.67 8
Sparse-NonSparse [59]	13.4	0.08 4	0.23 7	0.07 8	0.22 20	0.73 22	0.18 21	0.28 10	0.64 9	0.19 18	0.14 4	0.71 7	0.08 6	0.67 11	0.99 11	0.48 11	0.49 23	1.06 8	0.32 15	0.14 29	0.11 1	0.28 44	0.49 9	0.98 7	0.73 17
ALD-Flow [73]	13.5	0.07 1	0.21 2	0.06 1	0.19 7	0.64 13	0.13 3	0.30 17	0.73 18	0.15 2	0.17 21	0.92 29	0.07 1	0.78 21	1.14 21	0.59 21	0.33 3	1.30 22	0.21 1	0.12 15	0.12 6	0.28 44	0.54 18	1.19 21	0.73 17
COFM [63]	13.6	0.08 4	0.26 24	0.06 1	0.18 3	0.62 8	0.14 8	0.30 17	0.74 19	0.19 18	0.15 8	0.86 20	0.07 1	0.79 22	1.14 21	0.74 36	0.35 5	0.87 2	0.28 6	0.14 29	0.12 6	0.28 44	0.49 9	0.94 4	0.71 12
Efficient-NL [65]	13.9	0.08 4	0.22 5	0.06 1	0.23 30	0.73 22	0.18 21	0.32 22	0.75 20	0.18 12	0.14 4	0.72 9	0.08 6	0.60 3	0.88 3	0.43 6	0.57 32	1.11 14	0.35 24	0.14 29	0.13 18	0.25 34	0.48 7	0.90 3	0.63 5
TC-Flow [48]	14.0	0.07 1	0.21 2	0.06 1	0.15 1	0.59 4	0.11 1	0.31 20	0.78 23	0.14 1	0.16 15	0.86 20	0.08 6	0.75 18	1.11 19	0.54 16	0.42 10	1.40 27	0.25 3	0.11 9	0.12 6	0.29 49	0.62 24	1.35 25	0.93 36
LSM [41]	14.5	0.08 4	0.23 7	0.07 8	0.22 20	0.73 22	0.18 21	0.28 10	0.64 9	0.19 18	0.14 4	0.70 5	0.09 12	0.66 8	0.97 7	0.48 11	0.50 24	1.06 8	0.33 18	0.15 37	0.12 6	0.29 49	0.50 12	0.99 10	0.73 17
Ramp [67]	14.6	0.08 4	0.24 14	0.07 8	0.21 17	0.72 20	0.18 21	0.27 7	0.62 7	0.19 18	0.15 8	0.71 7	0.09 12	0.66 8	0.97 7	0.49 13	0.51 26	1.09 11	0.34 23	0.15 37	0.12 6	0.30 52	0.48 7	0.96 5	0.72 13
Classic+NL [31]	16.2	0.08 4	0.23 7	0.07 8	0.22 20	0.74 26	0.18 21	0.29 14	0.65 14	0.19 18	0.15 8	0.73 10	0.09 12	0.64 6	0.93 6	0.47 8	0.52 27	1.12 15	0.33 18	0.16 44	0.13 18	0.29 49	0.49 9	0.98 7	0.74 21
TV-L1-MCT [69]	16.6	0.08 4	0.23 7	0.07 8	0.24 33	0.77 28	0.19 29	0.32 22	0.76 22	0.19 18	0.14 4	0.69 4	0.09 12	0.72 14	1.03 12	0.60 22	0.54 30	1.10 13	0.35 24	0.11 9	0.12 6	0.20 18	0.54 18	1.04 14	0.84 28
Direct ZNCC [71]	17.0	0.09 15	0.25 18	0.07 8	0.19 7	0.70 18	0.13 3	0.43 30	1.00 33	0.15 2	0.13 1	0.55 1	0.08 6	0.86 29	1.23 27	0.73 32	0.53 29	1.22 17	0.38 32	0.14 29	0.13 18	0.27 41	0.44 2	0.99 10	0.44 1
IROF-TV [56]	18.2	0.09 15	0.25 18	0.08 22	0.22 20	0.77 28	0.19 29	0.30 17	0.70 16	0.19 18	0.18 25	0.93 31	0.11 28	0.73 15	1.04 14	0.56 19	0.44 14	1.69 42	0.31 12	0.09 3	0.11 1	0.12 4	0.50 12	1.08 16	0.73 17
MDP-Flow [26]	19.1	0.09 15	0.25 18	0.08 22	0.19 7	0.54 2	0.18 21	0.24 4	0.55 6	0.20 27	0.16 15	0.91 25	0.09 12	0.74 16	1.06 16	0.61 24	0.46 18	1.02 7	0.35 24	0.12 15	0.14 30	0.17 11	0.78 40	1.68 43	0.97 40
OF-Mol [49]	20.9	0.08 4	0.23 7	0.07 8	0.28 41	0.99 45	0.20 36	0.28 10	0.64 9	0.19 18	0.16 15	0.80 14	0.09 12	0.75 18	1.12 20	0.50 14	0.52 27	1.09 11	0.33 18	0.16 44	0.13 18	0.30 52	0.56 21	1.08 16	0.78 23
Sparse Occlusion [57]	21.0	0.09 15	0.24 14	0.08 22	0.22 20	0.63 10	0.19 29	0.38 28	0.91 27	0.18 12	0.17 21	0.85 19	0.09 12	0.75 18	1.09 18	0.47 8	0.34 4	1.00 5	0.26 4	0.22 62	0.22 67	0.28 44	0.53 17	1.13 19	0.67 8
OFH [39]	21.2	0.10 25	0.25 18	0.09 31	0.19 7	0.69 16	0.14 8	0.43 30	1.02 35	0.17 7	0.17 21	1.08 38	0.08 6	0.87 30	1.25 29	0.73 32	0.43 11	1.69 42	0.32 15	0.10 4	0.13 18	0.18 14	0.59 23	1.40 28	0.74 21
TrajectoryFlow [60]	21.9	0.10 25	0.26 24	0.07 8	0.20 14	0.73 22	0.13 3	0.37 27	0.94 28	0.15 2	0.13 1	0.67 3	0.07 1	0.82 24	1.23 27	0.54 16	0.66 35	1.42 30	0.44 33	0.16 44	0.14 30	0.37 60	0.58 22	1.25 22	0.79 24
NL-TV-NCC [25]	22.0	0.10 25	0.26 24	0.08 22	0.22 20	0.72 20	0.15 12	0.35 25	0.85 25	0.16 5	0.15 8	0.70 5	0.09 12	0.79 22	1.16 24	0.51 15	0.78 40	1.38 24	0.48 38	0.16 44	0.15 40	0.26 36	0.55 20	1.16 20	0.55 3
CostFilter [42]	22.1	0.10 25	0.27 30	0.08 22	0.20 14	0.63 10	0.15 12	0.22 2	0.45 2	0.18 12	0.19 30	0.88 24	0.12 32	0.60 3	0.90 5	0.28 1	0.75 39	1.19 16	0.50 39	0.21 61	0.24 70	0.40 62	0.46 3	1.02 12	0.62 4
SimpleFlow [52]	23.5	0.09 15	0.24 14	0.08 22	0.24 33	0.78 30	0.20 36	0.43 30	0.96 31	0.21 31	0.16 15	0.77 12	0.09 12	0.71 13	1.04 14	0.55 18	1.47 56	1.59 38	0.76 50	0.13 23	0.12 6	0.22 26	0.50 12	1.04 14	0.72 13
Occlusion-TV-L1 [68]	24.8	0.09 15	0.26 24	0.07 8	0.22 20	0.74 26	0.18 21	0.51 38	1.15 40	0.21 31	0.18 25	0.91 25	0.10 25	0.87 30	1.25 29	0.72 29	0.47 19	1.38 24	0.36 29	0.10 4	0.12 6	0.11 2	0.83 42	1.78 45	0.96 39
Adaptive [20]	26.8	0.09 15	0.26 24	0.06 1	0.23 30	0.78 30	0.18 21	0.54 40	1.19 44	0.21 31	0.18 25	0.91 25	0.10 25	0.88 33	1.25 29	0.73 32	0.50 24	1.28 20	0.31 12	0.14 29	0.16 47	0.22 26	0.65 28	1.37 27	0.79 24
DPOF [18]	27.8	0.12 42	0.33 42	0.08 22	0.26 37	0.80 34	0.20 36	0.24 4	0.49 4	0.20 27	0.19 30	0.83 17	0.13 35	0.66 8	0.98 9	0.40 4	1.11 48	1.41 29	0.57 43	0.25 68	0.14 30	0.55 68	0.51 15	1.02 12	0.54 2
Adapt-Window [34]	27.8	0.10 25	0.24 14	0.09 31	0.19 7	0.59 4	0.15 12	0.27 7	0.64 9	0.17 7	0.18 25	0.82 16	0.11 28	0.74 16	1.07 17	0.56 19	1.78 65	1.73 45	0.95 60	0.22 62	0.16 47	0.45 65	0.70 32	1.28 23	0.88 32
ACK-Prior [27]	28.6	0.11 32	0.25 18	0.09 31	0.18 3	0.59 4	0.13 3	0.27 7	0.64 9	0.16 5	0.15 8	0.78 13	0.09 12	0.82 24	1.14 21	0.71 28	1.90 67	1.90 52	0.99 63	0.23 66	0.17 53	0.49 67	0.77 38	1.44 29	0.91 34
Complementary OF [21]	29.3	0.11 32	0.28 31	0.10 40	0.18 3	0.63 10	0.12 2	0.31 20	0.75 20	0.18 12	0.19 30	0.97 33	0.12 32	0.97 44	1.31 40	1.00 49	1.78 65	1.73 45	0.87 58	0.11 9	0.12 6	0.22 26	0.68 29	1.48 30	0.95 37
ComplOF-FED-GPU [36]	30.6	0.11 32	0.29 34	0.10 40	0.21 17	0.78 30	0.14 8	0.32 22	0.79 24	0.17 7	0.19 30	0.99 34	0.11 28	0.89 34	1.29 35	0.73 32	1.25 50	1.74 47	0.64 47	0.14 29	0.13 18	0.30 52	0.64 26	1.50 32	0.83 27
Classic++ [32]	31.0	0.09 15	0.25 18	0.07 8	0.23 30	0.78 30	0.19 29	0.43 30	1.00 33	0.22 34	0.20 34	1.11 39	0.10 25	0.87 30	1.30 37	0.66 26	0.47 19	1.62 39	0.33 18	0.17 49	0.14 30	0.32 57	0.79 41	1.64 39	0.92 35

Figure A.1. Screen shot of Middlebury EPE table at the writing of the dissertation (July 2012). There are 73 methods in total and only the higher-ranking ones are shown.



Average angle error	avg.	Army (Hidden texture)			Mequon (Hidden texture)			Schefflera (Hidden texture)			Wooden (Hidden texture)			Grove (Synthetic)			Urban (Synthetic)			Yosemite (Synthetic)			Teddy (Stereo)		
		GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1
	rank	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 [61]	6.5	2.80 1	7.42 1	2.20 2	2.71 14	7.24 2	2.55 30	2.61 2	6.24 2	2.45 28	2.30 2	12.7 5	1.16 2	2.30 1	3.02 1	1.70 1	2.62 2	6.95 1	2.09 2	2.29 18	3.46 13	1.89 16	1.38 3	3.06 4	1.29 3
ADF [72]	8.7	2.98 4	8.32 6	2.28 3	2.27 3	8.35 9	1.81 4	3.55 15	9.74 16	2.17 15	3.15 23	16.8 24	1.29 5	2.64 4	3.55 5	1.81 2	3.02 4	9.08 3	2.38 4	2.29 18	3.48 15	2.07 21	1.34 2	3.03 2	1.11 2
Layers++ [38]	10.9	3.11 6	8.22 4	2.79 18	2.43 7	7.02 1	2.24 16	2.43 1	5.77 1	2.18 17	2.13 1	9.71 1	1.15 1	2.35 2	3.02 1	1.96 4	3.81 22	11.4 18	3.22 26	2.74 34	4.01 37	2.35 27	1.45 4	3.05 3	1.79 10
IROF++ [62]	11.8	3.17 10	8.69 9	2.61 8	2.79 15	9.61 15	2.33 19	3.43 9	8.86 11	2.38 22	2.87 13	14.8 13	1.52 17	2.74 7	3.57 6	2.19 8	3.20 10	9.70 10	2.71 11	1.96 8	3.45 12	1.22 5	1.80 11	4.06 13	2.50 20
ALD-Flow [73]	12.0	2.82 2	7.86 2	2.16 1	2.84 18	10.1 19	1.86 8	3.73 17	10.4 19	1.67 3	3.10 19	16.8 24	1.28 4	2.69 6	3.60 7	1.85 3	2.79 3	11.3 17	2.32 3	2.07 10	3.25 5	3.10 48	2.03 18	5.11 20	1.94 12
MDP-Flow2 [40]	12.2	3.32 16	8.76 12	2.85 20	2.18 1	7.47 4	1.85 7	2.77 4	6.95 4	2.06 12	3.25 25	17.3 30	1.59 23	2.87 13	3.73 11	2.32 11	3.15 8	11.1 16	2.65 7	2.04 9	3.64 21	1.60 9	1.88 12	4.49 15	1.49 4
Efficient-NL [65]	13.0	3.01 5	8.29 5	2.30 5	3.12 29	10.3 21	2.40 20	3.83 20	9.97 18	2.08 13	2.76 11	14.4 11	1.45 9	2.64 4	3.51 4	2.07 7	3.06 5	8.23 2	2.49 6	2.53 27	3.73 23	2.46 29	1.91 13	3.32 7	2.40 19
Sparse-NonSparse [59]	13.1	3.14 8	8.75 11	2.76 15	3.02 26	10.6 23	2.43 24	3.45 12	8.96 12	2.36 20	2.66 6	13.7 7	1.42 6	2.85 12	3.75 13	2.33 12	3.28 11	9.40 7	2.73 12	2.42 23	3.31 6	2.69 32	1.47 5	3.07 5	1.66 7
LSM [41]	14.2	3.12 7	8.62 8	2.75 14	3.00 25	10.5 22	2.44 25	3.43 9	8.85 10	2.35 19	2.66 6	13.6 6	1.44 7	2.82 9	3.68 8	2.36 14	3.38 14	9.41 8	2.81 16	2.69 32	3.52 18	2.84 36	1.59 7	3.38 9	1.80 11
TC-Flow [48]	14.4	2.91 3	8.00 3	2.34 6	2.18 1	8.77 10	1.52 1	3.84 21	10.7 23	1.49 1	3.13 20	16.6 23	1.46 12	2.78 8	3.73 11	1.96 4	3.08 6	11.4 18	2.66 8	1.94 6	3.43 11	3.20 52	3.06 26	7.04 26	4.08 45
Ramp [67]	14.8	3.18 12	8.83 14	2.73 13	2.89 21	10.1 19	2.44 25	3.27 7	8.43 7	2.38 22	2.74 10	14.2 9	1.46 12	2.82 9	3.69 10	2.29 9	3.37 13	9.31 6	2.93 18	2.62 30	3.38 10	3.19 51	1.54 6	3.21 6	2.24 16
COFM [63]	16.2	3.17 10	9.90 29	2.46 7	2.41 5	8.34 8	1.92 9	3.77 18	10.5 20	2.54 30	2.71 9	14.9 15	1.19 3	3.08 19	3.92 18	3.25 42	3.83 23	10.9 13	3.15 23	2.20 16	3.35 7	2.91 43	1.62 9	2.56 1	2.09 13
Classic+NL [31]	16.7	3.20 14	8.72 10	2.81 19	3.02 26	10.6 23	2.44 25	3.46 13	8.84 9	2.38 22	2.78 12	14.3 10	1.46 12	2.83 11	3.68 8	2.31 10	3.40 15	9.09 4	2.76 15	2.87 39	3.82 27	2.86 39	1.67 10	3.53 10	2.26 18
TV-L1-MCT [69]	17.2	3.16 9	8.48 7	2.71 12	3.28 33	10.8 29	2.60 33	3.95 23	10.5 20	2.38 22	2.69 8	13.9 8	1.45 9	2.94 15	3.79 14	2.63 27	3.50 18	9.75 11	3.06 21	2.08 11	3.35 7	2.29 25	1.95 15	3.89 12	2.71 23
SimpleFlow [52]	19.6	3.35 17	9.20 18	2.98 24	3.18 30	10.7 27	2.71 36	5.06 30	12.6 28	2.70 31	2.95 15	15.1 16	1.58 21	2.91 14	3.79 14	2.47 20	3.59 20	9.49 9	2.99 19	2.39 21	3.46 13	2.24 24	1.60 8	3.56 11	1.57 5
Direct ZNCC [71]	20.6	3.50 22	8.96 15	2.70 11	2.46 9	9.21 14	1.83 6	5.20 31	13.2 31	1.61 2	2.48 3	10.2 2	1.49 16	3.32 27	4.34 29	2.60 24	4.60 37	14.6 34	4.31 45	2.62 30	3.64 21	3.09 47	1.96 16	4.70 16	1.58 6
CostFilter [42]	20.9	3.84 27	9.64 26	3.06 26	2.55 12	8.09 6	2.03 12	2.69 3	6.47 3	1.88 7	3.66 33	16.8 24	1.88 31	2.62 3	3.34 3	1.99 6	4.05 28	11.0 15	3.65 35	4.16 60	7.18 71	4.66 61	1.16 1	3.36 8	0.87 1
OF-Mol [49]	21.5	3.19 13	8.76 12	2.77 16	3.84 40	14.0 44	2.69 35	3.44 11	8.78 8	2.39 26	2.98 16	15.8 19	1.53 18	2.96 16	3.89 17	2.34 13	3.40 15	9.30 5	2.73 12	2.83 37	3.92 32	2.98 46	2.46 20	4.98 19	2.89 26
MDP-Flow [26]	21.6	3.48 21	9.46 23	3.10 28	2.45 8	7.36 3	2.41 21	3.21 6	8.31 6	2.78 34	3.18 24	17.8 32	1.70 26	3.03 17	3.87 16	2.60 24	3.43 17	12.6 22	2.81 16	2.19 15	3.88 30	1.60 9	4.13 38	9.96 41	3.86 42
IROF-TV [56]	22.3	3.40 20	9.29 20	2.95 23	2.99 24	11.1 31	2.53 29	3.81 19	9.81 17	2.44 27	3.25 25	16.9 27	1.78 29	3.27 26	4.10 22	2.93 36	4.47 33	16.0 42	3.53 31	1.70 3	3.21 4	1.12 3	1.91 13	4.75 17	2.19 15
Sparse Occlusion [57]	24.5	3.62 24	9.12 16	2.90 21	2.92 22	9.08 13	2.56 31	4.49 26	11.8 26	2.11 14	3.14 21	15.8 19	1.57 20	3.26 25	4.22 27	2.36 14	3.52 19	10.9 13	2.66 8	5.10 69	6.32 67	3.15 50	2.02 17	4.92 18	1.71 8
OFH [39]	24.7	3.90 30	9.77 28	3.62 39	2.84 18	11.0 30	2.04 13	5.52 34	14.4 35	1.89 8	3.52 28	20.5 41	1.60 24	3.18 21	4.06 21	2.82 32	3.86 24	14.1 31	3.59 32	1.77 5	3.62 20	1.81 13	2.64 23	7.08 28	2.15 14
NL-TV-NCC [25]	24.7	3.89 29	9.16 17	2.98 24	2.87 20	9.69 16	1.99 10	4.44 25	11.6 25	1.76 5	2.64 4	11.8 3	1.48 15	3.49 39	4.60 43	2.47 20	4.67 41	13.5 27	4.26 44	2.83 37	4.57 50	2.84 36	2.62 22	6.00 24	2.25 17
TrajectoryFlow [60]	24.9	3.71 25	9.92 30	2.77 16	2.61 13	9.96 18	1.81 4	4.59 28	12.6 28	1.68 4	2.65 5	12.5 4	1.45 9	3.48 38	4.69 50	2.42 19	4.19 31	13.3 26	4.18 41	2.87 39	4.11 39	4.33 60	2.56 21	5.70 21	3.00 28
Occlusion-TV-L1 [68]	25.6	3.59 23	9.61 24	2.64 10	2.93 23	10.6 23	2.41 21	6.16 38	15.2 36	2.70 31	3.32 27	17.0 28	1.68 25	3.38 31	4.44 34	2.82 32	3.10 7	13.2 25	2.68 10	2.17 13	3.52 18	1.46 6	4.63 44	11.1 51	3.53 34
Complementary OF [21]	26.2	4.44 41	11.2 37	4.04 44	2.51 11	9.77 17	1.74 3	3.93 22	10.6 22	2.04 10	3.87 36	18.8 34	2.19 36	3.17 20	4.00 19	2.92 35	4.64 39	13.8 28	3.64 34	2.17 13	3.36 9	2.51 30	3.08 27	7.04 26	3.65 37
Adaptive [20]	27.2	3.29 15	9.43 22	2.28 3	3.10 28	11.4 33	2.46 28	6.58 40	15.7 41	2.52 29	3.14 21	15.6 17	1.56 19	3.67 45	4.46 36	3.48 47	3.32 12	13.0 24	2.38 4	2.76 36	4.39 47	1.93 18	3.58 32	8.18 31	2.88 25
ACK-Prior [27]	27.7	4.19 36	9.27 19	3.60 37	2.40 4	8.21 7	1.65 2	3.40 8	8.96 12	1.84 6	2.87 13	14.4 11	1.44 7	3.36 30	4.15 23	3.07 39	6.35 58	16.1 44	4.90 49	4.21 61	4.80 54	6.03 67	3.29 30	5.99 23	2.82 24
DPOF [18]	28.1	4.67 48	12.6 44	3.30 30	3.57 38	10.6 23	3.12 42	3.09 5	7.50 5	2.32 18	3.06 18	14.8 13	1.82 30	3.21 23	4.18 25	2.79 31	4.47 33	12.5 21	3.33 27	4.09 59	3.92 32	6.96 68	2.09 19	4.39 14	1.74 9
Adapt-Window [34]	30.0	4.07 33	9.32 21	3.54 36	2.42 6	7.97 5	1.99 10	3.47 14	8.99 14	2.05 11	3.55 30	17.0 28	1.97 32	3.34 28	4.21 26	2.82 32	5.93 51	14.8 36	4.83 48	4.32 62	4.61 52	5.39 65	3.27 29	5.89 22	3.16 29
ComplOF-FED-GPU [36]	31.4	4.28 38	11.3 38	3.70 40	3.25 31	13.0 38	2.16 14	4.06 24	11.2 24	1.95 9	3.91 37	19.2 36	2.01 33	3.20 22	4.15 23	2.64 28	4.61 38	16.1 44	3.90 37	2.98 44	3.77 25	3.69 56	2.85 24	7.44 29	2.53 21
Aniso. Huber-L1 [22]	32.8	3.71 25	10.1 31	3.08 27	4.36 45	13.0 38	3.77 46	6.92 43	15.3 38	3.60 45	3.54 29	15.9 21	2.04 34	3.38 31	4.45 35	2.47 20	3.88 25	12.9 23	2.74 14	3.37 49	4.36 46	2.85 38	3.16 28	7.52 30	2.90 27
Classic++ [32]	34.5	3.37 19	9.67 27	2.91 22	3.28 33	12.1 35	2.61 34	5.46 33	14.1 34	3.00 36	3.63 32	20.2 39	1.70 26	3.24 24	4.34 29	2.60 24	4.65 40	16.0 42	3.60 33	3.09 46	3.94 35	3.28 54	4.64 45	10.4 47	3.71 39

Figure A.2. Screen shot of Middlebury AAE table at the writing of the dissertation (July 2012). There are 73 methods in total and only the higher-ranking ones are shown.

Table A.1. Models. Average end-point error (EPE) on the Middlebury optical flow benchmark (*test set*). The ranking information was at the publication of our conference paper [**158**] (June 2010); the average EPE ranks for **Adaptive**, **Complementary OF**, **Classic++**, and **Classic+NL** are 26.8, 29.3, 31.0, and 16.2 at the writing of the dissertation (July 2012).

	Rank	**Average**	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
HS	24.6	0.501	0.12	0.25	0.45	0.24	0.95	0.83	0.24	0.93
Classic-C	14.9	0.408	0.10	0.23	0.45	0.20	0.88	0.47	0.16	0.77
Classic-L	19.8	0.530	0.10	0.24	0.47	0.21	0.92	1.23	0.20	0.87
HS-brightness	N/A	0.759	0.21	0.89	1.13	0.42	0.93	0.70	0.18	1.61
Classic-C-brightness	N/A	0.726	0.39	0.95	1.12	0.42	0.87	0.48	0.13	1.45
Classic-L-brightness	N/A	0.603	0.17	0.64	0.84	0.32	0.90	0.48	0.13	1.34
HS [**159**]	35.1	0.872	0.22	0.61	1.01	0.78	1.26	1.43	0.16	1.51
BA (Classic-L) [**159**]	30.9	0.746	0.18	0.58	0.95	0.49	1.08	1.43	0.15	1.11
Adaptive [**184**]	11.5	0.401	0.09	0.23	0.54	0.18	0.88	0.50	0.14	0.65
Complementary OF [**204**]	10.1	0.485	0.10	0.20	0.35	0.19	0.87	1.46	0.11	0.60
Classic++	13.4	0.406	0.09	0.23	0.43	0.20	0.87	0.47	0.17	0.79
Classic++Gradient	15.1	0.430	0.08	0.17	0.49	0.21	0.94	0.55	0.17	0.83
Classic+NL	6.6	0.319	0.08	0.22	0.29	0.15	0.64	0.52	0.16	0.49
Classic+NL-Full	6.8	0.316	0.08	0.24	0.28	0.15	0.63	0.49	0.16	0.50

Table A.2. Models. Average angular error (AAE) on the Middlebury optical flow benchmark (*test set*). The ranking information was at the publication of our conference paper [**158**] (June 2010); the average AAE ranks for **Adaptive**, **Complementary OF**, **Classic++**, and **Classic+NL** are 26.2, 27.2, 34.5, and 16.7 at the writing of the dissertation (July 2012).

	Rank	**Average**	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
HS	27.1	4.914	4.45	3.76	5.73	4.69	3.63	6.59	4.91	5.55
Classic-C	16.1	3.971	3.76	3.28	5.77	3.78	3.28	4.54	2.97	4.39
Classic-L	19.8	4.353	3.79	3.43	5.88	3.92	3.34	5.23	3.72	5.51
HS-brightness	N/A	8.465	7.66	12.30	15.30	8.08	3.56	5.95	3.87	11.00
Classic-C-brightness	N/A	7.938	8.79	13.40	14.80	7.23	3.29	4.10	2.57	9.32
Classic-L-brightness	N/A	6.544	6.30	8.72	11.70	5.99	3.31	4.52	2.70	9.11
HS [**159**]	36.4	8.720	8.01	9.13	14.20	12.40	4.64	8.21	4.01	9.16
BA (Classic-L) [**159**]	31.8	7.165	6.81	8.77	13.00	8.29	4.18	6.19	3.63	6.45
Adaptive [**184**]	12.1	3.680	3.29	3.10	6.58	3.14	3.67	3.32	2.76	3.58
Complementary OF [**204**]	11.3	3.476	4.44	2.51	3.93	3.87	3.17	4.64	2.17	3.08
Classic++	15.1	3.920	3.37	3.28	5.46	3.63	3.24	4.65	3.09	4.64
Classic++Gradient	14.8	3.981	3.10	2.42	5.93	3.90	3.28	4.38	3.22	5.62
Classic+NL	6.0	2.904	3.20	3.02	3.46	2.78	2.83	3.40	2.87	1.67
Classic+NL-Full	6.1	2.843	3.23	3.23	3.34	2.73	2.73	3.03	2.83	1.62

Table A.3. Models and pre-processing. Average end-point error (EPE) on the Middlebury *training set* for the classical model and different penalty functions. By default, the input sequences were preprocessed using ROF texture decomposition; "brightness" means no preprocessing is performed. The statistical significance is tested using the Wilcoxon signed rank test between each method and the baseline (**Classic-C**).

	Average	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3	signif.	p-value
Classic-C	0.298	0.281	0.152	0.165	0.093	0.158	0.627	0.348	0.562	—	—
Classic-C-brightness	0.288	0.268	0.166	0.215	0.134	0.146	0.584	0.352	0.437	0	0.9453
HS	0.384	0.337	0.219	0.189	0.118	0.204	0.688	0.463	0.853	**1**	**0.0078**
HS-brightness	0.387	0.335	0.226	0.252	0.154	0.185	0.639	0.564	0.743	**1**	**0.0078**
Classic-L	0.319	0.294	0.193	0.175	0.095	0.166	0.648	0.374	0.604	**1**	**0.0078**
Classic-L-brightness	0.325	0.292	0.207	0.274	0.145	0.158	0.588	0.451	0.484	0	0.2969

Table A.4. Models and pre-processing. Average angular error (AAE) on the Middlebury *training set* for the classical model and different penalty functions. By default, the input sequences were preprocessed using ROF texture decomposition; 'brightness' means no preprocessing is performed. The statistical significance is tested using the Wilcoxon signed rank test between each method and the baseline (**Classic-C**).

	Average	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3	signif.	p-value
Classic-C	3.518	4.463	3.097	1.960	2.981	2.298	6.116	2.529	4.698	—	—
Classic-C-brightness	3.580	4.059	3.270	2.449	4.321	2.127	5.683	2.684	4.049	0	1.0000
HS	4.660	5.486	4.562	2.209	3.801	2.850	6.776	4.078	7.519	**1**	**0.0078**
HS-brightness	4.782	5.513	4.621	2.892	4.977	2.581	6.156	4.859	6.658	**1**	**0.0078**
Classic-L	3.910	4.993	3.928	3.022	4.543	2.178	5.756	2.876	3.980	0	0.2500
Classic-L-brightness	3.874	4.635	4.181	3.323	4.502	2.221	5.709	2.970	3.452	0	0.3125

Table A.5. Pre-Processing. Average end-point error (EPE) on the Middlebury *training set* for the baseline method (**Classic-C**) using different pre-processing techniques. The regularization weight λ parameter was tuned for each method to achieve optimal performance. The statistical significance is tested using the Wilcoxon signed rank test between each method and the baseline (**Classic-C**).

	Average	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3	signif.	*p*-value
Classic-C	0.298	0.281	0.152	0.165	0.093	0.158	0.627	0.348	0.562	—	—
Gradient	0.305	0.288	0.141	0.167	0.092	0.165	0.614	0.385	0.588	0	0.4609
Gaussian	0.281	0.268	0.146	0.226	0.141	0.137	0.582	0.335	0.413	0	0.5469
Gaussian + Dx + Dy	0.290	0.280	0.126	0.174	0.105	0.154	0.588	0.470	0.420	0	0.6406
Dx + Dy	0.301	0.286	0.122	0.166	0.099	0.161	0.616	0.443	0.518	0	1.0000
Sobel edge[**172**]	0.417	0.334	0.149	0.184	0.130	0.194	0.757	0.451	1.135	**1**	**0.0156**
Laplacian [**92**]	0.430	0.374	0.170	0.176	0.096	0.175	0.756	0.464	1.232	**1**	**0.0078**
Laplacian 1:1	0.301	0.296	0.179	0.193	0.109	0.157	0.606	0.349	0.520	0	0.6641
Texture 4:1	0.286	0.271	0.159	0.175	0.100	0.154	0.587	0.349	0.490	0	0.5312
Unnormalized texture	0.298	0.279	0.152	0.166	0.092	0.158	0.623	0.348	0.563	0	0.3750

Table A.6. Additional results for HS. Average end-point error (EPE) on the Middlebury *training set*. The statistical significance is tested using the Wilcoxon signed rank test between each method and **HS**.

	Average	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3	signif.	*p*-value
HS	0.384	0.337	0.219	0.189	0.118	0.204	0.688	0.463	0.853	—	—
HS-Down-0.95	0.386	0.333	0.220	0.189	0.117	0.200	0.651	0.522	0.856	0	0.8125
HS 20× **MF**	0.365	0.299	0.214	0.184	0.104	0.196	0.699	0.431	0.792	**1**	**0.0469**

Table A.7. Model and Methods. Average end-point error (EPE) on the Middlebury *training set* for the baseline model (**Classic-C**) using different algorithm and modeling choices. The statistical significance is tested using the Wilcoxon signed rank test between each method and the baseline (**Classic-C**).

	Average	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3	signif.	p-value
Classic-C	0.298	0.281	0.152	0.165	0.093	0.158	0.627	0.348	0.562	—	—
3 **warping steps**	0.304	0.283	0.122	0.163	0.095	0.150	0.622	0.357	0.644	0	0.9688
Down-0.5	0.298	0.280	0.152	0.166	0.092	0.158	0.626	0.349	0.562	0	1.0000
Down-0.95	0.298	0.281	0.151	0.168	0.099	0.165	0.661	0.339	0.523	0	0.9375
w/o GNC	0.354	0.303	0.160	0.171	0.105	0.183	0.835	0.316	0.759	0	0.1094
Bilinear	0.302	0.284	0.144	0.167	0.099	0.160	0.637	0.363	0.563	0	0.1016
w/o TAVG	0.306	0.288	0.149	0.167	0.093	0.163	0.647	0.345	0.593	0	0.1562
Central	0.300	0.272	0.156	0.169	0.092	0.159	0.608	0.349	0.597	0	0.7266
7-point [**38**]	0.302	0.282	0.168	0.171	0.091	0.163	0.601	0.360	0.584	0	0.3125
Deriv-warp	0.297	0.283	0.153	0.165	0.092	0.159	0.636	0.333	0.552	0	0.9531
Bicubic-II	0.290	0.276	0.132	0.152	0.083	0.142	0.624	0.338	0.571	**1**	**0.0391**
Deriv-warp-II	0.287	0.264	0.155	0.152	0.085	0.145	0.616	0.333	0.546	**1**	**0.0156**
Warp-deriv-II	0.288	0.267	0.155	0.151	0.085	0.147	0.630	0.328	0.542	**1**	**0.0391**
C-L ($\lambda = 0.6$)	0.303	0.290	0.158	0.171	0.094	0.158	0.611	0.367	0.579	0	0.1562
L-C ($\lambda = 2$)	0.306	0.281	0.174	0.173	0.096	0.164	0.662	0.343	0.557	0	0.1562
GC-0.45 ($\lambda = 3$)	0.292	0.280	0.145	0.165	0.092	0.154	0.612	0.340	0.546	**1**	**0.0156**
GC-0.25 ($\lambda = 0.7$)	0.298	0.283	0.128	0.169	0.094	0.150	0.617	0.353	0.594	0	1.0000
MF 3×3	0.305	0.287	0.155	0.168	0.094	0.162	0.616	0.372	0.583	0	0.1016
MF 7×7	0.305	0.281	0.152	0.173	0.095	0.174	0.676	0.330	0.557	0	0.5625
2× **MF**	0.300	0.279	0.152	0.167	0.093	0.163	0.650	0.339	0.555	0	1.0000
5× **MF**	0.305	0.278	0.152	0.171	0.093	0.172	0.682	0.329	0.561	0	0.6875
w/o MF	0.352	0.307	0.168	0.199	0.113	0.217	0.705	0.423	0.684	**1**	**0.0078**
Classic++	0.285	0.271	0.128	0.153	0.081	0.139	0.614	0.336	0.555	**1**	**0.0078**

Table A.8. Average end-point error (EPE) on the Middlebury *training set* for the proposed new objective with the area term and alternating optimization (**Classic-C-A**) and its improved models. The statistical significance is tested using the Wilcoxon signed rank test between each method and the baseline (**Classic-C**).

	Average	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3	signif.	*p*-value
Classic-C	0.298	0.281	0.152	0.165	0.093	0.158	0.627	0.348	0.562	—	—
Classic-C-A	0.305	0.281	0.140	0.159	0.092	0.167	0.676	0.334	0.594	0	0.8125
Classic-C-A-noRep	0.309	0.279	0.139	0.161	0.093	0.157	0.653	0.370	0.619	0	0.5781
Classic-C-A-II	0.296	0.278	0.153	0.166	0.091	0.168	0.656	0.329	0.531	0	0.7188
Classic-C-A-CGD	0.305	0.281	0.148	0.161	0.093	0.159	0.697	0.344	0.560	0	0.5625

Table A.9. Average end-point error (EPE) on the Middlebury *training set* for the proposed new objective with the weighted area term and its variants. The statistical significance is tested using the Wilcoxon signed rank test between each method and the baseline (**Classic+NL**).

	Average	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3	signif.	*p*-value
Classic+NL	0.221	0.238	0.131	0.152	0.073	0.103	0.468	0.220	0.384	—	—
Classic+NL-Full	0.222	0.252	0.135	0.156	0.074	0.097	0.469	0.214	0.382	0	0.8203
Classic+NL-Fast	0.221	0.233	0.117	0.151	0.076	0.098	0.464	0.210	0.421	0	0.3125
RGB	0.240	0.243	0.131	0.155	0.081	0.109	0.501	0.236	0.468	**1**	**0.0156**
HSV	0.231	0.245	0.131	0.152	0.074	0.110	0.492	0.222	0.424	**1**	**0.0312**
LUV	0.226	0.241	0.131	0.149	0.074	0.104	0.460	0.223	0.427	0	0.5625
Gray	0.253	0.253	0.133	0.158	0.086	0.125	0.547	0.242	0.479	**1**	**0.0078**
w/o color	0.283	0.258	0.128	0.157	0.087	0.155	0.633	0.303	0.543	**1**	**0.0156**
w/o occ	0.226	0.243	0.131	0.152	0.073	0.103	0.488	0.230	0.386	0	0.1250
w/o spa	0.223	0.237	0.132	0.154	0.073	0.102	0.475	0.213	0.398	0	0.5625
$\sigma_2 = 5$	0.221	0.240	0.131	0.151	0.073	0.104	0.466	0.208	0.392	0	1.0000
$\sigma_2 = 10$	0.224	0.238	0.132	0.153	0.073	0.102	0.485	0.228	0.384	0	0.2500
$\lambda = 1$	0.236	0.245	0.151	0.164	0.080	0.120	0.430	0.243	0.459	0	0.1406
$\lambda = 9$	0.244	0.249	0.137	0.160	0.091	0.111	0.577	0.201	0.426	0	0.1016
11×11	0.223	0.240	0.131	0.151	0.074	0.103	0.451	0.234	0.397	0	0.5938
19×19	0.220	0.238	0.132	0.154	0.073	0.103	0.470	0.210	0.384	0	0.8750

APPENDIX B

Gradient Formulae for Chapter 5

We derive the gradient for each individual term. From these individual terms, it is easy to obtain the gradient formula for the overall objective. Most of the derivations are straightforward and we only elaborate where there are subtle points.

B.1. Gradients w. r. t. the Support Function

Temporal coherence term. We can write the temporal coherence term using matrix vector multiplication form. Let $\mathbf{W}_{tk}$ be the bi-linear warping matrix according to the flow field $(\mathbf{u}_{tk}, \mathbf{v}_{tk})$ and $\mathbf{g}_{tk}(:)$ be the vectorized support function in a column-major way.

$$E_{\text{time}}(\mathbf{g}_{tk}, \mathbf{g}_{t+1,k}) = ||\mathbf{g}_{tk}(:) - \mathbf{W}_{tk}\mathbf{g}_{t+1,k}(:)||^2. \tag{64}$$

Using matrix derivative, we have

$$\nabla_{\mathbf{g}_{tk}} E_{\text{time}}(\mathbf{g}_{tk}, \mathbf{g}_{t+1,k}) = 2\Big(\mathbf{g}_{tk}(:) - \mathbf{W}_{tk}\mathbf{g}_{t+1,k}(:)\Big), \tag{65}$$

$$\nabla_{\mathbf{g}_{tk}} E_{\text{time}}(\mathbf{g}_{t-1,k}, \mathbf{g}_{tk}) = 2\mathbf{W}_{t-1,k}^{\mathcal{T}}\Big(\mathbf{W}_{t-1,k}\mathbf{g}_{tk}(:) - \mathbf{g}_{t-1,k}(:)\Big), \tag{66}$$

where $^{\mathcal{T}}$ means the matrix/vector transpose operator. Note that there is no need to construct the matrices $\mathbf{W}_{t-1,k}$ and $\mathbf{W}_{tk}$ in the implementation. We only need to perform the bi-linear interpolation operation and its transpose.

Data term. Similarly, the support function $\mathbf{g}_{tk}$ appears in the data term at time t and $t-1$. What is subtle is that the support function of a front layer influences the data term of the layers behind. We will first give the gradient of the soft layer assignment w. r. t. the support function, which plays a major role of the later derivations. Recall that the soft layer assignment is

$$\tilde{s}_{tk}^{p} = \begin{cases} \sigma(\lambda_e g_{tk}^p)\prod_{k'=1}^{k-1}\sigma(-\lambda_e g_{tk'}^p), & 1 \leq k < K \\ \prod_{k'=1}^{K-1}\sigma(-\lambda_e g_{tk'}^p), & k = K. \end{cases} \tag{67}$$

Using the property of the logistic function $\sigma'(x) = \sigma(x)\sigma(-x)$, we can obtain the gradient of the soft thresholding function (67) w. r. t. the support function as

$$\frac{\partial \tilde{s}_{tk}^p}{\partial g_{tl}^p} = \begin{cases} 0, & k < l \\ \lambda_e \tilde{s}_{tk}^p \sigma(-\lambda_e g_{tl}^p), & k = l \\ -\lambda_e \tilde{s}_{tk}^p \sigma(\lambda_e g_{tl}^p), & k > l. \end{cases} \tag{68}$$

Note that the $k < l$ case means that the support function of a layer behind does not influence the data term of the layers in front of it.

Let $\mathbf{M}_{tk}$ be a diagonal matrix with the element at the pth row pth column being $\sum_{q\in\mathcal{N}_{tk}^p}\rho_d(I_t^p-I_{t+1}^q)$. We can rewrite the data term as

$$E_{\text{data}}(\mathbf{u}_t,\mathbf{v}_t,\mathbf{g}_t,\mathbf{g}_{t+1})=\sum_{k=1}^{K}\tilde{\mathbf{s}}_{tk}^{\mathcal{T}}(:)\mathbf{M}_{tk}\mathbf{W}_{tk}\tilde{\mathbf{s}}_{t+1,k}(:) \tag{69}$$

and the gradients of the data term w. r. t. to the support functions are

$$\nabla_{\mathbf{g}_{tk}}E_{\text{data}}(\mathbf{u}_t,\mathbf{v}_t,\mathbf{g}_t,\mathbf{g}_{t+1})=\sum_{k=1}^{K}\nabla_{\mathbf{g}_{tk}}\tilde{\mathbf{s}}_{tk}^{\mathcal{T}}(:)\mathbf{M}_{tk}\mathbf{W}_{tk}\tilde{\mathbf{s}}_{t+1,k}(:) \tag{70}$$

$$\nabla_{\mathbf{g}_{tk}}E_{\text{data}}(\mathbf{u}_{t-1},\mathbf{v}_{t-1},\mathbf{g}_{t-1},\mathbf{g}_t)=\sum_{k=1}^{K}\tilde{\mathbf{s}}_{t-1,k}^{\mathcal{T}}(:)\mathbf{M}_{t-1,k}\mathbf{W}_{t-1,k}\nabla_{\mathbf{g}_{tk}}\tilde{\mathbf{s}}_{tk}(:), \tag{71}$$

where $\nabla_{\mathbf{g}_{tk}}\tilde{\mathbf{s}}_{tk}(:)$ is a diagonal matrix because $\frac{\partial\tilde{s}_{tk}^p}{\partial g_{tk}^q}=0$ for $p\neq q$.

Color-modulated spatial term. It is easy to obtain the gradient of E_{space} w. r. t. the support function because of its quadratic form:

$$\frac{\partial E_{\text{space}}(\mathbf{g}_{tk})}{\partial g_{tk}^p}=\sum_{q\in\Gamma^p}2w_q^p(g_{tk}^p-g_{tk}^q). \tag{72}$$

B.2. Gradients w. r. t. the Horizontal Flow Field

Due to symmetry, we only give the gradient formulae for the horizontal flow field; the vertical case is analogous.

Temporal coherence term. Using the chain rule, we obtain

$$\frac{\partial E_{\text{time}}(\mathbf{g}_{tk},\mathbf{g}_{t+1,k},\mathbf{u}_{tk},\mathbf{v}_{tk})}{\partial u_{tk}^p}=\sum_{q\in\Gamma_p}2\big(g_{t+1,k}^q-g_{tk}^p\big)\left(\frac{\partial g_{t+1,k}}{\partial x}\right)^q. \tag{73}$$

Data term. The data term is different from the standard data term for optical flow estimation in that the warped soft layer assignment $\tilde{s}_{t+1,k}(i+u_{tk}^p,j+v_{tk}^p)$ also depends on the flow field. As a result

$$\begin{aligned}\frac{\partial E_{\text{data}}(\mathbf{u}_t,\mathbf{v}_t,\mathbf{g}_t,\mathbf{g}_{t+1})}{\partial u_{tk}^p}=&-\rho_d'(\mathbf{I}_t^p-\mathbf{I}_{t+1}^q)\Big(\frac{\partial\mathbf{I}_{t+1}}{\partial x}\Big)^q\tilde{s}_{tk}^p\tilde{s}_{t+1,k}^q\\&+\Big(\rho_d(\mathbf{I}_t^p-\mathbf{I}_{t+1}^q)-\lambda_d\Big)\tilde{s}_{tk}^p\left(\frac{\partial\tilde{s}_{t+1,k}}{\partial x}\right)^q.\end{aligned} \tag{74}$$

Spatial prior term. Before each warping step, we estimate the affine flow field $(\bar{\mathbf{u}}_{\theta_{tk}},\bar{\mathbf{v}}_{\theta_{tk}})$ for each layer. We fix $\mathbf{u}_{tk}$ and minimize $E_{\text{aff}}(\mathbf{u}_{tk},\theta_{tk})$ w. r. t. the parameters θ_{tk} using iterated reweighted least squares methods.

With the affine flow field fixed, we can obtain the gradient of the spatial term w. r. t. the flow field as

$$\frac{\partial E_{\text{aff}}(\mathbf{u}_{tk},\theta_{tk})}{\partial u_{tk}^p}=\sum_{q\in\Gamma^p}\rho_s'\Big((u_{tk}^p-\bar{u}_{\theta_{tk}}^p)-(u_{tk}^q-\bar{u}_{\theta_{tk}}^q)\Big). \tag{75}$$

With these gradient formulae, it is straightforward to perform the incremental estimation for the flow field [**133**].

Bibliography

[1] `http://vision.middlebury.edu/flow`.

[2] `http://gpu4vision.icg.tugraz.at`.

[3] `http://www.cs.brown.edu/people/dqsun`.

[4] `http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize`.

[5] Yair Adato, Todd Zickler, and Ohad Ben-Shahar. A polar representation of motion and implications for optical flow. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1145–1152, 2011.

[6] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, November 1984.

[7] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sanchez. Symmetrical dense optical flow estimation with occlusions detection. *International Journal of Computer Vision*, 75(3):371–385, December 2007.

[8] P. Anandan and R. Weiss. Introducing a smoothness constraint in a matching approach for the computation of optical flow fields. In *DARPA Image Understanding Workshop*, pages 186–196, 1985.

[9] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 33(5):898–916, May 2011.

[10] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 26(3):10, July 2007.

[11] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *IEEE International Conference on Computer Vision*, pages 777–784, June 1995.

[12] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *International Journal of Computer Vision*, 97(3), May 2012.

[13] A. Bab-Hadiashar and D. Suter. Robust optic flow computation. *International Journal of Computer Vision*, 29(1):59–77, August 1998.

[14] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 28(3):70:1–70:11, July 2009.

[15] S. Baker and T. Kanade. Super-resolution optical flow. Technical report, CMU, 1999.

[16] S Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, March 2011.

[17] S Baker, D. Scharstein, J.P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[18] S. Baker, R.S. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 434–441, 1998.

[19] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.

[20] A.C. Berg, T.L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume I, pages 26–33, 2005.

[21] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, volume 588, pages 237–252, 1992.

[22] Stan Birchfield and Carlo Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *IEEE International Conference on Computer Vision*, pages 489–495, 1999.

[23] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[24] M. J. Black and P. Anandan. A model for the detection of motion over time. In *IEEE International Conference on Computer Vision*, pages 33–37, 1990.

[25] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 296–302, 1991.

[26] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63:75–104, 1996.

[27] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion boundaries. *International Journal of Computer Vision*, 38(3):231–245, July 2000.

[28] M. J. Black, Y. Yacoob, A. D. Jepson, and D. J. Fleet. Learning parameterized models of image motion. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 561–567, 1997.

[29] Michael J. Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *European Conference on Computer Vision*, pages 485–493, 1992.

[30] M.J. Black and A.D. Jepson. Estimating optical-flow in segmented images using variable-order parametric models with local deformations. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 18(10):972–986, October 1996.

[31] A. Blake and A. Zisserman. *Visual Reconstruction.* The MIT Press, Cambridge, Massachusetts, 1987.

[32] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 23(11):1222–1239, November 2001.

[33] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2009.

[34] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36, 2004.

[35] T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In *European Conference on Computer Vision*, volume I, pages 471–483, 2006.

[36] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision*, volume V, pages 282–295, 2010.

[37] Thomas Brox and Jitendra Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 33(3):500–513, March 2011.

[38] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, February 2005.

[39] A Buades, B Coll, and JM Morel. A non-local algorithm for image denoising. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65, 2005.

[40] P. J. Burt, C. Yen, and X. Xu. Local correlation measures for motion analysis: A comparative study. *Processings of IEEE Pattern Recognition and Image Processing*, pages 269–274, 1982.

[41] Daniel Butler, Jonas Wulff, and Michael Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, 2012.

[42] Zhuoyuan Chen, Ying Wu, and Jiang Wang. Decomposing and regularizing sparse/non-sparse components for motion field estimation. In *CVPR*, pages 1176–1183, 2012.

[43] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 24(5):603–619, May 2002.

[44] D. Cremers and S. Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005.

[45] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*, pages II: 428–441, 2006.

[46] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 17(5):474–487, May 1995.

[47] T.J. Darrell and A.P. Pentland. Robust estimation of a multi-layered motion representation. In *Workshop on Visual Motion*, pages 173–178, 1991.

[48] AP Dempster, NM Laird, and DB Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B*, 39:1–38, 1977.

[49] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[50] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[51] G. Farneback. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *IEEE International Conference on Computer Vision*, volume 1, pages 171–177, 2001.

[52] D. Feldman and D. Weinshall. Motion segmentation and depth ordering using an occlusion detector. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 30(7):1171–1185, July 2008.

[53] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, September 2004.

[54] Cornelia Fermüller, David Shulman, and Yiannis Aloimonos. The statistics of optical flow. *Computer Vision and Image Understanding*, 82(1):1–32, April 2001.

[55] D. J. Fleet, M. J. Black, and O. Nestares. Bayesian inference of visual motion boundaries. In *Exploring Artificial Intelligence in the New Millennium*, pages 139–174. Morgan Kaufmann Pub., 2002.

[56] David J. Fleet and Allan D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.

[57] William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, October 2000.

[58] B.J. Frey and N. Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 27(9):1392–1416, September 2005.

[59] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, Providence, USA, 2012.

[60] Michael A. Gennert and Shahriar Negahdaripour. Relaxing the brightness constancy assumption in computing optical flow. Technical report, MIT, Cambridge, MA, USA, 1987.

[61] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *SIAM Multiscale Modeling and Simulation*, 7:1005–1028, 2008.

[62] F. Glaer, G. Reynolds, and P. Anandan. Scene matching by hierarchical correlation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 432–441, 1983.

[63] B. Glocker, T. H. Heibel, N. Navab, P. Kohli, and C. Rother. Triangleflow: Optical flow with triangulation-based higher-order likelihoods. In *European Conference on Computer Vision*, pages 272–285, 2010.

[64] Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. Video object annotation, navigation, and composition. In *ACM Symposium on User Interface Software and Technology*, pages 3–12, 2008.

[65] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2141–2148, 2010.

[66] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0-1 optimization. *Mathematical Programming*, 28(2):121–155, February 1984.

[67] H.W. Haussecker and D.J. Fleet. Computing optical flow with physical models of brightness variation. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 23(6):661–673, June 2001.

[68] X.M. He and A.Y. Yuille. Occlusion boundary detection using pseudo-depth. In *European Conference on Computer Vision*, volume IV, pages 539–552, 2010.

[69] G E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, August 2002.

[70] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 31(9):1582–1599, September 2009.

[71] G. Hirzinger, K. Landgettel, and W. Snyder. Automated tv tracking of moving objects: The DFVLR tracker & related approaches. In *International Conference on Pattern Recognition*, pages 1255–1261, 1980.

[72] B.K.P. Horn. *Robot Vision*. MIT Press, 1986.

[73] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 16(1-3):185–203, August 1981.

[74] I.T. Hsiao, A. Rangarajan, and G. Gindi. A new convex edge-preserving median prior with applications to tomography. *IEEE Transactions on Medical Imaging*, 22(5):580–585, May 2003.

[75] S.C. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representations. In *International Conference on Pattern Recognition*, volume A, pages 743–746, 1994.

[76] A. Humayun, O. Mac Aodha, and G. J. Brostow. Learning to Find Occlusion Regions. In *IEEE International Conference on Computer Vision and Pattern Recognition*, number 2161–216, 2011.

[77] M. Irani, P. Anandan, and D. Weinshall. From reference frames to reference planes: Multi-view parallax geometry and applications. In *European Conference on Computer Vision*, volume II, pages 829–845, 1998.

[78] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 760–761, 1993.

[79] A.D. Jepson, D.J. Fleet, and M.J. Black. A layered motion representation with occlusion and compact spatial support. In *European Conference on Computer Vision*, volume I, pages 692–706, 2002.

[80] Kui Jia, Xiaogang Wang, and Xiaoou Tang. Optical flow estimation using learned sparse model. In *IEEE International Conference on Computer Vision*, pages 2391–2398, 2011.

[81] N. Jojic and B.J. Frey. Learning flexible sprites in video layers. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume I, pages 199–206, 2001.

[82] Michael I. Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[83] A. Kannan, B. Frey, and N. Jojic. A generative model of dense optical flow in layers. Technical Report TR PSI-2001-11, University of Toronto, August 2001.

[84] R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(6):1153–1160, December 1981.

[85] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts - a review. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 7:1274–1279, July 2007.

[86] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions via graph cuts. In *IEEE International Conference on Computer Vision*, pages 508–515, 2001.

[87] Philipp Krähenbühl and Vladlen Koltun. Efficient nonlocal regularization for optical flow. In *European Conference on Computer Vision*, 2012.

[88] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, February 2001.

[89] M.P. Kumar, P.H.S. Torr, and A. Zisserman. Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76(3):301–319, March 2008.

[90] R. Kumar, P. Anandan, and K. Hanna. Shape recovery from multiple views: A parallax based approach. In *International Conference on Pattern Recognition*, volume A, pages 685–688, 1994.

[91] C. Lei and Y.-H. Yang. Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In *IEEE International Conference on Computer Vision*, pages 1562–1569, 2009.

[92] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[93] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 32(8):1392–1405, August 2010.

[94] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3369–3376, 2011.

[95] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 23(3):303–308, August 2004.

[96] Yingying Li and Stanley Osher. A new median formula with applications to PDE based denoising. *Communications in Mathematical Sciences*, 7(3):741–753, September 2009.

[97] Y.P. Li and D.P. Huttenlocher. Learning for optical flow using stochastic optimization. In *European Conference on Computer Vision*, volume II, pages 379–391, 2008.

[98] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis.* PhD thesis, MIT, 2009.

[99] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[100] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson. Motion magnification. In *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, pages 519–526. ACM, 2005.

[101] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 33(12):2368–2382, December 2011.

[102] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 33(1):978–994, January 2011.

[103] Ce Liu and Deqing Sun. A bayesian approach to adaptive video super resolution. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 209–216, 2011.

[104] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[105] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conferences on Artificial Intelligence*, pages 674–679, 1981.

[106] Oisin Mac Aodha, Gabriel J. Brostow, and Marc Pollefeys. Segmenting video into classes of algorithm-suitability. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1778–1785, 2010.

[107] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman, New York, NY, USA, 1982.

[108] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application toevaluating segmentation algorithms and measuring ecological statistics. In *IEEE International Conference on Computer Vision*, volume 2, pages 416–423, 2001.

[109] B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):126–143, October 2001.

[110] J. McCann and N. S. Pollard. Local layering. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 28(3), August 2009.

[111] Étienne Mémin and Patrick Pérez. Joint estimation-segmentation of optic flow. In *European Conference on Computer Vision*, volume II, pages 563–577, 1998.

[112] Étienne Mémin and Patrick Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, February 2002.

[113] Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *Uncertainty in Artificial Intelligence*, pages 362–369, 2001.

[114] R. D. Morris, X. Descombes, and J. Zerubia. The Ising/Potts model is not well suited to segmentation tasks. In *Proceedings of the IEEE Digital Signal Processing Workshop*, pages 263–266, 1996.

[115] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 9(2):220–228, March 1987.

[116] K. Mutch and W. Thompson. Analysis of accretion and deletion at boundaries in dynamic scenes. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 7(2):133–138, March 1985.

[117] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 8(5):565–593, September 1986.

[118] R.K. Namdev, A. Kundu, K.M. Krishna, and C.V. Jawahar. Motion segmentation of multiple objects from a freely moving monocular camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4092–4099, 2012.

[119] M. Nicolescu and G. Medioni. Motion segmentation with accurate boundaries - a tensor voting approach. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 382–389, 2003.

[120] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *IEEE International Conference on Computer Vision*, pages 1583–1590, 2011.

[121] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, November 1988.

[122] Yuri Ostrovsky, Ethan Meyers, Suma Ganesh, Umang Mathur, and Pawan Sinha. Visual parsing after recovery from blindness. *Psychological Science*, 20(12):1484–1491, December 2009.

[123] M. Otte and H.H. Nagel. Optical flow estimation: Advances and comparisons. In *European Conference on Computer Vision*, volume A, pages 49–60, 1994.

[124] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, April 2006.

[125] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision*, 81(1):24–52, January 2009.

[126] W. H. Press, W. T. Vetterling, S. A.. Teukolsky, and B. P. Flannery. *Numerical Recipes in C++: the art of scientific computing*. Cambridge University Press, New York, NY, USA, 2002.

[127] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, December 1971.

[128] X. Ren. Local grouping for optical flow. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[129] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3017–3024, 2011.

[130] M.G. Ross and L.P. Kaelbling. Learning static object segmentation from motion segmentation. In *AAAI Conference on Artificial Intelligence*, volume 2, pages 956–961, 2005.

[131] M.G. Ross and L.P. Kaelbling. Segmentation according to natural examples: Learning static segmentation from motion segmentation. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 31(4):661–676, April 2009.

[132] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume II, pages 860–867, 2005.

[133] S. Roth and M. J. Black. On the spatial statistics of optical flow. *International Journal of Computer Vision*, 74(1):33–50, August 2007.

[134] S. Roth and M. J. Black. Steerable random fields. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[135] S. Roth and M.J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, April 2009.

[136] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 23(3):309–314, August 2004.

[137] Daniel L. Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. In *Advances in Neural Information Processing Systems*, pages 551–558, 1993.

[138] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, November 1992.

[139] S. Sadanand and **J. J. Corso**. Action bank: A high-level representation of activity in video. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1234–1241, 2012.

[140] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80(1):72–91, October 2008.

[141] H. S. Sawhney. 3D geometry from planar parallax. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 929–934, 1994.

[142] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[143] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, April 2002.

[144] U. Schmidt, Q. Gao, and S. Roth. A generative perspective on mrfs in low-level vision. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1751–1758, 2010.

[145] T. Schoenemann and D. Cremers. High resolution motion layer decomposition using dual-space graph cuts. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008.

[146] A. Shekhovtsov, I. Kovtun, and V. Hlavac. Efficient MRF deformation model for non-rigid image matching. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[147] J.B. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *IEEE International Conference on Computer Vision*, pages 1154–1160, 1998.

[148] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 22(8):888–905, August 2000.

[149] D. Shulman and J.-Y. Herve. Regularization of discontinuous flow fields. In *Workshop on Visual Motion*, pages 81–86, 1989.

[150] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, volume 2, pages 702–718, 2000.

[151] E P Simoncelli, E H Adelson, and D J Heeger. Probability distributions of optical flow. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 310–315, 1991.

[152] A. Singh. An estimation-theoretic framework for image-flow computation. In *IEEE International Conference on Computer Vision*, pages 168–177, 1990.

[153] A. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *International Journal of Computer Vision*, 82(3):325–357, May 2009.

[154] F. Steinbrucker, T. Pock, and D. Cremers. Large displacement optical flow computation withoutwarping. In *IEEE International Conference on Computer Vision*, pages 1609–1614, 2009.

[155] Liam Stewart, Xuming He, and Richard Zemel. Learning flexible features for conditional random fields. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 30(8):1145–1426, August 2008.

[156] C. Strecha, R. Fransens, and L.J. Van Gool. A probabilistic approach to large displacement optical flow and occlusion detection. In *Workshopt on Statistical Methods in Video Processing*, pages 71–82, 2004.

[157] E. Sudderth and M. Jordan. Shared segmentation of natural scenes using dependent Pitman-Yor processes. In *Advances in Neural Information Processing Systems*, pages 1585–1592, 2009.

[158] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2432–2439, 2010.

[159] D. Sun, S. Roth, J. P. Lewis, and M. J. Black. Learning optical flow. In *European Conference on Computer Vision*, pages 83–97, 2008.

[160] D. Sun, E. B. Sudderth, and M. J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *NIPS*, pages 2226–2234, 2010.

[161] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1768–1775, 2012.

[162] Deqing Sun and Ce Liu. Non-causal temporal prior for video deblocking. In *European Conference on Computer Vision*, 2012.

[163] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2233–2240, 2011.

[164] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[165] N. A. Thacker, A. F. Clark, J. L. Barron, J. Ross B., P. Courtney, W. R. Crum, V. Ramesh, and C. Clark. Performance characterization in computer vision: A guide to best practices. *Computer Vision and Image Understanding*, 109(3):305–334, March 2008.

[166] A. Thayananthan, M. Iwasaki, and R. Cipolla. Principled fusion of high-level model and low-level cues for motion segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[167] W.B. Thompson. Exploiting discontinuities in optical flow. *International Journal of Computer Vision*, 30(3):163–173, Dec. 1998.

[168] W.B. Thompson and S.T. Barnard. Lower-level estimation and interpretation of visual motion. *Computer*, 14:20–28, August 1981.

[169] P.H.S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 23(3):297–303, March 2001.

[170] Daniel Toth, Til Aach, and Volker Metzler. Illumination-invariant change detection. In *4th IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 3–7, 2000.

[171] W. Trobin, T. Pock, D. Cremers, and H. Bischof. An unbiased second-order prior for high-accuracy motion estimation. In *Pattern Recognition (Proceedings of DAGM)*, pages 396–405, 2008.

[172] T. Vaudrey and R. Klette. Residual images remove illumination artifacts! In *Pattern Recognition (Proceedings of DAGM)*, pages 472–481, Berlin, Heidelberg, 2009. Springer-Verlag.

[173] P.A. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.

[174] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. Modeling temporal coherence for optical flow. In *IEEE International Conference on Computer Vision*, pages 1116–1123, 2011.

[175] Martin J. Wainwright and Eero P. Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In *Advances in Neural Information Processing Systems*, pages 855–861, 1999.

[176] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1030–1037, 2010.

[177] C. Wang, M. de La Gorce, and N. Paragios. Segmentation, ordering and multi-object tracking using graphical models. In *IEEE International Conference on Computer Vision*, pages 747–754, 2009.

[178] H. Wang, A. Klaser, C. Schmid, and C.L. Liu. Action recognition by dense trajectories. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3169–3176, 2011.

[179] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.

[180] J.Y.A. Wang and E.H. Adelson. Layered representation for motion analysis. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 361–366, 1993.

[181] Y-S Wang, H-C Lin, O. Sorkine, and T-Y Lee. Motion-based video retargeting with optimized crop-and-warp. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 29(4):90:1–90:9, 2010.

[182] William H. Warren, Bruce A. Kay, Wendy D. Zosh, Andrew P. Duchon, and Stephanie Sahuc. Optic flow is used to control human walking. *Nature:Neuroscience*, 4(2):213–216, February 2001.

[183] A. Wedel, T. Pock, J. Braun, U. Franke, and D. Cremers. Duality TV-L1 flow with fundamental matrix prior. In *Image and Vision Computing New Zealand*, pages 1–6, 2008.

[184] A. Wedel, T. Pock, and D. Cremers. Structure- and motion-adaptive regularization for high accuracy optic flow. In *IEEE International Conference on Computer Vision*, pages 1663–1668, 2009.

[185] A. Wedel, T. Pock, C. Zach, D. Cremers, and H. Bischof. An improved algorithm for TV-L1 optical flow. In *Dagstuhl Motion Workshop*, pages 23–45, 2008.

[186] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 520–526, 1997.

[187] Y. Weiss and E.H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 321–326, 1996.

[188] M. Werlberger, T. Pock, and Horst Bischof. Motion estimation with non-local total variation regularization. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2464–2471, 2010.

[189] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. In *BMVC*, pages 108.1–108.11, 2009.

[190] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, December 1945.

[191] J. Wills, S. Agarwal, and S. Belongie. What went where. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 37–454, 2003.

[192] J. Wills, S. Agarwal, and S.J. Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. *International Journal of Computer Vision*, 68(2):125–143, June 2006.

[193] J. Wills and S.J. Belongie. A feature-based approach for determining dense long range correspondences. In *European Conference on Computer Vision*, volume III, pages 170–182, 2004.

[194] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *PAMI*, 27(10):1644–1659, October 2005.

[195] J.J. Xiao, H. Cheng, H.S. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *European Conference on Computer Vision*, volume I, pages 211–224, 2006.

[196] L Xu, J. Chen, and J. Jia. A segmentation based variational model for accurate optical flow estimation. In *European Conference on Computer Vision*, volume I, pages 671–684, 2008.

[197] L. Xu, J. Jia, and Y.i Matsushita. Motion detail preserving optical flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1293–1300, 2010.

[198] H. Yalcin, M. J. Black, and R. Fablet. The dense estimation of motion and appearance in layers. In *IEEE Workshop on Image and Video Registration*, pages 777–784, 2004.

[199] K.J. Yoon and I.S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 28(4):650–656, April 2006.

[200] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Pattern Recognition (Proceedings of DAGM)*, pages 214–223, 2007.

[201] Shaohua Kevin Zhou, Jie Shao, Bogdan Georgescu, and Dorin Comaniciu. Boostmotion: Boosting a discriminative similarity function for motion estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1761–1768, 2006.

[202] Y. Zhou and H. Tao. Background layer model for object tracking through occlusion. In *IEEE International Conference on Computer Vision*, volume 2, pages 1079–1085, 2003.

[203] S. Zhu, Y. Wu, and D. Mumford. Filters random fields and maximum entropy (FRAME): To a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, March 1998.

[204] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, and H.-P. Seidel. Complementary optic flow. In *Energy Minimization Methods in Computer Vision and Pattern*, pages 207–220, 2009.

[205] C.W. Zitnick, N. Jojic, and Sing Bing Kang. Consistent segmentation for optical flow estimation. In *IEEE International Conference on Computer Vision*, volume 2, pages 1308–1315, 2005.