UNIVERSITY OF CALIFORNIA

Los Angeles

Unsupervised Learning for Object Representations by Watching and Moving

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Yanchao Yang

2019

ABSTRACT OF THE DISSERTATION

Unsupervised Learning for Object Representations by Watching and Moving

by

Yanchao Yang
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2019
Professor Stefano Soatto, Chair

The power of deep neural networks comes mainly from huge labeled datasets. Even though it shines on many computer vision tasks, supervised learning bears little hope to hack into the core of intelligent visual systems. On the other side, unsupervised learning is believed to be the future of AI; however, its performance is always inferior compared to the supervised counterpart. The goal of our research is to develop unsupervised learning algorithms for computer vision tasks while matching or even outperforming the supervised ones. Our key is a representation that is as informative as the supervisory labels, which can be constructed from an unlimited amount of unlabeled data. In theory, this representation contains richer information than the processed supervisory signal. Moreover, we develop algorithms that can utilize existing labeled datasets to expedite the information extraction from the unlimited unlabeled data. Our research is lined up in an order similar to the visual development in early infancy, such that we can also investigate the interplay between different visual functionalities. The final goal is to develop a robotic visual system akin to a human's, that can automatically acquire semantics from concepts of objects fostered by basic perceptions of motion and depth with the minimum amount of human supervision.

The dissertation of Yanchao Yang is approved.

Guy Van den Broeck

Stanley J Osher

Ying Nian Wu

Stefano Soatto, Committee Chair

University of California, Los Angeles

2019

To my parents, Song and Jinhua

And my wife, Chen

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I first want to thank my advisor Stefano Soatto, for his full support in shaping out my Ph.D. by ceaselessly discussing fundamental questions and sharing with me the profound understanding of the world. I also want to thank my thesis committee: Stanley Osher, Ying Nian Wu and Guy Van den Broeck, for their helpful advice on the content discussed here. Ganesh Sundaramoorthi has an important role in introducing me to computer vision in the early stage of my research career. The UCLA Vision Lab provides a vital, dependable and inspiring environment for both academic research and fruitful living. Thus I would like to thank my colleagues and collaborators: Alex Wong, Antonio Loquercio, Xiaohan Fei, Shay Deutsch, Pratik Chaudhari, Alessandro Achille, Xinzhu Bei, Safa Cicek, Peng Zhao, Tong He, Alexandre Tiard, Konstantine Tsotsos, Jingming Dong, Brian Taylor, Sim-lin Lau, and all the others. As usual, a single ”thanks” to my parents represents all that I want to say to them in this lifetime or the next. Note that, all the words here, would not come into incarnation without the advent of my wife, who constantly supports me in every aspect. Also, I want to thank my daughter, who accelerates the generation of this manuscript by a magnitude and should be credited if there is any surprise in the manuscript.

VITA

2007 - 2011 Bachelor of Engineering, Electronic Information Engineering, University of Science and Technology of China, Hefei, China.

2011 - 2013 Master of Science, Electrical Engineering, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia.

2014 - 2018 Research Assistant, Computer Science Department, University of California, Los Angeles, CA.

CHAPTER 1

Introduction

Deep Neural Networks (DNNs) have improved the performance on many computer vision tasks, for example, image classification [HZR16], object detection [RDG16], semantic segmentation [CPK18] and image captioning [DAG15]. However, vision is still far from being solved in at least three aspects. Firstly, neural networks for different vision tasks are usually trained in a supervised manner, with the performance strongly depends on the quality of labeled datasets, and there is no guarantee that the trained networks generalize to unseen data when deployed to real-world applications. Secondly, each task is mostly being solved independently from the other tasks, different datasets for different tasks, and the interplay between tasks becomes hard to capture and understand. Thirdly, how these visual abilities are developed through "unsupervised"[1] experience/exploration in the natural world is still not clear. Certainly, a full understanding of all these aspects is indispensable for the construction of general intelligence of vision.

Before the revival of deep learning, there is already a large amount of work on the mechanisms of visual functionalities in both the field of computer vision and neural science. Most of the algorithms proposed are unsupervised due to the lack of big datasets specifically invented for a task and efficient computational devices. The representations and corresponding algorithms are explicitly designed through our understanding of the underlying principles. They are predictable since each computation is known to accomplish a certain subtask, and explainable as each subtask contributes to the final output in an explicit manner. However, the difficulty lies in the online incorporation (learning) of new principles into the model, which is usually related to the modification of the representations and the algorithms. A

[1] no manual label or very few manual labels are provided.

re-programming is often needed which deviates from the goal to build a machine that can constantly learn from new experiences without too much human intervention.

The focus of this thesis is mainly on unsupervised methods for the visual functionalities sequentially developed in the infant visual system. Because we want to minimize the reliance on manually labeled data as much as possible, such that the learning method proposed can learn from the unlimited amount of unlabeled data or the experience of an autonomous robot, by adjusting the network parameters. Why do we investigate the visual functionalities emerge in early infancy? Because the order of development of these visual functionalities may provide an "optimal"[2] guide map for learning in the sense that if one functionality emerges after the others, then the earlier ones could be utilized by the later, which is always a harder and more complex problem. For example, learning object may help to learn categorization and semantics, again learning motion may help to discover objects. The milestones of the visual development in early infancy are listed in Tab. 1.1. For more details, please refer to [Wat96, Kau95, GYP84, Xu99].

month	visual functionality
1	visual motion perception
3	moving object detection and tracking
5 - 7	depth perception
> 12	object categorization & language

Table 1.1: Milestones of the visual development in early infancy.

Next, we provide a summary of our technical contributions harvested along the way to develop a 6-month-old infant visual system, with the abilities of motion perception, depth perception, and moving object detection all learned in an unsupervised manner. Despite the preference for unsupervised learning, our algorithms do allow incorporation of existing labeled data to regulate the learning process. We then have a brief overview of the organization

[2] by nature

of this manuscript.

1.1 Summary of Contributions

Our first contribution is a framework for multiple objects precise shape tracking. To maintain the quality of the shapes, occlusions between objects and self-occlusions are jointly determined with the dense warp between two consecutive video frames via Sobolev gradient descent. This joint problem we have formulated naturally encompasses coarse-to-fine deformation inference without an explicit regularizer and the associated weighting constant. To partition and group unoccluded regions to various objects, we leverage on the complementarity of motion and appearance cues by introducing a novel data term that encompasses both. We derive an efficient numerical scheme and test it against competing methods on benchmark datasets and obtain state-of-the-art performance.

Next, we describe a system to causally process a video to discover "objects" without initialization or supervision. Objects are defined by generic regularities of the scene, that manifest in the images as simply-connected regions with occluding boundaries that deform smoothly over time. The result is a method for detecting and tracking regions that can be used to prime object labeling or semi-supervised training of object detectors. Such regions can be of interest per se in video analysis, as they produce a video "segmentation" that can be evaluated using benchmark datasets.

Our third contribution is a method to compute optical flow at multiple scales of motion, without resorting to multi-resolution or combinatorial methods. It addresses the key problem of small objects moving fast and resolves the artificial binding between how large an object is and how fast it can move before being diffused away by classical scale-space. Even with no learning, it achieves top performance on the most challenging optical flow benchmark. Moreover, the results are interpretable, and indeed we list the assumptions underlying our method explicitly. The key to our approach is the matching progression from slow to fast, as well as the choice of the interpolation method, or equivalently the prior, to fill in regions where the data allows it. We use several off-the-shelf components, with relatively low sensitivity to parameter tuning.

The fourth contribution is a method that learns rich priors on the set of possible flows that are statistically compatible with an image. Classical computation of optical flow involves generic priors (regularizers) that capture rudimentary statistics of images, but not long-range correlations or semantics. On the other hand, fully supervised methods learn the regularity in the annotated data, without explicit regularization and with the risk of overfitting. Given our supervisedly learned prior, one can easily learn the full map to infer optical flow directly from two or more images, without any need for (additional) supervision. We introduce a novel architecture, called Conditional Prior Network (CPN), and show how to train it to yield a conditional prior. When used in conjunction with a simple optical flow architecture, the CPN beats all variational methods and all unsupervised learning-based ones using the same data term. It performs comparably to fully supervised ones, that however are fine-tuned to a particular dataset. Our method, on the other hand, performs well even when transferred between datasets.

The fifth contribution is a deep learning system to infer the posterior distribution of a dense depth map associated with an image, by exploiting sparse range measurements, for instance from a lidar. While the lidar may provide a depth value for a small percentage of the pixels, we exploit regularities reflected in the training set to complete the map so as to have a probability over depth for each pixel in the image. We exploit the Conditional Prior Network, which allows associating a probability to each depth value given an image, and combine it with a likelihood term that uses the sparse measurements. Optionally we can also exploit the availability of stereo during training, but in any case, only require a single image and a sparse point cloud at run-time. We test our approach on both unsupervised and supervised depth completion using the KITTI benchmark and improve the state-of-the-art in both.

The sixth contribution is an adversarial contextual model for detecting moving objects in videos. A deep neural network is trained to predict the optical flow in a region using information from everywhere else but that region (context), while another network attempts to make such context as uninformative as possible. The result is a model where hypotheses naturally compete with no need for explicit regularization or hyper-parameter tuning. Although our method requires no supervision whatsoever, it outperforms several methods that are pre-trained on large annotated datasets. Our model can be thought of as a generalization of classical variational generative region-based segmentation, but in a way that avoids explicit regularization or solution of partial differential equations at run-time.

The content presented here has also been released through the following articles:

1 . Yanchao Yang, Ganesh Sundaramoorthi, and Stefano Soatto. Self-occlusions and disocclusions in causal video object segmentation. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015 [YSS15b].

2 . Yanchao Yang, Brian Taylor, and Stefano Soatto. Building Object Hypotheses by Generic Detection and Tracking in Video. Under Review, 2016.

3 . Yanchao Yang and Stefano Soatto. S2F: Slow-To-Fast Interpolator Flow. Conference on Computer Vision and Pattern Recognition (CVPR), 2017 [YS17].

4 . Yanchao Yang and Stefano Soatto. Conditional Prior Networks for Optical Flow. European Conference on Computer Vision (ECCV), 2018 [YS18].

5 . Yanchao Yang, Alex Wong and Stefano Soatto. Dense Depth Posterior (DDP) from Single Image and Sparse Range. Conference on Computer Vision and Pattern Recognition (CVPR), 2019 [YWS19].

6 . Yanchao Yang, Antonio Loquercio, Davide Scaramuzza and Stefano Soatto. Unsupervised Moving Object Detection via Contextual Information Separation. Conference on Computer Vision and Pattern Recognition (CVPR), 2019 [YLS19].

1.2 Organization of the Thesis

This thesis is divided into three parts, corresponding to the milestones of the visual development in early infancy.

Chapter 2 to 3 form the first part of the thesis. The focus of this part is motion perception, specifically, optical flow estimation. Chapter 2 describes a method that relies on sparse feature matching and dense interpolation for optical flow. The key idea is to match features extracted at different scales *of motion* in a manner that slow motion is first matched, then fast motion and large deformation. Without resorting to the classical multi-scale technique embodied by image hierarchy, how large an object is and how fast it can move are disentangled. Then an interpolation scheme utilizing the topology of the scene is used to fill in the gaps where matching is not found. Quality of the interpolated flow depends heavily on how well the topology used for interpolation represents the scene depicted by the image. This observation motivates the Conditional Prior Networks (CPN) as described in Chapter 3. Instead of engineering the scene topology with heuristics coming from our understanding of the physical world, we propose to learn it from a dataset where images and corresponding compatible flows are given. When the CPN is trained correctly with a bottleneck, it can represent the scene topology up to a conditional prior, which can be used for unsupervised learning of optical flow estimation from a pair of images in any scenarios.

The second part in Chapter 4 develops a deep learning framework to infer the posterior distribution of a dense depth map corresponds to an image. This posterior in its form is divided into two parts, with one part the likelihood part that represents the fidelity of the dense depth given some sparse depth measurements, and the other part a conditional prior term that measures the compatibility of the dense depth given the current observed image and all past experience. Instead of processing the given image and sparse depth measurements alone, which could not generate better decision or control action than the raw input (Data Processing Inequality), the network trained using our framework is able to harvest side information from the previously seen images and corresponding dense depth maps.

Chapter 5 to 6 form the third part of this thesis with the main focus on general object detection and tracking. Chapter 5 develops a system to discover objects without manual initialization by causally processing a video. It consists of two modules, with one module provides pseudo object measurements utilizing occlusion cues and the other module tracks these pseudo measurements in precise shape for temporal association based on explicit modeling of occlusions and dis-occlusions in a Sobolev framework. In Chapter 6, we ask a more fundamental question on what makes an object an object. If there is an answer, what is the representation and algorithm we should use to individualize the objects out of the scene? Finally, this quest turns out to give us the first *adversarial contextual model* to detect moving objects in images. It is fully unsupervised in terms that it can learn the representation of objects and how to detect moving objects by only watching videos. It captures the desirable features of variational region-based segmentation, but it does not require solving a partial differential equation (PDE) at run-time, nor to pick regularizers. It even outperforms several methods using supervision (ground truth object masks) for training.

We conclude in Chapter 7 with a discussion on future directions to develop general intelligence for robot vision based on the ideas explored in this thesis.

CHAPTER 2

Optical Flow via Matching and Interpolation

Optical flow has been a core concern in Computer Vision for over two decades. It is a building block in many low-level vision tasks, and plays a role in a large number of applications, from autonomous navigation to video post-production, only to mention a few. An overview of recent developments is in [SRB14]. Most existing optical flow algorithms struggle with *small things that move fast*. This phenomenon does not have a dramatic impact on the benchmarks since the problem being with small objects makes it such algorithms are not penalized too harshly. Nevertheless, small objects are important: humans can effortlessly pick out a bee flying at a distance.

In analyzing the root causes for the failure by most algorithms to capture small things moving fast, we honed in on a fundamental problem with classical scale-space, which trades off spatial frequencies (by blurring and down-sampling images) with temporal anti-aliasing (to compute temporal derivatives) as illustrated in Fig. 2.1. This ties the *size* of objects to the *speed* at which they can move before being blurred-away in the multi-resolution pyramid that is routinely used in multi-scale/multi-resolution stages common to most variational optical flow techniques. This multi-scale structure is also common in convolutional neural network architectures, so optical flow schemes based on them are typically subject to similar failure modes.

The case of fast motion has been tackled head-on in many recent works on large displacement optical flow, for instance [BBM09, RWH15b, BTS15, TV15, WRH13, CW13, BYJ14, CJL13] and references therein. Several methods are proposed, mixing sparse matching with interpolation [WB15, RWH15b], a philosophy we adopt. Some have used coarse-to-fine matching that maintains the native resolution [RWH15a, BTS15, HSL, YLS15, SY12], or

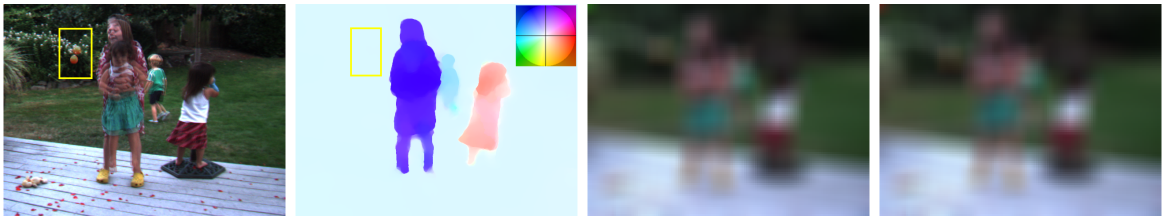


Figure 2.1: Small things moving fast. 1st: two images from the Middlebury dataset (shown superimposed) with the fast-moving ball highlighted, a classic failure mode of multi-resolution optical flow (2nd: the inset *color wheel* shows the map from color to image displacement). Small objects disappear at coarse resolution, where large motions are computed (3rd, 4th), and are never recovered in a differential-based variational scheme.

other multi-scale approaches in a combinatorial setting [TV15, DOR15]. Other samples of relevant related work include [WC11, XDJ12, BDB13, YL15, CSH11]. [XJM12] addresses the problem of lost details in the coarse-to-fine matching by not completely relying on the flow propagated from the upper levels.

However, to the best of our knowledge, none addresses specifically the interplay of size and motion in multi-scale processing, and proposes an iteration that increases the region-of-interest, acting on a decreasing residual domain on the image. In particular, [TV15] addresses matching from small to large displacements, however, it follows the standard scale-space of [BBM09], and focuses on a novel descriptor inspired by sparse coding. Also, [WB15] learns a basis from the computed flow, which however follows a standard approach to scale-space. Both significantly underperform our method on the benchmarks.

In our proposed scheme for multi-scale matching, the scale-space variable is not the amount of diffusion/subsampling of spatial resolution, but instead the size of the interest region on which local matching is based, at the native resolution. Thus, like others have done before, we perform *multi-scale without multi-resolution*. The iteration is instead over the radius of the region-of-interest, whereby regions with larger and larger radii operate on smaller and smaller subsets of the image domains. Slower objects are matched first, and then faster and smaller ones, hence the name S2F.

Clearly, the prior or regularization model plays a key role in optical flow. Rather than delegating it to a dataset and a generic function approximator, we discuss the specific model assumptions made in our method, and the topology with respect to which we consider pixels to be "nearby." In other words, *we hand-engineer the prior,* almost anathema in the age of Deep Learning.

Despite the absence of any learning, our algorithm achieves top performance in the most challenging optical flow benchmark, Sintel. More importantly, we can at least try to *explain* the performance, which we do in Sect. 2.3. Before doing so, we summarize the motivations and the actual algorithm in Sect. 2.1, and describe empirical tests in Sect. 2.2.

2.1 Rationale and Underlying Assumptions

Given two (grayscale) images $I_1, I_2 : D \subset \mathbb{R}^2 \to \mathbb{R}^+$, optical flow is a map $w : \mathbb{R}^2 \to \mathbb{R}^2$ defined at points $x \in D \subset \mathbb{R}^2$ implicitly by $I_1(x) = I_2(w(x)) + n(x)$, where $n(x)$ is an uninformative (white) residual. *Optical flow* is related to *motion field* (the projection of the displacement of points in space when seen in I_1 and I_2 [VP89]) under several assumptions on the scene around the (pre-image) point $X \in \mathbb{R}^3$ of $x \in D$, including: (i) Lambertian reflection and constant illumination, (ii) co-visibility.

When (i) is violated, there is in general no relation between optical flow and motion field. When (ii) is violated (occlusion) there exists *no* transformation w mapping x in image I_1 onto a corresponding point in image I_2. When w exists, it may not be unique, *i.e.*, (iii) flow can be non-identifiable, which happens when the irradiance ("intensity") is not *sufficiently exciting* (*e.g.,* constant). This issue is usually addressed via regularization, by allowing a prior to fill in the flow from sufficiently exciting areas.

A final assumption that is not necessary but common to many algorithms, is (iv) *small displacement* $w(x) \simeq x$. This allows using differential operations (regularized gradient) that facilitate variational optimization. This issue is not present in a combinatorial setting, where any large displacement is allowed, but at a prohibitive computational cost. In the variational setting, the issue is usually addressed via *multi-scale* methods, where temporal anti-aliasing is performed by spatial smoothing, through the creation of *multi-resolution* image pyramids (smoothed and sub-sampled versions of an image [Lin13]), where large displacements at fine-scale correspond to small displacements at coarse-scale.

Small things moving fast

There is a fundamental problem with multi-scale approaches based on classical scale-space, in that it couples *spatial* and *temporal* frequencies. In other words, it ties the *size* of objects to their allowable *speed*. This is manifested in typical failure cases with *small things moving fast* (Fig. 2.1). In general, the size of an object and the speed at which it moves are independent, and they should be treated as such, rather than be coupled for mathematical convenience. How then to address the spatial variability of image velocity?

Multi-scale without multi-resolution

Our approach to avoid the pitfall of multi-resolution, while addressing the intrinsically space-varying scale of motion and respecting the assumptions underlying optical flow computation, is to design a method that is multi-scale but not multi-resolution. It operates at the native resolution, using increasingly large regions-of-interest operating on a decreasing subset of the image domain. Instead of using *spatial blurring* as the scale parameter, it uses *speed*, or magnitude of displacement. This is the key to our method, and explains the name "slow-to-fast". The next section sketches a generic implementation of our algorithm, and subsequent sections detail our choices of components and parameters.

Sketch of S2F-IF

Call $\phi(x; w, I_1, I_2)$ the point-wise cost function used by any baseline optical flow algorithm, for instance $\phi(x; \hat{w}, I_1, I_2) = |I_1(x) - I_2(\hat{w}(x))|$, where we may omit some of the arguments when obvious from the context. Then:

1. Choose an initial radius $r > 0$;

2. Use a *baseline optical flow* algorithm to compute putative forward $\hat{w}$ and backward $\hat{w}^{-1}$ displacements; point-wise residual ρ, where $\hat{w} = \arg\min_w \int_D \phi(x; w, I_1, I_2)dx$, $\rho(x) = \phi(x; \hat{w})$, and $\hat{w}^{-1} = \arg\min_w \int_D \phi(x; w, I_2, I_1)dx$. Also compute forward-backward (f-b) compatibility $b(x) \doteq \|I_{2\times 2} - \hat{w} \circ \hat{w}^{-1}(x)\|$.

   Test violations of (i) and (ii) using the residual $\rho(x)$ and f-b compatibility $b(x)$ respectively, aggregated on a region/window $\mathcal{B}(r)$ with radius r, using a conservative threshold.

   This leaves a (typically sparse) set of points $\mathcal{D} = \{x_i\}_{i=1}^{N(r)}$, and yields their (by assumption, typically small) displacements $w_i = w(x_i)$.

3. *Interpolate* the sparse matches to fill unmatched regions $D \backslash \mathcal{D}$ that violated (i)-(iv), based on a choice of prior/regularizer, leading again to a dense field $\tilde{w}$ and point-wise residual $\tilde{\rho}(x) = \phi(x; \tilde{w})$. Given flow at each point, check f-b compatibility after warping; large residuals are considered occlusions (violations of (ii)).

4. Optionally partition I_1 into piecewise constant regions $\{S_j\}_{j=1}^M$ (*super-pixels*), to facilitate computation, and expand $\mathcal{D}$ to include simply-connected regions with small residual $S_j \cap \chi(\tilde{\rho} < \epsilon_r)$.

5. Mask the matched regions $\mathcal{D}$ from the images, $I_1 \leftarrow I_1 \cdot \chi(D \backslash \mathcal{D})$, and similarly for the warped $I_2 \circ \tilde{w}$, where the dot indicates point-wise multiplication (matched regions are now black).

6. $r \leftarrow r + \delta$, and go to step 2. We use $\delta \geq 1$ pixels, and terminate when r reaches the size of the image, or no more matches could be found.

Several comments are now in order:

- We choose $r = 5, 8$ pixels in (1.) for KITTI and Sintel respectively as in [BTS15]; we use [BTS15] as a baseline optical flow in (2.), and the *census transform* to test compatibility with (i)-(ii). We reject points that fail either the residual ($\epsilon_r = 30$) or the f-b test($\epsilon_c = 1, 5$). We choose [RWH15b] for interpolation in (3.), and [DZ15] for superpixelization. Finally, we use $\delta = 1, 2$ pixels for the scale increment.

- Step 2 implements a conservative *sparse matching* procedure for regions of size r, that leads to a set of sparse matches. Our choice [BTS15] can be replaced by any other conservative sparse matching.

- The matched region $\mathcal{D}$ typically grows monotonically, so the procedure either terminates with a non-empty unmatched set, if no further matches could be found, or each pixel is matched $\mathcal{D} = D$.

- In theory, the process should be terminated before each pixel is matched, as displacement is not defined in occluded region. In practice, all pixels are typically matched, exploiting the regularizer imposed by the interpolation step.

- The first regions of the scene to be matched are the ones that are (i) Lambertian, with (ii) sufficiently exciting radiance, are (iii) co-visible, and (iv) moving slowly. As iterations progress, smaller and smaller regions that are moving faster and faster are matched. For this reason, we call this scheme *Slow-To-Fast* (S2F) Interpolator Flow (IF), as the final solution is influenced heavily by the prior.

- The crucial characteristic of the algorithm above, which is responsible for edging the state-of-the-art, is its lossless multi-scale nature, that is the search at multiple scales of motion, without changing the resolution of the images.

- The algorithm is relatively insensitive to the choice of component algorithms at each step, although the most crucial is the choice of interpolation, which we discussed at in Sect. 2.3.2

2.2 Experiments

2.2.1 Qualitative results

Fig. 2.1 illustrates the key characteristic of our method in comparison to most alternate methods, which we choose to represent with a close-to state-of-the-art baseline [SRB14]. Small objects that move fast are diffused away by scale-space by the time their displacement becomes small enough for a variational optical flow algorithm to resolve. Modifying spatial frequencies (smoothing and down-sampling) to achieve temporal anti-aliasing (to enable approximation of temporal derivatives with first differences) ties the *size* of objects with their *speed*, in ways that are detrimental. Our approach treats them as independent, thus enabling us to capture their motion. It should be mentioned that combinatorial search-base schemes are not subject to this limitation, but suffer from prohibitive computational complexity.

Fig. 2.2 illustrates the various stages of evolution of our algorithm, corresponding to the sketch in Sect. 2.1.

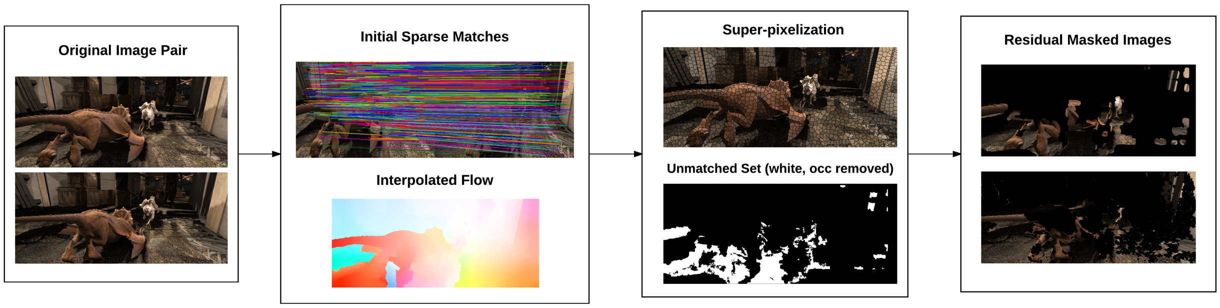


Figure 2.2: Visualization of the stages of our algorithm: Original images (left), initial sparse matches (middle-left, step 2), interpolated flow (middle-left, step 3), super-pixelization (middle-right, step 4), matched set (middle-right, step 5) and residual masked image (right) after the first iteration.

Fig. 2.3 shows the evolution of the matched domain, which typically shrinks monotonically to encompass the entire image domain, with the last, unmatched region filled in by the regularizer.

2.2.2 Benchmark comparisons

Fig. 2.4 shows representative samples for the benchmarks used. The Middlebury dataset [SZS08] comprises 12 pairs of images of mostly static man-made scenes seen under a short baseline. There are few small objects, and none moves fast in the only 8 ground-truthed pairs.

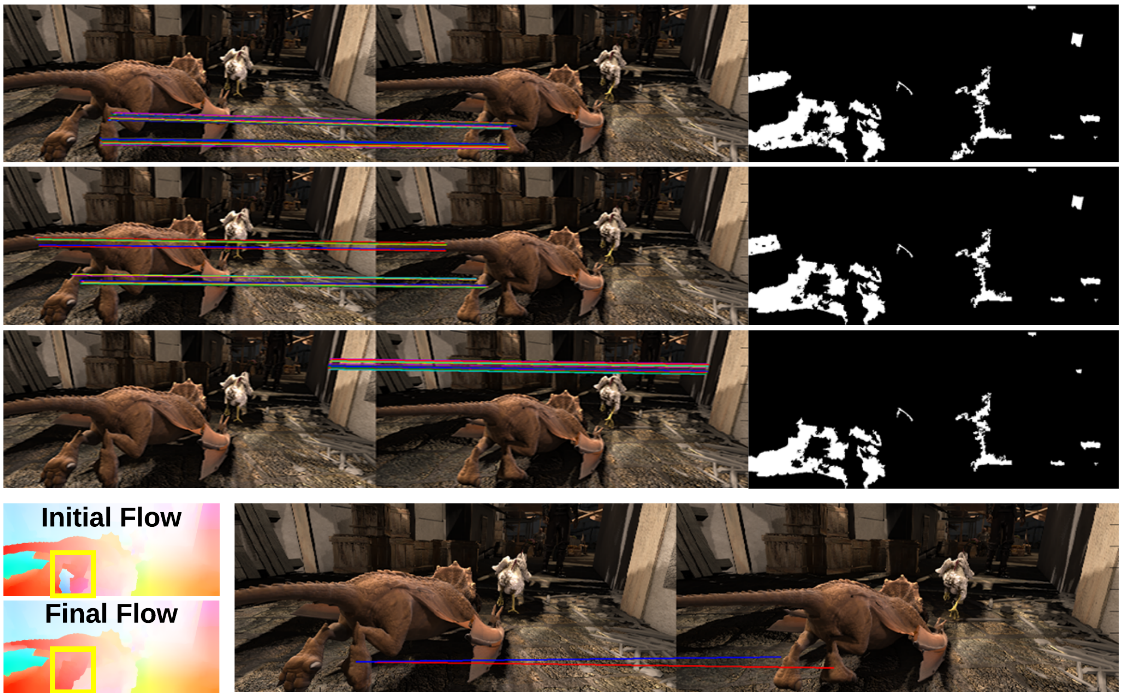


Figure 2.3: Matched regions as the iteration evolves from the first (top row) to the last (bottom rows). The unmatched region (white) shrinks in size, until it converges to regions that are compatible with the hypotheses, but where there is no unique match (third row). On these, the regularizer has license to fill in (bottom), where we highlight details on the legs of the dinosaur, where the overall procedure corrects initial matching errors of the baseline flow algorithm.

The only pairs showing large displacement of small objects are the 4 with no ground truth, including the one shown in Fig. 2.4, which are unfortunately not included in the evaluation. Our algorithm estimates flow more accurately on these sequences. In overall performance, our method ranks in the middle-of-the-pack on this dataset. As a sanity check, we use the Middlebury dataset to compare against the algorithms that report top performance on Sintel, which is a larger dataset showing a wider variety of motions, including large displacement of small objects. The results in Tab. 2.1, show our algorithms comparing favorably. The fact that top performers on Sintel are different from top performers on Middlebury suggests that one of the datasets, or both, are easily overfitted. Middlebury only has 12 image pairs, only 8 of which with ground truth, none of them with large displacement.

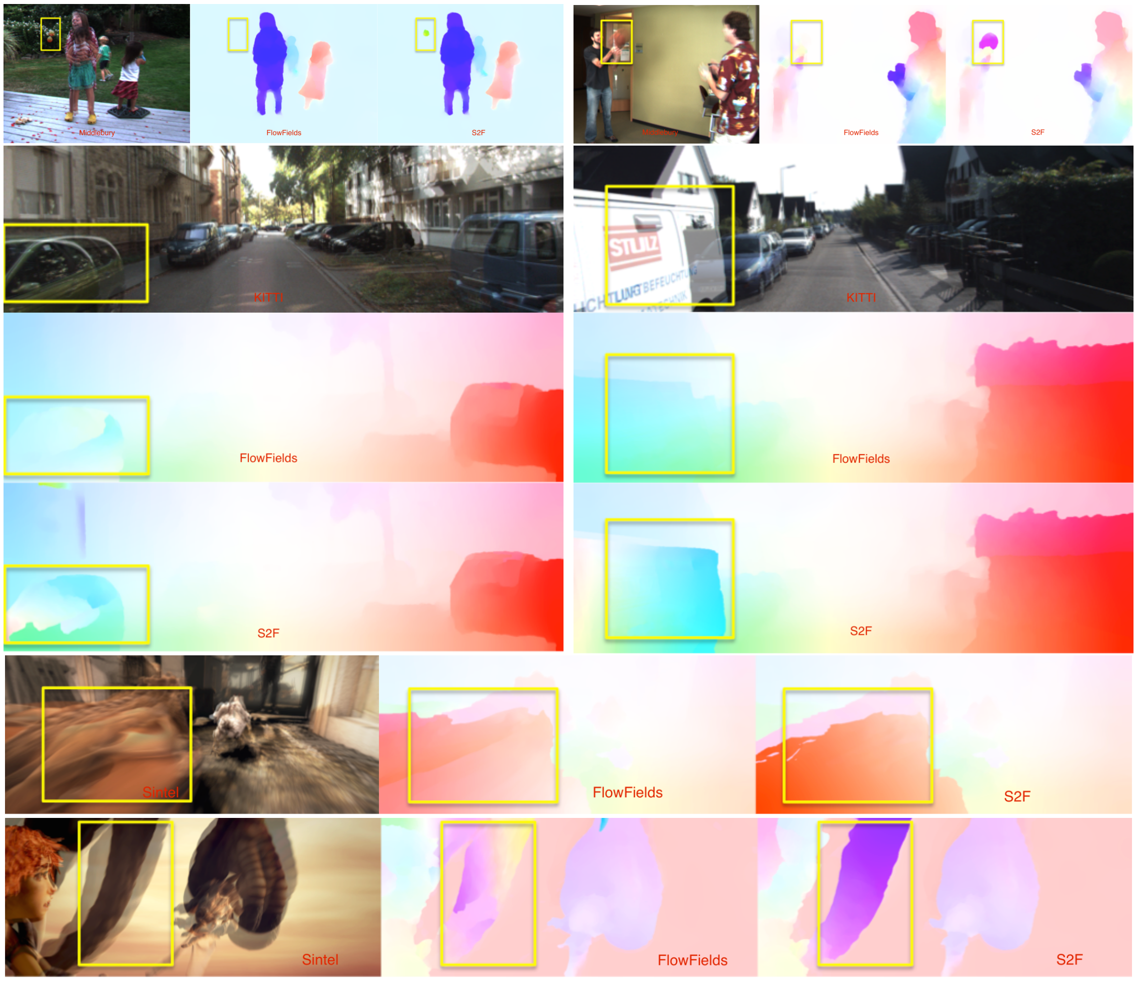


Figure 2.4: Representative samples from various datasets: Middlebury (row 1), KITTI (rows 2, 3, 4), Sintel (rows 5,6). We compare the component flow [BTS15] (FlowFields), with ours (S2F). Details are highlighted in yellow rectangles.

A better benchmark is the KITTI dataset [GLS13], which consists of outdoor driving sequences, with sparse ground truth. Quantitative comparisons with competing algorithms are shown in Tab. 2.2. We use default parameters, not fine-tuned for the dataset, and show competitive performance. As expected, we outperform the baseline flow algorithm we use as a component, shown as the last line on the table as FlowField-. It should be noticed that the same algorithm has been fine-tuned to the KITTI dataset by the authors, shown on the table as FlowFields, with a considerable improvement in performance, suggesting that this dataset can also be overfitted. Since the parameters chosen for the test are not disclosed,

Method	**Avg. Rank**	**Method**	**Avg. Rank**
CPM-Flow [HSL16]	53.7	EpicFlow [RWH15b]	57.4
DeepFlow2 [WRH13]	54.0	FlowNetS [FDI15]	80.4
S2F-IF	**38.6**	FlowFields [BTS15]	41.2

Table 2.1: Average endpoint error on Middlebury for the top-performing algorithms on Sintel. Full ranking can be accessed directly on the Middlebury flow page `http://vision.middlebury.edu/flow/eval/`.

we use the same parameters of the baseline as released, with no fine-tuning for the dataset. We feel that this test is more representative than reporting the best score with different parameters for each dataset.

Method	**Out-Noc**	**Out-All**	**Avg-Noc**	**Avg-All**
CPM-Flow [HSL16]	5.79 %	13.70 %	1.3 px	3.2 px
EpicFlow [RWH15b]	7.88 %	17.08 %	1.5 px	3.8 px
DeepFlow2 [WRH13]	6.61 %	17.35 %	1.4 px	5.3 px
FlowNetS [FDI15]	37.05 %	44.49 %	5.0 px	9.1 px
FlowFields [BTS15]	5.77 %	14.01 %	1.4 px	3.5 px
S2F-IF	6.20 %	15.68 %	1.4 px	3.5 px
FlowField- [BTS15]	6.49 %	15.94 %	1.5 px	3.9 px

Table 2.2: Comparison on the KITTI dataset. Our method uses as a component FlowField- for flow computation. As expected, it improves its performance. The same algorithm, however, fine-tuned to the dataset (indicted as FlowFields, for which no parameters are disclosed) further improves performance. We do not fine-tune ours, and simply report our performance with the same tuning for all datasets. Out-Noc indicates the percentage of pixels with error larger than 3 pixels in non-occluded regions, whereas Out-All indicates the percentage of outliers among all pixels. Avg denotes the average end-point error, again for non-occluded, or all pixels.

Again, we use the same settings as in [BTS15] on the Sintel dataset [BWS12a], which is a synthetic one, but challenging in that it includes fast motion, motion blur, and has precise ground truth. We report the performance in the official benchmark in Tab. 2.3, with our algorithm exhibiting top performance in overall end-point error at the time of writing.

Method	all	mat.	unmat.	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
FlowFields [BTS15]	5.81	2.62	31.79	4.85	2.23	**1.68**	1.15	3.73	33.89
FlowFields+ [BTS15]	5.70	2.68	30.35	**4.69**	**2.11**	1.79	1.13	**3.33**	34.16
SPM-BPv2 [LMB15]	5.81	2.75	30.74	4.73	2.25	1.93	**1.04**	3.46	35.11
FullFlow [CK16]	5.89	2.83	30.79	4.90	2.50	1.91	1.13	3.37	35.59
CPM-Flow [HSL16]	5.96	2.99	30.17	5.03	2.41	2.14	1.15	3.75	35.13
EpicFlow [RWH15b]	6.28	3.06	32.56	5.20	2.61	2.21	1.13	3.72	38.02
DeepFlow2 [WRH13]	6.92	3.09	38.16	5.20	2.81	2.14	1.18	3.85	42.85
S2F-IF	**5.41**	**2.54**	**28.79**	4.74	2.19	1.71	1.15	3.46	**31.26**

Table 2.3: Comparison on the Sintel dataset. all, mat., unmat., respectively stands for end-point error, among all, matched, and unmatched pixels (second through the fourth column). dX-Y stands for error restricted to pixels between X and Y of objects boundaries, thus discounting error at occluded regions. sX-Y stands for pixels with displacements between X and Y pixels. Our method is competitive on all counts, and shines for large displacements, as expected.

These results illustrate the benefit in specifically handling multi-scale phenomena without sacrificing resolution and confusing spatial statistics with temporal ones. More qualitative comparisons can be found in Fig. 2.5.

The next section gives more details on our choice of component methods for the generic algorithm described in Sect. 2.1.

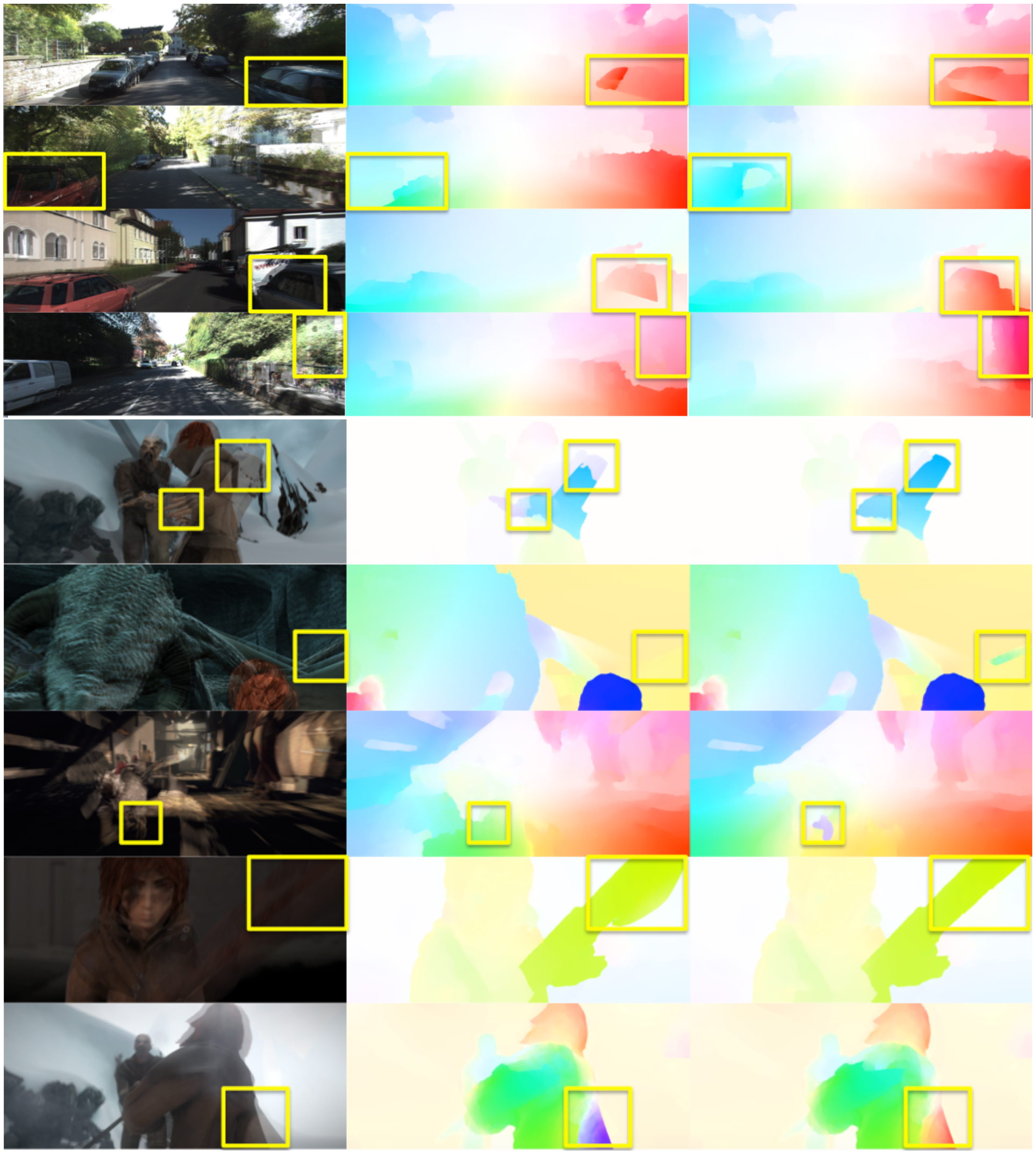


Figure 2.5: More comparisons to [BTS15]. Left to right: overlapped image pairs, results of [BTS15], results of S2F-IF. Details are highlighted using yellow rectangles. KITTI: 1st-4th row; Sintel: 5th-9th row.

2.3 Technical Details

The basic algorithm was described in Sect. 2.1, and consists of sparse matching, followed by interpolation, followed by testing for violation of the hypotheses, where the iteration is with respect to a growing radius for the region of interest, which operates on smaller and smaller residual unmatched portion of the image domain.

2.3.1 Sparse matching

Step 2 of our algorithm results in a sparse set of regions being matched over short displacements. This is not because we actively seek for sparse matches with small displacement. On the contrary, we start with a dense flow, specifically [BTS15], but then conservatively reject all regions that fail hypotheses (i)-(ii) based on residual or f-b compatibility. This naturally results in a sparse set, because sufficient excitation conditions (which are tested through f-b compatibility) require large gradients in two independent directions, which is typically only satisfied on a sparse subset of the image domain. Conceptually, any other sparse matching would do, and the algorithm is not very sensitive to the choice of method for this step, which we therefore do not further discuss.

2.3.2 Interpolation

The algorithm is sensitive to the choice of prior, which in our case corresponds to the choice of the interpolation algorithm. To describe and motivate our choice, let $x, y \in D \subset \mathbb{R}^2$ be two points on the pixel lattice, with distance $d(x, y)$ for some choice of norm. We are interested in inferring the value of the displacement $w(x)$ at x from observations performed at y. We assume a parametric form for the likelihood function

$$p_\theta(w(x)|y) = \mathcal{N}\left(Ax + b; \Sigma(x, y)\right). \tag{2.1}$$

whereby the displacement w at x is a Gaussian random vector having as mean an affine deformation, depends on y, of the point x, with an uncertainty

$$\Sigma(x, y) = \beta^2 \exp\left(d(x, y)\right) I_{2\times 2} \tag{2.2}$$

that grows exponentially with the distance of the observation point. The parameters $\theta = \{A, b\}$ can be inferred via maximum-likelihood, given a sample $\mathcal{D} = \{x_i, w_i\}_{i=1}^N$, where

$w_i = w(x_i)$, as

$$\begin{aligned} \hat{A}, \hat{b} &= \arg\max_{\theta} \prod_{i=1}^{N} p_\theta(w_i|x) \\ &= \arg\max_{A,b} \prod_{i=1}^{N} \mathcal{N}(Ax_i + b; \Sigma(x_i, x)) \\ &= \arg\min_{A,b} \sum_{i=1}^{N} \frac{\|w_i - Ax_i - b\|_2^2}{\beta^2 \exp\left(d(x_i, x)\right)} \end{aligned} \tag{2.3}$$

leaving β as a tuning parameter. This is essentially the locally-weighted (LA) estimator in Eq. (2) of [RWH15b]. Note that $p_\theta(w(x)|x) = \mathcal{N}(Ax + b; \beta^2 I_{2\times2})$ and the parameters θ (which are the sufficient statistics of the dataset $\mathcal{D}$ for the displacement $w(x)$) are a function of the location x. We make this explicit by writing $\theta = \{A(x), b(x)\}$. A point-estimate, for instance the conditional mean, of the displacement can be obtained at each point x,

$$w(x) = A(x)x + b(x). \tag{2.4}$$

This approach follows [RWH15b] to avoid solving a variational optimization problem with explicit regularization, which is instead implicit in the finite-dimensional class of transformations (affine) and the finite data sample $\mathcal{D}$. The behavior of this interpolation method hinges critically on the choice of distance d in (2.3), which we describe next.

2.3.3 Topology

The distance between two points $d(x, y)$ can be based on the topology of the image domain, for instance $d_2(x, y) = \|x - y\|_2$, where nearby pixels are considered close, or the topology of the image range, for instance $d_I(x, y) = \|I(x) - I(y)\|$, where pixels with similar intensity are considered close. Ideally, we would like to use the topology of the *scene*, and consider points $x, y \in D$ close if the distance between their pre-images (back-projection) onto the scene $X, Y \in \mathbb{R}^3$ is close. This would be a geodesic distance, assuming the scene to be multiply-connected and piecewise smooth, infinite if X, Y are on different connected components.

Since we do not have a model of the scene, we use a proxy, whereby the distance between two points on the same connected component X, Y is the distance between their projections $x = \pi(X), y = \pi(Y)$ on the image, whereas the distance between points on different connected components adds a term proportional to their depth differential relative to the distance from the camera.

While we do not know their depth, disconnected components result in occlusion regions with area proportional to the relative depth differential, where the optical flow residual $\phi(x) = \min_w \|I_1(x) - I_2(w(x))\|$ is generally large. Therefore, we can take the path-integral of optical flow residual as a proxy of the geodesic distance:

$$d_w(x, y) \doteq \min_{\gamma} \int_{\gamma_{x \to y}} \phi(z) dz \tag{2.5}$$

where $\gamma_{x \to y}$ is any path from x to y.

We can also assume that objects are smoothly colored, and therefore large intensity changes can be attributed to points being on different objects. Clearly this is not always the case, as smooth objects can have sharp material transitions, but nevertheless one can restrict the topology to simply connected components of the piecewise smooth albedo, and define d_I as

$$d_I(x, y) \doteq \min_{\gamma} \int_{\gamma_{x \to y}} |\nabla I(z)| dz \tag{2.6}$$

and similarly bypass the minimization by using a cordal distance. Various product distances, and various approximations to the geodesic, can be derived, for instance those in [RWH15b]. We use (2.6) in our algorithm.

2.3.4 Hypotheses (i)-(iv) testing

The key to our algorithm is the multi-scale iteration, starting from large regions that move slowly, eventually matching small regions that move fast. At each iteration, hypotheses of (i) Lambertian reflection and constant illumination, and (ii) co-visibility (large residual) are tested conservatively relative to a fixed radius of the region of interest. Furthermore, backward-forward compatibility tests (iii) sufficient excitation; where failed, the regularizer (which in our case is implicit in the interpolation scheme) has license to take over.

While it would be desirable to have an integrated Bayesian framework where the thresholds are automatically determined by competing hypotheses, in practice these stages boil down to threshold selection. Importantly, the algorithm is not extremely sensitive to the choice of thresholds.

2.3.5 Computational cost

The computational cost of our algorithm is essentially dictated by the choice of components. Run-time depends on the complexity of the motion, since the length of our iteration is data-dependent. On average, it takes about 1m per pair of frames in Sintel, where images are of size 1024×436, on a commodity 4-core 3.1GHz desktop. We have observed convergence in as little as 20s, and as long as 2m. This includes all component elements of our pipeline. On smaller images, for instance, Middlebury's, (300×400), our algorithm runs in about 15s/pair of frames. On KITTI, that has 400×1234 pixels per image, our algorithm runs, on average, at 1.5m per pair of frames.

CHAPTER 3

Learning Conditional Prior for Optical Flow

As seen in the previous chapter, the computation of optical flow always involves generic priors (regularizers) that capture rudimentary statistics of images, but not long-range correlations or semantics. Generic priors for regularizing optical flow have been used for decades, starting with Horn & Schunk's ℓ^2 norm of the gradient, to ℓ^1, Total Variation, etc.

Consider Fig. 3.1: A given image (left) could give rise to many different optical flows (OF) depending on what another image of the same scene looks like: It could show a car moving to the right (top), or the same apparently moving to the left due to camera motion to the right (middle), or it could be an artificial motion because the scene was a picture portraying the car, rather than the actual physical scene. A single image biases, but does not constrain, the set of possible flows the underlying scene can generate. We wish to leverage the information an image contains about possible compatible flows to learn better priors than those implied by generic regularizers. Note that all three flows in Fig. 3.1 are equally valid under a generic prior (piecewise smoothness), but not under a natural prior (cars moving in the scene).

A regularizer is a criterion that, when added to a data fitting term, constrains the solution of an inverse problem. These two criteria (data term and regularizer) are usually formalized as an energy function, which is minimized to, ideally, find a unique global optimum.[1] In variational OF, the regularizer captures very rudimentary low-order statistics [BSL11, BWS05, PBB06, BA93, XJM12], for instance the high kurtosis of the gradient distribution. This does not help with the scenario in Fig. 3.1. There has been a recent surge of (supervised) learning-based approaches to OF [DFI15, IMS17, RB17], that do not have

[1] We use the terms regularizer, prior, model, or assumption, interchangeably and broadly to include any restriction on the solution space, or bias on the solution, imposed without full knowledge of the data. In OF, the full data is (at least) two images.

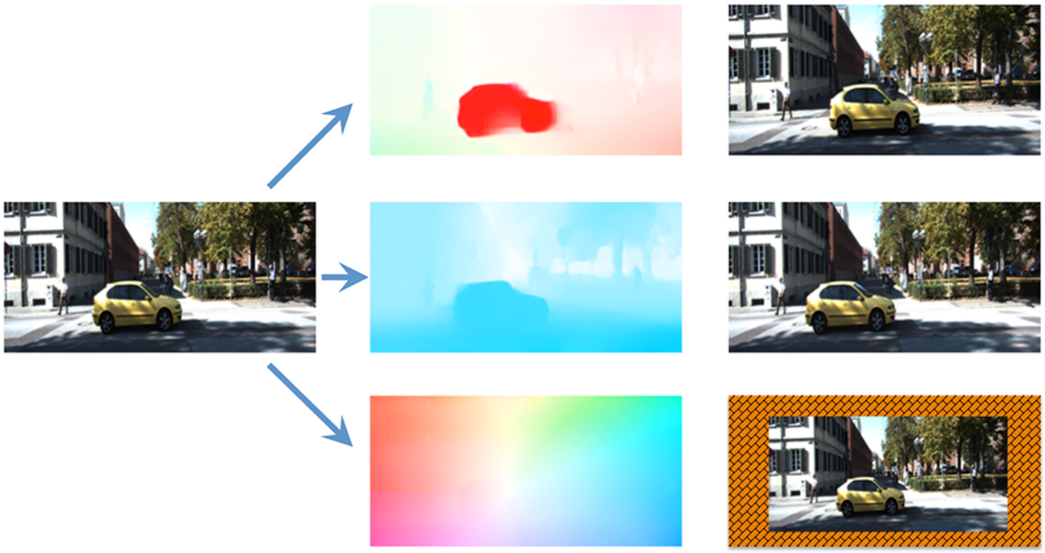


Figure 3.1: A single image biases, but does not constrain, the set of optical flows that can be generated from it, depending on whether the camera was static but objects were moving (top), or the camera was moving (center), or the scene was flat (bottom) and moving on a plane in an un-natural scenario. Flow fields here are generated by our CPNFlow.

explicit regularization nor do they use geometric reprojection error as a criterion for data fit. Instead, a map is learned from pairs of images to flows, where regularization is implicit in the function class [CS16],[2] in the training procedure [CS17] (e.g. noise of stochastic gradient descent – SGD), and in the datasets used for training (e.g. Sintel [BWS12b], Flying Chair [DFI15]).

Our method does not attempt to learn geometric optics anew, even though black-box approaches are the top performers in several benchmarks. Instead, we seek to learn richer priors on the set of possible flows that are statistically compatible with an image (Fig. 3.1).

Supervised learning methods typically rely on synthesized datasets, due to the extreme difficulty in obtaining ground truth flows for realistic videos. Recently, unsupervised optical flow learning methods have flourished, making use of a vast amount of unlabeled videos. Although unsupervised optical flow learning methods are able to learn from an unlimited

[2] In theory, deep neural networks are universal approximants, but there is a considerable amount of engineering in the architectures to capture suitable inductive biases.

amount of data, when compared to variational methods, their performance usually falls behind. Unsupervised learning-based approaches use the same or similar loss functions as variational methods [JHD16, RYN17, MHR18, AP16], including priors, but restrict the function class to a parametric model, for instance convolutional neural networks (CNNs) trained with SGD, thus adding implicit regularization [CS17], which is minute when explicit regularizer is applied. Again, the priors only encode first-order statistics, which fail to capture the phenomena in Fig. 3.1.

We advocate learning a conditional prior, or regularizer, from data, but do so once and for all, and then use it in conjunction with any data fitting term, with any model and optimization one wishes.

What we learn is a prior in the sense that it imposes a bias on the possible solutions, but it does not alone constraint them, which happens only in conjunction with a data term. Once the prior is learned, in a supervised fashion, one can also learn the full map to infer optical flow directly from data, without any need for (additional) supervision. In this sense, our method is *"semi-unsupervised"*: Once *we* learn the prior, *anyone* can train an optical flow architecture entirely unsupervised. The key idea here is to learn a prior for the set of optical flows that are statistically compatible with *a single image*. Once done, we train a relatively simple network *in an unsupervised fashion* to map *pairs of images* to optical flows, where the loss function used for training includes explicit regularization in the form of the conditional prior, added to the reprojection error.

Despite a relatively simple architecture and low computational complexity, our method beats all variational ones and all unsupervised learning-based ones. It is on par or slightly below a few fully supervised ones, that however are fine-tuned to a particular dataset, and are extremely onerous to train. More importantly, available fully supervised methods perform best *on the dataset on which they are trained.* Our method, on the other hand, performs well even when the prior is trained on one dataset and used on a different one. For instance, a fully-supervised method trained on Flying Chair beats our method on Flying Chair, but underperforms it on KITTI and vice-versa (Tab. 3.1). Ours is consistently among the top in all datasets. More importantly, our method is complementary, and can be used in conjunction with more sophisticated networks and data terms.

Formalization

Let $I_1, I_2 \in \mathbb{R}_+^{H\times W\times 3}$ be two consecutive images and $f : \mathbb{R}^2 \to \mathbb{R}^2$ the flow, implicitly defined in the co-visible region by $I_1 = I_2 \circ f + n$ where $n \sim P_n$ is some distribution. The posterior $P(f|I_1, I_2) \propto P_n(I_1 - I_2 \circ f)$ can be decomposed as

$$\begin{aligned}\log P(f|I_1, I_2) &= \log P(I_2|I_1, f) + \log P(f|I_1) - \log P(I_2|I_1)\\ &\approx \log P(I_2|I_1, f) + \log P(f|I_1) \qquad (3.1)\end{aligned}$$

We call the first term (data) **prediction error**, and the second **conditional prior**. It is a prior in the sense that, given I_1 alone, many flows can have high likelihood for a suitable I_2. However, it is informed by I_1 in the sense of capturing image-dependent regularities such as flow discontinuities often occurring at *object boundaries*, which may or may not correspond to generic image discontinuities. A special case of this model assumes a Gaussian likelihood (ℓ^2 prediction error) and an ad-hoc prior of the form

$$E(f, I_1, I_2) = \int (I_1(x) - I_2(x + f(x)))^2 dx + \int \alpha(x, I_1)\|\nabla f(x)\|^2 dx \qquad (3.2)$$

where α is a scalar function that incorporates our belief in an irradiance boundary of I_1 corresponding to an object boundary.[3] Image-dependent priors as in Eq. (3.2) include [KK12, RBP14, CK16, DKA95, PVP94, BBP04, XJM12]. This type of conditional prior has several limitations: First, in the absence of *semantic context*, it is not possible to differentiate occluding boundaries (where f can be discontinuous) from material boundaries (irradiance discontinuities), or illumination boundaries (cast shadows) where f is smooth. Second, the image I_1 only informs the flow *locally*, through its gradient, and does not capture global regularities. Fig. 3.2 shows that flow fails to propagate into homogeneous region. This can be mitigated by using a fully connected CRF [SM12] but at a heavy computational cost.

[3] When α is constant, we get an even more special case, the original Horn & Schunk model where the prior is also Gaussian and unconditional (independent of I_1).

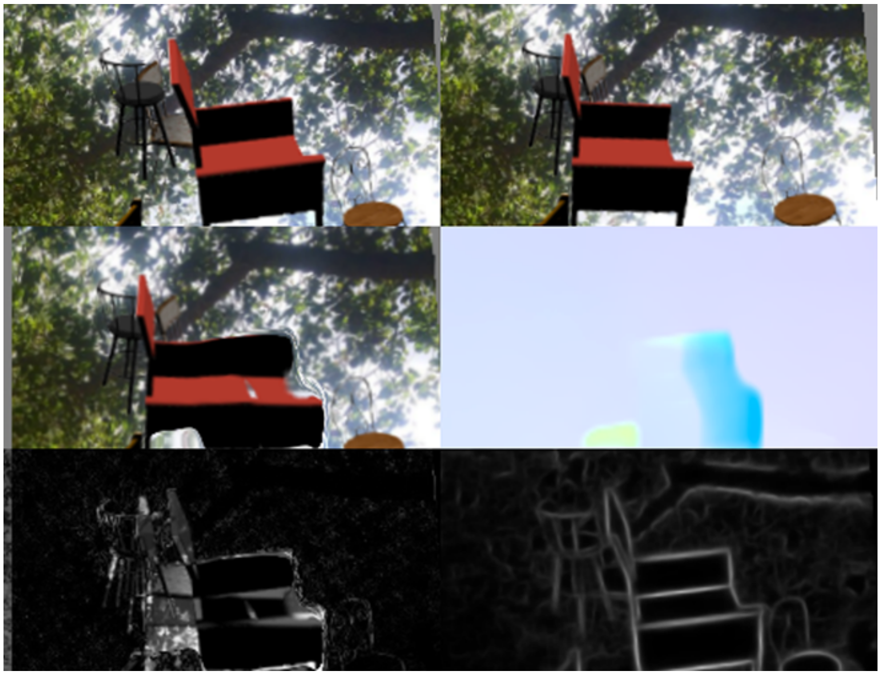


Figure 3.2: First row: two images I_1, I_2 from the Flying Chair dataset; Second row: warped image $I_2 \circ \hat{f}$ (left) using the flow (right) estimated by minimizing Eq. (3.2); Third row: residual $n = \|I_1 - I_2 \circ f\|$ (left) compared to the edge strength of I_1 (right). Note the flow estimated at the right side of the chair fails to propagate into the homogeneous region where the image gradient is close to zero.

Our goal can be formalized as *learning the conditional prior* $P(f|I_1)$ in a manner that exploits the semantic context of the scene[4] and captures the global statistics of I_1. We will do so by leveraging the power of deep convolutional neural networks trained end-to-end, to enable which we need to design differentiable models, which we do next.

3.1 Method

To learn a conditional prior we need to specify the inference criterion (loss function), which we do in Sect. 3.1.2 and the class of functions (architecture), with respect to which the loss

[4] The word "semantic" is often used to refer to *identities* and *relations* among discrete entities (objects). What matters in our case is the *geometric and topological relations* that may result in occluding boundaries on the image plane. The name of an object does not matter to that end, so we ignore identities and do not require object labels.

is minimized end-to-end. We introduce our choice of architecture next, and the optimization in Sect. 3.1.4.

3.1.1 Conditional Prior Network (CPN)

We construct the conditional prior from a modified autoencoder trained to reconstruct a flow f that is compatible with the given (single) image I. We call this a Conditional Prior Network (CPN) shown in Fig. 3.3.

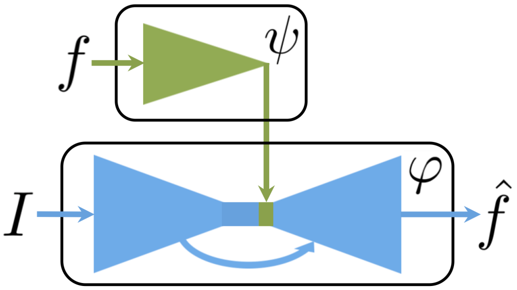


Figure 3.3: Conditional Prior Network (CPN) architecture for learning $P(f|I)$: ψ is an encoder of the flow f, and φ is a decoder that has full access to the image I.

In a CPN, ψ encodes only the flow f, then φ takes the image I and the output of ψ to generate a reconstruction of f, $\hat{f} = \varphi(I, \psi(f))$. Both ψ and φ are realized by pure convolutional layers with subsampling (striding) by two to create a bottleneck. Note that φ is a U-shape net [DFI15] with skip connections, at whose center a concatenation with $\psi(f)$ is applied. Before we articulate the reasons for our choice of architecture, and argue that it is better than an ordinary autoencoder that encodes both f and I in one branch, we introduce the choice of loss function and how it is trained next.

3.1.2 Loss function

We are given a dataset D sampled from the joint distribution $D = \{(f_j, I_j)\}_{j=1}^n \sim P(f, I)$, with n samples. We propose approximating $P(f|I)$ with a CPN as follows

$$Q_{w_\varphi, w_\psi}(f|I) = \exp\left(-\|\varphi(I, \psi(f)) - f\|^2\right) \propto P(f|I) \tag{3.3}$$

where w_φ, w_ψ are the parameters of φ and ψ respectively. Given I, for every flow f, the above returns a positive value whose log, after training, is equal to the negative squared autoencoding loss. To determine the parameters that yield an approximation of $P(f|I)$, we should solve the following optimization problem

$$w^*_\varphi, w^*_\psi = \arg\min_{w_\varphi, w_\psi} \mathbb{E}_{I\sim P(I)} \mathbb{KL}(P(f|I) \| Q_{w_\varphi,w_\psi}(f|I)) \tag{3.4}$$

where the expectation is with respect to all possible images I, and $\mathbb{KL}$ is the Kullback-Leibler divergence between $P(f|I)$ and the CPN $Q_{w_\varphi,w_\psi}(f|I)$. We show that the above is equivalent to:

$$\begin{aligned}
w^*_\varphi, w^*_\psi &= \arg\min_{w_\varphi, w_\psi} \mathbb{E}_{I\sim P(I)} KL(P(f|I) \| Q_{w_\varphi,w_\psi}(f|I)) \\
&= \arg\min_{w_\varphi, w_\psi} \int_I P(I) KL(P(f|I) \| Q_{w_\varphi,w_\psi}(f|I)) \\
&= \arg\min_{w_\varphi, w_\psi} \int_I P(I) \int_f (P(f|I) log \frac{P(f|I)}{Q_{w_\varphi,w_\psi}(f|I)}) df dI \\
&= \arg\min_{w_\varphi, w_\psi} \int_I \int_f P(I)P(f|I) log P(f|I) df dI - \int_I \int_f P(I)P(f|I) log Q_{w_\varphi,w_\psi}(f|I) df dI \\
&= \arg\max_{w_\varphi, w_\psi} \int_I \int_f P(f,I) log Q_{w_\varphi,w_\psi}(f|I) df dI \\
&= \arg\max_{w_\varphi, w_\psi} - \int_I \int_f P(f,I) \|\varphi_{w_\varphi}(I, \psi_{w_\psi}(f)) - f\|^2 df dI \\
&= \arg\min_{w_\varphi, w_\psi} \int_I \int_f P(f,I) \|\varphi_{w_\varphi}(I, \psi_{w_\psi}(f)) - f\|^2 df dI
\end{aligned} \tag{3.5}$$

which is equivalent to minimizing the empirical autoencoding loss since the ground truth flow is quantized, $\sum_{j=1}^n \|\hat{f}_j - f_j\|^2$. If the encoder had no bottleneck (sufficient information capacity), it could overfit by returning $\hat{f} = \varphi_{w_\varphi}(I, \psi_{w_\psi}(f)) = f$, rendering the conditional prior $Q_{w_\varphi,w_\psi}(f|I)$ uninformative (constant). Thus we introduce an information regularizer (bottleneck) on the encoder ψ leading to the **CPN training loss**

$$w^*_\varphi, w^*_\psi = \arg\min_{w_\varphi, w_\psi} \mathbb{E}_{I\sim P(I)} \mathbb{KL}(P(f|I) \| Q_{w_\varphi,w_\psi}(f|I)) + \beta \boldsymbol{I}(f, \psi_{w_\psi}(f)) \tag{3.6}$$

where $\beta > 0$ modulates complexity (information capacity) and fidelity (data fit), and $\boldsymbol{I}(f, \psi_{w_\psi}(f))$ is the mutual information between the flow f and its representation (code)

$\psi_{w_\psi}(f)$. When β is large, the encoder is lossy, thus preventing $Q_{w_\varphi,w_\psi}(f|I)$ from being uninformative.[5]

3.1.3 Reasoning behind the CPN structure

Now we show our reasoning that leads us to the current CPN structure instead of the ordinary autoencoder which encodes both f and I in one branch as follows:

$$Q_{w_\varphi,w_\psi}(f|I) = \exp\left(-\|\varphi \circ \psi(f,I) - f\|^2\right) \tag{3.7}$$

where φ is a decoder and ψ is an encoder of both f and I, parameterized by w_φ, w_ψ respectively. The optimal parameters w_φ^*, w_ψ^* should be obtained by minimizing the average KL divergence between the proposed conditional Q and $P(f|I)$:

$$w_\varphi^*, w_\psi^* = \arg\min_{w_\varphi,w_\psi} E_{I\sim P(I)} KL(P(f|I)\|Q_{w_\varphi,w_\psi}(f|I)) \tag{3.8}$$

similarly to former subsection, we can show that the above optimization problem is equivalent to:

$$\begin{aligned} w_\varphi^*, w_\psi^* &= \arg\max_{w_\varphi,w_\psi} \int_I \int_f P(f,I) \log[Q_{w_\varphi,w_\psi}(f|I)] df dI \\ &= \arg\min_{w_\varphi,w_\psi} \int_I \int_f P(f,I) \|\varphi_{w_\varphi} \circ \psi_{w_\psi}(f,I) - f\|^2 df dI \end{aligned} \tag{3.9}$$

However, we are not done as ψ is an encoder with limited capacity, thus ψ is not one-to-one, which makes the following subset non-empty:

$$\boldsymbol{I}_{\psi,f} = \{I|\psi_{w_\psi}(f,I) = \psi\}. \tag{3.10}$$

We can rewrite the optimization problem Eq. (3.9) as:

$$\begin{aligned} w_\varphi^*, w_\psi^* &= \arg\min_{w_\varphi,w_\psi} \int_f \int_\psi \int_{I\in\boldsymbol{I}_{\psi,f}} P(f,I)\|\varphi_{w_\varphi} \circ \psi_{w_\psi}(f,I) - f\|^2 dI d\psi df \\ &= \arg\min_{w_\varphi,w_\psi} \int_f \int_\psi \|\varphi_{w_\varphi}(\psi) - f\|^2 \left(\int_{I\in\boldsymbol{I}_{\psi,f}} P(f,I) dI\right) d\psi df \\ &= \arg\min_{w_\varphi,w_\psi} \int_f \int_\psi \|\varphi_{w_\varphi}(\psi) - f\|^2 P(f) P_{w_\psi}(\psi|f) d\psi df \end{aligned} \tag{3.11}$$

[5] the decoder φ imposes no architectural bottleneck due to skip connections.

$P_{w_\psi}(\psi|f)$ is a probability measure induced by the encoder ψ. Thus, the original optimization problem is essentially minimizing the following quantity:

$$w_\varphi^*, w_\psi^* = KL\left(P(f)P_{w_\psi}(\psi|f) \| Q_{w_\varphi}(\psi, f)\right) \tag{3.12}$$

During the optimization process, the encoder is trying to push $P(f)P_{w_\psi}(\psi|f)$ towards $Q_{w_\varphi}(\psi, f)$ and the decoder is pushing from the other side. After optimization:

$$Q_{w_\varphi, w_\psi}(f|I) = \exp\left(-\|\varphi \circ \psi(f, I) - f\|^2\right) \propto P(f)P_{w_\psi}(\psi(f, I)|f) \tag{3.13}$$

which is not $P(f|I)$ nor $P(f, I)$! In order to let $Q_{w_\varphi, w_\psi}(f|I)$ approximate $P(f|I)$, the condition $P_{w_\psi}(\psi|f) = P(I|f)$ should be true. And this is satisfied when ψ imposes no compression on I. i.e. $\psi : (f, I) \to (\psi(f), I)$, which enforces Eq. (3.10) to be a singleton, and now we have the proposed CPN structure.

3.1.4 Training a CPN

While the first term in Eq. (3.6) can simply be the empirical autoencoding loss, the second term can be realized in many ways, *e.g.*, an ℓ^2 or ℓ^1 penalty on the parameters w_ψ. Here we directly increase the bottleneck β by decreasing the coding length ℓ_ψ of ψ. Hence the training procedure of the proposed CPN can be summarized as follows:

1. Initialize the coding length of the encoder ℓ_ψ with a large number ($\beta = 0$).
2. Train the encoder-decoder ψ, φ jointly by minimizing $e = \dfrac{1}{n}\sum_{j=1}^{n} \|\hat{f}_j - f_j\|^2$ until convergence. The error at convergence is denoted as e^*.
3. If $e^* > \lambda$, training done.[6]
   Otherwise, decrease ℓ_ψ, (increase β), and goto step 2.

It would be time consuming to train for every single coding length ℓ_ψ. We only iteratively train for the integer powers, $2^k, k \leq 10$.

[6] in our experiments, $\lambda = 0.5$.

Inference: suppose the optimal parameters obtained from the training procedure are w_{ψ}^{*}, w_{φ}^{*}, then for any given pair (f, I), we can use $Q_{w_{\varphi}^{*}, w_{\psi}^{*}}(f|I)$ as the conditional prior up to a constant. In the next section we add a data discrepancy term to the (log) prior to obtain an energy functional for learning direct mapping from images to optical flows.

3.1.5 Semi-unsupervised learning optical flow

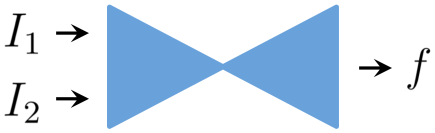


Figure 3.4: FlowNet architecture for learning the mapping from I_1, I_2 to optical flow f.

Unlike a generative model such as a variational autoencoder [KW13], where sampling is required in order to evaluate the probability of a given observation, here (f, I) is directly mapped to a scalar using Eq. (3.3), thus differentiable w.r.t f, and suitable for training a new network as shown in Fig. 3.4 to predict optical flow given images I_1, I_2, by minimizing the following compound loss:

$$\begin{aligned} E(f|I_1, I_2) &= \int_{\Omega \backslash O} \rho(I_1(x) - I_2(x + f(x)))dx - \alpha \log[Q_{w_{\varphi}^{*}, w_{\psi}^{*}}(f|I_1)] \\ &= \int_{\Omega \backslash O} \rho(I_1(x) - I_2(x + f(x)))dx + \alpha \|\varphi^{*}(I_1, \psi^{*}(f)) - f\|^2 \quad (3.14) \end{aligned}$$

with $\alpha > 0$, $Q_{w_{\varphi}^{*}, w_{\psi}^{*}}$ our learned conditional prior, and $\rho(x) = (x^2 + 0.001^2)^{\eta}$ the generalized Charbonnier penalty function [BW05]. Note that the integration in the data term is on the co-visible area, i.e. the image domain Ω minus the occluded area O, which can be set to empty for simplicity or modeled using the forward-backward consistency as done in [MHR18] with a penalty on O to prevent trivial solutions. In the following section, we describe our implementation and report results and comparisons on several benchmarks.

3.2 Experiments

3.2.1 Network details

CPN: we adapt the FlowNetS network structure proposed in [DFI15] to be the decoder φ, and the contraction part of FlowNetS to be the encoder ψ in our CPN respectively. Both parts are shrunk versions of the original FlowNetS with a factor of 1/4; altogether our CPN has 2.8M parameters, which is an order of magnitude less than the 38M parameters in FlowNetS. As we mentioned before, the bottleneck in Eq. (3.6) is controlled by the coding length ℓ_ψ of the encoder ψ, here we make the definition of ℓ_ψ explicit, which is the number of the convolutional kernels in the last layer of the encoder. In our experiments, $\ell_\psi = 128$ always satisfies the stopping criterion described in Sect. 3.1.4, which ends up with a reduction rate of 0.015 in the dimension of the flow f.

CPNFlow: we term our flow prediction network CPNFlow. The network used on all benchmarks for comparison is the original FlowNetS with no modifications, letting us focus on the effects of different loss terms. The total number of parameters is 38M. FlowNetS is the most basic network structure for learning optical flow [DFI15], *i.e.*, only convolutional layers with striding for dimension reduction, however, when trained with loss Eq. (3.14) that contains the learned conditional prior (CPN), it achieves better performance than the more complex network structure FlowNetC [DFI15], or even stack of FlowNetS and FlowNetC. Please refer to Sect. 3.2.4 for details and quantitative comparisons.

3.2.2 Datasets for training

Flying Chairs is a synthesized dataset proposed in [DFI15], by superimposing images of chairs on background images from Flickr. Randomly sampled 2-D affine transformations are applied to both chairs and background images. Thus there are independently moving objects together with background motion. The whole dataset contains about 22k 512×384 image pairs with ground truth flows.

MPI-Sintel [BWS12b] is collected from an animation that made to be realistic. It contains scenes with natural illumination, objects moving fast, and articulated motion. Final and clean versions of the dataset are provided. The final version contains motion blur and fog effects. The training set contains only $1,041$ pairs of images, much smaller compared to Flying Chairs.

KITTI 2012 [GLU12] and 2015 [MG15] are the largest real-world datasets containing ground truth optical flows collected in a driving scenario. The ground truth flows are obtained from simultaneously recorded video and 3-D laser scans, together with some manual corrections. Even though the multi-view extended version contains roughly 15k image pairs, ground truth flows exist for only 394 pairs of image, which makes fully supervised training of optical flow prediction from scratch under this scenario infeasible. However, it provides a base for unsupervised learning of optical flow, and a stage to show the benefit of semi-unsupervised optical flow learning, that utilizes both the conditional prior (CPN) learned from the synthetic dataset, and the virtually unlimited amount of real-world videos.

3.2.3 Training details

We use Adam [KB14] as the optimizer with its default parameters in all our experiments. We train our conditional prior network (CPN) using Flying Chairs dataset due to its large amount of synthesized ground truth flows. The initial learning rate is 1.0e-4, and is halved every 100k steps until the maximum 600k training steps. The batch size is 8, and the autoencoding loss after training is around 0.6.

There are two versions of our CPNFlow, i.e. CPNFlow-C and CPNFlow-K. Both employ the FlowNetS structure, and they differ in the training set on which Eq. (3.14) is minimized. CPNFlow-C is trained on Flying Chairs dataset, similarly, CPNFlow-K is trained on KITTI dataset with the multi-view extension. The consideration here is: when trained on Flying Chairs dataset, the conditional prior network (CPN) is supposed to only capture the statistics of the affine transformations (a) CPNFlow-C is to test whether our learned prior works properly or not. If it works, (b) CPNFlow-K tests how the learned prior generalizes to real-world scenarios. Both CPNFlow-C and CPNFlow-K have the same training schedule with the initial learning rate 1.0e-4, which is halved every 100k steps until the maximum 400k steps.[7] Note that in [RYN17], layer-wise loss adjustment is used during training to simulate coarse-to-fine estimation, however, we will not adopt this training technique to avoid repeatedly interrupting the training process. In a similar spirit, we will not do network stacking as in [MHR18, IMS17], which increases both the training complexity and the network size.

In terms of data augmentation, we apply the same augmentation method as in [DFI15] whenever our network is trained on Flying Chairs dataset with a cropping of 384x448. When trained on KITTI, resized to 384x512, only vertical flipping, horizontal flipping and image order switching are applied. The batch size used for training on Flying Chairs is 8 and on KITTI is 4.

3.2.4 Benchmark results

Tab. 3.1 summarizes our evaluation on all benchmarks mentioned above, together with quantitative comparisons to the state-of-the-art methods from different categories: Fully supervised, variational, and unsupervised learning methods. Since CPNFlow has the same network structure as FlowNetS, and both CPNFlow-C and FlowNetS are trained on Flying Chairs dataset, the comparison between CPNFlow-C and FlowNetS shows that even if CPNFlow-C is trained without knowing the correspondences between pairs of image and the ground truth flows, it can still achieve similar performance compared to the fully supervised ones on the synthetic dataset MPI-Sintel. When both are applied to KITTI, CPNFlow-C achieves 11.2% and 21.6% improvement over FlowNetS and FlowNetC respectively on KITTI 2012 Train, hence CPNFlow generalizes better to out of domain data.

One might notice that FlowNet2 [IMS17] consistently achieves the highest score on MPI-Sintel and KITTI Train, however, it has a totally different network structure where several FlownetS [DFI15] and FlowNetC [DFI15] are stacked together, and it is trained in a sequential manner, and on additional datasets, e.g. FlyingThings3D [MIH16] and a new dataset designed for small displacement [IMS17], thus not directly comparable to CPNFlow. How-

[7] $\alpha = 0.1, \eta = 0.25$ for CPNFlow-C, and $\alpha = 0.045, \eta = 0.38$ for CPNFlow-K.

		Chairs	Sintel Train		Sintel Test		KITTI Train		KITTI Test	
	Methods	test	clean	final	clean	final	2012	2015	2012	2015
Sup	FlowNetS [DFI15]	2.71	4.50	5.45	7.42	8.43	8.26	—	**9.1**	—
	FlowNetC [DFI15]	**2.19**	4.31	5.87	7.28	8.81	9.35	—	—	—
	SPyNet [RB17]	2.63	4.12	5.57	6.69	8.43	9.12	—	10.1	—
	FlowNet2 [IMS17]	—	**2.02**	**3.14**	**3.96**	**6.02**	**4.09**	**10.06**	—	—
Var	Classic-NL [SRB10]	—	6.03	7.99	7.96	9.15	—	—	16.4	—
	LDOF [BM11]	**3.47**	**4.29**	6.42	**7.56**	**9.12**	13.7	—	12.4	—
	HornSchunck [SRB14]	—	7.23	8.38	8.73	9.61	—	—	**11.7**	**41.8%**
	DIS-Fast [KTD16]	—	5.61	**6.31**	9.35	10.13	**11.01**	**21.2**	14.4	—
Unsup	DSTFlow [RYN17]	5.11	6.93	7.82	10.40	11.11	16.98	24.30	—	—
	DSTFlow-ft [RYN17]	5.11	(**6.16**)	(**6.81**)	10.41	11.27	10.43	16.79	12.4	**39%**
	BackToBasic [JHD16]	5.30	—	—	—	—	11.30	—	**9.9**	—
	UnFlowC [MHR18]	—	—	—	—	—	7.11	14.17	—	—
	UnFlowC-oc [MHR18]	—	—	8.64	—	—	3.78	8.80	—	—
	UnFlowCSS-oc [MHR18]	—	—	7.91	**9.37**	10.22	**3.29**	**8.10**	—	—
	DenseNetF [ZN17]	**4.73**	—	—	—	**10.07**	—	—	11.6	—
	CPNFlow-C	**3.81**	**4.87**	**5.95**	**7.66**	**8.58**	7.33	14.61	—	—
	CPNFlow-K	4.37	6.46	7.12	—	—	3.76	9.63	4.7	30.8%
	CPNFlow-K-o	—	7.01	7.52	—	—	**3.11**	**7.82**	**3.6**	**30.4%**

Table 3.1: Quantitative evaluation and comparison to the state-of-the-art optical flow estimation methods coming from three different categories. Sup: Fully supervised, Var: Variational methods, and Unsup: Unsupervised learning methods. The performance measure is the end-point-error (EPE), except for the last column where percentage of erroneous pixels is used. The best performer in each category is highlighted in bold, and the number in parentheses is fine-tuned on the tested dataset. For more detailed comparisons on KITTI test sets, please refer to the online benchmark website: `http://www.cvlibs.net/datasets/kitti/eval_flow.php`.

ever, when we simply apply the learned conditional prior to train our CPNFlow on KITTI using Eq. (3.14), the final network CPNFlow-K surpasses FlowNet2 by 8% on KITTI 2012 Train, yet the training procedure of CPNFlow is much simpler, and there is no need to switch between datasets nor between different modules of the network.

Since the emergence of unsupervised training of optical flow [JHD16], there has not been a single method that beats the variational methods, as shown in Tab. 3.1, even if both variational methods and unsupervised learning methods are minimizing the same type of loss function. One reason might be that when we implement the variational methods, we could apply some “secret” operations as mentioned in [SRB10], e.g. median filtering, such that implicit regularization is triggered. Extra data term can also be added to bias the optimization, as in [BM11], sparse matches are used as a data term to deal with large displacements. However, when combined with our learned conditional prior, even the simplest data term would help unsupervisedly train a network that outperforms the state-of-the-art variational optical flow methods. As shown in Tab. 3.1 our CPNFlow consistently achieves similar or better performance than LDOF [BM11], especially on KITTI 2012 Train, the improvement is at least 40%.

Compared to unsupervised optical flow learning, the advantage of our learned conditional prior becomes obvious. Although DenseNetF [ZN17] and UnFlowC [MHR18] employ more powerful network structures than FlowNetS, their EPEs on MPI-Sintel Test are still 1.5 higher than our CPNFlow. Note that in [MHR18], several versions of result are reported, e.g. UnFlowC: trained with brightness data term and second order smoothness term, UnFlowC-oc: census transform based data term together with occlusion modeling and bidirectional flow consistency penalty, and UnFlowCSS-oc: a stack of one FlowNetC and two FlowNetS’s sequentially trained using the same loss as in UnFlowC-oc. Our CPNFlow-K outperforms UnFlowC by 47% on KITTI 2012 Train and 32% on KITTI 2015 Train. When occlusion reasoning is effective in Eq. (3.14) as done in [MHR18], our CPNFlow-K-o outperforms UnFlowC-oc by 17.7% on KITTI 2012 Train, 11.1% on KITTI 2015 Train, and 12.9% on Sintel Train Final, even without a more robust census transform based data term and flow consistency penalty, which demonstrates the effectiveness of our learned conditional prior across different data terms. Note that our CPNFlow-K-o even outperforms UnFlowCSS-oc, which is far more complex in training and network architecture.

Fig. 3.5, Fig. 3.6, Fig. 3.7 show the visual comparisons on MPI-Sintel, KITTI 2012 and KITTI 2015 respectively. Note that our CPNFlow is generally much smoother, and at the

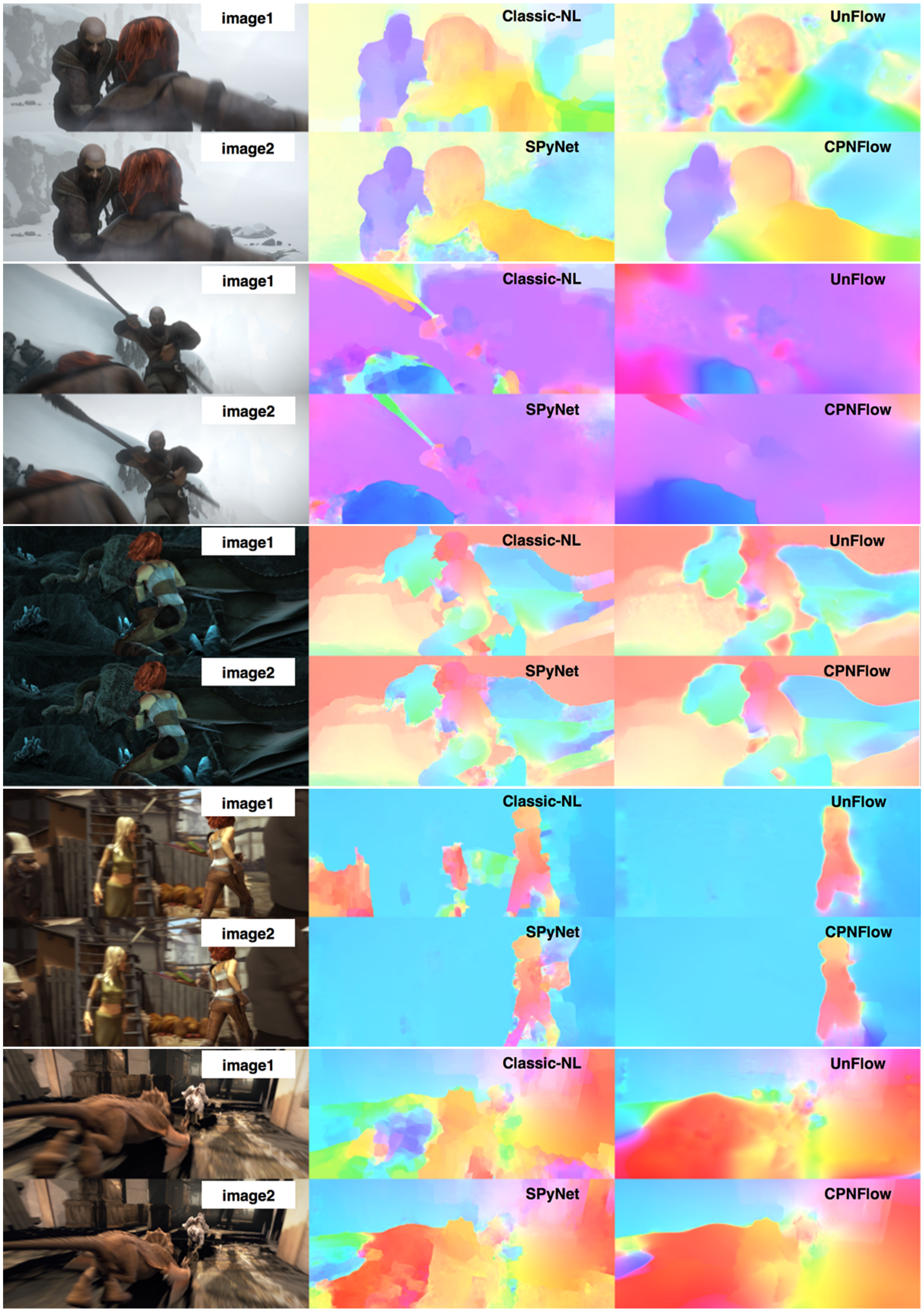


Figure 3.5: Visual comparison on MPI-Sintel. Variational: CLassic-NL [SRB10], Supervised: SPyNet [RB17], Unsupervised: UnFlowC [MHR18] and our CPNFlow-C.

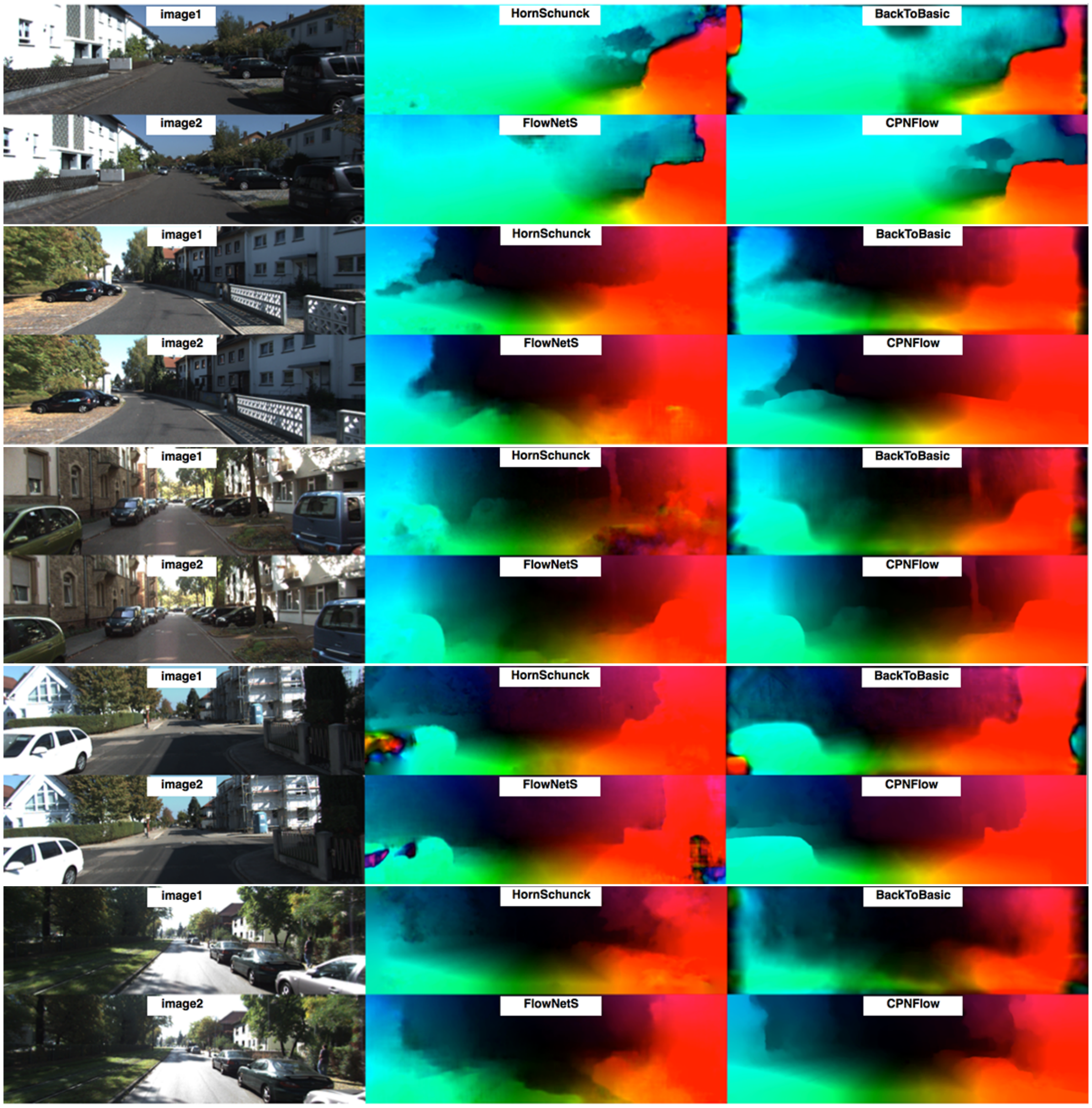


Figure 3.6: Visual comparison on KITTI 2012. Variational: HornSchunck [SRB14], Supervised: FlowNetS [DFI15], Unsupervised: BackToBasic [JHD16] and our CPNFlow-K.

same time sharper at object boundaries, e.g. the girl in the 3rd, 4th rows and the dragon in the 5th row in Fig. 3.5. This demonstrates that our conditional prior network (CPN) is capable of learning high level (semantic) regularities imposed by object entities. In Fig. 3.6, we can also observe that discontinuities in the flow fields align well with object boundaries, for example, the cars in all pairs. This, again, demonstrates that our learned conditional prior is able to generalize to different scenarios. The error of the estimated flows is also displayed in Fig. 3.7.

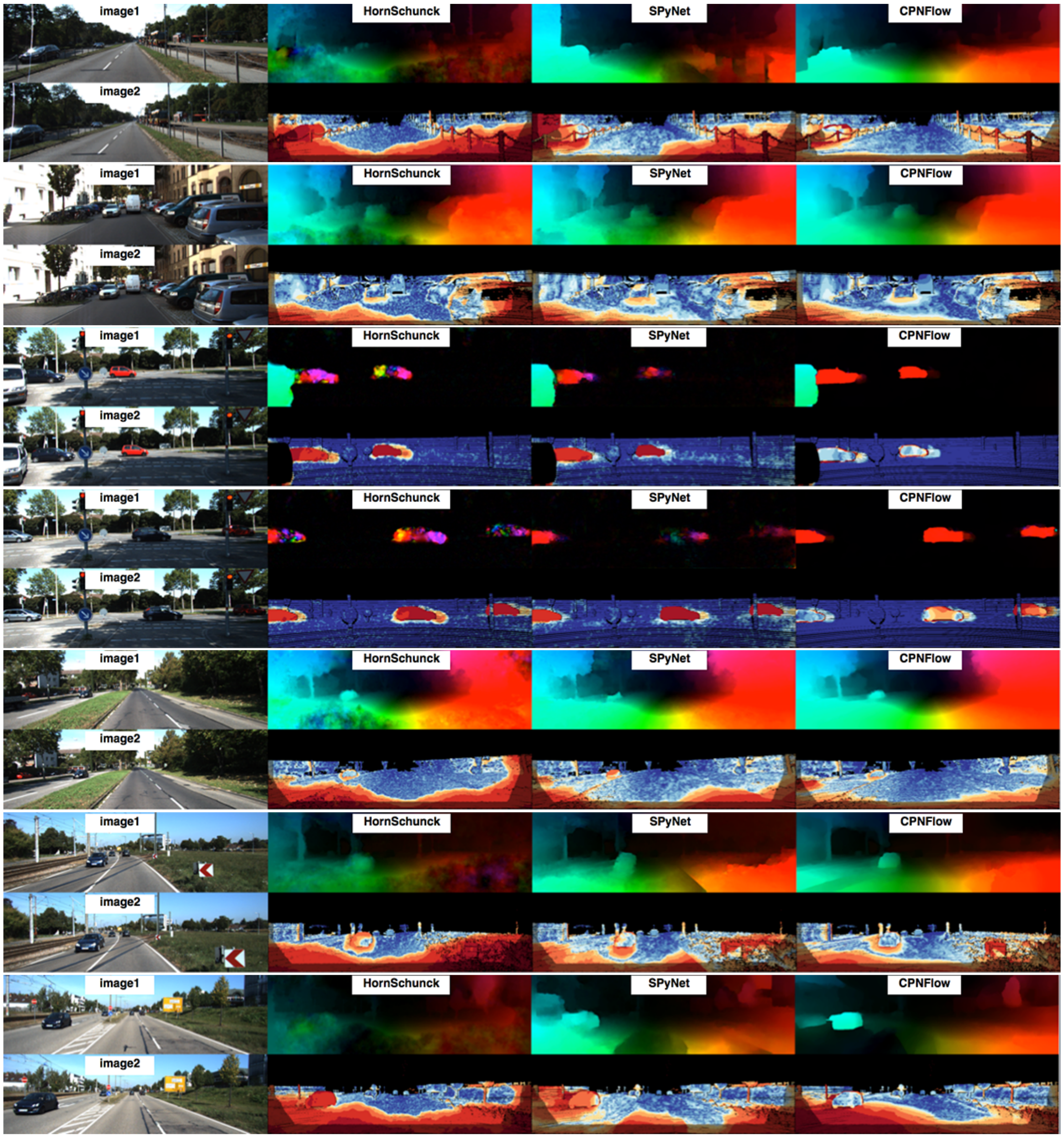


Figure 3.7: Visual comparison on KITTI 2015. Variational: HornSchunck [SRB14], Supervised: SPyNet [RB17] and our CPNFlow-K. The 2nd row in each pair shows the end-point-error of the estimated flow, red is high and blue is low.

3.3 Discussion

It would be tempting to use a GAN [GPM14] to learn the prior distribution of interest. A GAN can be thought of as a method to learn a map g such that its push-forward g_* maps two distributions, one known μ, and one we can sample from, p, so $\hat{g} = \arg\min \mathbb{KL}(g_*\mu||p)$.

It does so via an adversarial process such that a generative model G will capture the data distribution p_{data}. If we sample from the generative model G, we will have samples that are equivalently sampled from p_{data}, in order to evaluate $p_{data}(x)$ of a sample x, we can not circumvent the sampling step, thus making the method unsuitable for our purpose where we want a differentiable scalar function.

Our work entails constructing an autoencoder of the flow, so it naturally relates to [KW13]. Similarly, evaluating the probability of a test example is intractable, even if we can approximately evaluate the lower bound of the probability of a data point, which again cannot be computed in closed form due to the expectation over the noise.

CHAPTER 4

Dense Depth Posterior from Single Image & Sparse Range

In this chapter, we shift a bit from motion perception to depth perception. Since motion and depth are two closely related problems in computer vision, algorithms developed for motion estimation, e.g. optical flow, can usually be applied to depth perception, e.g. stereo. However, here we focus on a slightly different flavor. We present a deep learning system to infer the posterior distribution of a dense depth map associated with an image, by exploiting sparse range measurements, for instance from a lidar or a SLAM system. While the sparse depth value is only valid on a small percentage of the pixels, we exploit regularities reflected in the training set (past experience) to complete the depth map so as to have a probability over depth for each pixel in the image. Optionally we can also exploit the availability of stereo during training, but in any case, only require a single image and a sparse point cloud at run-time.

Depth completion is highly ill-posed: There are infinitely many dense depth maps that are compatible with a given image and a sparse point cloud. Any point-estimate, therefore, depends critically on the prior assumptions made. Ideally, one would compute the entire posterior distribution of depth maps, rather than a point-estimate, given an image and a sparse point cloud. The posterior affords to reason about confidence, integrating evidence over time, and in general, is a (Bayesian) sufficient representation that accounts for all the information in the data.

In autonomous navigation, a sparse point cloud from lidar may be insufficient to make planning decisions: Is the surface of the road in Fig. 4.1 (middle, better viewed when enlarged) littered with pot-holes, or is it a smooth surface? Points that are nearby in image topology, projecting onto adjacent pixels, may be arbitrarily far in the scene. For instance, pixels that straddle an occluding boundary correspond to large depth gaps in the scene. While the lidar may not measure every pixel, if we know it projects onto a tree, trees tend to stand out from the ground, which informs the topology of the scene. On the other hand, pixels that straddle illumination boundaries, like shadows cast by trees, seldom correspond to large depth discontinuities.

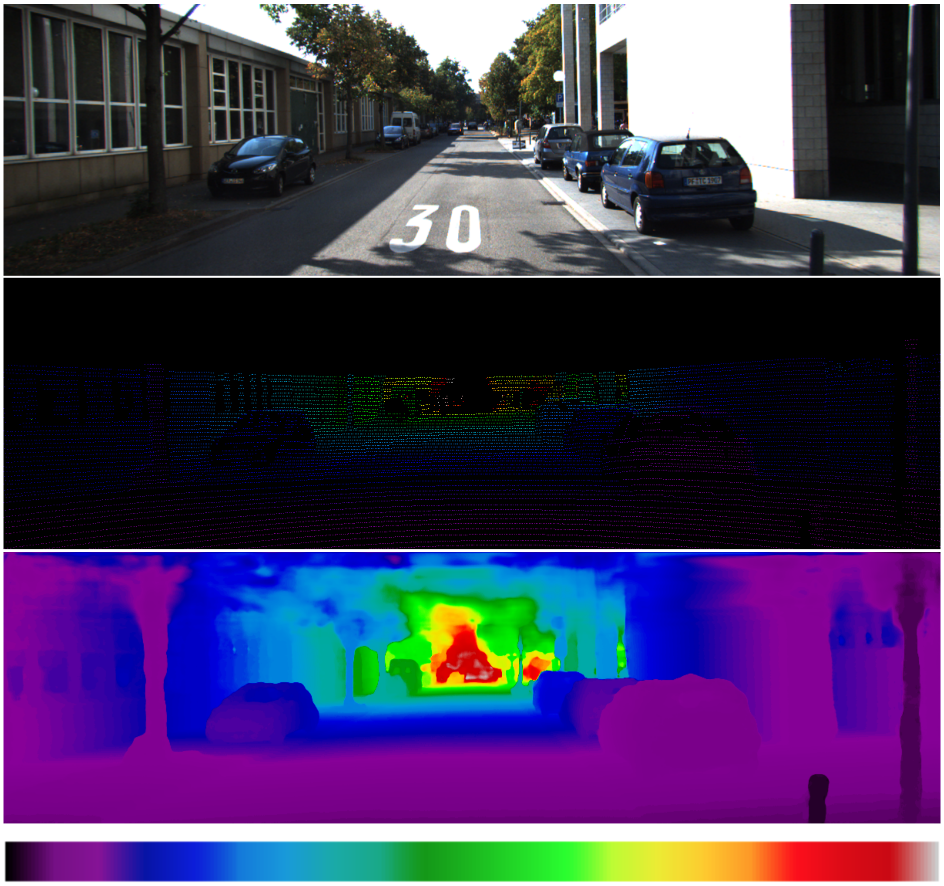


Figure 4.1: An image (top) is insufficient to determine the geometry of the scene; a point cloud alone (middle) is similarly ambiguous. Lidar returns are shown as colored points, but black regions are uninformative: Are the black regions holes in the road surface, or due to radiometric absorption? Combining a single image, the lidar point cloud, and previously seen scenes allows inferring a dense depth map (bottom) with high confidence. Color bar from left to right: zero to infinity.

Structured light sensors typically provide dense depth measurements with about 20% missing values; At this density, the problem is akin to inpainting [CS12, LRL14, SC13] that use morphological operations [KHW18, PGA16]. There is no need for annotated datasets [GLU12, SF11, SHK12]. The regime we are interested in involves far sparser point clouds ($> 90\%$ missing values).

Depth completion is the process of assigning a depth value to each pixel. Supervised deep learning-based methods [EFK18, HFY18, RPY18, USS17, ZF18] minimize the corresponding loss between prediction (from a single RGB image and its associated sparse depth measurements) and ground truth depth. [USS17] trains a deep network to regress depth using a sparse convolutional layer that discounts the invalid depth measurements while [HFY18] proposes a sparsity-invariant upsampling layer, sparsity-invariant summation, and joint sparsity-invariant concatenation and convolution. [EFK18] treat the binary validity map as a confidence map and adapts normalized convolution for confidence propagation through layers. [DVP18a] implements an approximation of morphological operators using the contra-harmonic mean (CHM) filter [MAS13] and incorporates it as a layer in a U-Net architecture for depth completion. [CWL18] proposes a deep recurrent auto-encoder to mimic the optimization procedure of compressive sensing for depth completion, where the dictionary is embedded in the neural network. [ZF18] predicts surface normals and occlusion boundaries from the RGB image, which gives a coarse representation of the scene structure. When the dense depth map is not available for training, we have the unsupervised depth completion [MWA18, WBZ18, ZBS17]. [MCK18] proposes minimizing the photometric constancy loss among a sequence of images with a second-order smoothness prior. [FNP16, XGF16] and unsupervised single image depth prediction [GBC16, GMB17] proposed using novel view synthesis to hallucinate the existence of a novel view using an image reconstruction loss. In the case of stereo pairs, [GBC16, GMB17] propose training networks to predict the disparities of an input image by reconstructing the unseen right view of a stereo pair given the left image as input. [GMB17] additionally proposed edge-aware smoothness and left-right consistency. We also exploit stereo as in Sect. 4.1.3, where we incorporate only the stereo photometric reconstruction term. While methods [EFK18, HFY18, MCK18, RPY18, USS17, ZF18] learn a representation for the depth completion task through ground truth supervision, they do not have any explicit modeling of the semantics of the scene. Recently, [SSP16] explored this direction by predicting object boundary and semantic labels through a deep network and using them to construct locally planar elements that serve as input to a global energy minimization for depth completion. [CWY18] proposes to complete the depth by anisotropic diffusion with a recurrent convolution network, where the affinity matrix is computed locally from an image. [JDW18] also trains a U-Net for joint depth completion and semantic segmentation in the form of multitask learning in an effort to incorporate semantics in the learning process.

We wish to infer the entire posterior estimate over depths. Sparse range measurements serve to ground the posterior estimate in a metric space. This could then be used by a decision and control engine downstream. We exploit Conditional Prior Network (CPN) [YS18] to learn the conditional prior to take into account scene semantics rather than using a local smoothness assumption. We leverage this technique and formulate depth completion as a maximum a-posteriori problem by factorizing it into a likelihood term and a conditional prior term, making it possible to explicitly model the semantics induced regularity of a single image.

Side information. If the dense depth map is obtained by processing the given image and sparse point cloud alone, the quality of the resulting decision or control action could be no better than if the raw data was fed downstream (Data Processing Inequality). However, if depth completion can exploit a prior or aggregate experience from previously seen images and corresponding dense depth maps, then it is possible for the resulting dense depth map to improve the quality of the decision or action, assuming that the training set is representative. To analyze a depth completion algorithm, it is important to understand what prior assumptions, hypotheses or side information is being exploited.

Goal. We seek methods to *estimate the geometry and topology of the scene given an image, a sparse depth map, and a body of training data consisting of images and the associated dense depth maps.* Our assumption is that the distribution of seen images and corresponding depth maps is representative of the present data (image and sparse point cloud) once restricted to a sparse domain.

Our method yields the full posterior over depth maps, which is much more powerful than any point estimate. For instance, it allows reasoning about confidence intervals. Since there is no benchmark dataset to evaluate the accuracy of the posterior, we elect the simplest point estimate possible, which is the maximum. It should be noted, however, that when there are multiple hypotheses with similar posterior, the point estimate could jump from one mode to another, and yet the posterior being an accurate representation of the unknown variable. More sophisticated point estimators, for instance, taking into account memory, or spatial distribution, non-maximum suppression, etc. could be considered, but here we limit ourselves to the simplest one.

Key idea. While an image alone is insufficient to determine a depth map, certain depth maps are more probable than others given the image and a previously seen dataset. The key to our approach is a conditional prior model $P(d|I, \mathcal{D})$ that scores the compatibility of each dense depth map d with the given image I based on the previously observed dataset $\mathcal{D}$. This is computed using a Conditional Prior Network (CPN) [YS18] in conjunction with a model of the likelihood of the observed sparse point cloud z under the hypothesized depth map d, to yield the posterior probability and, from it, a maximum a-posteriori (MAP) estimate of the depth map for evaluation:

$$\hat{d} = \arg\max_{d} P(d|I, z) \propto P(z|d) P_{\mathcal{D}}(d|I) \tag{4.1}$$

Here $D \subset \mathbb{R}^2$ is the image domain, sampled on a regular lattice of dimension $N \times M$, $I : D \to \mathbb{R}^3$ is a color image, with the range quantized to a finite set of colors, $d : D \to \mathbb{R}+$ is the dense depth map defined on the lattice D, which we represent with an abuse of notation as a vector of dimension MN: $d \in \mathbb{R}_+^{NM}$. $\Omega \subset D$ is a sparse subset of the image domain, with cardinality $K = |\Omega|$, where the function d takes values $d(\Omega) = z \in \mathbb{R}_+^K$. Finally, $\mathcal{D} = \{d_j, I_j\}_{j=1}^n$ is a dataset of images I_j and their corresponding dense depth maps $d_j \in \mathbb{R}_+^{NM}$. Since we do not treat $\mathcal{D}$ as a random variable but a given set of data, we write it as a subscript. In some cases, we may have additional data available during training, for instance stereo imagery, in which case we include it in the dataset, and discuss in detail how to exploit it in Sect. 4.1.3.

4.1 Method

In order to exploit a previously observed dataset $\mathcal{D}$, we use the Conditional Prior Network (CPN) described in the previous Chapter, integrated into the loss for training the Depth Completion Network (DCN) shown in Fig. 4.2. Conditional Prior Networks infer the probability of an optical flow given a single image. During training, ground truth optical flow is encoded (upper branch in Fig. 4.2-A), concatenated with the encoder of an image (lower branch), and then decoded into a reconstruction of optical flow.

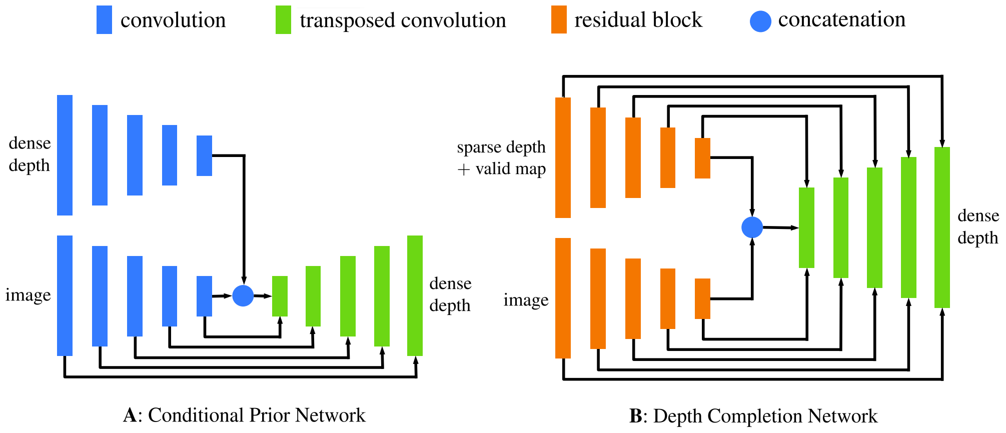


Figure 4.2: (A): the architecture of the Conditional Prior Network (CPN) to learn the conditional of the dense depth given a single image. (B): Our proposed Depth Completion Network (DCN) for learning the mapping from a sparse depth map and an image to a dense depth map. Connections within each encoder/decoder block are omitted for simplicity.

In our implementation, the upper branch encodes dense depth, concatenated with the encoding of the image, to produce a dense reconstruction of depth at the decoder, together with a normalized likelihood that can serve as a posterior score. We consider a CPN as a function that, given an image (lower branch input) maps any sample putative depth map (upper branch input) to a positive real number, which represents the conditional probability of the input dense depth map given the image.

We denote the ensemble of parameters in the CPN as w^{CPN}; with an abuse of notation, we denote the decoded depth with $d' = w^{CPN}(d, I)$. When trained with a bottleneck imposed on the encoder (upper branch), the reconstruction error is proportional to the conditional distribution:

$$Q(d, I; w^{CPN}) = e^{-\|w^{CPN}(d,I)-d\|^{\eta}} \propto P_{\mathcal{D}}(d|I) \tag{4.2}$$

where, η indicates the specific norm used for calculating Q. In Sect. 4.2.2 and Sect. 4.3, we show the training details of CPN for a conditional prior on dense depth maps, and also quantitatively show the effect of different choices of the norm η.

In order to obtain a posterior estimate of depth, the CPN needs to be coupled with a likelihood term.

4.1.1 Supervised single image depth completion

Supervised learning of dense depth assumes the availability of ground truth dense depth maps. In the KITTI depth completion benchmark [USS17], these are generated by accumulating the neighboring sparse lidar measurements. Even though it is called ground truth, the density is only $\sim$ 30% of the image domain, whereas the density of the unsupervised benchmark is $\sim$ 5%. The training loss in the supervised modality is just the prediction error:

$$L(w) = \sum_{j=1}^{N} \|\phi(z_j, I_j; w) - d_j\|^{\gamma} \tag{4.3}$$

where ϕ is the map from sparse depth z and image I to dense depth, realized by a deep neural network with parameters w, and $\gamma = 1$ fixed in the supervised training.

Our network structure for ϕ is detailed in Fig. 4.2-B, which has a symmetric two-branch structure, each encoding different types of input: one sparse depth, the other an image; skip connections are enabled for two branches. Note that our network structure is unique among all the top performing ones on the KITTI depth completion benchmark: We do not use specifically-designed layers for sparse inputs, such as sparsity invariant layers [HFY18, USS17]. Instead of early fusion of sparse depth and image, our depth defers fusion to decoding, which entails fewer learnable parameters. To elaborate, Ma's [MCK18] encoder contains a total of $\approx$12.1M parameters and ours $\approx$6.2M. We use the same decoder, which contains $\approx$1.6M. This gives Ma's architecture a total of $\approx$13.7M and ours $\approx$7.8M – an effective 48.8% reduction in the encoder and a 43% reduction overall. A related idea was also proposed in [JDW18]; instead of a more sophisticated NASNet block [ZVS], we use more common ResNet block [HZR16]. Although simpler than competing methods, our network achieves state-of-the-art performance (Sect. 4.3).

4.1.2 Unsupervised single image depth completion

Supervised learning requires ground truth dense depth, which is hard to come by. Even the "ground truth" provided in the KITTI benchmark is only 30% dense and interpolated from even sparser maps. When only sparse independent measurements of depth are available, for instance from lidar, with less than 10% coverage (e.g. 5% for KITTI), we call depth completion *unsupervised* as the only input are sensory data, from images and a range measurement device, with no annotation or pre-processing of the data.

The key to our approach is the use of a CPN to score the compatibility of each dense depth map d with the given image I based on the previously observed data $\mathcal{D}$. In some cases, we may have additional sensory data available *during training*, for instance, a second image taken with a camera with a known relative pose, such as stereo. In this case, we include the reading from the second camera in the training set $\mathcal{D}$, as described in Sect. 4.1.3. When only a single image is given, the CPN (4.2) is combined with a model of the likelihood of the observed sparse point cloud z under the hypothesized depth map d:

$$P(z|d) \propto e^{-\|z-d(\Omega)\|^\gamma} \tag{4.4}$$

which is simply a Gaussian around the hypothesized depth, restricted to the sparse subset Ω, when $\gamma = 2$. The overall loss is:

$$\begin{aligned} L^u(w) = -\sum_{j=1}^N logP(d_j|I_j, z_j, \mathcal{D}) \\ = \sum_{j=1}^N \|z_j - d_j(\Omega)\|^\gamma + \alpha \sum_{j=1}^N \|w^{CPN}(d_j, I_j) - d_j\|^\eta \\ = \sum_{j=1}^N \|z_j - \phi(z_j, I_j; w)(\Omega)\|^\gamma + \alpha \sum_{j=1}^N \|w^{CPN}(\phi(z_j, I_j; w), I_j) - \phi(z_j, I_j; w)\|^\eta \end{aligned} \tag{4.5}$$

Note that γ, η control the actual norm used during training, as well as the modeling of the likelihood and conditional distribution. We experiment with these parameters in Sect. 4.3.1, and show the quantitative analysis there.

4.1.3 Disparity supervision

Some datasets come with stereo imagery. We want to be able to exploit it, but without having to require its availability at inference time. We exploit the strong relation between depth and disparity. In addition to the sparse depth z and the image I, we are given a second image I' as part of a stereo pair, which is rectified (standard pre-processing), to first-order we assume that there exists a displacement $s = s(x), x \in D$ such that

$$I(x) \approx I'(x+s) \tag{4.6}$$

which is the intensity constancy assumption. We model, again simplistically, disparity s as $s = FB/d$, where F is the focal length and B is the baseline (distance between the optical centers) of the cameras. Hence, we can synthesize s from the predicted d, thus to constrain the recovery of 3-d scene geometry. More specifically, we model the likelihood of seeing I' given I, d as:

$$P(I'|I,d) \propto e^{-\frac{\sum\limits_{x} \|I(x) - I'(x+s(d(x)))\|}{\delta^2}} \tag{4.7}$$

However, the validity of the intensity constancy assumption is affected by complex phenomena such as translucency, transparency, inter-reflection, etc. In order to mitigate the error in the assumption, we could also employ a perceptual metric of structural similarity (SSIM) [WBS04]. SSIM scores corresponding 3×3 patches $p(x), p'(x) \in \mathbb{R}_+^{3\times 3}$ centered at x in I and I', respectively, to measure their local structural similarity. A higher score denotes more similarity; hence we can subtract the scores from 1 to form a robust version of Eq. 4.7. We use $P_{raw}(I'|I,d)$ and $P_{ssim}(I'|I,d)$ to represent the likelihood measured in raw photometric value and SSIM score respectively. When the stereo pair is available, we can form the conditional prior as follows:

$$P(d|I,I',\mathcal{D}) \propto P(I'|I,d,\mathcal{D})P(d|I,\mathcal{D}) = P(I'|I,d)P_{\mathcal{D}}(d|I) \tag{4.8}$$

Similar to the training loss Eq. (4.5) for unsupervised single image setting, we can derive the loss for stereo setting as follows:

$$
\begin{aligned}
L^s(w) = &-\sum_{j=1}^{N} logP(d_j|I_j, I'_j, z_j, \mathcal{D}) \\
&= L^u(w) + \beta \sum_{j,x} \|I_j(x) - I'_j(x + s(d_j(x)))\| \quad (4.9)
\end{aligned}
$$

where $d_j = \phi(z_j, I_j; w)$ and L^u is the loss defined in Eq. (4.5). Note that, the above summation term is the instantiation for $P_{raw}(I'|I, d)$, which can also be replaced by the SSIM counterpart. Rather than choosing one or the other, we compose the two with tunable parameters β_c and β_s, our final loss for stereo setting depth completion is:

$$
L^s(w) = L^u(w) + \beta_c\psi_c + \beta_s\psi_s \quad (4.10)
$$

with ψ_c represents the raw intensity summation term in Eq. (4.9), and ψ_s for the SSIM counterpart. Next, we elaborate our implementation details and evaluate the performance of our proposed method in different depth completion settings.

4.2 Implementation Details

4.2.1 Network architecture

We modify the public implementation of CPN [YS18] by replacing the input of the encoding branch with a dense depth map. Fusion of the two branches is simply a concatenation of the encodings. The encoders have only convolutional layers, while the decoder is made of transposed convolutional layers for upsampling.

Our proposed network, unlike the base CPN, as seen in Fig. 4.2-A, contains skip connections between the layers of the depth encoder and the corresponding decoder layers, which makes the network symmetric. We also use ResNet blocks [HZR16] in the encoders instead of pure convolutions. A stride of 2 is used for downsampling in the encoder and the number of channels in the feature map after each encoding layer is $[64*k, 128*k, 256*k, 512*k, 512*k]$. In all our experiments, we use $k = 0.25$ for the depth branch, and $k = 0.75$ for the image branch, taking into consideration that an RGB image has three channels while depth map only has one channel.

4.2.2 Training Procedure

We begin by detailing the training procedure for CPN. Once learned, we apply CPN as part of our training loss and do not need it during inference. In order to learn the conditional prior of the dense depth maps given an image, we require a dataset with images and corresponding dense depth maps. We are unaware of any real-world dataset for outdoor scenes that meets our criterion. Therefore, we train the CPN using the Virtual KITTI dataset [GWC16]. It contains 50 high-resolution monocular videos with a total of $21,260$ frames, together with ground truth dense depth maps, generated from five different virtual worlds under different lighting and weather conditions. The original Virtual KITTI image has a large resolution of 1242×375, which is too large to feed into a normal commercial GPU. So we crop it to 768×320 and use a batch size of 4 for training. The initial learning rate is set to $1e^{-4}$, and is halved every 50,000 steps 300,000 steps in total.

We implement our approach using TensorFlow [ABC16]. We use Adam [KB14] to optimize our network with the same batch size and learning rate schedule as the training of CPN. We apply histogram equalization and also randomly crop the image to 768×320. We additionally apply random flipping both vertical and horizontal to prevent overfitting. In the case of unsupervised training, we also perform a random shift within a 3×3 neighborhood to the sparse depth input and the corresponding validity map. We use $\alpha = 0.045$, $\beta = 1.20$ for Eq. (4.9), and the same α is applied with $\beta_c = 0.15$, $\beta_s = 0.425$ for Eq. (4.10). We choose $\gamma = 1$ and $\eta = 2$, but as one may notice in Eq. (4.2), the actual conditional prior also depends on the choice of the norm η. To show the reasoning behind our choice, we will present as an empirical study in Fig. 4.3 to show the effects of the different pairing of norms with a varying α by evaluating each model on the RMSE metric.

In the next section, we report representative experiments in both the supervised and unsupervised benchmarks.

4.3 Experiments

We evaluate our approach on the KITTI depth completion benchmark [USS17]. The dataset provides $\sim 80k$ raw image frames and corresponding sparse depth maps. The sparse depth maps are the raw output from the Velodyne lidar sensor, each with a density of about 5%. The ground truth depth map is created by accumulating the neighboring 11 raw lidar scans, with roughly 30% pixels annotated. We use the officially selected 1,000 samples for validation and we apply our method to 1,000 testing samples, with which we submit to the official KITTI website for evaluation.

Method	iRMSE	iMAE	RMSE	MAE	Rank
Dimitrievski [DVP18b]	3.84	1.57	1045.45	310.49	13.0
Cheng [CWY18]	2.93	1.15	1019.64	279.46	7.5
Huang [HFY18]	2.73	1.13	841.78	253.47	6.0
Ma [MCK18]	2.80	1.21	**814.73**	249.95	5.5
Eldesokey [EFK18]	2.60	1.03	829.98	233.26	4.75
Jaritz [JDW18]	2.17	0.95	917.64	234.81	3.0
Ours	**2.12**	**0.86**	836.00	**205.40**	**1.5**

Table 4.1: Quantitative results on the supervised KITTI depth completion benchmark. Our method achieves state of the art performance in three metrics, iRMSE, iMAE, and MAE. [MCK18] performs better than us by 2.6% on the RMSE metric; however, we outperform [MCK18] on all other metrics by 24.3%, 28.9% and 17.8% on the iRMSE, iMAE and MAE, respectively. The last column is the average rank over ranks on all the four metrics.

4.3.1 Norm Selection

As seen in Eq. (4.5), γ, η control the actual norms (penalty functions) applied to the likelihood term and conditional prior term respectively, which in turn determine how we model the distributions. General options are from the binary set $\{1, 2\}$. i.e. $\{\mathcal{L}_1, \mathcal{L}_2\}$, however, there is

	Validation Set		Test Set			
Loss	RMSE	MAE	iRMSE	iMAE	RMSE	MAE
Ma [MCK18]	1384.85	358.92	4.07	1.57	1299.85	350.32
L^u	1325.79	355.86	3.69	1.37	1285.14	353.16
$L^u + \psi_c$	1320.26	353.24	3.63	1.34	1274.65	349.88
$L^u + \psi_c + \psi_s$	**1310.03**	**347.17**	**3.58**	**1.32**	**1263.19**	**343.46**

Table 4.2: Quantitative results on the unsupervised KITTI depth completion benchmark. Our baseline approach using CPN as a regularizer outperforms [MCK18] on the iRMSE, iMAE and RMSE metrics on the test set, whereas [MCK18] marginally performs better than us on MAE by 0.8%. We note that [MCK18] achieves this performance using photometric supervision. When including our photometric term (Eq. (4.10)), we outperform [MCK18] on every metric and achieve state-of-the-art performance.

currently no agreement on which one is better suited for the depth completion task. [MCK18] shows $\gamma = 2$ gives significant improvement for their network, while both [USS17, JDW18] claim to have better performance when $\gamma = 1$ is applied. In our approximation of the posterior in Eq. (4.5), the choice of the norms gets more complex as the modeling (norm) of the conditional prior will also depend on the likelihood model. Currently, there is no clear guidance on how to make the best choice, as it may also depend on the network structure. Here we try to explore the characteristic of different norms, at least for our network structure, by conducting an empirical study on a simple version (channel number of features reduced) of our depth completion network using different combinations of γ and η. As shown in Fig. 4.3, the performance on the KITTI depth completion validation set varies in a wide range with different γ, η. Clearly for our depth completion network, $\mathcal{L}_1$ is always better than $\mathcal{L}_2$ on the likelihood term. And the lowest RMSE is achieved when a $\mathcal{L}_2$ is also applied on the conditional prior term. Thus the best coupling is $\gamma = 1, \eta = 2$ for Eq. (4.5).

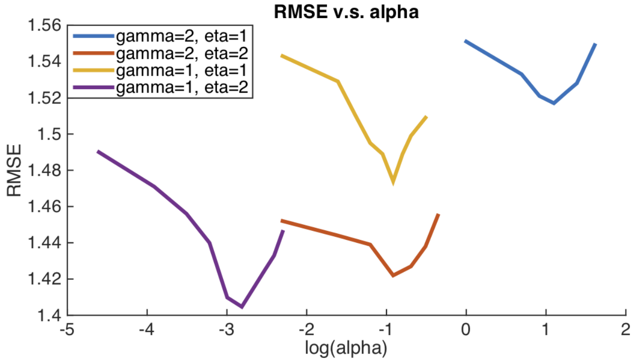


Figure 4.3: This plot shows the empirical study on the choice of norms γ, η in the likelihood term and the conditional prior term respectively. Each curve is generated by varying α in Eq. (4.5) with fixed γ, η. And the performance is measured in RMSE.

4.3.2 Supervised depth completion

We evaluate the proposed Depth Completion Network described in Sect. 4.1.1 on the KITTI depth completion benchmark. We show a quantitative comparison between our approach and the top performers on the benchmark in Tab. 4.1. Our approach achieves the state-of-the-art in three metrics by outperforming [EFK18, JDW18], who each held the state-of-the-art in different metrics on the benchmark. We improve over [JDW18] in iRMSE and iMAE by 2.3% and 9.5%, respectively, and [EFK18] in MAE by 11.9%. [MCK18] performs better on the RMSE metric by 2.6%; however, we outperform [MCK18] by 24.3%, 28.9% and 17.8% on the iRMSE, iMAE and MAE metrics, respectively. Note in the online table of KITTI depth completion benchmark[1], all methods are solely ranked by the RMSE metric, which may not fully reflect the performance of each method. Thus we propose to rank all methods by averaging over the rank numbers on each metric, and the overall ranking is shown in the last column of Tab. 4.1. Not surprisingly, our depth completion network gets the smallest rank number due to its generally good performance on all metrics.

Fig. 4.4 shows a qualitative comparison of our method to the top performing method on the test set of the KITTI benchmark. We see that our method produces depths that are more consistent with the scene with fewer artifacts (e.g. grid-like structures [MCK18], holes in

[1] `http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_completion`

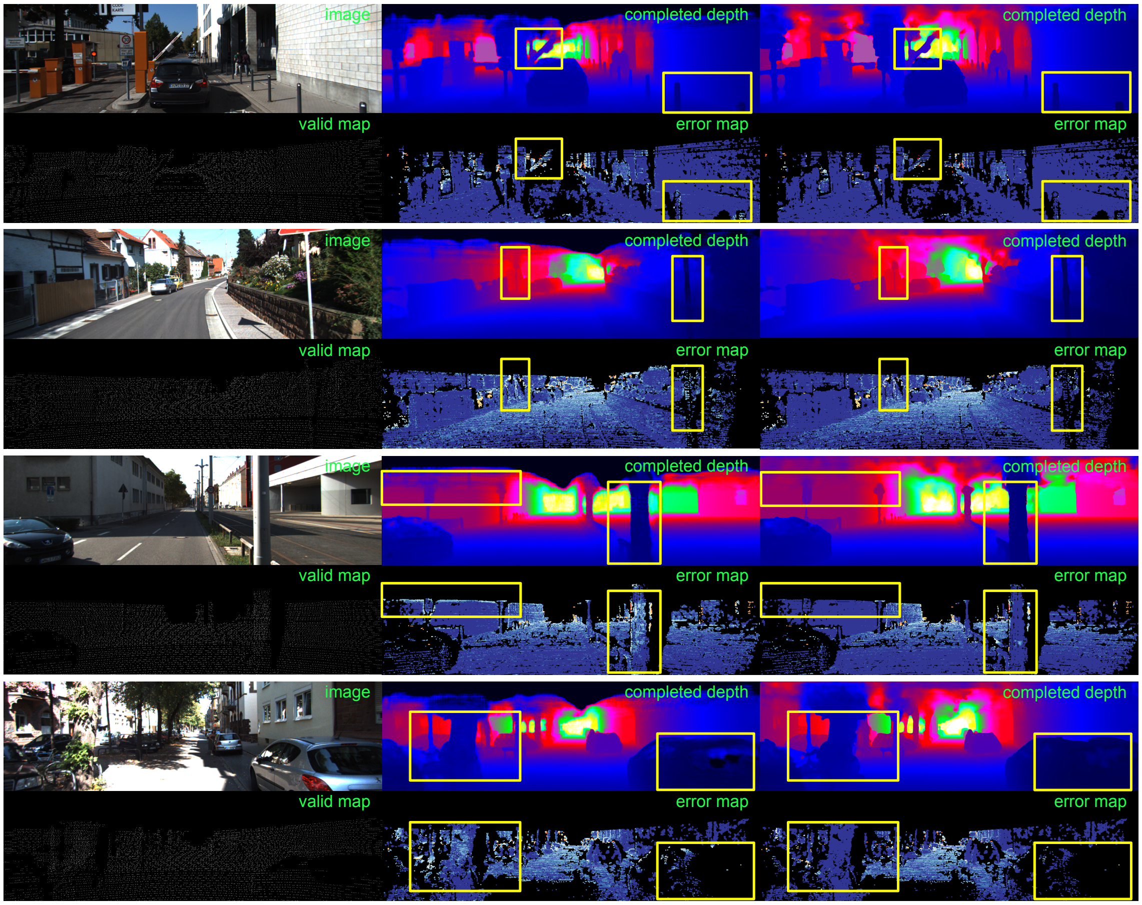


Figure 4.4: Qualitative comparison to Ma et al. [MCK18] on KITTI depth completion test set using ground truth depth as supervision. From left to right: image and validity map of the sparse measurements, dense depth results and corresponding error map of [MCK18] and our results and error map. Warmer colors in the error map denote higher error. The yellow rectangles highlight the regions for detailed comparison. Note that our network consistently performs better on fine and far structures and our completed dense depth maps have less visual artifacts.

objects [EFK18]). Also, our network performs consistently better on fine and far structures, which may be traffic signs and poles on the roadside that provide critical information for safe driving as shown in the second row in Fig. 4.4.

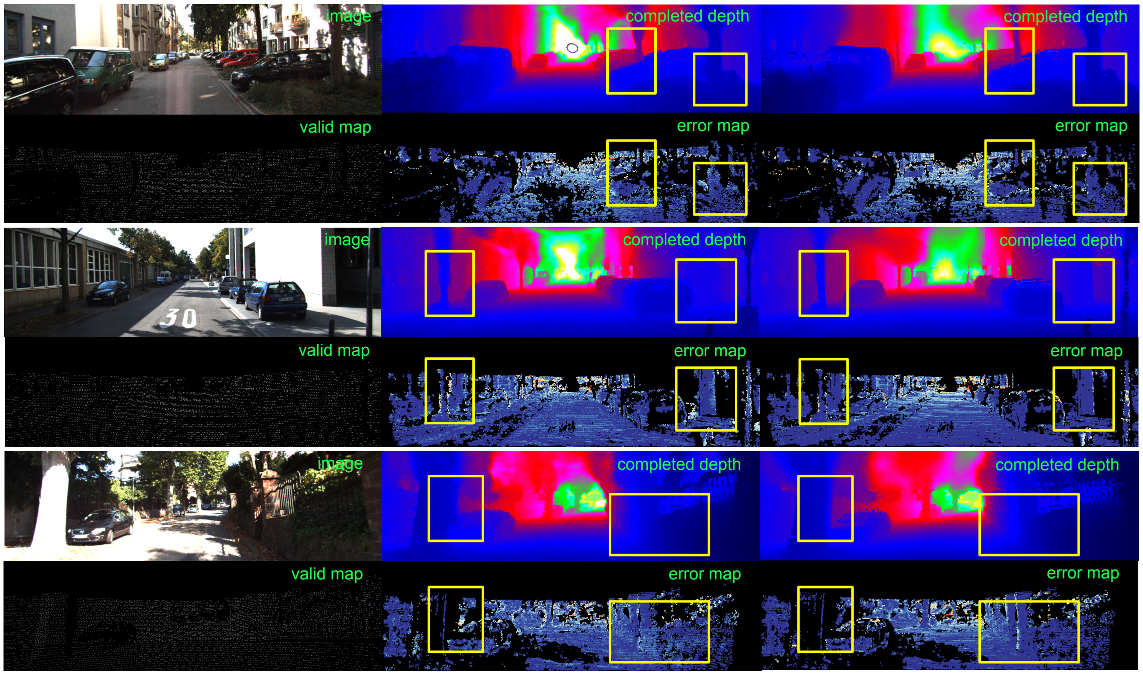


Figure 4.5: Qualitative comparison to Ma et al. [MCK18] on KITTI depth completion test set in the unsupervised setting. From left to right: image and validity map of the sparse measurements, dense depth results and corresponding error map of [MCK18] and our results and error map. Warmer colors in the error map denote higher error. The yellow rectangles highlight the regions for detailed comparison. Note again that our network consistently performs better on fine and far structures and our completed dense depth maps have less visual artifacts. This includes the circle in the center of their prediction (row 1).

4.3.3 Unsupervised depth completion

We show that our network can also be applied to unsupervised setting using only the training loss Eq. (4.5) to achieve the state-of-the-art results as well. We note that the simplest way for the network to minimize the data term is to directly copy the sparse input to the output, which will make the learning inefficient. To facilitate the training, we change the stride of the first layer from 1 to 2 and replace the final layer of the decoder with a nearest neighbor upsampling.

We show a quantitative comparison (Tab. 4.2) between our method and that of [MCK18] along with an ablation study on our loss function. We note that the results of [MCK18] are achieved using their full model, which includes their multi-view photometric term. Our approach using just Eq. (4.5) is able to outperform [MCK18] in every metric with the exception of MAE where [MCK18] marginally beats us by 0.8%. By applying our reconstruction loss Eq. (4.9), we outperform [MCK18] in every metric. Moreover, our full model Eq. (4.10) further improves over all other variants and is state-of-the-art in unsupervised depth completion. We present a qualitative comparison between our approach and that of [MCK18] in Fig. 4.5. Visually, we observe the results of [MCK18] still contain the artifacts as seen before. The artifacts, i.e. circles, as detailed in Fig. 4.5, are signs that their network is probably overfitted to the input sparse depth, due to the lack of semantic regularity. Our approach, however, does not suffer from these artifacts; instead, our predictions are globally correct and consistent with the scene geometry.

4.4 Summary

In this work, we have described a system to infer a posterior probability over the depth of points in the scene corresponding to each pixel, given an image and a sparse aligned point cloud. Our method leverages a Conditional Prior Network, that allows the association of a probability to each depth value based on a single image, and combines it with a likelihood term for sparse depth measurements. Moreover, we exploit the availability of stereo imagery in constructing a photometric reconstruction term that further constrains the predicted depth to adhere to the scene geometry.

We have tested the approach both in a supervised and unsupervised setting. It should be noted that the difference between "supervised" and "unsupervised" in the KITTI benchmark is more quantitative than qualitative: the former has about 30% coverage in depth measurements, the latter about 5%. We show in Tab. 4.1 and 4.2 that our method achieves state-of-the-art performance in both supervised and unsupervised depth completion on the KITTI benchmark. Although we outperform other methods on score metrics that measures the deviation from the ground truth, we want to emphasize that our method does not simply produce a point estimate of depth, but provides a confidence measure, that can be used for more downstream processing, for instance for planning, control and decision making.

We have explored the effect of various hyperparameters, and are in the process of expanding the testing to real-world environments, where there could be additional errors and uncertainty due to possible time-varying misalignment between the range sensor and the camera, or between the two cameras when stereo is available, faulty intrinsic camera calibration, and other nuisance variability inevitably present on the field that is carefully weeded out in evaluation benchmarks such as KITTI. This experimentation is a matter of years, and well beyond the scope of this paper. Here we have shown that a suitably modified Conditional Prior Network can successfully transfer knowledge from prior data, including synthetic ones, to provide context to input range values for inferring missing data. This is important for downstream processing as the context can, for instance, help differentiate whether gaps in the point cloud are free space or photometrically homogeneous obstacles, as discussed in our motivating example in Fig. 4.1.

CHAPTER 5

Deformable Object Tracking

From now on, we assume the existence of an autonomous robot equipped with the perception of motion and depth. The next question we are going to ask is how could this robot "see" objects in the 3D scene. Here "seeing" the objects has two folds of meaning, first, if there is an object in the scene, how does it know which subset in the projected domain on the image corresponds to the object? Then, if it is given already the corresponding subset, how would the robot maintain a representation that enables itself to temporally track the object no matter how the scene changes? We leave the first part to the next chapter, and here we focus on precise shape tracking assuming that we know where the object is in the beginning.

In order to track object shape precisely, we propose a method to detect disocclusion in video sequences of three-dimensional scenes and to partition the disoccluded regions into objects, defined by coherent deformation corresponding to surfaces in the scene. We jointly infer occlusion and deformation fields that are piecewise smooth in a Sobolev framework where no explicit regularizer is needed. It then partitions the disoccluded region and groups its components with objects by leveraging on the complementarity of motion and appearance cues: Where appearance changes within an object, motion can usually be reliably inferred and used for grouping. Where appearance is close to constant, it can be used for grouping directly.

Persistent tracking of three-dimensional (3D) objects in video presents long-standing challenges unless they are flat [WA94a], or the video is short. As surfaces move in 3D relative to the viewer, previously unseen portions of the scene become visible and will have to be attributed to different objects to maintain tracking. Such *disocclusion* phenomena are the focus of our investigation. Persistent object tracking in video touches upon a large body of work in video segmentation [GKH10a, LKG11, XXC12, JG14], tracking [XBM04, BWS09, GCS04, LKH13, DCS13], optical flow [HS81, BA96, BBP04, ZPB07, BBM09], and motion segmentation [WA94a, SWS13a]. In dealing with visibility phenomena, our work relates to the literature on occlusion detection. There is a literature on detecting occluding boundaries from static images or short-baseline video (see [BSC00, SBM11] and references therein). Our work is related to [AS12a] that partitions the image domain into (flat) layers like [WA94a], but in a convex optimization setting after relaxing the ℓ_0 norm to ℓ_1. We detect occlusions without the need for such a relaxation and without the need for regularization of the deformation field, which can cause over-smoothing in some regions, and under-smoothing in others. Instead, following [YS15] we employ a Sobolev approach [SYM07] to infer deformation fields that are by construction smooth in a naturally coarse-to-fine manner. On a short time-scale, such deformation fields are related to optical flow, except for when the flow is explicitly partitioned into regions, as in *motion segmentation*. There, the flow field is often assumed to be piecewise parametric. Here we allow each component to be a generic diffeomorphism to handle articulated and deforming objects without over-segmenting them. Other motion segmentation approaches perform clustering of optical flow, often non-causally [OMB14b, GKH10a]. Taylor et al. [TKS15a] perform layer segmentation in longer video sequences leveraging occlusion cues, but do not explicitly address the interplay of motion and intensity cues in disocclusion. Similarly, [SWS13a] performs layered segmentation by grouping. Only intensity cues are used for the disocclusion in [CF13, YS15].

Consider a camera rotating around a box in Fig 5.1: Both the occluded and unoccluded regions involve portions of different objects, in this case just the box and the "background." Occlusions have been addressed by [SBM11, AS12a]. We focus on disocclusions, by determining the disoccluded area (Sect. 5.1), partitioning it and grouping each portion with an object (Sect. 5.2).

Grouping unseen portions of the scene into different objects requires prior assumptions on their properties. One could assume that the "appearance" or "texture" of objects is homogeneous (i.e. their reflectance exhibits spatially stationary statistics) and leverage on the similarity of image color histograms to partition and group disoccluded regions. How-

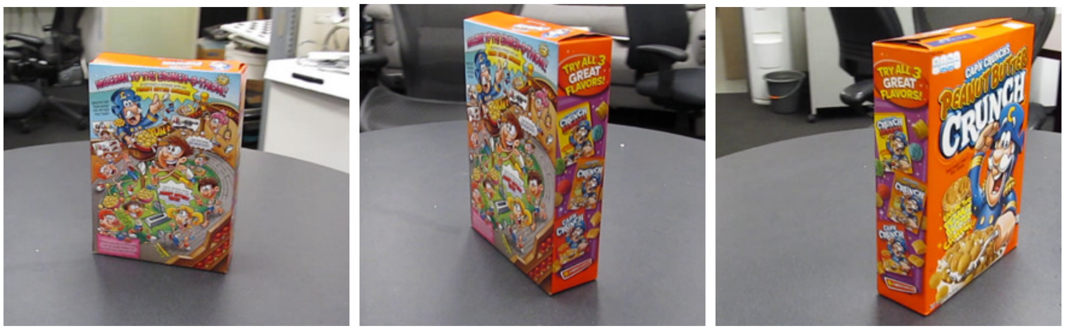


Figure 5.1: Relative motion between a three-dimensional scene and the camera (here rotating around the box) causes *disocclusion*, i.e. regions of the image domain where previously unseen portions of the scene project to. Unless objects in the scene are flat, the disocclusion include portions of different objects. Persistent tracking requires detecting the disocclusion and attributing their components to different objects.

ever, this assumption often fails, as in Fig. 5.1. Alternatively, one could assume that the "apparent motion" of objects is homogeneous (i.e. the deformation undergone by the image domain is smooth within objects, and discontinuous across). However, when objects exhibit "textureless" surfaces (i.e. constant reflectance), such a deformation is undetermined, and cannot be used for grouping.

Fortunately, motion and appearance cues are complementary: When one fails to be informative, the other may be. When the disoccluded region exhibits complex appearance, motion can be reliably inferred and exploited for grouping. Otherwise, when the disoccluded region is textureless, photometric statistics are spatially homogeneous and can be reliably used for grouping. Of course, both cues can fail if an object has a piecewise constant appearance, and the transition happens right at the disocclusion (Fig. 5.2). However, these are accidental phenomena that do not persist in long temporal sequences.

Here, objects are layouts of piecewise smooth and smoothly deforming surfaces in 3D supporting Lambertian reflection seen under constant illumination throughout a video sequence. There can be multiple objects moving independently, in addition to the viewer (or equivalently background) motion. Under these assumptions, the domain of a video image of a scene can be partitioned into two types of regions: Those that are *co-visible*, that un-

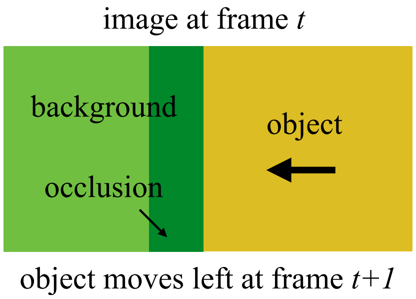


Figure 5.2: Illustration of an error that arises in segmentation by grouping pixels only based on motion residuals. The object (dark yellow) moves to the left to occlude a portion of the background (dark green). Pixels in the occluded region are likely to be classified incorrectly in frame t if only motion residuals are used since both residuals are large. When the background is constant in the occluded region and around it, classifying by residuals almost certainly leads to misclassifications.

der the stated assumptions are a smooth deformation of regions in the previous frame, and those that are *disoccluded*, i.e. whose pre-image under perspective projection is a portion of a surface that was not visible in the previous frame(s). In addition, *occluded* regions are subsets of a region that, in the previous frame, was occupied by an object different than the current one. These have been addressed by others [AS12a].

Disoccluded regions in a video are the occluded regions in the video played backwards. Because we eventually aim at real-time closed-loop operation we wish to process the data *causally*. Furthermore, parts of objects can appear in a frame and disappear in the next, a case which forward-backward sweeps would not address (Sect. 5.3). With an abuse of nomenclature, we refer to "objects" as both the connected surfaces in 3D, and the subsets of the (2D) image domain where they project. We extend the Sobolev framework of [YS15] to multiple objects to detect disocclusions. This framework naturally encompasses coarse-to-fine deformation inference without an explicit regularizer and the associated weighting constant (Sect. 5.1).

5.1 Sobolev Warps and Occlusions

We seek to partition the domain D of a time-varying color image $I_t : D \subset \mathbb{R}^2 \to \mathbb{R}^+$ for $t = 1, 2, \ldots,$ into a collection $\{R_i^t\}_{i=1}^N$ of *regions* R_i^t. We omit the time index hereafter for simplicity. These regions are also called "objects," that *move coherently*, as defined next.

The (apparent) motion of each region R_i, also referred to as *warp* or *deformation*, is defined in the domain of the image I_t as the map $w_i : R_i \to \mathbb{R}^2$ that transforms I_{t+1} back to I_t. Assuming the scene is Lambertian, illumination is constant, and the image is corrupted by additive zero-mean Gaussian noise, the maximum-likelihood estimate of w_i is obtained by minimizing $E_{\text{warp}}(w_i, O_i)$, given by

$$E_{\text{warp}} = \int_{R_i \backslash O_i} (I_{t+1}(w_i(x)) - I_t(x))^2 \, \mathrm{d}x + \beta \int_{O_i} \mathrm{d}x, \tag{5.1}$$

where $O_i \subset R_i$ is the (unknown) *occluded region* that is visible at time t but not at time $t+1$. Note that, although w_i is defined on all of R_i, the data I_{t+1}, I_t only provides evidence in the *co-visible* region $R_i \backslash O_i$. To avoid the trivial solution $O_i = R_i$ and thus w_i undetermined, we put a penalty on the occluded area as in [AS12a].

Eq. (5.1) is reminiscent of many *optical flow* estimation algorithms [HS81, BA96, BBP04, ZPB07], but there are important differences: First, each warp is restricted to a subset $R_i \subset D$ with no compatibility condition or relation among the different warps. Second, *there is no regularizer* for the warps. Most motion segmentation or optical flow schemes either assume that each warp belongs to a (small-dimensional) parametric family such as the group of affine transformations, or impose a penalty on the (piecewise) smoothness of w_i. Instead, we leverage on the Sobolev framework [SYM07] to impose regularity in a naturally coarse-to-fine framework, while allowing the warps to be arbitrary diffeomorphisms (smooth maps with a smooth inverse). So, rather than adding a regularizer for the warps in (5.1), we compute each warp as the integral of a smooth time-varying vector field that, at each instant, belongs to a Sobolev space. This allows us to efficiently optimize (5.1) without imposing global regularization, which may be too much for fine-scale objects, and too little for large ones.

Given the warp w_i, the optimal occlusion O_i is

$$O_i = \{x \in R_i \,:\, (I_{t+1}(w_i(x)) - I_t(x))^2 > \beta\}. \tag{5.2}$$

Substituting the expression above into the energy, we obtain

$$E_{\text{warp}}(w_i) = \int_{R_i} \rho(I_{t+1}(w_i(x)) - I_t(x))\,\mathrm{d}x, \tag{5.3}$$

which now depends only on the warp w_i, and where

$$\rho(x) = x^2 \text{ for } x^2 < \beta \quad \text{and} \quad \rho(x) = \beta \text{ for } x^2 \geq \beta \tag{5.4}$$

With this, we can finally clarify the the notion of "coherent motion" used to define the regions R_i: *A region R_i moves coherently if there is a warp w_i that is smooth according to the Sobolev metric, that minimizes* (5.3)*.*

The gradient of E_{warp}, $G_i : w_i(R_i) \to \mathbb{R}^2$, with respect to the Sobolev metric has been computed by [YS15] and is

$$G_i(x) := \nabla_{Sob} E(w_i)(x) = \text{avg}(F_i) + \frac{1}{\alpha}\tilde{G}_i(x), \tag{5.5}$$

where $\alpha > 0$ is a parameter that will be eliminated below, $F_i : w_i(R_i) \to \mathbb{R}^2$ is

$$F_i = \nabla I_{t+1} \nabla \rho(I_{t+1} - I_t \circ w_i^{-1}) \det \nabla w_i^{-1}, \tag{5.6}$$

$\text{avg}(F_i)$ is the average over $w_i(R)$, ∇ is the spatial gradient, and $\tilde{G}_i$ satisfies the partial differential equation (PDE):

$$\begin{cases} -\Delta \tilde{G}_i(x) = F_i(x) - \text{avg}(F_i) & x \in w_i(R_i) \\ \nabla \tilde{G}_i(x) \cdot N = 0 & x \in \partial w_i(R_i) \\ \text{avg}(\tilde{G}_i) = 0 & \end{cases}, \tag{5.7}$$

where Δ is the Laplacian, N is normal to $\partial w_i(R_i)$, $\tilde{G}_i$ is the *deformation*, and $\text{avg}(F_i)$ is the *translation*.

To extend the framework to multiple regions, we extend each warp w_i to the entire domain D by imposing $\Delta \tilde{G}_i(x) = 0$ for $x \in D \backslash R_i$ and a Dirichlet condition on ∂R_i. The extension is continuous, but not differentiable across R_i.[1]

Starting with the identity map $w_i(x) = x$, we deform it by the gradient descent (5.5) as follows. Define $\phi_i^{0,\tau} : D \to D$ and $\phi_i^{\tau,0} : D \to D$ as the evolving warp and its inverse where τ is an artificial time variable parameterizing the evolution. The inverse is needed to compute F_i. The evolution of the warps according to the gradient descent of E_{warp} is

$$G_i^\tau = \nabla_{Sob} E_{\text{warp}}(\phi_i^{0,\tau}), \tag{5.8}$$

$$\partial_\tau \phi_i^{\tau,0}(x) = \nabla \phi_i^{\tau,0}(x) \cdot G_i^\tau(x), \tag{5.9}$$

$$\partial_\tau \phi_i^{0,\tau}(x) = -G_i^\tau(\phi_i^{0,\tau}(x)) \tag{5.10}$$

for all $x \in D$. This gives a coarse-to-fine evolution. One can eliminate the parameter α by noting the independence of the deformation and translation components on α in (5.5). This gives Algorithm 1, which decreases the energy.

Algorithm 1 Sobolev Warp Computation

```
Set $\phi_i^{\tau,0}(x) = \phi_i^{0,\tau}(x) = x$ for $\tau = 0$
repeat
    repeat
        Let $\alpha \to \infty$ so $G_i^\tau = \text{avg}(F_i^\tau)$ is a translation
        Translate: Perform one iteration of (5.9)-(5.10)
    until avg($F_i^\tau$) = 0.
    Deform: Do one iteration of (5.9)-(5.10) with $G_i^\tau = \tilde{G}_i^\tau$
until $\tilde{G}_i^\tau = 0$
Set $w_i = \phi_i^{0,\tau_\infty}$ where $\tau_\infty$ is the convergence time
```

In the next section, we will need to compute the occlusion so that it can be removed in

[1] While one can define the Sobolev metric over the entire domain D [BMT05], thus naturally having a regular gradient defined over the entire domain D, this is avoided to enable capturing fine-scale structures in a manner that is not influenced by neighboring large-scale structures, for instance an arm swinging near the torso of a person.

the next frame. It can be computed at the end of the evolution as

$$O_i^{\tau_\infty} = \{x \in R_i \,:\, (I_{t+1}(\phi_i^{0,\tau_\infty}(x)) - I_t(x))^2 > \beta\}. \tag{5.11}$$

5.2 Causal Shape Tracking

If the motion of each region R_i was reliably inferred, one could attempt to propagate forward the R_i to segment the next frame. Unfortunately, regions that become *disoccluded* between t and $t+1$ are not included in any of the R_i. While this is not a major problem if we are interested in only two adjacent frames, t and $t+1$, as the area of the occluded/disoccluded regions is small, as time goes by the disocclusion typically grows. The challenge becomes to assign the various components of the disocclusion to regions. This is illustrated in Fig. 5.1: So long as the scene is populated by *non-flat* surfaces, multiple objects contribute to the disoccluded region.

We assume a partition into objects at time $t-1$ and propagate it forward to time t. The disocclusion, i.e., the part of the domain D not covered by the propagated segmentation, is initially assigned to regions based on estimated warps, and this is refined by minimizing the energy in Section 5.2.1.

5.2.1 Complementarity of motion and appearance

Of course both appearance and motion cues are obtained from image irradiance. What we mean by "cues" is bottom-up computation that leverages on the assumption of smooth spatial variation of image irradiance (appearance cues) versus smooth temporal variation of the same (motion cues).

To attribute disoccluded regions to any of the existing objects, we can leverage the photometric regularity and assign each segment to the object that has similar "texture" or motion. We favor the latter, as objects can have spatially-varying appearance, as in the cereal box in Fig. 5.1. This fails when the object and the background are textureless, as in Figure 5.2, or when they exhibit similar fine-scale texture. However, in this case grouping by appearance is straightforward. We leverage on this complementarity by exploiting preferentially motion regularity, consistent with our definition of objects, resorting to appearance regularity when the photometry is not suitable to reliably estimate motion.

Textureless regions: To leverage on this complementarity, we use the local standard deviation $\sigma_i(x)$ of I_t in a neighborhood $B_{x,r'} \cap R_i$ where $B_{x,r'} = \{y \in D : |x - y| \leq r'\}$ is the ball of radius r' centered at point x. We can then define a measure of local constancy of any region local to a point x as the minimum standard deviation over all regions that intersect the ball:

$$\underline{\sigma}(x) = \min_{i,\, B_{x,r'} \cap R_i \neq \emptyset} \sigma_i(x). \tag{5.12}$$

Low values of $\underline{\sigma}(x)$ indicate that the underlying intensity is not *sufficiently exciting* and therefore motion estimates can be expected to be unreliable.

Motion ambiguity function: Grouping by residuals also should not be done when current warp residuals are large. Define the forward, backward and minimum residuals as

$$\mathrm{Res}_i^f(x) = (I_{t+1}(w_i^f(x)) - I_t(x))^2 \tag{5.13}$$

$$\mathrm{Res}_i^b(x) = (I_t(w_i^b(x)) - I_{t-1}(x))^2 \tag{5.14}$$

$$\mathrm{Res}_i(x) = \min\{\mathrm{Res}_i^f(x), \mathrm{Res}_i^b(x)\} \tag{5.15}$$

where w_i^f and w_i^b are the current forward and backward warps of region R_i. The backward residual is used to remove some ambiguity in Fig. 5.2 as sometimes occluded pixels at time $t + 1$ are visible at time $t - 1$, and hence the backward motion is reliable. The minimum of Res_i over all regions that intersect with a ball around x,

$$\underline{\mathrm{Res}}(x) = \min_{i,\, B_{x,r'} \cap R_i \neq \emptyset} \mathrm{Res}_i(x), \tag{5.16}$$

is small when motion cues are reliable. We define the *motion ambiguity function*, $\mathrm{maf} : D \to \{0, 1\}$, which indicates whether motion cues are unreliable, as

$$\mathrm{maf}(x) = \begin{cases} 1 & \text{if } \underline{\sigma}(x) < k/r' \text{ or } \underline{\mathrm{Res}}(x) > \beta \\ 0 & \text{otherwise} \end{cases}, \tag{5.17}$$

where $k > 0$ is a parameter, the sensitivity to which is studied empirically in Sect. 5.3. maf is 1 if the pixel is in or borders a constant region or if all the current motion residuals are large.

Complementary data term: The cost for $x \in R_i$ is

$$f_i(x) = (1 - \text{maf}(x))\text{Res}_i(x) - \text{maf}(x) \log p_{i,x}(I_t(x)), \tag{5.18}$$

where $p_{i,x}$ are local normalized histograms [BC09] of the image I_t within the region R_i. Therefore, if the motion is reliable, as defined by the maf, the cost is the residual of the pixel in the region and if the motion is unreliable, the cost is the fidelity of pixel to the local intensity distribution of the region R_i. The data energy for region R_i is then:

$$E^i_{\text{data}} = \int_{R_i} f_i(x)\,\mathrm{d}x. \tag{5.19}$$

This complementary data term is a key feature in resolving disocclusions (Fig. 5.3).

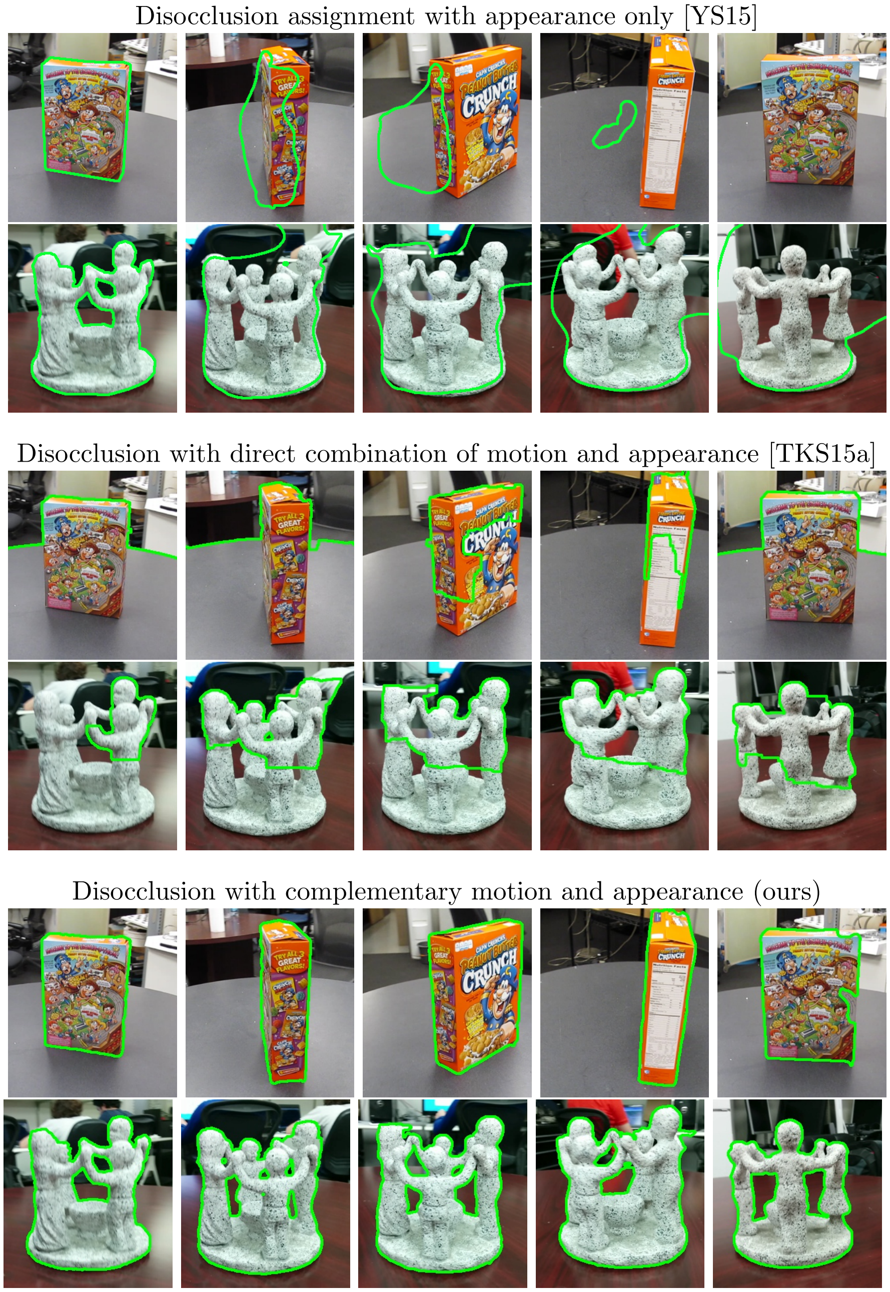
5.2.2 Temporal and Spatial Regularity

To leverage on temporal and spatial regularity of the regions, we first note that the warps are regular by construction within the Sobolev framework. We also note that, in between frames, disoccluded regions are small, adjacent to the object they belong to, and typically result in an updated region of similar shape. Thus, if R'_i is the forward warping of the i^{th} region from frame t to $t+1$, we bias the final regions R_i to be close to R'_i in shape and location.

To this end, we construct a local shape similarity prior. Measuring the similarity of R_i and R'_i generally requires knowledge of point correspondences. Similar to ICP [BM92], we assume that $x \in R_i$ corresponds to its closest point in R'_i, $\text{cl}_i(x)$, which can be computed efficiently with Fast Marching [Set96]. Define the local shape similarity, $S_i : R_i \to \mathbb{R}^+$, of R_i within the ball $B_{r,x}$ to R'_i within $B_{r,\text{cl}_i(x)}$ as follows:

$$S_i(x) = \frac{1}{|B_{x,r}|} \int_{B_{x,r}} |\mathbf{1}_{R_i}(y) - \mathbf{1}_{R'_i}(\text{cl}_i(x) - x + y)| \mathrm{d}y, \tag{5.20}$$

where $\mathbf{1}_R$ is the indicator function of R, and $|B_{x,r}|$ is the area of $B_{r,x}$. See Figure 5.4. The score measures the difference between the shapes $R_i \cap B_{x,r}$ and $R'_i \cap B_{\text{cl}_i(x),r}$ using translation



Disocclusion assignment with appearance only [YS15]

Disocclusion with direct combination of motion and appearance [TKS15a]

Disocclusion with complementary motion and appearance (ours)

Figure 5.3: Rotating around an object. Disoccluded parts of an object that have different appearance than the visible parts in the previous frame (cereal box) pose difficulties to existing algorithms. Labeled above are various strategies for addressing disocclusions. To show that our method performs even under self-similar appearance, we have included the statue.

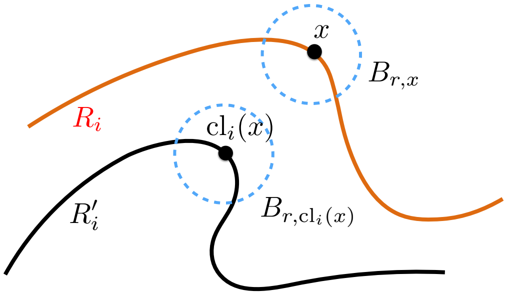


Figure 5.4: Illustration of the quantities in the local shape similarity term, S_i. R_i' is the forward warped region and R_i is a candidate in frame $t+1$. The region R_i in a ball around x is compared to R_i' in a ball around $\mathrm{cl}_i(x)$, the closest point on R_i' to x to from $S_i(x)$.

invariant set symmetric difference. The shape similarity energy is:

$$E^i_{\text{shape}} = \int_{R_i} S_i(x)\,\mathrm{d}x. \tag{5.21}$$

In addition, to bias regions R_i towards being close to R_i', let $d_{R_i'}$ denote the distance function to $\partial R_i'$, and define

$$E^i_{\text{dist}} = \int_{R_i} d_{R_i'}(x)\,\mathrm{d}x. \tag{5.22}$$

Finally, we induce spatial regularity of R_i, i.e., nearby points x and y are penalized if they do not belong to the same region. Let

$$W_{R_i} = G_s * (1 - \mathbf{1}_{R_i}) \tag{5.23}$$

be a Gaussian smoothing of standard deviation s of the complement of the indicator function of R_i [ET06]. A large value of $W_{R_i}(x)$ implies that $x \in R_i$ is near many points of $D \backslash R_i$. We induce spatial regularity of R_i by

$$E^i_{\text{smooth}} = \int_{R_i} W_{R_i}(x)\,\mathrm{d}x. \tag{5.24}$$

5.2.3 Overall Model and Optimization Method

The assumptions underlying our model are captured by the following energy, which is minimized with respect to the regions R_i:

$$E_{\text{seg}} = \sum_{i=1}^{N} E^i_{\text{data}} + \gamma_{ls} E^i_{\text{shape}} + \gamma_d E^i_{\text{dist}} + \gamma_s E^i_{\text{smooth}}, \tag{5.25}$$

where $\gamma_{ls}, \gamma_d, \gamma_s > 0$ are weights. We optimize the energy above by a first order approximation to the gradient descent, ignoring terms that involve integrals over R_i. They could be easily included, at a high computational cost and modest performance gain. By defining

$$H_i(x) = f_i(x) + \gamma_{ls} S_i(x) + \gamma_d d_{R_i'}(x) + \gamma_s W_{R_i}(x), \tag{5.26}$$

we arrive at our optimization scheme in Algorithm 2.

Algorithm 2 Assigning Disocclusion to Regions

```
 1: // initialize $R_i$ for gradient descent
 2: Compute propagation of segmentation, $R_i'$ using (5.27)
 3: Compute disocclusion $\mathbf{D} = D \setminus \cup_i R_i'$
 4: Compute warps of $R_i'$ using Algorithm 1
 5: Compute $H_i$ by substituting $R_i$ with $R_i' \cup \mathbf{D}$
 6: Set $R_i = R_i' \cup \{x \in \mathbf{D} \, : \, H_i(x) \le H_j(x), \, \forall j\}$
 7: // end initialize
 8: **repeat** // first order approximation of gradient descent
 9:     Update warps of $R_i$ using Algorithm 1
10:     Compute $H_i$
11:     $R_i^{\text{new}} = \{x \in D \, : \, d_{R_i}(x) < \varepsilon, \, H_i(x) \le H_j(x), \forall j\}$
12:     Update regions by $R_i = R_i^{\text{new}}$
13: **until** $R_i$'s do not change between iterations
```

Algorithm 2 first computes an initialization of regions R_i to the gradient descent (lines 2-6). This is accomplished by propagating forward the segmentation at time $t-1$ to t:

$$R_i' = \{x \in D \, : \, \mathbf{1}_{R_i^t \setminus O_i^t}(w_i^{-1}(x)) \ge \mathbf{1}_{R_j^t \setminus O_j^t}(w_j^{-1}(x)), \, \forall j\} \tag{5.27}$$

where $O_i^t \subset R_i^t$ is the part of the i^{th} region that is occluded at frame t (5.11), which is removed, and w_i is the warp from $t-1$ to t. R_i' does not partition all of D because of disocclusion. Therefore, the disoccluded region $\mathbf{D} = D \setminus \cup_i R_i'$ is initially assigned based on motion cues computed from R_i' and other terms in H_i.

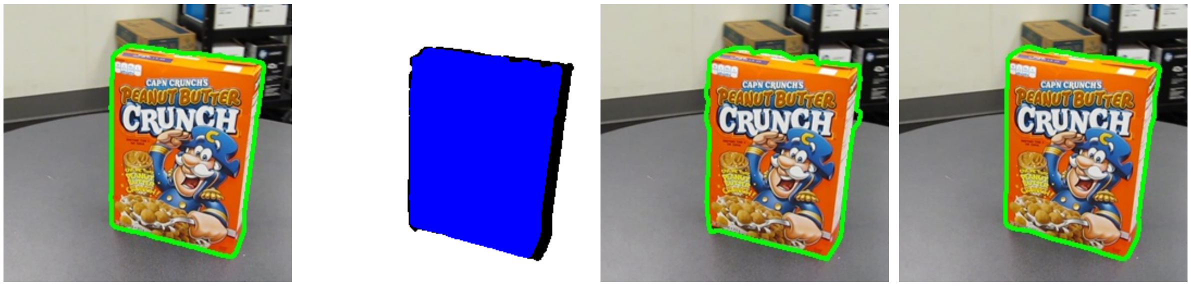


Figure 5.5: [Left]: Segmentation from the frame t. [Middle, left]: the propagation of the segmentation from frame t to $t+1$ (black regions indicate disoccluded regions). [Middle, right]: initialization of the regions. [Right]: final segmentation.

With this initialization, the first order approximation to the gradient descent is computed (lines 9-12). Note that the condition, $d_{R_j}(x) < \varepsilon$, is to allow pixel changes only within a band of the boundaries of the current regions so as to approximate the gradient descent. Each step of the warp computation (from t to $t+1$ and from t to $t-1$) in line 9 requires only a few iterations in Algorithm 1 since the warps in the previous iteration of line 9 are close to the final. See Fig. 5.5 for an example of various stages of this method.

5.2.4 Initialization for the First Frame

So far we have assumed that, at time t, we have a partition at time $t-1$. This is the case during regime operation when processing a video sequence, but not when $t = 0$. For certain applications, such as interactive video segmentation [BWS09, BWS10], one can assume that the user provides an initial partition. More in general, a number of methods could be employed to obtain an initial partition, using a variety of cues, including semantic labeling from trained detectors. While this process may be costly, it only needs to be performed once as our method affords us the ability to correct initial errors based on motion and appearance regularity.

In the next section, we present results for an initialization performed by clustering optical flow (with regularity (5.24) using Classic-NL [SRB10]) during a longer initial temporal segment, until enough motion is observed (see Fig. 5.6).

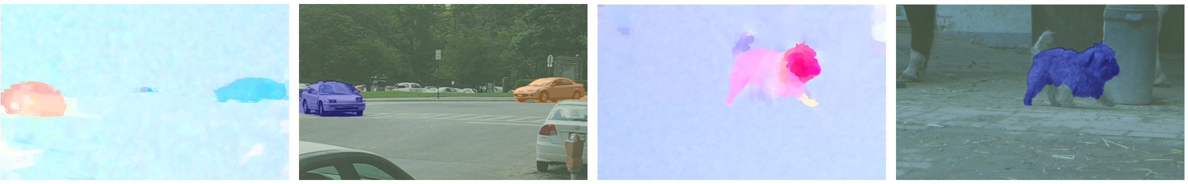


Figure 5.6: Illustration of initialization method in the first frame. [Left]: Aggregation of optical flow fields, [Right]: initial segmentation in the first frame.

5.3 Experiments

Our algorithm aims to track *objects* in precise shape, thus we test it on benchmarks with ground truth object annotation: the Freiburg-Berkeley Motion Seg. (FBMS-59) [OMB14b], and SegTrack (v1 & v2) [TFN12, LKH13]. FBMS-59's two sets - training (29 sequences) and test (30 sequences), range between 19-800 frames with multiple objects. SegTrack v2 consists of 14 sequences ranging from 29-279 frames with multiple objects. SegTrack v1 is an earlier version with single objects, which we use to expand the comparison to more methods.

Evaluation: FBMS-59 scores a subset of frames (3-41). Results are reported in terms of precision, recall, F-measure, and the number of objects with $F \geq 0.75$. SegTrack (v1 & v2) evaluates, on all frames, the number of pixels incorrectly classified. Results are reported as average intersection over union overlap.

Comparisons: On FBMS-59, we compare against a baseline approach [GKH10a], one based on clustering motion tracks [OMB14b], one segmenting based on occlusion, motion and appearance cues [AS12a], and finally a most recent one integrating motion, appearance, occlusion, and temporal regularity [TKS15a]. On SegTrack, we compare to [CF13] that attempts to solve disocclusions using only appearance and to other state-of-the-art methods [LKH13, LKG11, ML12, JG14, WDL15].

Initialization: On FBMS-59, we report results of our method automatically initialized as described in Sect. 5.2.4. On SegTrack our method is initialized by the user in frame 1 and compared with similarly initialized methods and also automated methods. Typically, sequences in SegTrack do not have enough object motion in the first few frames to ensure proper initialization.

	Training set (29 sequences)				Test set (30 sequences)			
	P	R	F	$N/65$	P	R	F	$N/69$
[GKH10a]	79.17	47.55	59.42	4	77.11	42.99	55.20	5
[OMB14b]	81.50	63.23	71.21	16	74.91	60.14	66.72	20
[AS12a]	87.20	59.60	70.81	17	79.64	50.73	61.98	7
[TKS15a]	85.00	67.99	75.55	21	82.37	58.37	68.32	17
[TKS15a]-NC	83.00	70.10	76.01	23	77.94	59.14	67.25	15
ours	**89.53**	**70.74**	**79.03**	**26**	**91.47**	**64.75**	**75.82**	**27**

Table 5.1: FBMS-59 results. Average precision (P), recall (R) and f-measure (F) over all sequences in the training and test datasets of FMS-59. Higher values indicate superior performance. All methods are fully automatic. Methods [TKS15a] and our method are causal. The other methods process the whole video in batch.

Parameters: For FBMS-59, we tune the parameters on a few sequences in the training dataset, and then fix them over both training and test datasets. On SegTrack, we fix parameters over all sequences to the same values. Sensitivity of key parameters is addressed later.

Results on FBMS-59: Are in Table 5.5. Figure 5.7 shows some representative outcomes. Overall our method is more accurate in all measures, even compared to non-causal (NC) methods that process the video in batch. This suggests that good disocclusion is key to accurate object segmentation.

Failure Cases on FBMS-59: The main source of error is the automatic initialization in frame 1. This could be mitigated by running our method on multiple candidate initializations, although initialization is not our focus here. To show that better initialization would resolve failures, we show that the results of the 10 most inaccurate cases (typically when an object failed to be detected) improves with user annotation in the first frame (Tab. 5.2, Fig. 5.8). Fig. 5.9 shows that our method recovers from errors in the first frame (short of failed detection).

Forward-Backward Sweeps on FBMS-59: Although disocclusions are backward-

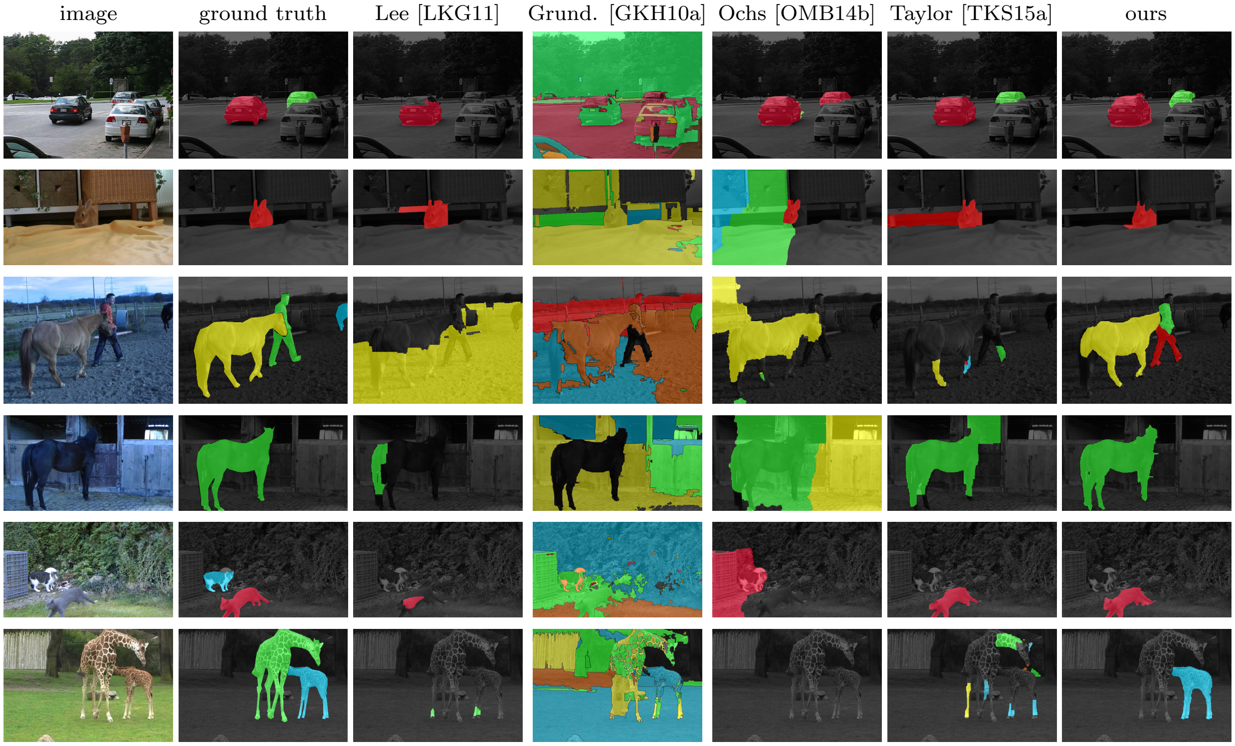


Figure 5.7: Sample Visual Results on FBMS-59. Comparison of various state-of-the-art methods. Only a single frame on various sequences are shown. Failure cases (bottom two) in our method typically arise when not enough motion is present in the first few frames.

	marple9	cats4	farm1	goats1	giraffes1	all
ours (auto)	0.7950	0.7723	0.6730	0.6166	0.7515	0.7217
ours (manual)	0.9782	0.9025	0.7519	0.7505	0.9255	0.8617

Table 5.2: Failure cases on FBMS-59 in Fig. 5.7 can be enhanced with user annotation in the first frame. Thus, the main source of error in our method is the initialization. Results are in terms of F-measure.

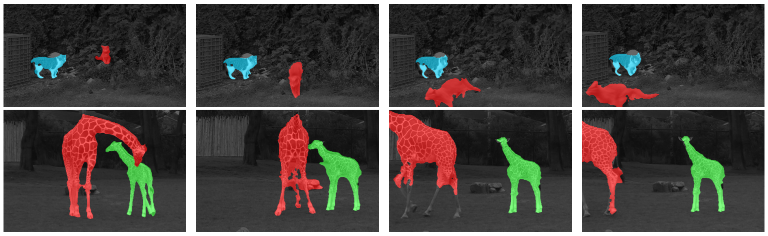


Figure 5.8: Sample failure cases (various frames) on FBMS-59 in Fig. 5.7 are enhanced with user annonation in the first frame.

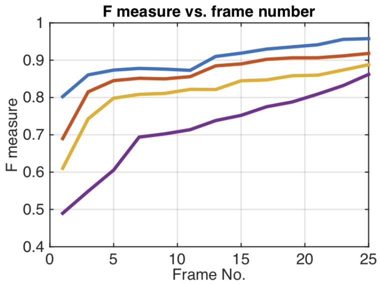


Figure 5.9: Results (on FBMS-59) with four different levels of errors, in initialization. Errors are mitigated in subsequent frames.

	human	ours	[WDL15]	[JG14]	[CF13]	[ML12]	[LKG11]
Mean	347	**409**	535	874	455	677*	740*
Birdfall	130	**144**	163	189	265	189	288
Cheetah	308	623	806	1170	**570**	806	905
Girl	762	**835**	1904	2883	841	1698	1785
Monkeydog	306	**252**	342	333	289	472	521
Parachute	299	**169**	275	228	310	221	201
Penguin	279	**429**	571	443	456	-	136285

Table 5.3: SegTrack v1 results. Evaluation is performed in terms of the number of pixels classified incorrectly; smaller values indicate superior results. Note that our method, [WDL15], [JG14], and [CF13] use user annotation in frame 1, and [ML12], [LKG11] do not.

occlusions, addressed extensively in the literature [SBM11, AS12a], computing disocclusions via forward-backward sweeps followed by a grouping procedure does not perform as well as our method. We compare to the non-causal version of [TKS15a], consisting of one forward and one backward pass. Then, advanced grouping is performed based on motion, appearance, temporal continuity, and constraints imposed by occlusions/disocclusions. The result, labeled [TKS15a]-NC in Tab. 5.5, is worse than ours on all measures. This reaffirms that forward-backward sweeps is not an adequate approach to resolve disocclusions.

Results on SegTrack: Tab. 5.3. We let the user annotate the first frame, as in [JG14, CF13, WDL15]. Our method outperforms all others on all but one sequence. That our method outperforms [CF13] reaffirms our that exploiting complementary motion and appearance cues is beneficial. Results on v2 (Tab. 5.4, Fig. 5.10) show that our method out-performs fully automated ones but also those using user annotation.

	ours	[WDL15]	[LKH13]	[LKG11]	[GKH10a]
Mean per object	**76.4**	71.8	65.9	45.3	51.8
Mean per sequence	**77.0**	72.2	71.2	57.3	50.8
Girl	**91.6**	84.6	89.2	87.7	31.9
Birdfall	77.3	**78.7**	62.5	49.0	57.4
Parachute	96.1	94.4	93.4	**96.3**	69.1
CheetahDeer	62.4	**66.1**	37.3	44.5	18.8
CheetahCheetah	**52.2**	35.3	40.9	11.7	24.4
Monkeydog-Monkey	**84.1**	82.2	71.3	74.3	68.3
Monkeydog-Dog	**43.7**	21.1	18.9	4.9	18.8
Penguin1	94.0	**94.2**	51.5	12.6	72.0
Penguin2	82.1	**91.8**	76.5	11.3	80.7
Penguin3	78.4	**91.9**	75.2	11.3	75.2
Penguin4	86.3	**90.3**	57.8	7.7	80.6
Penguin5	**77.1**	76.3	66.7	4.2	62.7
Penguin6	**89.0**	88.7	50.2	8.5	75.5
Drifting Car1	**82.3**	67.3	74.8	63.7	55.2
Drifting Car2	**77.6**	63.7	60.6	30.1	27.2
Hummingbird1	39.0	**58.3**	54.4	46.3	13.7
Hummingbird2	69.0	50.7	72.3	**74.0**	25.2
Frog	**76.7**	56.3	72.3	0	67.1
Worm	83.4	79.3	82.8	**84.4**	34.7
Soldier	**84.0**	81.1	83.8	66.6	66.5
Monkey	85.1	**86.0**	84.8	79.0	61.9
Bird of Paradise	**96.1**	93.0	94.0	92.2	86.8
BMXPerson	**92.8**	88.9	85.4	87.4	39.2
BMXBike	32.5	5.70	24.9	**38.6**	32.5

Table 5.4: SegTrack v2. The evaluation is performed in terms of the overlap of the best segments; larger values indicate superior results. Our method and [WDL15] uses user annotation in frame 1.

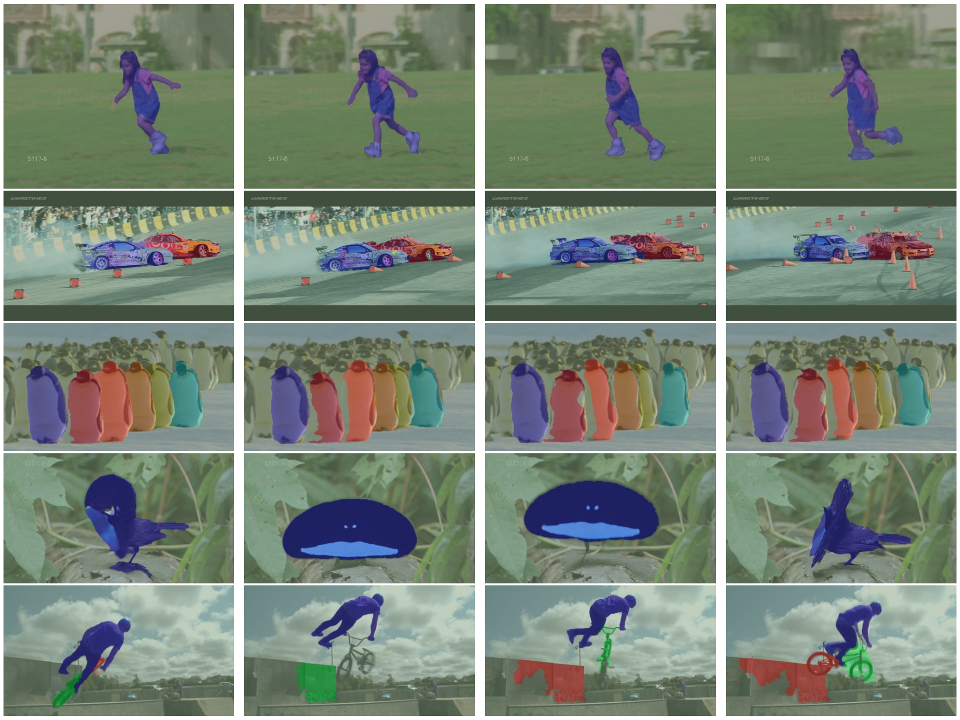


Figure 5.10: Sample SegTrack v2 results of our method.

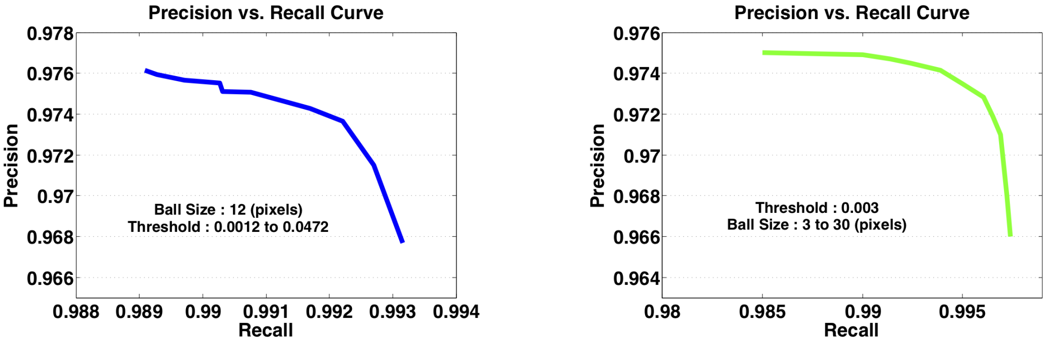


Figure 5.11: Analysis of sensitivity of key parameters (the threshold and ball size of the textureless detector). [Left]: ROC curve fixing the ball size and varying the threshold. [Right]: ROC curve fixing the threshold and varying the ball size.

Sensitivity to Key Parameters: These include the ball size r' and the threshold parameter k in our textureless region detector (5.12) and (5.17). To this end, we plot PR curves (measured in terms of correct/incorrectly classified pixels) by fixing one parameter and varying the other and vice-versa. Results (5.11) on the cereal box and statue sequences show that within the operating range, precision does not drop much as recall is increased.

Computational cost and implementation: The costliest component is solving for the warps. This requires solving a linear PDE, for which there are many available fast-solvers that could be leveraged. We used conjugate gradient, which can be sped up. The overall cost of our algorithm varies with the amount of deformation between frames. Using a 3.1GHz 12-core processor, processing one frame on FBMS-59 takes on average 30 secs.

5.4 Integrating Tracking and Detection

In the previous sections, we propose a method for handling occlusion and disocclusion in object tracking that does not require explicit motion regularization, operates naturally in a coarse-to-fine framework, and leverages complementary motion and appearance cues. However, it assumes that a current estimate of the partition into objects is given at time t to infer the same at t+1. If the given partition is nonsensical, most likely so will be the output of our inference scheme. This issue is particularly cogent at time t = 0. It can be addressed by spawning multiple trackers corresponding to different initialization hypotheses, later aggregating them through a voting scheme.

5.4.1 Methodology

Our goal is to partition a video $I_t : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ for $t = 1, 2, \ldots$, where Ω is the image domain, into regions $\{R_t^i\}_{i=1}^{N_t}$, corresponding to (projections of partially) visible objects onto the current frame, the number of which can vary over time. We call a point-estimate of these regions based on prior image measurements the "objects state," which is maintained by a tracker. We proceed recursively: At each instant t and given measurements up to t, the objects state, $R^i_{t|t}$ (Fig. 5.12, row 1), serves to *predict* the regions at the next instant of time, $R^i_{t+1|t}$ (Fig. 5.12, row 2). In the meantime, a *pseudo-measurement* module (object detector) processes the next frame I_{t+1} or small batch of frames (for us, 3) to produce object hypotheses or "proposals" P^j_{t+1} (Fig. 5.12, row 3) used to *update* the current estimate of the existing regions, $R^i_{t+1|t+1}$, $i = 1, \ldots, N_t$, as well as to detect new objects not currently represented in the state of the tracker (marked in orange in Fig. 5.12). This update, described in Sect. 5.4.2, also ensures that visibility constraints are satisfied, so each point on the image I_{t+1} corresponds to a single object, which is equivalent to enforcing opacity of objects in the scene. Initially, we assume that the scene is empty, $N_0 = 0$.

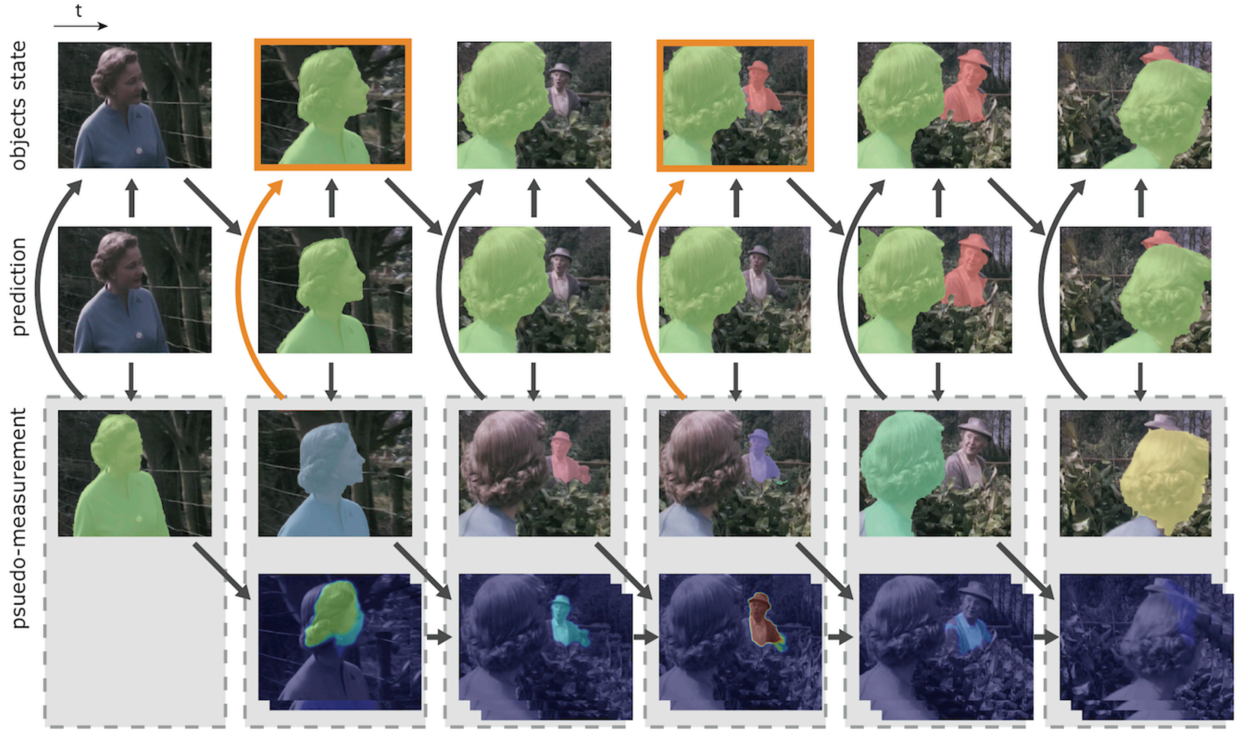


Figure 5.12: Illustration of our framework causally processing a video. The pseudo-measurement (object detector) (bottom 2 rows) proposes regions in the image that correspond to objects in the scene, which are integrated with the prediction (row 2) into the final object labeling (row 1). We begin with no objects in the state (left column) and an object is only added to the state after being detected across multiple frames (highlighted by orange arrows).

Prediction: Given the current state of objects $\{R^i_{t|t}\}_{i=1}^{N_t}$, we predict their next state $\{R^i_{t+1|t}\}_{i=1}^{N_{t+1}}$ by warping the current regions with ω^i_t, defined on the regions R^i_t. The occluded portion of each region O^i_t must be predicted and removed, while the disoccluded parts D^i_{t+1} must be estimated and added, yielding $\{R^i_{t+1|t} = \omega^i_t(R^i_{t|t} \setminus O^i_t) \cup D^i_{t+1}\}\forall i$. We employ deformable shape tracker developed in previous sections that automatically infers disocclusions [YSS15a]. Since each object is predicted *independently*, the update phase must manage conflicts among different objects predicted to overlap in the next image, in order to make our method viable.

Object detection: The pseudo-measurement module provides object proposals $\{P_{t+1}^j\}, j = 0, 1, 2, \ldots$ using the image at time I_{t+1} or a small batch of images (short-term memory; long-term memory is represented by the objects state), to produce hypotheses of objects informed by occlusion relations. These exploit assumptions of spatial connectedness of objects as well as the topology of the scene. Object proposals also spawn new objects that appear and are not attributed to existing predicted ones. Occlusions have been used extensively to prime object detection or layer segmentation [WA94a, BM98, AS12b]. Again, we employ [AS12b] as our pseudo-measurement module.

To be robust to gross failure of the pseudo-measurement module, we require objects to be detected in multiple frames before they are added to the objects state in the tracker. This is useful when the motion or the scene is very complex (see Fig. 5.13 for an example where multiple horses are moving independently), as occlusion detection and subsequently object detection from occlusions can be considerably noisy in these cases. This reduces the number of false positives in our output and helps to keep the computational cost of prediction, which scales linearly with the number of objects tracked, manageable.

Thus, object proposals are accumulated over time before being declared as a true object. For the accumulation, each proposal $P_t^j, j = 1, 2, \ldots, M_t$ is associated with a confidence score as follows:

$$s(P_t^j) = \frac{\int_{\partial P_t^j} e(x)dx}{|P_t^j|} \tag{5.28}$$

where $|P_t^j|$ is the area of proposal P_t^j and $e(x)$ measures the intensity and motion discontinuity strength, defined as

$$e(x) \doteq \max\{c_1 e_{image}(x) + c_2 e_{motion}(x), \varepsilon\}, \tag{5.29}$$

where e_{image} and e_{motion} are the output of an edge detector on the image data and motion field, respectively, and $0 < \varepsilon \ll 1$. We then associate each proposal $P_{t+1}^j, j = 1, 2, \ldots$ from the current frame with a warped proposal $\omega_t(\bar{P}_t^k), k = 1, 2, \ldots, M_t$ from the past if P_{t+1}^j and

$\omega_t(\bar{P}_t^k)$ achieve a sufficient overlap as measured by the intersection over union score:

$$IU(P_{t+1}^j, \omega_t(\bar{P}_t^k)) = \frac{P_{t+1}^j \cap w_t(\bar{P}_t^j)}{P_{t+1}^i \cup w_t(\bar{P}_t^k)} \tag{5.30}$$

Here we abuse the notation and let ω_t refer to the warp of the entire image domain Ω from time t to $t+1$, which is computed during prediction. Each matched proposal contributes its shape and score to the accumulated proposal. We define the running score or likelihood of $\bar{P}_{t+1}^k$ recursively as

$$p_{t+1}^k(x) = s(P_{t+1}^j)\mathbb{I}(P_{t+1}^j) + p_t^k(w_t(x)), \tag{5.31}$$

where $\mathbb{I}(P_{t+1}^j) = \{x | x \in P_{t+1}^j\}$, the indicator function of the proposal, and $p_\tau^k(x) = s(P_\tau^k)\mathbb{I}(P_\tau^k)$ when the proposal was first added into the proposal pool at time τ. Finally, proposals with p_{t+1}^k exceeding a nominal threshold $\lambda = 0.5$ are promoted to objects added to the objects state maintained by the tracker as

$$R_{t+1}^k = \{x, p_{t+1}^k(x) > \lambda\} \tag{5.32}$$

and removed from the proposal pool.

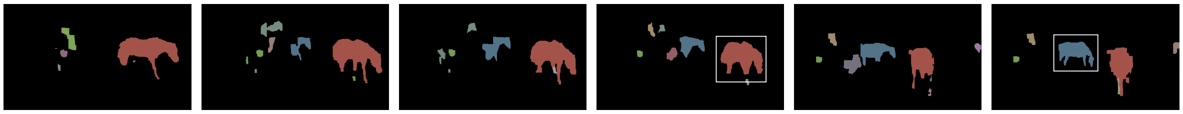


Figure 5.13: Object detections from our pseudo-measurement module in six nonconsecutive but chronological frames from a video of horses. In this difficult scenario containing multiple deforming objects with complex motion dynamics, object detection generates many false positives. In this work, we trust consistently appearing regions to correspond to true objects in the scene. The accumulation of evidence from detections identifies the red segment (by the 4^{th} frame) and the blue segment (by the 6^{th} frame) as real objects and allows us to ignore the remaining proposals as noisy measurements.

5.4.2 Update

At time $t+1$, given a set of object predictions $\{R_{t+1|t}^i = \omega_t^i(R_{t|t}^i \setminus O_t^i) \cup D_{t+1}^i\}, i = 1, 2, \ldots, N_t$ and object proposals from the pseudo-measurement module $\{P_{t+1}^j\}, k = 1, 2, \ldots,$ we wish to update the object state $\{R^i_{t+1|t+1}\}, i = 0, 1, 2, \ldots, N_{t+1}$. Notice that the number of objects could be different than number predicted.

Updating each object region: For each proposal P^j_{t+1}, we first try to assign it to one of the existing objects $R^i_{t+1|t}$ if $\text{IU}(P^j_{t+1}, R^i_{t+1|t}) > \tau, \tau \in (0, 1)$. This would suggest that the proposal comes from an already-present object $R^i_{t|t+1}$, and P^j_{t+1} and $R^i_{t+1|t}$ are then fused to generate the estimate of $R^i_{t+1|t+1}$. Let ∂P^j_{t+1} and $\partial R^j_{t+1|t}$ be the boundaries to the proposal and predicted object respectively, we define weight functions of pixel location x, with respect to the corresponding boundary,

$$g_{\partial P^j_{t+1}}(x) = \exp\left(-\frac{d^2_{\partial P^j_{t+1}}(x)}{\delta^2}\right) \tag{5.33}$$

where $d_{\partial P^j_{t+1}}$ is the distance function of the boundary ∂P^j_{t+1}, and $\delta > 0$. $g_{\partial R^i_{t+1|t}}(x)$ is defined similarly for $R^i_{t+1|t}$. Additionally, we define

$$Coh(x) \doteq \max\{\sum_t \mathbb{I}_{\cup_{P^j_t}}(\omega_t^{-1}(x)), 1\} \tag{5.34}$$

where $\mathbb{I}_{\cup_{P^j_t}}$ is the indicator function of the union of object proposals from the detector at time t. $Coh(x)$ will have a high value if proposals are frequently generated at x. The update of the objects state $\hat{R}^i_{t+1|t+1}$ is then obtained as

$$\hat{R}^i_{t+1|t+1} = \arg\max_R \int_{\partial R} \frac{e(x)}{Coh(x)} \left(g_{\partial P^j_{t+1}}(x) + g_{\partial R^i_{t+1|t}}(x)\right) \tag{5.35}$$

$$s.t.\ R \subset R^i_{t+1|t} \cup P^j_{t+1}, \quad R \supset R^i_{t+1|t} \cap P^j_{t+1} \tag{5.36}$$

This problem can be solved efficiently with graph cuts. Specifically, we convert the above to the standard form

$$E(l) = \sum_{x \in \Omega} f(x, l(x))dx + \sum_{|y-x|<\epsilon,\ x,y \in \Omega} v(l(x), l(y)) \tag{5.37}$$

where the data term $f(x, l(x))$ is the cost of assigning label $l(x)$ to pixel x, and the regularizer $v(l(x), l(y))$ is the cost of assigning labels $l(x), l(y)$ to neighbouring pixels x, y, so that graph cuts can return the label map l^* that minimizes the energy in eq. 5.37. Here we only solve a binary graph cuts problem (i.e. $l(x) = 1$ for the object region and $l(x) = 0$ for the background). The data term is

$$f(x, l(x)) = \begin{cases} 0, x \in R^i_{t+1|t} \cap P^j_{t+1},\ l(x) = 1 \\ \eta_1, x \in R^i_{t+1|t} \cup P^j_{t+1}, x \notin R^i_{t+1|t} \cap P^j_{t+1},\ l(x) = 1 \\ \eta_2, x \in \Omega \setminus (R^i_{t+1|t} \cup P^j_{t+1}),\ l(x) = 1 \\ \eta_2, x \in R^i_{t+1|t} \cap P^j_{t+1},\ l(x) = 0 \\ \eta_1, x \in R^i_{t+1|t} \cup P^j_{t+1}, x \notin R^i_{t+1|t} \cap P^j_{t+1},\ l(x) = 0 \\ 0, x \in \Omega \setminus (R^i_{t+1|t} \cup P^j_{t+1}),\ l(x) = 0, \end{cases} \tag{5.38}$$

where η_1, η_2 are positive numbers and $\eta_2 \gg \eta_1$. This ensures that the boundary is decided only in the area where the proposal and the predicted object region disagree. The overlap of the the predicted region and the proposal region is forced to object and the complement of the union of the two regions is set to background. The regularizer $v(l(x), l(y))$ is

$$\begin{cases} \dfrac{Coh(x)}{\left(g_{\partial P^j_{t+1}}(x) + g_{\partial R^i_{t+1|t}}(x)\right) e(x)} + \dfrac{Coh(y)}{\left(g_{\partial P^j_{t+1}}(y) + g_{\partial R^i_{t+1|t}}(y)\right) e(y)},\ l(x) \neq l(y) \\ 0,\ l(x) = l(y) \end{cases} \tag{5.39}$$

Given l^* is the solution of Eq. (5.37) with (5.38)-(5.39); The updated objects state $\hat{R}^i_{t+1|t+1}$ in Eq (5.36) is given by:

$$\hat{R}^i_{t+1|t+1} = \{x, l^*(x) = 1\}. \tag{5.40}$$

Enforcing opacity for all object regions: After updating object regions that had overlapped with new proposals, we append the set of promoted object proposals, resulting in the combined set $\{\hat{R}^i_{t+1|t+1}\}_{i=1}^{N_{t+1}} = \{\hat{R}^i_{t+1|t+1}\}_{i=1}^{N_t} \cup \{\hat{R}^k_{t+1|t+1}\}_{k=N_t+1}^{N_{t+1}}$.

Note that the updated objects might overlap one another, which does not satisfy the visibility constriant $\cap_{i=0}^{N_{t+1}} \hat{R}^i_{t+1|t+1} = \emptyset$ and $\cup_{i=0}^{N_{t+1}} \hat{R}^i_{t+1|t+1} = \Omega$. To ensure that only a single object label occupies each pixel, we obtain our final object estimates by solving a multi-label graph cut problem described below. The data term is

$$f(x, l(x)) = \begin{cases} 0, x \in \hat{R}^i_{t+1|t+1}, x \notin \cup_{j \neq i} \hat{R}^j_{t+1|t+1}, l(x) = i \\ \eta_1, x \in \hat{R}^i_{t+1|t+1} \cup \hat{R}^j_{t+1|t+1}, j \neq i, l(x) = i \\ \eta_2, x \in \Omega \setminus \hat{R}^i_{t+1|t+1}, l(x) = i \end{cases} \tag{5.41}$$

where $l(x)$ could have multiple values in the set $\{1, 2, \ldots, N_{t+1}\}$. Here, $\hat{R}^0_{t+1|t+1}$ represents the background region and is automatically set to be the complement of the union of all objects. The regularity term is:

$$v(l(x), l(y)) = \begin{cases} \frac{1}{e(x)} + \frac{1}{e(y)},\ l(x) \neq l(y) \\ 0,\ l(x) = l(y) \end{cases} \tag{5.42}$$

Suppose l^* is the solution of the above graph cut problem, then the final update of the objects is:

$$R^i_{t+1|t+1} = \{x, l^*(x) = i\} \tag{5.43}$$

A high-level view of this update procedure is summarized in Alg. 3. In Fig. 5.14, we show examples where both the prediction and the proposals are incorrect, but the update corrects the mistakes of both.

5.4.3 Testing of the integrated system

Our integrated system segments video into object labels. We evaluate on two popular datasets, FBMS59 [OMB14a] and BVSD [GNJ13]. We used the toolbox of [DZ13] for edge detection on both the image and motion field to obtain e_{image} and e_{motion} mentioned in Eq. 5.29. The linear weights are set to $c_1 = 0.1, c_2 = 0.9$ for Moseg and $c_1 = 0.3, c_2 = 0.7$ for BVSD. For both datasets, $\delta = 5, \varepsilon = 0.0001$.

Following the evaluation protocol of [OMB14a], we report *precision*, *recall*, *F-measure*, and the number of extracted objects (labeled regions with F-measure ≥ 0.75) in Tab. 5.5.

In Fig. 5.15, we visually compare our output with other state-of-the-art methods on sample frames from FBMS59. Our conservative approach to labeling object yields little to

Algorithm 3 Update Algorithm

```
procedure UPDATE
    for each $P^j_{t+1}$ do
        if $P^j_{t+1}$ overlaps sufficiently with existing object $R^i_{t+1|t}$ then
            $\hat{R}^j_{t+1|t+1} \leftarrow$ apply update scheme for each object region to $P^j_{t+1}$ and $R^i_{t+1|t}$
        else if $P^j_{t+1}$ overlaps sufficiently with existing proposal $\bar{P}^k_{t+1|t}$ then
            $p^k_{t+1|t+1}(x) \leftarrow p^k_{t+1|t}(x) + s(P^j_{t+1})\mathbb{I}_{P^j_{t+1}}(x)$
        else
            append $P^j_{t+1}$ to the proposal pool
            $M_{t+1} = M_t + 1$
            $p^{M_t}_{t+1|t+1}(x) = s(P^j_{t+1})\mathbb{I}_{P^j_{t+1}}(x)$
        end if
    end for
    Enforcing opacity for all object regions.
end procedure
```

	Training set (29 sequences)				Test set (30 sequences)			
	P	R	F	$N/65$	P	R	F	$N/69$
[GKH10b]	79.17	47.55	59.42	4	77.11	42.99	55.20	5
[OMB14a]	81.50	63.23	71.21	16	74.91	60.14	66.72	20
[AS12b]	87.20	59.60	70.81	17	79.64	50.73	61.98	7
[TKS15c]	85.00	67.99	75.55	21	82.37	58.37	68.32	17
[TKS15c]-NC	83.00	70.10	76.01	23	77.94	59.14	67.25	15
[YSS15a]	**89.53**	70.74	79.03	26	**91.47**	64.75	75.82	27
ours	87.43	**77.03**	**81.9**	**29**	89.19	**70.74**	**78.9**	**32**

Table 5.5: FBMS-59 results. Average precision (P), recall (R) and f-measure (F) over all sequences in the training and test datasets of FMS-59. Higher values indicate superior performance. All methods are fully automatic. Methods [TKS15c] and our method are causal. The other methods process the whole video in batch.

	Boundary			Volume		
	P	R	F	P	R	F
[OMB14a]	0.566	0.100	0.170	0.146	0.852	0.249
[TKS15c]	**0.760**	0.186	0.299	0.136	0.870	0.234
ours	0.697	**0.198**	**0.308**	**0.164**	**0.874**	**0.276**

Table 5.6: Quantitative results on BVSD dataset. Average precision (P), recall (R) and f-measure (F) over all sequences in the test datasets of BVSD. Higher values indicate superior performance.

no spurious object detections, even when the sequence has complex motion with multiple independently moving objects (rows $2, 3, 4$ in Fig. 5.15 and additionally, Fig. 5.13). Moreover, our system shows better visual performance than the others due to the interplay between the two modules in our system, as each component accounts for the failures of the other. Our system can still fail when objects are far away and do not generate sufficient occlusions for the pseudo-measurement modular to produce an object proposal (see Fig. 5.15 row 8).

In Tab. 5.6, we show our quantitative results on the BVSD dataset. We report both boundary precision-recall (BPR), a commonly used metric in image segmentation, and volume precision-recall (VPR), which quantifies the spatio-temporal overlap of our labels and the groundtruth regions.

For a VGA frame, our method takes approximately 2 minutes with 3 objects in the state on a single core, and around 1 minute per frame when running in parallel running on a standard desktop (8 core i7-3770 3.4GHz with 16GB RAM).

The developed causal video object detection and tracking system can discover objects and localize their boundaries in video. It should be noted that we wish to track the evolving occluding boundary of objects as a geometric entity, as that is informed by the shape of the object in 3D. One could also track bounding boxes, in our case, we make the separation of geometric and photometric properties explicit, which can be useful for the analysis and classification of objects that are defined solely by their shape, irrespective of reflectance, or vice-versa.

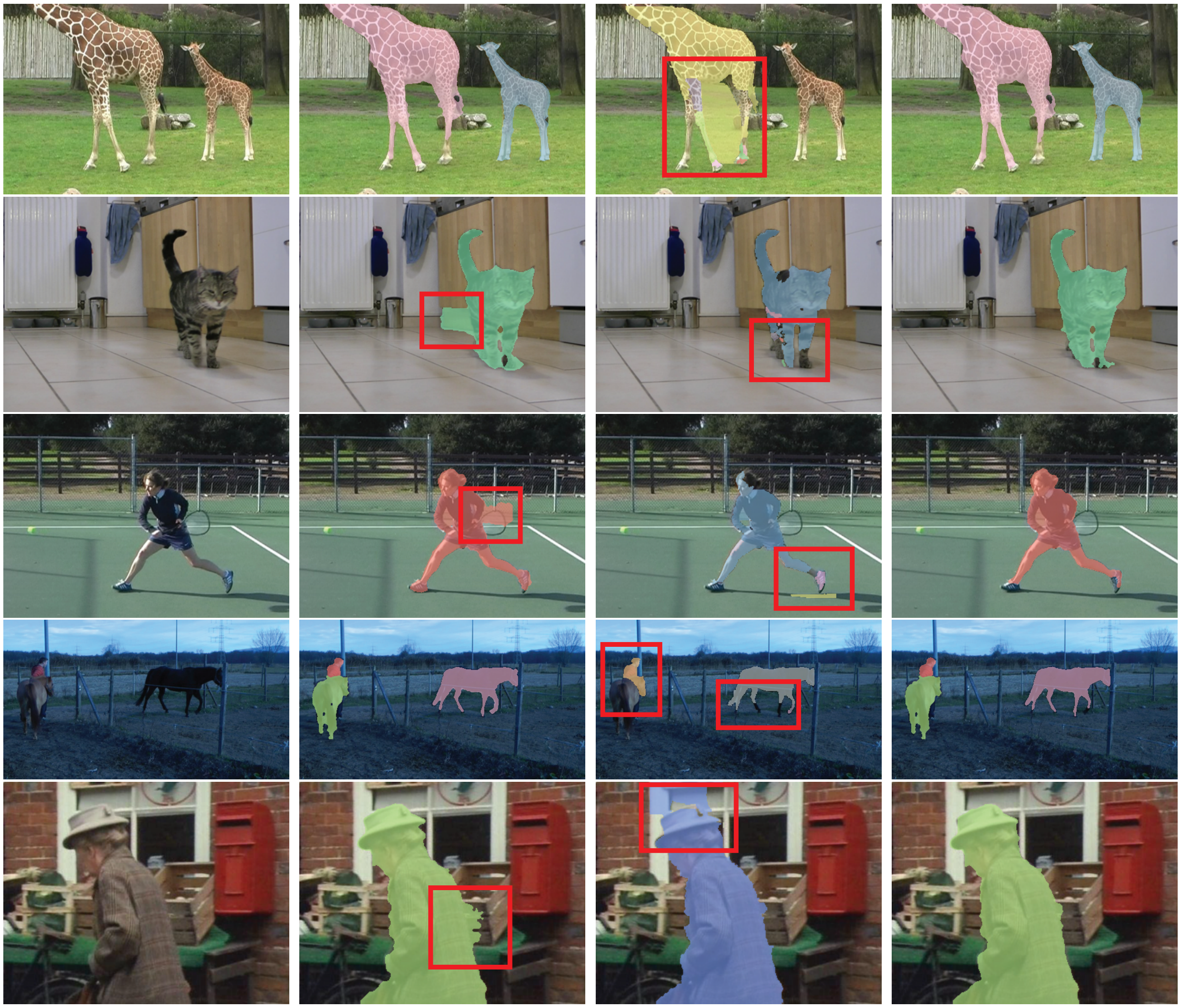


Figure 5.14: Illustration of the update of tracking and detection. From left to right: original image, prediction from tracker, object proposal from detector, updated estimate of objects. Errors made by prediction or detection are marked by the red rectangles, which are corrected in the final estimate in 4th column.

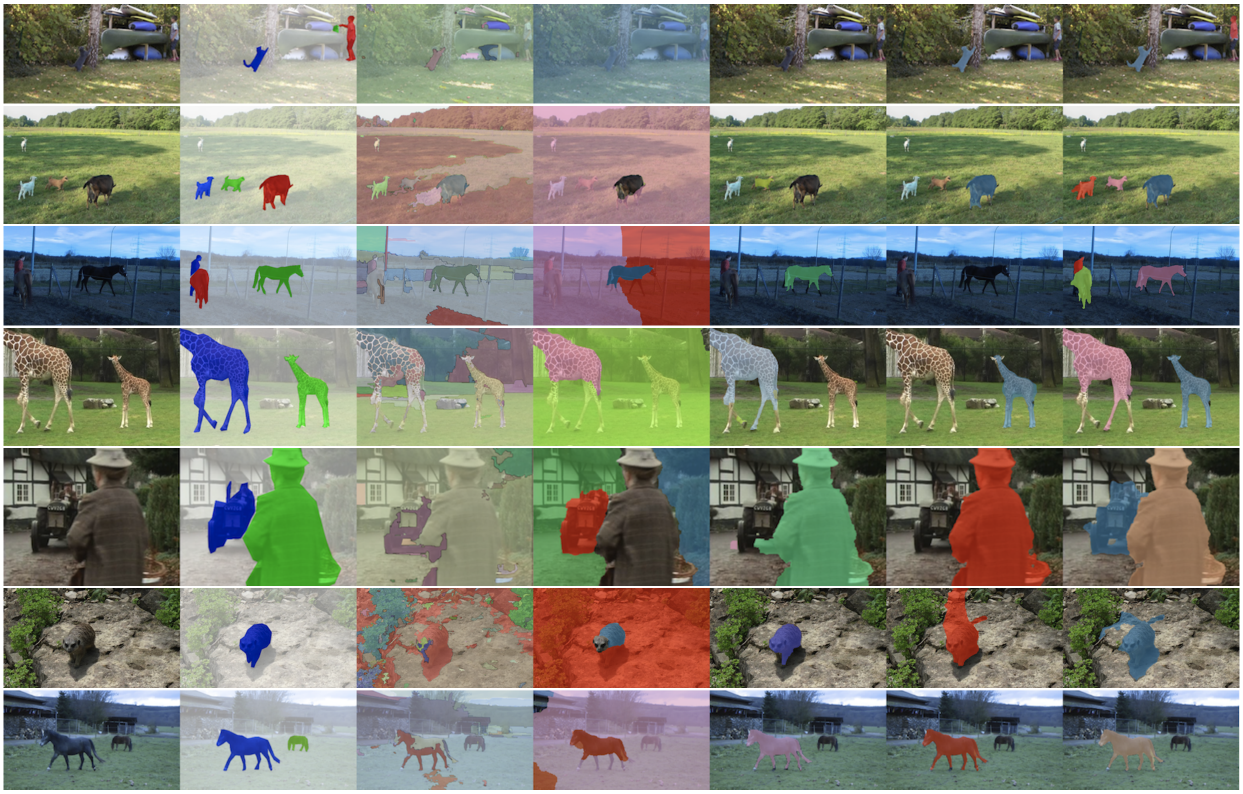


Figure 5.15: Visual Comparison on FBMS59. From left to right: Original frame, groundtruth, [GKH10c], [OMB14a], [TKS15c], [YSS15a], Ours. Notice the tracker and detector independently perform worse that our method, were we obtain all of the goats (row 2) and both horses and the man (row 3). We miss the horse in the far field (row 7) as the object remains stationary throughout the video.

CHAPTER 6

Adversarial Contextual Model for Unsupervised Moving Object Detection

Even though occlusion cues provide a representation to individualize an object from the scene as seen in the previous chapter, the quality of the detection heavily depends on the estimation of occlusions. And by Information Processing Inequality, the processed occlusion contains less information than what we can infer from the raw optical flows or images. In this chapter, we want to develop a richer representation of objects, which can be used for robust real-time object detection, in an unsupervised manner. And this is also closely related to the question asked by Marr [Mar82], what makes an object so special that it should be recoverable as a region in an image?

We propose an adversarial contextual model for detecting moving objects in images. A deep neural network is trained to predict the optical flow in a region using information from everywhere else but that region (context), while another network attempts to make such context as uninformative as possible. The result is a model where hypotheses naturally compete with no need for explicit regularization or hyper-parameter tuning. Although our method requires no supervision whatsoever, it outperforms several methods that are pre-trained on large annotated datasets. Our model can be thought of as a generalization of classical variational generative region-based segmentation, but in a way that avoids explicit regularization or solution of partial differential equations at run-time.

Consider Fig. 6.1: Even relatively simple objects, when moving in the scene, cause complex discontinuous changes in the image. Being able to rapidly detect independently moving objects in a wide variety of scenes from images is functional to survival for animals and

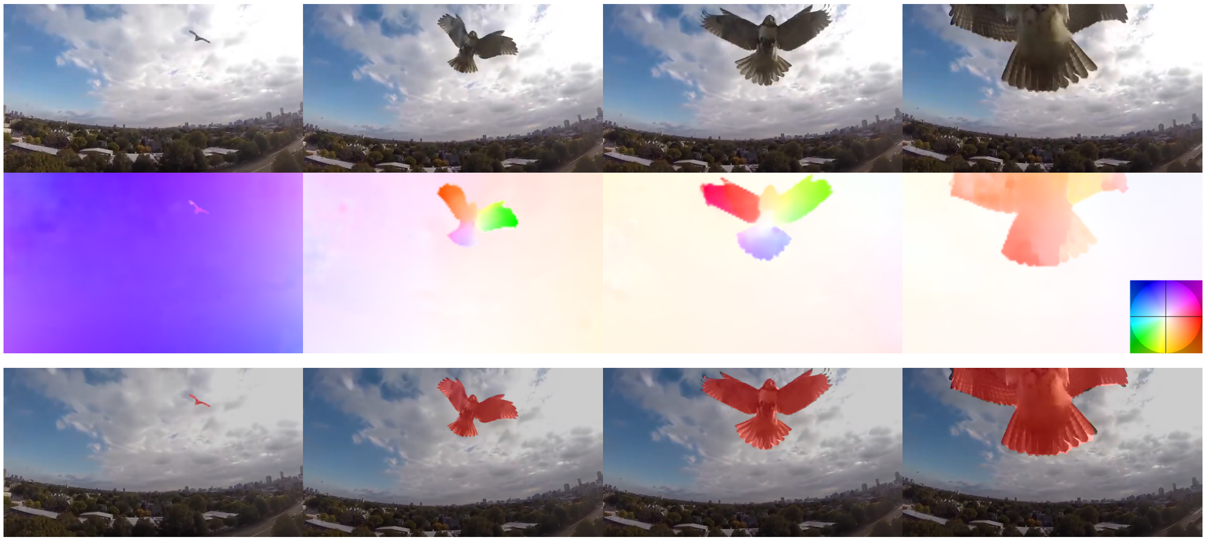


Figure 6.1: An encounter between a hawk and a drone (top). The latter will not survive without being aware of the attack. Detecting moving objects is crucial to survival of animal and artificial systems alike. Note that the optical flow (middle row) is quite diverse within the region where the hawk projects: It changes both in space and time. Grouping this into a moving object (bottom row) is our goal in this work. Note the object is detected by our algorithm across multiple scales, partial occlusions from the viewpoint, and complex boundaries.

autonomous vehicles alike. We wish to endow artificial systems with similar capabilities, without the need to pre-condition or learn similar-looking backgrounds. This problem relates to motion segmentation, foreground/background separation, visual attention, video object segmentation as we discuss in Sect. 6.2. For now, we use the words "object" or "foreground" informally[1] to mean (possibly multiple) connected regions of the image domain, to be distinguished from their surrounding, which we call "background" or "context," according to *some* criterion.

Since objects exist in the scene, not in the image, a method to infer them from the latter rests on an operational definition based on measurable image correlates. We call moving objects regions of the image whose motion cannot be explained by that of their surroundings. In other words, the motion of the background is uninformative of the motion of the foreground and vice-versa. The "information separation" can be quantified by the informa-

[1] The precise meaning of these terms will be formalized in Sect. 6.1.

tion reduction rate (IRR) between the two as defined in Sect. 6.1. This naturally translates into an adversarial inference criterion that has close connections with classical variational region-based segmentation, but with a twist: Instead of learning a generative model of a region that explains the image *in that region* as well as possible, our approach yields a model that tries to explain it *as poorly as possible* using measurements from *everywhere else but* that region.

In generative model-based segmentation, one can always explain the image with a trivial model, the image itself. To avoid that, one has to impose model complexity bounds, bottlenecks or regularization. Our model does not have access to trivial solutions, as it is forced to predict a region without looking at it. What we learn instead is a contextual adversarial model, without the need for explicit regularization, where foreground and background hypotheses compete to explain the data with no pre-training nor (hyper)parameter selection. In this sense, our approach relates to adversarial learning and self-supervision as discussed in Sect. 6.2.

The result is a completely unsupervised method, unlike many recent approaches that are called unsupervised but still require supervised pre-training on massive labeled datasets and can perform poorly in contexts that are not well represented in the training set. Despite the complete lack of supervision, our method performs competitively even compared with those that use supervised pre-training (Sect. 6.3).

Our method captures the desirable features of variational region-based segmentation: Robustness, lack of thresholds or tunable parameters, no need for training. However, it does not require solving a partial differential equation (PDE) at run-time, nor to pick regularizers, or Lagrange multipliers, nor to restrict the model to one that is simple-enough to be tractable analytically. It also exploits the power of modern deep learning methods: It uses deep neural networks as the model class, optimizes it efficiently with stochastic gradient descent (SGD), and can be computed efficiently at run time. However, it requires no supervision whatsoever.

While our approach has close relations to both classical region-based variational segmentation and generative models, as well as modern deep learning-based self-supervision,

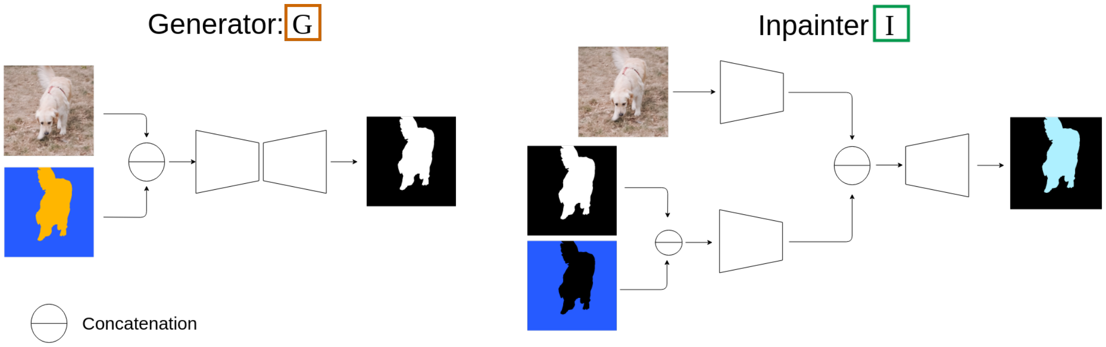


Figure 6.2: During training our method entails two modules. One is the generator (G), which produces a mask of the object by looking at the image and the associated optical flow. The other module is the inpainter (I), which tries to inpaint back the optical flow masked out by the corresponding mask. Both modules employ the encoder-decoder structure with skip connections, except that the inpainter (I) is equipped with two separate encoding branches. See section 6.3.1 for network details.

discussed in detail in Sect. 6.2, to the best of our knowledge, it is the first *adversarial contextual model* to detect moving objects in images.

6.1 Method

We call "moving object(s)" or "foreground" any region of an image whose motion is unexplainable from the context. A "region of an image" Ω is a compact and multiply-connected subset of the domain of the image, discretized into a lattice D. "Context" or "background" is the complement of the foreground in the image domain, $\Omega^c = D \backslash \Omega$. Given a measured image I and/or its optical flow to the next (or previous) image u, foreground and background are uncertain, and therefore treated as random variables. A random variable u_1 is "unexplainable" from (or "uninformed" by) another u_2 if their mutual information $\mathbb{I}(u_1; u_2)$ is zero, that is if their joint distribution equals the product of the marginals, $P(u_1, u_2) = P(u_1)P(u_2)$.

More specifically, the optical flow $u : D_1 \to D_2 \subset \mathbb{R}^2$ maps the domain of an image $I_1 : D_1 \to \mathbb{R}^3_+$ onto another D_2 of I_2, so that if $x_i \in D_1$, then $x_i + u_i \in D_2$, where $u_i = u(x_i)$ up to a discretization into the lattice and cropping of the boundary (occlusions). Ideally,

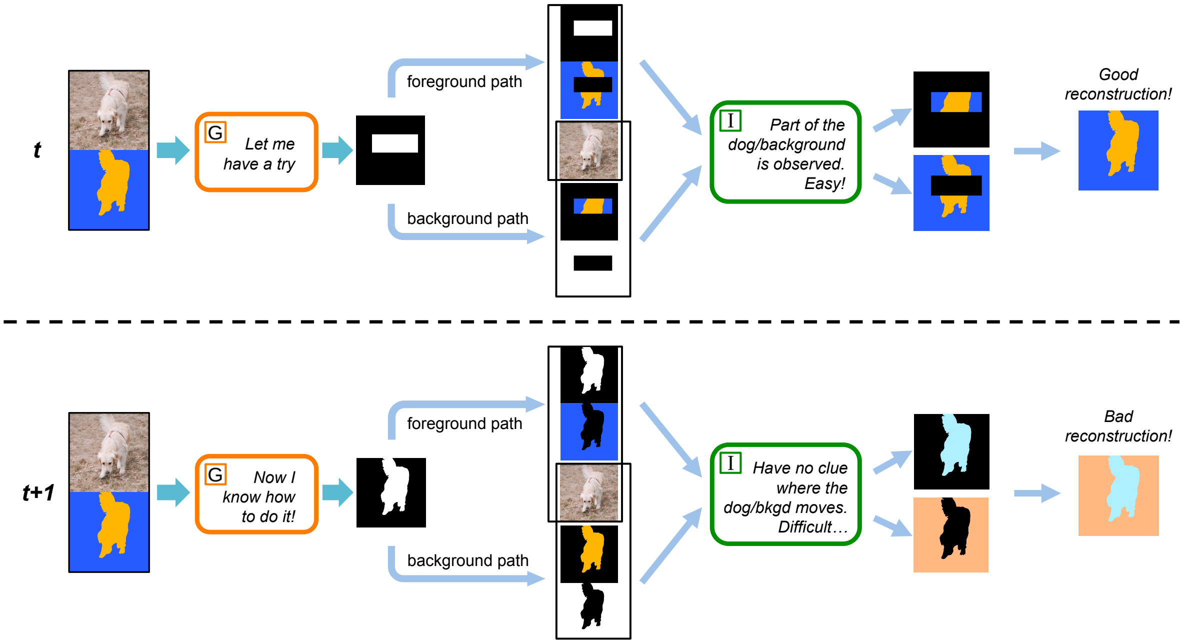


Figure 6.3: The two diagrams above show the learning process of the mask generator (G), when the inpainter (I) has already learned the conditionals to accurately inpaint the masked flow. The upper diagram shows that when the mask generated is not covering the object precisely, the inpainter will be informed about the optical flow in the mask by the flow in the complement mask, and be able to make a good reconstruction. Similarly for the complement mask. However, in the lower diagram, when the object is precisely masked against the background, the inpainter (I) only observes the flow in the context, and has no information to predict the flow inside the object. Note that a randomly initialized inpainter (I) also knows nothing about the conditionals, thus we propose to jointly train both the generator (G) and the inpainter (I) in an adversarial manner as in Sect. 6.1.

if the brightness constancy constraint equation that defines optical flow was satisfied, we would have $I_1 = I_2 \circ u$ point-wise.

If we consider the flow at two locations i, j, we can formalize the notion of foreground as a region Ω that is uninformed by the background:

$$\begin{cases} \mathbb{I}(u_i, u_j | I) > 0, i, j \in \Omega \\ \mathbb{I}(u_i, u_j | I) = 0, i \in \Omega, j \in D \setminus \Omega. \end{cases} \tag{6.1}$$

As one would expect, based on this definition, a subset of an object informs the remaining part of this object, and a subset of the foreground does not inform a subset of the background.

6.1.1 Loss function

We now operationalize the definition of foreground into a criterion to infer it. We use the information reduction rate (IRR) γ, which takes two subsets $\boldsymbol{x}, \boldsymbol{y} \subset D$ as input and returns a positive scalar:

$$\gamma(\boldsymbol{x}|\boldsymbol{y}; I) = \frac{\mathbb{I}(u_{\boldsymbol{x}}, u_{\boldsymbol{y}}|I)}{\mathbb{H}(u_{\boldsymbol{x}}|I)} = 1 - \frac{\mathbb{H}(u_{\boldsymbol{x}}|u_{\boldsymbol{y}}, I)}{\mathbb{H}(u_{\boldsymbol{x}}|I)} \tag{6.2}$$

where $\mathbb{H}$ denotes (Shannon) entropy. It is zero when the two variables are independent, but the normalization prevents the trivial solution (empty set).[2] As proven next, objects as we defined them are the regions that minimize the following loss function

$$\mathcal{L}(\Omega; I) = \gamma(\Omega|\Omega^c; I) + \gamma(\Omega^c|\Omega; I). \tag{6.3}$$

Note that $\mathcal{L}$ *does not have a complexity term*, or regularizer, as one would expect in most region-based segmentation methods. This is a key strength of our approach, that involves no modeling hyperparameters, as we elaborate on in Sect. 6.2.

Before we start the proof, we make the following two statements:

statement 1: *a subset of an object informs the remaining part of this object.* If the object is Ω, and there is a subset $\hat{\Omega} \subset \Omega$, suppose $i \in \hat{\Omega}$, $j \in \Omega \setminus \hat{\Omega}$ respectively, then: $\mathbb{I}(u_{\hat{\Omega}}, u_{\Omega\setminus\hat{\Omega}}|I) \geq \mathbb{I}(u_i, u_{\Omega\setminus\hat{\Omega}}|I) \geq \mathbb{I}(u_i, u_j|I) > 0$ by Eq. (6.1).

[2] A small constant $0 < \epsilon \ll 1$ is added to the denominator to avoid singularities, and whenever $\boldsymbol{x} \neq \emptyset$, $\mathbb{H}(u_{\boldsymbol{x}}|I) \gg \epsilon$, thus we will omit ϵ from now on.

statement 2: *a subset of the foreground does not inform a subset of the background.* Suppose Ω is the foreground, if $\hat{\Omega} \subset \Omega$, and $\Omega' \subset D \setminus \Omega$, then $\mathbb{I}(u_{\hat{\Omega}}, u_{\Omega'}|I) = 0$. Otherwise, we can find at least two pixels $i \in \hat{\Omega}$, $j \in \Omega'$ such that $\mathbb{I}(u_i, u_j|I) > 0$, which is contradictory to definition Eq. (6.1).

Proof: First we show that the estimate Ω^* right on the object achieves the minimum value of the loss function, since:

$$\mathcal{L}(\Omega^*; I) = \gamma(\Omega^*|D \setminus \Omega^*; I) + \gamma(D \setminus \Omega^*|\Omega^*; I) = \frac{\mathbb{I}(u_{\Omega^*}, u_{D\setminus\Omega^*}|I)}{\mathbb{H}(u_{\Omega^*}|I)} + \frac{\mathbb{I}(u_{D\setminus\Omega^*}, u_{\Omega^*}|I)}{\mathbb{H}(u_{D\setminus\Omega^*}|I)} = 0 \tag{6.4}$$

by statement (2) above. Thus Ω^* achieves the minimum value of the loss Eq. (6.3). Now we need to show that Ω^* is unique, for which, we just need to check the following two inclusive cases for $\hat{\Omega} \neq \Omega^*$ (note that $\mathcal{L}(\emptyset; I) = \mathcal{L}(D; I) = 1.0$ as $0 < \epsilon \ll 1$ is added to the denominator):

- $\hat{\Omega}$ is either a subset of foreground or a subset of background: $\hat{\Omega} \cap D \setminus \Omega^* = \emptyset$ or $\hat{\Omega} \cap \Omega^* = \emptyset$.
- $\hat{\Omega}$ is neither a subset of foreground nor a subset of background: $\hat{\Omega} \cap D \setminus \Omega^* \neq \emptyset$ and $\hat{\Omega} \cap \Omega^* \neq \emptyset$.

In both cases $\mathcal{L}(\hat{\Omega}; I)$ is strictly larger than 0 with some set operations under statements (1,2) above. Thus the object satisfies the definition Eq. (6.1) is a unique optima of the loss Eq. (6.3).

Tame as it may look, (6.3) is intractable in general. For simplicity we indicate the flow inside the region(s) Ω (foreground) with $u^{\text{in}} = \{u_i,\ i \in \Omega\}$, and similarly for u^{out}, the flow in the background Ω^c. The only term that matters in the IRR is the ratio $\mathbb{H}(u^{\text{in}}|u^{\text{out}}, I)/\mathbb{H}(u^{\text{in}}|I)$, which is

$$\frac{\int \log P(u^{\text{in}}|u^{\text{out}}, I) dP(u^{\text{in}}|u^{\text{out}}, I)}{\int \log P(u^{\text{in}}|I) dP(u^{\text{in}}|I)} \tag{6.5}$$

that measures the information transfer from the background to the foreground. This is minimized when knowledge of the background flow is sufficient to predict the foreground. To enable computation, we have to make draconian, yet common, assumptions on the underlying probability model, namely that

$$\begin{aligned} P(u^{\text{in}} = x|I) &\propto \exp\left(-\frac{\|x\|^2}{\sigma^2}\right) \\ P(u^{\text{in}} = x|u^{\text{out}} = y, I) &\propto \exp\left(-\frac{\|x - \phi(\Omega, y, I)\|^2}{\sigma^2}\right) \end{aligned} \tag{6.6}$$

where $\phi(\Omega, y, I) = \int u^{\text{in}} dP(u^{\text{in}}|u^{\text{out}}, I)$ is the conditional mean given the image and the complementary observation. Here we assume $\phi(\Omega, \emptyset, I) = 0$, since given a single image the most probable guess of the flow is zeros. With these assumptions, (6.5) can be simplified, to

$$\frac{\int \|u^{\text{in}} - \phi(\Omega, u^{\text{out}}, I)\|^2 dP(u^{\text{in}}|u^{\text{out}}, I)}{\int \|u^{\text{in}}\|^2 dP(u^{\text{in}}|I)} \approx \frac{\sum_{i=1}^N \|{u_i}^{\text{in}} - \phi(\Omega, {u_i}^{\text{out}}, I)\|^2}{\sum_{i=1}^N \|{u_i}^{\text{in}}\|^2} \tag{6.7}$$

where $N = |\mathcal{D}|$ is the cardinality of $\mathcal{D}$, or the number of flow samples available. Finally, our loss (6.3) to be minimized can be approximated as

$$\mathcal{L}(\Omega; I) = 1 - \frac{\sum_{i=1}^N \|{u_i}^{\text{in}} - \phi(\Omega, {u_i}^{\text{out}}, I)\|^2}{\sum_{i=1}^N \|{u_i}^{\text{in}}\|^2 + \epsilon} + 1 - \frac{\sum_{i=1}^N \|{u_i}^{\text{out}} - \phi(\Omega^c, {u_i}^{\text{in}}, I)\|^2}{\sum_{i=1}^N \|{u_i}^{\text{out}}\|^2 + \epsilon}. \tag{6.8}$$

In order to minimize this loss, we have to choose a representation for the unknown region Ω and for the function ϕ.

6.1.2 Function class

The region Ω that minimizes (6.8) belongs to the power set of D, that is the set of all possible subsets of the image domain, which has exponential complexity.[3] We represent it with the indicator function

$$\begin{aligned} \chi : D &\to \{0, 1\} \\ i &\mapsto 1 \text{ if } i \in \Omega;\ 0 \text{ otherwise} \end{aligned} \tag{6.9}$$

so that the flow inside the region Ω can be written as ${u_i}^{\text{in}} = \chi u_i$, and outside as ${u_i}^{\text{out}} = (1 - \chi) u_i$.

[3] In the continuum, it belongs to the infinite-dimensional set of compact and multiply-connected regions of the unit square.

Similarly, the function ϕ is non-linear, non-local, and high-dimensional, as it has to predict the flow in a region of the image of varying size and shape, given the flow in a different region. In other words, ϕ has to capture the context of a region to *recover* its flow.

Characteristically for the ages, we choose both ϕ and χ to be in the parametric function class of deep convolutional neural networks, as shown in Fig. 6.2, the specifics of which are in Sect. 6.3.1. We indicate the parameters with w, and the corresponding functions ϕ_{w_1} and χ_{w_2}. Accordingly, after discarding the constants, the *negative* loss (6.8) can be written as a function of the parameters

$$\mathcal{L}(w_1, w_2; I) = \frac{\sum\limits_i \|\chi_{w_2}(u_i - \phi_{w_1}(\chi_{w_2}, {u_i}^{\text{out}}, I))\|^2}{\sum\limits_i \|{u_i}^{\text{in}}\|^2} + \frac{\sum\limits_i \|(1 - \chi_{w_2})(u_i - \phi_{w_1}(1 - \chi_{w_2}, {u_i}^{\text{in}}, I)\|^2}{\sum\limits_i \|{u_i}^{\text{out}}\|^2} \tag{6.10}$$

ϕ_{w_1} is called the *inpainter network*, and must be chosen to *minimize* the loss above. At the same time, the region Ω, represented by the parameters w_2 of its indicator function χ_{w_2} called *mask generator network*, should be chosen so that u^{out} is as uninformative as possible of u^{in}, and therefore the same loss is *maximized* with respect to w_2. This naturally gives rise to a minimax problem:

$$\hat{w} = \arg\min_{w_1}\max_{w_2} \mathcal{L}(w_1, w_2; I). \tag{6.11}$$

This loss has interesting connections to classical region-based segmentation, but with a twist as we discuss next.

6.2 Relation to Prior Work

To understand the relation of our approach to classical methods, consider the simplest model for region-based segmentation [CV01]

$$L(\Omega, c_i, c_o) = \int_\Omega |u^{\text{in}}(x) - c_i|^2 dx + \int_{\Omega^c} |u^{\text{out}}(x) - c_o|^2 dx \tag{6.12}$$

typically combined with a regularizing term, for instance the length of the boundary of Ω. This is a convex infinite-dimensional optimization problem that can be solved by numerically integrating a partial differential equation (PDE). The result enjoys significant robustness to noise, provided the underlying scene has piecewise constant radiance and is measured by image irradiance, to which it is related by a simple "signal-plus-noise" model. Not many scenes of interest have piecewise constant radiance, although this method has enjoyed a long career in medical image analysis. If we enrich the model by replacing the constants c_i with smooth functions, $\phi_i(x)$, we obtain the celebrated Mumford-Shah functional [MS89], also optimized by integrating a PDE. Since smooth functions are an infinite-dimensional space, regularization is needed, which opens the Pandora box of regularization criteria, not to mention hyperparameters: Too much regularization and details are missed; too little and the model gets stuck in noise-induced minima. A modern version of this program would replace $\phi(x)$ with a parametrized model $\phi_w(x)$, for instance a deep neural network with weights w pre-trained on a dataset $\mathcal{D}$. In this case, the loss is a function of w, with natural model complexity bounds. Evaluating ϕ_w at a point inside, $x \in \Omega$, requires knowledge of the entire function u *inside* Ω, which we indicate with $\phi_w(x, u^{\text{in}})$:

$$\int_\Omega |u^{\text{in}}(x) - \phi_w(x, u^{\text{in}})|^2 dx + \int_{\Omega^c} |u^{\text{out}}(x) - \phi_w(x, u^{\text{out}})|^2 dx. \tag{6.13}$$

Here, a network can just map $\phi_w(x, u^{\text{in}}) = u^{\text{in}}$ providing a trivial solution, avoided by introducing (architectural or information) bottlenecks, akin to explicit regularizers. We turn the table around and use the outside to predict the inside and vice-versa:

$$\int_\Omega |u^{\text{in}}(x) - \phi_w(x, u^{\text{out}})|^2 dx + \int_{\Omega^c} |u^{\text{out}}(x) - \phi_w(x, u^{\text{in}})|^2 dx \tag{6.14}$$

After normalization and discretization, this leads to our loss function (6.8). The two regions compete: for one to grow, the other has to shrink. In this sense, our approach relates to region competition methods, and specifically Motion Competition [CS05], but also to adversarial training, since we can think of ϕ as the "discriminator" presented in a classification problem (GAN [ACB17]), reflected in the loss function we use. This also relates to what is called "self-supervised learning," a misnomer since there is no supervision, just a loss function that does not involve externally annotated data. Several variants of our approach can be constructed by using different norms, or correspondingly different models for the joint and marginal distributions (6.6).

More broadly, the ability to detect independently moving objects is primal, so there is a long history of motion-based segmentation, or moving object detection. Early attempts to explicitly model occlusions include the layer model [WA94b] with piecewise affine regions, with computational complexity improvements using graph-based methods [SM98] and variational inference [CS04, BBW06, SWS13b, YSS15b] to jointly optimize for motion estimation and segmentation; [OMB14c] use of long-term temporal consistency and color constancy, making however the optimization more difficult and sensitive to parameter choices. Similar ideas were applied to motion detection in crowds [BC06], traffic monitoring [BMC97] and medical image analysis [EGS11]. Our work also related to the literature on visual attention [IK00, BDR15].

More recent data-driven methods [TAS17b, TAS17a, CTW17a, SWZ18] learn discriminative spatio-temporal features and differ mainly for the type of inputs and architectures. Inputs can be either image pairs [SWZ18, CTW17a] or image plus dense optical flow [TAS17b, TAS17a]. Architectures can be either time-independent [TAS17a], or with recurrent memory [TAS17b, SWZ18]. Overall, those methods outperform traditional ones on benchmark datasets [OMB14c, PPM16], but at the cost of requiring a large amount of labeled training data and with evidence of poor generalization to previously unseen data.

It must be noted that, unlike in Machine Learning at large, it is customary in video object segmentation to call *"unsupervised"* methods that *do* rely on massive amounts of manually annotated data, so long as they do not require manual annotation at run-time. We adopt the broader use of the term where unsupervised means that there is no supervision of any kind both at training and test time.

Like classical variational methods, our approach does not need any annotated training data. However, like modern learning methods, our approach learns a contextual model, which would be impossible to engineer given the complexity of image formation and scene dynamics.

6.3 Experiments

We compare our approach to a set of state-of-the-art baselines on the task of video object segmentation to evaluate the accuracy of detection. We first present experiments on a controlled toy-example, where the assumptions of our model are perfectly satisfied. The aim of this experiment is to get a sense of the capabilities of the presented approach in ideal conditions. In the second set of experiments, we evaluate the effectiveness of the proposed model on three public, widely used datasets: Densely Annotated VIdeo Segmentation (DAVIS) [PPM16], Freiburg-Berkeley Motion Segmentation (FBMS59) [OMB14c], and SegTrackV2 [TFR10]. Provided the high degree of appearance and resolution differences between them, these datasets represent a challenging benchmark for any moving object segmentation method. While the DAVIS dataset has always a single object per scene, FBMS and SegTrackV2 scenes can contain multiple objects per frame. We show that our method not only outperforms the unsupervised approaches, but even edges out other supervised algorithms that, in contrast to ours, have access to a large amount of labeled data with precise manual segmentation at training time. For quantitative evaluation, we employ the most common metric for video object segmentation, *i.e.* the mean Jaccard score, a.k.a. intersection-over-union score, $\mathcal{J}$.

6.3.1 Implementation and Networks Details

Generator, $\boldsymbol{G}$: Depicted on the left of Fig. 6.3, the generator architecture is a shrunk version of SegNet [BKC17]. Its encoder part consists of 5 convolutional layers each followed by batch normalization, reducing the input image to $\frac{1}{4}$ of its original dimensions. The encoder is followed by a set of 4 atrous convolutions with increasing radius (2,4,8,16). The decoder part consists of 5 convolutional layers, that, with upsampling, generate an output with the same size of the input image. As in SegNet [BKC17], a final softmax layer generates the probabilities for each pixel to be foreground or background. The generator input consists of an RGB image I_t and the optical flow $u_{t:t+\delta T}$ between I_t and $I_{t+\delta T}$, to introduce more variations in the optical flows conditioned on image I_t. At training time, δT is randomly sampled from the uniform distribution $\mathcal{U} = [-5, 5]$, with $\delta T \neq 0$. The optical flow $u_{t:t+\delta T}$ is generated with the pretrained PWC network [SYL18], given its state-of-the-art accuracy and efficiency. The generator network has a total of 3.4M parameters.

Inpainter, *I*: We adapt the architecture of CPN [YS18] to build our inpainter network. Its structure is depicted on the right of Fig. 6.3. The input to this network consists of the input image I_t and the flow masked according to the generator output, χu, the latter concatenated with χ, to make the inpainter aware of the region to look for context. Differently from the CPN, these two branches are balanced, and have the same number of parameters. The encoded features are then concatenated and passed to the CPN decoder, that outputs an optical flow $\hat{u} = \phi(\chi, (1-\chi)u, I_t)$ of the same size of the input image, whose inside is going to be used for the difference between u^{in} and the recovered flow inside. Similarly, we can run the same procedure for the complement part. Our inpainter network has a total of 1.5M parameters.

At test time, only the generator G is used. Given I_t and $u_{t:t+\delta T}$, it outputs a probability for each pixel to be foreground or background, $P_t(\delta T)$. To encourage temporal consistency, we compute the temporal average:

$$\overline{P_t} = \sum_{\delta T=-5, \neq 0}^{\delta T=5} P_t(\delta T) \tag{6.15}$$

The final mask χ is generated with a CRF [KK11] post-processing step on the final $\overline{P_t}$.

6.3.2 Experiments in Ideal Conditions

Our method relies on basic, fundamental assumptions: *The optical flow of the foreground and of the background are independent.* To get a sense of the capabilities of our approach in ideal conditions, we artificially produce datasets where this assumption is fully satisfied. The datasets are generated as a modification of DAVIS2016 [PPM16], FMBS [OMB14c], and SegTrackV2 [TFR10]. While images are kept unchanged, ground truth masks are used to artificially perturb the optical flow generated by PWC [SYL18] such that foreground and background are statistically independent. More specifically, a different (constant) optical flow

	DAVIS [PPM16]	FBMS59 [OMB14c]	SegTrackV2 [TFR10]
$\mathcal{J} \uparrow$	92.5	88.5	92.1

Table 6.1: **Performance under ideal conditions:** When the assumptions made by our model are fully satisfied, our approach can successfully detect moving objects.. Indeed, our model reaches near maximum Jaccard score in all considered datasets.

field is sampled from a uniform distribution independently at each frame, and associated to the foreground and the background, respectively. It can be observed that in Table 6.1, our method reaches very high performance in all considered datasets. This confirms the validity of our algorithm and that our loss function (6.11) is a valid and tractable approximation of the functional (6.3).

6.3.3 Performance on Video Object Segmentation

As previously stated, we use the term *Unsupervised* with a different meaning with respect to its definition in literature of video object segmentation. In our definition and for what follows, the supervision refers to the algorithm's usage of ground truth object annotations at training time. In contrast, the literature usually defines methods as semi-supervised, if at test time they assume the ground-truth segmentation of the first frame to be known [BWL, MCC18]. This could be posed as tracking problem [YS15] since the detection of the target is human generated. Instead, here we focus on moving object detection and thus we compare our approach to the methods that are usually referred to as "unsupervised" in the video object segmentation domain. However we make further differentiation on whether the ground truth object segmentation is needed (supervised) or not (truly unsupervised) during training.

In this section we compare our method with other 8 methods that represent the state of the art for moving object segmentation. For comparison, we use the same metric defined above, which is the Jaccard score $\mathcal{J}$ between the real and predicted masks.

Table 6.2 shows the performance of our method and the baseline methods on three

	PDB	FSEG	LVO	ARP	FTS	NLC	SAGE	CUT	Ours
DAVIS2016 $\mathcal{J}\uparrow$	**77.2**	70.7	75.9	**76.2**	55.8	55.1	42.6	55.2	71.5
FBMS59 $\mathcal{J}\uparrow$	**74.0**	68.4	65.1	59.8	47.7	51.5	61.2	57.2	**63.6**
SegTrackV2 $\mathcal{J}\uparrow$	60.9	61.4	57.3	57.2	47.8	**67.2**	57.6	54.3	62.0
DNN-Based	Yes	Yes	Yes	No	No	No	No	No	Yes
Pre-Training Required	Yes	Yes	Yes	No	No	No	No	No	No

Table 6.2: **Moving Object Segmentation Benchmarks:** We compare our approach with 8 different baselines on the task of moving object segmentation (PDB [SWZ18], FSEG [JXG17], LVO [TAS17b], ARP [KK17], FTS [PF13], NLC [FI14], SAGE [WSP15], CUT [KAB15]). In order to do so, we use three popular datasets, *i.e.* DAVIS2016 [PPM16], FBMS59 [OMB14c], and SegTrackV2 [TFR10]. Methods in blue require ground truth annotations at training time and are pre-trained on image segmentation datasets. In contrast, methods in red are unsupervised and not require any ground-truth annotation. Our approach is top-two in all the considered benchmarks, comparing to the other unsupervised methods. **Bold** indicates best among all methods, while **Bold Red** and red represent the best and second best for unsupervised methods, respectively.

popular datasets, DAVIS2016 [PPM16], FBMS59 [OMB14c] and SegTrackV2 [TFR10]. Our approach is top-two in each of the considered datasets, and even outperforms baselines that need a large amount of labelled data at training time, *i.e.* FSEG [JXG17].

As can be observed in Table 6.2, unsupervised baselines typically perform well in one dataset but significantly worse in others. For example, despite being the best performing unsupervised method on DAVIS2016, the performance of ARP [KK17] drops significantly in the FBMS59 [OMB14c] and SegTrackV2 [OMB14c] datasets. ARP outperforms our method by 6.5% on DAVIS, however, *our method outperforms ARP by 6.3% and 8.4%, on FBMS59 and SegTrackV2 respectively.* Similarly, NLC [FI14] and SAGE [WSP15] are extremely competitive in the Segtrack and FBMS59 benchmarks, respectively, but not in others. NLC outperforms us on SegTrackV2 by 8.4%, however *we outperform NLC by 29.8% and 24.7%, on DAVIS and FBMS respectively.*

It has been established that being second-best in multiple benchmarks is more indicative of robust performance than being best in one [PL13]. Indeed, existing unsupervised approaches for moving object segmentation are typically highly-engineered pipeline methods which are tuned on one dataset but do not necessarily generalize to others. Also, consisting of several computationally intensive steps, extant unsupervised methods are generally orders of magnitude slower than our method (Table 6.3).

Interestingly, a similar pattern is observable for supervised methods. This is particularly evident on the SegTrackV2 dataset [TFR10], which is particularly challenging since several frames have very low resolution and are motion blurred. Indeed, supervised methods have difficulties with the covariate shift due to changes in the distribution between training and testing data. Generally, supervised methods alleviate this problem by pre-training on image segmentation datasets, but this solution clearly does not scale to every possible case. In contrast, our method can be finetuned on any data without the need for the latter to be annotated. As a result, our approach outperforms the majority of unsupervised methods as well as all the supervised ones, in terms of segmentation quality and training efficiency.

6.3.4 Qualitative experiments and Failure Cases

In Fig. 6.4 we show a qualitative comparison of the detection generated by our and others' methods on the DAVIS dataset. Our algorithm can segment precisely the moving object regardless of cluttered background, occlusions, or large depth discontinuities. The typical failure case of our method is the detection of objects whose motion is due to the primary object. An example is given in the last row of Fig. 6.4, where the water moved by the surfer is also classified as foreground by our algorithm.

	ARP [KK17]	FTS [PF13]	NLC [FI14]	SAGE [WSP15]	CUT [KAB15]	Ours
Runtime(s)	74.5	0.5	11.0	0.88	103.0	**0.098**
DNN-based	No	No	No	No	No	Yes

Table 6.3: **Run-time analysis**: Our method is not only effective (top-two in each considered dataset), but also orders of magnitude faster than other unsupervised methods. All timings are indicated without optical flow computation.

6.3.5 Training and Runtime Analysis

The generator and inpainter network's parameters are trained at the same time by optimizing the functional (6.11). The training time is approximately 6 hours on a single GPU Nvidia Titan XP. Since both our generator and inpainter networks are relatively small, we can afford very fast training/finetuning times. This stands in contrast to larger modules, *e.g.* PDB [SWZ18], that require up to 40 hrs of training.

At test time, predictions $\overline{P_t}$ (defined in eq. 6.15) are generated at 3.15 FPS, or with an average time of 320ms per frame, including the time to compute optical flow with PWC [SYL18]. Excluding the time to generate optical flow, our model can generate predictions at 10.2 FPS, or 98ms per frame. All previous timings do not include the CRF post-processing step. Table 6.3 compares the inference time of our method with respect to other unsupervised methods. Since our method at test time requires only a pass through a relatively shallow network, it is orders of magnitude faster than other unsupervised ap-

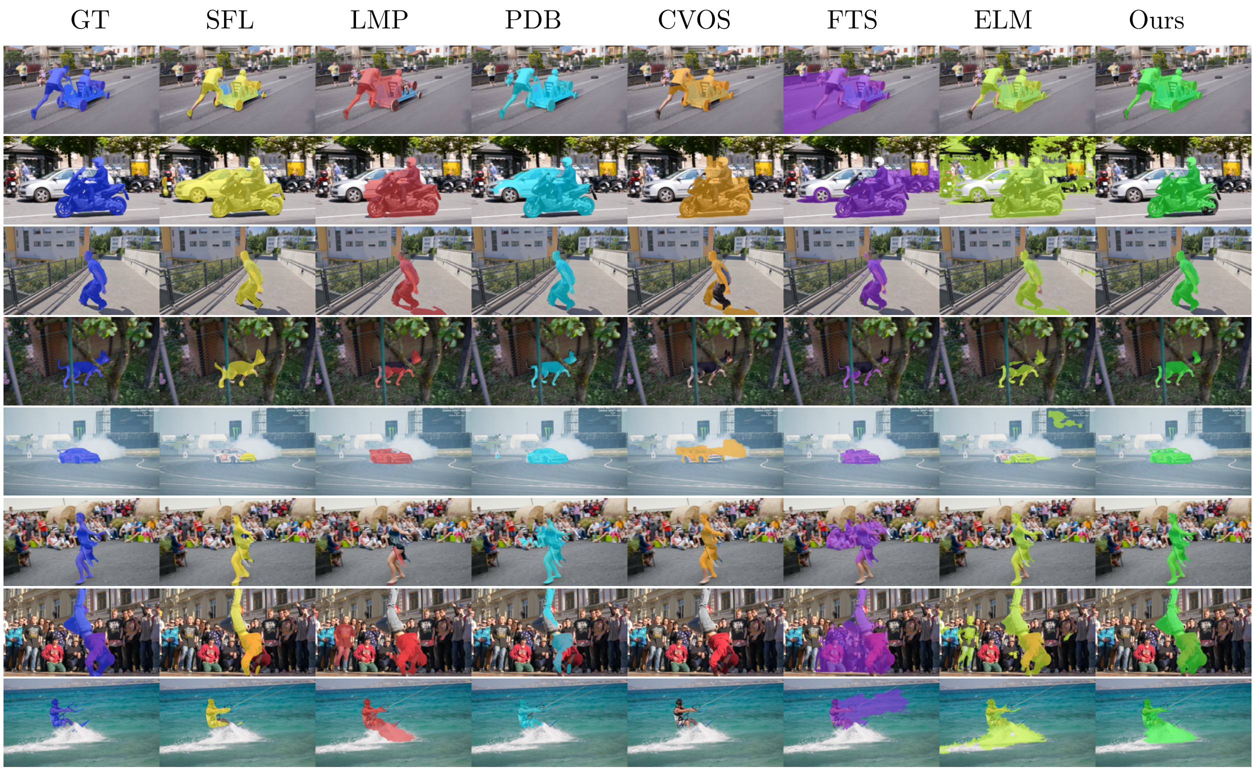


Figure 6.4: **Qualitative Results:** We qualitatively compare the performance of our approach with several state-of-the-art baselines as well as the Ground-Truth (GT) mask (SFL[CTW17b], LMP[TAS17a], PDB[SWZ18], CVOS[TKS15b], FTS[PF13], ELM[LS18]). Our prediction are robust to background clutter, large depth discontinuities and occlusions. The last row shows a typical failure case of our method, *i.e.* objects which are moved by the primary objects are detected as foreground (water is moved by the surfer in this case).

proaches.

6.4 Discussion

We have introduced a contextual adversarial model for moving object detection, based on information separation between foreground and background. The foreground (object) can be multiply-connected. Our model shows some strengths, and has limitations.

The strengths relate to the ability of the model to learn complex relations between foreground and background, that allows us to separate objects in ways that a generative model cannot, even when one plays with regularization parameters extensively. This is made possible by using modern deep neural network architectures, like SegNet [BKC17] and CPN [YS18], but does not require pre-training on massive annotated datasets.

This can be seen as a strength but also a limitation: If massive datasets are available, why not use them? In part because even massive is not large enough: There is evidence that models pre-trained on all available datasets still suffer performance drops whenever a new benchmark appears that has a significant covariate shift, as we show in the experiments. Moreover, our method outperforms models that require pre-training, despite being in principle at a disadvantage compared to them. In principle, the networks trained using our method can still finetune on the limited annotated datasets.

Our method is illustrated on motion-based segmentation, where the starting point is optical flow. One could argue that optical flow is costly, local, error-prone, all valid concerns. Our approach is general and could be applied to other statistics than optical flow.

Another possible limitation is that we do not make full use of the image: In some cases, the optical flow is ambiguous, in others intensity, but rarely is the combination of the two insufficient. Again, our framework allows in theory exploitation of both, and we intend to expand in this direction. Having said that, our method does make use of the image as a conditioning factor in the inpainter network.

Our definition of objects, and the resulting inference criterion, is related to generative model-based segmentation and region-based methods popular in the nineties, but with an important difference: Instead of using the evidence inside a region to infer a model of that region that is as accurate as possible, we use evidence *everywhere else but* that region to infer a model within, and we seek that model to be as bad as possible. This relation explored in detail in Sect. 6.2, forces learning a contextual model of the image, which is not otherwise the outcome of a generative model in region-based segmentation. For instance, if we choose a rich enough model class, we can trivially model the appearance of an object inside an image region as the image itself. This is not an option in our model: We can only predict the inside of a region by looking outside of it. This frees us from having to impose modeling assumptions to avoid trivial solutions, but requires a much richer class of function to harvest contextual information.

This naturally gives rise an adversarial (min-max) optimization: An inpainter network, tries to hallucinate the flow inside from the outside. Another network, a discriminator or regressor, tries to force the inpainting network to do the lousiest possible job.

CHAPTER 7

Look into the Future

Conceptualization of objects is a building block for high-level semantics related tasks. To build object representations, an agent would have to see objects in videos at first. We follow the course of visual development in early infancy to construct the components enabling the perception of objects, with a prioritization on unsupervised learning, such that the agent can learn from unlimited video data. We have presented Conditional Prior Networks (CPN) that help us achieving state-of-the-art performance in unsupervised optical flow prediction and depth estimation, by harvesting regularity of the scene from previous observations. We have also introduced the Adversarial Contextual Model (ACM) for unsupervised object detection using the contextual information separation criteria. With no manual annotation, the network trained using ACM achieves state-of-the-art performance on video object segmentation benchmarks, demonstrating its potential for building object representations with the minimum amount of human supervision.

It would be extremely interesting to see if we can incorporate all the unsupervised components into a robot, such that it can move and detect objects around. Moreover, given the ability of the robot to interact with the environment, we should expect that semantics would appear based on the concept of objects, where all the information about affordance, utility, and dynamics is grounded. One step further, we would expect natural language to appear among robots if they are going to collectively pursue a goal in the form of adaptation.

REFERENCES

[ABC16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. "TensorFlow: A System for Large-Scale Machine Learning." In *OSDI*, volume 16, pp. 265–283, 2016. 53

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein gan." *arXiv preprint arXiv:1701.07875*, 2017. 101

[AP16] Aria Ahmadi and Ioannis Patras. "Unsupervised convolutional neural networks for motion estimation." In *Image Processing (ICIP), 2016 IEEE International Conference on*, pp. 1629–1633. IEEE, 2016. 26

[AS12a] Alper Ayvaci and Stefano Soatto. "Detachable object detection: Segmentation and depth ordering from short-baseline video." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **34**(10):1942–1951, 2012. 62, 64, 65, 75, 76, 78

[AS12b] Alper Ayvaci and Stefano Soatto. "Detachable Object Detection: Segmentation and Depth Ordering From Short-Baseline Video." *PAMI*, 2012. 83, 88

[BA93] Michael J Black and P Anandan. "A framework for the robust estimation of optical flow." In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pp. 231–236. IEEE, 1993. 24

[BA96] M.J. Black and P. Anandan. "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields." *Computer vision and image understanding*, **63**(1):75–104, 1996. 62, 65

[BBM09] Thomas Brox, Christoph Bregler, and Jitendra Malik. "Large displacement optical flow." In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 41–48. IEEE, 2009. 8, 9, 62

[BBP04] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. "High accuracy optical flow estimation based on a theory for warping." In *European conference on computer vision*, pp. 25–36. Springer, 2004. 27, 62, 65

[BBW06] Thomas Brox, Andrés Bruhn, and Joachim Weickert. "Variational Motion Segmentation with Level Sets." In *IEEE European Conference on Computer Vision (ECCV)*, pp. 471–483. 2006. 102

[BC06] G.J. Brostow and R. Cipolla. "Unsupervised Bayesian Detection of Independent Motion in Crowds." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2006. 102

[BC09] Thomas Brox and Daniel Cremers. "On local region models and a statistical interpretation of the piecewise smooth Mumford-Shah functional." *International journal of computer vision*, **84**(2):184–193, 2009. 70

[BDB13] Jim Braux-Zin, Romain Dupont, and Adrien Bartoli. "A general dense image matching framework combining direct and feature-based costs." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 185–192, 2013. 9

[BDR15] Zoya Bylinskii, Ellen M DeGennaro, Rishi Rajalingham, Harald Ruda, Jinxia Zhang, and John K Tsotsos. "Towards the quantitative evaluation of visual attention models." *Vision research*, **116**:258–268, 2015. 102

[BKC17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **39**(12):2481–2495, 2017. 103, 110

[BM92] Paul J Besl and Neil D McKay. "Method for registration of 3-D shapes." In *Robotics-DL tentative*, pp. 586–606. International Society for Optics and Photonics, 1992. 70

[BM98] Lothar Bergen and Fernand Meyer. "Motion Segmentation and Depth Ordering Based on Morphological Segmentation." In *ECCV*, 1998. 83

[BM11] Thomas Brox and Jitendra Malik. "Large displacement optical flow: descriptor matching in variational motion estimation." *IEEE transactions on pattern analysis and machine intelligence*, **33**(3):500–513, 2011. 37, 38

[BMC97] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. "A real-time computer vision system for measuring traffic parameters." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997. 102

[BMT05] M.F. Beg, M.I. Miller, A. Trouvé, and L. Younes. "Computing large deformation metric mappings via geodesic flows of diffeomorphisms." *International Journal of Computer Vision*, **61**(2):139–157, 2005. 67

[BSC00] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. "Image inpainting." In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417–424. ACM Press/Addison-Wesley Publishing Co., 2000. 62

[BSL11] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. "A database and evaluation methodology for optical flow." *International Journal of Computer Vision*, **92**(1):1–31, 2011. 24

[BTS15] Christian Bailer, Bertram Taetz, and Didier Stricker. "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation." In *Proceedings of the IEEE international conference on computer vision*, pp. 4015–4023, 2015. 8, 12, 13, 16, 17, 18, 19, 20

[BW05] Andres Bruhn and Joachim Weickert. "Towards ultimate motion estimation: Combining highest accuracy with real-time performance." In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pp. 749–755. IEEE, 2005. 33

[BWL] Linchao Bao, Baoyuan Wu, and Wei Liu. "CNN in MRF: Video Object Segmentation via Inference in A CNN-Based Higher-Order Spatio-Temporal MRF." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 105

[BWS05] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods." *International journal of computer vision*, **61**(3):211–231, 2005. 24

[BWS09] X. Bai, J. Wang, D. Simons, and G. Sapiro. "Video snapcut: robust video object cutout using localized classifiers." *ACM Transactions on Graphics (TOG)*, **28**(3):70, 2009. 62, 74

[BWS10] X. Bai, J. Wang, and G. Sapiro. "Dynamic color flow: a motion-adaptive color model for object segmentation in video." *ECCV 2010*, pp. 617–630, 2010. 74

[BWS12a] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. "A naturalistic open source movie for optical flow evaluation." In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pp. 611–625. Springer-Verlag, October 2012. 18

[BWS12b] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. "A naturalistic open source movie for optical flow evaluation." In *European Conference on Computer Vision*, pp. 611–625. Springer, 2012. 25, 34

[BYJ14] Linchao Bao, Qingxiong Yang, and Hailin Jin. "Fast edge-preserving patchmatch for large displacement optical flow." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3534–3541, 2014. 8

[CF13] Jason Chang and John W Fisher. "Topology-Constrained Layered Tracking with Latent Flow." In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 161–168. IEEE, 2013. 62, 75, 78

[CJL13] Zhuoyuan Chen, Hailin Jin, Zhe Lin, Scott Cohen, and Ying Wu. "Large displacement optical flow from nearest neighbor fields." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2443–2450, 2013. 8

[CK16] Qifeng Chen and Vladlen Koltun. "Full flow: Optical flow estimation by global optimization over regular grids." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4706–4714, 2016. 18, 27

[CPK18] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *IEEE transactions on pattern analysis and machine intelligence*, **40**(4):834–848, 2018. 1

[CS04] Daniel Cremers and Stefano Soatto. "Motion Competition: A Variational Approach to Piecewise Parametric Motion Segmentation." *IEEE International Journal of Computer Vision*, **62**(3):249–265, 2004. 102

[CS05] Daniel Cremers and Stefano Soatto. "Motion competition: A variational approach to piecewise parametric motion segmentation." *International Journal of Computer Vision*, **62**(3):249–265, 2005. 101

[CS12] Massimo Camplani and Luis Salgado. "Efficient spatio-temporal hole filling strategy for kinect depth maps." In *Three-dimensional image processing (3DIP) and applications Ii*, volume 8290, p. 82900E. International Society for Optics and Photonics, 2012. 44

[CS16] Nadav Cohen and Amnon Shashua. "Inductive bias of deep convolutional networks through pooling geometry." *arXiv preprint arXiv:1605.06743*, 2016. 25

[CS17] Pratik Chaudhari and Stefano Soatto. "Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks." *arXiv preprint arXiv:1710.11029*, 2017. 25, 26

[CSH11] Jan vCech, Jordi Sanchez-Riera, and Radu Horaud. "Scene flow estimation by growing correspondence seeds." In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 3129–3136. IEEE, 2011. 9

[CTW17a] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. "SegFlow: Joint Learning for Video Object Segmentation and Optical Flow." In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 102

[CTW17b] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. "SegFlow: Joint Learning for Video Object Segmentation and Optical Flow." In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 109

[CV01] Tony F Chan and Luminita A Vese. "Active Contours Without Edges." *IEEE TRANSACTIONS ON IMAGE PROCESSING*, **10**(2), 2001. 100

[CW13] Haw-Shiuan Chang and Yu-Chiang Frank Wang. "Superpixel-based large displacement optical flow." In *2013 IEEE International Conference on Image Processing*, pp. 3835–3839. IEEE, 2013. 8

[CWL18] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. "Deep Convolutional Compressed Sensing for LiDAR Depth Completion." *arXiv preprint arXiv:1803.08949*, 2018. 45

[CWY18] Xinjing Cheng, Peng Wang, and Ruigang Yang. "Depth Estimation via Affinity Learned with Convolutional Spatial Propagation Network." In *European Conference on Computer Vision*, pp. 108–125. Springer, Cham, 2018. 46, 54

[DAG15] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. "Long-term recurrent convolutional networks for visual recognition and description." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015. 1

[DCS13] Caglayan Dicle, Octavia I Camps, and Mario Sznaier. "The way they move: Tracking multiple targets with similar appearance." In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 2304–2311. IEEE, 2013. 62

[DFI15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. "Flownet: Learning optical flow with convolutional networks." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2758–2766, 2015. 24, 25, 29, 34, 36, 37, 40

[DKA95] Rachid Deriche, Pierre Kornprobst, and Gilles Aubert. "Optical-flow estimation while preserving its discontinuities: A variational approach." In *Asian Conference on Computer Vision*, pp. 69–80. Springer, 1995. 27

[DOR15] Tali Dekel, Shaul Oron, Michael Rubinstein, Shai Avidan, and William T Freeman. "Best-Buddies Similarity for robust template matching." In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2021–2029. IEEE, 2015. 9

[DVP18a] Martin Dimitrievski, Peter Veelaert, and Wilfried Philips. "Learning Morphological Operators for Depth Completion." In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 450–461. Springer, 2018. 45

[DVP18b] Martin Dimitrievski, Peter Veelaert, and Wilfried Philips. "Learning morphological operators for depth completion." In *Advanced Concepts for Intelligent Vision Systems*, 2018. 54

[DZ13] Piotr Dollár and C. Lawrence Zitnick. "Structured Forests for Fast Edge Detection." In *ICCV*, 2013. 87

[DZ15] Piotr Dollár and C Lawrence Zitnick. "Fast edge detection using structured forests." *IEEE transactions on pattern analysis and machine intelligence*, **37**(8):1558–1570, 2015. 12

[EFK18] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. "Propagating Confidences through CNNs for Sparse Data Regression." *arXiv preprint arXiv:1805.11913*, 2018. 45, 54, 56, 57

[EGS11] Ahmed Elnakib, Georgy Gimelfarb, Jasjit S Suri, and Ayman El-Baz. "Medical image segmentation: a brief survey." In *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*, pp. 1–39. Springer, 2011. 102

[ET06] Selim Esedog, Yen-Hsi Richard Tsai, et al. "Threshold dynamics for the piecewise constant Mumford–Shah functional." *Journal of Computational Physics*, **211**(1):367–384, 2006. 72

[FDI15] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. "Flownet: Learning optical flow with convolutional networks." *arXiv preprint arXiv:1504.06852*, 2015. 17

[FI14] Alon Faktor and Michal Irani. "Video Object Segmentation by Non-Local Consensus voting." In *British Machine Vision Conference (BMVC)*. British Machine Vision Association, 2014. 106, 107, 108

[FNP16] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. "Deepstereo: Learning to predict new views from the world's imagery." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5515–5524, 2016. 45

[GBC16] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. "Unsupervised cnn for single view depth estimation: Geometry to the rescue." In *European Conference on Computer Vision*, pp. 740–756. Springer, 2016. 45

[GCS04] Camillo Gentile, Octavia Camps, and Mario Sznaier. "Segmentation for robust tracking in the presence of severe occlusion." *Image Processing, IEEE Transactions on*, **13**(2):166–178, 2004. 62

[GKH10a] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. "Efficient hierarchical graph-based video segmentation." In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2141–2148. IEEE, 2010. 62, 75, 76, 77, 79

[GKH10b] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. "Efficient Hierarchical Graph-based Video Segmentation." In *CVPR*, 2010. 88

[GKH10c] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. "Efficient Hierarchical Graph Based Video Segmentation." In *CVPR*, 2010. 91

[GLS13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. "Vision meets Robotics: The KITTI Dataset." *International Journal of Robotics Research (IJRR)*, 2013. 16

[GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3354–3361. IEEE, 2012. 35, 45

[GMB17] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. "Unsupervised monocular depth estimation with left-right consistency." In *CVPR*, volume 2, p. 7, 2017. 45

[GNJ13] Fabio Galasso, S. Naveen, Tatiana J. Cardenas, Thomas Brox, and Bernt Schiele. "A Unified Video Segmentation Benchmark: Annotation, Metrics and Analysis." In *ICCV*, 2013. 87

[GPM14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In *Advances in neural information processing systems*, pp. 2672–2680, 2014. 41

[GWC16] A Gaidon, Q Wang, Y Cabon, and E Vig. "Virtual Worlds as Proxy for Multi-Object Tracking Analysis." In *CVPR*, 2016. 53

[GYP84] Carl E Granrud, Albert Yonas, and Linda Pettersen. "A comparison of monocular and binocular depth perception in 5-and 7-month-old infants." *Journal of experimental child psychology*, **38**(1):19–32, 1984. 2

[HFY18] Zixuan Huang, Junming Fan, Shuai Yi, Xiaogang Wang, and Hongsheng Li. "HMS-Net: Hierarchical Multi-scale Sparsity-invariant Network for Sparse Depth Completion." *arXiv preprint arXiv:1808.08685*, 2018. 45, 49, 54

[HS81] Berthold KP Horn and Brian G Schunck. "Determining optical flow." *Artificial intelligence*, **17**(1-3):185–203, 1981. 62, 65

[HSL] Yinlin Hu, Rui Song, and Yunsong Li. "Efficient Coarse-to-Fine PatchMatch for Large Displacement Optical Flow.". 8

[HSL16] Yinlin Hu, Rui Song, and Yunsong Li. "Efficient coarse-to-fine patchmatch for large displacement optical flow." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5704–5712, 2016. 17, 18

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 1, 50, 52

[IK00] Laurent Itti and Christof Koch. "A saliency-based search mechanism for overt and covert shifts of visual attention." *Vision research*, **40**(10-12):1489–1506, 2000. 102

[IMS17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. "FlowNet 2.0: Evolution of Optical Flow Estimation With Deep Networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2462–2470, 2017. 24, 36, 37

[JDW18] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. "Sparse and Dense Data with CNNs: Depth Completion and Semantic Segmentation." In *2018 International Conference on 3D Vision (3DV)*, pp. 52–60. IEEE, 2018. 46, 50, 54, 55, 56

[JG14] Suyog Dutt Jain and Kristen Grauman. "Supervoxel-Consistent Foreground Propagation in Video." In *Computer Vision–ECCV 2014*, pp. 656–671. Springer, 2014. 62, 75, 78

[JHD16] J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness." In *European Conference on Computer Vision*, pp. 3–10. Springer, 2016. 26, 37, 38, 40

[JXG17] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. "FusionSeg: Learning to Combine Motion and Appearance for Fully Automatic Segmentation of Generic Objects in Videos." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 106, 107

[KAB15] Margret Keuper, Bjoern Andres, and Thomas Brox. "Motion Trajectory Segmentation via Minimum Cost Multicuts." In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 106, 108

[Kau95] Franz Kaufmann. "Development of motion perception in early infancy." *European Journal of Pediatrics*, **154**(4):S48–S53, 1995. 2

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*, 2014. 35, 53

[KHW18] Jason Ku, Ali Harakeh, and Steven Lake Waslander. "In Defense of Classical Image Processing: Fast Depth Completion on the CPU." *CoRR*, **abs/1802.00036**, 2018. 45

[KK11] Philipp Krähenbühl and Vladlen Koltun. "Efficient inference in fully connected crfs with gaussian edge potentials." In *Advances in neural information processing systems*, pp. 109–117, 2011. 104

[KK12] Philipp Krähenbühl and Vladlen Koltun. "Efficient nonlocal regularization for optical flow." In *European Conference on Computer Vision*, pp. 356–369. Springer, 2012. 27

[KK17] Yeong Jun Koh and Chang-Su Kim. "Primary Object Segmentation in Videos Based on Region Augmentation and Reduction." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 106, 107, 108

[KTD16] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. "Fast optical flow using dense inverse search." In *European Conference on Computer Vision*, pp. 471–488. Springer, 2016. 37

[KW13] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114*, 2013. 33, 42

[Lin13] Tony Lindeberg. *Scale-space theory in computer vision*, volume 256. Springer Science & Business Media, 2013. 11

[LKG11] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. "Key-segments for video object segmentation." In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1995–2002. IEEE, 2011. 62, 75, 77, 78, 79

[LKH13] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. "Video segmentation by tracking many figure-ground segments." In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 2192–2199. IEEE, 2013. 62, 75, 79

[LMB15] Yu Li, Dongbo Min, Michael S Brown, Minh N Do, and Jiangbo Lu. "Spm-bp: Sped-up patchmatch belief propagation for continuous mrfs." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4006–4014, 2015. 18

[LRL14] Si Lu, Xiaofeng Ren, and Feng Liu. "Depth enhancement via low-rank matrix completion." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3390–3397, 2014. 44

[LS18] Dong Lao and Ganesh Sundaramoorthi. "Extending Layered Models to 3D Motion." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 435–451, 2018. 109

[Mar82] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Henry Holt and Co., Inc., New York, NY, USA, 1982. 92

[MAS13] Jonathan Masci, Jesús Angulo, and Jürgen Schmidhuber. "A learning framework for morphological operators using counter–harmonic mean." In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pp. 329–340. Springer, 2013. 45

[MCC18] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. "Video Object Segmentation Without Temporal Information." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 105

[MCK18] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. "Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera." *arXiv preprint arXiv:1807.00275*, 2018. 45, 49, 54, 55, 56, 57, 58, 59

[MG15] Moritz Menze and Andreas Geiger. "Object Scene Flow for Autonomous Vehicles." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 35

[MHR18] Simon Meister, Junhwa Hur, and Stefan Roth. "UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss." In *AAAI*, New Orleans, Louisiana, February 2018. 26, 33, 36, 37, 38, 39

[MIH16] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4040–4048, 2016. 36

[ML12] Tianyang Ma and Longin Jan Latecki. "Maximum weight cliques with mutex constraints for video object segmentation." In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 670–677. IEEE, 2012. 75, 78

[MS89] David Mumford and Jayant Shah. "Optimal approximations by piecewise smooth functions and associated variational problems." *Communications on pure and applied mathematics*, **42**(5):577–685, 1989. 101

[MWA18] Reza Mahjourian, Martin Wicke, and Anelia Angelova. "Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints." *arXiv preprint arXiv:1802.05522*, 2018. 45

[OMB14a] P. Ochs, J. Malik, and T. Brox. "Segmentation of Moving Objects by Long Term Video Analysis." *PAMI*, **36**(6), 2014. 87, 88, 89, 91

[OMB14b] Peter Ochs, Jitendra Malik, and Thomas Brox. "Segmentation of moving objects by long term video analysis." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **36**(6):1187–1200, 2014. 62, 75, 76, 77

[OMB14c] Peter Ochs, Jitendra Malik, and Thomas Brox. "Segmentation of Moving Objects by Long Term Video Analysis." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **36**(6):1187–1200, 2014. 102, 103, 104, 105, 106, 107

[PBB06] Nils Papenberg, Andrés Bruhn, Thomas Brox, Stephan Didas, and Joachim Weickert. "Highly accurate optic flow computation with theoretically justified warping." *International Journal of Computer Vision*, **67**(2):141–158, 2006. 24

[PF13] Anestis Papazoglou and Vittorio Ferrari. "Fast Object Segmentation in Unconstrained Video." In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 106, 108, 109

[PGA16] Cristiano Premebida, Luis Garrote, Alireza Asvadi, A Pedro Ribeiro, and Urbano Nunes. "High-resolution LIDAR-based depth mapping using bilateral filter." *arXiv preprint arXiv:1606.05614*, 2016. 45

[PL13] Yu Pang and Haibin Ling. "Finding the best from the second bests-inhibiting subjective bias in evaluation of visual tracking algorithms." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2784–2791, 2013. 107

[PPM16] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 102, 103, 104, 105, 106, 107

[PVP94] Marc Proesmans, Luc Van Gool, Eric Pauwels, and André Oosterlinck. "Determination of optical flow and its discontinuities using non-linear diffusion." In *European Conference on Computer Vision*, pp. 294–304. Springer, 1994. 27

[RB17] Anurag Ranjan and Michael J Black. "Optical Flow Estimation Using a Spatial Pyramid Network." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4161–4170, 2017. 24, 37, 39, 41

[RBP14] René Ranftl, Kristian Bredies, and Thomas Pock. "Non-local total generalized variation for optical flow estimation." In *European Conference on Computer Vision*, pp. 439–454. Springer, 2014. 27

[RDG16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016. 1

[RPY18] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. "SBNet: Sparse Blocks Network for Fast Inference." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8711–8720, 2018. 45

[RWH15a] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. "Deepmatching: Hierarchical deformable dense matching." *International Journal of Computer Vision*, pp. 1–24, 2015. 8

[RWH15b] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. "EpicFlow: Edge-preserving interpolation of correspondences for optical flow." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1164–1172, 2015. 8, 12, 17, 18, 21, 22

[RYN17] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. "Unsupervised Deep Learning for Optical Flow Estimation." In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 26, 36, 37

[SBM11] Patrik Sundberg, Thomas Brox, Michael Maire, Pablo Arbeláez, and Jitendra Malik. "Occlusion boundary detection and figure/ground assignment from optical flow." In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2233–2240. IEEE, 2011. 62, 78

[SC13] Ju Shen and Sen-Ching S Cheung. "Layer depth denoising and completion for structured-light rgb-d cameras." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1187–1194, 2013. 44

[Set96] James A Sethian. “A fast marching level set method for monotonically advancing fronts.” *Proceedings of the National Academy of Sciences*, **93**(4):1591–1595, 1996. 70

[SF11] Nathan Silberman and Rob Fergus. “Indoor scene segmentation using a structured light sensor.” In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 601–608. IEEE, 2011. 45

[SHK12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. “Indoor segmentation and support inference from rgbd images.” In *European Conference on Computer Vision*, pp. 746–760. Springer, 2012. 45

[SM98] Jianbo Shi and J. Malik. “Motion segmentation and tracking using normalized cuts.” In *IEEE International Conference on Computer Vision (ICCV)*, 1998. 102

[SM12] Charles Sutton, Andrew McCallum, et al. “An introduction to conditional random fields.” *Foundations and Trends® in Machine Learning*, **4**(4):267–373, 2012. 27

[SRB10] Deqing Sun, Stefan Roth, and Michael J Black. “Secrets of optical flow estimation and their principles.” In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2432–2439. IEEE, 2010. 37, 38, 39, 74

[SRB14] Deqing Sun, Stefan Roth, and Michael J Black. “A quantitative analysis of current practices in optical flow estimation and the principles behind them.” *International Journal of Computer Vision*, **106**(2):115–137, 2014. 8, 13, 37, 40, 41

[SSP16] Nick Schneider, Lukas Schneider, Peter Pinggera, Uwe Franke, Marc Pollefeys, and Christoph Stiller. “Semantically guided depth upsampling.” In *German Conference on Pattern Recognition*, pp. 37–48. Springer, 2016. 45

[SWS13a] Deqing Sun, Jonas Wulff, Erik B Sudderth, Hanspeter Pfister, and Michael J Black. “A fully-connected layered model of foreground and background flow.” In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2451–2458. IEEE, 2013. 62

[SWS13b] Deqing Sun, Jonas Wulff, Erik B. Sudderth, Hanspeter Pfister, and Michael J. Black. “A Fully-Connected Layered Model of Foreground and Background Flow.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 102

[SWZ18] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. “Pyramid Dilated Deeper ConvLSTM for Video Salient Object Detection.” In *IEEE European Conference on Computer Vision (ECCV)*. 2018. 102, 106, 108, 109

[SY12] Ganesh Sundaramoorthi and Yanchao Yang. "Matching through features and features through matching." *arXiv preprint arXiv:1211.4771*, 2012. 8

[SYL18] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 104, 108

[SYM07] Ganesh Sundaramoorthi, Anthony Yezzi, and Andrea C Mennucci. "Sobolev active contours." *International Journal of Computer Vision*, **73**(3):345–366, 2007. 62, 65

[SZS08] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. "A comparative study of energy minimization methods for markov random fields with smoothness-based priors." *IEEE transactions on pattern analysis and machine intelligence*, **30**(6):1068–1080, 2008. 14

[TAS17a] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. "Learning Motion Patterns in Videos." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 102, 109

[TAS17b] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. "Learning Video Object Segmentation with Visual Memory." In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 102, 106

[TFN12] David Tsai, Matthew Flagg, Atsushi Nakazawa, and James M Rehg. "Motion coherent tracking using multi-label MRF optimization." *International journal of computer vision*, **100**(2):190–202, 2012. 75

[TFR10] David Tsai, Matthew Flagg, and James Rehg. "Motion Coherent Tracking with Multi-label MRF optimization." In *British Machine Vision Conference (BMVC) 2010*. British Machine Vision Association, 2010. 103, 104, 105, 106, 107

[TKS15a] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. "Causal Video Object Segmentation From Persistence of Occlusions." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. 62, 71, 75, 76, 77, 78

[TKS15b] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. "Causal video object segmentation from persistence of occlusions." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4268–4276, 2015. 109

[TKS15c] Brian Taylor, Vasiliy Karasev, and Stefano Soattoc. "Causal Video Object Segmentation from Persistence of Occlusions." In *CVPR*. IEEE, 2015. 88, 89, 91

[TV15] Radu Timofte and Luc Van Gool. "Sparse flow: Sparse matching for small to large displacement optical flow." In *2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 1100–1106. IEEE, 2015. 8, 9

[USS17] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. "Sparsity Invariant CNNs." *2017 International Conference on 3D Vision (3DV)*, pp. 11–20, 2017. 45, 49, 54, 55

[VP89] Alessandro Verri and Tomaso Poggio. "Motion field and optical flow: Qualitative properties." *IEEE Transactions on pattern analysis and machine intelligence*, **11**(5):490–498, 1989. 10

[WA94a] John Y.A. Wang and Edward H. Adelson. "Representing Moving Images with Layers." *TIP*, 1994. 61, 62, 83

[WA94b] J.Y.A. Wang and E.H. Adelson. "Representing moving images with layers." *IEEE Transactions on Image Processing*, **3**(5):625–638, 1994. 102

[Wat96] John Wattam-Bell. "Visual motion processing in one-month-old infants: Preferential looking experiments." *Vision Research*, **36**(11):1671–1677, 1996. 2

[WB15] Jonas Wulff and Michael J Black. "Efficient sparse-to-dense optical flow estimation using a learned basis and layers." In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 120–130. IEEE, 2015. 8, 9

[WBS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. "Image quality assessment: from error visibility to structural similarity." *IEEE transactions on image processing*, **13**(4):600–612, 2004. 51

[WBZ18] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. "Learning Depth from Monocular Videos using Direct Methods." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2022–2030, 2018. 45

[WC11] Andreas Wedel and Daniel Cremers. "Optical flow estimation." In *Stereo Scene Flow for 3D Motion Analysis*, pp. 5–34. Springer, 2011. 9

[WDL15] Longyin Wen, Dawei Du, Zhen Lei, Stan Z. Li, and Ming-Hsuan Yang. "JOTS: Joint Online Tracking and Segmentation." In *Computer Vision and Pattern Recognition, IEEE Conference on*, 2015. 75, 78, 79

[WRH13] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. "Deepflow: Large displacement optical flow with deep matching." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1385–1392, 2013. 8, 17, 18

[WSP15] Wenguan Wang, Jianbing Shen, and Fatih Porikli. "Saliency-aware geodesic video object segmentation." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 106, 107, 108

[XBM04] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. "Real-time combined 2D+ 3D active appearance models." In *CVPR (2)*, pp. 535–542, 2004. 62

[XDJ12] Li Xu, Zhenlong Dai, and Jiaya Jia. "Scale invariant optical flow." In *Computer Vision–ECCV 2012*, pp. 385–399. Springer, 2012. 9

[XGF16] Junyuan Xie, Ross Girshick, and Ali Farhadi. "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks." In *European Conference on Computer Vision*, pp. 842–857. Springer, 2016. 45

[XJM12] Li Xu, Jiaya Jia, and Yasuyuki Matsushita. "Motion detail preserving optical flow estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(9):1744–1757, 2012. 9, 24, 27

[Xu99] Fei Xu. "Object individuation and object identity in infancy: The role of spatiotemporal information, object property information, and language." *Acta psychologica*, **102**(2-3):113–136, 1999. 2

[XXC12] Chenliang Xu, Caiming Xiong, and Jason J Corso. "Streaming hierarchical video segmentation." In *Computer Vision–ECCV 2012*, pp. 626–639. Springer, 2012. 62

[YL15] Jiaolong Yang and Hongdong Li. "Dense, accurate optical flow estimation with piecewise parametric model." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1019–1027, 2015. 9

[YLS15] Yanchao Yang, Zhaojin Lu, and Ganesh Sundaramoorthi. "Coarse-to-fine region selection and matching." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5051–5059, 2015. 8

[YLS19] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. "Unsupervised Moving Object Detection via Contextual Information Separation." *arXiv preprint arXiv:1901.03360*, 2019. 5

[YS15] Yanchao Yang and Ganesh Sundaramoorthi. "Shape tracking with occlusions via coarse-to-fine region-based sobolev descent." *IEEE transactions on pattern analysis and machine intelligence*, **37**(5):1053–1066, 2015. 62, 64, 66, 71, 105

[YS17] Yanchao Yang and Stefano Soatto. "S2F: Slow-to-fast interpolator flow." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2087–2096, 2017. 5

[YS18] Yanchao Yang and Stefano Soatto. "Conditional prior networks for optical flow." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 271–287, 2018. 5, 46, 47, 52, 104, 110

[YSS15a] Yanchao Yang, Ganesh Sundaramoorthi, and Stefano Soatto. "Self-Occlusions and Disocclusion in Causal Video Object Segmentation." In *ICCV*, 2015. 82, 88, 91

[YSS15b] Yanchao Yang, Ganesh Sundaramoorthi, and Stefano Soatto. "Self-occlusions and disocclusions in causal video object segmentation." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4408–4416, 2015. 5, 102

[YWS19] Yanchao Yang, Alex Wong, and Stefano Soatto. "Dense Depth Posterior (DDP) from Single Image and Sparse Range." *arXiv preprint arXiv:1901.10034*, 2019. 5

[ZBS17] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. "Unsupervised learning of depth and ego-motion from video." In *CVPR*, volume 2, p. 7, 2017. 45

[ZF18] Yinda Zhang and Thomas Funkhouser. "Deep Depth Completion of a Single RGB-D Image." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 175–185, 2018. 45

[ZN17] Yi Zhu and Shawn Newsam. "DenseNet for dense flow." *arXiv preprint arXiv:1707.06316*, 2017. 37, 38

[ZPB07] C. Zach, T. Pock, and H. Bischof. "A duality based approach for realtime TV-L 1 optical flow." *Pattern Recognition*, pp. 214–223, 2007. 62, 65

[ZVS] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. "Learning transferable architectures for scalable image recognition.". 50