UNIVERSITAT POLITÈCNICA DE CATALUNYA

Programa de Doctorat:

AUTOMÀTICA, ROBÒTICA I VISIÓ

Tesi Doctoral

Enhancing low-level features with mid-level cues

Eduard Trulls Fortuny

Directors:
Francesc Moreno Noguer
Alberto Sanfeliu Cortés

Febrer 2015

Abstract

Local features have become an essential tool in visual recognition. Much of the progress in computer vision over the past decade has built on simple, local representations such as SIFT or HOG. SIFT in particular shifted the paradigm in feature representation. Subsequent works have often focused on improving either computational efficiency, or invariance properties.

This thesis arguably belongs to the latter group. Invariance is a particularly relevant aspect if we intend to work with *dense* features—extracted for every pixel in an image. The traditional approach to *sparse* matching is to rely on stable interest points, such as corners, where scale and orientation can be reliably estimated, thus enforcing invariance. This is not applicable to dense features, which need to be computed on arbitrary points. Dense features have been shown to outperform sparse matching techniques in many recognition problems, and form the bulk of our work.

In this thesis we present strategies to enhance low-level, local features with mid-level, global cues. We devise techniques to construct better features, and use them to handle complex ambiguities, occlusions and background changes. To deal with ambiguities, we explore the use of motion to enforce temporal consistency with optical flow priors. We also introduce a novel technique to exploit segmentation cues, and use it to extract features invariant to background variability. For this, we downplay image measurements most likely to belong to a region different from that where the descriptor is computed. In both cases we follow the same strategy: we incorporate mid-level, 'big picture' information into the construction of local features, and proceed to use them in the same manner as we would the baseline features.

We apply these techniques to different feature representations, including SIFT and HOG, and use them to address canonical vision problems such as stereo and object detection, demonstrating that the introduction of global cues yields consistent improvements. We prioritize solutions that are simple, general, and efficient.

Our main contributions are as follows: (a) An approach to dense stereo reconstruction with spatiotemporal features, which unlike existing works remains applicable to wide baselines. (b) A technique to exploit segmentation cues to construct dense descriptors invariant to background variability, such as occlusions or background motion. (c) A technique to integrate bottom-up segmentation with recognition efficiently, amenable to sliding window detectors.

Keywords: feature descriptors, dense features, wide-baseline stereo, optical flow, segmentation, object detection, deformable part models.

Contents

Contents iii

1 Introduction 1
- 1.1 Objectives . . . 3
- 1.2 Methodology . . . 5
- 1.3 Contributions . . . 9
  - 1.3.1 Publications . . . 9
- 1.4 Thesis Overview . . . 10

2 Overview 11
- 2.1 Components . . . 11
- 2.2 Low-level features . . . 12
  - 2.2.1 SIFT . . . 15
  - 2.2.2 Daisy . . . 16
  - 2.2.3 Stequel . . . 18
  - 2.2.4 SID . . . 19
  - 2.2.5 Scale-less SIFT (SLS) . . . 22
  - 2.2.6 Metric learning and binary descriptors . . . 23
  - 2.2.7 Histograms of Oriented Gradients . . . 24
- 2.3 Mid-level cues . . . 26
  - 2.3.1 Optical flow . . . 26
  - 2.3.2 Segmentation . . . 28
- 2.4 Applications . . . 33
  - 2.4.1 Wide-baseline stereo . . . 33
  - 2.4.2 Large-displacement motion . . . 37
  - 2.4.3 Object detection . . . 38

3 A spatiotemporal approach to wide-Baseline stereo 41
- 3.1 Introduction . . . 41
- 3.2 Related work . . . 44
- 3.3 Spatiotemporal descriptor . . . 45
- 3.4 Depth Estimation with spatiotemporal constraints . . . 48
- 3.5 Handling occlusions with spatial masks . . . 49

3.6 Computational cost and implementation details . . . 52
3.7 Experimental evaluation . . . 52
3.7.1 Parameter selection . . . 53
3.7.2 Wide baseline experiments . . . 54
3.7.3 Image noise experiments . . . 54
3.7.4 Experiments with occlusion masks . . . 55
3.7.5 Experiments with real sequences . . . 58
3.8 Summary and future work . . . 59

4 Dense segmentation-aware descriptors **63**
4.1 Introduction . . . 63
4.2 Related work . . . 65
4.3 Segmentation cues . . . 66
4.4 Segmentation-aware descriptor construction . . . 68
4.5 Experimental evaluation . . . 70
4.5.1 Large displacement, multi-layered motion . . . 70
4.5.2 Wide-baseline stereo . . . 72
4.6 Summary and future work . . . 78

5 Segmentation-aware DPMs **79**
5.1 Introduction . . . 79
5.2 Related work . . . 83
5.3 Deformable Part Models . . . 84
5.4 Superpixel-grounded DPMs . . . 85
5.4.1 Segmentation mask 'alpha-blending' . . . 86
5.5 Superpixel-grounded descriptors . . . 88
5.6 Experimental evaluation . . . 90
5.6.1 Object detection on the PASCAL VOC . . . 90
5.6.2 Large-displacement motion . . . 90
5.7 Summary and future work . . . 91

6 Concluding remarks **97**
6.1 Future work . . . 98
6.2 Current trends in recognition . . . 99

Bibliography **101**

List of Figures **117**

List of Tables **119**

Acknowledgements

Thanks to my advisors Francesc and Alberto, and also to Iasonas, who was an advisor in all but name.
Thanks to my family and friends for their support in such a long journey. You know who you are.

This work has been partly sponsored by the following:

- A four-year PhD scholarship by Universitat Politècnica de Catalunya.
- A three-month scholarship to visit the Center for Visual Computing in Paris in 2012, by the Ministerio de Educación y Ciencia of the Spanish Government (MHE2011).
- A three-month scholarship to visit the Center for Visual Computing in Paris in 2013, by the Generalitat de Catalunya (BE-DGR 2012).
- Projects Rob-TaskCoop (DPI2010-17112), PAU+ (DPI2011-27510), MIPRCV (Consolider-Ingenio 2010, CSD2007-00018), ARCAS (FP7-ICT-2011-287617), the ERA-net CHISTERA project VISEN (PCIN-2013-047), grant ANR-10-JCJC-0205, MOBOT (FP7-ICT-2011-600796).

Chapter 1

Introduction

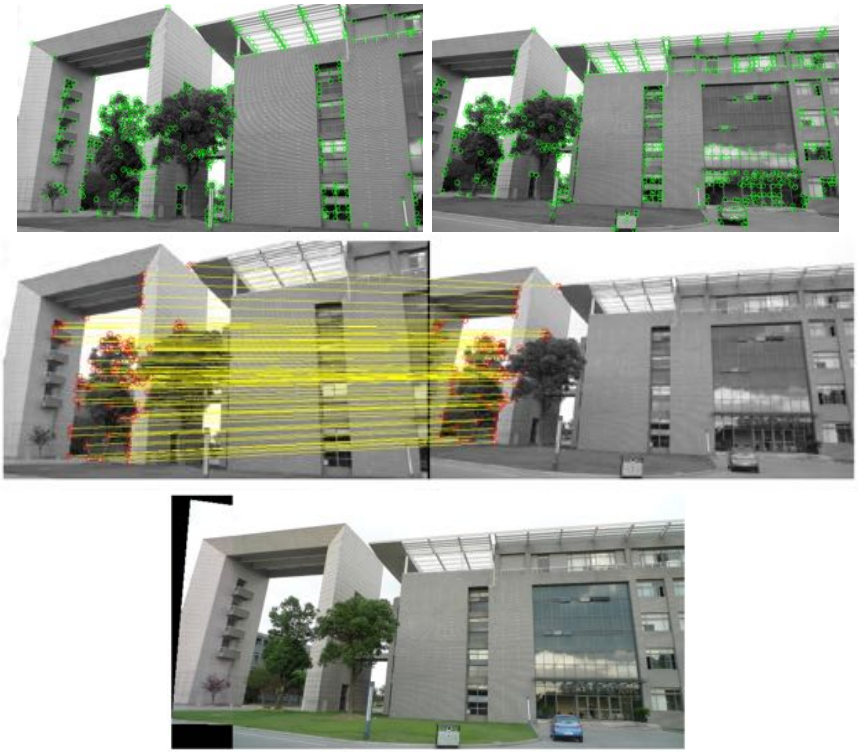
In the early stages of artificial intelligence research, in the 1950s and 60s, it was conjectured that the perception problem could be solved in a decade or so. It is a well-known anecdote that in 1966, artificial intelligence pioneer Marvin Minsky assigned a first-year undergraduate student, as a summer problem, "connect a television camera to a computer and get the machine to describe what it sees." (Crevier, 1994). This story illustrates that the complexity of the problem was grossly overestimated. Many years later, and despite the best efforts or a growing community of researchers, computer vision is far from a 'solved' problem. Extracting high-level information, e.g. what is a person doing and why, from image data, i.e. raw matrices of numbers, has proven a formidable task.

After the success experienced in recent years regarding the application of convolutional neural networks to computer vision problems[1], most notably by Hinton's group (Krizhevsky et al., 2012), there is a growing belief in some circles of our community that we may finally be on the verge of overcoming what we can call "feature engineering". Why painstakingly design hand-crafted, compact, invariant feature representations when we can learn them from large amounts of raw image data with models capable of accommodating millions of parameters? While we share this hope, we also believe that tools such as feature descriptors will remain useful for a long time to come.

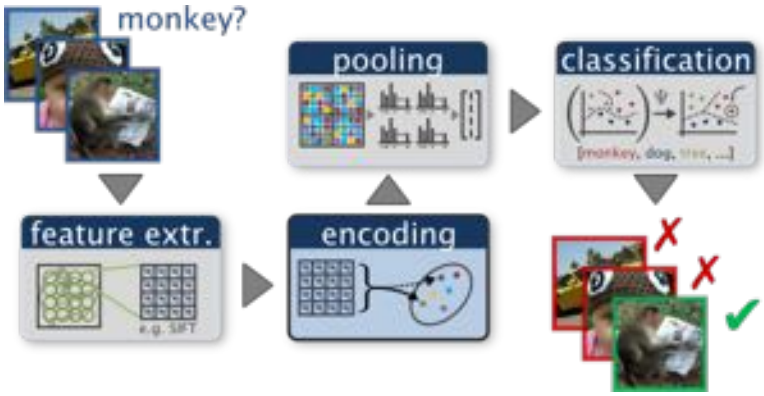
There is plenty of evidence to support this thesis. Much of the progress in visual recognition over the past decade has built on local, low-level features such as SIFT (Lowe, 2004) or HOG (Dalal and Triggs, 2005). The SIFT paper remains the most cited work in computer vision since the turn of the century, by a considerable margin[2]. New invariant descriptors are introduced every year in the major computer vision conferences. Many, if not most, modern computer vision applications rely on low-level features, and SIFT and its successors have become indispensable tools in matching,

[1] Convolutional Neural Networks (CNN) are a type of Artificial Neural Network (ANN) that was popular in the 90s, but fell out of fashion until Geoffrey Hinton's group introduced an architecture which performed exceedingly well on the 2012 Imagenet challenge. CNNs are arguably one of the 'hottest' topics in computer vision at the time of this writing, and must be acknowledged in any comprehensive study on feature representations. We will discuss them in chapter 6.

[2] According to Google Scholar rankings and to the best of our ability.



(a) Sparse feature matching with outlier detection. **Top row:** Local features are computed for interest points, extracted with the Harris detector (Harris and Stephens, 1988). **Middle row:** We can use RANSAC or similar strategies to estimate the homography between a pair of images while discarding outliers. **Bottom row:** The images can be aligned and stiched together using said homography. By Lin Zhang.



(b) A modern approach to image classification. Descriptors are extracted densely, at arbitrary scales, and clustered into visual words. Geometry can be preserved, to some extent, pooling with spatial pyramids. The features are then stacked and fed to a SVM classifier. Reproduced from the project website for (Chatfield et al., 2011).

Figure 1.1: Two practical examples that showcase the use of feature descriptors in computer vision applications: (a) stereo matching with sparse features, and (b) image classification with dense features. Both rely on SIFT descriptors from VLFEAT (Vedaldi and Fulkerson, 2008).

recognition, and retrieval. The design of optimal feature descriptors remains a major topic of research in computer vision. Features matter.

Local features based on small patches such as SIFT are very discriminative, but they don't give us the 'whole picture'. They are usually paired with tools or algorithms that work at the image level. For image recognition we can e.g. build histograms of features (Csurka et al., 2004; Lazebnik et al., 2006), with classifiers on top. Features are also indispensable in geometric computer vision. For instance, in stereo reconstruction we can apply tools such as RANSAC to discard outliers (Fischler and Bolles, 1981) or graph cuts to enforce piecewise-smoothness (Boykov et al., 2001). Similarly, 3D pose or homography estimation relies on accurate feature correspondences. Fig. 1.1 shows two examples of how we use low-level features for higher-level reasoning in computer vision applications.

1.1 Objectives

Features are thus the foundation for many computer vision applications. They have proved very reliable; so much so that they are often taken as-is, despite their potential shortcomings. Most research is focused on using features *in smarter ways*, rather than building *smarter features*. In practice, many state-of-the-art applications in 2014 still rely on SIFT (the original paper being published in 1999) or HOG (2005). Most research *on features* is focused on building faster or more compact representations, often leaving invariance a secondary consideration. Our question is: can we feed higher-level information back into the features to make them more reliable and discriminative? The main goal of this thesis is to investigate strategies to enhance local, low-level features with global, mid-level cues, building on the large body of work on feature design and application.

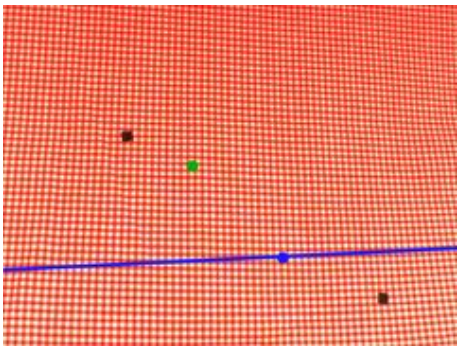
We illustrate the kind of problems we intend to address in Figs. 1.2-1.4. Fig. 1.2 demonstrates the problem of matching very complex ambiguities, where multiple false positives are unavoidable. Notice that this problem would persist even if we were to consider large, and therefore more informative, patches, given the regularity of the image structures; we must find other means to disambiguate. Fig. 1.3 shows the effect of background structures creeping into local features computed around object boundaries. This problem is related to occlusions in stereo reconstruction, pictured in Fig. 1.4, which are inevitable for wide camera baselines.

Additionally, many such techniques work only over reliable, *sparse* features. In this thesis we are primarily interested in *dense* features, i.e. those computed for every pixel in the image, a recent trend which has proven very powerful in stereo and recognition.

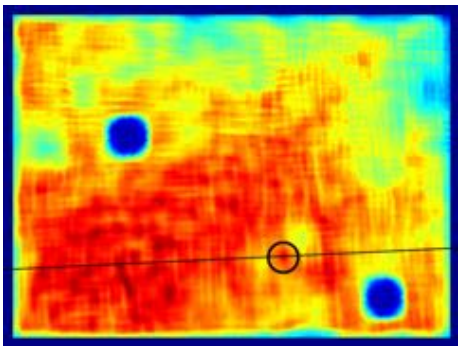
To summarize, the objectives laid out at the beginning of this thesis were as follows:

- Our main goal is to investigate novel techniques to leverage global, mid-level cues into the construction of local features.
- Rather than reinvent the wheel, we would rather build on popular, reliable features—while monitoring the state of the art on descriptors and algorithms.
- We are primarily interested in dense features.
- Our findings should be applied to relevant computer vision problems, likely in an application-dependent manner.
- The resulting, enhanced features should be as transparent as possible to the user.

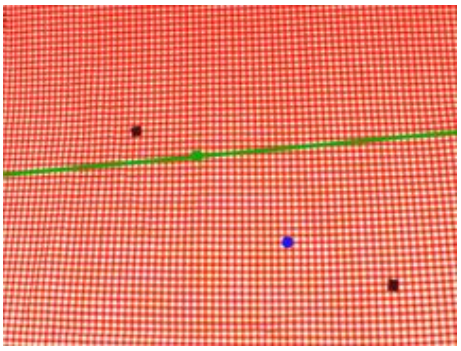
(a) Data acquisition



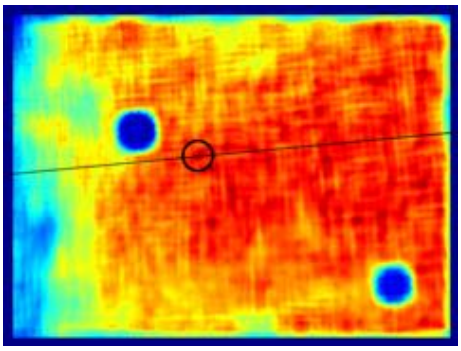
(b) I_l, $\mathbf{x_1^l}$ (green), $\mathbf{x_2^l}$ (blue)



(d) Similarity for $\mathbf{D_r}(\mathbf{x_2^r})$ and $\mathbf{D_l}(\mathbf{x}), \forall \mathbf{x} \in I_l$



(c) I_r, $\mathbf{x_2^r}$ (blue), $\mathbf{x_1^r}$ (green)



(e) Similarity for $\mathbf{D_l}(\mathbf{x_1^l})$ and $\mathbf{D_r}(\mathbf{x}), \forall \mathbf{x} \in I_r$

Figure 1.2: Matching with complex ambiguities. **Top row:** Two images of the same scene from a different viewpoint, as pictured in **(a)**: **(b)** left image I_l; **(c)** right image I_r. We plot two arbitrary, non-corresponding points: $\mathbf{x_1^l}$ over the left image, in green, and $\mathbf{x_2^r}$ over the right image, in blue. We show their corresponding epipolar lines and the ground truth match over the image plane of the other camera. The correct match must lie on the epipolar line of the same color (see Sec. 2.4.1.1 for details on epipolar geometry). We compute dense SIFT descriptors $\mathbf{D_{\{l,r\}}}$, aligned with the epipolar lines to enforce rotation invariance. On **(d)** we show the similarity between the descriptor for $\mathbf{x_2^r}$ and every descriptor over I_l (dark red is better). Notice that while we obtain a good match for the correct point, marked with a circle, other points also obtain a good score; even over the small subset determined by the epipolar lines. On **(e)** we show its counterpart for $\mathbf{x_1^l}$ and I_r.

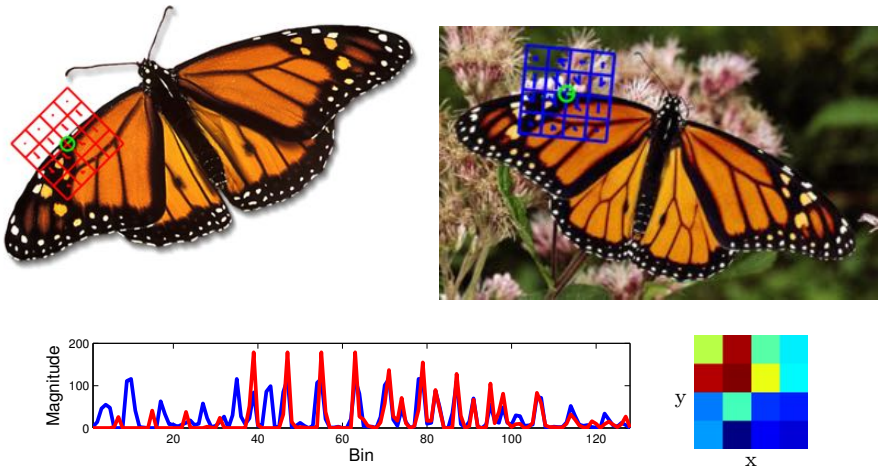


Figure 1.3: Matching with background interference. We show images of two monarch butterflies, strikingly similar in appearance and pose, one over a white background and another in the wild. We plot the SIFT descriptors for two corresponding points found by hand, in red (left image) and blue (right image), close to object boundaries. The arrow on the green circle indicates the keypoint's orientation. On the bottom left we plot the descriptor values, ordered: orientation bin first, x-bin second, y-bin third. On the bottom right we plot the similarity between descriptors, averaged over the orientation bins (blue is better, red is worse). The orientation histograms are similar for foreground cells but not for background cells. This hurts correspondence matching around object boundaries.

The types of features, cues and applications are strongly correlated. In Sec. 1.2 we explore these design goals in further detail and substantiate our choices, and in Sec. 1.3 we present our primary contributions.

1.2 Methodology

Our work is focused on local, low-level features. While research in this area remains strong, new publications are often variations of existing works. Most feature descriptors appearing after SIFT follow its template, and aim to improve either efficiency, e.g. the trade-off between accuracy and computational concerns, or invariance. Two recent trends are dense descriptors and binary descriptors. Most dense descriptors are also variations of SIFT, with dense SIFT arguably being the most popular. Binary descriptors can either be computed from real-valued descriptors, by learning the most effective transformations, or computed directly from image patches—while they do offer a new perspective their performance often lags behind SIFT-type descriptors, and have found relatively little use outside of real-time applications. We will explore the state of the art in feature descriptors in the following chapter.

In this thesis we approach the problem from a different angle. We build on well-known, time-tested features such as SIFT or HOG, and enhance them with higher-level information. In the following chapters we present three different techniques which all contain three distinct elements:

Figure 1.4: Matching with occlusions. **Top row:** Two images from wide-baseline stereo, with large occluded areas. **Bottom row:** On the left, we show the ground truth depth map, from the viewpoint of the rightmost camera. Occlusions, determined from ground truth visibility maps, are shown in red. On the right, we dilate the occlusion map by 25 pixels, and plot the result in yellow. We can think of these as the coordinates of the pixels that will be affected if we compute descriptors over a circular area of radius 25 pixels, as occluded areas creep into their domain—pixels closer to occlusion boundaries will suffer more. As the baseline increases, more and more pixels suffer from this problem: over 30% visible pixels in this case.

- 'Low-level' features: *local*, intensity-based features such as SIFT or HOG. They take an image patch as input and transform it into a high-dimensional representation that is both discriminative and invariant to common transformations, such as lighting or perspective changes.
- 'Mid-level' cues: representations extracted at a higher abstration level. In this work, we rely on *image-level* data such as optical flow or segmentation. They give us a low-dimensional embedding, such as velocity or region membership, for every pixel in the image.
- Target application: a computer vision problem which relies on low-level features, such as stereo or object detection.

Computer vision applications often rely on local features along with optimization or machine learning techniques. For instance, in dense stereo we can use feature descriptors to determine the most likely matches and global optimization techniques to find a good trade-off between per-pixel accuracy and piecewise-smooth depth estimates, as well as occlusions (see Sec. 2.4.1). Most research takes place at higher abstraction levels, particularly for complex problems such as object recognition 'in the wild', which nevertheless rely on local features. Good features are thus an absolute necessity.

In our approach, we add an intermediate step. Rather than attempting to build new features from the same image patches, we bring into play mid-level cues wich con-

tain image-level information, such as segmentation, and use them to enhance existing features. Continuing with the stereo example, we propose to use segmentation information to separate areas likely to belong to different objects. Taking this information into account while constructing the descriptors allows us to deal with scenarios such as that portrayed in Fig. 1.4; our work on segmentation-aware features is presented in chapters 4 and 5.

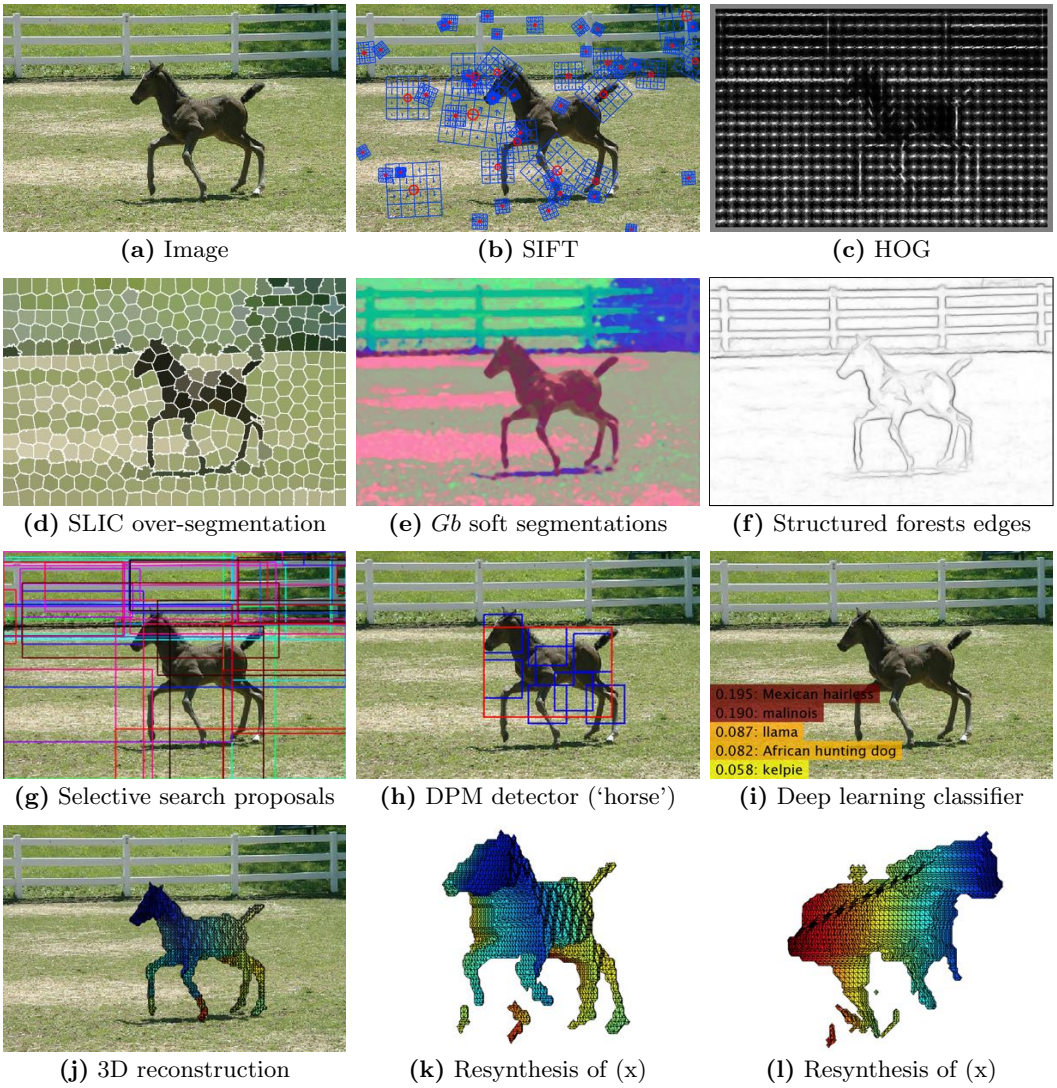
Our enhanced features are of the same type as the original features, and can thus be plugged into existing frameworks without many (or any) adjustments. The mid-level cues are naturally *application-dependent*. In chapter 2 we give an overview of all the technologies used in our work, breaking them into low-level features, mid-level cues and application-level tools.

Fig. 1.5 illustrates how different computer vision algorithms fit into this classification. Features are extracted and inferences made at different abstraction levels, for the same image, which is pictured in (a); note that we do not use all of them in our work, and not all those we use are pictured.

- Images (b-c) correspond to local, low-level, gradient-based features. These are the kind of **low-level features** we build on.
- Processes (d-f) operate at the image level and make inferences about pixel groupings in the image, without delving into semantic classification or 3D geometry. (d) shows an over-segmentation of the image into superpixels. (e) shows a 'soft' segmentation, where the similarity between pixels is encoded in a low-dimensional space which contains global information about image regions (in practice there are 10 such soft segmentations; we show the first three in RGB space). In (f) we show boundaries extracted with a state-of-the-art detector. (g) shows image windows likely to contain objects, an approach that is sometimes favored over exhaustive search due to computational constraints. These are the kind of **mid-level cues** we enhance local features with.
- The images in (h) and (i) show the results for two filters trained, respectively, for object detection (Deformable Part Models) and fine-grained image classification (Convolutional Neural Networks). (j-l) show a class-based 3D reconstruction of the object (a foal), obtained with a single image. These are **target applications**.

We can trace a parallel between this classification and the hierarchical architectures used to break the computer vision problem into smaller parts. Artificial vision has been studied from many different perspectives, including psychology, neuroscience and computation. Some models are inspired by modern developments in our understanding of primate vision, such as Marr's serminal work in computational neuroscience (Marr, 1982), while others tackle the problem from an image processing viewpoint. In a modern, 'practical' division of the stages of computer vision, 'low-level' or 'early' vision tasks involve operating on raw pixel data to produce useful features, such as noise reduction or edge detection, while 'mid-level' vision involves making inferences in a general way to reconstruct parts of the scene, such as determining its geometry, camera/object motion, or the location and pose of objects contained within. Finally, 'high-level' vision involves symbolic representations of the image, such as the recognition of specific objects and events. While the latter is clearly is our ultimate goal, holistic image understanding remains unsolved.

Many of the works showcased in Fig. 1.5 rely on others, also pictured. In this thesis we present novel new ways to combine image representations extracted at different



(a) Image **(b)** SIFT **(c)** HOG

(d) SLIC over-segmentation **(e)** *Gb* soft segmentations **(f)** Structured forests edges

(g) Selective search proposals **(h)** DPM detector ('horse') **(i)** Deep learning classifier

(j) 3D reconstruction **(k)** Resynthesis of (x) **(l)** Resynthesis of (x)

Figure 1.5: Features and inferences in computer vision, at varying levels of complexity. **(a)** An image from the PASCAL VOC. **(b)** SIFT descriptors (blue) on interest points (red) (Lowe, 2004). **(c)** HOG features (Dalal and Triggs, 2005). **(d)** Over-segmentation with SLIC superpixels (Achanta et al., 2012). **(e)** Soft segmentations from Leordeanu et al. (2012); we show the first three segmentations, in RGB space. **(f)** Boundaries computed with the structured forests detector of Dollár and Zitnick (2013). **(g)** The top 50 bounding-box object proposals with the selective search algorithm of Uijlings et al. (2013). **(h)** Object detection with Deformable Part Models (Felzenszwalb et al., 2010b), for the 'horse' filter, trained over PASCAL VOC. The bounding box for the object is marked in red and the blue boxes indicate the location of the parts. **(i)** Image classification with Caffe (Jia, 2013), trained over Imagenet; the top 5 labels and their respective scores overlaid on the image (all of which except for 'llama' are dog breeds). Images **(j-l)** show the dense, per-object 3D reconstructions of Vicente et al. (2014), over the image (j), and resynthesized over two different viewpoints (k-l); colors encode depth, with blue being closest to the camera and red the farthest.

abstraction levels. In the 'big picture', we can see it as a small step in the line of interactive hierarchical models, where lower-level processes initiate higher-level processes which in turn feed back into lower-level processes (Rumelhart et al., 1986).

1.3 Contributions

There are three primary contributions in this thesis:

1. We present an approach to build spatiotemporal features for 3D stereo reconstruction. Unlike existing works based on the spatiotemporal volume, defined by the concatenation of frames across time, our approach does not require a knowledge of the geometry of the scene and is thus applicable to wide-baseline stereo. This work was published in (Trulls et al., 2012).
2. We present an approach to exploit segmentation cues to construct dense features that can robustly deal with occlusion and background changes. To accomplish this we build soft segmentation masks for every pixel, and use them to downplay measurements coming from areas likely to belong to different regions. This work was published in (Trulls et al., 2013). Additionally, in this thesis we extend this technique to boundary cues.
3. We propose a procedure to combine bottom-up segmentation, in the form of SLIC superpixels, with sliding window detectors. To do this we build soft segmentation masks similar to those of our previous work, and use them to split block-wise HOG features into object-specific and background changes. We apply this to Deformable Part Models (DPM) for object detection. This work was published in (Trulls et al., 2014).

We prioritize solutions that are simple, general and fast. Our focal point is to build enhanced features, which can thereafter be used in place of the original ones. We apply this principle to multiple problems, such as stereo, motion estimation and object detection. We publish our code whenever possible[3].

1.3.1 Publications

The following is a list of the publications derived from this thesis:

Eduard Trulls, Alberto Sanfeliu, and Francesc Moreno-Noguer. *Spatiotemporal descriptor for wide-baseline stereo reconstruction of non-rigid and ambiguous scenes.* Proceedings of the European Conference on Computer Vision, 2012.

Eduard Trulls, Iasonas Kokkinos, Alberto Sanfeliu, and Francesc Moreno-Noguer. *Dense segmentation-aware descriptors.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.

Eduard Trulls, Stavros Tsogkas, Iasonas Kokkinos, Alberto Sanfeliu, and Francesc Moreno-Noguer. *Segmentation-aware deformable part models.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[3] https://github.com/etrulls

Eduard Trulls, Iasonas Kokkinos, Alberto Sanfeliu, and Francesc Moreno-Noguer. *Dense segmentation-aware descriptors.* In Ce Liu and Tal Hassner (Eds.), Dense correspondences in Computer Vision (under revision). Springer, 2014.

1.4 Thesis Overview

This thesis is structured in the following manner: three chapters relating to each of our three main publications, plus an introductory chapter to lay the groundwork and present material common to several or all of our papers. Here is a summary of the chapters:

Chapter 2: Overview. This introductory chapter presents some of the tools used throughout our work, from low-level features to optimization algorithms, and explains how they fall into the structure outlined in Sec. 1.2.

Chapter 3: A spatiotemporal approach to wide-baseline stereo. This chapter presents a spatiotemporal descriptor and a stereo algorithm. We show how to use it to reconstruct highly ambiguous, wide-baseline stereo sequences. This chapter is largely based on our 2012 ECCV paper.

Chapter 4: Segmentation-aware descriptors. This chapter introduces our technique to build segmentation-aware descriptors. We apply it to wide-baseline stereo reconstruction and to large displacement optical flow. This work is a collaboration with Iasonas Kokkinos of Ecole Centrale Paris, who authored the paper that inspired this work (Kokkinos and Yuille, 2008) and who served as advisor for this project. This chapter is largely based on our 2013 CVPR paper.

Chapter 5: Segmentation-aware Deformable Part Models. This chapter shows how to extend the approach introduced in the previous chapter to the more challenging problem of object detection. We apply it to a popular sliding window detector, Deformable Part Models (DPM), and demonstrate increased performance on the PASCAL VOC (Everingham et al., 2010). This work is a collaboration with Iasonas Kokkinos and Stavros Tsogkas of Ecole Centrale Paris. Prof. Kokkinos served as an advisor, while S. Tsogkas provided help on several extensions of this work which at the time of this writing remain unpublished, and are not part of this thesis. E. Trulls was the primary researcher and developer for this project. This chapter is largely based on our 2014 CVPR paper.

Chapter 6: Concluding remarks. The thesis concludes with a short summary of our efforts, an interpretation on where our work stands in a field currently undergoing significant developments, and considerations about open questions and directions for future research.

Chapter 2

Overview

In this thesis we develop approaches to build better low-level features for computer vision applications. This chapter explores elements common to the different techniques featured in this dissertation. We will first discuss the state of the art on local feature descriptors, and later introduce components which will be presented in further detail in the following chapters, as they become relevant. The goal of this chapter is to give the reader an global perspective of the different tools used in our work and how they interrelate.

We follow the division introduced in section 1.2: (a) local, low-level features; (b) global, mid-level cues; and (c) target applications. Sec. 2.1 lists the different components used in our work. Sec. 2.2 reviews local features in computer vision in detail. Sec. 2.3 introduces the mid-level cues we augment features with. Sec. 2.4 describes the computer vision problems we validate our contributions on, and the components used to tackle them, which consist of optimization techniques and machine learning algorithms. Some of these categories include canonical vision problems such as stereo or segmentation, for which we do not aim to provide a thorough overview; we focus instead on the techniques exploited in this work.

2.1 Components

Table 2.1 presents a comprehensive list of the building blocks used in this thesis, and indicates how each entry relates to the following, 'stand-alone' chapters:

Chapter 3: Spatiotemporal descriptors. We present an approach to build spatiotemporal descriptors based on Daisy descriptors (Tola et al., 2010) and optical flow priors. We apply them to dense, wide-baseline stereo, using SIFT, Daisy, and the Spatiotemporal Quadric Element (Stequel) of Sizintsev and Wildes (2009) as a baseline.

Chapter 4: Segmentation-aware descriptors. We show how to exploit segmentation cues to build descriptors that can deal with occlusions and background changes. We apply our technique to SIFT and to the Scale- and Invariant-Descriptor (SID) of Kokkinos and Yuille (2008). We use three different seg-

Type	C3	C4	C5	Work & citation
Features	✓	✓	✓	SIFT (Lowe, 2004)
('low level')	✓	✓		Daisy (Tola et al., 2010)
	✓			Stequel (Sizintsev and Wildes, 2009)
		✓		SLS (Hassner et al., 2012)
		✓		SID (Kokkinos and Yuille, 2008)
			✓	HOG (Dalal and Triggs, 2005)
Global cues	✓			Optical flow
('mid level')		✓		Normalized cut eigenvectors (Maire et al., 2008)
		✓		Soft segmentations (Leordeanu et al., 2012, 2014)
		✓		Structured forests detector (Dollár and Zitnick, 2013)
			✓	SLIC superpixels (Achanta et al., 2012)
Applications	✓	✓		Wide-baseline stereo
(target)		✓	✓	Large-displacement motion w/ SIFT-flow (Liu et al., 2011)
			✓	Object detection w/ DPMs (Felzenszwalb et al., 2010b)

Table 2.1: Index of the technologies used in this thesis, and how they relate to the work presented in the following chapters.

mention cues: (a) soft segmentations derived from the eigenvectors of the normalized cut relaxation (Maire et al., 2008); (b) the soft segmentations provided by Leordeanu et al. (2012); and (c) boundary estimates, obtained with the state-of-the-art detector of Dollár and Zitnick (2013). We demonstrate improvements in large-displacement optical flow within the registration framework of Liu et al. (2011), who extend optical flow from raw pixel values to dense descriptors; and also in wide-baseline stereo. We use SIFT, Daisy and SID as a baseline, as well as the Scale-less SIFT (SLS), a SIFT-based descriptor invariant to scale changes (Hassner et al., 2012).

Chapter 5: Segmentation-aware Deformable Part Models. We show how to extend our segmentation-based approach to construct background-invariant features for the more challenging problem of object detection. We use SLIC superpixels to extract soft segmentation masks on a scale-, position-, and object-dependent manner, and use them to split low-level features based on Histograms of Oriented Gradients (HOG) into object and background channels. We use these background-invariant features to build sliding window detectors following the Deformable Part Models (DPM) paradigm (Felzenszwalb et al., 2010b).

2.2 Low-level features

Matching points across images is the backbone of a great number of computer vision applications. Local feature descriptors have proved very successful at this task, and have been applied to problems as diverse as wide baseline matching (Schaffalitzky and Zisserman, 2002; Tuytelaars and Van Gool, 2004; Tola et al., 2010); 3D pose estimation (Moreno-Noguer et al., 2007; Lepetit and Fua, 2006); recognition of objects (Lowe,

2004; Ferrari et al., 2004; Dorko and Schmid, 2003; Fergus et al., 2003; Opelt et al., 2004), scenes (Fei-Fei and Perona, 2005), and texture (Lazebnik et al., 2003); robot navigation (Se et al., 2002; Maimone et al., 2007); image (Mikolajczyk and Schmid, 2001) and video (Sivic and Zisserman, 2003) retrieval; image stitching (Szeliski, 2004; Brown and Lowe, 2007); and structure from motion (Pollefeys et al., 2004; Snavely et al., 2006).

Features are often used *sparsely*, i.e. only for salient points. Extracting a local descriptor in this manner involves two steps:

- *Feature detection*, i.e. finding a salient point, which is distinctive and where scale and rotation can be reliably estimated.
- *Feature representation*, i.e. computing the representation itself from the image patch, at a given scale and orientation.

Good *feature detectors* should be accurate and repeatable, and provide distinctive features. Some examples include the Harris corner detector (Harris and Stephens, 1988), SUSAN (Smith and Brady, 1997), and FAST (Rosten and Drummond, 2006), a modern corner detector designed for real-time applications based on machine learning techniques. The SIFT detector convolves the image with a difference of gaussians kernel at multiple scales (Lowe, 2004). The detector used with SURF (Bay et al., 2008), a SIFT-like descriptor designed towards efficiency, uses integral images to approximate the determinant of hessian (Lindeberg, 1998). Detection algorithms can be tailored to specific applications; e.g. Matas et al. (2004) introduce maximally stable extremal regions (MSER) for wide-baseline stereo. Mikolajczyk et al. (2005) presents a thorough comparison of affine covariant region detectors. Our work is concerned exclusively with dense features, where feature detection is unnecessary, and we will thus pay no further attention to the subject.

Going back to *feature representation*, the standard way to solve the correspondence problem is by building representations from salient image patches, in a manner that is invariant to common transformations such as illumination or viewpoint changes. We refer to these local representations as feature descriptors. We can then measure the similarity between points in this transformed space. These representations usually take the form of a multi-dimensional array of floating point values, and the affinity between a pair of descriptors can be computed e.g. with their euclidean distance.

Feature descriptors have proven very adept at describing in a succinct and informative manner the neighborhood around an image point. Optimal descriptors should be (a) *invariant*, to common transformations such as scaling, rotation, illumination or viewpoint changes, and possibly to more advanced problems such as non-rigid deformations (depending of course on the application); and (b) *discriminative*, i.e. their representation of the patch should be distinctive. Additional considerations include their size and computational efficiency (which are interrelated). Efficiency concerns include feature detection, if relevant, and computing both the features and the actual correspondences. A recent trend is to abandon real-valued descriptors in favor of binary representations, which are cheaper to store and faster to operate with; we will introduce some of them in Sec. 2.2.6, but in our work we rely on traditional real-valued features.

The simplest way to describe a pixel is with its grayscale value or a small patch around it. Metrics to compare two such features would therefore be, respectively, pixel differencing or the sum of square differences (SSD). We can achieve some invariance to

illumination changes using the normalized cross-correlation (NCC), which is the dot product between two normalized vectors. A comparison of correlation-based techniques can be found in (Martin and Crowley, 1995). These techniques are still in use for applications such as narrow-baseline stereo, where occlusions and perspective changes are not significant and the similarity can be computed by cross-correlating square windows.

More elaborate techniques to represent appearance or shape rely on histograms, which can be of simple data such as pixel intensities, or take advantage of more expressive representations. For instance, Johnson and Hebert (1997) introduced histograms of relative positions in range data for 3D object recognition, an approach that was later extended to images by Lazebnik et al. (2003), histogramming intensity and distance to the center. Another early descriptor based on histograms is that of Zabih and Woodfill (1994), which relies on histograms of the ordering of relative pixel intensities. Related work exists on the discipline of texture classification, including cooccurrence matrices (Davis et al., 1979), polarograms (Davis, 1981), texture anisotropy (Chetverikov, 1982), filters such as Gabor or wavelet transforms (Randen and Husoy, 1999).

Another histogram-based descriptor is Shape Context (Belongie and Malik, 2002), intended to describe shapes for object recognition, which is conceptually similar to SIFT but is based on edges instead of gradients; edges are extracted with the Canny detector (Canny, 1986). Geometric Blur (Berg and Malik, 2001) extended Shape Context to gradients instead of contours.

Many of these works rely on image gradients, which have proved to be very rich features for recognition, and whose use in computer vision reached maturity with SIFT. SIFT combines a scale-invariant region detector with a carefully engineered descriptor based on 3D histograms of oriented gradients, with local spatial histogramming over 4×4 blocks around the point, and normalization. This approach is related to biological models of the primary visual cortex: Edelman et al. (1997) posit that neurons in the visual cortex respond to gradients at particular orientations—but the location of the gradient response is allowed to shift, and does not have to be precisely localized. SIFT represented a great leap forward on the state of the art of feature descriptors, and remains the most popular among its brethren. The SIFT descriptor will be described in the following section.

Despite its popularity, SIFT was not designed with computational efficiency in mind, and a host of recent works have been developed to facilitate the use of descriptors in real-time applications.

A PCA (Principal Components Analysis)-based extension to SIFT was proposed by Ke and Sukthankar (2004), applying PCA to the normalized gradient patch, in place of SIFT's smoothed weighting scheme. Another derivative work is GLOH (Mikolajczyk and Schmid, 2005), which computes a SIFT analog on a log-polar location grid with 17 location bins and 16 orientation bins, resulting in a 272 bin histogram; the size of the descriptor is reduced to 128 with PCA, estimated from patches collected from a large set of images. The SURF descriptor (Bay et al., 2008) is based on Haar wavelet responses and employs integral images to compute the histogram bins, reducing the computational load—at the cost of having all gradients contribute equally to their respective bins.

A complementary line of research, also concerned with efficiency, is that of *dense descriptors*, typically computed for every pixel in the image. Recent works such as Dense SIFT (DSIFT) (Vedaldi and Fulkerson, 2008), Daisy (Tola et al., 2010) or the

dense Scale-invariant Descriptors of Kokkinos et al. (2012b) (SID) have demonstrated that it is possible to compute dense descriptors efficiently and use them as a generic low-level image representation on a par with filter banks. This negates the need for feature detection, which is not feasible on arbitrary points.

Other researchers have tackled problems, at the descriptor level, which are often addressed at higher abstraction levels, such as non-rigid deformations, scale, and occlusions. These are closer in spirit to the works presented in this thesis. Current techniques accommodate scale changes by limiting their application to singular points, where scale can be reliably estimated, or alternately sampling densely at arbitrary scales. In contrast, Kokkinos and Yuille (2008) rely on a log-polar transformation and Fourier Transform properties to achieve scale invariance by design, while Hassner et al. (2012) introduce a scale-invariant version of SIFT (SLS) based on SIFT descriptors computed at multiple scales. Non-rigid deformations have been seldom addressed with region descriptors, but recent advances show that kernels based on heat diffusion geometry can effectively describe local features of deforming surfaces, which has been applied to images with the DaLI descriptor (Moreno-Noguer, 2011), as well as 3D models (Bronstein and Kokkinos, 2010). Regarding occlusions, Tola et al. (2010) demonstrated performance improvements in multi-view stereo from the treatment of occlusion as a latent variable and enforcing spatial consistency with graph cuts.

2.2.1 SIFT

The Scale Invariant Feature Transform, commonly known as SIFT, has become the *de facto* standard in many computer vision applications which require correspondence matching. SIFT *descriptors* are built aggregating oriented gradients at a selected scale and orientation around a point, over three dimensions: the spatial coordinates and the gradient orientation.

The standard SIFT descriptor is defined by the 4×4 grid pictured in Fig. 2.1, with 8 orientation bins. The scale and orientation obtained on the detection step determine the size and orientation of the grid. As such, scale- and rotation-invariance is contingent on the *detector*. Gradient responses are weighted with a gaussian kernel centered on the pixel location, to give stronger weights to gradient orientations near the interest point, and spread over multiple bins to avoid boundary effects. The final descriptor is the concatenation of histograms and is size $4 \times 4 \times 8 = 128$.

This array is normalized to increase the robustness of the descriptor against affine illumination changes. Lowe (2004) proposed a two-step approach: normalization to unit length, clipping the values to a threshold (0.2), and renormalization if necessary.

SIFT was introduced as a descriptor for sparse interest points, along with its own feature detector, but it has proven very successful when sampled densely—particularly for image classification (Nowak et al., 2006; Bosch et al., 2006), where the descriptors are often clustered into a vocabulary of visual words and exploited with a bag-of-words model (Csurka et al., 2004) or spatial pyramids (Lazebnik et al., 2006). Dense SIFT has also been applied to image alignment with SIFT-flow (Liu et al., 2011), which will be described in Sec. 2.4.2. Given that SIFT was originally formulated for sparse matching it is not designed with computational efficiency in mind, but it can be computed densely for constant scales and orientations. The VLFEAT library (Vedaldi and Fulkerson, 2008) offers a fast implementation of dense SIFT, which will be referred throughout this document as DSIFT.

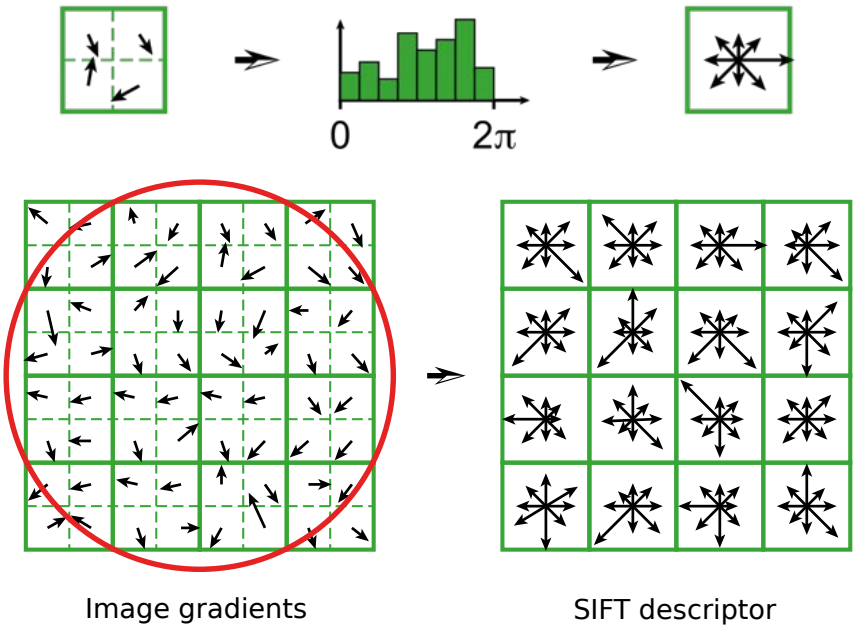


Figure 2.1: SIFT descriptors are 3-D histograms of image gradients over the spatial coordinates and gradient orientations. The image gradients are weighted by a gaussian window, indicated by the red circle on the bottom left figure. The length of the arrows corresponds to the sum of gradient magnitudes on a given direction. Released by Indif under CC-BY-SA-3.0, edited by the author.

2.2.2 Daisy

Another important work towards dense features is Daisy (Tola et al., 2010), a descriptor specifically designed for dense computation. Unlike SIFT, Daisy's histograms are aggregations of image gradients obtained from convolutions over gradient images, which are faster to compute, and allow the algorithm to use larger, more informative patches. Daisy is used in chapters 3 and 4, and we will describe it in detail in this section.

Analogous to SIFT's 4×4 cell structure, Daisy employs a polar grid centered on the pixel for which we want to compute a descriptor, $\mathbf{x}$. The grid (Fig. 2.2, left) $g[k, n]$ is defined by N equally spaced points over K equally spaced rays leaving $\mathbf{x}$. The number of grid points is thus $P = 1 + K \cdot N$; the center of the grid $\mathbf{x}$ is also sampled. For each point we compute a histogram of gradients with H orientation bins, so that the size of the descriptor is $S = P \cdot H$.

To extract Daisy descriptors efficiently, we first compute H gradient maps from the input image. These are defined as the gradient norms at each location, if they are positive, and are set to zero otherwise; this preserves the polarity of intensity changes. Each orientation map is then convolved with multiple gaussian kernels of increasing size to obtain the convolved orientation maps, so that the size of the kernel determines the size of the region. Gaussian filters are separable, and the convolution for larger kernels can be obtained from consecutive convolutions with smaller kernels, which makes for a significant improvement in computational efficiency.

The number of rings N also determines the number of kernels to convolve the gradient images with: outer rings use convolution maps computed over larger regions,

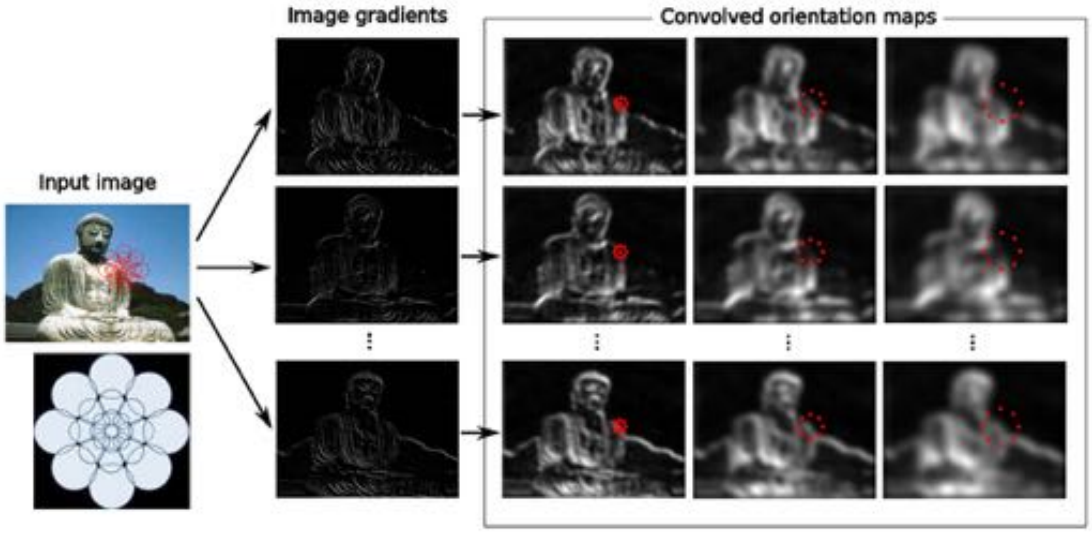


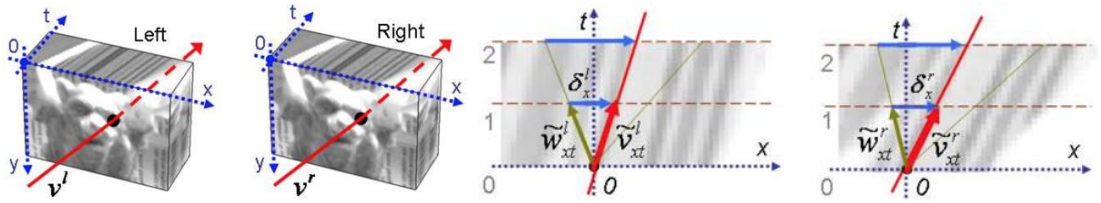
Figure 2.2: To compute dense Daisy descriptors, we first obtain image gradients in $H = 8$ directions (3 are pictured). The gradient images are then convolved with $N = 3$ gaussian kernels of increasing size. The grid $g[k, n]$, centered on feature point $\mathbf{x}$, and defined by $N = 3$ equally-spaced concentric rings and $K = 8$ equally spaced rays, is pictured over the original image (and below it, magnified). The circles represent the size of the regions considered at each grid point, which are larger for the outer rings. Building a Daisy descriptor for $\mathbf{x}$ is then a matter of sampling the convolved orientation maps at grid points $g[k, n]$, centered on $\mathbf{x}$, where n determines the spatial coordinates and also the magnitude of the smoothing, as illustrated in the right-hand side of the figure. These sampled values are concatenated into a $S = 200$-dimensional Daisy descriptor.

to gain a degree invariance against small rotations. In Fig. 2.2, this is represented with circles of increasing size centered on each grid coordinate. We thus compute $N \times H$ convolved orientation maps, which contain a weighted sum of gradient norms around each pixel at H orientations, smoothed N times. A Daisy descriptor for point $\mathbf{x}$ is obtained sampling the convolved orientation maps at the coordinates of grid $g[k, n]$, centered on $\mathbf{x}$. This process is illustrated in Fig. 2.2. In this manner, the convolved orientation maps are repeatedly sampled as we extract descriptors densely, without recomputing them. Tola et al. (2010) use $H = 8$ orientation bins, $N = 3$ rings and $K = 8$ points per ring, so that a Daisy descriptor is size $S = 200$.

The orientation histograms are normalized to unit length separately for each of the P grid points, rather than normalizing the descriptor as a whole. This is part of a technique introduced in (Tola et al., 2010) to deal with partial occlusions by disabling certain grid coordinates, which will be presented in Sec. 3.5. Following their formulation, the score for the match between a pair of Daisy descriptors $\mathbf{D}_i$ and $\mathbf{D}_j$ is defined as:

$$d\left(\mathbf{D}_i, \mathbf{D}_j\right) = \frac{1}{P} \sum_{p=1}^{P} \left\| \mathbf{D}_i^{[p]} - \mathbf{D}_j^{[p]} \right\|_2, \tag{2.1}$$

where $\mathbf{D}^{[p]}$ denotes the histogram for grid point p.



(a) **(b)**

Figure 2.3: The Spatiotemporal Quadric Element (Stequel). (a) The spacetime volume, for a binocular system. (b) A $x - t$ slice of the spatiotemporal volume, for each camera, with a pair of vectors which can be put in correspondence (for binocular stereo). Figure reproduced from (Sizintsev and Wildes, 2009).

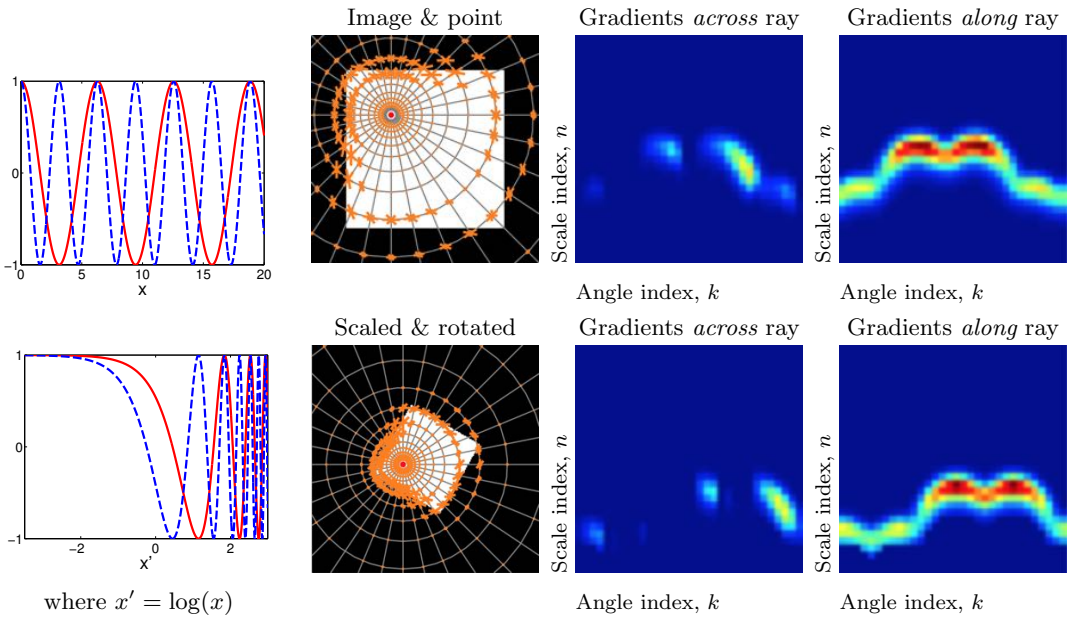
Notice how SIFT, Daisy and several other descriptors introduced in the previous section all follow similar principles, differing in the strategies used for feature extraction and pooling. This was explored by Winder and Brown (2007) in a paper which studied a large number of gradient- and steerable filter-based configurations, with the goal of determining the combination of techniques that yields the best compromise between discriminative power and the footprint of the descriptor. The configurations explored included generalizations of SIFT, GLOH, Daisy and the spin images of Lazebnik et al. (2003). Their proposal, which was shown to outperform state-of-the-art descriptors on sparse point matching, has a shape remarkably similar to that of Daisy (without its concerns towards efficiency for dense computation).

2.2.3 Stequel

Sizintsev and Wildes (2009) present an approach to spatiotemporal stereo based on primitives that encapsulate spatial and temporal information, and the match constraints required to bring them into correspondence across narrow-baseline binocular views. Their approach is based on the spatiotemporal volume: orientations on the image plane capture the spatial distribution of objects and scene, whereas orientations in the temporal dimension encapsulate dynamic content. The combination of appearance and motion can be used to resolve ambiguities better than with either type of data alone.

Their stereo primitives are called Spatiotemporal Quadric Elements, or Stequel. To build them they first apply 3D steerable filters, which are convolved with the data across a set of 3D orientations, obtaining a measure of the energy along each of these directions, for every point in spacetime. The steerable filter responses are used to construct the spatiotemporal quadric, a feature representation designed to capture local orientation information as well as the variance of spacetime about that orientation.

The left and right channels of the stereo system are filtered separately. The spatiotemporal orientation will generally change across different viewpoints, and we cannot match stequels directly, but we can exploit the epipolar constraints given by a calibrated, binocular stereo camera to obtain comparable features. This procedure is outlined in Fig. 2.3. Stequels are used as match primitives for both local and global matchers. They are embedded in a coarse-to-fine, local, block-matching algorithm with shiftable windows, and also in a global graph-cuts algorithm with occlusion reasoning, operating at the finest scale. The disparity estimates are brought to sub-pixel precision



(a) Scale change (1D) **(b)** Scaling & rotation (2D)

Figure 2.4: Building invariant representations for 1D and 2D signals with the Fourier Modulus Transform technique. **(a)** The left column demonstrates, for a 1D signal, how the logarithmic transformation $x' = \log(x)$ turns scaling into translation: the red-solid $f(x) = \cos(x)$ and blue-dashed $g(x) = \cos(2x)$ functions differ by a scale factor of two. The transformation $x' = \log(x)$ delivers $f'(x') = \cos(\log(x)) = \cos(x')$ and $g'(x') = \cos(\log(x) - \log(2)) = \cos(x' - \log(2))$. Note that now $f'(x')$ and $g'(x')$ only differ by a translation. **(b)** We show an image, scaled and rotated (left), and image measurements $g[k, n]$ for two corresponding points. For clarity, we show two (out of H) gradient components: *across* rays (center column), and *along* rays (right column). The needle length over the grid points indicates the magnitude of the directional derivative. Sampling densely enough, image rotation and scaling turn into translations (shifts) of this matrix, except for the introduction and removal of some entries at the finer/larger scales. Notice that this point is arbitrary, where estimating scale and orientation would be hard, or unfeasible.

with a Lucas-Kanade type refinement. The stequels are built from $5 \times 5 \times 5$ windows across the spatiotemporal volume. This procedure does not require explicit motion recovery—either optical flow or scene flow—but it can be used to estimate 3D scene flow. We use stequels to benchmark our work on spatiotemporal stereo on chapter 3.

2.2.4 SID

Feature detectors are able to find stable scales and orientations only for a small subset of the pixels. Feature matching strategies can thus be *sparse*, and computed only for reliable interest points, or *dense*, using arbitrary scales and orientations. In several applications it can be desirable to construct a scale-invariant descriptor densely, for instance when establishing dense image correspondence in the presence of scale changes. In such cases scale selection is not appropriate, as it is only reliably applicable around

a few singular points (e.g. blobs or corners). A work towards this goal is the Scale- and rotation-Invariant descriptor (SID) by Kokkinos and Yuille (2008). We will describe this descriptor in detail as it provides the foundation for our dense, segmentation-aware descriptors (chapter 4).

SID relies on a combination of log-polar sampling and spatially-varying smoothing, converting image scalings and rotations into translations. Invariance to rotation and scaling can then be guaranteed adapting the Fourier Transform Modulus/Fourier-Mellin technique, which is translation-invariant, to the construction of descriptors, bypassing the need for scale detection. In Fig. 2.4-(a) we show an illustration of the technique for a one-dimensional signal, and in Fig. 2.4-(b) we show how to adapt it to image descriptors.

We first consider describing a one-dimensional signal $f(x),\ x > 0$ in a manner that will not change when the signal is scaled as $f(x/a),\ a > 0$. Using the domain transformation $x' = \log(x)$ we can define a new function f' such that

$$f'(x') \doteq f(x), \quad \text{where } x' = \log x, \tag{2.2}$$

which is what we will be referring to as the 'logarithmically transformed' version of f. This is illustrated in Fig. 2.4-(a). For this particular transformation, dilating f by a will amount to translating f' by a constant, $\log(a)$:

$$f'(x' - \log(a)) \quad = \quad f(x/a), \tag{2.3}$$

meaning that we turn dilations of f to translations of f'.

We can thus extract a scale-invariant quantity based on the shifting-in-time property. Defining $f_a(x') = f'(x' - \log(a))$, and denoting by $F_a(\omega)$ the Fourier Transform of $f_a(x)$ we then have:

$$F_a(\omega) \quad = \quad F_1(\omega)e^{-j\log(a)\omega}, \quad \text{and,} \tag{2.4}$$

$$|F_a(\omega)| \quad = \quad |F_1(\omega)|. \tag{2.5}$$

From Eq. 2.5 we conclude that changing a will not affect $|F_a(\omega)|$, the Fourier Transform Modulus (FTM) of f_a, which can thus be used as a scale-invariant descriptor of f.

In Fig. 2.4-(b) we show that 2D scalings and rotations can similarly be converted into a translation with a log-polar transformation of the signal—and then eliminated with the FTM technique. The principle behind this approach is commonly used in tasks involving global transformations such as image registration (Casasent and Psaltis, 1976; Wolberg and Zokai, 2000) and texture classification (Stein and Hebert, 2005). The original paper (Kokkinos and Yuille, 2008) used features based on the monogenic signal (Felsberg and Sommer, 2001) to build sparse descriptors, while a later version (Kokkinos et al., 2012b) enabled its dense application by exploiting the Daisy pipeline to compile features efficiently. We now we describe the latter.

Adapting the FTM technique to the construction of local descriptors requires firstly a discrete formulation. We construct a descriptor around a point $\mathbf{x} = (x_1, x_2)$ by sampling its neighborhood along K rays leaving $\mathbf{x}$ at equal angle increments $\theta_k = 2\pi k/K,\ k = 0, \ldots, K-1$. Along each ray we use N points whose distances from $\mathbf{x}$ form a geometric progression $r_n = c_0 a^n$. The signal measurements on those points provide us with a matrix of size $K \times N$:

$$g[k, n] = f\left[x_1 + r_n \cos(\theta_k), x_2 + r_n \sin(\theta_k)\right], \tag{2.6}$$

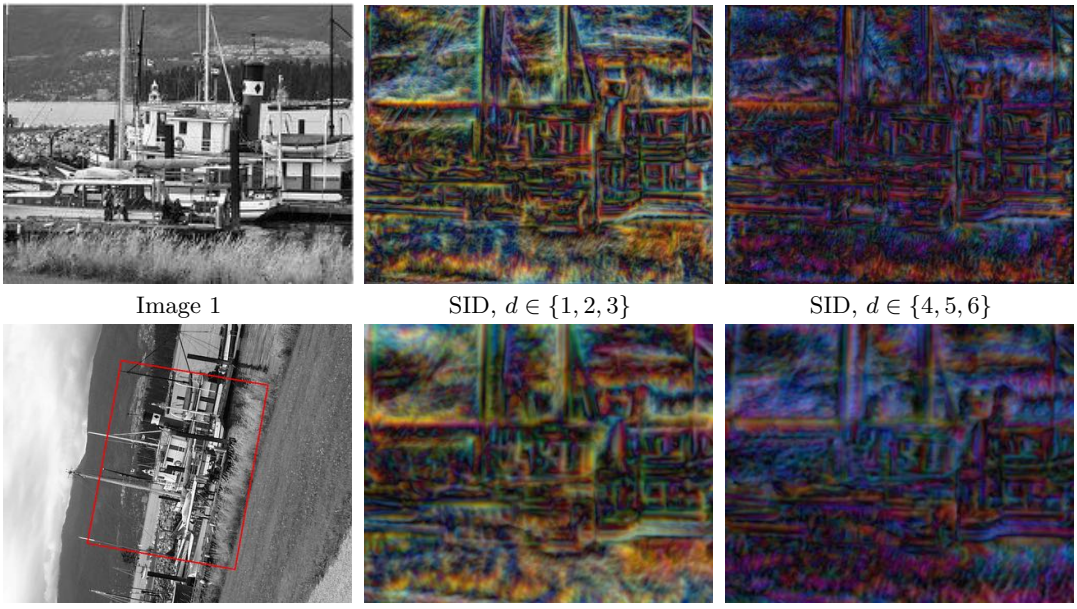


Figure 2.5: Visualization of dense SID: the location of Image 1 within Image 2 is indicated by the red box. The scaling transformation amounts to an area change in the order of four. After computing dense SID descriptors for every image, we align the descriptors for the bottom row, within the red box, with those for the top row, for easy inspection, and visualize their lower-frequency components in RGB space: [1-3] in the middle column and [4-6] in the right column (low-frequency components typically contain the most energy). We demonstrate that the descriptors are effectively invariant to scaling and rotation.

By design, isotropic image scalings, and image rotations, amount to translations over the radial and angular dimensions of this representation, respectively.

From the time-shifting property of the Discrete-Time Fourier Transform (DTFT), we know that if $g[k, n] \overset{\mathcal{F}}{\leftrightarrow} G(j\omega_k, j\omega_n)$ are a DTFT pair, then:

$$g[k - c, n - d] \overset{\mathcal{F}}{\leftrightarrow} G(j\omega_k, j\omega_n)e^{-j(\omega_k c + \omega_n d)}, \tag{2.7}$$

This means that the magnitude of the DTFT $|G(j\omega_k, j\omega_n)|$ is unaffected by signal translations. Applying this observation to the descriptor we realize that this provides a scale- and rotation-invariant quantity. Alternatively, we can apply the Fourier Transform only over scales, to obtain a scale-invariant but rotation-dependent quantity (and vice-versa, although a rotation-invariant and scale-sensitive variant is arguably less appealing). This can be useful in scenes with scaling changes but no rotations, where we would be discarding useful information. We will refer to the scale- and rotation-invariant descriptor as **SID** and to the scale-invariant but rotation-sensitive descriptor as **SID-Rot**.

For memory- and time-efficient dense computation, Kokkinos et al. (2012b) combine Daisy with steerable filtering (Freeman and Adelson, 1991) and recursive Gaussian convolutions (Deriche, 1987). The image measurements on $g[k, n]$ are obtained filtering the image with Gaussian filters (Geusebroek et al., 2003), following a 'foveal' smoothing

pattern, with a smoothing scale that is linear in the distance from the descriptor's center; as with Daisy, grid points further away from the center account for larger image regions (see Fig. 2.2). We extract directional derivatives at H' orientations, offset by the current ray's orientation (see e.g. 2.4-(b) for the components along, and perpendicular to the ray). We preserve the polarity as in (Tola et al., 2010), so that the effective number of bins of the orientation histograms is $H = 2 \cdot H'$. In Fig. 2.5 we show the values of the lowest-frequency coefficients of densely computed descriptors on two images related by scaling and rotation. We see that the descriptor values are effectively invariant, despite a scaling factor in the order of 2.

For the invariance assumption to hold, we must sample the neighborhood of the image point very densely: typical values are $K = 32$ rays and $N = 28$ scales. This creates two problems. Firstly: it produces a very large descriptor (size $S = K \times N \times H$). There are strategies to deal with this: in addition to compression or binarization techniques, which will be introduced in Sec. 2.2.6, we can drop low-energy, high-frequency components after computing the Fourier Transform. Secondly, and most importantly: scale invariance requires large image patches, so that in practical situations the descriptor will suffer from background interference and occlusions. In chapter 4 we devise a method to overcome this limitation, exploiting global segmentation cues to filter out features likely to belong to different regions. We apply this procedure over the spatial domain, before the Fourier Transform.

2.2.5 Scale-less SIFT (SLS)

As we argued in the previous section, dense descriptors should not rely on scale detection, which is not feasible on most image pixels, and neither should they fall back to arbitrary scales instead. SID introduced a technique to build inherently scale-invariant descriptors, based on properties of the Fourier Transform. An alternative strategy is presented by the Scale-less SIFT, or SLS (Hassner et al., 2012), a scale-invariant descriptor designed as an efficient representation of multi-scale SIFT descriptors.

This work builds on two key observations. The first observation is that feature descriptors can change their values significantly across scales, even in low-contrast areas—where we would expect SIFT values to be approximately uniform. This implies that selecting a single scale, arbitrary or not, may not be a reliable strategy when the images exhibit scale changes. The second observation is that rather than use a single scale, we can represent single pixels with descriptors computed at multiple scales and match them *as a set*. This carries a considerable computational cost.

A byproduct of the latter is the realization that SIFT descriptors do not change drastically across different scales. This indicates they are embedded in a low-dimensional sub-space. Rather than devise strategies to match sets of multi-scale SIFT, Hassner et al. (2012) propose to find a compact representation of the these sub-spaces. This is a well-studied problem, and they propose to employ the substance-to-point mapping of Basri et al. (2010) to produce the Scale-less SIFT.

The aforementioned observations of Hassner et al. (2012) are summarized in Fig. 2.6. Notice how:

- Low-contrast points (particularly true for the smaller scales) do not result in uniform SIFT histograms.
- The set of SIFT descriptors for the image on the left matches the set of SIFT descriptors for the image on the right—at higher scales. Matching sets of SIFT

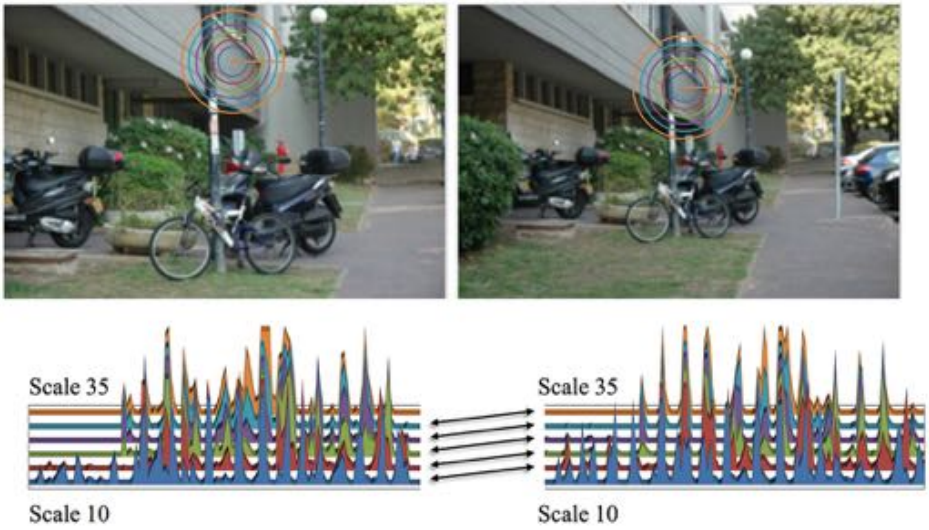


Figure 2.6: Top: two images of the same scene at different scales. Bottom: histograms for SIFT descriptors computed at a set of 6 scales, for the same point, with the same orientation. The descriptors for the image on the left match those for the image on the right, at higher scales. Figure from (Hassner et al., 2012).

descriptors seems a viable, if costly, strategy.

- The SIFT histograms change gradually across scales. This suggests that sets of multi-scale SIFT descriptors are redundant, and more compact representations can be obtained; Hassner et al. (2012) present one way to achieve this.

SLS gives clearly better results than SIFT in the presence of scaling transformations. The dimensionality of the final descriptor is very large (8256), but the authors make available a much smaller PCA-based version (528), developed after the paper was published. This however still requires computing the full descriptor, which comes at a significant computational price; see chapter 4 for a comparison with other descriptors. We use SLS in chapter 4 to benchmark our segmentation-aware descriptors. Notice that unlike SID, it is not rotation-invariant.

2.2.6 Metric learning and binary descriptors

Feature descriptors are usually high-dimensional, e.g. SIFT is size 128, and Daisy 200. This can pose problems in applications such as large-scale retrieval, due to increased storage. Likewise, matching becomes computationally expensive, as it involves Euclidean distances between long feature vectors. This affects, in particular, any application that relies on dense descriptors. There have been attempts to address these problems, including vector quantization (Tuytelaars and Schmid, 2007; Winder et al., 2009), where the vector features are quantized into a small number of bits; and dimensionality reduction (Hua et al., 2007; Mikolajczyk and Schmid, 2005; Winder et al., 2009), particularly with PCA.

An alternative approach is supervised metric learning, where labeled training data is used to learn transformations that render short binary codes whose distances are small for positive training pairs and large otherwise. Binarization (a) greatly reduces the size of the descriptors and makes storage cheaper, and (b) the similarity between

two descriptors can be computed with the Hamming distance, i.e. bit-wise operations, which are very fast.

Binarization from real-valued descriptors is usually performed by multiplying the descriptors by a projection matrix $\mathbf{P}$, subtracting a thresholding vector $\mathbf{t}$ and computing the sign of the result, i.e. $\mathbf{y} = \mathtt{sign}(\mathbf{P}\mathbf{x} + \mathbf{t})$. The binarized descriptor is a string of bits of a length given by the projection matrix/thresholding vector. In supervised binarization techniques based on a linear projection, the matrix and thresholds are selected so as to preserve similarity relationships. This implies a difficult non-linear optimization problem for which a global optimum is not guaranteed (Strecha et al., 2012). An alternative is spectral hashing (Weiss et al., 2008), which does not suffer from this problem.

A recent metric learning approach is that by Strecha et al. (2012), the basic idea of which is to find a mapping from descriptor space to Hamming space (binary strings) with a transformation of the aforementioned type. Their method involves two steps, to determine $\mathbf{P}$ and $\mathbf{t}$. They propose a technique to compute the projection matrix $\mathbf{P}$ in closed form; the thresholding vector $\mathbf{t}$ is then adjusted to maximize recognition rates. Their procedure is equivalent to classical Linear Discriminant Analysis; hence the name of the algorithm: LDA-hash.

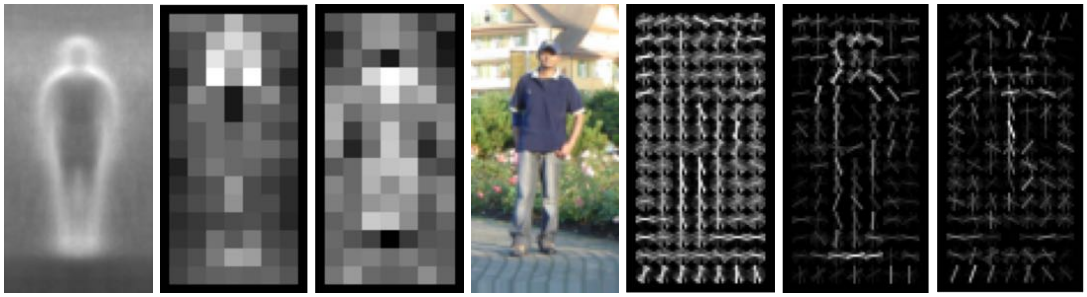
Binary descriptors are cheap to store, fast to match, and perfect candidates for real-time applications. Metric learning approaches, however, still rely on sophisticated, high-dimensional representations, such as SIFT, which are then transformed into a binary representation. An alternative way to attack this problem is to compute binary descriptors directly, eliminating the computational overhead. Modern binary descriptors are built applying simple operations, such as the difference over intensities, directly on raw image patches.

The current interest in binary descriptors was sparked by BRIEF (Calonder et al., 2010), which is built by concatenating the results of 256 intensity difference tests on pre-defined locations, drawn randomly. These intensity comparisons can be understood as approximations of gradients along random orientations. BRISK (Leutenegger et al., 2011) substitutes the random sampling of BRIEF with a uniform sampling scheme on a circular pattern, and the descriptor is a concatenation of short-distance intensity comparisons. The rotation is estimated at the feature detection level, and the pattern is shifted according to the dominant orientation, which makes it rotation-invariant. Two other BRIEF variants are ORB (Rublee et al., 2011), which uses a learning method to decorrelate the difference tests, and FREAK (Alahi et al., 2012), which uses a sampling scheme inspired by the human visual system. Recent works have also studied the advantages in learning the entire descriptor from raw image patches (Brown et al., 2011; Simonyan et al., 2014; Trzcinski et al., 2013).

While we do not employ binarization in this thesis, the descriptors introduced in chapters 3 and 4 suffer from high dimensionality, and we are currently investigating the application of metric learning techniques to our work.

2.2.7 Histograms of Oriented Gradients

Histograms of Oriented Gradients (HOG) are features designed for object detection. They were first introduced by Dalal and Triggs (2005) for the purpose of pedestrian detection. HOG consist in histograms of gradient orientations in localized image regions, and as such are related to SIFT and shape contexts. HOG differ from them in



(a) (b) (c) (d) (e) (f) (g)

Figure 2.7: HOG features and trained detectors. **(a)** Average gradient over the training samples. **(b)** shows the maximum positive weight of the learned SVM on a cell, and **(c)** the same for negative SVM weights. For a test image **(d)**, we show the HOG features in **(e)**, and the features weighted by the positive **(f)** and negative **(g)** SVM weights; (f) and (g) show only the dominant orientation for each cell. This figure was reproduced from (Dalal and Triggs, 2005).

that they are computed on a dense grid of uniformly spaced cells, without orientation alignment, at a single scale, and in that they use contrast normalization over larger regions of the image for better invariance against illumination changes. As Lowe did for SIFT, Dalal and Triggs experimented with multiple settings and combinations and determined that the best results were obtained for fine-scale gradients, fine orientation binning and coarse spatial binning.

Their paper proposes 9-bin orientation histograms computed over cells of 8×8 pixels, which are L2-normalized separately four times over blocks of 2×2 cells, so that every cell contributes to the descriptor more than once. This results in a $4 \times 9 = 36$-dimensional feature vector per cell. HOG features are then concatenated for all adjacent cells with a predetermined aspect ratio, so that e.g. 64×128 pixels, or 7×15 blocks (after discarding boundary blocks), produce HOG features size 3780. These feature vectors are subsequently used to train linear SVM classifiers. This procedure is illustrated in figure 2.7. HOG detectors are typically used in a sliding window fashion, where a filter is applied at every position and scale of an image. A detection hypothesis, as well as the training examples, are thus determined by a bounding box. HOG features are not robust to object rotation, which is one of the reasons the original algorithm targeted upright pedestrians.

This work was extended by Zhu et al. (2006), using regions of a variable size which are learned with AdaBoost (Freund and Schapire, 1995). The classifiers are organized in a cascade, which helps reject many negatives using a small subset of detectors and speeds up the detection process, following the ideas of Viola and Jones (2001). HOG are also the features used in the Deformable Part Models framework (Felzenszwalb et al., 2010b), which is a direct extension of the original HOG paper and arguably the dominant paradigm in object detection, at least until 2013. We will introduce DPMs in Sec. 2.4.3, and use them in chapter 5.

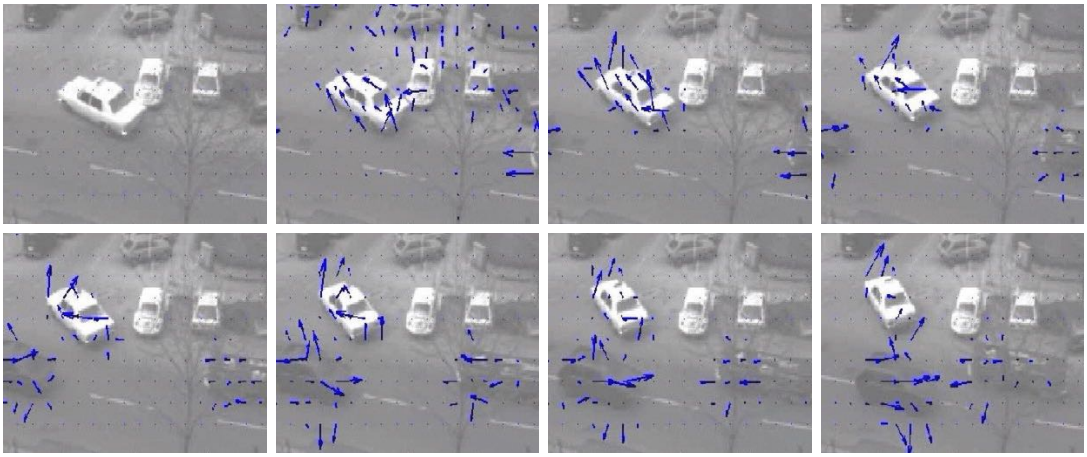


Figure 2.8: Optical flow in computer vision. The figure shows eight images from the taxi sequence in the Karlsruhe dataset. The direction and length of the arrows represent the orientation and magnitude of optical flow on each pixel. Reproduced from (Browning et al., 2007).

2.3 Mid-level cues

The algorithms introduced in this thesis use two types of *mid-level cues*: optical flow and segmentation. Although they are unrelated, we group them under the same category because, as we argued:

- Unlike the features presented in the previous section, which are strictly local, they make pixel-level inferences (velocity, membership) over the whole image domain, relying on global optimizations.
- They play the same role in our proposals; i.e., are used to feed a modicum of global information into local features.

Both optical flow and image segmentation are canonical—yet unsolved—computer vision problems, and there is a very large body of work on either subject. In this section we will define the problem, some of the more relevant works, and give an intuition on how we integrate these features in our work.

2.3.1 Optical flow

Optical flow is the apparent motion of objects on a 2D plane caused by the relative motion between an observer and the rest of the scene. While this subject finds its roots in biology and psychology (Gibson, 1950), it has become a field of study in modern computer vision and robotics, where it is used to refer to the problem of estimating the 2D motion across sequences of time-ordered images, as discrete displacements or continuous velocities, that we call flow field or optical field estimates (see Fig. 2.8 for an example). Given assumptions on the structure of the flow field and the enviroment, optical flow can also be used to recover 3D motion (to within a scale factor), which is often referred to as scene flow (Vedula et al., 1999; Huguet and Devernay, 2007; Wedel et al., 2008). Aside from its importance as a stand-alone problem, flow fields have been used in many computer vision applications, including object detection (Dalal

(a) (b)

Figure 2.9: **(a)** We follow the color-coding scheme for two-dimensional flow fields of (Baker et al., 2011). This figure was reproduced from (Liu et al., 2011). **(b)** Two consecutive images from the MOSEG dataset (Brox and Malik, 2010a), along with their forward (left) and backward (right) flow fields, estimated with the optical flow algorithm provided by Sun et al. (2010).

et al., 2006), segmentation (Ochs and Brox, 2011; Fragkiadaki et al., 2012) and tracking (Brox et al., 2010), and pose estimation (Fragkiadaki et al., 2013).

Despite rapid progress in the field, most optical flow algorithms follow the seminal work of Horn and Schunck (1981), which combines a data term that assumes constance of some image property, such as intensity, with a spatial term that models the expected variation of the flow fields across the image. Their formulation relies on brightness constancy and the assumption that the flow fields are smooth, and is not robust to outliers. It has been since improved on by many researchers, but remains surprisingly competitive combined with modern optimization methods that were intractable at the time of its writing: see (Sun et al., 2014) for details.

Optical flow plays a small role in our work and a thorough revision of the field falls outside of the scope of this document. A classic survey of optical flow algorithms is that by Beauchemin and Barron (1995). Modern optical flow algorithms are usually benchmarked over the Middlebury dataset (Baker et al., 2011) and more recently with the KITTI dataset (Geiger et al., 2013). For an up-to-date reference on the state of the art please refer to (Sun et al., 2014). Fig. 2.9 shows and example for flow fields given a pair of images and the color-coding scheme used throughout this document. We use optical flow cues to build spatiotemporal feature descriptors in chapter 3.

A relevant, related work is SIFT-flow (Liu et al., 2011), an optical flow algorithm relying on feature descriptors for the data term. We use SIFT-flow for large-displacement motion estimation in chapter 4 (see Sec. 2.4.2 for details).

2.3.2 Segmentation

Segmentation refers to process of assigning a label to every pixel in an image, so that pixels with the same label share some properties, such as intensity, color or texture. This is a very broad definition that spans both basic pixel-wise similarity measures such as simple thresholding over intensity values, and complex semantic segmentation algorithms that aim to not only segment a whole object but also identify the category it belongs to.

Again, providing a thorough overview of the origins and state of the art in image segmentation falls beyond the scope of this thesis—the subject is simply too broad. The problem has been attacked from many angles, including simple binary tresholding (Klaus and Horn, 1986), k-means or mean-shift clustering, edge detection techniques (Maire et al., 2008), graph partitioning methods (Boykov et al., 2001), binary partition trees (Salembier and Garrido, 2000), watershed transformations (Najman and Schmitt, 1996), random walks (Grady, 2006) and many others, up to powerful semantic representations (Carreira et al., 2012). We apply segmentation cues to feature descriptors in chapter 4, and to object recognition in chapter 5. For the former, we use 'soft' segmentation cues, while for the latter we rely on oversegmentation with superpixels, which are 'hard' segments.

In chapter 4, we exploit segmentation cues to generate figure-ground segmentations on a per-pixel basis. We rely on soft segmentations that are used as intermediate, mid-level cues by two state-of-the-art edge detection algorithms: *Pb* (Maire et al., 2008) and *Gb* (Leordeanu et al., 2012). We also show how to extend this approach to work directly on edge cues, for which we use a recent detector that obtains a performance comparable or superior to both *Gb* and *Pb*, while significantly relaxing their computational requirements: the structured forests detector (Dollár and Zitnick, 2013). Notice that these are all 'soft' cues. The former (soft segmentations) provide pixel embeddings into a low-dimensional subpace which captures information about the different regions in the image; similar values indicate that pixels are likely to belong to the same region. Boundaries, in turn, measure the probability that adjacent pixels belong to the same region.

In chapter 5 we use the spatial support provided by SLIC superpixels (Achanta et al., 2012) to augment HOG features used by sliding window detectors. These 'hard' superpixels cluster image regions with similar properties (Fig. 1.5-(d)).

Note that neither of these rank among the most sophisticated image segmentation techniques. The soft segmentations we obtain from (Maire et al., 2008) and (Leordeanu et al., 2012) were introduced not to be used by themselves, but as input to their respective boundary detection systems. Despite its excellent performance, the structured forests detector extracts simple features and can run at multiple frames per second, and for some configurations, in real-time. SLIC superpixels were also designed with efficiency in mind and are often used as a pre-processing step to cluster pixels into over-segmented regions (over-segmentation occurs when regions corresponding to the same object are split apart, and is more lenient than under-segmentation, where regions belonging to different objects are clustered together). Our motivation for relying on these algorithms is grounded on the problems we tackle, such as dense descriptors and sliding window detection, which require practical, efficient solutions. Our choices will be substantiated in the relevant chapters.

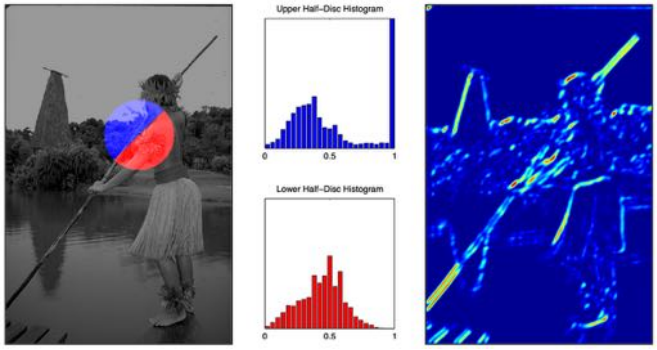


Figure 2.10: To build the posterior probability of boundary $Pb(x, y, \theta)$, consider two half-disks at angle θ, centered on pixel (x, y), as pictured in the left panel. The plots in the middle panel show the histograms of intensity values for each half-disk; in practice, more channels are used. The distance between histograms can be understood as a measure of the magnitude of the gradient at point (x, y) and angle θ, and thus as a measure of Pb. The figure on the right panel shows the Pb for each pixel, for the specific scale and orientation pictured in the left panel. Figure reproduced from (Arbeláez et al., 2011).

2.3.2.1 'Eigen' soft segmentations (Pb detector)

We explore two different approaches to extract the soft segmentations that we use as pixel embeddings. First, we turn to state-of-the-art boundary detector of Maire et al. (2008). The segmentation problem is coupled to that of contour detection or figure-ground assignment, as region boundaries and edges are closely related. Edge detection techniques have been used as a cue for segmentation algorithms, and vice-versa.

There is of course a large body of work on boundary detection, but most approaches fall into one of two groups. The first one consists of algorithms that use local information to determine the presence of a boundary, and includes classical approaches such as Canny (Canny, 1986) or oriented energy filters (Perona and Malik, 1990). Modern, more discriminant approaches are more sophisticated, employing a combination of cues and learning techniques. The second group consists of those that try to extract a global impression of the image. The Pb boundary detector belongs to this family, and builds on the seminal paper on normalized cuts (Shi and Malik, 1997).

Shi and Malik (1997) approach perceptual grouping as a global optimization problem, and rely on spectral graph theory to make it tractable. They construct a weighted graph where nodes are the points in the feature space (pixels) and edges encode the similarity between every pair of pixels. The latter is obtained in terms of the 'intervening contour' cue, which measures the presence of strong boundaries between two pixels. One can phrase the segmentation problem as a global optimization of the *normalized cut* objective defined on the (discrete) labelling of this graph, which is NP-hard. Relaxing this problem, however, yields a tractable, generalized eigenvector problem.

Maire et al. (2008) follow this formulation, using richer features, built on a combination of multiple cues as input to the spectral clustering stage. Their detector relies on the posterior probability of boundary of Martin et al. (2004), $Pb_\sigma(x, y, \theta)$, which measures the difference in multiple image channels (local brightness, color, texture) at different orientations. Fig. 2.10 illustrates this technique. In order to detect both fine

and coarse structures, they propose to repeat this process at multiple scales. We can use the resulting, multiscale Pb, called mPb, to estimate the affinity between pixels, by looking for an 'intervening contour', i.e. the presence of large gradients between them. Maire et al. (2008) use the maximal value of mPb along the line connecting two pixels. These local affinities are subsequently 'globalized' by finding the eigenvectors of the relaxed normalized cut criterion:

$$(D - W)\mathbf{v} = \lambda D\mathbf{v}, \tag{2.8}$$

where $\mathbf{v} \in \mathbb{R}^P$ is the relaxed solution for the P image pixels, W is the affinity matrix, which encodes the affinity between pixels, and D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$.

Even though this eigenvector problem can be solved exactly (albeit slowly), turning the computed eigenvectors into a segmentation is not obvious. Shi and Malik (1997) apply eigenvector-level clustering to obtain a partition into 'hard' regions. In practice, this often breaks regions where the eigenvectors have smooth gradients, a problem that has been addressed in following works. Maire et al. (2011) take the top M eigenvectors and extract their contours with Gaussian directional derivatives, at multiple orientations, and combine them into the *globalized* probability of boundary, gPb. In the final step, the gPb is used to extract image contours (for conveniency, we refer to this detector as 'Pb').

This is a post-processing technique, that does not necessarily offer the optimal solution to the segmentation problem, which is application-dependent. Instead, we propose to use the eigenvectors directly. In particular, we construct $\mathbf{y}(\mathbf{x})$ by weighting the first $M = 10$ eigenvectors by a quantity dependent on their corresponding eigenvalues:

$$\mathbf{y}(\mathbf{x}) = \left[\frac{1}{\sqrt{\lambda_1}}\mathbf{v_1}(\mathbf{x}), \ldots, \frac{1}{\sqrt{\lambda_M}}\mathbf{v_M}(\mathbf{x})\right]^T \tag{2.9}$$

so that lower-energy eigenvectors (global structures) have a stronger weight. These weighted eigenvectors $\mathbf{y}(\cdot)$ can be understood as *pixel embeddings*, which bring closer pixels likely to lie in the same region and pull apart those which do not belong together. We thus stay closer in spirit to the 'Laplacian eigenmaps' works of (Belkin and Niyogi, 2003). We can compute the affinity between two pixels as the euclidean distance in the embedded space. For simplicity, we refer to these pixel embeddings as 'Eigen'. See Fig. 2.11-(b) for indicative examples.

2.3.2.2 'SoftMask' soft segmentations (Gb detector)

As an alternative to the 'Eigen' embeddings, we also explore the *generalized* boundary detector Gb of Leordeanu et al. (2012). This paper presents a method to compute soft segmentations, which are one of multiple cues used as input to their boundary detection system.

Their method consists in representing semantic regions with color distributions, under the assumption that the color distribution for any patch in the image can be expressed as a linear combination of a finite number of color probability distributions on regions of interest. They use local color models, built around each pixel, to construct a large set of figure-ground segmentations. These segmentations are then projected onto a lower dimensional subspace through PCA. As before, we can take the top $M = 8$

components, which provides us with low-dimensional pixel embeddings. The main advantage of these features is that they are obtained at a substantially smaller computational cost, whereas building and solving for Eq. 2.8 is expensive for any but the smallest images.

We use their soft segmentations directly, without further processing. We refer to these embeddings, and their associated masks, which will be introduced in chapter 4, as 'SoftMask'. Fig. 2.11-(c) shows examples over several images. Note that the 'SoftMask' embeddings have higher granularity than the 'Eigen' embeddings; they are noisier, but also better able to capture smaller features. We display some boundary maps in Fig. 2.11-(d).

2.3.2.3 Structured forests detector

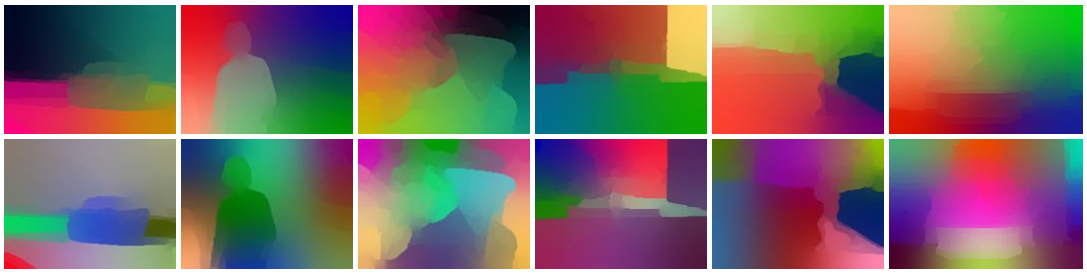
The third method we explore relies on the state-of-the-art structured forests boundary detector of Dollár and Zitnick (2013), which has excellent detection performance while operating at multiple frames per second. This approach learns decision trees with structured learning techniques, mapping structured labels to a discrete space on which standard information gain measures can be evaluated. Its performance is comparable or superior to the state of the art in standard datasets, and can run (for some settings) in real-time. Note that unlike the two previous methods, boundary data does not provide an embedding, but rather measures the probability that two adjacent pixels may belong to different regions. As such we cannot measure the affinity between pixels with their euclidean distance, as we do for 'Eigen' and 'SoftMask'. In Sec. 4.4 we show how to adapt the 'intervening countour' technique of Shi and Malik (1997) for this purpose.

2.3.2.4 SLIC superpixels

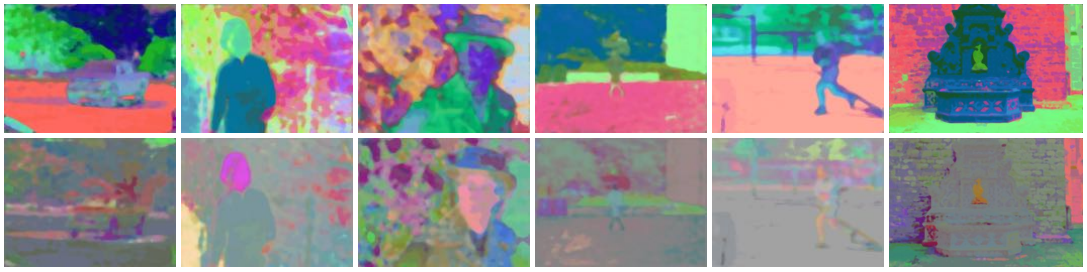
We can define superpixels as perceptually meaningful, atomic regions. The motivation behind them is to replace the rigidity of the pixel lattice, dispensing with the redundancy typically found in images. Superpixel algorithms do not attempt to provide a exact segmentation of object boundaries. In practice, they are often used to over-segment the image, grouping pixels that belong to the same object, which serve as meaningful primitives that can be used at higher abstraction levels, while reducing their complexity. Computer vision applications are becoming increasingly reliant on them, and unsupervised over-segmentation of an image into superpixels is a common pre-processing step of many algorithms.

What makes a good superpixel algorithm depends on the application, but speed, regularity and adherence to object boundaries are primary concerns. Many state-of-the-art techniques rely on minimizing objective functions to enforce color homogeneity, often with graph-based approaches or by growing superpixels, which can be computationally expensive. Achanta et al. (2012) introduced the Simple Linear Iterative Clustering algorithm (SLIC), a superpixel algorithm based on a local version of k-means clustering in the 5-dimensional space of color and space. It can be used to generate regular superpixels with good boundary recall and a very low computational overhead, which makes it of practical use in many computer vision applications. Fig. 2.11-(e) shows some examples of SLIC superpixels computed at different scales. We rely on the implementation of VLFEAT (Vedaldi and Fulkerson, 2008), which has two parameters

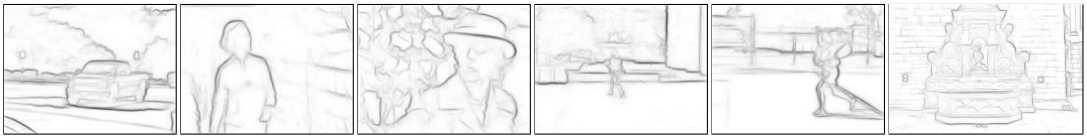
(a) Source image



(b) 'Eigen' embeddings. Top row: eigenvectors [1-3] in RGB space. Bottom row: [4-6].



(c) 'SoftMask' embeddings. Top row: soft segmentations [1-3], in RGB space. Bottom row: [4-6].



(d) Boundaries extracted with the structured forests detector.

(e) SLIC superpixels, at three different scales. Each superpixel is filled with its average RGB value.

Figure 2.11: The segmentation cues we use in our work as mid-level data.

that allow us to to tune the size of the superpixels and their regularity. In chapter 5 we exploit the spatial support of multi-scale SLIC superpixels to build very fast segmentations given an arbitrary bounding box, and show how to exploit them in the context of sliding-window detectors.

(a) (b)

Figure 2.12: Two stereo datasets with (a) narrow and (b) wide baselines. The narrow-baseline case contains significant partial occlusions, which are a rather exceptional situation. The wide-baseline data contains rotations, scalings and partial occlusions.

2.4 Applications

The algorithms presented in this dissertation target three different applications: wide-baseline stereo, large-displacement motion, and object detection. These are all canonical computer vision problems, and have been, and remain, extensively researched. This section will provide a background on the problem at hand, the tools we use to approach it, and the datasets used to evaluate our techniques.

2.4.1 Wide-baseline stereo

Stereo reconstruction can be formulated as the problem of computing a 3D representation of a scene given images extracted from different viewpoints. It is a classical computer vision problem, and has been studied for several decades. The stereo problem can be divided into *narrow-* and *wide*-baseline stereo, according to the spatial distribution of the cameras. There is not a hard division between these two categories, but narrow-baseline is usually concerned with rigs of two cameras separated by a short distance, and pointing in the same direction, while wide-baseline refers to the general case of cameras in any configuration. Fig. 2.12. illustrates the difference between the two. Wide-baseline stereo is a particularly relevant problem for our times, as cameras and cameraphones have become ubiquitous and the amount of data freely available on photo sharing websites has grown dramatically.

While narrow-baseline stereo is a well-understood problem, the same cannot be said for its wide-baseline counterpart. Narrow-baseline stereo is often attacked with simple similarity measures such as pixel differencing, SSD or NCC (see Sec. 2.2). As the viewpoint increases, the photo-consistency assumption weakens, perspective distortion and occlusions become a problem, and we cannot rely on such simple metrics. It is in these circumstances that feature descriptors come into play. Wide-baseline stereo has often been addressed as a multi-step process, using sparse matches as anchors or seeds (Strecha et al., 2003; Yao and Cham, 2006), which can result in gross reconstruction errors if the first matching stage is inaccurate. In this thesis we are interested in dense wide-baseline stereo, where we match two sets of dense descriptors to obtain dense depth maps in a single step.

2.4.1.1 Stereo camera calibration

Most stereo reconstruction algorithms require a geometrically calibrated system, which refers to the estimation of internal and external camera parameters. For pinhole cam-

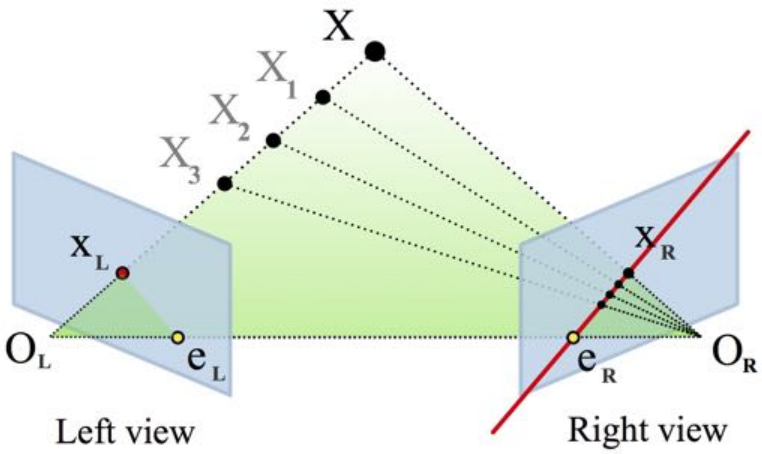


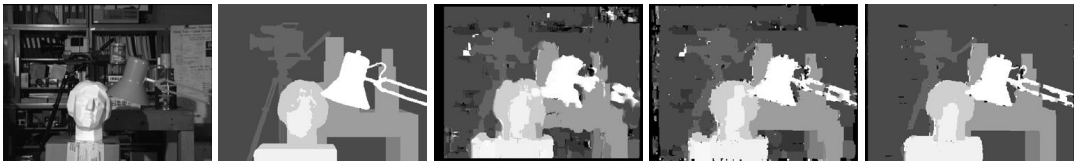
Figure 2.13: Epipolar geometry. Two pinhole cameras are looking at a 3D point $\mathbf{X}$, which is transformed by a perspective projection into 2D points $\mathbf{x_L}$ and $\mathbf{x_R}$ over the image planes for the left and the right cameras, respectively. $\mathbf{O_L}$ and $\mathbf{O_R}$, the centers of the projection for each camera, project into $\mathbf{e_L}$ and $\mathbf{e_R}$, respectively—the *epipoles*. The line $\mathbf{O_L}$-$\mathbf{X}$ is seen as a single point by the left camera, and as a line by the right camera, which lies on $\mathbf{e_R}$-$\mathbf{x_R}$ over the image plane—the *epipolar line*. Likewise, the plane determined by $\mathbf{X}$, $\mathbf{O_L}$ and $\mathbf{O_R}$ is the *epipolar plane*. Thus: for a calibrated stereo system, we can constrain the matches for a point on the image plane of the first camera to the corresponding epipolar line on the second camera. Figure by Arne Nordmann, released under CC-BY-SA-3.0.

era models, internal parameters include sensor properties such as focal length and lens distortion. External parameters encode the relative position between the cameras and with respect to a reference coordinate system. If the camera system is calibrated, we can use epipolar geometry to constrain the number of possible matches, so that a point on the image plane of the first camera must correspond to a point over a single line on the image plane of the second camera (or a fraction thereof, if we account for the range of the scene). This process is illustrated in Fig. 2.13. Alternatively, stereo algorithms targeting narrow baselines often apply a transformation process known as *image rectification* to project the images onto a common image plane, so that the correspondences can be queried over a horizontal line, which has computational advantages.

In chapter 3 we build our own dataset and use the Matlab Camera Calibration Toolbox (Bouguet, 2013) to calibrate the cameras. In chapter 4, we use a wide-baseline stereo dataset with calibration data (Strecha et al., 2008). We constrain the problem to epipolar matching in either case. For a comprehensive treatise of multiple view geometry and camera calibration, please refer to Hartley and Zisserman (2004).

2.4.1.2 Global optimization

Photometric constraints, i.e. metrics encompassing both simple pixel differencing and sophisticated feature descriptors, are not enough to solve complex stereo problems satisfactorily. Modern stereo algorithms use local features to estimate the similarity between points, and then impose global shape constraints to enforce spatial consistency.



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

Figure 2.14: Graph cuts for binocular stereo. (a) A frame from the Tsukuba stereo dataset—only one channel is shown. (b) Ground truth disparity. The following show reconstructions with (c) NCC, (d) simulated annealing, and (e) graph cuts.

This approach is not restricted to stereo: many early vision problems require estimating noisy, spatially-varying variables. These variables are often piecewise-smooth, i.e. they change smoothly inside an object and brusquely at object boundaries—notice the similarities with optical flow.

We can formulate these problems in terms of discrete energy minimization, where we want to assign every pixel $p \in \mathcal{I}$ to a finite set of L labels $f_p \in \mathcal{L}$, such as pixel disparities for narrow-baseline stereo (or 2D displacements for optical flow). Given a labeling for all pixels f, the energy is defined as the linear combination of two terms, $E(f) = E_d(f) + E_s(f)$, which encode, respectively the disagreement between the label assignment and the observed data; and the smoothness of the label assignment.

The data energy term is often formulated as the sum of the data costs for each separate pixel:

$$E_d(f) = \sum_{p \in \mathcal{I}} D_p(f_p) \tag{2.10}$$

where D_p encodes the quality of the assignment of labels to pixels—e.g. the distance between feature descriptors. There have been many proposals for the smoothness term E_s. A common formulation is:

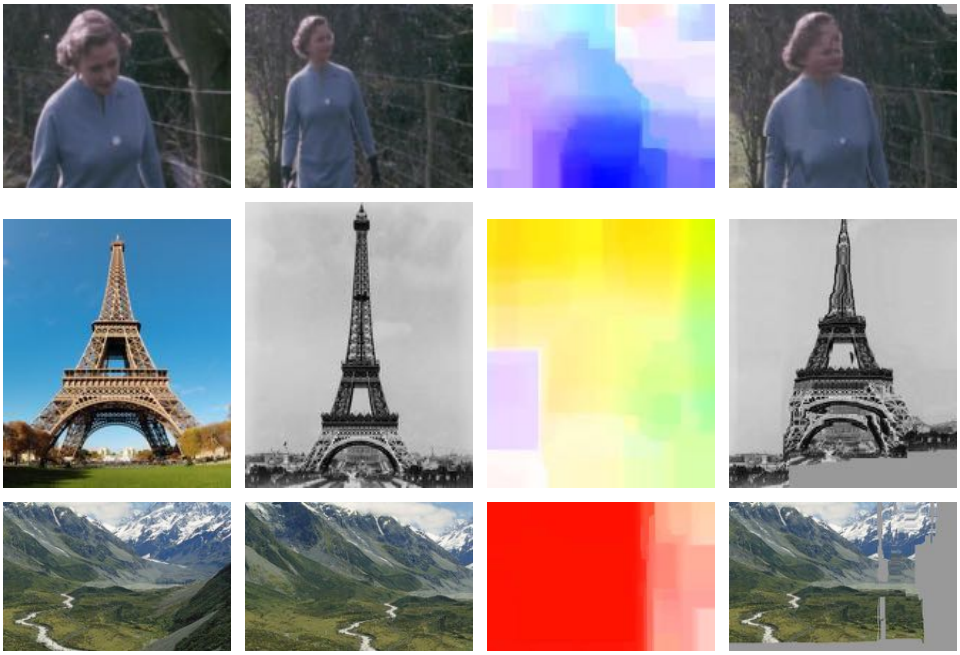
$$E_s(f) = \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q) \tag{2.11}$$

where p and q are two pixels with labels f_p and f_q, $\mathcal{N}$ is the set of interacting pixels, and $V_{p,q}$ a penalty function. The literature often refers to the data term E_d as the *unary* term and the smoothness term E_s as the *pairwise* term. A simple approach for the pairwise term is to restrict $\mathcal{N}$ to the set of adjacent pixels, and penalize differing labels on contiguous pixels:

$$V(\alpha, \beta) = \gamma \cdot T(\alpha \neq \beta) \tag{2.12}$$

where α and β are two labels, γ is the smoothness cost, which encodes the trade-off between the data and smoothness terms, and $T(\cdot)$ is 1 if the argument is true and 0 otherwise—we call this the Potts model.

Global minimization of energy functions of this form is NP-hard, even for the simplest discontinuity-preserving cases. On the other hand, they can often be reduced to instances of the maximum flow problem in a graph, for which accurate, fast approximations exist. The seminal work of Boykov et al. (2001) introduced efficient techniques to compute good, local minima based on graph cuts, which allows for the pairwise term of Eq. 2.12, among others. Graph-based energy minimization is by itself a field of research



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

Figure 2.15: Image registration with SIFT-flow. We register images **(a)** with **(b)**. **(c)** shows the flow fields. **(d)** shows image (b) warped over (a). The first row shows two images of the same scene, with translation and scaling effects. The second row shows two views of the same object, similar in shape but very different in appearance. The third row shows two windows cropped from the same (larger) image.

with many applications in computer vision. In this work, we employ two algorithms: graph cuts (Boykov et al., 2001), and tree-reweighted message-passing (TRW-S) (Kolmogorov, 2006), which is shown to obtain energy solutions lower than those of graph cuts on stereo problems with Potts pairwise costs. We use the former in chapter 3 and the latter in chapter 4. Many pixel labeling problems can be attacked with these strategies. Fig. 2.14 shows an application of graph cuts to binocular disparity estimation.

We now explain how to use these tools for stereo reconstruction. First, we rely on the calibration data to discretize space into L depth labels. We compute the unary costs as the distance between pairs of descriptors, subject to the epipolar constraints, and store the cost for the best match for every depth layer, thus building a cube of costs size $W \times H \times L$, where W and H are the width and height of the image. We use the Potts model for the pairwise costs, penalizing differing labels on adjacent pixels (with 4- or 8-connectivity). For wide-baseline stereo, we can also add an occlusion label with a constant cost to the graph (see Sec. 3.4). We then solve with either graph cuts or tree-reweighted message-passing[1].

[1]We used different algorithms for technical reasons. We do not compare one to the other, but rather use them, analogously, in different works: graph cuts in (Trulls et al., 2012), and TRW-S in (Trulls et al., 2013).

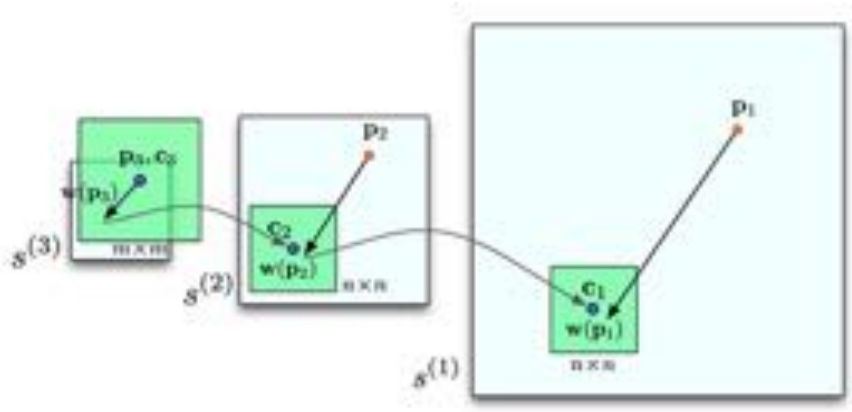


Figure 2.16: The SIFT-flow coarse-to-fine matching strategy. For simplicity, a single image is shown. A SIFT pyramid $\{s^{(k)}\}$ is built from $s^{(1)}$, which indicates the dense descriptor set; further pyramid levels are smoothed and downsampled from the previous level. This allows the matching algorithm to search over larger areas for the coarser levels (left) in fewer steps, making the optimization tractable. The flow estimates are refined in the finer levels (right). Image from (Liu et al., 2011).

2.4.2 Large-displacement motion

In this section we present SIFT-flow (Liu et al., 2011), an algorithm related to optical flow which uses densely sampled SIFT descriptors instead of raw pixel values for the unary matching. SIFT-flow is designed for image alignment, but unlike optical flow it is not meant to work only for two views of the same scene, but also different instances of 3D scenes belonging to the same category; hence the use of dense SIFT, which has been successful in image registration. Fig. 2.15 shows some examples of the scene alignment problem.

SIFT-flow follows the traditional optical flow formulation, with a data term, a smoothness term and a small displacement term. The data term encodes the match between SIFT descriptor pairs given a displacement, and the smoothness term enforces spatial regularization; the small displacement term constrains the flow vectors to be as small as possible when no other information is available. The objective function is optimized with a dual-layer loopy belief propagation algorithm. Differently from traditional optical flow formulations, SIFT-flow decouples the horizontal and the vertical flow in the smoothness term, for computational reasons. Even so, the direct optimization of this objective function remains largely intractable. This is due to the fact that scene alignment requires large search windows, as by definition a pixel in one image can match any pixel in the other image. This is addressed with a coarse-to-fine matching scheme with a SIFT pyramid: see Fig. 2.16. In a follow-up work, Kim et al. (2013) present a dense matching method that simultaneously regularizes match consistency at multiple levels—the entire image, coarse grid cells, and every single pixel—overcoming the rigidity of the spatial pyramids.

The code is publicly available and, in addition, it is amenable to any feature descriptor that can be computed densely. We use this framework to evaluate the performance of our segmentation-aware descriptors in chapter 4, and to a lesser degree in chapter 5, for the same problem.

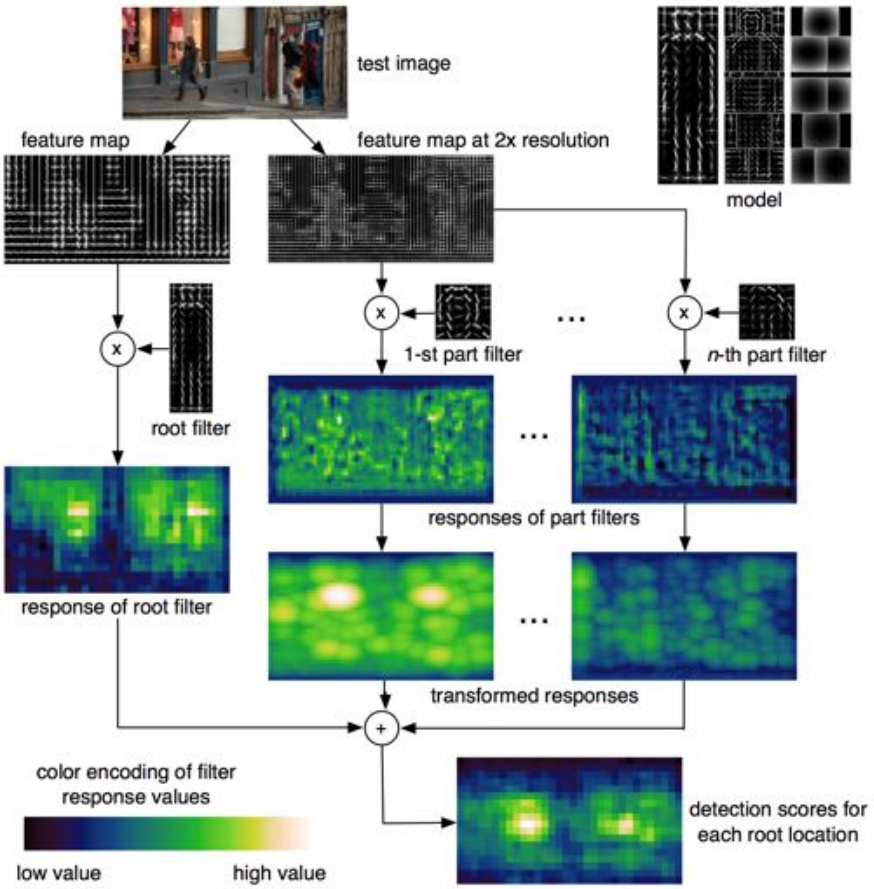


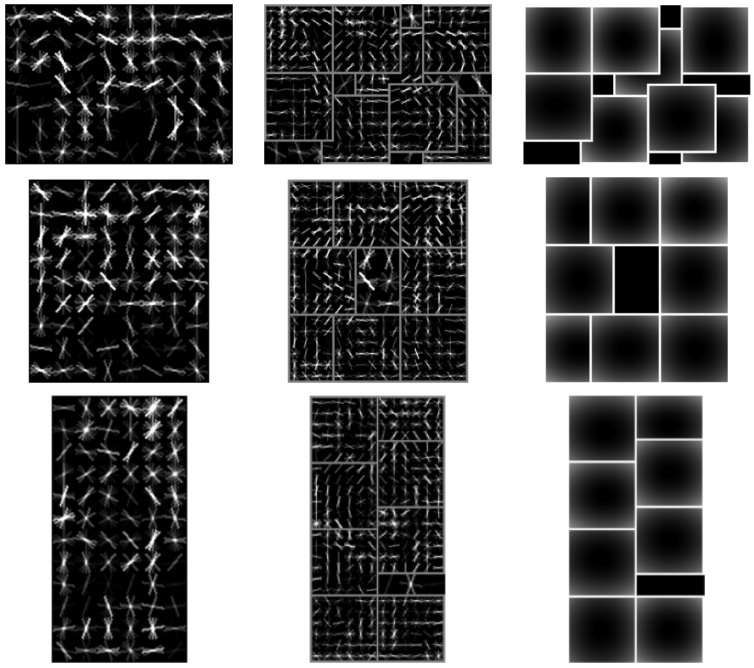
Figure 2.17: The DPM detection pipeline consists of a sliding window detector with root and part filters. The features for the part filters are computed at twice the resolution. The figure shows one root filter (left) and two part filters (center and right). The process is repeated at multiple scales. This figure was reproduced from (Girshick, 2012).

2.4.3 Object detection

Object detection, i.e. finding and identifying objects in an image, is one of the fundamental problems in computer vision. It is a very challenging task, as objects can vary greatly in appearance, illumination and viewpoint, as well as suffer from non-rigid deformations, occlusions and intra-class variation. Many approaches have been designed, going back to the very beginnings of the field.

Deformable Part Models (DPM) (Felzenszwalb et al., 2010b) formulate the problem of object recognition at the category level. They build on the work of Dalal and Triggs (2005), which combines HOG features and linear SVM training into a sliding window detector, and extend it with pictorial structures (Felzenszwalb and Huttenlocher, 2005). A DPM filter incorporates multiple HOG filters, one which serves as an anchor or 'root', plus local 'part' filters that can shift around it. The root filter models the global appearance of the object, while part filters enrich the models to better represent the data, and can capture distortions and intra-class variation.

The score for a specific arrangement of a root filter x_0 and n part filters $x_1, \ldots, x_n$



(a) (b) (c)

Figure 2.18: DPM filter for the 'horse' category, with three clusters. **(a)** Weights for the root filters. **(b)** Weights for the part filters, overlaid on top of the root filter. **(c)** Location of the part filters with respect to the root filter.

is given by the combination of a unary and a pairwise term:

$$S(x_0, x_1, \ldots, x_n) = \sum_{p=0}^{n} \langle w_p, G(x_p) \rangle + \sum_{p=1}^{n} D_p(x_p, x_0), \tag{2.13}$$

where $G(x_p)$ indicates the image-based features at position x_p, w_p is the template for part p, $\langle w_p, G(x_p) \rangle$ is the score obtained for placing part p in position x_p, and $D_p(x_p, x_0)$ is a quadratic penalty function that measures the spatial compatibility between the positions of part p and the root. The hypothesis of an object being present at a specific location is thus determined by the individual contributions of root and part filters matched with the HOG features, as their inner product, and penalized by discrepancies in the location of the parts with respect to their anchor positions are penalized. The DPM detection pipeline is pictured in Fig. 2.17.

This hierarchical approach to object modelling is very powerful, but hard to train, and Felzenszwalb et al. showed how to address many practical problems in their application. Compared to the standard HOG filter, the DPM framework incorporates more sophisticated machine learning techniques, such as an approach for data-mining hard negative examples with latent (hidden) variables during SVM training, and a number of engineering tricks of varying complexity, including: unsupervised left/right orientation discovery; a weakly supervised algorithm to determine the location of the parts (which are not generally labelled in the training data); mixture models using multiple clusters divided by aspect ratio; and features computed at twice the resolution for the part filters (bins of 4×4 pixels) than the root (8×8). Fig. 2.18 shows a representation

of the model used to detect the foal in Fig. 1.5, trained over the 'horse' category in PASCAL VOC. The model was applied at every image location using the detection pipeline outlined in Fig 2.17, and at multiple scales, resizing the target image and recomputing the HOG features.

This framework won the 2007 PASCAL VOC detection task, and provided a solid basis for extensions and refinements that have been continually raising the bar for object detection and pose estimation. Most works building on DPMs rely on standard HOG features[2]. In chapter 5 we propose a technique to combine bottom-up segmentation, in the form of SLIC superpixels, with DPMs. We follow a similar approach as in chapter 4 for descriptors, in that our goal is to 'clean up' the low-level features. Unlike that work, however, we do not commit to a single segmentation: we use a large pool of SLIC superpixels at multiple scales, and given a detection hypothesis (a window over the image) combine them in a scale-, position- and object-dependent manner to build soft segmentation masks. We devise an algorithm fast enough that can be applied to every hypothesis of a sliding-window detector, and show how to use it to build background-invariant features to train DPMs with.

[2]To be precise, Felzenszwalb et al. (2010b) use PCA to reduce feature dimensionality from 36 to 11. See chapter 5 for details.

Chapter 3

A spatiotemporal approach to wide-baseline stereo

Shape from stereo and shape from motion are two of the most well-known problems in computer vision. Their integrated research, however, has received comparatively little attention. In this chapter we present a spatiotemporal approach to stereo reconstruction. Existing works are often based on features which operate on the spatiotemporal volume, i.e. a stack of frames captured at different times, around the feature point. This is not applicable to wide-baseline setups, as the volumes generally cannot be matched as-is, due to large perspective changes. We propose an alternative solution to build spatiotemporal features based on the same principle, while avoiding this shortcoming. The core idea of our technique is to capture the *temporal changes* around a point from a given viewpoint, which we do by means of augmenting the descriptor features with the aid of optical flow priors. Fig. 3.1 illustrates the reasoning behind our design. This chapter is based on our 2012 ECCV paper (Trulls et al., 2012).

3.1 Introduction

Ever since the advent of SIFT (Lowe, 2004), feature descriptors have become an indispensable tool in matching, recognition, and retrieval. The current trend for new descriptors is to design fast an efficient algorithms, to facilitate their use in real-time applications (see Sec. 2.2 for details). A different thread of works, such as dense SIFT (Vedaldi and Fulkerson, 2008), Daisy (Tola et al., 2010) and SID (Kokkinos et al., 2012b) have demonstrated that it is possible to efficiently compute descriptors *densely*—i.e. for every pixel in the image—and use them as a generic low-level image representation.

Stereo reconstruction is generally performed by matching descriptors from two or more different images to determine the quality of each correspondence, and applying a global optimization scheme to enforce spatial consistency (see Sec. 2.4.1). This approach fails on scenes with poor texture or repetitive patterns, regardless of the descriptor or the underlying optimization scheme; wide baselines exacerbate the problem. In

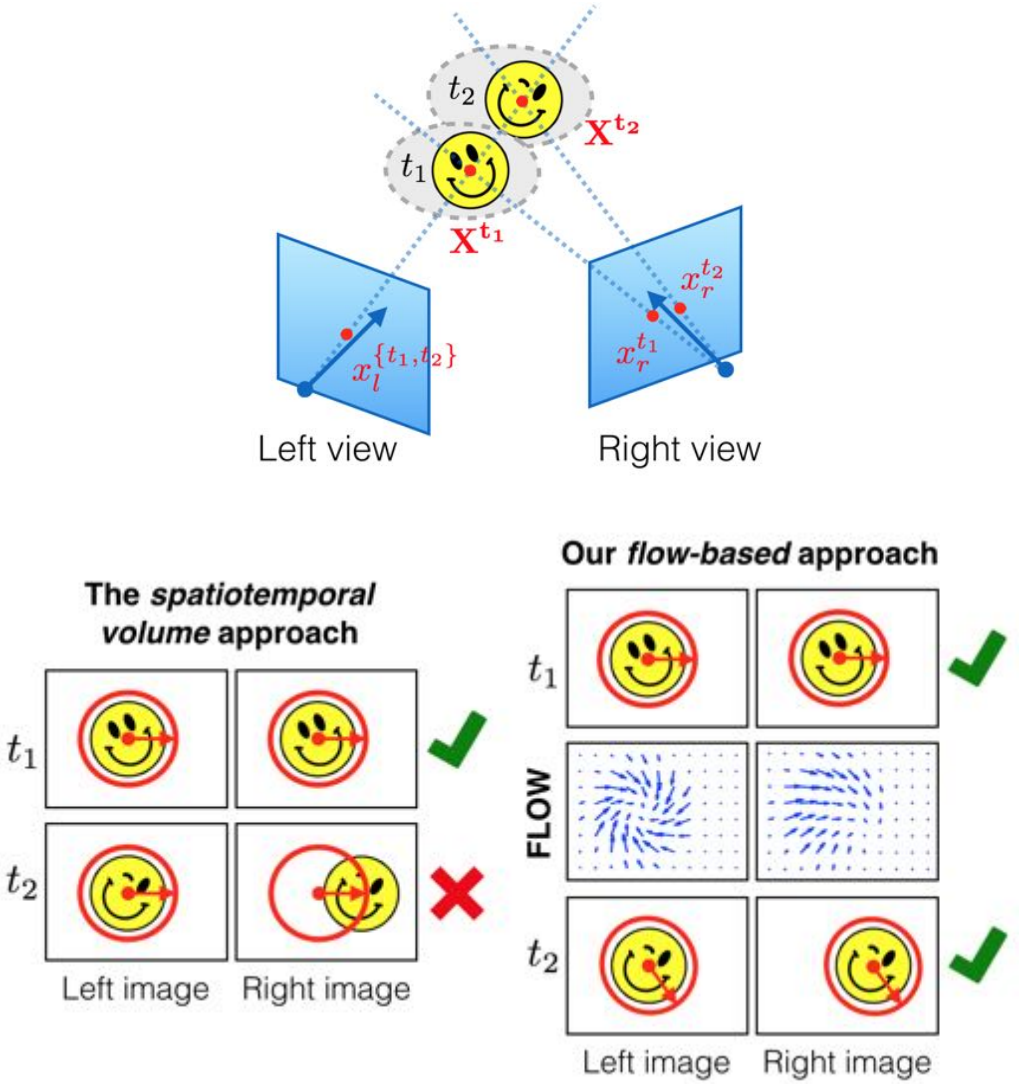


Figure 3.1: How can we build spatiotemporal features that remain reliable across wide baselines? **Top image:** we show a scene containing a single moving object at two different times, captured with a wide-baseline stereo setup. A point over the object is represented with a red dot. **Bottom left:** Existing works tackle this problem operating on the *spatiotemporal volume*, stacking features based on 2D gradients over different frames, or alternatively computing 3D gradients over the frame stack. The red circle indicates the image patch used to compute the descriptor marked with the red dot, and the red arrow indicates the orientation of the patch. Notice how this strategy does not hold for wide baselines, as the left and right channels change differently across time, according to their respective perspective transformations. **Bottom right:** In our approach, optical flow cues are computed for each camera separately and used to 'warp' the reference frame used to build features across different time frames. This allows us to stay fixated on the feature point, and build *consistent spatiotemporal features* which capture the dynamic information around the point—represented here by a 'wink' on t_2. This figure is exaggerated for effect.

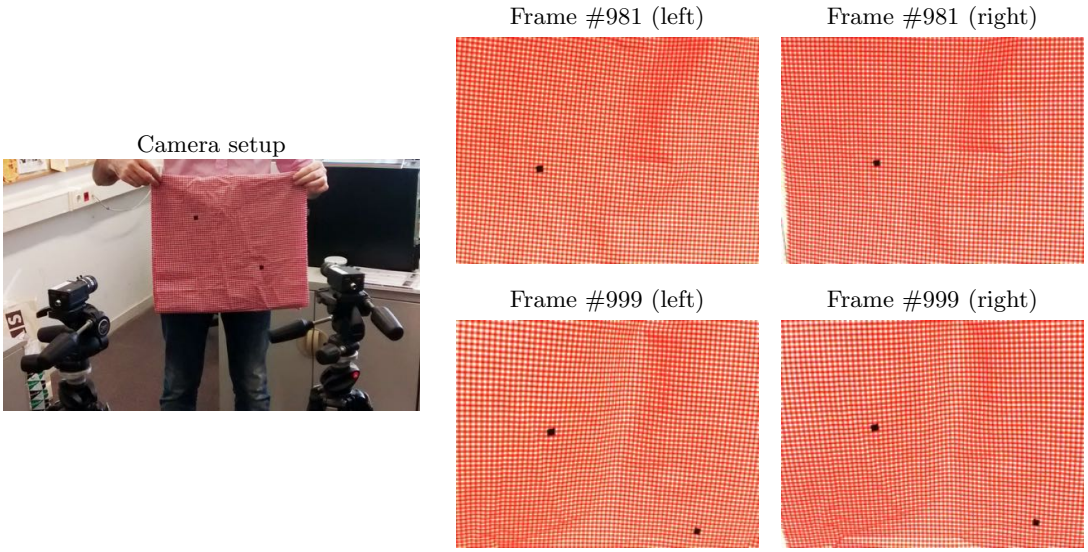


Figure 3.2: Samples from a highly-ambiguous, wide-baseline stereo sequence. Notice how hard it is to find correspondences with the naked eye; the black markers were in fact a visual aid to help us ensure that both cameras were properly aligned and looking at the same area while recording the sequence. We demonstrate that we can solve these ambiguities incorporating temporal information.

these situations we can attempt to solve the correspondence problem by incorporating dynamic information.

One approach for doing with descriptors based on oriented gradient histograms, such as SIFT and its variants, is to simply extend them to work over the spacetime volume, i.e. a stack of frames. Since the local temporal structure depends strongly on the camera view, most efforts in this direction have focused on monocular action recognition (Scovanner et al., 2007; Derpanis et al., 2010; Kläser et al., 2008; Laptev and Lindeberg, 2003).

For stereo reconstruction, spatiotemporal descriptors computed in this manner should be oriented according to the geometry of the camera setup. This approach was applied by Sizintsev and Wildes (2009) to disparity estimation for narrow-baseline stereo, but it remains unclear how to apply such a scheme to wide-baseline scenarios.

In this chapter we present a spatiotemporal approach to 3D stereo reconstruction applicable to wide-baseline stereo, based on augmenting 2D—i.e. single-frame—descriptors with optical flow priors instead of relying on 3D primitives over the spacetime volume. We compute, for each camera, dense Daisy descriptors for a given frame over the spatial domain, and then extend them over time using the flow fields as cues to warp the reference frame, which is used to build new Daisy descriptors. We finally concatenate the 2D features into a single 3D descriptor. As an additional contribution, we show how to apply a global optimization algorithm over the spatiotemporal domain, to enforce both spatial and temporal consistency.

We apply this technique to dynamic sequences of non-rigid objects with a high number of ambiguous correspondences and evaluate it on both synthetic and real sequences. We show that its reconstructions are more accurate and stable than those obtained with state-of-the-art descriptors. In addition, we demonstrate that our ap-

proach can be applied to wide-baseline setups with occlusions, and that it performs very robustly against image noise. An example of the kind of data our algorithm can work on is shown in Fig. 3.2.

These are the most important strengths of our approach:

Viewpoint invariance. We present an approach to build spatiotemporal descriptors applicable to any stereo baseline—a line of research largely unattended until now.

Robustness. Our descriptors rely on global optical flow estimates that prove reliable on complex scenes with large amount of image noise. We achieve improved results with spatiotemporal constraints. We also show how to handle occlusions.

Discriminating power. We prove that we can handle very complex ambiguities with the introduction of temporal data and constraints.

Our technique is composed of two parts: the spatiotemporal descriptors, presented in Sec. 3.3, and a spatiotemporal regularization scheme based on graph cuts, presented in Sec. 3.4. Our descriptors are based on Daisy and share many of its properties—in Sec. 3.5 we describe the algorithm for latent occlusion estimation introduced by Tola et al. (2010) and show how to extend it to our descriptors. Lastly, we present experiments on synthetic and real data.

3.2 Related work

SIFT remains the main reference among feature descriptors, showing great resilience against affine transformations on both the spatial and intensity domains. Given its computational cost, subsequent work has often focused on developing more efficient descriptors, such as PCA-SIFT (Ke and Sukthankar, 2004), GLOH (Mikolajczyk et al., 2005) or SURF (Bay et al., 2008), as well as binary descriptors and binarization techniques (see Sec. 2.2.6).

There remain a number of open problems in descriptor invariance, which include the treatment of non-rigid deformations, scale, and occlusions. Current techniques often accommodate scale changes by limiting their application to singular points (Lowe, 2004; Belongie and Malik, 2002; Bay et al., 2008), where scale can be reliably estimated. SID exploits a log-polar transformation to achieve scale invariance without scale detection. Non-rigid deformations have been seldom addressed with region descriptors, but recent advances have shown that kernels based on heat diffusion geometry can effectively describe local features of deforming surfaces (Moreno-Noguer, 2011). Regarding occlusions, Daisy demonstrated performance improvements in multi-view stereo from the treatment of occlusion as a latent variable.

Although regularization schemes such as graph cuts (Boykov et al., 2001) or formulations based on partial differential equations (Strecha et al., 2003) may improve the spatial consistency when matching pairs of images, in scenes with little texture or highly repetitive patterns the problem can be too challenging. Dynamic information can then be used to further discriminate amongst possible matches. Under controlled settings, the correspondence problem can be further relaxed via structured-light patterns (Zhang et al., 2003; Davis et al., 2003). Otherwise, more sophisticated descriptors need to be designed. For this purpose, SIFT-like descriptors have been extended to 3D data, although mostly for monocular cameras, as the local temporal structure varies strongly with large viewpoint changes. For instance, this has been applied to volumetric images on clinical data (Allaire et al., 2008), and to video sequences for action

recognition (Kläser et al., 2008; Derpanis et al., 2010; Laptev and Lindeberg, 2003; Rodriguez et al., 2008). One exception is the work by Sizintsev and Wildes (2009), who designed a spatiotemporal descriptor for disparity estimation on narrow baselines, based on spatiotemporal primitives called Stequels that can be reoriented and matched from slightly different viewpoints (see Sec. 2.2.3). The application of this approach to wide-baseline scenarios remains unexplored.

This problem is related to scene flow, i.e. simultaneously recovering 3D flow and geometry. In the work we present in this chapter, however, we do not intend to compute scene flow, as such techniques usually require strong assumptions such as known reflectance models or relatively small deformations to simplify the problem, and often use simple pixel-wise matching strategies that cannot be applied to ambiguous scenes or wide baselines. For instance, (Carceroni and Kutulakos, 2002) exploit a reflectance model to estimate the scene flow under known illumination conditions. Zhang et al. (2003) estimate an initial disparity map and compute the scene flow iteratively, exploiting image segmentation to maintain discontinuities. A more recent approach is that by Huguet and Devernay (2007), which couples dense stereo matching with optical flow estimation. This involves a set of partial differential equations which are solved numerically. More relevant to our work is the approach of (Wedel et al., 2008), which decouples stereo from motion while enforcing a scene flow consistent across different views; this approach is agnostic to the algorithm used for stereo.

We mention these works for completeness, but the method presented in this chapter is firmly embedded in the realm of feature descriptors, which we augment with the means of optical flow priors. Optical flow is often used as a cue in video applications. Numerous motion segmentation methods based on optical flow between pairs of frames have been proposed; e.g. Brox and Malik (2010b) use long-term point trajectories based on dense optical flow for object segmentation, whereas Palou and Salembier (2014) exploit flow information along with hierarchical segmentation to make inferences about occlusions, which are then used to extract depth order maps for monocular video. Using the flow estimates as features is relatively rare. Chaudhry et al. (2009) builds histograms of oriented optical flow estimates, which are used as features for human action recognition. To our knowledge, this is the first work that exploits optical flow cues in the construction of gradient-based low-level features such as descriptors.

3.3 Spatiotemporal descriptor

The approach presented in this section to build spatiotemporal descriptors could be applied, with minor tweaks, to any feature descriptor based on a spatial grid, such as SIFT or any of the variants listed in Sec. 2.2. We pick Daisy for the following reasons:

- It was designed from the ground up with efficiency in mind, at a relatively small size. This is particularly relevant to our 3D descriptor, as we can store the convolved orientation maps for different frames; this requires a relative increment in memory but saves computational time.
- Its grid structure can be very easily warped across time, unlike e.g. dense SIFT's, to follow the evolution of the patch around a point. Building the spatiotemporal descriptor is as simple as plucking additional values from the convolved orientation maps for different frames.

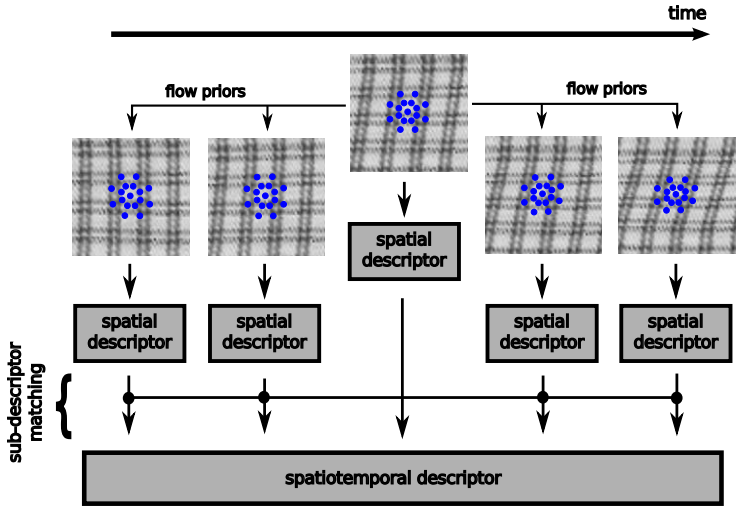


Figure 3.3: To build the spatiotemporal descriptors (for a single viewpoint), we first compute the Daisy descriptors for the central frame I_k, using an unwarped grid. Daisy descriptors for previous and past frames, $I_{k\pm b}, b \in \{1, \ldots, B\}$, are computed with a grid warped across time with the flow priors. To validate the latter, each descriptor is matched against the reference descriptor on I_k, and dropped if their distance exceeds a given threshold. Valid Daisy descriptors are then concatenated to create the spatiotemporal descriptor.

- It was specifically designed for wide-baseline stereo, with clever strategies to deal with occlusions which we will also take advantage of (see Sec.3.5).

We now describe how to compute dense spatiotemporal descriptors for a single camera. In the following section we will explain how to use two descriptor sets for stereo reconstruction.

To compute our spatiotemporal descriptors, we extend the Daisy pipeline in the following manner. For every new (gray-scale) frame I_k, we compute optical flow priors for each consecutive pair of frames from the same camera, in both the forward and backward directions: $\mathbf{F}^+_{k-1}$ (from I_{k-1} to I_k) and $\mathbf{F}^-_k$ (viceversa). To compute a full set of spatiotemporal descriptors for frame k using $T = 1 + 2 \cdot B$ frames we require frames $I_{k\pm b}, b \in \{1, \ldots, B\}$, and their respective flow priors. To compute the flow priors we use publicly available code from (Liu, 2009). Since the size of the descriptor grows linearly with T we use small values of B, between 1 and 5 (or equivalently, a total number of frames T from 3 to 11).

We then proceed with the standard Daisy pipeline (Sec. 2.2.2). For each frame we compute $H = 8$ oriented gradient maps, preserving gradient polarity, and repeteadly convolve them with gaussian kernels to obtain the convolved orientation maps, as many times as the number of rings on the descriptor grid, N; Tola et al. (2010) use $N = 3$. As explained in Sec. 2.2.2, building a Daisy descriptor is now just a matter of plucking the appropriate values from the convolved orientation maps, which nets us the gradient histograms. The size of the descriptor S is determined by H, N and the number of points per ring, K. We store the convolved orientation maps for frames I_{k-B} to I_{k+B}.

To build our spatiotemporal descriptor, we use the unwarped Daisy grid to compute the Daisy descriptor over the feature point on the central frame, F_k. We call this descriptor $\mathbf{D}_{\{k,0\}}$. We then warp the grid through time, from the central frame onto the others, by means of the optical flow priors. We translate each grid point independently, up to I_{k-B} (backward) and I_{k+B} (forward). In addition to warping the grid, we average the angular displacement of each grid point over the center of the grid to estimate the change in rotation over the whole patch between frames. Finally, we compute Daisy descriptors for every frame, using the warped grid and the new grid-wise orientation. A graphical illustration of this procedure is depicted in Fig. 3.3.

We denote 'warped' Daisy descriptors with a tilde: $\widetilde{\mathbf{D}}$. We denote the frame they belong to, with respect to the current frame k, with the subindex b: $\widetilde{\mathbf{D}}_{\{k,b\}}$, $b \in \{\pm 1, \ldots, \pm B\}$. The spatiotemporal descriptor $\widehat{\mathbf{D}}$, denoted with a hat, is assembled by concatenating the single-frame Daisy descriptors:

$$\widehat{\mathbf{D}}_{\{k\}}(\mathbf{x}) = \left\{ \mathbf{D}_{\{k,0\}}(\mathbf{x}), \left\{ \widetilde{\mathbf{D}}_{\{k,b\}}(\mathbf{x}), b \in \{\pm 1, \ldots, \pm B\} \right\} \right\} \tag{3.1}$$

Its size is at most $\widehat{S} = T \times S$, where S is the size of a single Daisy descriptor.

While assembling the spatiotemporal descriptor we perform an additional step, as follows. We match the central, unwarped Daisy descriptor $\mathbf{D}_{\{k,0\}}$ with each Daisy computed with the warped grid, $\widetilde{\mathbf{D}}_{\{k,b\}}$, and discard it if the matching score $d(\mathbf{D}_{\{k,0\}}, \widetilde{\mathbf{D}}_{\{k,b\}})$ falls beyond a certain threshold, for every warped frame $b \in \{\pm 1 \ldots \pm B\}$. We compute the distances between single-frame Daisy descriptors following Eq. 2.1. To determine the threshold we typically use a value smaller than the occlusion cost of the regularization process (see section 3.4). We use this procedure to discard bad flow priors, matches where the patch suffers significant transformations such as large distortions or lighting changes, and also as a safeguard against partial occlusions (at this level we have only a monocular sequence and are thus unable to make inferences about occlusions).

Note that we do not need to recompute the convolved orientation maps for each frame involved in the spatiotemporal descriptor, as we can store them in memory. Computing the descriptors is then reduced to sampling the convolved orientation maps at the appropriate grid coordinates on each frame. Additionally, to compute descriptors over different orientations we simply rotate the grid and shift the histograms circularly. We use interpolation over the convolved orientation map values for both the grid coordinates and the gradient orientations.

To compute the distance between two spatiotemporal descriptors $\widehat{\mathbf{D}}_i$ and $\widehat{\mathbf{D}}_j$, we average the distance between valid pairs of (warped) Daisy descriptors:

$$\widehat{d}\left(\widehat{\mathbf{D}}_i, \widehat{\mathbf{D}}_j\right) = \frac{1}{T^{\{i,j\}}} \sum_{b \in \mathcal{T}} \delta^{\{b\}} \cdot d\left(\widetilde{\mathbf{D}}_{\{i,b\}}, \widetilde{\mathbf{D}}_{\{j,b\}}\right) , \tag{3.2}$$

where $\delta^{\{b\}}$ is a set of binary flags that determine valid matching sub-descriptor pairs, i.e. where both sub-descriptors pass the validation process: $\delta^{\{b\}} = \delta_i^{\{b\}} \wedge \delta_j^{\{b\}}$, with $\wedge$ being the logical `and` operator. $T^{\{i,j\}} = \sum_{\forall b \in \mathcal{T}} \delta^{\{b\}}$ is the number of valid sub-descriptor matches after validating the warped frames. $\mathcal{T}$ denotes the group of frames used to build the descriptor, $\mathcal{T} = \{0, \pm 1 \ldots \pm B\}$. For conveniency, here we use $\widetilde{\mathbf{D}}$ for warped and unwarped descriptors alike. Note that in the worst case scenario we will always be able to match the sub-descriptors for the central frame, $\mathcal{T} = \{0\}$, and $T^{\{i,j\}} \in [1, T]$.

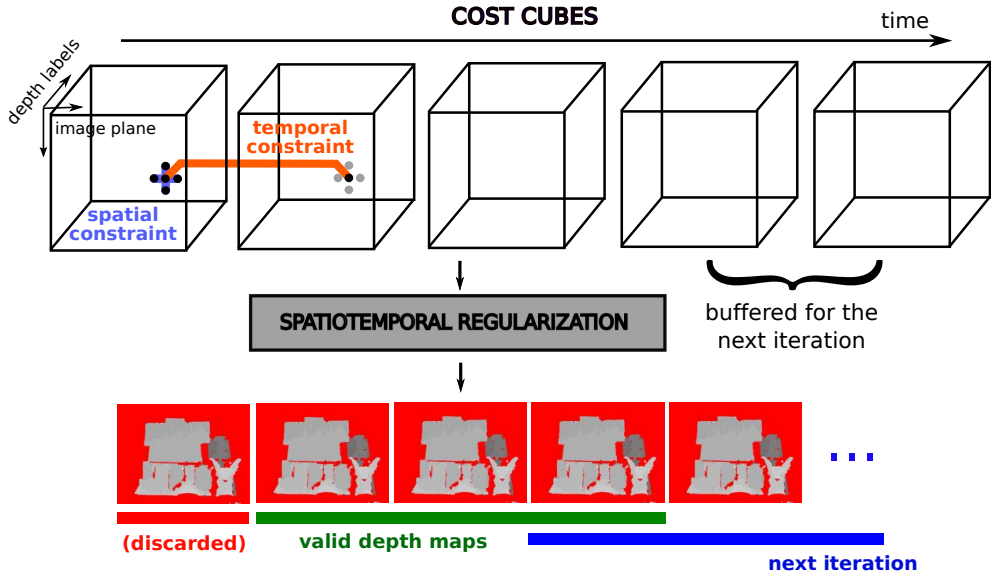


Figure 3.4: To perform spatiotemporal regularization, we apply a graph cuts algorithm over the spatiotemporal hypercube of costs size $W \times H \times L \times M$ (here displayed as a series of cost cubes). The smoothness function connects pixels on a spatial and temporal neighborhood. To avoid singularities on the frames at either end of the buffer, $k = \pm B$, we discard their resulting depth maps and obtain them from the next iteration, as pictured. We do not need to recompute the cost cubes.

3.4 Depth Estimation with spatiotemporal constraints

For stereo reconstruction we use a pair of calibrated monocular cameras. Since Daisy is not rotation-invariant we require the calibration data to compute the descriptors along the epipolar lines, rotating the grid. We do this operation for the central frame, and for frames forwards and backwards we use the flow priors to warp the grid and to estimate the patch rotation between the frames. As explained in Sec. 2.4.1, we discretize 3D space from the point of view of one of the two cameras, compute the depth for every possible match of descriptors constrained to the epipolar geometry, and store the match if it attains the best score for a given depth bin. We thus build a cube of distance values size $W \times H \times L$, where W and H are the width and height of the image and L is the number of layers we use to discretize 3D space. We can think of these values as costs, and then apply efficient global optimization techniques such as graph cuts (Boykov et al., 2001) to enforce piecewise smoothness. We thus obtain a good estimate that balances the pixel-wise fidelity to the data with a smoothness cost that penalizes the use of different labels on neighboring pixels. To deal with occlusions we incorporate an occlusion node in the graph structure with a constant cost. We use the same value, 20% of the maximum cost, for all the experiments.

We can use two strategies for global optimization: enforcing spatial consistency, as described in the previous paragraph, or enforcing both spatial and temporal consistency. To enforce *spatial consistency*, we can match two sets of spatiotemporal descriptors and run the optimization algorithm for a single frame. We use a smoothness function with the Potts model of Eq. 2.12, with a neighborhood of 4 pixels.

To enforce *spatial and temporal consistency*, we can match two sets of multiple spatiotemporal descriptors. To do this we use a hypercube of distances values across time, size $W \times H \times L \times M$, where M is the number of frames used in the optimization process. Note that M does not need to match T, the number of frames used to build the spatiotemporal descriptors. We then use 6 neighboring pixels for the smoothness function, so that each pixel (x, y, k) is linked to its 4 adjacent neighbors over the spatial domain and the two pixels that share its spatial coordinates on two adjacent frames, $(x, y, k-1)$ and $(x, y, k+1)$. This procedure is illustrated in Fig. 3.4. We then run the optimization algorithm over the spatiotemporal volume and obtain M separate depth maps. The value for M is constrained, in practice, by the amount of memory available; we use $M = 5$ for all the experiments presented in Sec. 3.7.

The spatiotemporal regularization strategy produces better, more stable results on dynamic scenes, but introduces one issue. The smoothness function operates over both the spatial and the temporal domain in the same manner, but we have a much larger granularity and a shorter buffer in the temporal domain (M, versus the image size $H \times W$). This results in over-penalizing label discontinuities across time. Smooth gradients on the temporal dimension are often lost in favor of two contiguous frames with the same depth maps. Additionally, the frames at either end of the spatiotemporal volume are linked to a single frame, as opposed to two, and so they are more strongly biased. To solve this we substitute the Potts model in the smoothness function for a truncated linear model:

$$V'(\alpha, \beta) = \texttt{min}\left(\gamma \cdot \frac{|\alpha - \beta|}{U}, \gamma\right) \tag{3.3}$$

where α and β are the depth labels for two connected pixels, γ is the smoothness cost, and U determines the width of the function. Note that for $U = 1$, $V' = V$, i.e. the Potts mode. In our experiments we use $U = L$. This effectively relaxes the spatial smoothness constraint, but the end results are better due to the integration of the temporal consistency constraints. The frames at either end of the spatiotemporal buffer, which are linked to a single neighbor, can still present this behavior. We solve this problem discarding their depth maps, which are recomputed in the following iteration, this time on a 'central' buffer position. We do not need to recompute the cost cubes for the discarded frames: we just store the two cost cubes at the end of the buffer, which become the first two for the buffer corresponding to the next iteration (see Fig. 3.4).

We use the spatiotemporal optimization strategy for most of the experiments on this chapter. We refer to the *spatial* regularization strategy as GC-2D and the *spatiotemporal* regularization strategy as GC-3D.

3.5 Handling occlusions with spatial masks

To handle occlusions we introduce occlusion binary masks over the spatiotemporal descriptors, following the procedure described by Tola et al. (2010) for iterative refinement of depth maps under occlusions. This work proposes using a set of $K+1$ binary masks, with K being the number of grid points on each ring. To enforce spatial coherence the masks are predefined as follows: one mask keeps all grid points enabled, and the remaining K enable the grid points in a 'half-moon' formation and disable the rest, over different orientations. The resulting masks are pictured in Fig. 3.5.

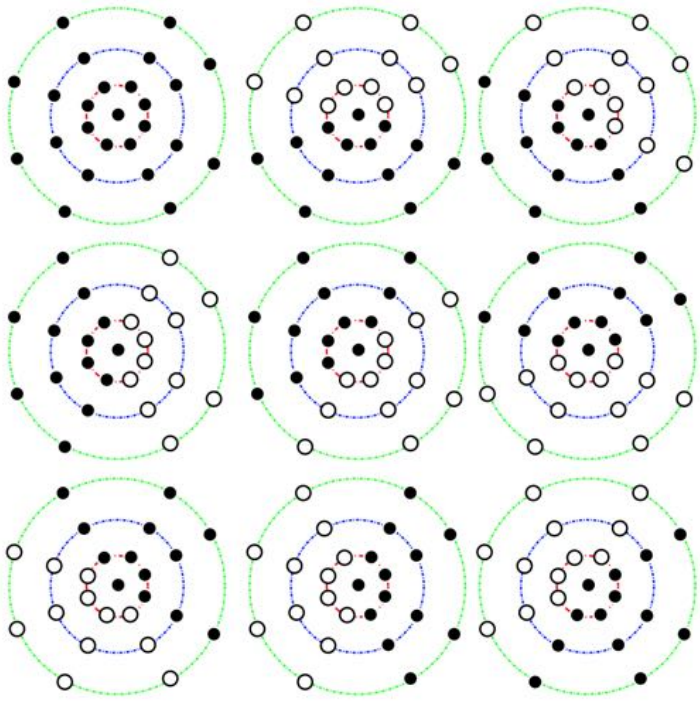


Figure 3.5: The spatial masks proposed in (Tola et al., 2010): black is enabled and white disabled. There is one fully enabled mask—i.e. no masking—and 8 more in a 'half-moon' configuration. The histograms corresponding to a disabled grid point are dropped while computing distances between descriptor pairs.

These masks are designed to be used in conjunction with the stereo algorithm described in the previous section. We run an initial stereo iteration, and use its result, i.e. the first depth map, to compute a score for each mask, for each pixel, penalizing differing depth values on enabled areas. We follow the procedure outlined in their paper to score the masks, which does not describe all the implementation details. For reference: we use the depth map values within a radius of 20 pixels of the feature point, the default grid radius use by Daisy being $R = 15$ pixels. For the 'half-moon' masks, we consider only the pixels which lie closer to an enabled grid point than to a disabled grid point.

We would then run a new iteration of the stereo algorithm, using for each pixel the mask with the highest score computed from the previous depth map. The histograms corresponding to disabled grid points are dropped from the similarity function. Given two Daisy descriptors $\mathbf{D}_i$ and $\mathbf{D}_j$.

$$d'(\mathbf{D}_i, \mathbf{D}_j) = \frac{1}{\sum_{p=1}^{P} \mathcal{M}^{[p]}} \sum_{p=1}^{P} \left\| \mathbf{D}_i^{[p]} - \mathbf{D}_j^{[p]} \right\|_2, \tag{3.4}$$

where $\mathcal{M}^{[p]} \in \{0, 1\}$ encodes the mask value at grid point $p = 1, \ldots, P$. We thus keep the advantages of matching large, discriminative patches, while preserving the boundaries around occlusions. Two or three iterations are often enough to refine the areas around occlusions. An actual run of this iterative process is illustrated in Fig. 3.6.

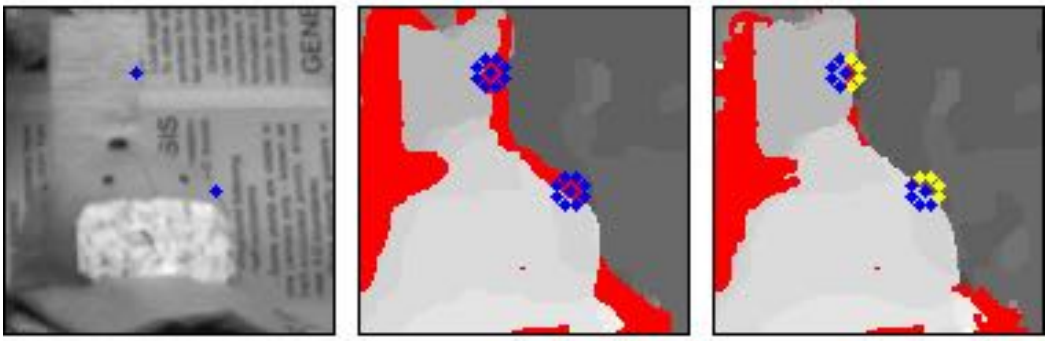


Figure 3.6: Using binary masks to refine the stereo reconstruction of scenes with occlusions, following (Tola et al., 2010). Left to right: reference image, the depth map after the first iteration, and a posterior refinement using the occlusion masks. Enabled grid points are colored blue, and disabled grid points are colored yellow. We show a simplified grid of 9 points with a single ring ($N = 1$), for simplicity. The images belong to the stereo video dataset of (Sizintsev and Wildes, 2009), extracted with a classical narrow-baseline stereo camera. Occlusion masks become more relevant in wide-baseline setups, for which unfortunately datasets with motion are scarce.

A limitation of this procedure is that it relies on depth estimates to choose the occlusion masks, so that early reconstruction errors can be hard to recover from in successive iterations. In chapter 4 we present a methodology based on segmentation cues to reason about occlusions, which we use to build soft masks, as opposed to binary, in a single pass.

As we discussed in the introduction to this chapter, the aim of this work is to devise a spatiotemporal approach to stereo that can be applied to wide baselines without prior knowledge of their geometry, and use it to tackle highly ambiguous correspondence problems (Fig. 3.2). Our main objective is to build spatiotemporal features *without* relying on the spatiotemporal volume, a principle whose applicability is limited to narrow baselines.

As such, we do not focus on dealing with occlusions. We must, however, consider them; for this, we rely the work of Tola et al. (2010). We integrate binary occlusion masks to our spatiotemporal descriptors, in the following manner. The masks can be scored in the same way, as they rely on estimated depth maps. To extend them to our spatiotemporal descriptors, we simply apply them to each single-frame (warped) Daisy, separately. For simplicity, we use a GC-2D optimization on the first iterations and a GC-3D optimization on the last iteration. The masks are recomputed at each step. In Sec. 3.11 we evaluate its performance. These experiments are performed over a narrow-baseline dataset, as to our knowledge there are no wide-baseline datasets incorporating both motion and occlusions, and ground truth data. The rest of the experiments do not use occlusion masks (we do, however, include an occlusion node in the graph cuts optimizer).

3.6 Computational cost and implementation details

For reference, a round of spatiotemporal stereo for two 480×360 images takes about 24 minutes on a 2 Ghz dual-core computer (with no parallelization), up from about 11 minutes for Daisy. This does not include the computation of the flow priors, which can be done very fast; the implementation we use requires approximately 1 minute per image. For optical flow we use an off-the-shelf implementation in Matlab. The rest of the code is C++. The bottleneck is matching the descriptors, as even after constraining the matches to the epipolar geometry, we need to compute approximately $W^2 \times H$ distances between descriptor pairs.

3.7 Experimental evaluation

Wide-baseline stereo datasets are scarce. To the best of our knowledge there are none for dynamic scenes, with available ground truth data. Therefore, in order to evaluate the accuracy of our approach we synthesized a series of stereo image sequences with ground truth depth.

To do so we generated a flat, triangulated 3D mesh, textured it with a repetitive pattern, and applied transformations over it to create synthetic images and ground truth depth maps. We generated views from two different camera poses, with narrow and wide baselines. The images contain the patterned mesh in the center, surrounded by black background. We used two different source patterns, one of a highly textured, natural image ('gravel'), and another one with a sparse, symmetric arrangement ('flowers').

We built three synthetic datasets:

1. A narrow-baseline dataset with mesh deformations, for which we consider two modes: one being the magnitude of the sinusoidal oscillations (deformations on a 'global' scale, such as those depicted in Fig. 3.7), and the other random noise over the mesh coordinates (deformations on a 'local' scale). We generate 5×5 combinations with 5 values for the magnitude of the 'global' deformations and 5 for the magnitude of the 'local' deformations.
2. A wide-baseline dataset, in which the angle between the optical axes of the two cameras increases at increments of $\pi/16$ radians (for a largest angle of $\pi/4$).
3. A dataset with a narrow stereo baseline with additive, gaussian image noise, independent for each camera.

Fig. 3.7 shows some samples from the wide-baseline dataset, as a reference.

We use the first dataset to gain an intuitive understanding of what kind of dynamic information our descriptor can take advantage of. The second and third synthetic datasets are used to benchmark our approach against state-of-the-art feature descriptors. Finally, we show qualitative results on two highly ambiguous real sequences without ground truth data. Additionally, we demonstrate the application of masks on a real dataset with occlusions (Sec. 3.11).

For a baseline, we use SIFT (from VLFEAT), Daisy, and the spatiotemporal Stequel descriptor. Each of our synthetic datasets contains 30 images of size 480×480 pixels, from each camera pose ('left' and 'right'). The experiments are performed over the 15 middle frames, and the spatiotemporal descriptors use the additional frames as required. We refer to our descriptor as Daisy-3D, and to the spatial Daisy descriptor

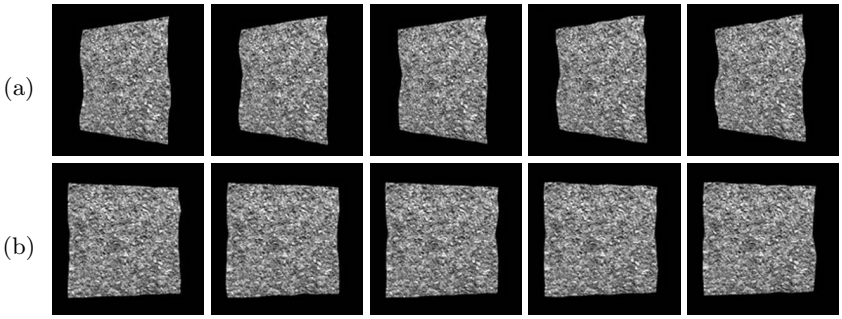


Figure 3.7: Five sample image pairs from the wide-baseline dataset (we show images 1, 4, 7, 10 and 13), from (a) the rightmost viewpoint and (b) the leftmost viewpoint.

as Daisy-2D. If necessary, we use Daisy-3D-2D and Daisy-3D-3D to distinguish between the spatial and spatiotemporal regularization schemes introduced in Sec. 3.4.

For Daisy-2D and SIFT, we apply a spatial regularization scheme. To provide a fair comparison with SIFT we compute the descriptors densely, at a constant scale, over a patch rotated along the epipolar lines, as we do for Daisy-2D and -3D—note that this negates the computational advantages of DSIFT. For the stequel algorithm we rectify the images with (Fusiello et al., 2000) and use the self-contained stequel binary code provided by Sizintsev and Wildes (2012), which contains its own regularization scheme and produces 3D depth maps directly. In neither case do we compute nor match descriptors for background pixels.

The experiments in Secs. 3.7.1-3.7.5 adhere to the following conventions. If we know the depth range of the scene beforehand, we can further constrain the possible matches to a segment of the epipolar line. Instead we choose a wide scene range, as the actual range of these synthetic scenes is very narrow and the correspondence problem would be very simple under those assumptions. We plot the results in terms of the error in number of depth layers, for ease of understanding. Whenever we show depth maps (we do so only for the experiments of Sec. 3.7.5, which do not have ground truth data) we prune the depth values to the actual scene range, to plot the results in a manner that is easy to interpret (this means that white and black pixels in the depth map plots are outside said range).

3.7.1 Parameter selection

The synthetic data of set (1) is used only for a preliminary study. As expected, noise displacements over the mesh coordinates of the synthetic sequence results in richer textures, which can be well discriminated by spatial descriptors and even in larger extend by the spatiotemporal descriptors. Affine and non-rigid transformations can also be discriminated, but to a lesser degree.

The optimal number of frames used to build the spatiotemporal descriptor, T, is often small, between 5 and 9. Larger values require a smaller footprint for the spatial descriptor to have them fit into memory, which reduces overall performance. For the Daisy parameters, it is consistently better to reduce the number of histograms (grid points) rather than bins (oriented gradients), as the latter prove more discriminant.

For Daisy-2D we use the configuration suggested in (Tola et al., 2010), with a grid radius of $R = 15$ pixels, $N = 3$ rings and $K = 8$ points per ring, and $H = 8$ gradient

orientations, resulting in a descriptor size $S = 200$. For SIFT we choose a grid size commensurate with that of Daisy. For Daisy-3D we use the same values, except for $N = 2$ rings. A single-frame descriptor is of size $S = 136$, which is the best compromise. We use $T = 7$ frames for the spatiotemporal features, yielding a final descriptor of size (at most) 136×7.

3.7.2 Wide baseline experiments

For this experiment we compare our descriptor with Daisy-2D, SIFT and Stequel. We match video pairs, from the rightmost camera position to the other four, each rotated $\pi/16$ while centered on to the mesh, for a maximum baseline of $\pi/4$. We reconstruct the depth maps from the rightmost viewpoint in every case.

Figs. 3.8 and 3.9 show the results for the 'gravel' and 'flower' patterns, respectively. The accuracy (see Figs. 3.8-(a) and 3.9-(a)) is defined as the percentage of pixels with an error below a threshold of 10% (left) and 5% (right) of the scene range, averaged over all the images in the sequence.

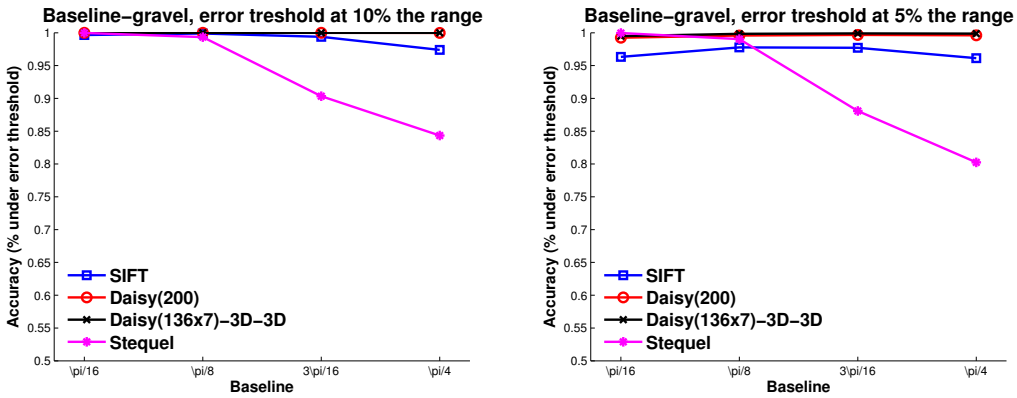
On the configuration with the narrowest baseline Stequel performs better than any of the other algorithms, due to sub-pixel accuracy from a Lucas-Kanade-like refinement—see (Sizintsev and Wildes, 2009) for details. The rest of the algorithms have slight quantization errors, due the discretization process of our regularization framework. Stequel still performs well at $\pi/8$, but shows systematic errors beyond that point—which are to be expected as it was not designed for wide baselines. The rest of the algorithms perform better in comparison, with the Daisy-based descriptors edging out SIFT. Wide baselines prove discriminative (without occlusions), particularly for the 'flowers' pattern. Note that we do not apply masks to this experiment.

Figs. 3.8-(b) and 3.9-(b) show qualitative results for one specific frame of each pattern. Colors encode the reconstruction error, expressed in terms of the layers used by the regularization algorithm: the very light yellow seen in most images is due to quantization errors (1 layer). Pixels in blue are marked as occluded, i.e. they are false negatives. The background is drawn in black, and background pixels are not included in the reconstruction.

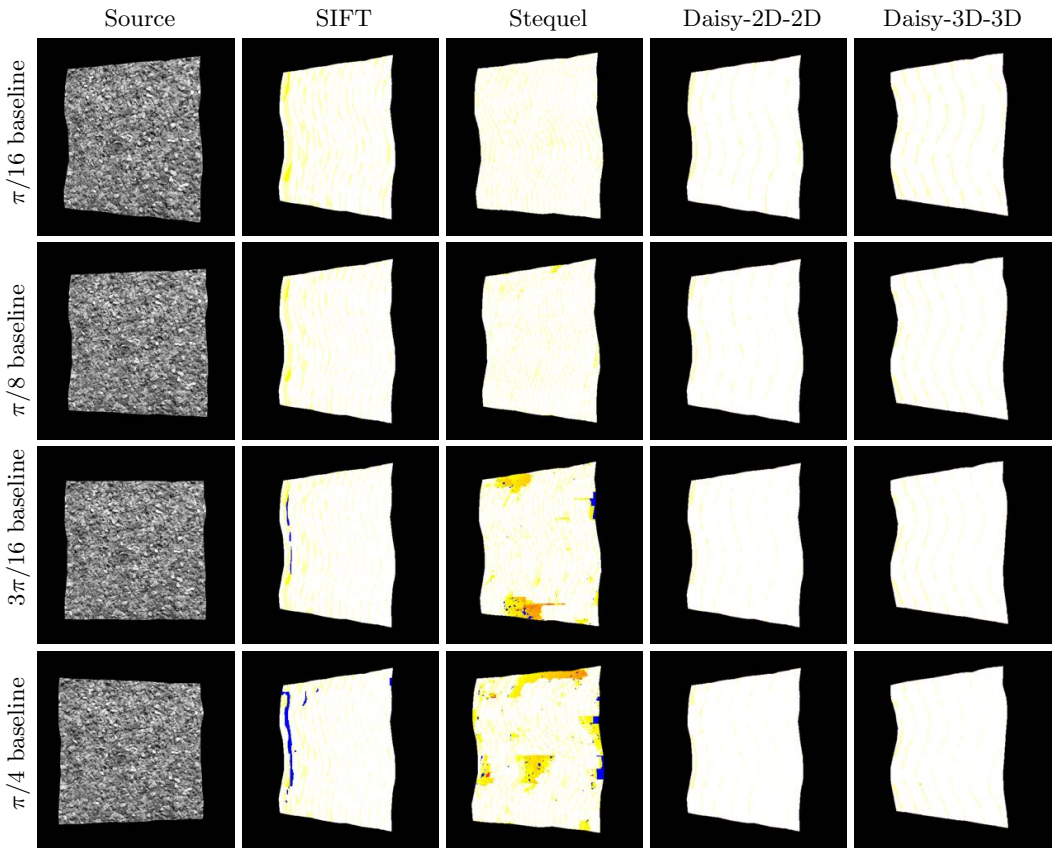
3.7.3 Image noise experiments

For these experiments we have 11 narrow-baseline video sequences with the with 'gravel pattern' and different levels of image noise in the grayscale values, from $\sigma = 0$ to 0.5, with increments of $\Delta\sigma = 0.05$. The results are shown in Fig. 3.10. For this experiment we consider our spatiotemporal descriptor with both spatial regularization (Daisy-3D-2D) and spatiotemporal regularization (Daisy-3D-3D), to ascertain that the descriptor itself is robust to image noise, thanks to the global optimization imposed on the flow priors, which while being noisy, remain able to capture the motion within the image.

Despite the expected distortion over the flow estimates, our descriptor significantly outperforms the others. Enforcing spatiotemporal consistency reduces the number of gross reconstruction errors but makes the estimates a bit less accurate: hence Fig. 3.10-(a) shows Daisy-3D-3D outperforming Daisy-3D-2D at a higher error threshold (left), which evens out at lower thresholds (right). SIFT performs slightly better than Daisy, and the Stequel descriptor breaks down with just a bit of noise, which affects each spatiotemporal volume independently.



(a) Quantitative results (averaged over all frames).



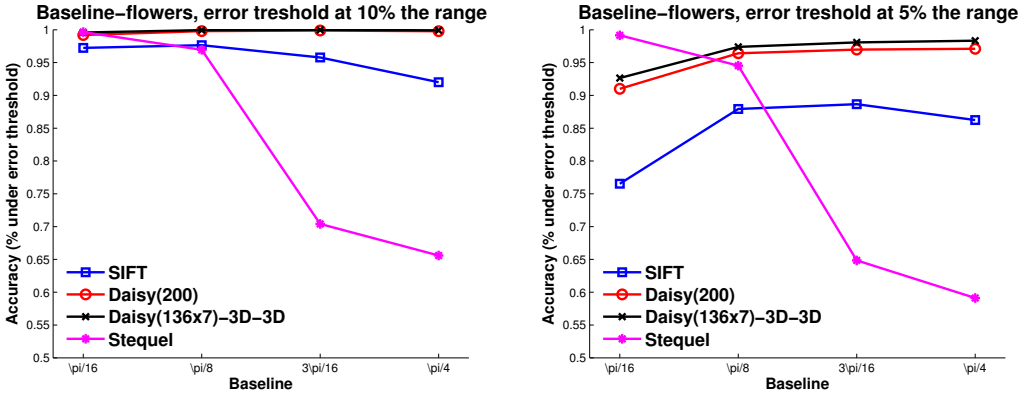
(b) Qualitative results (single frame).

Figure 3.8: Synthetic results, from narrow to wide-baseline, for the 'gravel' sequence.

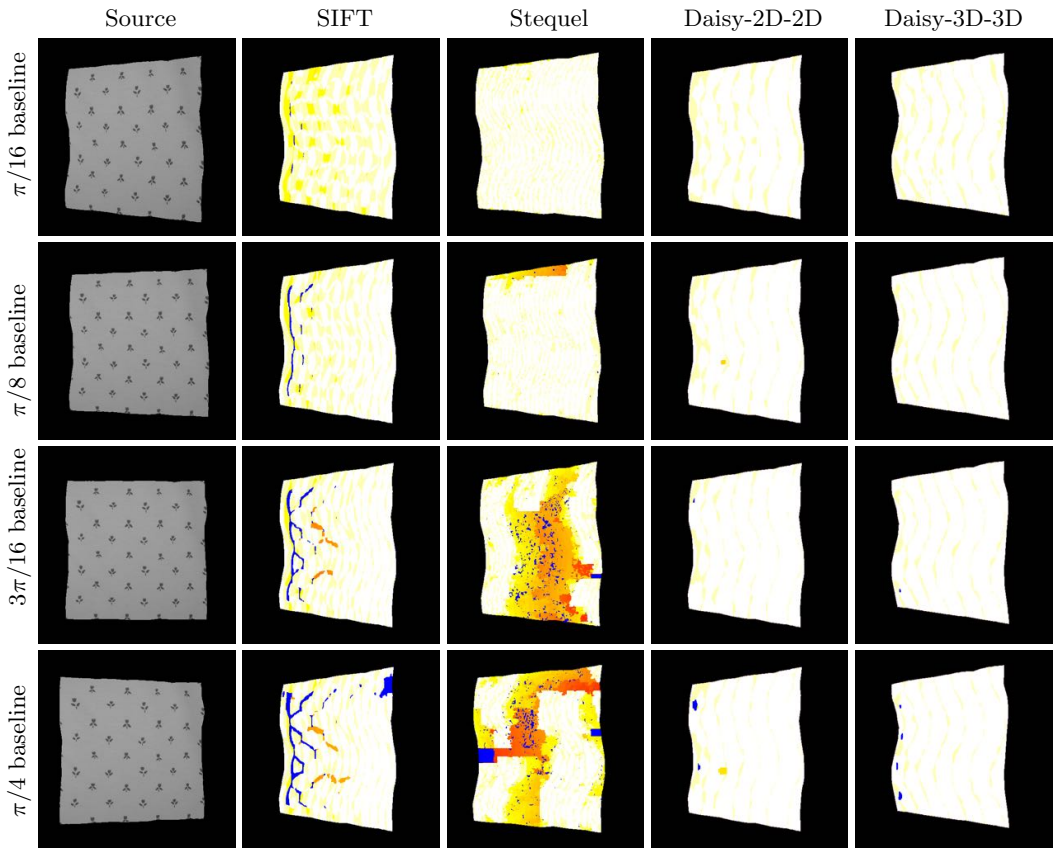
3.7.4 Experiments with occlusion masks

We have proven that our approach can deal with wide baselines (Sec. 3.7.2). In this section we demonstrate that we can extend the iterative masks of (Tola et al., 2010) to our spatiotemporal descriptor, to deal with occlusions, as discussed in Sec. 3.5.

To our knowledge, there are no video datasets with ground truth data for wide



(a) Quantitative results (averaged over all frames).

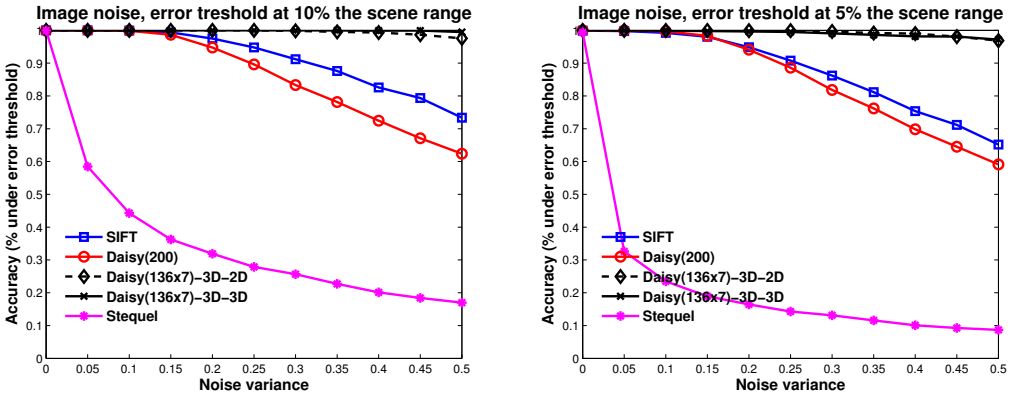


(b) Qualitative results (single frame).

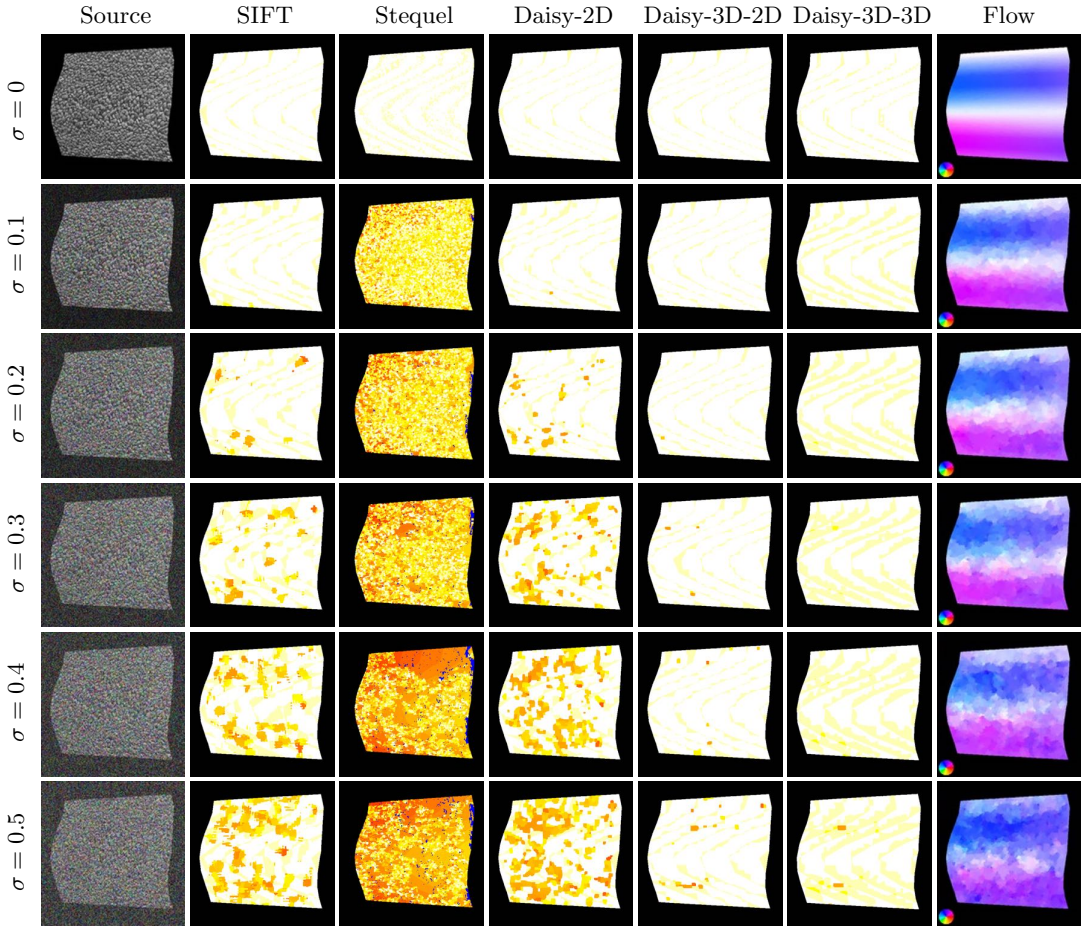
Figure 3.9: Synthetic results, from narrow to wide-baseline, for the 'flowers' sequence.

baseline stereo—which are arguably hard to capture. Our synthetic datasets do not consider occlusions. We use instead the narrow-baseline stereo dataset introduced by Sizintsev and Wildes (2009) to evaluate the Stequel descriptor. It contains a very richly textured scene, so we use a very small descriptor, with only 9 grid points and 4 gradient orientations (and hence of size 36).

Daisy already performs very well on these settings, and we do not expect significant



(a) Quantitative results (averaged over all frames).



(b) Qualitative results (single frame). Optical flow is always computed over the whole image, but we only use the foreground pixels—we paint the background black for clarity. The circle on the bottom left corner of the flow images shows the color coding used to display the flow velocities.

Figure 3.10: Results for the image noise experiments.

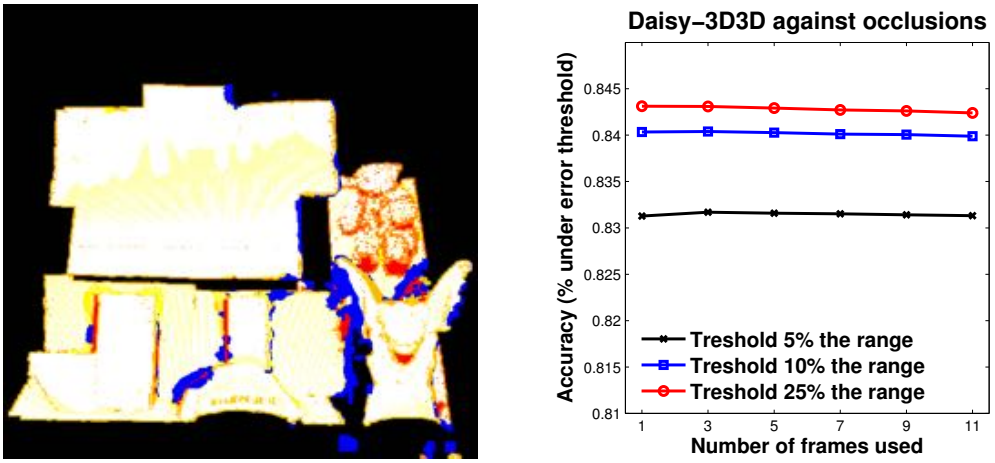


Figure 3.11: Benchmarking the spatiotemporal descriptor against occlusions. The image on the left displays the depth error with the latent occlusion estimation scheme described in Sec. 3.5 after four iterations, using Daisy-3D-3D built with $T = 11$ frames. The colors encode the error in terms of the quantized depth layers: white indicates a correct estimage, light yellow indicates an error of one layer (which is often due to quantization errors), and subsequent error values are coded by temperature. Red indicates occlusions, and the background is colored black. Blue indicates foreground pixels incorrectly labeled as occlusions, i.e. a false negative. The plot on the right shows accuracy values for three different error thresholds, for different values of T, the number of frames used to compute the spatiotemporal descriptor. We see that the spatiotemporal descriptor is not impaired in scenes with occlusions.

improvements with our spatiotemporal features. Our goal it to demonstrate that we are able to propagate the descriptors in time around occlusion areas, without impairing the reconstruction process—thus benefitting from the advantages shown in the previous section as well as Daisy's latent occlusion estimation. Fig. 3.11 shows that the reconstruction accuracy does not change significantly as the number of frames used to build the spatiotemporal descriptor increases. This is to be expected, as the scene contains few ambiguities and a low dynamic content. On the other hand, we observe that using a high number of frames to build the spatiotemporal descriptors does not result in singularities around occlusion areas.

3.7.5 Experiments with real sequences

In addition to our synthetic sequences, we apply the algorithm to two highly ambiguous real sequences. We do so qualitatively, as we cannot compute ground truth data at high rates through traditional means: structured light would alter the scene, most 3D laser range finders are not fast enough, and available time-of-flight cameras or motion sensor devices such as the Kinect are not accurate enough to capture small variations in depth. We require high framerates to extract an accurate flow, which would otherwise fail on sequences of this complexity (see the images of Fig. 3.2 for a reference).

We use two Point Grey Grasshoper cameras to record images at a framerate of approximately 100 Hz. For this experiment, we compute the optical flow priors at this framerate, but we run the stereo algorithm using one frame in three. We do

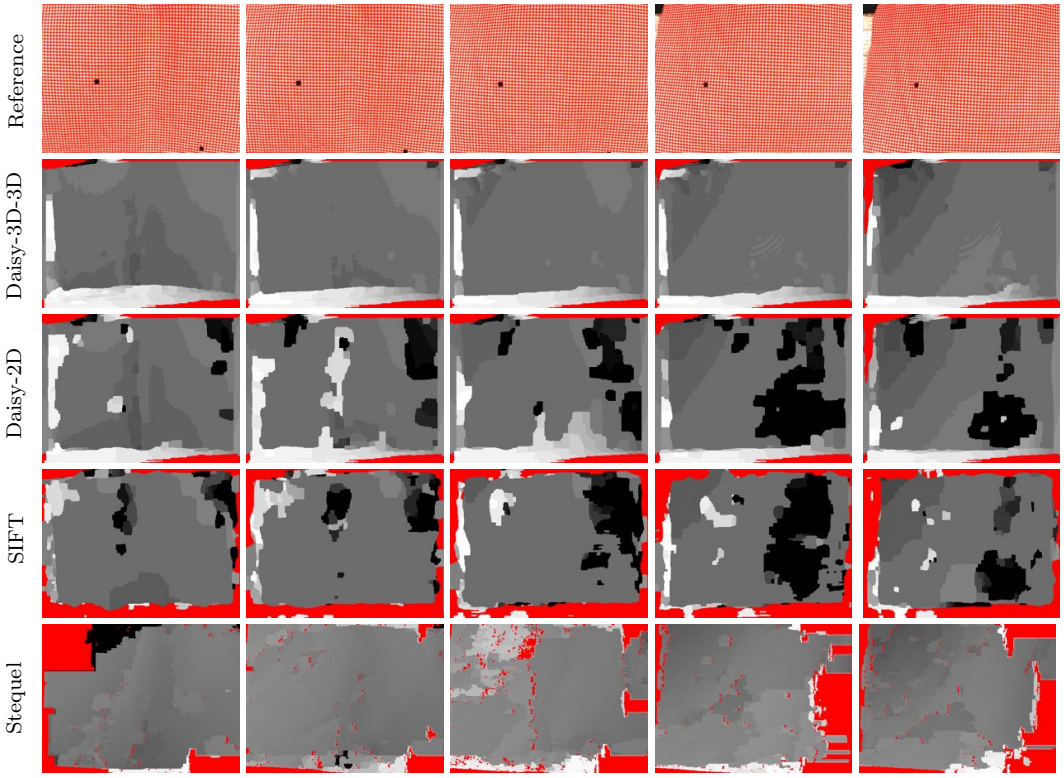


Figure 3.12: Qualitative reconstruction results for five consecutive frames of the first of two very challenging stereo sequences.

so because even though we have demonstrated that our spatiotemporal descriptor is very robust against perturbations of the flow data (see Sec. 3.7.3), video sequences of this nature can occasionally suffer from the aperture problem—we believe that, in practice, this constraint can be relaxed. Even without ground truth data, simple visual inspection shows that the depth estimate is generally much more stable in time and follows the expected motion. Figs. 3.12 and 3.13 show the respective depth estimates for five consecutive frames, for each algorithm. The algorithm we propose, Daisy 3D-3D, consistently outperforms the other methods.

3.8 Summary and future work

We have presented an approach to stereo reconstruction that can handle very challenging situations, including highly repetitive patterns, noisy images, occlusions, non-rigid deformations, and wide baselines. We have shown that these artifacts can be successfully addressed augmenting feature descriptors with temporal information.

The use of temporal information for 3D reconstruction is not by itself novel. Some approaches rely on active techniques such as structured light patterns, which cannot be applied in many realistic scenarios. Others have introduced spatiotemporal features based on cubic-shaped volumes of data in space and time. These techniques are limited to monocular applications, or to stereo with a known geometry, and narrow baselines. While a slight difference in viewpoint can indeed be very informative (Sizintsev and

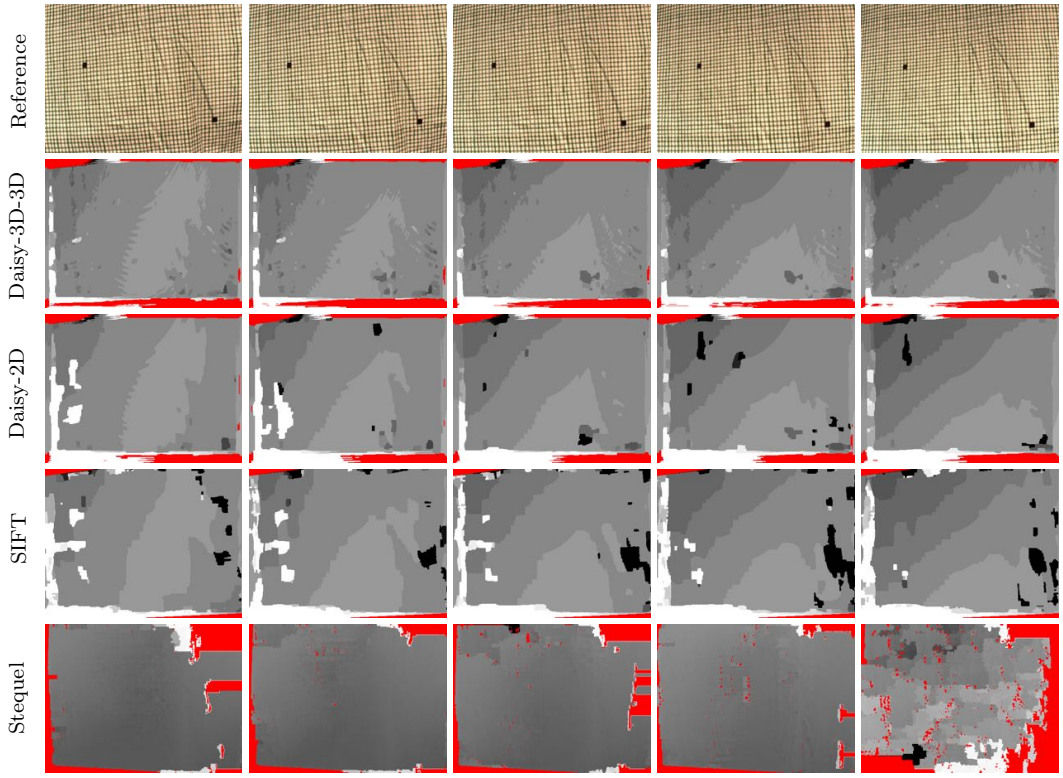


Figure 3.13: Qualitative reconstruction results for five consecutive frames of the second of two very challenging stereo sequences.

Wildes, 2009), it is unclear how to apply this strategy to the general case of wide stereo baselines.

Our stereo algorithm combines two components: a *spatiotemporal, dense descriptor*, and a *spatiotemporal regularization algorithm* for stereo.

Our *descriptor* also relies on spatiotemporal features, but rather than operating directly on the spatiotemporal volume, we use optical flow priors to 'build' its viewpoint-invariant equivalent. This allows us to stay fixated on the feature point—effectively tracking it—while capturing the changes in its neighborhood. This enables us to exploit the non-linear dynamics that individual pixels undergo over time to handle complex ambiguities.

With regards to the *stereo algorithm*, we consider the state-of-the-art advances shown by the Daisy paper, and build on them with a scheme to enforce temporal consistency.

We show that our approach is very robust with regards to noise in the optical flow estimates, and to other artifacts such as mislabeling due to aperture noise; the spatiotemporal regularization can cope with these errors.

One of the main limitations of our spatiotemporal descriptor is its high dimensionality. Our representation is arguably very redundant, as the pixel data for consecutive frames typically does not vary too much. We can address this problem with dimensionality reduction or quantization techniques, such as those described in Sec. 2.2.6, and which we intend to investigate in the future. Reducing the size of the descriptor would also alleviate its computational cost for dense stereo, where the bottleneck lies

in computing similarities between descriptor pairs.

This brings another question to mind: is our current approach to build the spatiotemporal descriptors optimal? In future work we will explore alternative strategies, perhaps involving the flow cues by themselves. We could for instance try to encode their statistics as use them as features for matching. Additionally, we know we cannot match the flow cues extracted from different stereo channels on wide-baseline rigs directly, for the same reason we cannot rely on the spatiotemporal volume, but this constraint may be relaxed for narrower baselines. We could also explore if the flow estimates can allow us to determine and track occlusions through time, as planes at different depths (with regards to the camera) will move with different speeds.

In addition, we believe that monocular approaches to non-rigid 3D reconstruction and action recognition may benefit from our methodology, specially if the videos contain significant viewpoint changes with regards to the subject or the scene. So far we have used our descriptor densely, which is arguably the best way to attack wide-baseline stereo, but different applications may benefit from a sparse approach. Note that while in this work we rely on calibration data to relax the complexity of the matching problem, we do not use it to build the spatiotemporal descriptors themselves. We could thus apply them on sparse, stable points found with any standard feature detector, without any modifications.

Lastly, we would like to investigate the application of our approach to scene flow estimation. We currently do not check the flow estimates for consistency; note that we perform alternative tests which are also effective: we do check the warped descriptors for temporal consistency and discard them when they match poorly, and we apply a global regularization to enforce spatiotemporal consistency. Narrower baselines and/or the use of calibration data would at the very least allow us to use simple heuristics to discard pairs of flow cues that do not make sense, and explore more complex strategies later. Given our concerns about computational costs, we could decouple stereo and motion as in (Wedel et al., 2008), while enforcing joint consistency.

Chapter 4

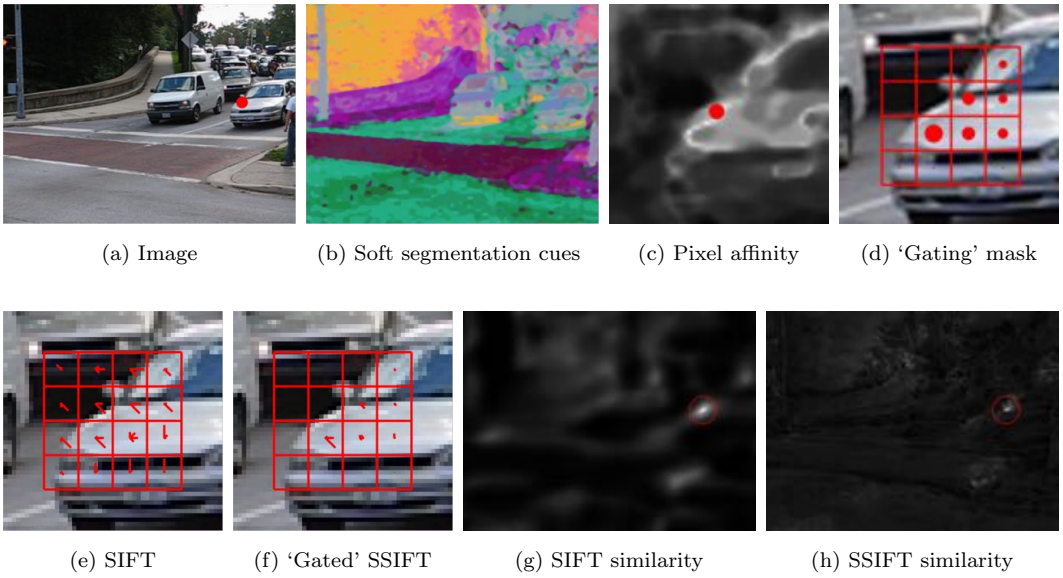
Dense segmentation-aware descriptors

Dense descriptors are becoming increasingly popular in a host of tasks, such as dense image correspondence, bag-of-words image classification, and label transfer. However the extraction of descriptors on generic image points, rather than select geometric features, e.g. blobs, requires rethinking how to achieve invariance to nuisance parameters. In this chapter we pursue invariance to background variability by introducing segmentation information within dense feature construction. Our goal is to construct feature descriptors that are contained within a single surface/object ('region' from now on). In this way changes in the background, e.g. due to layered motion, will not affect the description of a point in the interior of a region. Similarly, when a region is occluded by another region in front of it, even though we cannot recover its missing information, we can at least ignore irrelevant occluders. This chapter is based on our 2013 CVPR paper (Trulls et al., 2013).

4.1 Introduction

The core idea behind our method is to use segmentation cues to downplay features coming from image areas that are unlikely to belong to the same region as the center of the descriptor. Achieving this goal can benefit SIFT as well as any other descriptor, but its merit is most pertinent to SID. In particular, image scaling does not necessarily result in a cyclic permutation of the SID elements: the finest- and coarsest-level entries can change: see Fig. 2.4. As such the (circular) shifting relationship required to obtain the DTFT pair in Eq. 2.7 does not strictly hold. To remedy this issue SID typically uses finely-grained sampling over large image patches, so that the percentage of points where this change happens eventually becomes negligible; this however limits its applicability, since background structures and occlusions can easily creep into its construction. Overcoming this limitation was the original motivation for this work.

We also found inspiration in the iterative approach to latent occlusion detection by Tola et al. (2010). Their method, which we applied in the previous chapter, demon-



(a) Image (b) Soft segmentation cues (c) Pixel affinity (d) 'Gating' mask

(e) SIFT (f) 'Gated' SSIFT (g) SIFT similarity (h) SSIFT similarity

Figure 4.1: We exploit segmentation data to construct feature descriptors that are robust to background motion and/or occlusions. **(a)** Input image; we want to compute a descriptor for the pixel represented by a red dot. **(b)** Three soft segmentation cues represented in RGB space. Pixels with similar segmentation embeddings are likely to belong to the same region. **(c)** Affinity between the pixel represented by the red dot and its neighbors, defined as the euclidean distance over embedded space. **(d)** A 'gating' mask that encodes the reliability, i.e. region similarity, of the locations used to compute the SIFT descriptor **(e)**. We can thus construct a 'pure' descriptor SSIFT, **(f)** 'gating' the descriptor features with the mask. **(g-h)** show the distance between (e-f) and dense SIFT/SSIFT descriptors over the whole image. Note that the SSIFT similarity function peaks more sharply around the pixel (marked with a circle), indicating its higher distinctiveness. Additionally, removing the background response allows us to match the descriptor for this point with its counterparts across future frames.

strated that we can increase the performance of recognition systems through the combination of features computed over large, discriminative image patches, and occlusion reasoning. In contrast to their work, the technique we present in this chapter produces masks based on segmentation cues in a single step; is soft rather than binary; and is applicable to problems other than stereo.

The basic idea behind our method is illustrated in Fig. 4.1. Our main contribution is a new approach to suppress background information during descriptor construction. For this, we use soft segmentation masks to compute the affinity of a point with its neighbors (Fig. 4.1-(c)), and shun the information coming from regions which are likely to belong to other objects (Fig. 4.1-(f)).

Our treatment is applicable to any image point, i.e. dense, and its computational overhead, excluding the extraction of the segmentation cues, is in the order of a few seconds. We extract the soft segmentations once, *before* descriptor construction. We try three different segmentation cues: (a) 'Eigen' cues, from the the *Pb* eigenvectors of Maire et al. (2008); (b) 'SoftMask' cues, from the *Gb* soft segmentations of Leordeanu

et al. (2012); and (c) 'Edge' boundary cues, from the structured forests boundaries of Dollár and Zitnick (2013). Their computational time varies from about a minute per image (Pb) to multiple frames per second (structured forests).

We show how to integrate this idea with dense SIFT, and dense SID. With the latter we deliver descriptors that are: (a) densely computable; (b) scale- and/or rotation-invariant by design; and (c) robust to background variability. We explore the merit of our technique in conjunction with large displacement motion estimation and wide-baseline stereo, and demonstrate that exploiting segmentation information yields clear improvements.

These are the key strengths of our approach:

Generality. We apply it to *two* different descriptors (SIFT and SID), with *three* different segmentation cues ('Eigen', 'SoftMask', 'Edge'), for *two* different applications (motion and stereo). We demonstrate increased performance with respect to state-of-the art feature descriptors: dense SIFT, dense SID, and the Scale-less SIFT. Most importantly, we demonstrate that the introduction of segmentation results in systematically better results over their segmentation-agnostic counterparts.

Simplicity. It is application-independent, and no training is necessary. It takes features as input, and yields new features in turn, with the same format. As such it can be plugged into any application that relies on feature descriptors, with minimal fuss.

Small overhead. The soft segmentation masks can be computed and applied efficiently, in the order of seconds, for *dense* descriptors.

No tuning. Our algorithm has a *single parameter*, which can be used to adjust the 'hardness' of the masks. We fix it once and use it throughout all the experiments, even across different applications.

We next review related work. In Sec. 4.3 we present our method, and discuss the type of segmentation cues most useful for it. In Sec. 4.4 we show how to build dense soft segmentation masks and, in turn, segmentation-aware descriptors. Lastly, we benchmark our descriptors on standard datasets in large displacement motion estimation and wide-baseline stereo. We demonstrate that the introduction of segmentation cues yields systematic improvements.

4.2 Related work

After the seminal works of SIFT (Lowe, 2004) and Shape Contexts (Belongie and Malik, 2002), large strides have been made in improving performance (Berg and Malik, 2001; Mikolajczyk et al., 2005; Winder et al., 2009; Simonyan et al., 2012, 2014), efficiency (Bay et al., 2008; Özuysal et al., 2010; Calonder et al., 2010; Rublee et al., 2011), and decreasing memory requirements (Ke and Sukthankar, 2004; Brown et al., 2011; Strecha et al., 2012).

A complementary research direction that started with dense SIFT is to extract dense image descriptors. This is motived by experimental evidence that dense sampling of descriptors yields better performance in classification systems based on bag-of-words or spatial pyramids (Nowak et al., 2006; Bosch et al., 2006; Lazebnik et al., 2006; Chatfield et al., 2011), but also from applications such as dense stereo matching which require dense features.

A problem that emerges when computing dense descriptors is invariance. Unlike interest points, which allow for some estimation of local scale and orientation, estimating scale on arbitrary image locations is not obvious. Some recent works have addressed the treatment of scale- and/or rotation-invariance, including SID (Kokkinos and Yuille, 2008) and SLS (Hassner et al., 2012), introduced in chapter 2. Schmidt and Roth (2012) introduce a novel technique to obtain rotation-invariant features by learning image models with built-in invariance to linear transformations, such as rotations. In a paper published after (Trulls et al., 2013), Yang et al. (2014) leverage Daisy descriptors at multiple scales and rotations within the PatchMatch framework (Barnes et al., 2010). In trading off accuracy for speed, exchanging a Markov Random Field-based formulation (e.g. SIFT-flow) with efficient random searches that do not strictly enforce spatial coherence (PatchMatch), it becomes possible to perform feature matching in a high-dimensional label space. Scale and rotation are not the only invariance concerns addressed with local descriptors: DaLI (Moreno-Noguer, 2011) builds on kernels based on heat diffusion theory to extract features resilient to non-rigid image transformations and illumination changes.

In this chapter we push this line of works a step further, to also deal with background variability. Regarding occlusions, there is little work done at the descriptor level. Ott and Everingham (2009) exploit an implicit color segmentation of objects into foreground and background to augment HOG features, and boost the performance of a sliding-window detector. The Daisy paper (Tola et al., 2010) demonstrated clear performance improvements in multi-view stereo from treating occlusion as a latent variable and enforcing spatial consistency with a graph cuts regularization algorithm. Their method applies a predefined set of binary masks over the descriptors, disabling image measurements that are likely to come from occluded areas. To do so they rely on precomputed depth maps: the masks are applied iteratively, interleaved with successive rounds of stereo matching, yielding increasingly refined depth estimates. A possible criticism of this method is that errors in the first stages may not be recoverable later on. Additionally, the masks are made into very regular structures ('half-disks') to enforce spatial consistency: their shape is predetermined, which limits their adaptability.

That being said, our work is largely inspired by the performance improvements demonstrated in (Tola et al., 2010), which show that separating foreground and background results in a significant boost in performance. A marked difference with their approach is that ours is also applicable to the general case where a single image of the scene is available. Furthermore, we do not constrain the masks to be of a pre-determined shape, and show how this technique can be combined with other SIFT-type descriptors, particularly SID.

4.3 Segmentation cues

Our goal is to construct feature descriptors that are not only local, but also contained within a single region. In this way changes in the background, e.g. due to layered motion, will not affect the description of a point in the interior of a region. Similarly, when a region is occluded by another one which is in front of it, even though we cannot recover its missing information, we can at least ignore irrelevant occluders.

This problem is connected with segmentation, where one wants to extract a partition of the image into homogeneous regions. If we were able to use information only from

the region containing a point, we could make a descriptor invariant to background changes: as shown in the first column of Fig. 4.2, given the support of the region containing a pixel, we can identify the descriptor elements that come from different regions and set them to zero. However, and despite constant progress in this area, with new techniques appearing every year (Dollár and Zitnick, 2013; Humayun et al., 2014; Arbeláez et al., 2014), it is understood that the problem of image segmentation is still far from being solved. Therefore, we turn to algorithms that do not strongly commit to a single segmentation, but rather determine the affinity of a pixel to its neighbors in a soft manner. This soft affinity information is then incorporated into descriptor construction, in the form of a 'gating' signal.

We already introduced the segmentation cues we build on in Sec. 2.3.2. To recap: we explore three different means to extract soft segmentations:

1. Firstly, we turn to the work by Maire et al. (2008) on the globalized probability of boundary detector (*Pb*), described in Sec. 2.3.2.1. In short, their approach combines multiple cues into a probability of boundary feature, which in turn is used to estimate a boundary-based affinity. These local affinities are subsequently 'globalized' by finding the eigenvectors of the relaxed normalized cut criterion. Instead of trying to form a hard segmentation out of the resulting eigenvectors we use them as pixel embeddings, which bring closer those pixels which are likely to belong to the same region and pull apart pixels which are not likely to belong to the same region. The affinity between two pixels is then computed as the euclidean distance in the embedded space. We call these soft segmentations 'Eigen'.

2. Secondly, we turn to the soft segmentation masks of Leordeanu et al. (2012), a byproduct of their work on the generalized boundary detector (*Gb*), which combines multiple low- and mid-level image representations (such as intensity, color and, in a follow-up work, optical flow (Leordeanu et al., 2014)) with a closed-form solution. Additionally, the authors propose a novel method to extract soft segmentations, using local color models built around each pixel to construct a large set of figure-ground segmentations. These are then projected to a lower dimensional subspace through PCA, which results in a low-dimensional pixel embedding. They use these soft segmentations as additional cues for *Gb*. We utilize them as-is, with code provided by the authors. We call these soft segmentations 'SoftMask'. Their main advantage is that they are substantially cheaper to compute than the 'Eigen' embeddings.

3. Lastly, we show how to extend our formulation to exploit boundary data. Unlike the two previous methods, boundary data does not provide an embedding, but rather measures the probability that two adjacent pixels may belong to different regions. As such we cannot measure the affinity between pixels with their euclidean distance. In the following section we propose a novel formulation based on the 'intervening contour' technique (see Sec. 2.3.2.1). While our method is applicable to any boundary detector, we use the state-of-the-art detector of (Dollár and Zitnick, 2013), that obtains results comparable or superior to both *Pb* and *Gb* in a fraction of the time. We refer to these boundary cues as 'Edge'. This work was developed after the publication of (Trulls et al., 2013).

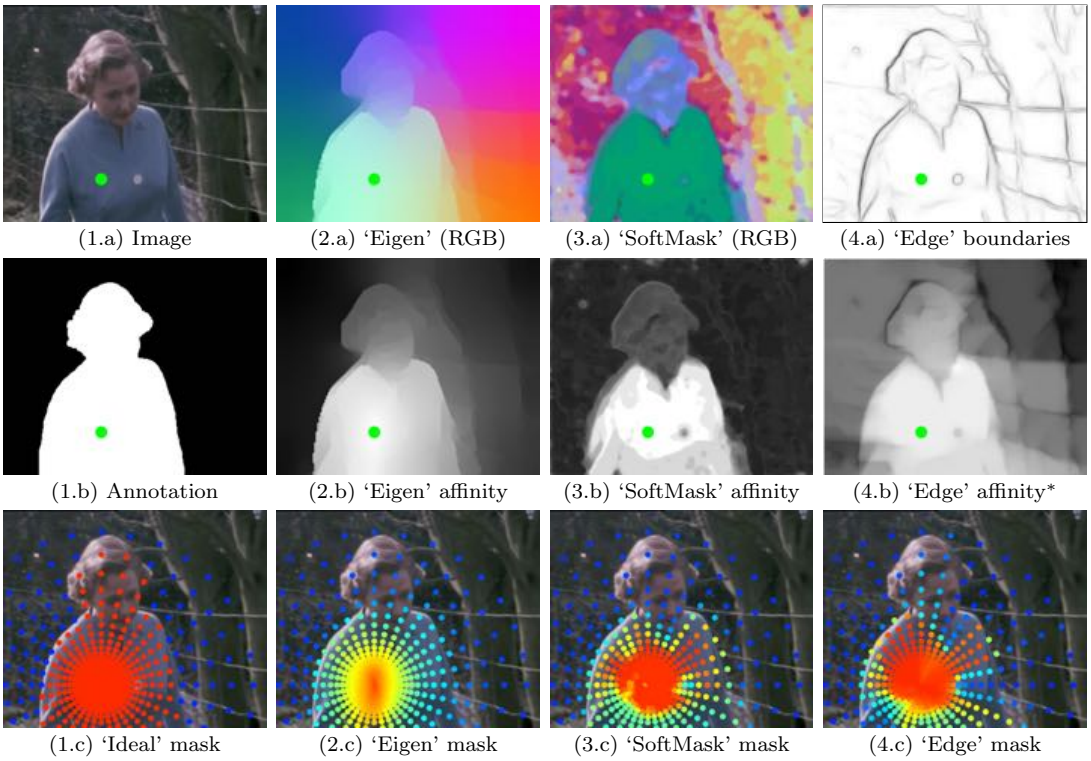


Figure 4.2: Segmentation-aware descriptor construction. **Column 1:** Given image (1.a) and a 'perfect' figure-ground segmentation (1.b), separating foreground and background measurements would be trivial: (1.c), grid points are marked in red if enabled and blue if disabled. This is unattainable—we propose alternative solutions based on different segmentation cues. **Columns 2-4:** Given segmentation cues ({2-4}.a), we can measure the 'affinity' between pairs of pixels: in ({2-4}.b) we show the affinity between the point represented by the green dot and the rest of the image. We use this per-pixel affinity to design 'gating' masks ({2-4}.c). We present procedures to leverage the Normalized-cut eigenvectors of (Maire et al., 2008) ('Eigen', column 2), the Generalized Pb soft segmentations of (Leordeanu et al., 2012) ('SoftMask', column 3) and the Structured Forests boundaries of (Dollár and Zitnick, 2013) ('Edge', column 4). Notice that (2.b) and (3.b) are for illustration: in practice we do this only for grid coordinates $\mathbf{x}[k, n]$ (pictured in the bottom row), where $[k, n]$ refer to the angular and radial coordinates, respectively. For the 'Edge' masks we use an affinity measure computed over the coordinates n, which does not lend itself to an image-based representation; for illustration purposes in (4.b) we show a distance transform instead (Borgefors, 1986). Please refer to Sec. 4.4 for details.

4.4 Segmentation-aware descriptor construction

We now describe how these segmentation cues can be used to render local descriptors robust to background variability. Our technique is equally applicable to any SIFT-type descriptor described by a grid, such as SIFT, Daisy and SID. We focus our efforts on SID, for the reasons outlined at the beginning of this chapter.

Namely, when constructing a descriptor around a point $\mathbf{x}$, we measure an affinity

$w[k,n] \in [0,1]$ between $\mathbf{x}$ and every other grid coordinate $\mathbf{x}[k,n]$, and multiply with it the respective measurements $\mathbf{D}$ extracted at angular and radial coordinates $[k,n]$:

$$\mathbf{D}'[k,n] = w[k,n] \cdot \mathbf{D}[k,n]. \tag{4.1}$$

In Eq. 4.1, $\mathbf{D}[k,n]$ represents for SID the concatenation of the H polarized and smoothed derivatives at $[k,n]$, and for SIFT the respective 8-dimensional orientation histogram. Multiplying by $w[k,n]$ effectively shuns measurements which come from the background. As such, the descriptor extracted around a point is affected only by points belonging to its region and remains robust to background variability. As our results indicate, replacing $\mathbf{D}$ with $\mathbf{D}'$ yields noticeable performance improvements.

Having provided the general outline of our method, we now describe three alternative methods to obtain the affinity function $w[k,n]$ used in Eq. 4.1, based on each of the segmentation cues considered in this chapter.

Given the globalized eigenvectors of Pb, we construct 'Eigen' embeddings $\mathbf{y}(\mathbf{x})$ by weighting the first $M = 10$ eigenvectors by a quantity dependent on their corresponding eigenvalues, as in Eq. 2.9, so that lower-energy eigenvectors (those corresponding to global structures) have a stronger weight. Pixel embeddings $\mathbf{y}(\mathbf{x})$ are thus also 10-dimensional.

Based on the assumption that the euclidean distance in $\mathbf{y}$ indicates how likely two points are to belong to the same region, we define the descriptor-level 'gating' signal, $w \in [0,1]$, between two points $\mathbf{x}, \mathbf{x}'$ as:

$$w = \exp\left(-\lambda \|\mathbf{y}(\mathbf{x}) - \mathbf{y}(\mathbf{x}')\|_2\right) \tag{4.2}$$

Here λ is a single scalar design parameter determining the sharpness of the affinity masks, which we set experimentally in Sec. 4.5.

For the 'SoftMask' embeddings we use the soft segmentations of Gb directly, without further processing, and likewise build the segmentation masks w following Eq. 4.2.

This procedure is not directly extensible to boundary data, which unlike the previous two methods does not provide an embedding, but rather measures the probability that two adjacent pixels may belong to different regions. In order to efficiently extend this measurement beyond adjacent pixels we adapt the 'intervening contour' technique of Shi and Malik (1997) to the descriptor coordinate system. We start by sampling the boundary responses on the log-polar grid of SID. We use smoothing prior to sampling, so as to achieve scale-invariant processing. This sampling provides us with a boundary strength signal $B[k,n]$ that complements the descriptor features in Eq. 2.6. We then obtain the affinity function $w[k,n]$ in Eq. 4.1 in terms of the running sum of $w[k,n]$ along the radial coordinate n:

$$w[k,n] = \exp(-\lambda \sum_{n'=0}^{n-d} B[k,n]). \tag{4.3}$$

We have introduced an additional quantity, d, in Eq. 4.3, which acts like a 'mask dilation' parameter. Namely, this allows us to postpone (by d) the decay of the affinity function w around region boundaries, thereby letting the descriptor profit from the shape information contained around boundaries. We have empirically observed that setting $d = 2$ or $d = 1$ yields a moderate improvement over $d = 0$. We refer to the segmentation masks computed in this manner as 'Edge'.

In Fig. 4.2 we show pixel affinities and segmentation masks derived from each of our three segmentation cues. For the 'Eigen' and 'SoftMask' masks, we define the affinity as the euclidean distance in the embedded space. For the 'Edge' masks, however, our affinity measure is computed over the radial coordinates n, and does not lend itself to an image-based representation; for illustration purposes, we show instead a distance transform operating over the boundary map (Borgefors, 1986). This is a richer representation that considers the full neighborhood of a pixel, rather than only the accumulated boundaries in direction k. Notice for instance how in Fig. 4.2, the eastbound ray ($\theta = 0$ in Eq. 2.6) crosses a small, foreign object (a button), and grid points further along the ray are disabled even though they belong to the same region as the center of the descriptor. It is not obvious how to exploit such formulations efficiently; we intend to explore this in the future.

4.5 Experimental evaluation

We consider two scenarios: video sequences with multi-layered motion, and wide baseline stereo. We explore the use of the different segmentation cues described in Sec. 4.3, and several dense descriptors: SID, Segmentation-aware SID, Dense SIFT, Segmentation-aware Dense SIFT, SLS, and Daisy. We use the 'S' prefix to indicate 'Segmentation-aware', so that for instance 'SSID' stands for our variant of SID.

4.5.1 Large displacement, multi-layered motion

In this experiment we estimate the motion of objects across time over the image plane, i.e. optical flow (Sec. 2.3.1). This problem is usually formulated as an optimization over a function that combines a data term, that assumes constancy of some image property (e.g. intensity), with a spatial term that models the expected variation of the flow fields across the image (e.g. piecewise-smoothness). Traditional optical flow methods rely on pixel values to solve the correspondence problem. We use SIFT-flow (Liu et al., 2011), which follows a similar formulation but exploits densely sampled SIFT descriptors rather than raw pixel values. The methodology underlying SIFT-flow, which was designed for image alignment, can be used in conjunction with any dense descriptor.

We test our approach on the Berkeley Motion Dataset (MOSEG) (Brox and Malik, 2010a), which itself is an extension of the Hopkins 155 dataset (Tron and Vidal, 2007). The MOSEG dataset contains 10 sequences of outdoor traffic taken with a handheld camera, 3 sequences of people in movement, and 13 sequences from the TV series *Miss Marple*. All of them exhibit multi-layered motion. For these experiments we consider only the traffic sequences, as in many of the others the 'objects' in the scene (e.g. people) disappear or occlude themselves (e.g. turn around), as the dataset was designed for long-term *tracking*. Ground truth object annotations, in the form of segmentation masks, are given for a subset of frames in every sequence, roughly one annotation every ten frames.

For each sequence, we match the first frame with all successive frames for which we have ground truth annotations, yielding 31 frame pairs. The images are resized to 33%, in particular to permit comparison with SLS, which has very high memory requirements. We design an evaluation procedure based on the segmentation annotations, proceeding as follows:

1. We compute flow fields for each descriptor type.
2. We use the flow estimates to warp the annotations for every frame $I_j, j > 0$ over the first frame (I_0).
3. We compute the overlap between the annotations for frame I_0 and the warped annotations for every frame $I_j, j > 0$ using the Dice coefficient (Dice, 1945) as a metric. Given two sets A and B, their Dice coefficient $2\,(|A \cap B|)\,/\,(|A| + |B|)$ measures the similarity between the sets.

We consider Dense SIFT (DSIFT) (Vedaldi and Fulkerson, 2008), SLS, SID, and SSID with 'Eigen', 'SoftMask' and 'Edge' embeddings. We use SLS both in its original form (Hassner et al., 2012) and a PCA variant developed after the publication of the paper: we refer to them as SLS-'paper' and SLS-PCA. A SLS descriptor is size 8256, whereas its PCA variant is size 528. The code for both was made publicly available by the authors.

For SID construction we rely on the dense implementation of (Kokkinos et al., 2012a). We take $K = 28$ rays, $N = 32$ points per ray and $H = 8$ polarity-preserving, oriented derivatives. We exploit the symmetry of the Fourier Transform Modulus to discard two quadrants, as well as the DC component, which is affected by additive lighting changes. The size of the descriptor is 3328 for SID and 3360 for SID-Rot. We refer to the publicly available code for further details[1].

For the SID-based descriptors we consider only their rotation-sensitive version, SID-Rot as the objects in the MOSEG sequences do not contain significant rotations, and in this case discarding orientation data to enforce invariance entails a loss of information and a decrease in performance. We use the same parameters for both SID and SSID unless stated otherwise. We select the values for the λ parameter of SSID (Eqs. 4.2 and 4.3) that render the best results: $\lambda = 0.7$ for 'Eigen', $\lambda = 37.5$ for 'SoftMask', and $\lambda = 27.5$ for 'Edge'.

Fig. 4.3 plots the results for every descriptor. Each column shows the average overlap for all frame pairs under consideration. The results are accumulated, so that the first column includes all frame pairs ($j \geq 10$), the second column includes frame pairs with a displacement of 20 or more frames ($j \geq 20$), and so on. We do so to prioritize large displacements. The sequences have varying lengths, so that the samples are skewed towards smaller displacements. As expected, SSID outperforms SID, in particular for large displacements, which are generally correlated with large j. The best overall results are obtained by SSID-Rot with 'SoftMask' embeddings, followed by SSID-Rot with 'Eigen' embeddings; the 'SoftMask' variant does better, despite its reduced computational cost. The 'Edge' boundaries also provide a boost over the segmentation-agnostic SID; while they do not perform as well as the 'SoftMask' embeddings, they reduce the cost of extracting the segmentation cues even more drastically.

Additionally, we use the flow fields to warp each image I_j, over I_0. Some large displacement warps are pictured in Fig. 4.4. Again, SSID outperforms the other descriptors.

4.5.1.1 Segmentation-aware SIFT

The application of soft segmentation masks over SID is particularly interesting because it alleviates its main shortcoming: fine sampling over large image areas to achieve

[1] https://github.com/etrulls/softseg-descriptors-release

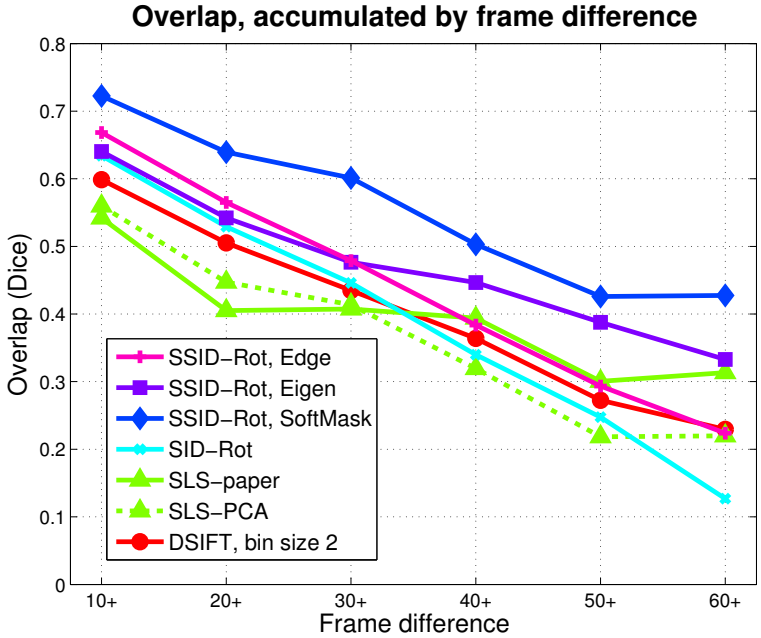


Figure 4.3: Overlap results over the MOSEG dataset (traffic sequences), for all the dense descriptors considered. Each column shows the average overlap for all frame pairs under consideration. The results are accumulated, so that the first column ('10+') includes all frame pairs, and subsequent columns ('j+') include frame pairs with a difference of j or more frames. For DSIFT we show only the results corresponding to the best scale.

invariance. We expect that this approach can be applied to other grid-based descriptors; namely SIFT. We thus extend the formulation to SIFT's 4×4 grid, using the 'SoftMask' embeddings, which consistently gave us better results with SSID. Fig. 4.5 shows the increase in performance over four different scales. The gains are systematic, but as expected the optimal sharpness parameter λ is strongly correlated with the spatial size of the descriptor grid. Fig. 4.6 displays the performance gains; note that the variability could be potentially accounted by the low number of samples (31 image pairs). This merits further study, in particular with regards to its application to the multiple scales considered in the construction of SLS descriptors (Hassner et al., 2012).

4.5.2 Wide-baseline stereo

For a second experiment, we consider stereo reconstruction. While *narrow-baseline* stereo (usually defined as two cameras separated by a short distance, pointing in the same direction) is well-understood, the same cannot be said for its *wide-baseline* counterpart.

Narrow-baseline stereo is often addressed with simple similarity measures such as pixel differencing, or block-wise operations such as SSD or NCC (Sec. 2.2). As the viewpoint increases, perspective distortion and occlusions become a problem and we cannot rely on such simple metrics. Feature descriptors are more robust, but the larger the area they consider, the more susceptible they are to occlusions. We show this problem can be alleviated incorporating segmentation cues into descriptor construction.

(a) Image 1

(b) Image 2

(c) DSIFT warp (2 to 1)

(e) SLS-PCA warp (2 to 1)

(d) SID-Rot warp (2 to 1)

(e) SSID-Rot 'SoftMask' warp (2 to 1)

Figure 4.4: Large displacement motion with SIFT-flow, for some of the descriptors considered in this work. We warp image '2' to '1', using the estimated flow fields. The ground truth segmentation masks are overlaid in red: a good registration should bring the object in alignment with the segmentation mask. We observe that segmentation-aware variant SSID does best, with noticeable improvements over its baseline, SID. Similar improvements were observed for SDSIFT over DSIFT.

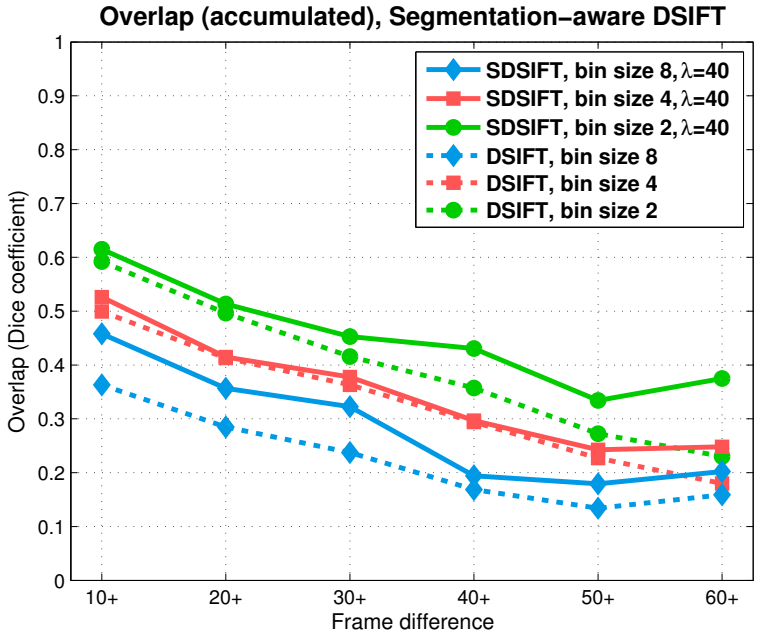


Figure 4.5: Overlap results over the MOSEG dataset for the segmentation-aware DSIFT and its baseline, at different scales (size of each bin, in pixels).

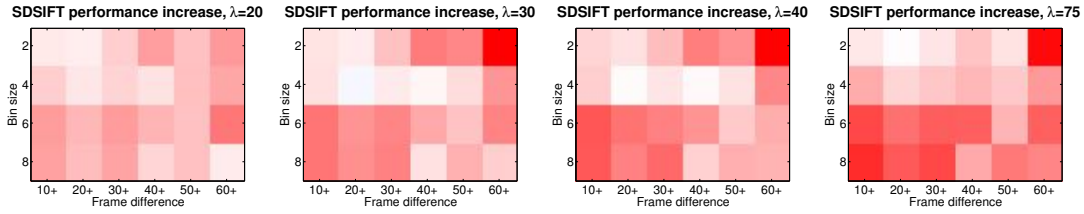


Figure 4.6: Increase in average overlap for the segmentation-aware SIFT over its baseline, for several SIFT scales, difference in frames j (accumulated), and λ values. White indicates no difference in overlap, with shades of red marking an increase in performance (the largest increase in overlap is 0.14). For clarification, note the correspondence between the figure on the third column ($\lambda = 40$) and Fig. 4.5. As expected, high λ values produce more aggressive segmentation masks and more discriminating descriptors, but the optimal λ varies with the SIFT scale.

For these experiments we use a set-up similar to that of the previous chapter:

1. We discretize 3D space into $L = 50$ depth bins, taking the reference frame of the rightmost camera.
2. Given a calibrated stereo system, we compute the distance between every pixel in one image and all the possible matching candidates over the other image, subject to epipolar constraints, within the scene range. It is in this step where we introduce specific descriptors representations.
3. We store the distance for the best match at every depth bin.
4. We feed the costs (distances) $N_x \times N_y \times L$, where N_x and N_y are the width and height of the image, to a global regularization algorithm, to enforce piecewise smoothness. Each pixel is assigned a label (depth bin) in $\mathcal{L} \in \{1, \ldots, L\}$.

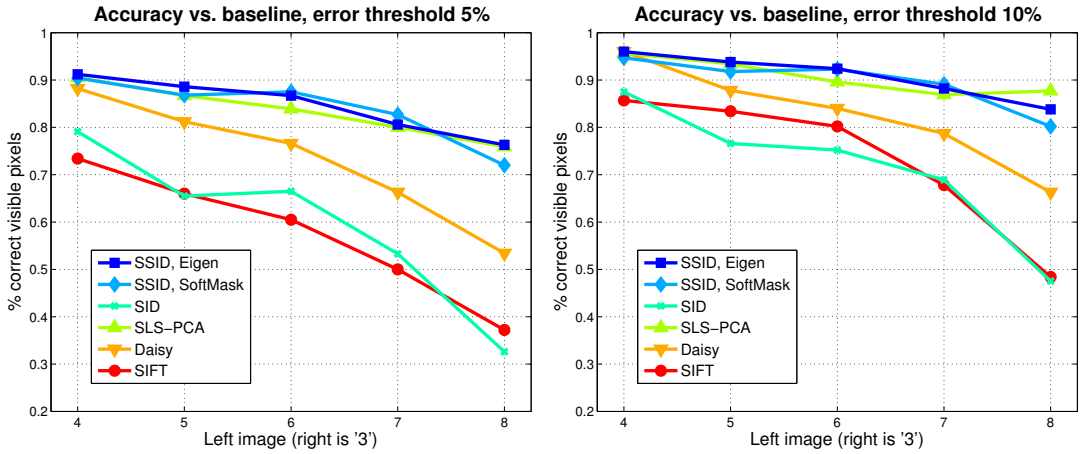


Figure 4.7: Accuracy at different baselines, for visible pixels only, for two error thresholds (expressed as a fraction of the scene range). Occlusions are not taken into account.

For the last step we use Tree-Reweighted Message Passing (Kolmogorov, 2006) with Potts pairwise costs; i.e. a constant penalty if adjacent pixels (with 4-connectivity) are assigned different depth labels, and no penalty otherwise. We add an additional label with a constant cost, to model occlusions. The occlusion layer clusters occluded pixels, penalizing 'false positives', that will typically have different labels from their neighbors, with high pairwise costs.

We use the wide baseline dataset of (Strecha et al., 2008), that contains two multi-view sets of high-resolution images with ground truth depth maps. We consider the 'fountain' set, as it contains much wider baselines in terms of angular variation than the 'herzjesu' set, which exhibits mostly fronto-parallel displacements. As in (Tola et al., 2010), we use a much smaller resolution, in our case 460×308.

First, we evaluate the accuracy of each descriptor. We compute depth maps using our stereo algorithm, and evaluate the error on every visible pixel, using the ground truth visibility maps from (Strecha et al., 2008), without accounting for occlusions. We consider DSIFT, SLS and Daisy, as well as SID and SSID. For DSIFT, SLS and Daisy we align the descriptors with the epipolar lines, to enforce rotation invariance, as in (Tola et al., 2010). For SID and SSID we consider only the *fully invariant* descriptors, and omit this step (we do not consider 'Edge' cues due to time constraints). We use SLS-PCA rather than SLS-'paper', which has much lower dimensionality: matching descriptors is the costliest step in dense stereo, and is correlated with descriptor size. We show the results on Fig. 4.7. Our SID-based segmentation-aware descriptors outperform the others, except for SLS at very large baselines; however, our approach does not require rotating the patch.

Most of the performance gains on wide-baseline stereo reported in (Tola et al., 2010) stem not from the Daisy descriptor itself, but from the novel approach to latent occlusion detection we described in the previous chapter. Their technique exploits a set of binary masks, 'half-disks' at different orientations, that disable image measurements from occluded areas: these are similar to our segmentation masks $\mathbf{w}[k, n]$, but binary (i.e. $\mathbf{w}' \in \{0, 1\}$) rather than soft ($\mathbf{w} \in [0, 1]$), and with a predetermined spatial structure. The most appropriate mask is determined on a per-pixel basis, using the

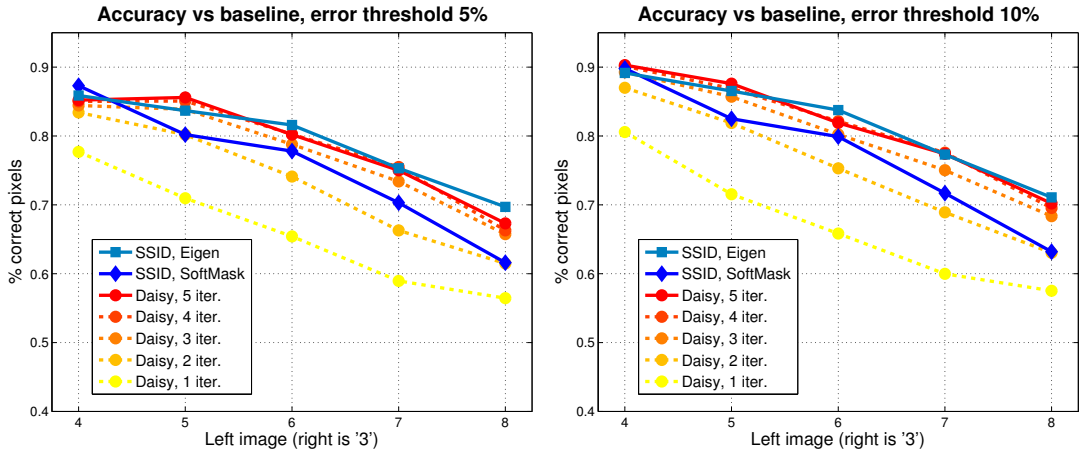


Figure 4.8: Accuracy of the iterative approach to occlusion estimation of (Tola et al., 2010) and our segmentation-based, single-shot approach, at different baselines, for two error thresholds (expressed as a fraction of the scene range).

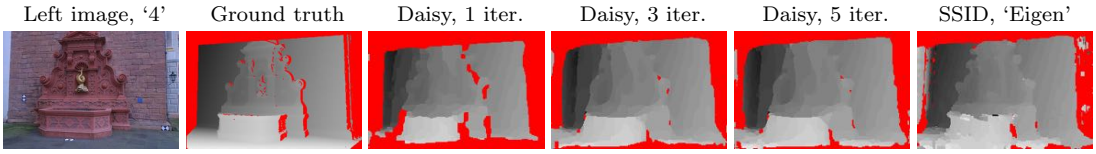
current depth estimates around the pixel to prioritize masks that disable regions with heterogenous label distributions. Subsequent iterations apply the highest-scoring masks to the descriptors as in Eq. 4.1, dropping measurements likely affected by occlusions from the similarity measure. A downside of this approach is that errors in the first iteration, which does not account for occlusions, can be hard to recover from.

Our previous experiment did not take occlusion masks into account. In a second experiment, we enable occlusion labels in the regularization step and pitch the state-of-the-art iterative technique of (Tola et al., 2010) against our segmentation-based, single-shot approach. We let the Daisy stereo algorithm run for 5 iterations, and show the results in Fig. 4.8. The performance of SSID with 'Eigen' embeddings is comparable or superior to that of Daisy for most baselines; we achieve this in a single step, and without relying on calibration data to enforce rotation invariance. Additionally, note that we set the λ parameter of Eqs. 4.2 and 4.3 based on the motion experiments, and do not adjust them for for the stereo problem. Fig. 4.9 shows the depth estimates at different baselines. Notice how, even though the algorithm can converge, the occlusion estimates for the first Daisy iteration at the largest baseline are very aggressive. The occlusion cost is a nuisance parameter which must be tuned carefully; and more so if the iterative masks are used.

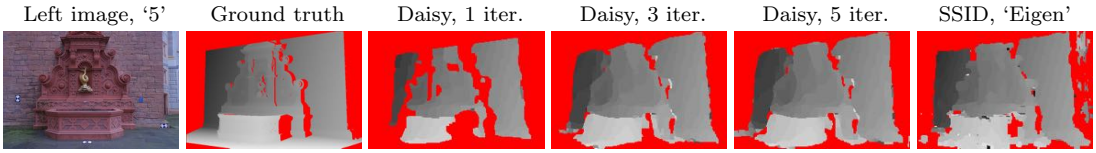
4.5.2.1 Computational requirements

The cost of computing dense SIFT descriptors with VLFEAT for an image of size 320×240 is under 1 second (MATLAB/C code). SLS (MATLAB) requires ~21 minutes. SID (a non-optimized MATLAB/C hybrid) requires ~71 seconds. SSID requires ~81 seconds, in addition to the extraction of the masks. Note that for all the experiments in this chapter we compute the 'Eigen'/'SoftMask' embeddings at the original resolution (e.g. 640×480) before downscaling the images. The 'SoftMask' embeddings (MATLAB) require ~7 seconds per image, and the 'Eigen' embeddings (MATLAB/C hybrid) ~280 seconds. Structured forest boundaries can be computed at multiple frames per second. The computational cost of matching two images with the SIFT-flow framework depends

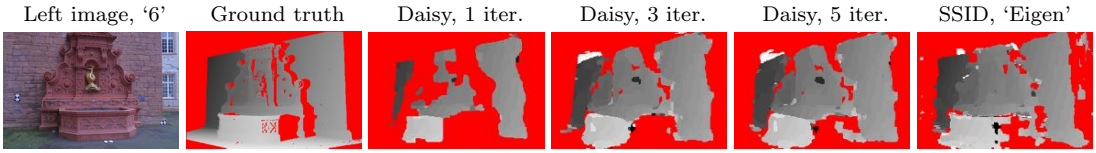
(a) Right image, '3'



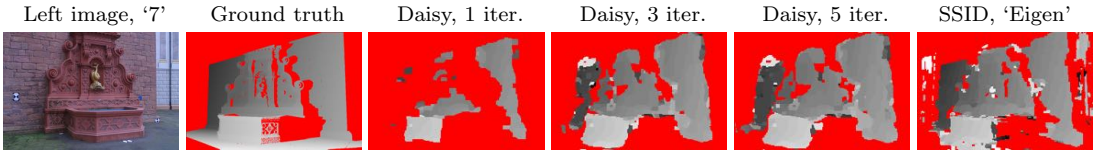
(b) Stereo reconstruction for {4,3}, over 3.



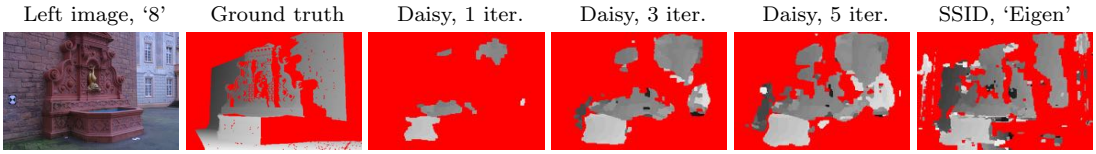
(c) Stereo reconstruction for {5,3}, over 3.



(d) Stereo reconstruction for {6,3}, over 3.



(e) Stereo reconstruction for {7,3}, over 3.



(f) Stereo reconstruction for {8,3}, over 3.

Figure 4.9: Qualitative comparison of the iterative stereo approach of Tola et al. (2010) with our segmentation-based, single-shot approach. We compute depth maps for image pairs $\{x, 3\}, x \in [8, 4]$, with an increasing baseline. All the reconstructions are over '3', the rightmost viewpoint, pictured in **(a)**. **(b-f):** the first column shows the reference images; the second column the ground truth; columns 3-5 the Daisy reconstructions at iterations 1, 3, and 5; and the rightmost column the single-shot SSID-'Eigen' reconstruction. Occluded pixels are marked in red.

on the size of the descriptors, varying from ~14 seconds for SIFT (the smallest) to ~80 seconds for SID/SSID, and ~10 minutes for SLS-'paper' (the largest).

4.6 Summary and future work

In this chapter we propose a method to address background variability at the descriptor level, incorporating segmentation data into their construction. Our method is general, simple, and carries a low overhead. We use it to obtain segmentation-aware descriptors with increased invariance properties, which are of the same form as their original counterparts, and can thus be plugged into any descriptor-based application with minimal adjustments. We apply our method to SIFT and to SID descriptors, obtaining with the latter dense descriptors that are simultaneously invariant to scale, rotation and background variability.

We demonstrate that our approach can deal with background changes in large-displacement motion, and with occlusions in wide-baseline stereo. For stereo, we obtain results with SID comparable to the mask-based, state-of-the-art latent occlusion estimation of Daisy (Tola et al., 2010). We (a) without relying on calibration data to obtain rotation-invariance; and (b) in a single step, rather than with iterative refinements. While similar in spirit (both methods 'gate' the features to achieve invariance against background variability), our approach is also applicable to the case where a *single image* is available.

Regarding future work, the immediate follow-up would be the application of our segmentation-aware descriptors to detection and classification of object categories. We have showed that our approach is amenable to the scenarios we have explored in this chapter, where the assumption of a static background is invalid. In addition to background motion, we have shown that our segmentation-aware SID can handle scaling, viewpoint changes and occlusions. But both of these scenarios consider *different views of the same scene*, which differ in time (motion) or viewpoint (stereo). It is unclear how to handle *similar views of different scenes* as the effect of inter-class variability on our segmentation-based approach remains in question. This is the focus of the next chapter, where we extend our segmentation-aware feature extraction technique to HOG features for recognition with sliding window detectors.

Additionally, the segmentation-aware SID suffers from high dimensionality, but is likely very redundant. This shortcoming could be addressed with spectral compression and also with metric learning (Strecha et al., 2012). The latter proved able to both drastically reduce dimensionality problems while still increasing the discriminative power of descriptors.

Chapter 5

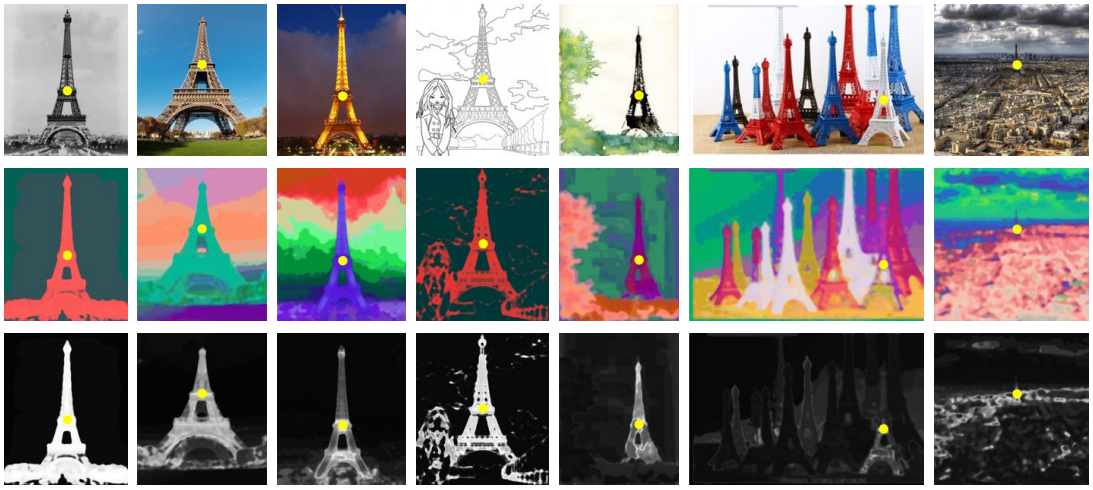
Segmentation-aware Deformable Part Models

We closed the last chapter wondering how to extend our segmentation-based approach to discard background variability to the general problem of object recognition, i.e. *similar views* of *different scenes*. We illustrate this problem in Fig. 5.1, where we show descriptors computed for the same point over very different views and representations of the same object (the Eiffel tower). We observe two issues. Firstly, high inter-class variability can result in segmentation masks that are not always consistent across images, such as for the illuminated tower in Fig. 5.1-(c), the drawing in Fig. 5.1-(e), and the figurine in Fig. 5.1-(f).

We could expect to address this issue with formulations more sophisticated than the exponential penalty function of Eq. 4.2, but it is not clear how to adjust them across very different images; in (Trulls et al., 2013) we obtained the best results with simple functions. Secondly, it is unclear how to deal with objects that can appear at any position and scale, as in Fig. 5.1-(g). We address these issues, and propose a technique, inspired by our previous work, to exploit segmentation cues for object detection, in the context of sliding window detectors. This chapter is based on our 2014 CVPR paper (Trulls et al., 2014).

5.1 Introduction

Sliding window classifiers are the method of choice for object detection in the high-recall regime, as these ensure that no objects 'fall through the cracks' of a segmentation front-end. However, even if a putative detection window is tightly surrounding the object, background structures can creep into the low-level features extracted from the image, adversely increasing the variability of the input signals. This is typically the case for highly deformable or non-convex objects (e.g. tables, cats, dogs) that do not naturally occupy a rectangular area, and therefore background structures often appear in the rectangular box containing them.



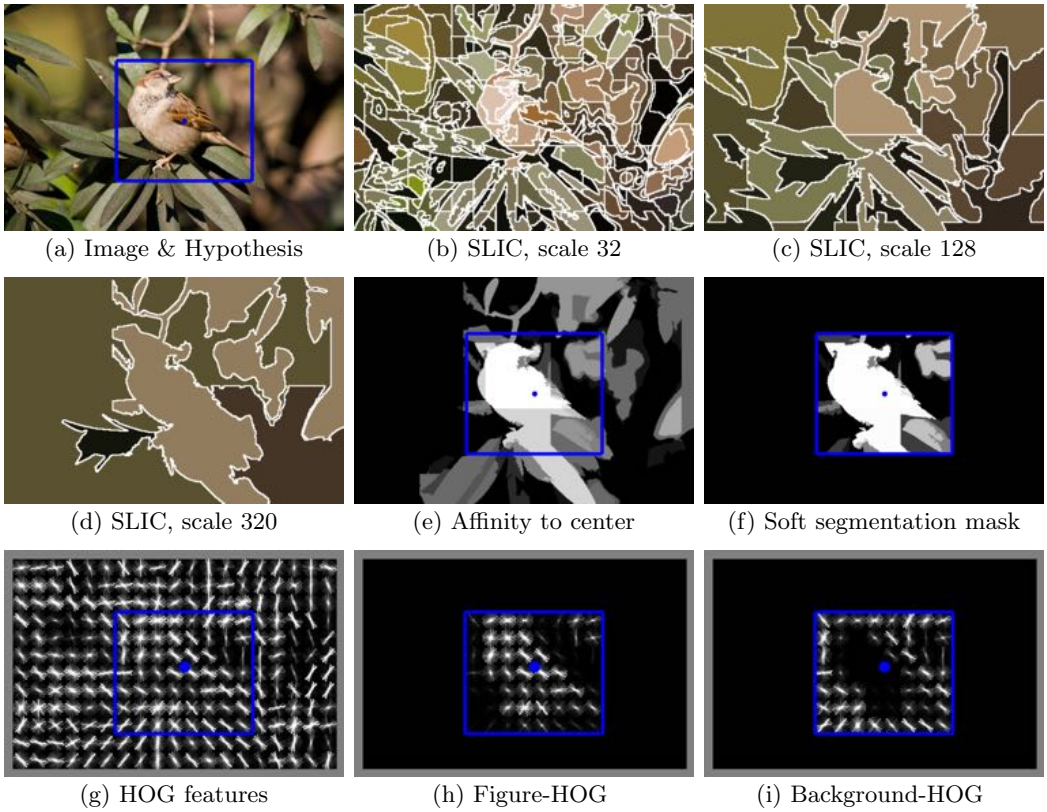
(a) (b) (c) (d) (e) (f) (g)

Figure 5.1: Pixel-wise segmentation masks obtained with the procedure presented in the previous chapter, for different representations of the Tour Eiffel. The top row shows the input image, the middle row the soft segmentations of Leordeanu et al. (2012), and the bottom row the affinity to the pixel represented by a yellow dot, as per Eq. 4.2. Our approach can fail under significant changes in appearance. This problem is greatly exacerbated when the object can be placed anywhere in the image and be of any size, which is often the case in object detection.

We propose to address this problem through segmentation. The combination of segmentation and recognition is a long-standing problem in computer vision; it is well understood that high-level priors are important for disambiguating low-level tasks, while segmenting low-level information can provide clean stimuli to subsequent processing. Even though segmentation and recognition could potentially be in a continuous feedback loop, the complexity of the resulting systems can be daunting when it comes to implementation, in particular due to the interleaving of segmentation and recognition. In particular, techniques that couple segmentation and recognition through the optimization of objective functions that reflect their interplay (Kumar et al., 2010; Kokkinos and Maragos, 2009; Gould et al., 2009; Gao et al., 2011; Maire et al., 2011) are still not practically exploitable in for fast object detection.

Rather than going for a full-blown coupling of segmentation and recognition, our approach is less intrusive, and thus easier to implement and transfer to other problems as well. We propose a technique to combine bottom-up segmentation, in the form of SLIC superpixels (Achanta et al., 2012), with sliding window detectors, such as Deformable Part Models (DPM) (Felzenszwalb et al., 2010b). As in the previous chapter, the merit of our approach lies in 'cleaning up' the low-level HOG features, based on the spatial support of SLIC superpixels. This can be understood as using segmentation cues to split feature variation into object-specific and background changes. In contrast to our work with descriptors, we do not commit to a single segmentation. Instead, we use a large pool of SLIC superpixels computed at multiple scales, and combine them in a scale-, position- and object-dependent manner to build soft segmentation masks. This allows us to deal with appearance changes and to address the multi-scale nature

of object detection, encompassing both large objects that occupy the whole image as well as the small components typically captured with parts filters (see Sec. 2.4.3). The segmentation masks can be computed fast enough that we are able to repeat this process over every candidate window, during training and detection, for both the root and part filters of DPMs. We use these masks to construct enhanced, background-invariant features to train DPMs.



(a) Image & Hypothesis (b) SLIC, scale 32 (c) SLIC, scale 128

(d) SLIC, scale 320 (e) Affinity to center (f) Soft segmentation mask

(g) HOG features (h) Figure-HOG (i) Background-HOG

Figure 5.2: Overview of our method. We take as input **(a)** an image and a detection hypothesis: a bounding box, represented in blue. We compute **(b-d)** a set of SLIC superpixels at different scales (we show 3 scales); once per image. We pick the superpixels which contain the center of the bounding box, i.e. the blue dot, and rank them according to how well they fit the box, using intersection over union as a metric. In **(e)** we show the aggregated response of the $N = 15$ best-matching superpixels, which can be seen as an affinity between the center of the box and the rest of its contents. We use this affinity to build a soft segmentation mask, pictured in **(f)**. Given HOG features **(g)**, we use the segmentation mask to split feature variation into a figure-HOG channel **(h)**, and a background channel **(i)**. These features are processed by a DPM-based classifier. Our algorithm operates over HOG blocks rather than pixels, for efficiency; here we use pixel data for illustration.

Our main technical contribution consists in improving the performance of low-level, gradient-based features such as HOG (Dalal and Triggs, 2005), and is inspired by recent advances in feature descriptors (Tola et al., 2008; Trulls et al., 2013) and sliding-window detectors (Ott and Everingham, 2009; Fidler et al., 2013). Our method is illustrated in Fig. 5.2. Firstly, we compute multi-scale SLIC superpixels, once per

image. At every position of a multi-scale sliding window detector, we construct a foreground mask 'on the fly', determining the most appropriate segmentation scales automatically. For this we use intersection over union with the detection window to rank the SLIC superpixels, rejecting segments that are unable to capture objects inside the box, by virtue of being too small, too large, or spilling over outside the detection window. We use the most relevant superpixels to build soft segmentation masks, measuring the affinity between pixels (or HOG blocks) and the putative object position, on a per-window basis. Rather than constructing a binary foreground mask through some discrete optimization procedure, as in e.g. (Gao et al., 2011), we do this in a soft manner, building on our work on segmentation-aware descriptors. As in the previous chapter, we use a segmentation 'hardness' design parameter, and introduce an additional parameter to determine 'how much' of the segmentation masks is desirable; both can be adapted per class, with cross-validation. Finally, we use the segmentation masks to build background-invariant features; in this case we split HOG features into foreground and background channels, as for some object categories the background can be informative.

Other than simplicity, a main advantage of our method is its computational efficiency. In particular, we exploit SLIC superpixels (Achanta et al., 2012), which have a computational overhead of a fraction of a second per image. Unlike (Fidler et al., 2013; Gao et al., 2011), our approach extends naturally to both root and part filters, while incurring a minimal additional computational cost. The segmentation process outlined in Fig. 5.2 is fast enough to be performed 'on the fly' for all object hypotheses in sliding window detection.

We validate our approach by applying it to the DPM filters of Felzenszwalb et al. (2010b). Keeping all other modelling parameters identical, our segmentation-based variant of HOG delivers consistent improvements in detection performance on PASCAL VOC 2007, outperforming standard DPMs in 17 out of 20 classes, yielding an average increase of 1.7% in AP (Average Precision). Lastly, we extend this method to segmentation-aware descriptors for dense SIFT matching with large-displacement optical flow; there we attain results comparable to those in our earlier work (Trulls et al., 2013), but in a fraction of the processing time used therein. The code for the work presented in this chapter is publicly available[1].

These are the key strengths of our approach:

Simplicity. We present an alternative formulation to the exponential function used to build background-invariant descriptors in chapter 4. Our masks involve two 'hardness' parameters, adjusted per class.

Speed. The segmentation masks can be computed fast enough that we can apply our procedure to exhaustive search with sliding window detectors. For reference, we can compute the masks for the root filter in ~2.1 seconds (266k sliding window iterations); this is already faster than convolving HOG features and filters, which takes ~5 seconds.

Generality. In addition to detection with DPMs, we show how to extend it to dense SIFT descriptors, tying it with our previous work.

The next section references the state of the art on segmentation in recognition. In Sec. 5.3 we briefly recap Deformable Part Models, introduced in chapter 2. We

[1] https://github.com/etrulls/dpm-masks-release

present our method for segmentation-aware DPMs in Sec. 5.4, and extend it to SIFT descriptors in Sec. 5.5.

5.2 Related work

In the previous decade several works extended segmentation techniques such as curve evolution (Rousson and Paragios, 2002; Tu et al., 2003; Cremers et al., 2002; Cremers, 2006; Moreno-Noguer et al., 2008; Kokkinos and Maragos, 2009) and graph cuts (Kumar et al., 2010; Lempitsky et al., 2008; Ladicky et al., 2010) to combine model-based information with region- and contour-based terms. However, with the exception of the rigid model of (Lempitsky et al., 2008), these works assume that a shortlist of object 'proposals' is available beforehand, and can thus help object detection only by pruning false positives, rather than helping objects 'pop up'.

A tighter coupling of segmentation and recognition is pursued in semantic segmentation, where object-specific appearance terms influence image labeling, e.g. (Shotton et al., 2006; Gould et al., 2009; Kumar and Koller, 2010), without necessarily relying on the outputs of an object detection module. Even though the latest techniques (Weiss and Taskar, 2013; Carreira et al., 2012) deliver compelling results, their impact on recognition performance has only very recently been explored (Fidler et al., 2013). Finally, such techniques can be computationally demanding, involving some form of discrete optimization for segmentation, or object-tailored cascades (Weiss and Taskar, 2013), meaning a substantial overhead for multi-category detection.

Coming to using a segmentation front-end for detection, originally (Russell et al., 2006; Malisiewicz and Efros, 2007; Pantofaru et al., 2008) used multiple image segmentations to obtain a rich set of object hypotheses in the context of learning. In (Ahuja and Todorovic, 2007; Gu et al., 2009), hierarchical segmentation was used to shortlist detection hypotheses, while Russakovsky and Ng (2010) used Steiner trees to search through a set of segmentations for the pairing of regions to object hypotheses.

The current state-of-the-art techniques (Uijlings et al., 2013; Manén et al., 2013) deliver a compact, yet sufficient set of proposals at a rate of multiple frames per second, thereby guiding the application of more demanding classifiers, such as bag-of-words. A more recent thread of works, relevant to the 'objectness' idea (Alexe et al., 2010), is that of learning to segment in an object-independent manner (Endres and Hoiem, 2014; Carreira and Sminchisescu, 2010). Still, these works can harm recall if object positions are missed by the segmentation front-end. In contrast, we do not use segmentation to shortlist object positions, but rather construct a segmentation 'on the fly' for every window position.

Turning to sliding-window variants, which are more similar in spirit to ours, Ramanan (2007) applies local figure-ground segmentations post-hoc to prune false positives in pedestrian detection; this however is not taking segmentation into account when training a classifier. In (Vedaldi and Zisserman, 2009) a model which explicitly accounts for truncated objects in both training and detection was shown to provide increased performance in detection. Gao et al. (2011) consider forming a binary segmentation mask per bounding box hypothesis using graph-cuts; they accelerate detection using branch-and-bound, but this still takes a couple of seconds for single root filters, while it is not straightforward how to extend their method to part-based models. In (Ott and Everingham, 2009) the Fisher criterion is used to create a per-patch, soft figure-ground

segmentation, which is then summarized through a HOG descriptor. By contrast we bring superpixels into play, and also learn to detect from segmentation-sensitive HOG features.

Most recently, Fidler et al. (2013) combined semantic segmentation results with DPM-based detection, by constructing additional features that measure the overlap of a putative bounding box and the region assigned to an object hypothesis, given by a semantic segmentation. This yields substantial improvements in performance, yet requires running first the semantic segmentation algorithm of Carreira et al. (2012), which requires multiple seconds per frame, on a 6-core machine, for feature extraction. Our approach yields more modest improvements, but incurs a negligible additional computational cost. It can thus be understood as a segmentation-sensitive variant of DPMs that is geared towards efficiency.

5.3 Deformable Part Models

Deformable Part Models (Sec. 2.4.3) have provided a solid basis for extensions and refinements that have been continually raising the bar for object detection and pose estimation. Special emphasis has been put on reducing computational complexity at test time by adopting coarse-to-fine processing (Pedersoli et al., 2011), cascades (Felzenszwalb et al., 2010a), or the branch and bound algorithm in combination with k-d trees (Kokkinos, 2011). Azizpour and Laptev (2012) mitigate the dependence of the latent-SVM on heuristic rules to initialize the location of the part filters by introducing part annotations for a subset of the PASCAL VOC classes. Other works aim at reducing the size of DPMs by using shared parts (Ott and Everingham, 2011) or by building a large vocabulary of part templates, using linear combinations of a considerably smaller set of basis parts (Pirsiavash and Ramanan, 2012). A recent paper introduces part sharing with 'shufflets' (Kokkinos, 2013), shiftable mid-level structures shared across different types of objects, which speeds up detections with very little performance loss.

These works rely essentially on the same type of features, i.e. local gradient data. HOG features aggregate edge strengths at neighbouring blocks; they have been proven particularly effective for human and object detection, and are simple and fast to compute. The combination of bottom-up and top-down information has shown to be more effective than relying on either exclusively in a broad range of tasks, from low-level problems such as boundary and symmetry detection (Arbeláez et al., 2011; Leordeanu et al., 2012; Tsogkas and Kokkinos, 2012) or constructing feature descriptors (Stein and Hebert, 2005; Tola et al., 2010; Trulls et al., 2013), to higher-level problems such as object detection. We argue that complementing local HOG features with higher-level, global knowledge extracted from a set of segmentation cues (SLIC superpixels) we can boost the performance of object detectors based on DPMs. We base our work on the latest release[2] of the discriminatively trained DPM filters of Felzenszwalb et al. (2010b).

DPMs were introduced in chapter 2. To recap, they represent objects as a star-shaped graphical model, with a 'root' node at the center, corresponding to the entire object domain, and 'leaf' nodes indicating the deformable object parts. As seen in Sec. 2.4.3, the score for a specific arrangement of a root filter x_0 and n part filters

[2] http://people.cs.uchicago.edu/~rbg/latent-release5

$x_1, \ldots, x_n$ is given by the combination of a unary and a pairwise term. Per Eq. 2.13:

$$S(x_0, x_1, \ldots, x_n) = \sum_{p=0}^{n} \langle w_p, G(x_p) \rangle + \sum_{p=1}^{n} D_p(x_p, x_0), \tag{2.13}$$

where $G(x_p)$ indicates the image-based features at position x_p, w_p is the template for part p, $\langle w_p, G(x_p) \rangle$ is the score obtained for placing part p in position x_p, and $D_p(x_p, x_0)$ is a quadratic penalty function over the parts. This configuration allows the filters to capture the variance in pose appearance, while restricting part placements that would lead to unrealistic configurations.

In this chapter we are primarily concerned with building better unary features, which are defined as the inner product of the HOG pyramid features G_k at scale k and the DPM filters:

$$\sum_{x',y'} F[x', y'] \cdot G_k[x + x', y + y'], \tag{5.1}$$

where F are rectangular templates of dimensionality d, the same as the data; and x and y map to HOG blocks. We thus convolve each filter with each HOG pyramid level; see Fig.2.17 for an illustration. Rather than use 36-dimensional HOG features as in (Dalal and Triggs, 2005), Felzenszwalb et al. (2010b) show that a PCA-based representation can bring the features down to $d = 11$ with little to no loss in performance, while speeding up learning and detection. In practice, they observe that the top eigenvectors are nearly constant along each row and column of the 4×9 measurement matrix (i.e. 4 blocks over 9 orientations), and define an alternative feature representation summing over each row and column, so that $d = 4 + 9 = 13$. Computing these features is much less costly as it does not involve PCA projection, and they have a simple interpretation: 9 orientation features plus a measure of the overall gradient energy over 4 different areas around a cell.

5.4 Superpixel-grounded DPMs

Our contribution lies in modifying the local features G used in the unary term of Eq. 2.13, so as to exploit segmentation information. In particular, inspired by the recent success of integrating segmentation and image descriptors in (Tola et al., 2010; Trulls et al., 2013), we apply a similar approach to feature extraction for object recognition. For this, as illustrated in Fig. 5.2, we efficiently compute a large pool of image segments which are then combined to build segmentation masks for any putative object hypothesis. These foreground and background masks allow us to decouple the effects of background changes from class-specific appearance variability.

Our segment hypotheses are obtained using SLIC superpixels (Achanta et al., 2012) with the implementation of (Vedaldi and Fulkerson, 2008) in a fraction of a second. We extract superpixels over 7 scales, ranging from 200-250 down to 10 superpixels per image, and for five different regularisation values; these parameters are chosen by inspection and used for all experiments. We make our code available, and therefore omit exact parameter values. This strategy provides us with a large pool of candidate segments of different sizes, valid both for objects that can take up the whole image and also for small image parts.

For every candidate detection hypothesis (i.e. every sliding window step) we consider only the superpixels which contain the center of the hypothesis' bounding box.

We then use intersection over union with the candidate window and each valid superpixel as a matching metric, and select the top $k = 15$ matching superpixels. This procedure allows us to automatically determine the most appropriate scale for each detection window, rejecting superpixels that are too small or too large, as well as those that fall too much outside the window.

Fig. 5.3 shows the best- and worst-matching superpixels for several detection windows for some PASCAL VOC images, and the resulting pixel-wise masks, including typical failure cases. In Fig. 5.3-(5) a 'foreign' object (a cushion) lies on the center of the box containing the object (a sofa). The box in Fig. 5.3-(8) exhibits significant appearance changes, due to a dark screen over a light casing. Furthermore, categories like 'person', in Fig. 5.3-(9), are notoriously hard to segment. We also show how our approach tackles parts such as the head, in Fig. 5.3-(10), or the hands, in Fig. 5.3-(11).

Averaging the top superpixels provides us with an affinity measure $f \in [0, 1]$, indicating how likely it is that two pixels or blocks belong to the same region. Indexing HOG blocks (or 'cells') by i, we denote this affinity measure with $f[i]$, where i ranges over the filter size (e.g. $i \in [1, 6] \times [1, 6]$ for a 6×6 part filter). We use this affinity to build segmentation masks over the window using a sigmoid function parameterized by a 'segmentation hardness' parameter λ:

$$M[i] = \frac{1}{1 + \exp\left(-\frac{10}{1-\lambda}(f[i] - \lambda)\right)} \tag{5.2}$$

This expression ensures that for $f[i] = 1$ we will have $M[i] \approx 1$ regardless of λ, so $\lambda \in [0, 1)$ can be determined in a per-category manner through cross-validation. Fig. 5.4 shows additional examples over multiple scales and object categories, including typical failure cases.

We use these soft masks as weights over the HOG features, to separate those that share an affinity with the center of the bounding box: $G^+[i] = M[i] \cdot G[i]$. As the background can be informative for some object categories, we also consider the complementary set of features, $G^-[i] = (1 - M[i]) \cdot G[i]$. Our extended feature array is the concatenation of (i) the original features, (ii) the figure-ground channels, and (iii) the mask itself:

$$G^{seg}[i] = \left[G[i], G^+[i], G^-[i], M[i]\right]. \tag{5.3}$$

These extended, segmentation-aware features can be applied to the DPM training/detection pipeline directly. The cost of computing and scoring the superpixels and building the masks is small compared to the actual cost of the convolution; training is however more costly, due to increased feature dimensionality. Our implementation extends the fast convolution with SSE extensions (a set of instructions that can greatly increase performance) of the latest DPM code release. In the future we will also consider exploiting recent advances on fast DPM detection (Kokkinos, 2011, 2013).

5.4.1 Segmentation mask 'alpha-blending'

The assumptions behind our segmentation method may not hold for certain categories. For instance, in the case of bicycles, the center of a bounding box does often not belong to the object, whereas bottles may contain transparencies or specularities, and people or man-made objects like vehicles are often composed by elements with very different appearance which are hard to segment as a whole; some examples are illustrated

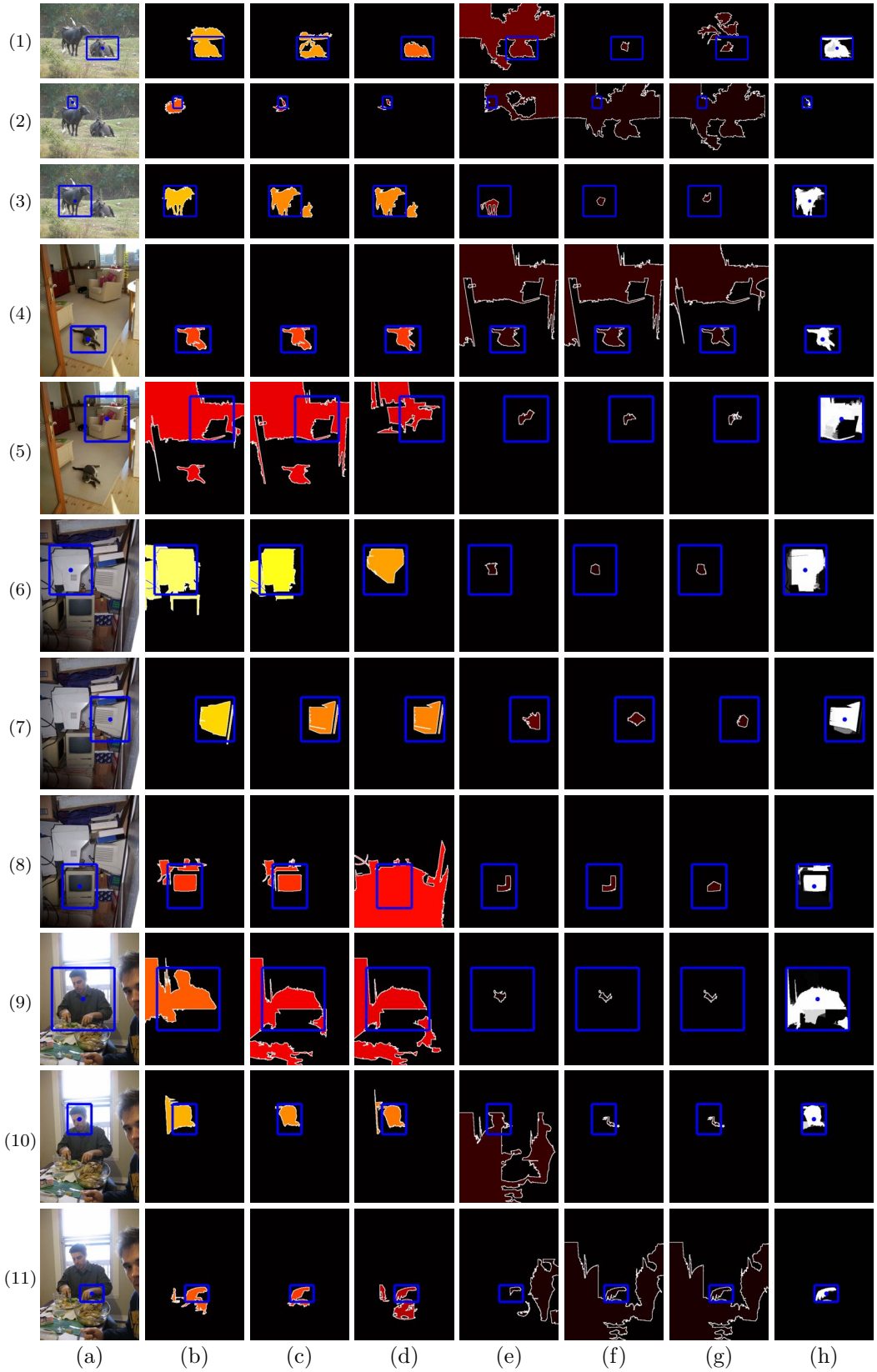
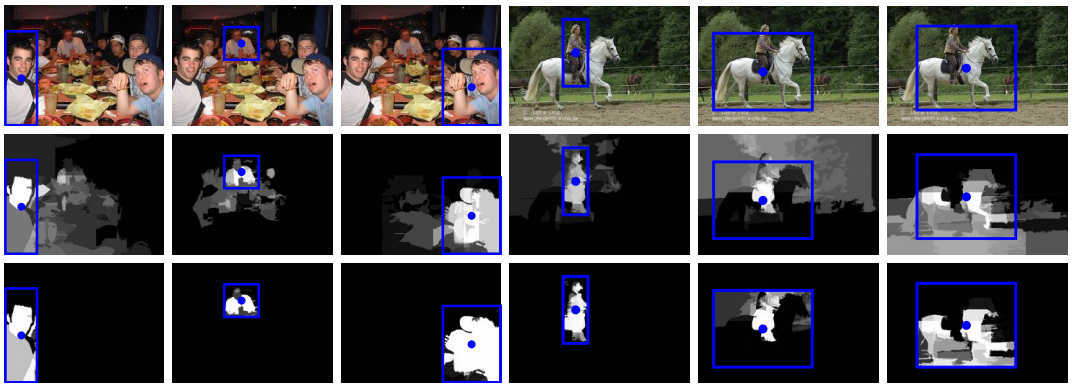


Figure 5.3: (a) Input image and detection window (blue rectangle). (b-d) Top-3 scoring superpixels. (e-g) Lowest-3. (h) Segmentation mask. Superpixels are color-coded by score. (5,8) are failure cases.



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

Figure 5.4: Examples of our segmentation masks over different scales and object categories. **Top row:** input image and detection window (blue rectangle). **Middle row:** pixel affinity. **Bottom row:** soft segmentation mask. Note that even when the center of the bounding box does not contain the object, as is the case in (c), ranking the superpixels by how well they match the window can help us recover. This can still be problematic over extreme examples such as those (e-f), where the rider correctly detected in (d) occludes the middle part of the horse, effectively breaking the object in two. We deal with these scenarios by keeping the original HOG features.

Fig. 5.8. As suggested by Table 5.1, using segmentation features for such categories may actually result in a loss in performance.

We address this problem with a strategy similar to that of 'alpha-blending' for images. Namely, given a design parameter $\alpha \in [0, 1]$, we define new masks M_α as:

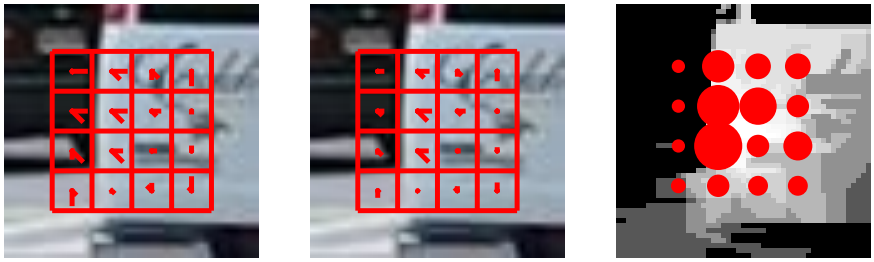
$$M_\alpha[i] = (1 - \alpha)1[i] + \alpha M[i], \tag{5.4}$$

where $1[i]$ indicates the unit function, and apply these over the feature array G as before. For $\alpha = 1$ (or 100%) we have our full-blown segmentation-sensitive features, while for α tending towards 0, the foreground-HOG channel G^+ becomes equal to the HOG features G, while G^- tends to 0. For intermediate values of α we work with features that blend between these two extremes.

For both λ and α we use cross-validation to separately fix the right parameter values per object category.

5.5 Superpixel-grounded descriptors

Having described our method on using superpixels to decompose HOG features into foreground and background channels, we now describe how we can use similar ideas to address the problem we had originally considered in (Trulls et al., 2013). There we introduced a methodology to build soft segmentation masks for dense SIFT and SID descriptors based on low-dimensional pixel embeddings derived from the eigenvectors of Maire et al. (2008) and the soft segmentations of Leordeanu et al. (2012). These embeddings were used to compute pixel affinities with the euclidean distance over embedded space; we then built soft segmentation masks with an exponential function.



(a) SIFT (b) SSIFT (c) Soft mask

Figure 5.5: SIFT (a), and the SLIC-based segmentation-aware SIFT (b). The response of background pixels is greatly attenuated. (c) shows the soft segmentation mask computed from SLIC superpixels, and its magnitude at the SIFT grid coordinates, which are the weights applied over the descriptor bins.

Following Eq. 4.2, we define:

$$w^{exp}[i] = \exp\left(-\lambda(1 - f[i])\right) \tag{5.5}$$

where f is the SLIC-based affinity, computed between grid positions i and the center of the descriptor (rather than pixels or HOG blocks in the DPM formulation), and λ is a design parameter. In this chapter we used a sigmoid function instead (Eq.5.2). We thus define an alternative mask function, following this formulation:

$$w^{sigm}[i] = \frac{1}{1 + \exp\left(-\frac{10}{1-\lambda}(f[i] - \lambda)\right)}. \tag{5.6}$$

In the experiments of Sec. 5.6.2 we will consider both functions to build SLIC-grounded descriptors. Note that in either case, the λ parameters can be used to adjust the 'hardness' of the segmentation masks. As before, the last step is to 'gate' the SIFT features with the soft segmentation masks, as per Eq. 4.1.

To adapt this technique to descriptors we face additional issues. In our approach to object detection, we rely on a detection window to determine the object's support, effectively searching for superpixels that are commensurate with the window. This technique does not translate directly to feature descriptors. In fact, descriptors so not typically contain whole objects (or parts of objects), but rather features of arbitrary scale. Here, we are mainly insterested in capturing *region discontinuities*.

We thus adjust the technique of Sec. 5.4 in the following ways:

1. We use SLIC superpixels that are approximately the same size or larger (at least 50%) than the image patch. With this we avoid the appearance of artificial boundaries inside homogeneous regions due to over-segmentation.
2. We use all the valid superpixels that contain the current pixel, rather than ranking them with intersection-over-union and picking the top n superpixels. This simplifies the technique.

It remains unclear how to extend this approach to SID, given the large image patches it considers. In this chapter, we show how to build background-invariant SIFT descriptors with SLIC-based segmentation masks; a full integration of the procedure introduced in this chapter with the techniques introduced in (Trulls et al., 2013) is left

for future work. Qualitative results for this method are shown in Fig. 5.5; a quantitative evaluation follows in the next section.

5.6 Experimental evaluation

We present two experiments: the DPM-based object detector introduced in Sec. 5.4, and the segmentation-aware SIFT of Sec. 5.5.

5.6.1 Object detection on the PASCAL VOC

We evaluate the performance of our approach on the detection task of the PASCAL VOC 2007, where the goal is to predict the bounding boxes of each object present in the test images. VOC contains 20 object categories and a total of 9,963 annotated images, divided into three sets: training (`train`, 2,501 images), validation (`val`, 2,510 images), and test (`test`, 4,952 images). We train the filters over `train`, and evaluate them over `val` to cross-validate the model parameters. Finally, we retrain with the chosen parameters over `trainval` and report results over `test`.

Detection performance is evaluated with the precision/recall curves, and summarized by the average precision (AP). Both can be obtained with the standard development kit provided by the organizers of the challenge (Everingham et al., 2010). Detections must be provided in the same format as the annotations, i.e. as bounding boxes, and are determined as true or false positives based on the overlap with the ground truth: their intersection over union must exceed 50%. Multiple detections of the same object in an image are considered as false detections.

We use standard DPM as a baseline, with the same parameters and settings as the segmentation-aware DPM; we only change the low-level features. Regarding our detector, we consider five different values for the 'segmentation hardness' parameter, $\lambda \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$, and pick the best value for each object category with two-fold cross-validation. As explained in Sec. 5.4, we also apply 'alpha-blending' to determine 'how much' of the segmentation masks is desirable over different object categories. In particular, after determining λ we follow the same procedure for 'alpha-blending', with α values 100% (i.e. no blending), 75%, 50%, and 25%.

Our approach outperforms standard DPM on 17 out of 20 classes, for an average improvement of 1.7% AP (1.3% without 'alpha-blending'). We report the results in terms of average precision (AP) in Table 5.1, and the precision/recall curves in Fig. 5.6. We display the per-class increase in performance over DPM in Fig. 5.7. Fig. 5.8 shows some examples of the soft masks generated by our filters on the PASCAL VOC.

5.6.2 Large-displacement motion

We follow the procedure outlined in the previous chapter (Sec. 4.5.1) to compute large displacement flow estimates with dense SIFT and SIFT-flow. We evaluate descriptor performance with 31 image pairs of traffic sequences with ground truth segmentation annotations from the Berkeley Motion Dataset (Brox and Malik, 2010a), all of which feature multi-layered motion. Our metric is the Dice overlap coefficient (Dice, 1945) between the ground truth mask for the first frame and the ground truth for the k-th frame warped over the first frame with the flow estimates.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	—
DPM	31.5	54.9	7.8	11.1	**29.9**	**51.2**	53.9	23.3	22.2	25.1	—
Ours (λ)	**33.2**	55.9	**11.9**	**12.3**	28.9	49.1	53.8	26.6	**23.1**	25.4	—
Ours (λ, α)	**33.2**	**56.0**	**11.9**	11.8	28.6	49.1	**54.1**	**26.7**	**23.1**	**27.3**	—

	table	dog	horse	mbik	pers	plant	sheep	sofa	train	tv	Avg.
DPM	29.2	10.6	56.8	44.5	**40.4**	13.6	20.2	30.2	42.2	42.0	32.0
Ours (λ)	**29.9**	**14.1**	**59.8**	45.8	39.4	**14.7**	21.2	34.3	44.4	**42.6**	33.3
Ours (λ, α)	**29.9**	13.5	**59.8**	**46.8**	39.8	**14.7**	**22.8**	**36.1**	**46.7**	**42.6**	**33.7**

Table 5.1: AP performance (%) on the PASCAL VOC 2007, for DPM (first row) and for our method, with cross-validated λ (second row), and with 'alpha-blending' (third row). We cross-validate α only for the best λ per category; i.e. we do not run a full α/λ sweep. Entries where the second and third rows are equal correspond to those where $\alpha = 100\%$, i.e. no 'alpha-blending' is used.

The results are shown in Fig. 5.9, for different scales (i.e. the spatial size of the descriptor cell). We include only the best λ for every case. The dataset provides ground truth annotations roughly every ten frames, and the results are accumulated, so that e.g. the first bin contains every frame pair, the second bin contains every frame pair separated by 20 or more frames, and so on. We report results for the segmentation masks computed with both the exponential function of Eq. 5.5 and the sigmoid function of Eq. 5.6.

The SLIC-based SDSIFT show better performance than DSIFT, and closely match the results obtained in the previous chapter. The SLIC-based masks are however faster to compute. Furthermore, when using 'hard' superpixel segmentations the affinity between pixels can be computed through binary membership operations, rather than euclidean distances as in Eq. 4.2, which would result in yet another acceleration.

5.7 Summary and future work

The combination of segmentation and recognition is a long-standing problem in computer vision. In this chapter we have presented a simple technique to combine bottom-up segmentation with object detection, using SLIC superpixels to build soft segmentation masks. We extract a large pool of multi-scale SLICs and combine them in a scale-, position- and object-dependent manner. This procedure is fast enough that we can repeat it at every step of a sliding-window detector. We focus on building better features and plug them into a standard, state-of-the-art detection pipeline, i.e. Deformable Part Models. We use the segmentation masks to 'clean up' the HOG features, for both the root and part filters. We evaluate our segmentation-enhanced features on the PASCAL VOC and demonstrate consistent improvements in all but three categories.

We also extend the same design principle to build background-invariant SIFT descriptors, following our work in the previous chapter. We remove the descriptor features

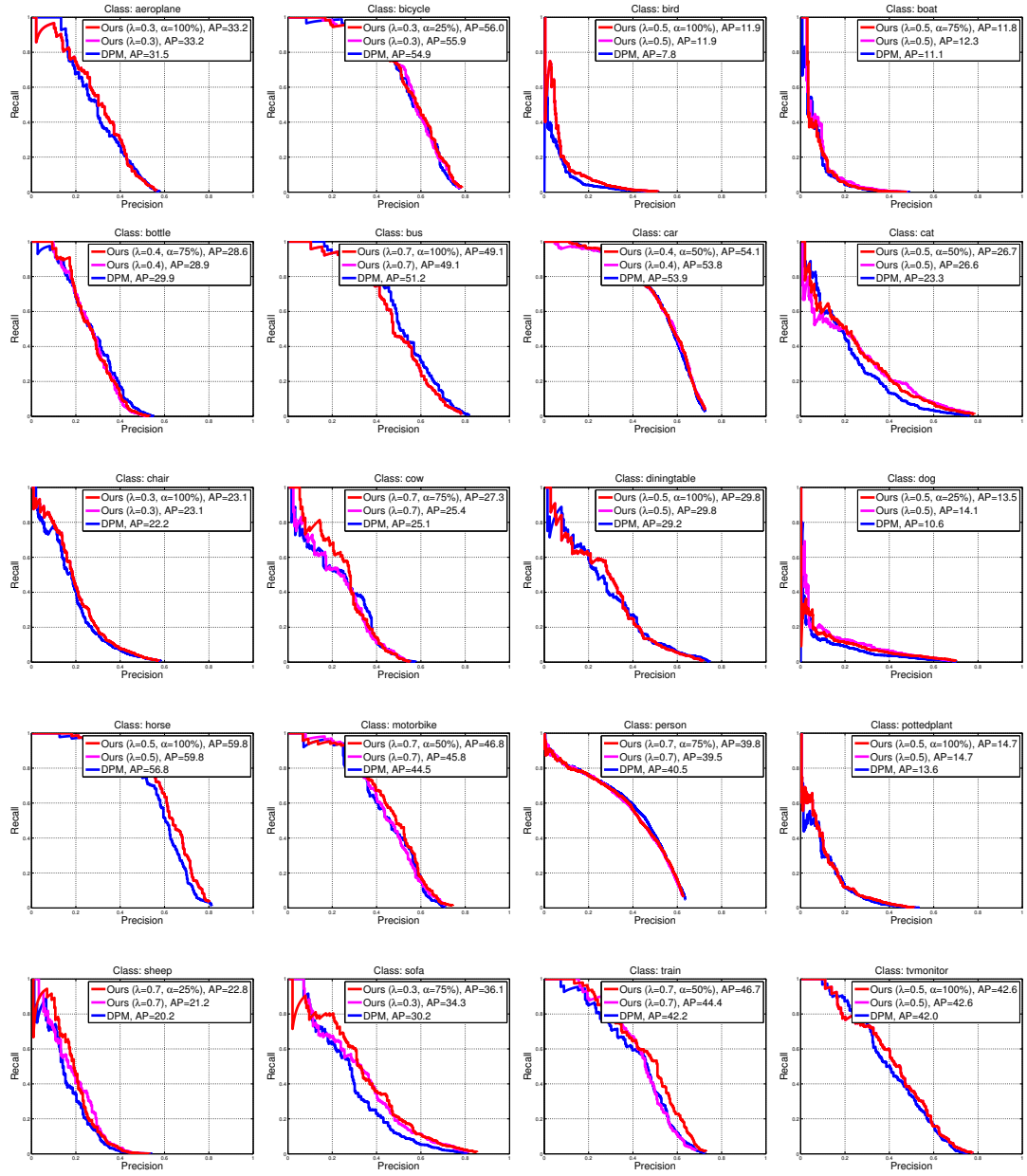


Figure 5.6: Precision/recall curves on the PASCAL VOC 2007, for: (a) standard DPMs (blue); (b) segmentation-aware DPMs with cross-validated λ (magenta); and (c) segmentation-aware, alpha-blended DPMs (red). Note that (b) and (c) are the same filter if $\alpha = 100\%$.

corresponding to regions which share little affinity with the center of the descriptor, and thus make them more robust against background motion and occlusions. Again, this process is fast enough that we can use it to compute dense descriptors.

Regarding future work, an obvious extension would be to try to pick the right λ for every object instance, rather than object category, effectively determining the 'hardness' of the masks on a per-object basis. It is unclear how to fit this into the training pipeline.

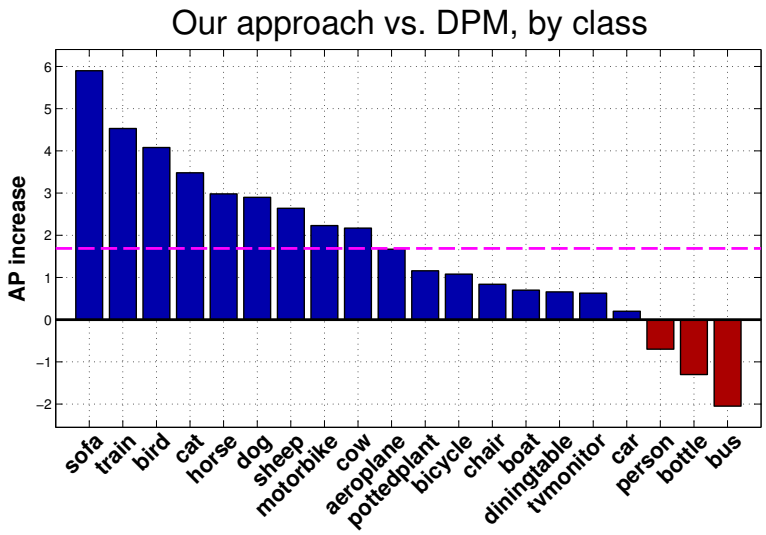


Figure 5.7: Increase in performance (AP in %) between our segmentation-aware DPM, with 'alpha-blending', and standard DPM, for every class. The dashed line in magenta marks the average AP increase.

A possible criticism of our method is that it takes as a reference the center of a bounding box, which may not actually contain the object. This is hardly a problem in practice, as we prioritize segments with a size commensurate to that of the detection window, thus rejecting those that would capture small details such as the piece of background visible through a bike frame, or the hole in a doughnut. Regardless, our approach can fail in some cases, as pictured in Fig. 5.8. This inspires us to design richer models to build the masks, perhaps considering the statistics of the superpixels. So far, we have explored only soft segmentations, defined as the weighted sum of 'hard' segments. We intend to explore binary figure/ground segmentations based on grabcut-like optimizations (Rother et al., 2004); these procedures are typically expensive, but fast variants exist (Tang et al., 2013).

Lastly, we are currently exploring alternatives to SLIC superpixels. These include:

- *'Soft' segmentation cues*, such as those used in the chapter 4.
- *'Hard' segmentations*, such as those of Carreira and Sminchisescu (2010), which formed the basis of the high-level semantic segmentations exploited in (Fidler et al., 2013). Faster alternatives include (Felzenszwalb and Huttenlocher, 2004; Arbeláez et al., 2014; Humayun et al., 2014).
- *Boundary features*, which encapsulate high-level information in a soft manner. Recent works have introduced boundary detectors that are not only accurate but also increasingly efficient (Leordeanu et al., 2012; Lim et al., 2013; Dollár and Zitnick, 2013). We already explored this venue in chapter 4, but it is not clear how to deal with large scale differences, which often happen in object detection.

Regarding feature descriptors, we have integrated this approach to SIFT, and demonstrate an increase in performance for the motion experiments. But as we argued in the previous chapter, our segmentation-based approach is particularly interesting for descriptors like SID which rely on fine sampling over large areas. We believe this is only a starting point for leveraging segmentation in feature extraction for recognition,

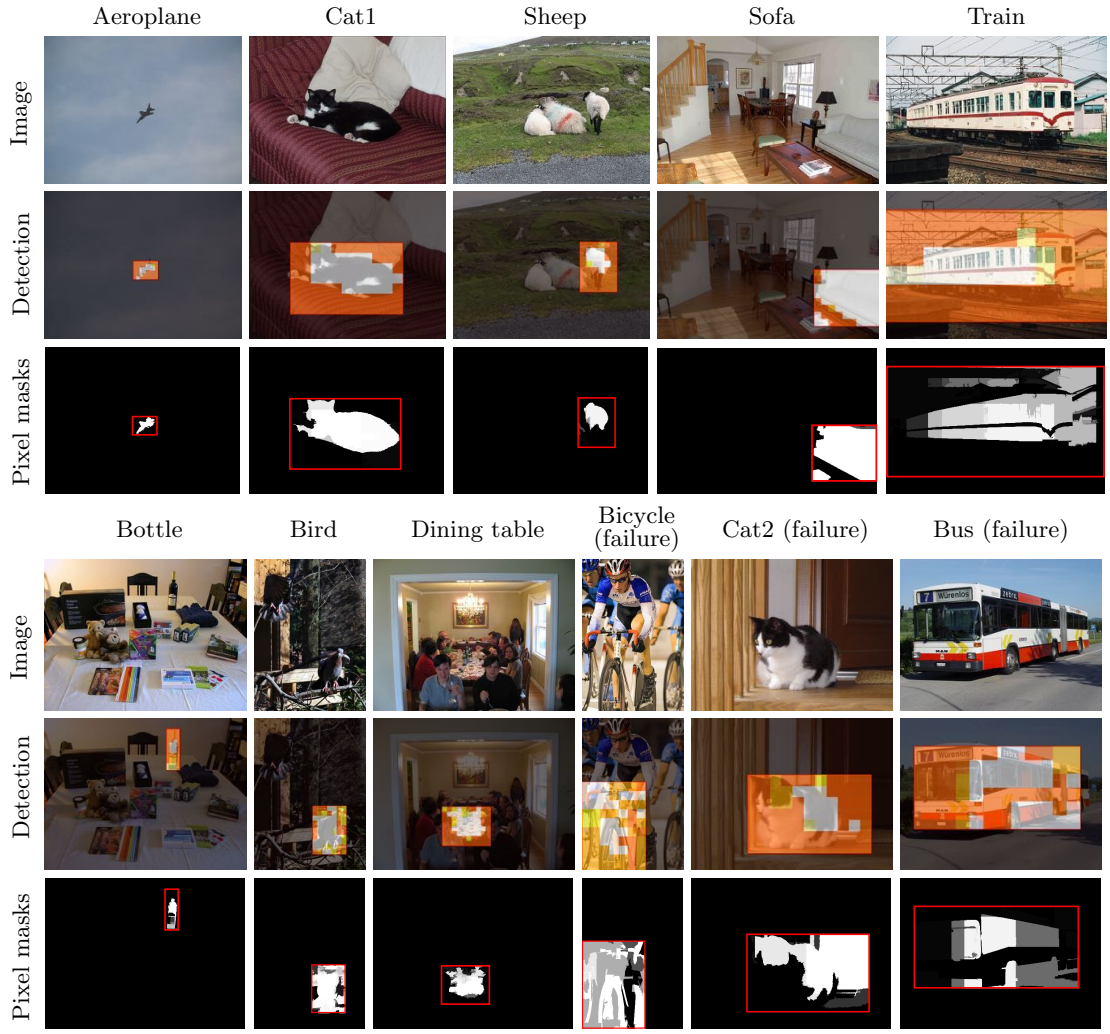


Figure 5.8: Top row: input image. **Middle row:** detection hypothesis, with the segmentation mask generated by our filters overlaid on top, from white (on) to red (off); we show only the root filter, for clarity. **Bottom row:** masks computed at a pixel level; note that the actual filter response is in HOG blocks (middle row). We include some failure cases, such as 'cat2' and 'bus', where the center of the bounding box has a different appearance than the rest of the object (due to patterned fur in the former and side windows in the latter), or 'bicycle', which contains an object hard to segment.

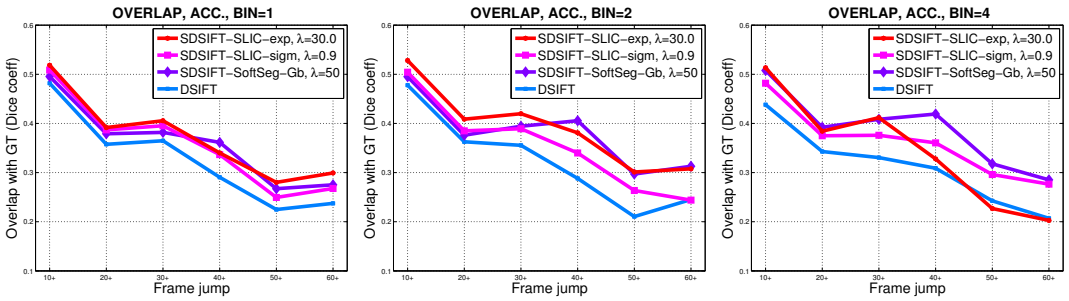


Figure 5.9: Overlap with the ground truth annotations of (Brox and Malik, 2010a), for DSIFT (blue); DSIFT with 'SoftMask' segmentation masks (purple); and DSIFT with SLIC-based masks (magenta and red, for masks built with the exponential function of Eq. 5.5 and the sigmoid of Eq. 5.6, respectively), at different scales.

and the work undertaken in this chapter has provided us with ideas for further work on the subject which we are currently pursuing, with the ultimate goal of publishing a comprehensive study of segmentation cues for feature descriptors and sliding window detectors. Additionally, the introduction of segmentation in convolutional network classifiers is one of our current research directions.

Chapter 6

Concluding remarks

The main objective of this thesis was to explore strategies to enhance local features with global, mid-level cues. We have presented three works towards that goal. We have demonstrated that 'top-level' problems such as 3D reconstruction or object detection can be addressed not only with increasingly complex models, but also with 'smarter' features.

Our main contributions were outlined in the introductory section. To recap:

1. We have presented a novel approach to build spatiotemporal features. We built on Daisy descriptors, augmented with flow priors—both are computed over a single viewpoint, which allows us to construct 'perspective-invariant' features, applicable to wide-baseline stereo. We used them for 3D stereo reconstruction, along with spatiotemporal regularization constraints, demonstrating an increase in performance.
2. We have devised a technique to exploit soft segmentation cues, using them to build feature descriptors invariant to occlusions and background changes. We applied this principle to SIFT and to SID. With the latter we deliver descriptors with very strong invariant properties. We have presented two different types of segmentation cues, and benchmarked our features on two different applications.
3. Concerned with the more general problem of recognition across different objects and scenes, we have proposed an alternative to the previous technique. We used SLIC superpixels to extract segmentation masks in a position-, scale- and object-dependent manner. We do so efficiently and for every putative detection of a sliding window classifier. We used our segmentation masks to split HOG features into object and background channels, feeding them into the standard Deformable Part Models training pipeline, demonstrating improvements over the PASCAL VOC.

These works share some important characteristics:

- We take low-level features (oriented gradient histograms) and enhance them with global cues, feeding a measure of 'big picture' information into local characteristics.

- Following recent trends in recognition, we focus on dense descriptors—while keeping efficiency in mind. For DPMs, we show how to enhance features in a manner that can be applied to every hypothesis of a sliding window detector, with a small overhead.
- Our algorithms are simple and generic, with little need for tuning. We do not require data to learn from (with the exception of the DPM training pipeline, which works at a higher abstraction level), and the parameters we do use have physical meaning and can be adjusted empirically, on a per application basis.
- We build enhanced features that can be plugged into standard frameworks and applications, and benchmarking them against their respective baseline features is straightforward. We do so throughout this thesis. Our modules can be seen as a 'black box' between feature *extraction* and feature *use*.
- We make our code public, for future reference.

6.1 Future work

We have indicated open avenues for future research in each chapter. The most important are:

On our spatiotemporal descriptors. (a) We intend to explore more efficient feature representations, which in their current form are probably highly redundant due to concatenating similar descriptors across time. (b) We also intend to investigate the use of the flow itself as a feature, particularly if we pair our reconstruction algorithm with scene flow estimation, enforcing consistent motion in addition to feature similarity. (c) Our features could be used for monocular applications, densely or on interest points. While many datasets on action recognition focus on a static camera and background, our approach could prove useful in more complicated, and practical, settings.

On segmentation-aware descriptors and DPMs. (a) In both cases (descriptors and object detectors) we intend to investigate additional segmentation cues, as well as the use of combinations of cues at the same time, along with the new strategies to build the masks. We are exploring new soft segmentations, 'hard' multi-scale segmentations, semantic segmentations, and binary cues. (b) Likewise, we are considering new alternatives to build the masks, including fast graph-based optimizers to build binary segmentations on a per-pixel or -window basis. (c) Regarding the segmentation-aware descriptors, we are exploring strategies to reduce their size, first with spectral analysis, and later with metric learning techniques (Sec. 2.2.6).

Our works are interrelated. On one hand, we work on depth, motion and segmentation, which are three basic building blocks of perception systems: parts of a whole. On the other, we use similar principles, that is: we build from reliable local features, and exploit global priors to operate on their spatial structure, focusing on simplicity and generality whenever possible. Our spatiotemporal descriptors could use segmentation cues, in addition to flow-based cues, particularly around object boundaries. Segmentation itself can be used as a cue to make inferences about geometry, and reason about structure (Hoiem et al., 2007a) and occlusions (Hoiem et al., 2007b). Our segmentation-aware descriptors could be used along with flow data for multi-layered motion, as sharp discontinuities in the flow fields are themselves indicative of object

boundaries. Depth cues, either estimates or actual measurements from RGB-D sensors, could be used instead of or along with our segmentation cues to find occlusions. Lastly, developments on our efforts in recognition with DPMs provide insight to our work with feature descriptors, and vice-versa.

6.2 Current trends in recognition

The focal point of the work presented in this thesis was building better low-level features for registration and recognition. As such, we would be remiss if we did not mention the upheaval the field is currently undergoing. We will examine two separate problems, feature learning and object detection.

Regarding **feature learning**, a comprehensive study of features in computer vision circa 2012-2014 cannot ignore the resurgence of deep learning. Convolutional neural networks (CNN) are arguably one of the 'hot topics' in computer vision.

CNNs are a type of Artificial Neural Network (ANNs), particularly relevant for computer vision applications. ANNs are a family of classifiers which consist of densely-connected networks of simple computational elements, called neurons—a neuron will e.g. compute a weighted sum of its inputs, subtract a threshold, and pass the result to a non-linear function. ANNs often have one or more hidden layers, which transform the input with a learned non-linear transformation, projecting it into a space where it can become linearly separable. They can have simple topologies, but in practice there are reasons to use multiple hidden layers, which is known as 'deep learning'.

In CNNs, neurons are placed so that they respond to overlapping regions of the image, exploiting spatially local correlation. Small groups of neurons process small portions of the input image—the 'convolutional' layers. They have much fewer connections and parameters than classical ANNs, and are thus easier to train, which makes them amenable to work with pixel data. They were relatively popular in the 90s but fell out of fashion until 2012, when Krizhevsky et al. (2012) showed drastic improvements on the Imagenet Large Scale Visual Recognition Challenge (ILSVRC) image classification task, which contains half a million labeled images of 1000 different classes (Russakovsky et al., 2014). Their network (Fig. 6.1) contains five convolutional layers, some of which are followed by pooling layers, and three fully-connected layers at the end—650,000 neurons and 6 million parameters in all. This architecture contains a few twists over previous proposals, which account for its success.

This work kickstarted a new wave of interest in deep learning for computer vision, with dozens of papers on different recognition problems at CVPR 2014 and the upcoming ECCV 2014, many of which outperform the state of the art (He et al., 2014; Zhang et al., 2014b; Gkioxari et al., 2014; Toshev and Szegedy, 2014; Taigman et al., 2014; Erhan et al., 2014; Hariharan et al., 2014; Ciresan et al., 2013). Most notably, Krizhevsky's architecture was recently applied by Girshick et al. (2014) to obtain a relative improvement of over 30% with respect to the previous top performer on the detection task of the PASCAL VOC 2012.

Until now, the most successful features for recognition (SIFT, HOG) have been carefully crafted. It is now a legitimate question whether we are on the brink of overcoming the necessity to worry about 'feature engineering'—and learn them instead. As we stated in the introduction, we believe they are not going away anytime soon. Firstly, there are aspects of deep belief networks that we do not truly grasp yet—

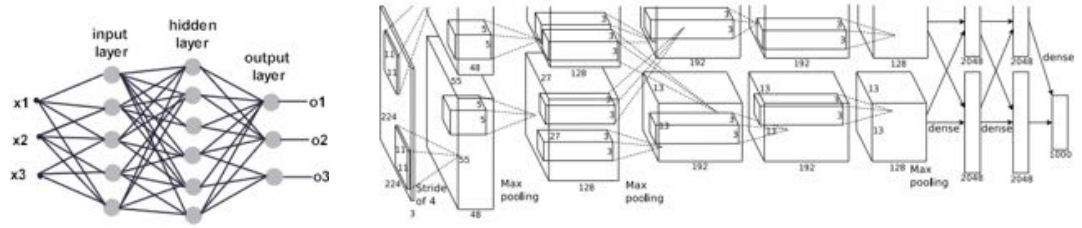


Figure 6.1: Two ANN architectures. Left: a basic multi-layer perceptron, with a single hidden layer. Right: the deep learning model introduced by Krizhevsky et al. (2012), which won the 2012 ILSVRC—featuring five convolutional layers, some of which are followed by pooling layers, and a classic three-layer, fully-connected Perceptron at the end. The output of the last fully-connected layer is fed into a softmax which produces a distribution over the 1000 classes of the ILSVRC.

recent papers attempt to understand the reasons behind their success (Szegedy et al., 2014; Chatfield et al., 2014; Zeiler and Fergus, 2014). Secondly, they have performed excellently on categorization tasks, but it remains unclear how to apply them to other problems, such as inferences on scene geometry. They can also be costly to train and deploy, and learning rich features requires large amounts of annotated data. CNNs are powerful—but not universal.

In addition, **object detection** is also undergoing a paradigm shift, moving away from exhaustive, sliding-window approaches in favor of efficient search with segmentation cues. This line of research started with 'Objectness' (Alexe et al., 2010), CPMC (Carreira and Sminchisescu, 2010) and Selective Search (van de Sande et al., 2011), with more works following in their footsteps (Manén et al., 2013; Cheng et al., 2014; Weiss and Taskar, 2013; Zitnick and Dollár, 2014). The specifics vary, but the general idea is to generate a large (~10k-100k) number of proposals (segments or windows) that describe the image regions most likely to contain *objects*. These proposals are overlapping, and often noisy and inaccurate, but put together offer an alternative to exhaustive exploration. They are often pruned to a more manageable number (~1k) and evaluated separately.

Such systems have historically underperfomed on the recall side, but the tide is turning, specially combined with rich features based on CNNs that do not fit naturally into the sliding window formulation. This was the approach followed by (Girshick et al., 2014): pre-training the networks on a data-rich auxiliary task (the ILSVRC) and transferring them to detection on the PASCAL VOC, using Selective Search to find the most relevant windows on detection. A significant advantage with respect to sliding-window detectors, as DPMs, is that we do not need to worry about aspect ratios—each putative detection window is resized into a square image, which is a requirement of Krizhevsky's architecture, and the network will learn all the latent variables—or that is the hope.

Even so, sliding windows remain a viable option for detection, and development on DPMs, very active the last few years, has not stalled. Technical complications impede their use with features *learned* with CNNs, but recent works are attempting to bridge the gap (Zhang et al., 2014a; Savalle et al., 2014), and we expect more developments to come.

Bibliography

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282. 8, 12, 28, 31, 80, 82, 85

Ahuja, N. and Todorovic, S. (2007). Learning the taxonomy and models of categories present in arbitrary images. *Proceedings of the International Conference on Computer Vision*. 83

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). FREAK: Fast Retina Keypoint. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 24

Alexe, B., Deselaers, T., and Ferrari, V. (2010). What is an object? *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 83, 100

Allaire, S., Kim, J., Breen, S., Jaffray, D., and Pekar, V. (2008). Full orientation invariance and improved feature selectivity of 3D SIFT with application to medical image analysis. *Computer Vision and Pattern Recognition Workshops*. 44

Arbeláez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916. 29, 84

Arbeláez, P., Pont-Tuset, J., Barron, J. T., Marques, F., and Malik, J. (2014). Multiscale combinatorial grouping. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 67, 93

Azizpour, H. and Laptev, I. (2012). Object detection using strongly-supervised deformable part models. *Proceedings of the European Conference on Computer Vision*. 84

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*. 27

Barnes, C., Shechtman, E., Goldman, D. B., and Finkelstein, A. (2010). The generalized PatchMatch correspondence algorithm. *Proceedings of the European Conference on Computer Vision*. 66

Basri, R., Hassner, T., and Zelnik-Manor, L. (2010). Approximate nearest subspace search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 22

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding*. 13, 14, 44, 65

Beauchemin, S. S. and Barron, J. L. (1995). The computation of optical flow. *ACM Computing Surveys*. 27

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6). 30

Belongie, S. and Malik, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24). 14, 44, 65

Berg, A. and Malik, J. (2001). Geometric blur for template matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 14, 65

Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*. 68, 70

Bosch, A., Zisserman, A., and Munoz, X. (2006). Scene classification via pLSA. *Proceedings of the European Conference on Computer Vision*. 15, 65

Bouguet, J.-Y. (2013). Camera calibration toolbox for MATLAB. http://www.vision.caltech.edu/bouguetj/calib_doc. 34

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239. 3, 28, 35, 36, 44, 48

Bronstein, M. and Kokkinos, I. (2010). Scale-invariant heat kernel signatures for non-rigid shape recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 15

Brown, M., Hua, G., and Winder, S. (2011). Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24, 65

Brown, M. and Lowe, D. (2007). Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*. 13

Browning, N. A., Grossberg, S., and Mingolla, E. (2007). Heading from optic flow of natural scenes during motion processing by cortical areas MT and MST. *Society for Neuroscience Conference*. 26

Brox, T. and Malik, J. (2010a). Berkeley motion segmentation dataset. http://lmb.informatik.uni-freiburg.de/resources/datasets/moseg.en.html. 27, 70, 90, 95

Brox, T. and Malik, J. (2010b). Object segmentation by long term analysis of point trajectories. *Proceedings of the European Conference on Computer Vision*. 45

Brox, T., Rosenhahn, B., Gall, J., and Cremers, D. (2010). Combined region- and motion-based 3D tracking of rigid and articulated objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 27

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. *Proceedings of the European Conference on Computer Vision*. 24, 65

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14, 29

Carceroni, R. L. and Kutulakos, K. N. (2002). Multi-view scene capture by surfel sampling: From video streams to non-rigid 3D motion, shape & reflectance. *International Journal of Computer Vision*. 45

Carreira, J., Caseiro, R., Batista, J., and Sminchisescu, C. (2012). Semantic segmentation with second-order pooling. *Proceedings of the European Conference on Computer Vision*. 28, 83, 84

Carreira, J. and Sminchisescu, C. (2010). Constrained parametric min-cuts for automatic object segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 83, 93, 100

Casasent, D. and Psaltis, D. (1976). Position, rotation, and scale invariant optical correlation. *Applied Optics*. 20

Chatfield, K., Lempitsky, V., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. *Proceedings of the British Machine Vision Conference*. 2, 65

Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. *Proceedings of the British Machine Vision Conference*. 100

Chaudhry, R., Ravichandran, A., Hager, G., and Vidal, R. (2009). Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 45

Cheng, M.-M., Zhang, Z., Lin, W.-Y., and Torr, P. (2014). BING: Binarized Normed Gradients for objectness estimation at 300fps. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 100

Chetverikov, D. (1982). Experiments in the rotation-invariant texture discrimination using anisotropy features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 14

Ciresan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. *Proc. of Medical Image Computing and Computer Assisted Intervention*. 99

Cremers, D. (2006). Dynamical statistical shape priors for level set-based tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 83

Cremers, D., Tischhauser, F., Weickert, J., and Schnorr, C. (2002). Diffusion Snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal of Computer Vision*. 83

Crevier, D. (1994). *AI: The Tumultuous Search for Artificial Intelligence*. Basic Books. 1

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. *ECCV Workshop on statistical learning in computer vision*. 3, 15

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1, 8, 12, 24, 25, 38, 81, 85

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. *Proceedings of the European Conference on Computer Vision*. 26

Davis, J., Ramamoothi, R., and Rusinkiewicz, S. (2003). Spacetime stereo: A unifying framework for depth from triangulation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 44

Davis, L. (1981). Polarograms: A new tool for image texture analysis. *Pattern Recognition*. 14

Davis, L., Johns, S., and Aggarwal, J. (1979). Texture analysis using generalized cooccurrence matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14

Deriche, R. (1987). Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*. 21

Derpanis, K., Sizintsev, M., Cannons, K., and Wildes, R. (2010). Efficient action spotting based on a spacetime oriented structure representation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 43, 45

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3). 71, 90

Dollár, P. and Zitnick, C. L. (2013). Structured forests for fast edge detection. *Proceedings of the International Conference on Computer Vision*. 8, 12, 28, 31, 65, 67, 68, 93

Dorko, G. and Schmid, C. (2003). Selection of scale-invariant parts for object class recognition. *Proceedings of the International Conference on Computer Vision*. 13

Edelman, S., Intrator, N., and Poggio, T. (1997). Complex cells and object recognition. *Proceedings of Neural Information Processing Systems*. 14

Endres, I. and Hoiem, D. (2014). Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 83

Erhan, D., Szegedy, C., Toshev, A., and Anguelov, D. (2014). Scalable object detection using deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 99

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338. 10, 90

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 13

Felsberg, M. and Sommer, G. (2001). The monogenic signal. *IEEE Transactions on Signal Processing*. 20

Felzenszwalb, P. F., Girshick, R. B., and McAllester, D. (2010a). Cascade object detection with deformable part models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2241–2248. 84

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010b). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9). 8, 12, 25, 38, 40, 80, 82, 84, 85

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*. 93

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79. 38

Fergus, R., Perona, P., and A.Zisserman (2003). Object class recognition by unsupervised scale-invariant learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 13

Ferrari, V., Tuytelaars, T., and Van Gool, L. (2004). Simultaneous object recognition and segmentation by image exploration. *Proceedings of the European Conference on Computer Vision*. 13

Fidler, S., Mottaghi, R., Yuille, A., and Urtasun, R. (2013). Bottom-up segmentation for top-down detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 81, 82, 83, 84, 93

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. 3

Fragkiadaki, K., Hu, H., and Shi, J. (2013). Pose from flow and flow from pose. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 27

Fragkiadaki, K., Zhang, G., and Shi, J. (2012). Video segmentation by tracing discontinuities in a trajectory embedding. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 27

Freeman, W. and Adelson, E. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 21

Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *Computational learning theory*. 25

Fusiello, A., Trucco, E., and Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12:16–22. 53

Gao, T., Packer, B., and Koller, D. (2011). A segmentation-aware object detection model with occlusion handling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 80, 82, 83

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*. 27

Geusebroek, J., Smeulders, A., and van de Weijer, J. (2003). Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12(8):938–943. 21

Gibson, J. (1950). *The Perception of the Visual World*. Houghton Mifflin. 26

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 99, 100

Girshick, R. B. (2012). *From rigid templates to grammars: Object detection with structured models*. PhD thesis, University of Chicago. 38

Gkioxari, G., Hariharan, B., Girshick, R., and Malik, J. (2014). R-CNNs for pose estimation and action detection. *arXiv:1406.5212v1*. 99

Gould, S., Gao, T., and Koller, D. (2009). Region-based segmentation and object detection. *Proceedings of Neural Information Processing Systems*. 80, 83

Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 28

Gu, C., Lim, J. J., Arbeláez, P., and Malik, J. (2009). Recognition using regions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1030–1037. 83

Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. (2014). Simultaneous detection and segmentation. *Proceedings of the European Conference on Computer Vision*. 99

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. *Alvey vision conference*, 15:50. 2, 13

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press. 34

Hassner, T., Mayzels, V., and Zelnik-Manor, L. (2012). On SIFTS and their scales. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 12, 15, 22, 23, 66, 71, 72

He, K., Zhang, X., Ren, S., and Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *Proceedings of the European Conference on Computer Vision*. 99

Hoiem, D., Efros, A. A., and Hebert, M. (2007a). Recovering surface layout from an image. *International Journal of Computer Vision*. 98

Hoiem, D., Stein, A. N., Efros, A. A., and Hebert, M. (2007b). Recovering occlusion boundaries from a single image. *Proceedings of the International Conference on Computer Vision*. 98

Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*. 27

Hua, G., Brown, M., and Winder, S. (2007). Discriminant embedding for local image descriptors. *Proceedings of the International Conference on Computer Vision*. 23

Huguet, F. and Devernay, F. (2007). A variational method for scene flow estimation from stereo sequences. *Proceedings of the International Conference on Computer Vision*. 26, 45

Humayun, A., Li, F., and Rehg, J. M. (2014). RIGOR: Reusing inference in graph cuts for generating object regions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 67, 93

Jia, Y. (2013). Caffe: An open source convolutional architecture for fast feature embedding. `http://caffe.berkeleyvision.org`. 8

Johnson, A. and Hebert, M. (1997). Object recognition by matching oriented points. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 14

Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 14, 44, 65

Kim, J., Liu, C., Sha, F., and Grauman, K. (2013). Deformable spatial pyramid matching for fast dense correspondences. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 37

Kläser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3D-gradients. *Proceedings of the British Machine Vision Conference*. 43, 45

Klaus, B. and Horn, P. (1986). *Robot Vision*. MIT Press. 28

Kokkinos, I. (2011). Rapid deformable object detection using dual-tree branch-and-bound. *Proceedings of Neural Information Processing Systems*. 84, 86

Kokkinos, I. (2013). Shufflets: Shared mid-level parts for fast object detection. *Proceedings of the International Conference on Computer Vision*. 84, 86

Kokkinos, I., Bronstein, M., Littman, R., and Bronstein, A. (2012a). Intrinsic shape context descriptors for deformable shapes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 71

Kokkinos, I., Bronstein, M., and Yuille, A. (2012b). Dense scale-invariant descriptors for images and surfaces. *INRIA Research Report 7914*. 15, 20, 21, 41

Kokkinos, I. and Maragos, P. (2009). Synergy between image segmentation and object recognition using the expectation maximization algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 80, 83

Kokkinos, I. and Yuille, A. (2008). Scale invariance without scale selection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10, 11, 12, 15, 20, 66

Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 36, 75

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Proceedings of Neural Information Processing Systems*. 1, 99, 100

Kumar, M. P. and Koller, D. (2010). Efficiently selecting regions for scene understanding. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 83

Kumar, P., Torr, P. H., and Zisserman, A. (2010). Objcut: Efficient segmentation using top-down and bottom-up cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):530–545. 80, 83

Ladicky, L., Sturgess, P., Alahari, K., Russell, C., and Torr, P. H. S. (2010). What, Where and How Many? Combining object detectors and CRFs. *Proceedings of the European Conference on Computer Vision*. 83

Laptev, I. and Lindeberg, T. (2003). Space-time interest points. *Proceedings of the International Conference on Computer Vision*. 43, 45

Lazebnik, S., Schmid, C., and Ponce, J. (2003). Sparse texture representation using affine-invariant neighborhoods. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 13, 14, 18

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3, 15, 65

Lempitsky, V., Blake, A., and Rother, C. (2008). Image segmentation by Branch-and-Mincut. *Proceedings of the European Conference on Computer Vision*. 83

Leordeanu, M., Sukthankar, R., and Sminchisescu, C. (2012). Efficient closed-form solution to generalized boundary detection. *Proceedings of the European Conference on Computer Vision*. 8, 12, 28, 30, 64, 67, 68, 80, 84, 88, 93

Leordeanu, M., Sukthankar, R., and Sminchisescu, C. (2014). Generalized boundaries from multiple image interpretations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 12, 67

Lepetit, V. and Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 12

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. *Proceedings of the International Conference on Computer Vision*. 24

Lim, J. J., Zitnick, C. L., and Dollár, P. (2013). Sketch tokens: A learned mid-level representation for contour and object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 93

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*. 13

Liu, C. (2009). *Beyond pixels: Exploring new representations and applications for motion analysis*. PhD thesis, MIT. 46

Liu, C., Yuen, J., and Torralba, A. (2011). SIFT flow: dense correspondence across difference scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5). 12, 15, 27, 37, 70

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. 1, 8, 12, 13, 15, 41, 44, 65

Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the Mars Exploration Rovers. *Journal of Field Robotics*. 13

Maire, M., Arbeláez, P., Fowlkes, C., and Malik, J. (2008). Using contours to detect and localize junctions in natural images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 12, 28, 29, 30, 64, 67, 68, 88

Maire, M., Yu, S. X., and Perona, P. (2011). Object detection and segmentation from joint embedding of parts and pixels. *Proceedings of the International Conference on Computer Vision*, pages 2142–2149. 30, 80

Malisiewicz, T. and Efros, A. A. (2007). Improving spatial support for objects via multiple segmentations. *Proceedings of the British Machine Vision Conference*. 83

Manén, S., Guillaumin, M., and Gool, L. V. (2013). Prime object proposals with randomized Prim's algorithm. *Proceedings of the International Conference on Computer Vision*. 83, 100

Marr, D. (1982). *Vision. A computational investigation into the human representation and processing of visual information*. MIT Press. 7

Martin, D., Fowlkes, C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549. 29

Martin, J. and Crowley, J. L. (1995). Comparison of correlation techniques. *Conference on Intelligent Autonomous System*. 14

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide baseline stereo from maximally stable extremal regions. *Image and vision computing*. 13

Mikolajczyk, K. and Schmid, C. (2001). Indexing based on scale invariant interest points. *Proceedings of the International Conference on Computer Vision*. 13

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14, 23

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*. 13, 44, 65

Moreno-Noguer, F. (2011). Deformation and illumination invariant feature point descriptor. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 15, 44, 66

Moreno-Noguer, F., Lepetit, V., and Fua, P. (2007). Accurate non-iterative O(n) solution to the PnP problem. *Proceedings of the International Conference on Computer Vision*. 12

Moreno-Noguer, F., Sanfeliu, A., and Samaras, D. (2008). Dependent multiple cue integration for robust tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 83

Najman, L. and Schmitt, M. (1996). Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 28

Nowak, E., Jurie, F., and Triggs, B. (2006). Sampling strategies for bag-of-features image classification. *Proceedings of the European Conference on Computer Vision*. 15, 65

Ochs, P. and Brox, T. (2011). Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. *Proceedings of the International Conference on Computer Vision*. 27

Opelt, A., Fussenegger, M., Pinz, A., and Auer, P. (2004). Weak hypotheses and boosting for generic object detection and recognition. *Proceedings of the International Conference on Computer Vision*. 13

Ott, P. and Everingham, M. (2009). Implicit color segmentation features for pedestrian and object detection. *Proceedings of the International Conference on Computer Vision*. 66, 81, 83

Ott, P. and Everingham, M. (2011). Shared parts for deformable part-based models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 84

Özuysal, M., Calonder, M., Lepetit, V., and Fua, P. (2010). Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3). 65

Palou, G. and Salembier, P. (2014). Depth order estimation for video frames using motion occlusions. *IET Computer Vision*. 45

Pantofaru, C., Schmid, C., and Hebert, M. (2008). Object recognition by integrating multiple image segmentations. *Proceedings of the European Conference on Computer Vision*. 83

Pedersoli, M., Vedaldi, A., and Gonzalez, J. (2011). A coarse-to-fine approach for fast deformable object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1353–1360. 84

Perona, P. and Malik, J. (1990). Detecting and localizing edges composed of steps, peaks and roofs. *Proceedings of the International Conference on Computer Vision*. 29

Pirsiavash, H. and Ramanan, D. (2012). Steerable part models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 84

Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., and Koch, R. (2004). Visual modeling with a hand-held camera. *International Journal of Computer Vision*. 13

Ramanan, D. (2007). Using segmentation to verify object hypotheses. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 83

Randen, T. and Husoy, J. H. (1999). Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14

Rodriguez, M., Ahmed, J., and Shah, M. (2008). Action MACH: A spatio-temporal maximum average correlation height filter for action recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 45

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. *Proceedings of the European Conference on Computer Vision*. 13

Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *Proceedings of ACM SIGGRAPH*. 93

Rousson, M. and Paragios, N. (2002). Shape priors for level set representations. *Proceedings of the European Conference on Computer Vision*. 83

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. R. (2011). ORB: An efficient alternative to SIFT or SURF. *Proceedings of the International Conference on Computer Vision*. 24, 65

Rumelhart, D. E., McClelland, J. L., and PDP Research Group (1986). *Parallel Distributed Processing, Volume 1*. MIT Press. 9

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2014). Imagenet large scale visual recognition challenge. *arXiv:1409.0575*. 99

Russakovsky, O. and Ng, A. Y. (2010). A Steiner tree approach to object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 83

Russell, B. C., Freeman, W. T., Efros, A. A., Sivic, J., and Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 83

Salembier, P. and Garrido, L. (2000). Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing*. 28

Savalle, P.-A., Tsogkas, S., Papandreou, G., and Kokkinos, I. (2014). Deformable part models with CNN features. *Parts and Attributes Workshop (ECCV)*. 100

Schaffalitzky, F. and Zisserman, A. (2002). Multi-view matching for unordered image sets. *Proceedings of the European Conference on Computer Vision*. 12

Schmidt, U. and Roth, S. (2012). Learning rotation-aware features: From invariant priors to equivariant descriptors. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 66

Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional SIFT descriptor and its application to action recognition. *Proceedings of ACM Multimedia*. 43

Se, S., Lowe, D., and Little, J. (2002). Global localization using distinctive visual features. *Proceedings of the International Conference on Intelligent Robots and Systems*. 13

Shi, J. and Malik, J. (1997). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 29, 30, 31, 69

Shotton, J., Johnson, M., and Cipolla, R. (2006). Semantic texton forests for image categorization and segmentation. *Proceedings of the European Conference on Computer Vision*. 83

Simonyan, K., Vedaldi, A., and Zisserman, A. (2012). Descriptor learning using convex optimisation. *Proceedings of the European Conference on Computer Vision*. 65

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24, 65

Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. *Proceedings of the International Conference on Computer Vision*. 13

Sizintsev, M. and Wildes, R. (2009). Spatiotemporal stereo via spatiotemporal quadric element (stequel) matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11, 12, 18, 43, 45, 51, 54, 56, 59

Sizintsev, M. and Wildes, R. P. (2012). Spatiotemporal stereo and scene flow via stequel matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 53

Smith, S. M. and Brady, J. M. (1997). SUSAN—a new approach to low level image processing. *International Journal of Computer Vision*. 13

Snavely, N., Seitz, S., and Szeliski, R. (2006). Phototourism: Exploring photo collections in 3D. *Proceedings of ACM SIGGRAPH*. 13

Stein, A. and Hebert, M. (2005). Incorporating background invariance into feature-based object recognition. *Workshop on Applications of Computer Vision*. 20, 84

Strecha, C., Bronstein, A. M., Bronstein, M. M., and Fua, P. (2012). LDA-hash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1). 24, 65, 78

Strecha, C., Tuytelaars, T., and Gool, L. V. (2003). Dense matching of multiple wide-baseline views. *Proceedings of the International Conference on Computer Vision*. 33, 44

Strecha, C., von Hansen, W., Gool, L. V., Fua, P., and Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 34, 75

Sun, D., Roth, S., and Black, M. J. (2010). Secrets of optical flow estimation and their principles. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 27

Sun, D., Roth, S., and Black, M. J. (2014). A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*. 27

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. *arXiv:1312.6199*. 100

Szeliski, R. (2004). Image alignment and stitching: A tutorial. Technical report, Microsoft Research, Tech. Rep. MSR-TR-2004-92. 13

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 99

Tang, M., Gorelick, L., Veksler, O., and Boykov, Y. (2013). Grabcut in one cut. *Proceedings of the International Conference on Computer Vision*. 93

Tola, E., Lepetit, V., and Fua, P. (2008). A fast local descriptor for dense matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 81

Tola, E., Lepetit, V., and Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5). 11, 12, 14, 15, 16, 17, 22, 41, 44, 46, 49, 50, 51, 53, 55, 63, 66, 75, 76, 77, 78, 84, 85

Toshev, A. and Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 99

Tron, R. and Vidal, R. (2007). Hopkins 155 dataset. `http://www.vision.jhu.edu/data.htm`. 70

Trulls, E., Kokkinos, I., Sanfeliu, A., and Moreno-Noguer, F. (2013). Dense segmentation-aware descriptors. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9, 36, 63, 66, 67, 79, 81, 82, 84, 85, 88, 89

Trulls, E., Sanfeliu, A., and Moreno-Noguer, F. (2012). Spatiotemporal descriptor for wide-baseline stereo reconstruction of non-rigid and ambiguous scenes. *Proceedings of the European Conference on Computer Vision*. 9, 36, 41

Trulls, E., Tsogkas, S., Kokkinos, I., Sanfeliu, A., and Moreno-Noguer, F. (2014). Segmentation-aware deformable part models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9, 79

Trzcinski, T., Christoudias, M., Fua, P., and Lepetit, V. (2013). Boosting binary keypoint descriptors. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 24

Tsogkas, S. and Kokkinos, I. (2012). Learning-based symmetry detection in natural images. *Proceedings of the European Conference on Computer Vision*. 84

Tu, Z. W., Chen, X., Yuille, A., and Zhu, S. C. (2003). Image parsing: Unifying segmentation, detection, and recognition. *Proceedings of the International Conference on Computer Vision*. 83

Tuytelaars, T. and Schmid, C. (2007). Vector quantizing feature space with a regular lattice. *Proceedings of the International Conference on Computer Vision*. 23

Tuytelaars, T. and Van Gool, L. (2004). Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*. 12

Uijlings, J., van de Sande, K., Gevers, T., and Smeulders, A. (2013). Selective search for object recognition. *International Journal of Computer Vision*. 8, 83

van de Sande, K., Uijlings, J., and Sebe, N. (2011). Segmentation as selective search for object recognition. *Proceedings of the International Conference on Computer Vision*. 100

Vedaldi, A. and Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org. 2, 14, 15, 31, 41, 71, 85

Vedaldi, A. and Zisserman, A. (2009). Structured output regression for detection with partial truncation. *Proceedings of Neural Information Processing Systems*. 83

Vedula, S., Baker, S., Rander, P., Collins, R., and Kanade, T. (1999). Three-dimensional scene flow. *Proceedings of the International Conference on Computer Vision*. 26

Vicente, S., Carreira, J., Agapito, L., and Batista, J. (2014). Reconstructing PASCAL VOC. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1:I–511. 25

Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., and Cremers, D. (2008). Efficient dense scene flow from sparse or dense stereo data. *Proceedings of the European Conference on Computer Vision*. 26, 45, 61

Weiss, D. and Taskar, B. (2013). SCALPEL: Segmentation Cascades with Localized Priors and Efficient Learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 83, 100

Weiss, Y., Torralba, A., and Fergus, R. (2008). Spectral hashing. *Proceedings of Neural Information Processing Systems*. 24

Winder, S., Hua, G., and Brown, M. (2009). Picking the best daisy. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 23, 65

Winder, S. A. J. and Brown, M. (2007). Learning local image descriptors. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 18

Wolberg, G. and Zokai, S. (2000). Robust image registration using log-polar transform. *IEEE International Conference on Image Processing*. 20

Yang, H., Lin, W.-Y., and Lu, J. (2014). DAISY Filter Flow: A generalized discrete approach to dense correspondences. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 66

Yao, J. and Cham, W. (2006). 3D modeling and rendering from multiple wide baseline images. *Signal Processing: Image Communication*. 33

Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. *Proceedings of the European Conference on Computer Vision*. 14

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. *Proceedings of the European Conference on Computer Vision*. 100

Zhang, L., Curless, B., and Seitz, S. M. (2003). Spacetime stereo: Shape recovery for dynamic scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 44, 45

Zhang, N., Donahue, J., Girshick, R., and Darrell, T. (2014a). Part-based R-CNNs for fine-grained category detection. *Proceedings of the European Conference on Computer Vision*. 100

Zhang, N., Paluri, M., Ranzato, M., Darrell, T., and Bourdev, L. (2014b). PANDA: Pose Aligned Networks for Deep Attribute Modeling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 99

Zhu, Q., Avidan, S., Yeh, M.-C., and Cheng, K.-T. (2006). Fast human detection using a cascade of histograms of oriented gradients. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 25

Zitnick, C. L. and Dollár, P. (2014). Edge boxes: Locating object proposals from edges. *Proceedings of the European Conference on Computer Vision*. 100

List of Figures

1.1 Feature descriptors in computer vision . . . 2
1.2 Matching with complex ambiguities . . . 4
1.3 Matching with background interference . . . 5
1.4 Matching with occlusions . . . 6
1.5 Features and inferences in computer vision . . . 8

2.1 Building a SIFT descriptor from image gradients . . . 16
2.2 Building a Daisy descriptor from image gradients . . . 17
2.3 The Spatiotemporal Quadric Element (Stequel) . . . 18
2.4 Building invariant representations with the Fourier Modulus Transform technique . . . 19
2.5 Visualization of dense SID . . . 21
2.6 SIFT behaviour across scales . . . 23
2.7 HOG features and trained detectors . . . 25
2.8 Optical flow in computer vision . . . 26
2.9 Flow field color coding and examples . . . 27
2.10 Building the posterior probability of boundary Pb . . . 29
2.11 Segmentation cues used in this thesis . . . 32
2.12 Narrow vs. wide baseline stereo . . . 33
2.13 Epipolar geometry . . . 34
2.14 Graph cuts for binocular stereo . . . 35
2.15 Image registration with SIFT-flow . . . 36
2.16 The SIFT-flow coarse-to-fine matching strategy . . . 37
2.17 The DPM detection pipeline . . . 38
2.18 DPM filter for the 'horse' category . . . 39

3.1 The motivation behind our flow-based approach . . . 42
3.2 A highly-ambiguous, wide-baseline stereo sequence . . . 43
3.3 Building the spatiotemporal descriptor with flow priors . . . 46
3.4 Enforcing spatiotemporal consistency with a global regularization . . . 48
3.5 Spatial masks for occlusions . . . 50
3.6 Refining stereo maps with binary occlusion masks . . . 51
3.7 Samples from our synthetic wide-baseline dataset . . . 53

3.8 Results for the baseline experiments: 'gravel' 55
3.9 Results for the baseline experiments: 'flowers' 56
3.10 Results for the image noise experiments 57
3.11 Benchmarking the spatiotemporal descriptor against occlusions 58
3.12 Qualitative results under complex ambiguities (1/2) 59
3.13 Qualitative results under complex ambiguities (2/2) 60

4.1 Building background-invariant descriptors with soft segmentation masks. . . 64
4.2 Segmentation-aware descriptor construction 68
4.3 Overlap results over the MOSEG dataset for all descriptors. 72
4.4 Large displacement matching with SIFT-flow using different descriptors . . 73
4.5 Overlap results over the MOSEG dataset for DSIFT and SDSIFT 74
4.6 Increase in average overlap for SDSIFT over DSIFT 74
4.7 Accuracy at different baselines 75
4.8 Quantitative comparison of iterative Daisy stereo with our single-shot approach 76
4.9 Qualitative comparison of iterative Daisy stereo with our single-shot approach 77

5.1 Segmentation masks for general object recognition 80
5.2 Soft segmentation masks for sliding-window detectors 81
5.3 Ranking the SLIC superpixels given a bounding box 87
5.4 Soft masks over different scales and object categories 88
5.5 Segmentation-aware DSIFT with SLIC-based masks 89
5.6 Precision/Recall curves on the PASCAL VOC 2007 92
5.7 AP increase between DPM and our segmentation-sensitive DPM 93
5.8 Qualitative detection results with the segmentation-aware DPM filters . . . 94
5.9 Performance of SIFT and SIFT with SLIC-based masks on MOSEG 95

6.1 Artificial Neural Networks for visual recognition 100

List of Tables

2.1 Index of the technologies used in this thesis 12

5.1 AP performance on the PASCAL VOC 2007 91