Video Object Segmentation

Author(s):
Perazzi, Federico

Publication Date:
2017

Permanent Link:

https://doi.org/10.3929/ethz-b-000184917 →

Diss. ETH No. 24353

Video Object Segmentation

A dissertation submitted to
ETH Zurich

for the Degree of
Doctor of Sciences

presented by
Federico Perazzi
MSc in Computer Science, ETH Zurich, Switzerland
born May 31, 1983
citizen of the Italian Republic

accepted on the recommendation of
Prof. Dr. Markus Gross, examiner
Dr. Alexander Sorkine-Hornung, co-examiner
Prof. Dr. Alexei Efros, co-examiner

2017

Abstract

The detection and segmentation of foreground objects in videos is a fundamental problem in computer vision research and a key component for a wide array of applications. Ranging from higher-level vision problems such as semantic scene understanding and video summarization, to low-level video post-production and editing tools, video segmentation encompass the entire spectrum of video related tasks. This diverse set of applications yields different objectives and impose different requirements in terms of quality, efficiency, and manual effort necessary.

This thesis investigates novel video object segmentation techniques, spanning different types of end applications. First, we study the problem of reducing or eliminating human effort to enable unsupervised segmentation of videos by proposing approaches to roughly estimate the primary or "salient" object. Next, we explore methods that operate in a semi-automatic fashion, *i.e*. with minimal human supervision, and methods that enable user control and interaction. Finally, we introduce a new dataset and evaluation methodology to enable a deeper understanding of the results and to point towards promising avenue for future research.

The first part of the thesis addresses the problem of discovering salient objects in still images. Motivated by psychological and neurobiological studies, we tackle the problem from complementary perspectives. On one hand, we combine, in a single-high dimensional Gaussian filtering framework, color contrast and spatial color distribution. On the other hand, we exploit spectral clustering properties to model common rules of photographic composition. These two approaches are orthogonal to each other and instrumental to unsupervised video object segmentation.

The second part of the thesis explores semi-automatic video segmentation techniques with different type of annotations and therefore different levels of human supervision and suitable applications. We demonstrate robustness to challenging situations such as occlusions, by estimating the maximum-a-posteriory of a fully connected graphical model built over object proposals. We leverage the discriminative power of fully convolutional networks trained on static images and initialized with precise segmentation masks or bounding-boxes.

The third part of the thesis is related to interactive segmentation techniques. In this domain, responsiveness is crucial to enable user interaction. Therefore we

propose to perform the segmentation on a sparse and regularly sampled data structure known as bilateral grid to provide iterative feedback in a fraction of the time of the previous approaches. Results demonstrates that the proposed approach is not only suitable for interactive segmentation but it is also able to generate high-quality results in semi-automatic settings, without any type of user interaction.

The fourth and concluding part of the thesis, introduces a new dataset and evaluation methodology specifically designed for the problem of segmenting foreground objects in videos. We analyze several state-of-the-art segmentation approaches as well as those proposed in this thesis to uncover their strengths and weaknesses and highlight promising directions for future works.

The novel approaches that will be presented in this thesis enabled improvements upon the state-of-the-art both in terms of accuracy and efficiency. Furthermore, the knowlegdge collected during the aforementioned studies has lead to the organization of the first workshop on video object segmentation that will be held at the Computer Vision and Pattern Recognition conference in 2017.

Sommario

L'identificazione e la segmentazione di oggetti nei video è un problema fondamentale nella ricerca in computer vision e componente chiave per una vasta gamma di applicazioni. A partire dai problemi di visione a livello superiore, come la comprensione semantica delle immagini e la classificazione dei video, per arrivare agli strumenti di video editing per la post-produzione, la segmentazione è uno strumento utilizzato nell'intero spettro delle applicazioni relative al video. Questa varietà di applicazioni introduce diversi obiettivi e impone requisiti diversi in termini di qualità, efficienza e sforzo manuale.

Questa tesi esamina nuove tecniche di segmentazione di oggetti presenti in video, utili per diversi tipi di applicazioni finali. In primo luogo, studiamo il problema della riduzione o dell'eliminazione dello sforzo umano per consentire la segmentazione non monitorata dei video proponendo approcci per individuare approssimativamente la locazione dell'oggetto primario o "saliente". Quindi, esploriamo metodi che operano in modo semi-automatico, cioè con una minima supervisione umana e metodi che consentono il controllo e l'interazione dell'utente. Infine, introduciamo un nuovo set di dati e una metodologia di valutazione per consentire una comprensione profonda dei risultati e delineare nuovi percorsi di ricerca futura.

La prima parte della tesi esamina il problema di scoprire oggetti salienti nelle immagini statiche. Motivati da studi psicologici e neurobiologici, affrontiamo il problema da prospettive complementari. Da una parte, combiniamo in un singolo framework di high-dimensional Gaussian filtering, il contrasto di colore e la distribuzione loro distribuzione all'interno dell'immagine. Dall'altra parte, sfruttiamo le proprietà di clustering spettrale per modellare regole comuni di composizione fotografica. Questi due approci sono ortogonali l'uno all'altro e necessari per lo sviluppo di tecniche di video segmentazione degli oggetti non supervisionata.

La seconda parte della tesi esplora tecniche semi-automatiche di segmentazione video con diversi tipi di annotazioni e quindi diversi livelli di supervisione umana e applicazioni. Qui dimostriamo robustezza a situazioni impegnative quali occlusioni, valutando la stima del massimo-a-posteriori di un modello grafico completamente connesso costruito su proposte di oggetti. Inoltre sfruttiamo il potere di-

scriminatorio di reti convoluzionali esercitate su immagini statiche e inizializzate con precise maschere di segmentazione o bounding-boxes.

La terza parte della tesi è legata a tecniche di segmentazione interattiva. In questo dominio, la velocità di risposta dell'algoritmo è cruciale per consentire l'interazione dell'utente. Pertanto proponiamo di eseguire la segmentazione su una struttura di dati efficiente, conosciuta come griglia bilaterale per fornire un feedback iterativo in una frazione del tempo degli approcci precedenti. I risultati dimostrano che l'approccio proposto non è solo adatto alla segmentazione interattiva, ma è anche in grado di generare risultati di alta qualità in impostazioni semi-automatiche, senza alcun tipo di interazione tra utenti.

La quarta e conclusiva parte della tesi, introduce un nuovo set di dati e una metodologia di valutazione specificatamente progettata per il problema di segmentazione di oggetti nei video. Analizziamo diversi approcci di segmentazione oltre a quelli proposti in questa tesi per scoprire i loro punti di forza e debolezza e mettere in evidenza le direzioni promettenti per la ricerca futura.

Gli algoritmi presentati in questa tesi hanno consentito miglioramenti dello stato dell'arte sia in termini di precisione che di efficienza. Inoltre, la conoscenza raccolta durante i suddetti studi ha portato all'organizzazione del primo workshop sulla segmentazione degli oggetti video che si terrà alla conferenza Computer Vision e Pattern Recognition nel 2017.

Acknowledgments

I would like to express my gratitude to the members of my examination committee, Prof. Dr. Markus Gross, Prof. Dr. Alexei Efros and Dr. Alexander Sorkine-Hornung.

I'm grateful to my advisor at Disney Research, Dr. Alexander Sorkine-Hornung for his guidance and support in overcoming the challenges I have been facing throughout my research studies. Furthermore, I would like to thank my fellow doctoral students for their feedback, cooperation and of course friendship.

Finally, I express my gratitude to my parents and to my sister for providing me with unconditioned support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them.

Contents

Abstract iii

Sommario v

Acknowledgements vii

Contents ix

Introduction 1
1.1 Contribution and Organization 3
1.2 Publications 5

Related Work 7
2.1 Salient Object Detection 7
2.2 Unsupervised Video Segmentation 11
2.2.1 Over-segmentation 12
2.2.2 Proposals-based Segmentation 12
2.2.3 Motion Segmentation 13
2.3 Semi-automatic Video Segmentation 13
2.3.1 Bounding-Box Tracking and Segmentation 14
2.3.2 Graph Based Video Segmentation 14
2.4 Interactive Video Segmentation 15
2.5 Datasets 16

Salient Object Detection 19
3.1 Saliency Filters 20
3.1.1 Method 21
3.2 Saliency Detection using Fiedler Vectors 27
3.2.1 Method 28
3.3 Results 31
3.3.1 Precision and Recall 32
3.3.2 Mean Absolute Error 33
3.4 Discussion 34

Semi-automatic Segmentation with Object Proposals **39**
4.1 Method . . . 40
4.1.1 Object Proposal Generation . . . 40
4.1.2 Fully Connected Proposal Labeling . . . 43
4.2 Implementation Details . . . 47
4.3 Results . . . 47
4.4 Discussion . . . 50

Learning Video Segmentation from Static Images **53**
5.1 Method . . . 54
5.1.1 Offline Training . . . 55
5.1.2 Online Training . . . 56
5.1.3 Variants . . . 57
5.2 Network implementation and training . . . 58
5.3 Results . . . 59
5.3.1 Experimental setup . . . 60
5.3.2 Ablation study . . . 60
5.3.3 Evaluation . . . 62
5.4 Conclusion . . . 64

Interactive Segmentation in Bilateral Space **67**
6.1 Method Overview . . . 68
6.1.1 Lifting . . . 68
6.1.2 Splatting . . . 69
6.1.3 Graph Cut . . . 70
6.1.4 Slicing . . . 72
6.2 Results . . . 73
6.2.1 Quantitative Evaluation . . . 74
6.2.2 Interactive Segmentation . . . 77
6.3 Discussion . . . 78

Dataset and Evaluation Methodology **79**
7.1 Dataset Description . . . 81
7.2 Evaluated Algorithms . . . 83
7.3 Experimental Validation . . . 85
7.3.1 Metrics Selection . . . 85
7.3.2 Metrics Validation . . . 87
7.4 Quantitative Evaluation . . . 88
7.4.1 Error Measure Statistics . . . 89
7.4.2 Attributes-based Evaluation . . . 91
7.5 Attributes Dependency . . . 94
7.6 Discussion . . . 95

Conclusion **103**
8.1 Future Works. 106

References **109**

CHAPTER 1

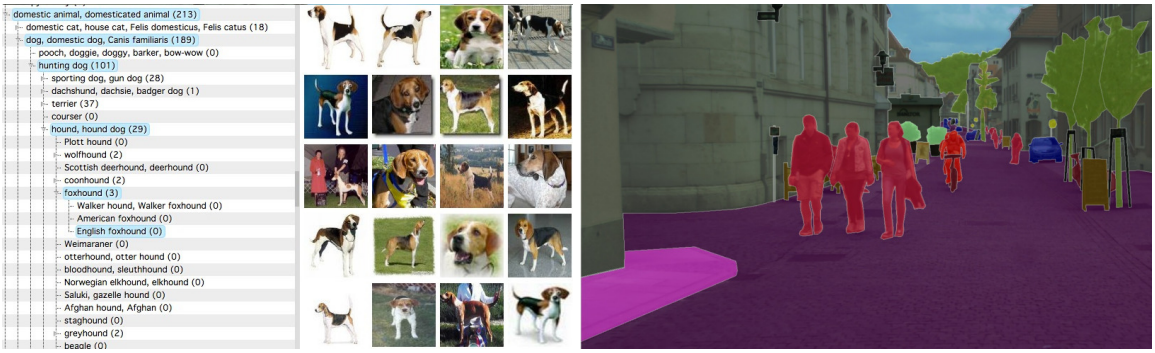
Introduction

A massive amount of video data is generated everyday by millions of people around the world and made publicly available on the Internet. This large amount of visual information is generally associated by users with labels to identify content and location. While these noisy labels represent a form of weak annotations useful for some supervised machine learning tasks such as scene classification and action recognition they do not provide enough context to leverage the rich spatio-temporal signal represented by videos.

At the other end of the spectrum, further away from noisy scene classification labels, lie dense, per-pixel accurate, manual annotations of videos, (Figure 1.1). This type of annotation enables a deeper level of visual scene understanding which is required, for example, in the context of self-driving cars, and video surveillance. Besides, visual understanding, pixel-wise annotations are ubiquitous in the media content post-production pipeline enabling independent processing of different image regions.

However, dense per-pixel annotations are tedious to obtain. Depending on the complexity of the scene, a trained human can process on average between 5 to 15 frames per day [Cordts et al., 2016]. Dense per-pixel video labeling, therefore, represents a significant investment both in terms of time and money and therefore large-scale datasets with per-pixel annotations are scarce.

As demonstrated by recent success on the tasks of object recognition and detection in still images, large-scale datasets are of fundamental importance to enable fast-paced progress in computer vision [Lin et al., 2014; Russakovsky et al., 2014; Torralba and Efros, 2011]. Thus, it is not a surprise



Classification Semantic Segmentation

Figure 1.1: *Different type of image annotations. Left: image is associated to weak labels specifying type of object and other semantic attributes. Right: dense per-pixel semantic labeling. Every pixel is associated with the object class it belongs to. Sources: ImageNet (www.image-net.org) and CityScapes (https://www.cityscapes-dataset.com).*

that one of the trend topics currently being investigated by the computer vision community is that of developing novel learning techniques to better exploit the information that a video source provides, while reducing or eliminating the manual effort required to manually label individual pixels.

One branch of computer vision that is related to the task above is know as video segmentation. Video segmentation refers to a broad range of computer vision techniques aiming to group perceptually or semantically similar regions in videos. This grouping enables the propagation of spatially dense, but temporally sparse labels (Figure 1.2) along successive video frames, reducing the amount of manual labour required to densely annotate a video.

Based on the type of grouping, video segmentation algorithms can be broadly classified into over-segmentation and object segmentation. While the former aims to group perceptually similar compact regions of a video, the latter aims to congregate pixels belonging to the same object instance. This thesis focuses on *video object segmentation*.

Besides low level tasks such as label propagation and video analysis, video object segmentation is instrumental for many high-level applications related to media content production. In particular video object segmentation is essential to special effect post-production. Complex editing, such as compositing, requires independent processing of several elements of a scene and video segmentation tools can help the artists to speed up their work flow. However, despite remarkable progress in recent years, video object segmentation still remains a challenging problem and most existing approaches still

Ground-truth Frame 16 Frame 38 Frame 69

Figure 1.2: *Sparse label propagation. The first frame is human annotated and it serves the purpose of initializating the algorithm that propagates the annotations forward to successive frames. Source: DAVIS (davischallenge.org).*

exhibit too severe limitations in terms of quality and efficiency to be applicable in practical applications, such as video post-production and editing in the visual effects industry.

In this thesis we investigate different statistical approaches to perform the task of video object segmentation while spanning different degrees of human supervision. First, we study low-level techniques to distinguish salient object from background regions as a form of bootstrapping for automatic video object segmentation algorithms. Next, we explore the usage of rough annotations such as bounding boxes or object proposal, *i.e.* regions that are likely to contain an objects, to initialize our algorithms in a semi-automatic fashion. Finally, we create a new dataset and propose a evaluation methodology that take into account three essential factors to assess the quality of the segmentation, namely region similarity, contour accuracy and temporal stability.

1.1 Contribution and Organization

This thesis aims to advance the field of video object segmentation proposing a new evaluation methodology and novel image based techniques for discovering and segmenting objects in videos. The main thread, delineating the structure of the thesis is the increasing amount of supervision, or human effort, required to perform the segmentation. Briefly, methods can be categorized as *unsupervised*, *semi-automatic* or *interactive*, depending whether they discover the object to segment without any human supervision, use few manually annotated frames as initialization, or allow user to repeatedly interact with them to provide feedback and improve the segmentation results. Note that the boundaries between these categories are fuzzy and several approaches are designed to operate in different modalities. In detail the thesis is structured as follows.

In *Chapter 2* we review the literature that is most closely related to this thesis.

The section begins with a selection of salient detection approaches which are often used to roughly locate the object to segment. Next we describe several state-of-the-art video object segmentation algorithms. Reflecting the overall structure of the thesis, they are grouped based on the amount and type of labeling. The section closes with an overview of existing datasets, which commonly used to benchmark the performance of video object segmentation algorithms.

Chapter 3 is related to unsupervised video segmentation. We introduce two different approaches to discover salient foreground objects in still images and videos. These approaches are instrumental to replace human annotations with a rough object localization. The first method implements the notion of color-contrast efficiently using high-dimensional gaussian filters. The second approach is based on the assumption that most of the image boundaries are non-salient and exploits known properties of the *Fiedler vector* to infer the saliency.

Chapter 4 we present a novel approach to perform video segmentation which is well suited to employ the saliency algorithms presented in the previous chapter, in order to operate in an unsupervised fashion. Our proposed technique exploits a fully connected spatiotemporal graph built over object proposals *i.e*. regions of an image that are likely to contain an object. The problem is formulated as a minimization of a novel energy function that combines appearance with long-range point tracks to ensure robustness to challenging situations such as occlusions.

In *Chapter 5* we investigate the usage of different types of manual annotations such as segments and bounding-boxes and propose a convolutional neural network (ConvNet) based, semi-supervised approach for video object segmentation. We couple the discriminative power of deep neural networks with an external guidance given in the form of a human annotated segmentation mask or a bounding box and demonstrate that highly accurate object segmentation in videos can be enabled by using a ConvNet trained with static images only. The novel idea of our approach is a combination of offline and online learning strategies, where the former serves the purpose of localizing the object from the previous frame estimate and the latter allows to capture the appearance of the specific object instance.

In *Chapter 6* we present an interactive approach to video segmentation that operates in bilateral space. This method enables near real-time user interaction and it is suitable for post-production applications that require higher level of accuracy. We design a new energy on the vertices of a regularly sampled spatio-temporal bilateral grid, which can be solved efficiently using a standard graph cut label assignment. Our formulation implicitly approx-

imates long-range, spatio-temporal connections between pixels while still containing only a small number of graph nodes and only local edges, yielding a method that is both efficient and robust to several challenging situations such as occlusions, appearance changes and non-linear deformations.

In *Chapter 7* we introduce a new dataset specifically designed for the task of video object segmentation. The dataset contains professionally annotated video sequences which have been carefully captured to cover multiple instances of major challenges typically faced in video object segmentation. The dataset is accompanied with a comprehensive evaluation of several state-of-the-art approaches. A series of attributes such as occlusions, fast-motion, non-linear deformation and motion-blur are associated to each video and evaluated independently enabling a deeper understanding of the results and pointing towards promising avenues for future research.

Chapter 8 concludes the thesis, summarizes its main contributions.

1.2 Publications

The technical contributions have led to top-tier conference publications and a Computer Vision and Pattern Recognition (CVPR) Workshop on video object segmentation.

- Saliency Filters: Contrast Based Filtering for Salient Region Detection, *F. Perazzi, P. Krähenbühl, Y. Pritch and A. Sorkine-Hornung, CVPR 2016, Providence, Rhode Island, USA.* (Chapter 3).
- Efficient Salient Foreground Detection for Images and Video using Fiedler Vectors, *F. Perazzi, O. Sorkine-Hornung, A. Sorkine-Hornung Eurographics Workshop on Intelligent Cinematography and Editing 2014, Zurich, Switzerland.* (Chapter 3).
- Fully Connected Object Proposals for Video Segmentation, *F. Perazzi, O. Wang, M. Gross, A. Sorkine-Hornung, ICCV 2015, Santiago, Chile.* (Chapter 4).
- Bilateral Space Video Segmentation, *N. Märki, F. Perazzi, O. Wang, A. Sorkine-Hornung CVPR 2016, Las Vegas, USA.* (Chapter 6).
- A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation, *F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, A. Sorkine-Hornung CVPR 2016, Las Vegas, USA.* (Chapter 7).

- Learning Video Object Segmentation From Static Images, *F. Perazzi*, A. Khoreva*, R. Benenson, B. Schiele, A. Sorkine-Hornung, CVPR 2017, Hawaii, USA.* (Chapter 5).

Although not relevant to the scope of this thesis, the following two conference paper were published during my PhD studies:

- Non-Polynomial Galerkin Projection on Deforming Meshes, *M. Stanton, Y. Sheng, M. Wicke, F. Perazzi, A. Yuen, S. Narasimhan, A. Treuille ACM Transactions on Graphics 32(4) - SIGGRAPH 2013.*
- Panoramic Video from Unstructured Camera Arrays, *F. Perazzi, A. Sorkine-Hornung, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, M. Gross Eurographics 2015, Computer Graphics Forum, Vol. 34, No. 2, Zurich, Switzerland.*

Motivated by the popularity gained by the DAVIS dataset and benchmark proposed in Chapter 7, we organized the First DAVIS Challenge for Video Object Segmentation. The objective is to promote and facilitate the development of research techniques aiming to separate foreground objects from background regions in video sequences.

- The DAVIS Challenge on Video Object Segmentation 2017, *J. Pont-Tuset, F. Perazzi, S. Caelles, A. Sorkine-Hornung, P. Arbeláez, L. Van Gool, CVPRW 2017, Hawaii , USA.*

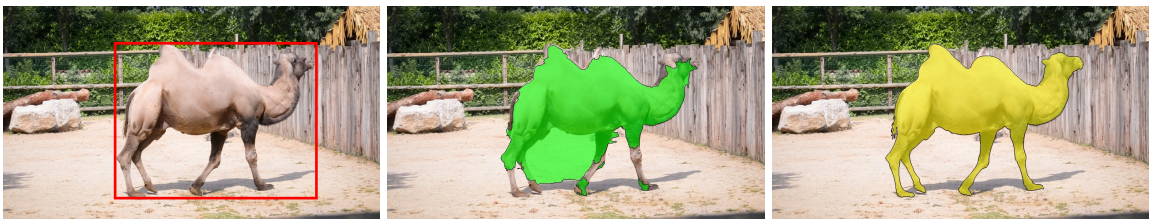
CHAPTER 2

Related Work

We categorize the body of literature related to this thesis based on the amount and type of annotations required. As briefly discussed in Chapter 1, video object segmentation approaches can be broadly classified as unsupervised, semi-automatic and interactive. Based on heuristics or supported by salient object detection mechanisms (§2.1), unsupervised video segmentation techniques (§2.2)) do not require any type of human supervision and instead discover the foreground object in a video sequences and proceed with the segmentation. In contrast semi-automatic approaches (§2.3) require some sort of human initialization. As discussed in details in Chapter 4 and Chapter 5 the level of supervision may vary and it can take the form of bounding-boxes, rough segmentations like object proposals, or ground-truth binary masks that precisely mark the object to be segmented in one or more video frames (Figure 2.1). Finally interactive approaches (§2.4) assume a human annotator in the cycle, such that the underlying algorithm can be guided towards the desired segmentation. The chapter concludes in Section 2.5 with an overview of existing dataset commonly used to benchmark video object segmentation algorithms.

2.1 Salient Object Detection

Automatic detection of salient image regions can alleviate, the tedious task of manually annotating video frames. Several video segmentation approaches have exploited saliency prior to initialize their segmentation in a fully automatic fashion [Papazoglou and Ferrari, 2013; Wang et al., 2015; Faktor and Irani, 2014]. Furthermore saliency detection is a useful tool with



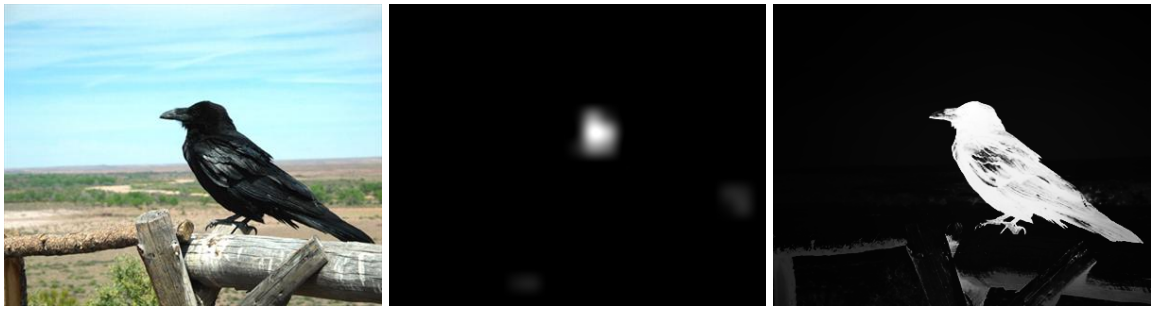
Bounding Box | Object Proposal | Segmentation Mask

Figure 2.1: *Types of input annotations. From left to right, bounding box, object proposal and accurate segmentation mask.*

applications in intelligent camera control, surveillance, video summarization and editing.

The pre-attentive human visual system is driven by bottomup, low-level stimuli such as color, contrast, orientation of edges, disparity and sudden movements. Depending on the nature of their features, methods that model bottom-up visual saliency can be categorized into biologically inspired or computationally based approaches, Figure 2.2. Works belonging to the first class [Itti et al., 1998; Harel et al., 2006] are generally based on the architecture proposed by Koch and Ullman [1985], in which the low-level stage processes features such as color, orientation of edges, or direction of movement. One implementation of this model is the work by Itti et al. [1998], which use a Difference of Gaussians approach to evaluate those features. However, as the evaluation by Cheng et al. [2011] shows, the resulting saliency maps are generally blurry, and often overemphasize small, purely local features, which renders this approach less useful for applications such as segmentation, detection, etc.

In contrast, computational methods may also be inspired by biological principles, but relate stronger to typical applications in computer vision and graphics. For example, frequency space methods [Hou and Zhang, 2007; Guo et al., 2008] determine saliency based on the amplitude or phase spectrum of the Fourier transform of an image. The resulting saliency maps better preserve the high level structure of an image than [Itti et al., 1998], but exhibit undesirable blurriness and tend to highlight object boundaries rather than its entire area. For colorspace techniques one can distinguish between approaches using local or global analysis of (color-) contrast. Local methods estimate the saliency of a particular image region based on immediate image neighborhoods, *e.g.*, based on dissimilarities at the pixel-level [Ma and Zhang, 2003], using multi-scale Difference of Gaussians [Itti and Baldi, 2005] or histogram analysis [Liu et al., 2007]. While such approaches are able to produce less blurry saliency maps, they are agnostic of global relations and



Input Gaze Prediction Salient Object Detection

Figure 2.2: *Difference between biologically inspired and computationally based saliency techniques. While the former aims to predict eye gaze, the latter aims to discover and uniformly highlight the most prominent object in the image.*

structures, and they may also be more sensitive to high frequency content like image edges and noise [Achanta et al., 2009].

Global methods take contrast relations over the complete image into account. For example, there are different variants of patch-based methods which estimate dissimilarity between image patches [Liu et al., 2007; Goferman et al., 2010; Wang et al., 2011]. While these algorithms are more consistent in terms of global image structures, they suffer from the involved combinatorial complexity, hence they are applicable only to relatively low resolution images, or they need to operate in spaces of reduced dimensionality [Duan et al., 2011], resulting in loss of small, potentially salient detail. The works of Singh et al. [2012] and Doersch et al. [2013] aim to extract a set of discriminative patches that occur frequently enough in images while being different from the other set of discriminative patches. While the final goal is not salient object detection, their output could be used a basis to compute patch-based global contrast.

The method of Achanta et al. [2009] also works on a per-pixel basis, but achieves globally more consistent results by computing color dissimilarities to the mean image color. They use Gaussian blur in order to decrease the influence of noise and high frequency patterns. However, their method does not account for any spatial relationship inside the image, and may highlight background regions as salient. Liu et al. [2007] combines multi-scale contrast, local contrast based on surrounding, context, and color spatial distribution to learn a Conditional Random Field (CRF) for binary saliency estimation. However, the significance of features in the CRF remains unclear. Ren et al. [2010] and Cheng et al. [2011] employ image segmentation as part of their saliency estimation. Ren et al. [2010] the segmentation solely to alleviate the negative influence of highly textured regions, noise and out-

liers during their subsequent clustering. Cheng et al. [2011], achieves high-quality results employing color dissimilarities between 3D color histogram bins. However, due to the use of larger-scale image segments in both approaches [Ren et al., 2010; Cheng et al., 2011], contrast measures involving spatial distribution cannot easily be formulated. Moreover, such methods have problems handling images with cluttered and textured background. Despite many improvements, the varying evaluation results in [Cheng et al., 2011] indicate that the actual significance of individual features and contrast measures in existing methods is difficult to assess. In Section 3.1 we propose to reduce the set of contrast measures to just two, namely, color uniqueness and distribution. These measures can be intuitively defined over abstract image elements, while still producing pixel-accurate saliency masks.

While contrast-based methods have proven to be very effective, their basic assumptions do not always hold. Therefore, research has also focused on additional visual cues. For example, Wei et al. [2012] note that image boundaries are most likely to be part of the background and introduce a measure of saliency based on the color-based geodesic distance between interior image regions and boundaries. Their method produces good results in high-recall areas, but it may suffer from non-smooth backgrounds, producing noisy saliency maps. Motivated by the same assumption of image boundaries being mostly non-salient, in Section 3.2 we propose an approach that leverage spectral clustering and effectively resolves the aforementioned issues of Wei et al. [2012] while producing more globally coherent saliency maps.

Since the publication of our studies [Perazzi et al., 2012; Perazzi et al., 2015a] our ideas have inspired several follow-up works. For example Cheng et al. [2013] improved our image abstraction with a Gaussian Mixture Model representation. Their formulation capture larger scale perceptually homogeneous elements, resulting in improved salient object region detection accuracy.

Recently, deep learning has re-defined the state-of-the-art of several visual tasks, ranging from scene classification [He et al., 2015] and object recognition [Ren et al., 2015] to inpainting [Pathak et al., 2016] and image colorization [Zhang et al., 2016]. Salient object detection is also taking advantage of the recent progresses and currently, most promising techniques are based on deep ConvNets. For instance, Li et al. [2015] extract deep features around multi-scale image regions and train a neural network regressor to determine their saliency score. Wang et al. [2015] train a deep neural network to learn local patch features and employ global contrast to determine the saliency value for each patch-centered image pixel. Similarly Zhao et al. [2015] inte-

Source Over Seg. Motion Seg. Object Seg.

Figure 2.3: *Different sub-tasks of video segmentation. From left to right, source image, over-segmentation of a video into supervoxels, motion segmentation, and video object segmentation.*

grate global and local context patches in a deep learning based pipeline. Li et al. [2016], observe that methods operating on patches tend to produce blurry saliency maps near the edges, therefore they propose a pixel-level fully convolutional stream paired with a segment-wise spatial pooling architecture that better models discontinuities along boundaries.

2.2 Unsupervised Video Segmentation

Unsupervised video object segmentation approaches extend the concept of salient object detection to videos [Papazoglou and Ferrari, 2013; Shen et al., 2015; Zhang et al., 2013; Taylor et al., 2015; Li et al., 2013]. They do not require any manual annotation and do not assume any prior information on the object to be segmented. Typically they are based on the assumption that object motion is dissimilar from the surroundings *i.e.* the motion is salient. To this end, Wang et al. [2015] use a saliency detector to locate the object and the geodesic between two superpixels on the image to compute a probability of a superpixel to belong to the foreground object. Instead, Faktor and Irani [2014] refine salient object detection using a Markov chain that fully connects the video frames. Besides using saliency, of some methods are based on object proposals and generate several ranked segmentation hypotheses [Lee et al., 2011; Zhang et al., 2013]. Unsupervised approaches are well suited for parsing large scale databases, they are bound to their underlying assumption and fail in cases it does not hold. While this thesis specifically address the topic of video object segmentation, unsupervised approaches have historically targeted over-segmentation, [Grundmann et al., 2010; Xu and Corso, 2012] or motion segmentation, [Brox and Malik, 2010; Fragkiadaki et al., 2012] and therefore this different domains will be briefly discussed in the following paragraphs. Visual output from the aforementioned sub-tasks, is shown in Figure 2.3. In Chapter 4 we employ sev-

eral strategies often used for unsupervised video segmentation. We formulate the problem of reducing overlapping segments into a foreground-background partition (§2.2.1) by minimizing a novel energy function which we solve optimally by inference on a fully connected Conditional Random Field (CRF). The fully connected graph is built over object proposals (§2.2.2). Furthermore, following a line of works (§2.2.3) aiming to segment coherent motion, we exploit point-tracks to increase stability of long term temporal connections.

2.2.1 Over-segmentation

Unconstrained motion can be handled by methods based on supervoxels [Grundmann et al., 2010; Xu and Corso, 2012; Hickson et al., 2014]. These methods generate an oversegmentation of the video into space-time homogeneous, perceptually distinct regions. They are important for early stage video preprocessing, but do not directly solve the problem of video object segmentation as they do not provide any principled approach to flatten the hierarchical decomposition of the video into a binary segmentation [Papazoglou and Ferrari, 2013].

2.2.2 Proposals-based Segmentation

Recent advances in state-of-the-art image analysis [Carreira et al., 2012; Girshick et al., 2014] have motivated the use of object proposals [Carreira and Sminchisescu, 2012; Endres and Hoiem, 2014; Arbeláez et al., 2014; Krähenbühl and Koltun, 2014] in video object segmentation. [Lee et al., 2011] discover clusters of key-segments in videos, coupling the notion of objectness and appearance similarity. Hypotheses are later ranked and the top scoring one is automatically selected for video segmentation. Their work is well suited to determine groups of segments with consistent appearance and motion, but disregards spatial and temporal relations between segments. Ma and Latecki [2012] account for these by imposing the selection of one proposal in every frame, formulating the problem as finding a maximum weighted clique in a locally connected graph with mutex constraints. However, the strict assumptions that the object should appear in every frame limits their efficacy in real world scenarios. Similar to Ma and Latecki [2012] , Zhang et al. [2013] create a layered *Directed Acyclic Graph* (DAG) which combines unary edges measuring the objectness of the object proposal and pairwise edges modeling affinities. A shortest path determines the video object segmentation. Both formulate the problem on a locally connected graph structure, requiring that objects appear in every frame.

2.2.3 Motion Segmentation

Significant progress has been achieved by methods designed to track keypoints over time and, more recently, over image regions [Brendel and Todorovic, 2009; Li et al., 2013; Varas and Marqués, 2014]. These methods, however, only consider two consecutive frames of video and are sensible to sudden motion and appearance changes (*i.e.* due to lighting). Related to tracking systems, Brox et al. [2010] propose an approach to segment motion by spectral clustering of long term point trajectories based on their motion affinity [Brox and Malik, 2010] and a variational approach [Ochs and Brox, 2011] to turn the resulting sparse trajectory clusters into dense regions. By defining the pairwise distance between trajectories as the maximum difference of their motion, they assume a translational motion model. Despite this being a reasonable approximation for spatially close point trajectories, these methods have difficulties to segment articulated bodies following non-rigid motion.

2.3 Semi-automatic Video Segmentation

Semi-automatic video object segmentation methods propagate a sparse manual labeling, generally given in the form of one or more annotated frames, to the entire video sequence. While being different from each other, they often solve an optimization problem with an energy defined over a graph structure [Ramakanth and Babu, 2014; Badrinarayanan et al., 2010; Vijayanarasimhan and Grauman, 2012]. To model long-range spatio-temporal connections some approaches use higher-order potentials [Jain and Grauman, 2014]. Semi-automatic segmentation is closely related to object tracking. While the scope of tracking is that of inscribing the object within a rectangular bounding box, video segmentation aims to delineate the object boundaries as accurately as possible. Due to the intrinsic objective similarity, several approaches have investigated approaches that improve segmentation quality by leveraging object tracking and vice versa, [Ren and Malik, 2007; Duffner and Garcia, 2013; Chockalingam et al., 2009; Xiao and Lee, 2016]. In Chapter 5 we propose a Convolutional Neural Network (ConvNet) based approach that, inspired by recent advances in object tracking, proceeds on a per-frame basis, and it is guided by the output of the previous frame towards the object of interest in the next frame. As detailed in the corresponding section this approach can handle different types of input annotations such as: bounding boxes or segments, making the system suitable for a diverse set applications.

2.3.1 Bounding-Box Tracking and Segmentation

Previous works have investigated approaches that improve segmentation quality by leveraging object tracking and vice versa [Ren and Malik, 2007; Duffner and Garcia, 2013; Chockalingam et al., 2009; Xiao and Lee, 2016]. More recent, state-of-the-art tracking methods are based on discriminative correlation filters over handcrafted features (e.g. HOG) and over frozen deep learned features [Danelljan et al., 2015; Danelljan et al., 2016], or are convnet based trackers on their own right [Held et al., 2016; Nam and Han, 2016]. Our approach is most closely related to the latter group. `GOTURN` [Held et al., 2016] proposes to train offline a convnet so as to directly regress the bounding box in the current frame based on the object position and appearance in the previous frame. `MDNet` [Nam and Han, 2016] proposes to use online fine-tuning of a convnet to model the object appearance. Our training strategy is inspired by `GOTURN` for the offline part, and `MDNet` for the online stage. Compared to the aforementioned methods our approach operates at pixel level masks instead of boxes. Differently from `MDNet`, we do not replace the domain-specific layers, instead finetuning all the layers on the available annotations for each individual video sequence.

2.3.2 Graph Based Video Segmentation

Images and videos naturally lend themselves to a regular graph structure where edges connect neighboring pixels in either a spatial or spatio-temporal configuration. Video segmentation can then be formulated as an optimization problem that tries to balance a coherent label assignment of neighboring vertices, while complying to a predetermined object model or user constraints. Graph-cuts techniques have long been used to efficiently solve this problem, both for image [Boykov and Jolly, 2001; Rother et al., 2004] and video segmentation [Li et al., 2005; Wang et al., 2005; Kohli and Torr, 2007; Price et al., 2009; Reso et al., 2014b; Dondera et al., 2014]. Building on this general framework, subsequent methods have lowered the computational cost by reducing the number of nodes in the graph using clustering techniques such as a per-frame watershed algorithm [Li et al., 2005; Price et al., 2009], mean-shift segmentation [Wang et al., 2005], or spatio-temporal superpixels [Reso et al., 2014b]. The last work showed high quality results when used with a contour-based EM optimization [Reso et al., 2014a] as well as faster, but still accurate results with online video seeds [Van den Bergh et al., 2013]. However, these methods still do not achieve interactive rates due to costly clustering steps, and allow only rough user control [Li et al., 2005], or require expensive per-pixel refinement on each

frame [Wang et al., 2005]. Additionally, the above clustering methods can fail in regions with poorly defined image boundaries. In Chapter 6 we propose to efficiently approximate non-local connections minimizing the graph energy in bilateral space. Minimize the energy function in bilateral space is efficient due to the reduced number of graph nodes and edges.

2.4 Interactive Video Segmentation

Supervised approaches assume manual annotation to be repeatedly added during the segmentation process, with a human correcting the algorithm results in an iterative fashion. These methods generally operate online, forward processing frames to avoid overriding of previous manual corrections. They guarantee high segmentation quality at the price of higher level of human supervision, hence they are well suited for specific scenarios such as video editing. In post-production, scene segmentation is regarded with the term *rotoscoping*. Rotoscoping is not driven only by the generic notion of object but also requires creative control and therefore it cannot be fully automated. Furthermore the task is extremely time-consuming and expensive. As a consequence, a large body of research have investigated this topic, with the aim of reducing the amount of human effort required to reach high quality.

The seminal work of Chuang et al. [2002] uses a Bayesian matting technique on top of back-forward flow propagated tri-maps to yield accurate soft-segmentation of moving objects. Agarwala et al. [2004] reformulates contour-based tracking as part of a user-driven key frame system. Based on user-defined key frames a space-time optimization problem finds the best interpolation of the roto-curves over time. Li et al. [2014], apply 3D graph cut based segmentation approach on the spatio-temporal video volume. Their algorithm partitions watershed segmentation regions into foreground and background while preserving temporal coherence. The resulting segmentation is further refined using 2D graph-cuts inside tracked boxes. Wang et al. [2014] aims to reduce user interaction proposing an algorithm that requires only one finger touch to identify the object of interest and perform the segmentation. Their approach proposes a new model for object segmentation that fuses edge, region, and geometric cues within a level set framework. To cope with a diverse set of situations, Price et al. [2009] propose an interactive approach that extract multiple features and learn how to combine automatically based on user input corrections, yielding a method that selectively applies the cues that are likely to segment the object in that particular scene context. Video SnapCut [Bai et al., 2009] uses overlapping local

classifiers that predict the foreground probability, which are propagated and refined over time. SnapCut was later integrated into Adobe After Effects as the Rotobrush tool. This approach was extended to a combination of local and global classifiers [Zhong et al., 2012] to improve robustness. Dondera et al. [2014], apply the spectral clustering method of Ng et al. [2002] on a graph of super-pixels in a 3D video volume. An initial segmentation is obtained without additional input, the user can then add constraints to correct the solution. Labels are then inferred using a conditional random field formulation. Fan et al. [2015] propose a method that propagates masks using nearest neighbor fields, and then refines the result with active contours on classified edge maps. As this is one of the top performing methods in the semi-automatic settings while still enabling user-interaction, in Chapter 6 we use it as a basis for our comparisons.

2.5 Datasets

Over the years, datasets and benchmarks have proven their fundamental importance in computer vision research, enabling targeted progress and objective comparisons in many fields [Torralba and Efros, 2011]. There exist several datasets for video segmentation, but none of them has been specifically designed for video *object* segmentation, the task of pixel-accurate separation of foreground objects from the background regions.

The *Freiburg-Berkeley Motion Segmentation* (MoSeg) dataset [Brox and Malik, 2010] is a popular dataset for motion segmentation, *i.e.* clustering regions with similar motion. Despite being recently adopted by works focusing on video object segmentation [Perazzi et al., 2015b; Taylor et al., 2015], the dataset does not fulfill several important requirements. Most of the videos have low spatial resolution, segmentation is only provided on a sparse subset of the frames, and the content is not sufficiently diverse to provide a balanced distribution of challenging situations such as fast motion and occlusions.

The *Berkeley Video Segmentation Dataset* (BVSD) [Sundberg et al., 2011] comprises a total 100, higher resolution sequences. It was originally meant to evaluate occlusions boundary detection and later extended to over- and motion-segmentation tasks (VSB100 [Galasso et al., 2013]). However, several sequences do not contain a clear object. Furthermore, the ground-truth, available only for a subset of the frames, is fragmented, with most of the objects being covered by multiple manually annotated, disjoint segments, and therefore, most of this dataset is not well suited for evaluating video object segmentation.

SegTrack [Tsai et al., 2010] is a small dataset composed of 6 densely annotated videos of humans and animals. It is designed to be challenging with respect to background-foreground color similarity, fast motion and complex shape deformation. Although it has been extensively used by several approaches, its content does not sufficiently span the variety of challenges encountered in realistic video object segmentation applications. Furthermore, the image quality is not anymore representative of modern consumer devices, and due to the limited number of available video sequences, progress on this dataset plateaued. Li et al. [2013] extended this dataset with 8 additional sequences. While this is certainly an improvement over the predecessor, it still suffers of the same limitations.

Other datasets exist, but they are mostly provided to support specific findings and thus are either limited in terms of total number of frames, [Chen and Corso, 2010; Tsai et al., 2010; Li et al., 2013; Grundmann et al., 2010], or do not exhibit a sufficient variety in terms of content [Tron and Vidal, 2007; Brostow et al., 2009; Badrinarayanan et al., 2010; Fragkiadaki and Shi, 2011; Gorelick et al., 2007; Brox and Malik, 2010; Fathi et al., 2011; Ren and Philipose, 2009]. Others cover a broader range of content but do not provide enough ground-truth data for an accurate evaluation of the segmentation [Grundmann et al., 2010; Prest et al., 2012]. Video datasets designed to benchmark tracking algorithms typically focus on surveillance scenarios with static cameras [Collins et al., 2005; Fisher, 2004; Oh et al., 2011], and usually contain multiple instances of similar objects [Wu et al., 2013] (e.g. a crowd of people), and annotation is typically provided only in the form of axis-aligned bounding boxes, instead of pixel-accurate segmentation masks necessary to accurately evaluate video object segmentation. Importantly, none of the aforementioned methods includes contemporary high resolution videos, which is an absolute necessity to realistically evaluate the actual practical utility of such algorithms.

In Chapter 7, we propose a new dataset specifically geared towards the task of video object segmentation. The dataset aims to overcome the limitations of the aforementioned datasets and it comes with a well defined evaluation protocol and an extensive benchmark of current state-of-the-art approaches.

CHAPTER 3

Salient Object Detection

In this chapter we investigate two orthogonal approaches aiming to discover salient objects. An object is salient when it stands out relative to neighboring image regions. The ability to automatically detect salient objects is particularly relevant to the task of video object segmentation as it enables semi-automatic techniques to operate in unsupervised mode, *i.e.* without need of manual annotations. Saliency detection finds its root in the mechanism of human attention. In particular the pre-attentive human visual system is driven by bottom-up, low-level stimuli such as color, contrast, orientation of edges, disparity and sudden movements [Koch and Ullman, 1985]. Depending on the nature of their features, methods that model bottom-up visual saliency can be categorized into biologically inspired or computationally based approaches. Biologically inspired methods aim to determine eye fixations, *i.e.*, a set of points or blobs in the image that are likely to attract the viewer's eye attention. As a result, saliency maps are often blurry and highlight sparse local features, making their usage in computer vision applications such video object segmentation or impracticable [Cheng et al., 2011]. In contrast, computational methods are often inspired by biological principles but strongly focus on their practical usage in computer vision and graphics. Central to those applications is the ability to determine salient objects, instead of eye fixation points. Hence, an important aspect to consider is the ability to segment and assign a uniform saliency value to the entire salient object [Chang et al., 2011; Achanta et al., 2008], preserving edges and producing a pixel-level accurate saliency map. In this chapter we propose two different computational approaches to tackle the task of salient object detection.

Perceptual research studies indicates that color-based contrast is a fundamental cue to determine bottom-up visual attention [Parkhurst et al., 2002; Einhauser et al., 2003]. In Section 3.1 we present a novel approach that derives a saliency estimate from two well-defined contrast measures based on the uniqueness and spatial distribution of color within an image. In Section 3.2 we propose to identify salient regions by eigenvalue analysis of a graph Laplacian that is defined over the color similarity of image superpixels. In this case, the underlying assumption is that the majority of pixels on image boundaries belong to non-salient background. Experiments demonstrates the complementary nature of the "background-prior" property to color contrast-based approaches.

3.1 Saliency Filters

Results from perceptual research [Reinagel and Zador, 1999; Parkhurst et al., 2002; Einhauser et al., 2003] indicate that the most influential factor in low-level visual saliency is *contrast*. However, the definition of contrast in previous works is based on various different types of image features, including color variation of individual pixels, edges and gradients, spatial frequencies, structure and distribution of image patches, histograms, multi-scale descriptors, or combinations thereof. The significance of each individual feature often remains unclear [Liu et al., 2007], and as recent evaluations show [Cheng et al., 2011] even quite similar approaches may exhibit considerably varying performance.

We reconsider the set of fundamentally relevant contrast measures and their definition in terms of image content. Our method is based on the observation that an image can be decomposed into basic, structurally representative elements that abstract away unnecessary detail, and at the same time allow for a very clear and intuitive definition of contrast-based saliency. Our first main contribution therefore is a concept and algorithm to decompose an image into perceptually homogeneous elements and to derive a saliency estimate from two well-defined contrast measures based on the uniqueness and spatial distribution of those elements. Both, local as well as the global contrast are handled by these measures in a unified way. Central to the contrast and saliency computation is our second main contribution; we show that all involved operators can be formulated within a single high-dimensional Gaussian filtering framework. Thanks to this formulation, we achieve a highly efficient implementation with linear complexity. The same formulation also provides a clear link between the element-based contrast estimation and the actual assignment of saliency values to all

Figure 3.1: *Illustration of the main phases of our algorithm. The input image is first abstracted into perceptually homogeneous elements. Each element is represented by the mean color of the pixels belonging to it. We then define two contrast measures per element based on the uniqueness and spatial distribution of elements. Finally, a saliency value is assigned to each pixel.*

image pixels. As we demonstrate in our experimental evaluation, each of our individual measures already performs close to or even better than existing approaches, and our combined method currently achieves the best ranking results on the public benchmark provided by [Liu et al., 2007; Achanta et al., 2009].

3.1.1 Method

We propose an algorithm that first decomposes the input image into basic elements. Based on these elements we define two measures for contrast that are used to compute per-pixel saliency. Hence, our algorithm consists of the following steps (Figure 3.1):

Abstraction. We decompose the image into perceptually homogeneous regions that preserve relevant structure, but abstract undesirable detail. Discontinuities between such regions, *i.e.* , strong contours and edges in the image, should be preserved as boundaries between individual elements. One approach to achieve this type of decomposition is an edge-preserving, localized oversegmentation based on color, Figure 3.1. Thanks to this abstraction, contrast between whole image regions can be evaluated using just those elements. Furthermore, we show that the quality of saliency maps is extremely robust to the number of elements. We can then define our two measures for contrast.

Element uniqueness. This first contrast measure implements the commonly employed assumption that image regions, which stand out from other regions in certain aspects, catch our attention and hence should be labeled more salient. We therefore evaluate how different each respective element is from all other elements constituting an image, essentially measuring the "rarity" of each element. In one form or another, this assumption has been

the basis for most previous algorithms for contrast-based saliency. However, thanks to our abstraction, variation on the pixel level due to small scale textures or noise is rendered irrelevant, while discontinuities such as strong edges stay sharply localized. As discussed in Section 2.1, previous multi-scale techniques often blur or lose this information.

Element distribution. While saliency implies uniqueness, the opposite might not always be true [Kadir and Brady, 2001]. Ideally colors belonging to the background will be distributed over the entire image exhibiting a high spatial variance, whereas foreground objects are generally more compact [Liu et al., 2007; Goferman et al., 2010].

The compactness and locality of our image abstracting elements allows us to define a corresponding second measure, which renders unique elements more salient when they are grouped in a particular image region rather than evenly distributed over the whole image. Techniques based on larger-scale image segmentation such as Cheng et al. [2011] lose this important source of information.

An example showing the differences between element uniqueness and element distribution is shown in Figure 3.2.

Saliency assignment. The two above contrast measures are defined on a per-element level. In a final step, we assign the actual saliency values to the input image to get a pixel-accurate saliency map. Thanks to this step our method can assign proper saliency values even to fine pixel-level detail that was excluded, on purpose, during the abstraction phase, but for which we still want a saliency estimate that conforms to the global saliency analysis.

3.1.1.1 Abstraction

For the image abstraction we use an adaptation of SLIC superpixels [Achanta et al., 2012] to abstract the image into perceptually uniform regions. SLIC superpixels segment an image using K-means clustering in RGBXY space. The RGBXY space yields local, compact and edge aware superpixels, but does not guarantee compactness. For our image abstraction we slightly modified the SLIC approach and instead use K-means clustering in geodesic image distance [Criminisi et al., 2010] in CIELab space. Geodesic image distance guarantees connectivity, while retaining the locality, compactness and edge awareness of SLIC superpixels.

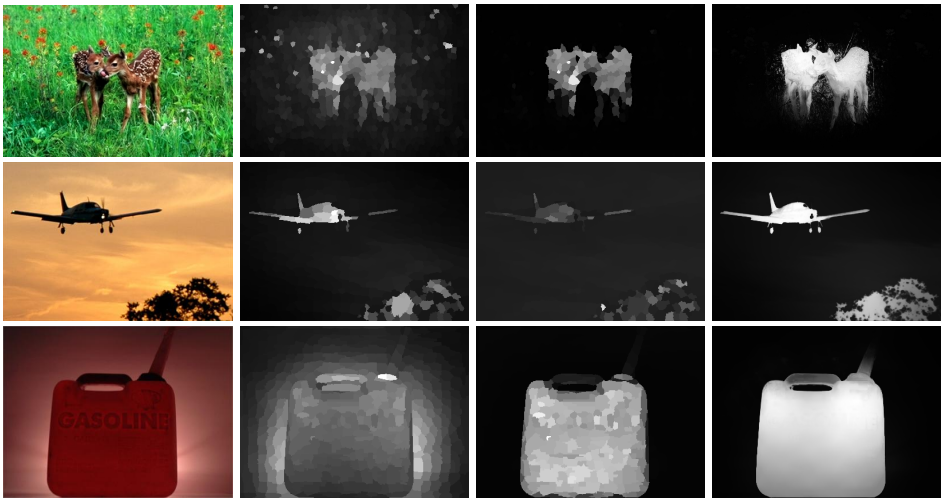


Figure 3.2: *Uniqueness, spatial distribution, and the combined saliency map. The uniqueness prefers rare colors, whereas the distribution favors compact objects. Combined together those measures provide better perfomance.*

3.1.1.2 Element uniqueness

Element uniqueness is generally defined as the rarity of a segment i given its position $\mathbf{p}_i$ and color in CIELab $\mathbf{c}_i$ compared to all other segments j:

$$U_i = \sum_{j=1}^{N} \|\mathbf{c}_i - \mathbf{c}_j\|^2 \cdot \underbrace{w(\mathbf{p}_i, \mathbf{p}_j)}_{w_{ij}^{(p)}}. \tag{3.1}$$

By introducing $w_{ij}^{(p)}$ we effectively combine global and local contrast estimation with control over the influence radius of the uniqueness operator. A local function $w_{ij}^{(p)}$ yields a local contrast term, which tends to overemphasize object boundaries in the saliency estimation [Ma and Zhang, 2003], whereas $w_{ij}^{(p)} \approx 1$ yields a global uniqueness operator, which cannot represent sensitivity to local contrast variation.

Moreover, evaluating Eq. (3.1) globally generally requires $O(N^2)$ operations, where N is the number of segments. This is why some related works downsample the image to a resolution where quadratic number of operations is feasible. Saliency maps computed on down-sampled images cannot preserve sharply localized contours and generally exhibit a high level of blurriness (§3.3). Cheng et al. [2011] approximate Eq. (3.1) using a histogram.

Achatan et al. [2009] approximate it as the distance to mean color. Both approximations are completely global with $w_{ij}^{(p)} = 1$.

We show that for a Gaussian weight $w_{ij}^{(p)} = \frac{1}{Z_i} \exp(-\frac{1}{2\sigma_p^2} \|\mathbf{p}_i - \mathbf{p}_j\|^2)$ Eq. (3.1) can be evaluated in linear time $O(N)$. σ_p controls the range of the uniqueness operator and Z_i is the normalization factor ensuring $\sum_{j=1}^{N} w_{i,j}^{(p)} = 1$. We decompose Eq. (3.1) by factoring out the quadratic error function:

$$\begin{aligned} U_i &= \sum_{j=1}^{N} \|\mathbf{c}_i - \mathbf{c}_j\|^2 w_{ij}^{(p)} \\ &= \mathbf{c}_i^2 \underbrace{\sum_{j=1}^{N} w_{ij}^{(p)}}_{1} - 2\mathbf{c}_i \underbrace{\sum_{j=1}^{N} \mathbf{c}_j w_{ij}^{(p)}}_{\text{blur } c_j} + \underbrace{\sum_{j=1}^{N} \mathbf{c}_j^2 w_{ij}^{(p)}}_{\text{blur } c_j^2}. \end{aligned} \tag{3.2}$$

Both terms $\sum_{j=1}^{N} \mathbf{c}_j w_{ij}^{(p)}$ and $\sum_{j=1}^{N} \mathbf{c}_j^2 w_{ij}^{(p)}$ can be evaluated using a Gaussian blurring kernel on color $\mathbf{c}_j$ and the squared color $\mathbf{c}_j^2$. Gaussian blurring is decomposable along x and y axis of the image and can thus be evaluated very efficiently.

In our implementation we use the permutohedral lattice embedding presented in Adams et al. [2010], which yields a linear time approximation of the Gaussian filter in arbitrary dimensions. The permutohedral lattice exploits the band limiting effects of Gaussian smoothing, such that a correspondingly filtered function can be well approximated by a sparse number of samples. Adams et al. [2010] use samples on simplices of a high dimensional lattice structure to represent the result of the filtering operation. They then evaluate the filter by downsampling the input values onto the lattice, blur along each dimension of the lattice and reconstruct the resulting signal by interpolation.

By using a Gaussian weight $w_{ij}^{(p)}$ we are able to evaluate Eq. (3.1) in linear time, without crude approximations such as histograms or distance to mean color. Parameter σ_p was set to 0.25 in all experiments, which allows for a balance between local and global effects. Examples for the uniqueness measure are shown in Figure 3.2b.

3.1.1.3 Element distribution

Conceptually, we define the element distribution measure for a segment i using the spatial variance D_i of its color $\mathbf{c}_i$, *i.e.*, we measure its occurrence

elsewhere in the image. As motivated before, low variance indicates a spatially compact object which should be considered more salient than spatially widely distributed elements. Hence we compute

$$D_i = \sum_{j=1}^{N} \|\mathbf{p}_j - \mu_i\|^2 \underbrace{w(\mathbf{c}_i, \mathbf{c}_j)}_{w_{ij}^{(c)}}, \tag{3.3}$$

where $w_{ij}^{(c)}$ describes the similarity of color $\mathbf{c}_i$ and color $\mathbf{c}_j$ of segments i and j, respectively, $\mathbf{p}_j$ is again the position of segment j, and $\mu_i = \sum_{j=1}^{N} w_{ij}^{(c)} \mathbf{p}_j$ defines the weighted mean position of color $\mathbf{c}_i$.

Again naive evaluation of Eq. (3.3) has quadratic runtime complexity. By choosing the color similarity to be Gaussian $w_{ij}^{(c)} = \frac{1}{Z_i} \exp(-\frac{1}{2\sigma_c^2}\|\mathbf{c}_i - \mathbf{c}_j\|^2)$, we can efficiently evaluate it in linear time:

$$\begin{aligned} D_i &= \sum_{j=1}^{N} \|\mathbf{p}_j - \mu_i\|^2 w_{ij}^{(c)} \\ &= \sum_{j=1}^{N} \mathbf{p}_j^2 w_{ij}^{(c)} - 2\mu_i \underbrace{\sum_{j=1}^{N} \mathbf{p}_j w_{ij}^{(c)}}_{\mu_i} + {\mu_i}^2 \underbrace{\sum_{j=1}^{N} w_{ij}^{(c)}}_{1} \\ &= \underbrace{\sum_{j=1}^{N} \mathbf{p}_j^2 w_{ij}^{(c)}}_{\text{blur } \mathbf{p}_j^2} - \underbrace{\mu_i}_{\text{blur } \mathbf{p}_j}{}^2. \end{aligned} \tag{3.4}$$

Here the position $\mathbf{p}_j$ and squared position $\mathbf{p}_j^2$ are blurred in the 3-dimensional color space. It can be efficiently evaluated by discretizing the color space and then evaluating a separable Gaussian blur along each of the L, a and b dimension. Since the Gaussian filter is additive, we can simply add position values associated to the same color. As in Eq. (3.2) we use the permutohedral lattice [Adams et al., 2010] as a linear approximation to the Gaussian filter in the Lab space.

The parameter σ_c controls the color sensitivity of the element distribution. We use $\sigma_c = 20$ in all our experiments. See Figure 3.2 for a visual comparison of uniqueness and spatial distribution.

In summary, by simple evaluation of two Gaussian filters we can compute two non-trivial, but intuitively defined contrast measures on a per-element basis. By filtering color values in the image, we compute the uniqueness of

an element, while filtering position values in the Lab color space gives us the element distribution. Next we will look at how to combine both measures, which have a different scaling and units associated to them, in order to compute a per-pixel saliency value.

3.1.1.4 Saliency assignment

We start by normalizing both uniqueness U_i and distribution D_i to the range $[0..1]$. We assume that both measures are independent, and hence we combine these terms as follows to compute a saliency value S_i for each element:

$$S_i = U_i \cdot \exp(-k \cdot D_i), \tag{3.5}$$

In practice we found the distribution measure D_i to be of higher significance and discriminative power. Therefore, we use an exponential function in order to emphasize D_i. In all our experiments we use $k = 6$ as the scaling factor for the exponential.

Figure 3.2 shows a visual comparison of the uniqueness U_i, distribution D_i and their combination S_i. As the final step, we need to assign a final saliency value to each image pixel, which can be interpreted as an up-sampling of the per-element saliency S_i. However, naive up-sampling by assigning S_i to every pixel contained in element i carries over all segmentation errors of the abstraction algorithm. Instead we adopt an idea proposed in the context of range image up-sampling [Dolson et al., 2010] and apply it to our framework. We define the saliency $\tilde{S}_i$ of a pixel as a weighted linear combination of the saliency S_j of its surrounding image elements

$$\tilde{S}_i = \sum_{j=1}^{N} w_{ij} S_j. \tag{3.6}$$

By choosing a Gaussian weight $w_{ij} = \frac{1}{Z_i} \exp(-\frac{1}{2}(\alpha \|\mathbf{c}_i - \mathbf{c}_j\|^2 + \beta \|\mathbf{p}_i - \mathbf{p}_j\|^2)$, we ensure the up-sampling process is both local and color sensitive. Here α and β are parameters controlling the sensitivity to color and position. We found $\alpha = \frac{1}{30}$ and $\beta = \frac{1}{30}$ to work well in practice.

As for our contrast measures in Eq. (3.1) and (3.3), Eq. (3.6) describes a high-dimensional Gaussian filter and can hence be evaluated within the same filtering framework [Adams et al., 2010]. The saliency value of each element is embedded in a five-dimensional space using its position $\mathbf{p}_i$ and its color value $\mathbf{c}_i$ in RGB (as we found it to outperform CIELab for up-sampling). Since our abstract elements do not have a regular shape we create a point

sample in RGBXY space at each pixel position $\tilde{\mathbf{p}}_i$ within a particular element and blur the RGBXY space along each of its dimensions. The per-pixel saliency values can then be retrieved with a lookup in that high-dimensional space using the pixel's position $\tilde{\mathbf{p}}_i$ and its color value $\tilde{\mathbf{c}}_i$ in the input image.

The resulting pixel-level saliency map can have an arbitrary scale. In a final step we rescale the saliency map to the range $[0..1]$ or to contain at least 10% saliency pixels.

In summary, our algorithm computes the saliency of an image by first abstracting it into small, perceptually homogeneous elements. It then applies a series of three Gaussian filtering steps in order to compute the uniqueness and spatial distribution of elements as well as to perform the final per-pixel saliency assignment. Qualitative results are shown in Figure 3.10, while we refer the reader to Section 3.3 for quantitative results.

3.2 Saliency Detection using Fiedler Vectors

In the previous section we described a method for detecting salient objects that implements the color-contrast assumption. While quantitative and qualitative results demonstrate the effectiveness of this prior, there are several scenarios where it fails. To this end, we investigate a new method that is based on the basic assumption that most of the image boundaries are covered by non-salient background. Background color priors and local color similarities are encoded in a graph structure defined over a superpixel segmentation of images or video frames. Starting from the eigenvalue decomposition of the graph Laplacian, we demonstrate that the eigenvector corresponding to the second smallest eigenvalue (Fiedler vector) provides a very effective and robust way to compute saliency masks. In addition, differently from previous approaches that use various heuristics or graph-cut segmentation to binarize saliency maps, the entries of the Fiedler vector yield both a continuous estimate and a content-adaptive binary partition.

Despite its computational simplicity, we show in our examples and evaluation that our method compares favorably to recent methods and efficiently handles various image and video types that are challenging for previous approaches. To demonstrate the complementary nature of our method, we also show that the performance can be further increased when combining our approach with the technique presented in Section 3.1, which is based on color-contrast.

3.2.1 Method

The algorithm consists of three simple steps. First, the input image is decomposed into superpixels (§3.2.1.1). Next, we compute a weighted graph $\mathcal{G}$ connecting adjacent superpixels to a dummy node representing the unknown background regions. Finally, a saliency score is assigned to each superpixel based on the eigenvalue analysis of the Laplacian matrix of G. Details are given in the following paragraphs.

3.2.1.1 Image representation

As a first step our algorithm, similarly to the technique proposed in Section 3.1, decomposes an input image into superpixels, as they provide an effective and perceptually meaningful level of abstraction, and remove unnecessary detail such as small scale non-salient structures and noise. To segment the image into superpixels we use a variant of Achanta et al. [2012] as described in Section 3.1, which is fast and preserves discontinuities such as edges well.

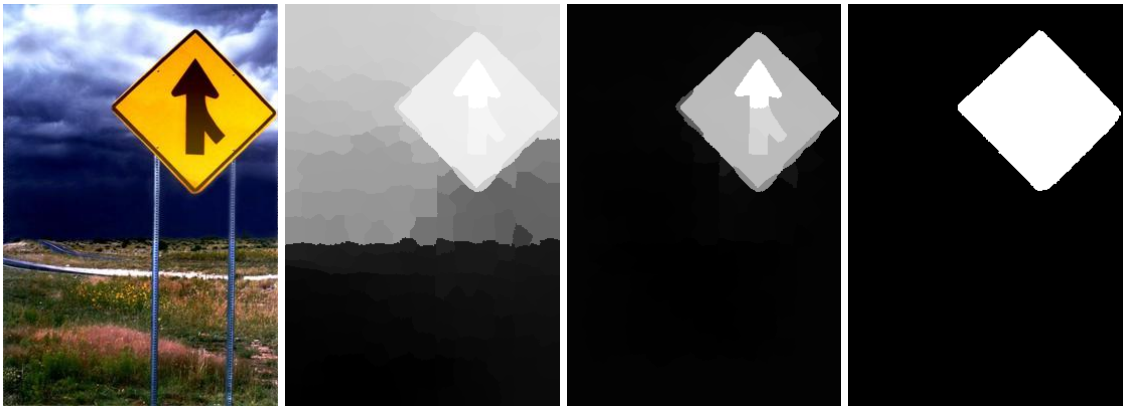
The superpixel-decomposition of the image induces an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertices $\mathcal{V}$ correspond to superpixels and the edges $\mathcal{E}$ represent an adjacency relationship between the superpixels. Similarly to segmentation algorithms such as Shi and Malik [1997], we model only local relationships, i.e. $(i, j) \in \mathcal{E}$ only if the superpixels corresponding to the nodes v_i, v_j share contiguous pixels in the image. We assign each node v_i the mean Lab color of the superpixel it belongs to, denoted as c_i. The Lab color space is chosen because its Euclidean metric mimics the human color perception. Each edge (i, j) is assigned a positive weight $w_{i,j}$ that measures the color similarity between superpixels v_i and v_j, higher values corresponding to higher similarity:

$$w_{i,j} = \frac{1}{\|c_i - c_j\|^2 + \epsilon} \tag{3.7}$$

where ϵ is a small constant to avoid infinite weights.

Performing a straightforward partitioning of the graph, e.g., using *Ratio-Cut* [Hagen and Kahng, 1992], to separate the superpixels into potential fore- and background regions is not sufficient to obtain a reliable salient object estimate. The quantitative evaluation on the MSRA dataset (§3.3) shows that the performance is substantially below state-of-art methods. See also Figure 3.3 for a representative saliency result.

We therefore incorporate a simple prior assuming that the majority of boundary superpixels belongs to non-salient background, motivated by re-



Source Image | No Augmentation | Augmentation | Ground Truth

Figure 3.3: *Graph augmentation with background prior. From left to right: source image, saliency map computed without our graph augmentation, saliency map using our method, and ground truth. The boundary prior and our graph augmentation are key to separating the background from the salient foreground object.*

cent studies in gaze prediction which indicate that humans have a tendency to focus attention on the center of an image. This is also reflected in various photographic rules and utilized in saliency estimation techniques such as Wei et al. [2012] . This prior is integrated by augmenting the graph with a background node b and a set of edges $\mathcal{U}$ connecting b to the nodes forming the image boundaries, i.e., to those superpixels that are in immediate contact with the image border.

The augmented graph is hence $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}_a)$ with $\mathcal{V}_a = \mathcal{V} \cup \{b\}$ and $\mathcal{E}_a = \mathcal{E} \cup \mathcal{U}$. The edge weights in $\mathcal{U}$ model the confidence of a node in being part of the background. We use the Euclidean distance to the mean boundary color. We assign the mean boundary color to b and compute the weights of the edges in $\mathcal{U}$ with Eq. (3.7). With this formulation, most of the edges in $\mathcal{U}$ are likely to be attached to background superpixels and carry high weight, while few edges (if any) are attached to salient regions and have low weights.

3.2.1.2 Saliency estimation

Denote $n = |\mathcal{V}_a|$. We compute an eigendecomposition of the weighted graph Laplacian matrix $L \in \mathbb{R}^{n \times n}$ of $\mathcal{G}_a$:

$$L_{i,j} = \begin{cases} -w_{i,j} & i \neq j,\ (i,j) \in \mathcal{E}_a \\ \sum_{(i,k)\in\mathcal{E}} w_{i,k} & i = j \\ 0 & \text{otherwise} \end{cases} . \tag{3.8}$$

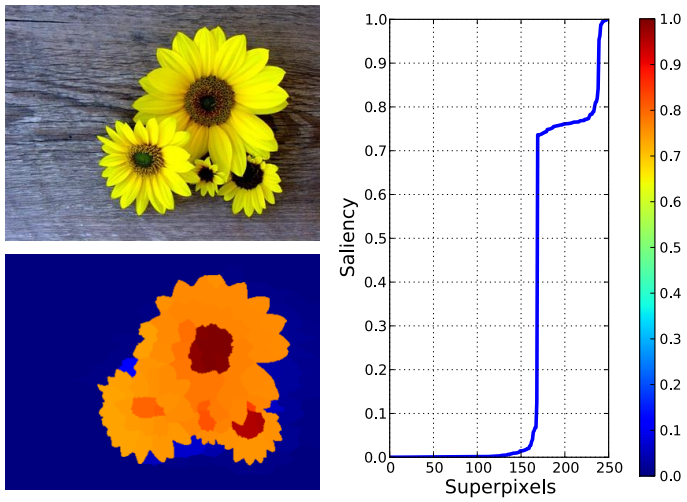


Figure 3.4: *Saliency computation using the Fiedler vector. In our approach the input image on the top left is represented by a graph structure that encodes color similarities between superpixels and a background color prior computed from the image boundary. The Fiedler vector of the graph Laplacian results in a continuously-valued saliency estimate for every superpixel, illustrated by the saliency map on the bottom left and the plot.*

The eigenvector f corresponding to the second smallest eigenvalue, also known as the *Fiedler* vector, represents an optimal *soft* segmentation of $\mathcal{G}_a$ according to a relaxed, continuously valued *RatioCut* objective [von Luxburg, 2007] by minimizing

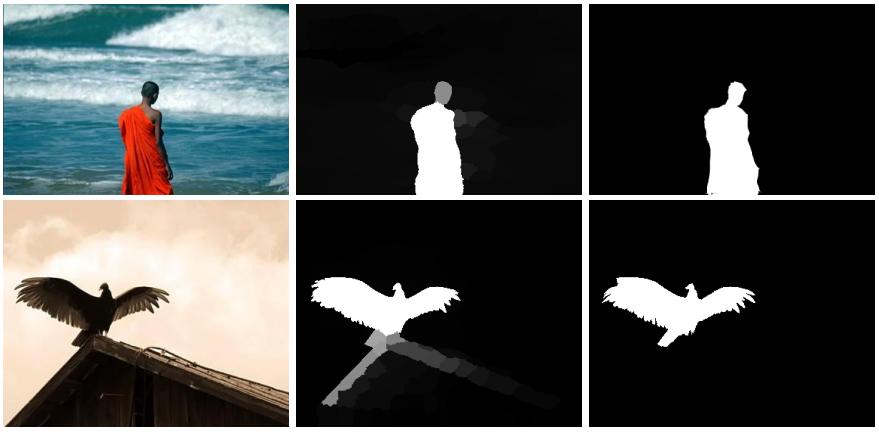
$$\min_f \sum_{i,j \in \mathcal{E}} w_{i,j}(f_i - f_j)^2. \tag{3.9}$$

The entries of this vector can be interpreted as a one-dimensional (linear) embedding of $\mathcal{G}_a$, where vertices are closer to each other if they are connected by large weights.

We found that this property of the Fiedler vector f provides a meaningful, continuously valued saliency score (Figure 3.4). We can derive either a saliency score $S_{\text{cont}} \in [0,1]^n$ or a binary partition $S_{\text{bin}} \in \{0,1\}^n$. Both measures are based on the sign of the entries of the Fiedler vector. Entries having the same sign as the entry f_b corresponding to the background node b will be less salient than those having the opposite sign. Hence we define the continuously-valued saliency score S_{cont} as:

$$S_{\text{cont}} = -\text{sign}(f_b) \cdot f \tag{3.10}$$

This sign-corrected S_{cont} is then scaled to the range $[0,1]$, possibly with pre-cropping of the value range such that the resulting mean saliency is at least 0.1.



Source Saliency Ground Truth

Figure 3.5: *Robustness to salient objects being part of the image boundaries. From left to right: source, our saliency map S_{cont}, ground truth saliency.*

A binary partition is obtained by discretizing the entries of the Fiedler vector f. This operation can be performed based on the sign of f_b, such that entries having opposite sign to f_b are defined as salient:

$$S_{\text{bin}}(i) = 1 \text{ if } f_i \cdot f_b \leq 0, \;\; S_{\text{bin}}(i) = 0 \text{ otherwise} \tag{3.11}$$

To subdivide the graph into more than two partitions, *i.e.*, to identify multiple individual salient objects, the entries of f can be interpreted as points in $\mathbb{R}$ and partitioned by a clustering algorithm such as k-means.

The above approach is quite robust even in challenging cases where the salient object is actually part of the image boundary. As long as the majority of the superpixels is part of the background, the graph partitioning correctly distinguishes between salient and non-salient areas (Figure 3.5).

3.3 Results

In order to compare our two proposed techniques, with respect to previous works, we evaluate the per-image saliency maps on a well established dataset, with manually labeled ground-truth saliency: the **MSRA** [Cheng et al., 2011; Achanta et al., 2009] dataset with 1000 images.

In accordance with Borji et al. [2012] we compare our result with several state-of-the-art approaches such as, context-aware saliency (CA [Goferman et al., 2010]) and global-contrast (RC [Cheng et al., 2011]). We also combine the methods proposed in Section 3.1 and Section 3.2 by simple averag-

ing of the saliency maps (denoted by FV+SF) in order to demonstrate the complementary nature of our approach to contrast-based techniques. Next we describe the two error measures commonly used to evaluate the performance of the aforementioned algorithms and discuss the results or our proposed approaches. In Section 3.4 we summarize conclusions, limitations and future works. The algorithm proposed in Section 3.1 is also used as a pre-processing baseline in our benchmark of video segmentation algorithms and therefore further evaluation can be found in Section 7.4.1.

In Figure 3.10 we show a qualitative evaluation of our approaches.

3.3.1 Precision and Recall

We evaluate the performance of our algorithm measuring its *precision* and *recall* rate. Precision corresponds to the percentage of salient pixels correctly assigned, while recall corresponds to the fraction of detected salient pixels in relation to the ground truth number of salient pixels.

High recall can be achieved at the expense of reducing the precision and vice-versa so it is important to evaluate both measures together. We perform two different experiments. In both cases we generate a binary saliency map based on some saliency threshold. In the first experiment we compare binary masks for every threshold in the range $[0..255]$. The resulting curves in Figure 3.6 show that our algorithms (SF,FV) consistently produces results closer to ground truth at every threshold and for any given recall rate.

In the second experiment we use the image dependent adaptive threshold proposed by [Achanta et al., 2009], defined as twice the mean saliency of the image:

$$T_a = \frac{2}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} S(x, y), \tag{3.12}$$

where W and H are the width and the height of the saliency map S, respectively. In addition to precision and recall we compute their weighted harmonic mean measure or *F-measure*, which is defined as:

$$F_\beta = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}. \tag{3.13}$$

Similar to Achanta et al. [2009] and Cheng et al. [2011] we set $\beta^2 = 0.3$.

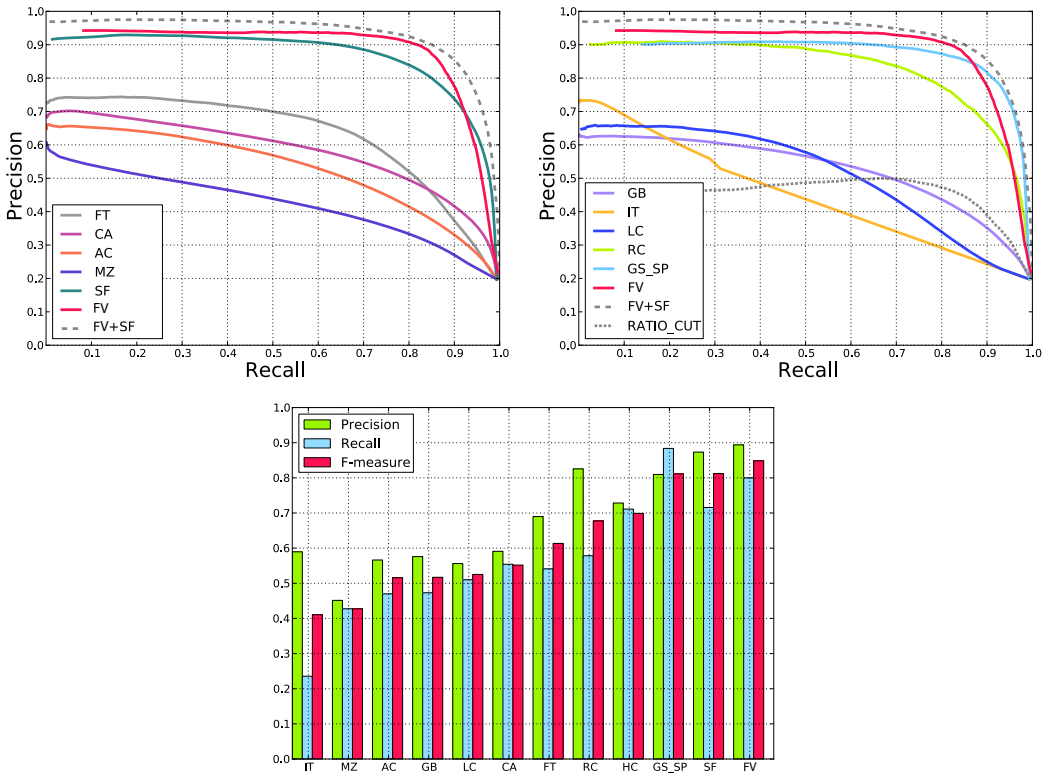


Figure 3.6: *Precision and recall rates for adaptive (top) and fixed (bottom) thresholds. We split adaptive threshold comparison of all methods into two plots for improved readability.*

3.3.2 Mean Absolute Error

Neither the precision nor recall measure consider the true negative saliency assignments, *i.e.*, the number of pixel correctly marked as non-salient. This favors methods that successfully assign saliency to salient pixels but fail to detect non-salient regions over methods that successfully detect non-salient pixels but make mistakes in determining the salient ones. Moreover, in some application scenarios [Avidan and Shamir, 2007] the quality of the weighted, continuous saliency maps may be of higher importance than the binary masks.

For a more balanced comparison that takes these effects into account we therefore propose to evaluate the *mean absolute error* (MAE) between the continuous saliency map S (prior to thresholding) and the binary ground truth GT. The mean absolute error is then defined as

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - GT(x,y)|, \tag{3.14}$$

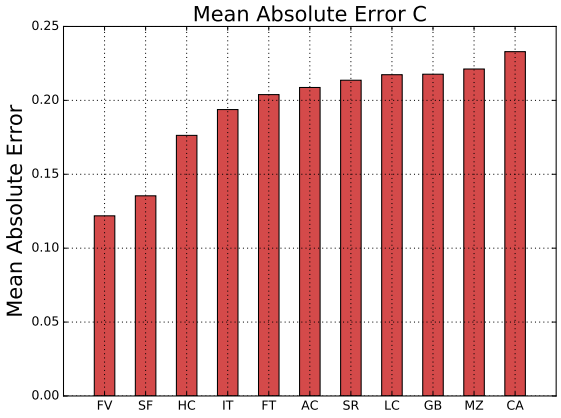


Figure 3.7: *Mean absolute error of the different saliency methods to ground truth. The proposed approaches (FV,SF) outperform the state-of-the-art.*

where W and H are again the width and the height of the respective saliency map and ground truth image.

Figure 3.7 shows that our methods also outperforms the other approaches in terms of the MAE measure, which provides a better estimate of the dissimilarity between the saliency map and ground truth.

3.4 Discussion

In this chapter we presented two methods for detecting salient objects in still images. In Section 3.1 we presented a method for saliency computation based on an image abstraction into structurally representative elements and contrast-based saliency measures, which can be consistently formulated as high-dimensional Gaussian filters. Our filter-based formulation allows for efficient computation and produces accurate per-pixel saliency maps. In Section 3.2 we presented a complementary method that combines the assumption of image boundaries covered mostly by background with soft graph segmentation using the Fiedler vector, yielding a continuously-valued solution to salient foreground detection and segmentation. Our approaches compare favorably to the state-of-the-art on a well established benchmark for salient object segmentation.

Limitations and Future Works. Saliency estimation based on color contrast may not always be feasible, *e.g.*, in the case of lighting variations, or when fore- and background colors are very similar. In such cases, the thresholding procedures used for all the above evaluations can result in noisy segmentations (Figure 3.8). One option to significantly reduce this effect is to perform

a single min-cut segmentation [Boykov and Kolmogorov, 2004] as a post process, using our saliency maps as a prior for the min-cut data term, and color differences between neighboring pixels for the smoothness term. The graph structure facilitates smoothness of salient objects and significantly improves the performance of our algorithms.

Similarly the method presented in Section 3.2 fails when the *boundary prior* does not old, *i.e.*, when a salient object covers most of image boundaries. Furthermore, in its current formulation, our approach is particularly effective for the detection of *single* salient objects. For example, in Figure 3.9 our algorithm correctly detects the salient object with the strongest separation from the background, but fails to detect the remaining pieces. Multiple salient objects could be retrieved by repeatedly segmenting the salient region [Lu et al., 2011] or using a sliding window approach [Feng et al., 2011]. For the latter, the shape of the Fiedler vector might well serve as an additional indicator of the number of salient objects within the window.

As discussed in the limitations, an interesting direction for future work is the detection of multiple salient objects. Moreover, combinations of different computational saliency methods with complementary properties can lead to improved accuracy of the computed saliency maps. Most promising saliency models [Pan et al., 2016; Liu and Han, 2016] are now built over ConvNets designed for semantic segmentation. An interesting direction could be to incorporate semantics in the model, instead of performing class agnostic saliency detection.

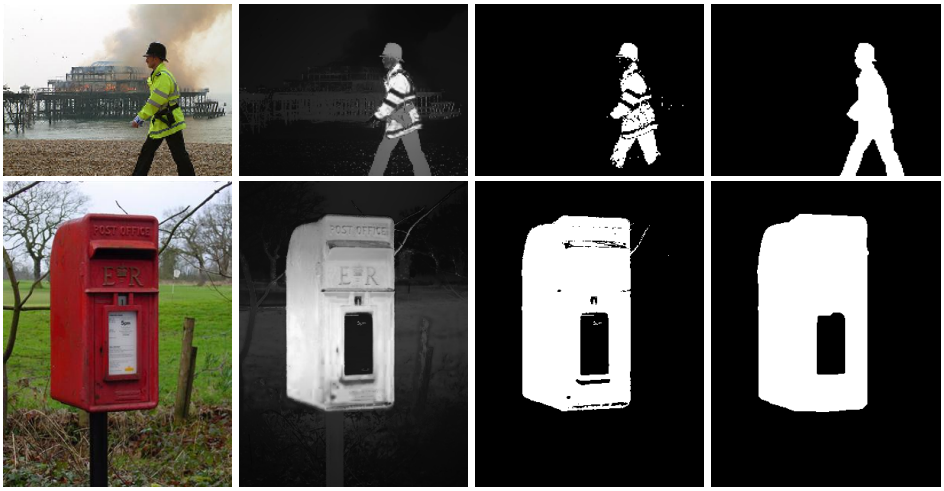


Figure 3.8: *Limitations of contrast based saliency (Section 3.1) and min-cut segmentation improvements. From left to right: Input image, saliency map computed with our method, the noisy result of simple thresholding, and min-cut segmentation applied to the saliency map.*

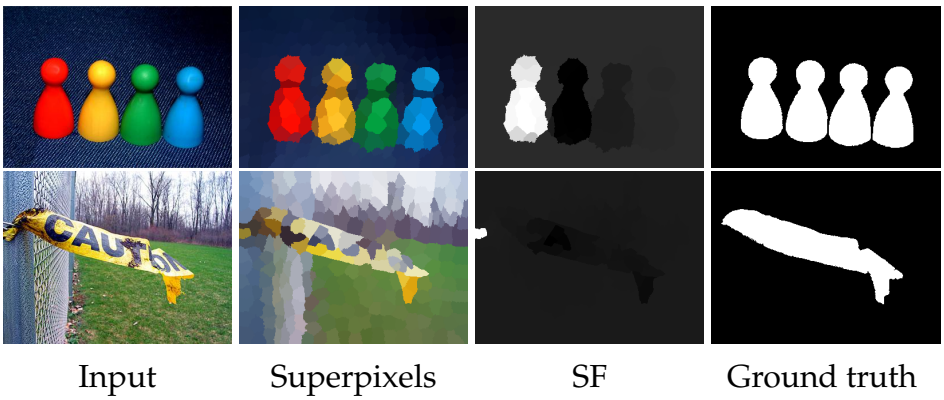


Figure 3.9: *Failure cases of saliency based on the Fiedler vector (Section 3.2). In the case of multiple disconnected objects our current algorithm correctly detects only the most salient one. Non-salient objects with distinctive colors cause the method to fail in some instances.*

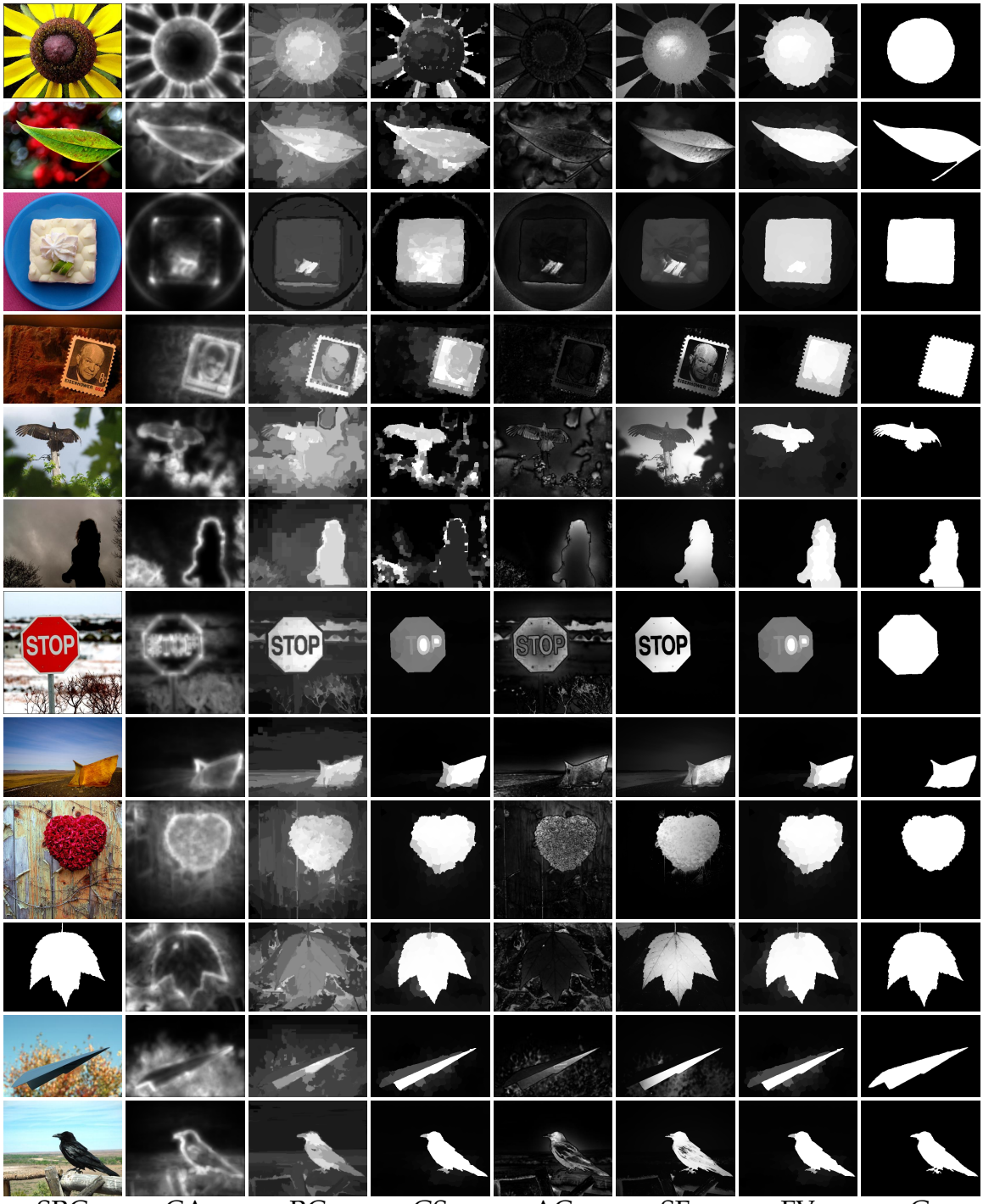


Figure 3.10: *Qualitative comparison of the results of our algorithms (SF, FV) with ground truth (GT) and several other state-of-the art approaches. Our methods consistently produce a foreground-background separation close to ground truth. Note their failure modes, when the prior they model does not hold. From left to right: source image (SRC), context-aware saliency (CA), global-contrast (RC), geodesic saliency (GS), salient region detection (AC), saliency filters (SF), fiedler saliency (FV) and ground-truth (GT).*

CHAPTER 4

Semi-automatic Segmentation with Object Proposals

In Chapter 3 we have discussed two different approaches to determine the location of a salient object in static images. While salient object detection techniques have demonstrated promising results applied on images containing a single foreground object, the subjective nature of saliency makes them less suited to process complex images with multiple objects. To this end, object proposals become handy. Object proposals indicate regions of an image that are likely to contain an object. Multiple proposals are extracted from a single image and assigned to an objectness score according to different heuristics, among them saliency. The striking advances that object proposals enabled in fields such as object recognition, have motivated their usage in several video object segmentation algorithms. These methods are often designed to operate unsupervised. An objectness score, which can incorporate saliency information, is assigned to each of the proposals and used to automatically select a set of temporally coherent segments that are likely to correspond to the foreground object. Typically the best proposal per-frame, is selected by minimizing an energy function defined over a locally connected spatiotemporal graph. While these methods have achieved state-of-art performance [Ma and Latecki, 2012; Li et al., 2013; Zhang et al., 2013], the sparse graph structure limits their ability to segment videos with fast motion and occlusions.

To overcome the aforementioned limitations of local graph connectivity, in this chapter, we propose an efficient alternative approach which exploits a fully connected spatiotemporal graph connecting all object proposals to

each other. The fully connected nature of the graph implies information exchange between both spatially and temporally distant object proposals, which makes our method robust to the difficult cases of fast frame-to-frame motion and object occlusions. We additionally propose an energy term that incorporates sparse but confident long range feature tracks, in order to ensure similar temporal labeling of objects. While previous approaches are constrained to the selection of one proposal per frame, our formulation enables the grouping of multiple overlapping proposals in the same frame, yielding robustness to outliers and incorrect proposal boundaries.

4.1 Method

Our method consists of three stages. Given an input video $\mathcal{V}$, for each frame $\mathcal{V}^t$ we compute a large set of object proposals $\mathcal{S}^t = \{s_i^t\}$, using existing techniques [Krähenbühl and Koltun, 2014]. The goal of this step is to generate a wide range of different proposals, such that a sufficient number of segments overlap with the object (§4.1.1). Then our method learns an SVM-based classifier in order to resample $\mathcal{S}$ into a smaller set of higher quality proposals $\bar{\mathcal{S}}$ (§4.1.1.1). Finally we refine this classification by solving for the maximum a posteriori inference on a densely connected CRF (§4.1.2). The fully connected graph structure is coupled with a novel energy function that considers overlap between point-tracks in the pairwise potentials, exploits temporal information, and ensures robustness to fast motion and occlusions.

4.1.1 Object Proposal Generation

Algorithms for computing object proposals are generally designed to have a high recall, proposing at least one region for as many objects in the image as possible. While the set of candidates must remain of limited size, the task of selecting positive samples is left to later stages, and the ratio of regions that truly belong to an object, *i.e.* precision, is usually not considered a measure of performance.

While other approaches leverage the high recall property by assuming that there is one good proposal per-frame, our goal is to exploit the redundancy in the data of multiple proposals with a high degree of overlap with the foreground object. In order to have a significant amount of such positive instances, we modified the parameters of Krähenbühl and Koltun [2014] that control seed placement and level set selection to generate around twenty thousands proposals per frame. Otherwise we consider the proposal generator as a black box, and other object proposal methods could be used instead.

It is important to note, however, that the resulting set of proposals is likely imbalanced, with potentially many more proposals on background regions than on foreground, depending on object size. Furthermore, many proposals will cover both foreground and background. These issues negatively impact segmentation, both in terms of quality and efficiency. To overcome this problem we train an SVM classifier and resample the pool of proposals.

4.1.1.1 Candidate Proposal Pruning

We introduce a per-frame pruning step with the goal of rebalancing the set of proposals and selecting only those with higher discriminative power, *i.e*. those that do not overlap both with foreground and background. The choice of the SVM is justified by its proven robustness to skewed vector spaces resulting from class imbalance [Wang and Japkowicz, 2008] and relatively fast performance. We train an SVM classifier which operates on elements of $\mathcal{S}$, separating those that overlap with foreground from those that belong to the background (§4.1.1.2), and then resample the set (§4.1.1.3). Finally, we use the output of the SVM to initialize the unaries of the CRF (§4.1.2.1).

4.1.1.2 Feature Extraction and Training

Features. From each of the proposals we extract a set of features that characterize its appearance, motion and objectness as summarized in Table 4.1. The global appearance and spatial support are defined in terms of average color, average position and area. The local appearance is encoded with Histogram of Oriented Gradients (HOG) [Lowe, 2004] computed over the proposal bounding box rescaled to 64x64 pixels and divided into 8x8, 50% overlapping cells quantized into 9 bins. The motion is defined with Histogram of Oriented Optical Flow (HOOF) [Chaudhry et al., 2009] extracted from the proposal bounding box rescaled to 64x64 pixels and quantized into 32 bins. The objectness is measured in terms of region boundaries encoded by 8x8 normalized gradients patches [Cheng et al., 2014]. The set of features is aggregated into a 1398 dimensional descriptor $\mathbf{x}_i \in \mathcal{X}$.

Training. The classifier is trained from a small set of proposals $\tilde{\mathcal{S}}$ known to belong to the foreground object. This set $\tilde{\mathcal{S}} = \{\tilde{s}_i\}$ may be either determined using automatic approaches such as the salient object detectors presented in Chapter 3, based on objectness [Endres and Hoiem, 2014; Lee et al., 2011], manually using interactive video editing tools, or a combination thereof. In our experiments we manually annotated 1 or 2 fore-

Feature	Description	Dim
(ACC)	Area, centroid, average color	6
(HOOF)	Histogram of Oriented Optical Flow	32
(NG)	Objectness via normalized gradients	64
(HOG)	Histogram of oriented gradients	1296

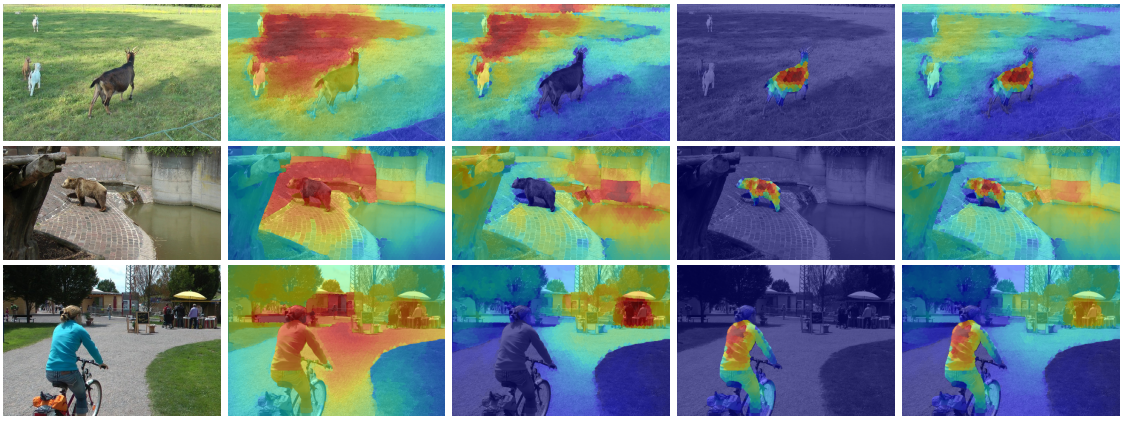
Table 4.1: *Set of features extracted from each object proposal by the SVM classifier and corresponding dimensionality.*

ground proposals per-sequence. $\tilde{\mathcal{S}}$ is augmented with all proposals that spatially overlap with one of its initial elements by a factor of more than a threshold τ (0.95 in our experiments). All remaining proposals are marked as background. A binary SVM classifier with linear kernel and soft margins is trained on the labeled data yielding the score function $\mathcal{C}(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i + b$ which measures the distance of the proposal $\tilde{s}_i$ with associated feature vector $\mathbf{x}_i$ from the decision surface $\mathbf{w}_\perp$. While $\text{sign}(\mathcal{C}(\mathbf{x}_i))$ is enough to classify proposals as either fore- or background, in Section 4.1.2 we can additionally include the distance from the hyperplane $\mathbf{w}^T\mathbf{x}_i + b \in [-\infty, +\infty]$ as the posterior probability $P(y_i|\mathbf{x}_i) \in [0, 1]$ in order to initialize the unary potentials of the CRF. We use *Platt Scaling* [Platt, 1999] to fit a logistic regressor $\mathcal{Q}$ to the output of the SVM and the true class labels, such that $\mathcal{Q}(\mathcal{C}(\mathbf{x}_i)) : \mathbb{R} \rightarrow P(y_i|\mathbf{x}_i)$. Parameters of the SVM are reported in Section 4.2.

4.1.1.3 Classification and Resampling

Given the trained classifier $\mathcal{C}$, we aim to roughly subdivide the set of object proposals $\mathcal{S}^t$ extracted at frame t into two spatially disjoint sets $\mathcal{S}^t_+$ and $\mathcal{S}^t_-$ such that $\bigcup \mathcal{S}^t_+$ lies within the foreground region and $\bigcup \mathcal{S}^t_-$ on the background. Initially we form $\mathcal{S}^t_+ = \{s^t_i | P(y_i|\mathbf{x}_i) > 0.5\}$. Next, we select elements from the set of proposals classified as background such that they do not overlap with $\mathcal{S}^t_+$, *i.e.*, $\mathcal{S}^t_- = \{s^t_i | \, |\mathcal{S}^t_+ \cap s^t_i| < \bar{\epsilon}\}$. The slack variable $\bar{\epsilon}$ is necessary to avoid $\mathcal{S}^t_- = \emptyset$, which can happen in videos where the foreground object occupies most of the frame. We initially set $\bar{\epsilon}$ to 0 and iteratively increment it with steps of 20 until the constraint $|\mathcal{S}^t_-| > 500$ is satisfied or the total amount of background proposal is reached. In our experiments we retain $\sim 10\%$ of the proposals generated (roughly 2000 proposals per-frame).

The positive impact of our pruning and resampling step on the quality of the video segmentation is shown in Section 4.3. The resulting classification can still be imprecise, but serves the purpose of rebalancing positive and



(a) Source (b) GOP (c) Background (d) Foreground (e) Resampled

Figure 4.1: *Left to right: distribution of object proposals on arbitrary frames. Colormaps are computed as the sum of the object proposals normalized to range [0,1]. Starting from a source image (a) we generate a set of geodesic object proposals with resulting distribution over the image (b). Note that many proposals fall on background regions. An SVM classifier (§4.1.1.2) resamples the set of proposals into foreground (c) and background (d). The new set (e), corresponding to the union of (c) and (d), is now balanced and contains proposals with higher discriminative power (§4.1.1.3).*

negative instances. The union of the two newly generated sets $\bar{\mathcal{S}}^t = \mathcal{S}_+^t \cup \mathcal{S}_-^t$ forms the input $\bar{\mathcal{S}} = \{\bar{\mathcal{S}}^t\}$ to the following step, which then provides a global solution considering spatial and temporal information jointly with the color appearance. Note that for ease of notation we refer to $\bar{\mathcal{S}}$ as $\mathcal{S}$ throughout the remaining part of paper. The original and newly generated distribution of proposals is visualized in Figure 4.1.

4.1.2 Fully Connected Proposal Labeling

In order to accurately classify elements of $\mathcal{S}$, we must enforce a smoothness prior that says that similar proposals should be similarly classified. Conditional random fields provide a natural framework to incorporate all mutual spatiotemporal relationships between proposals as well as our initial proposal confidences.

4.1.2.1 Inference

Let us define a set of labels $\mathcal{L} = \{\text{bg} = 0, \text{fg} = 1\}$, corresponding to background and foreground regions respectively. Let $\mathcal{F} = \{\mathbf{f}_i\}$ be a newly

generated set of features extracted from each element in $\mathcal{S}$, as defined in Eq. (4.3). Let us define the set of random variables $Y = \{y_i\}$, $y_i \in \mathcal{L}$. Consider a fully-connected random field $(Y, \mathcal{X} \cup \mathcal{F})$ defined over a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose nodes correspond to object proposals. Let $Z(\mathcal{X}, \mathcal{F})$ be the partition function. The posterior probability for this model is $P(Y|\mathcal{X}, \mathcal{F}) = \frac{1}{Z(\mathcal{F})} exp(-\mathrm{E}(Y|\mathcal{X}, \mathcal{F}))$ with the corresponding Gibbs energy defined over the set of all unary and pairwise cliques:

$$E(Y|\mathcal{X}, \mathcal{F}) = \sum_{i \in \mathcal{V}} \psi_u(y_i; \mathcal{X}) + \sum_{i,j \in \mathcal{E}} \psi_p(y_i, y_j; \mathcal{F}) \,. \tag{4.1}$$

Unary Potentials. The unary term ψ_u is directly inferred from the output of the SVM and the set of annotated proposals $\tilde{\mathcal{S}}$. We formulate an updated conditional probability $P(y_i|\mathbf{x}_i) = \lambda \cdot \mathcal{Q}(\mathcal{C}(\mathbf{x}_i)) + \frac{(1-\lambda)}{2}$, with the user-defined parameter $\lambda \in [0, 1]$ modulating the influence of the SVM prediction on the CRF initialization. For all experiments, we set the parameter λ to 0.1. We define ψ_u as a piecewise function

$$e^{-\psi_u(y_i, \mathcal{X})} = \begin{cases} l_i + \hat{\epsilon}, l_i \in \mathcal{L} & s_i \in \tilde{\mathcal{S}} \\ P(y_i|\mathbf{x}_i) & s_i \notin \tilde{\mathcal{S}} \end{cases} . \tag{4.2}$$

Pairwise Potentials. We define the label compatibility function μ to be the Potts model $\mu(y_i, y_j) = [y_i \neq y_j]$, a Gaussian kernel $k_*(x) = \exp(-\frac{x^2}{2\sigma_*^2})$, and scalar weights ω_*. In order to distinguish proposals that have similar appearance but belong to different image regions we define the pairwise potential ψ_p to be a linear combination of several terms that jointly incorporate color, spatial and temporal information:

$$\psi_p(y_i, y_j; \mathcal{F}) = [y_i \neq y_j] \cdot \big(\underbrace{\omega_c k_c(\mathcal{D}_c(c_i, c_j))}_{\text{appearance kernel}} + \underbrace{\omega_s k_s(\mathcal{D}_s(s_i, s_j))}_{\text{spatial kernel}} + \underbrace{\omega_p k_p(\mathcal{D}_p(p_i, p_j))}_{\text{trajectory kernel}} + \underbrace{\omega_t k_t(|t_i - t_j|)}_{\text{temporal kernel}} \big). \tag{4.3}$$

The color appearance $\mathcal{D}_c$ is defined in terms of the *chi-squared* kernel $\chi^2(c_i, c_j)$ where c_i and c_j are normalized RGB color histograms of proposals s_i and s_j, respectively, with 20 bins per dimension. The spatial relation between any pairs of proposals is defined in terms of the *intersection-over-union*: $\mathcal{D}_s(s_i, s_j) = 1 - \frac{|s_i \cap s_j|}{|s_i \cup s_j|}$. The last two kernels establish temporal connectivity among proposals, reducing the penalty of assigning different labels to those that are not intersected by the same trajectory or that belong to a different frame. The trajectory kernel exploits that the proposals we use consist of compact sub-regions in the form of superpixels. Let $p_i \subset s_i$ and $p_j \subset s_j$

be the set of superpixels that share at least one point-track with s_j or s_i, respectively. We define $\mathcal{D}_p$ based on the area that is intersected by common trajectories $\mathcal{D}_p(p_i, p_j) = 1 - \frac{|p_i \cup p_j|}{|s_i \cup s_j|}$. In the last term, t_i and t_j are the corresponding frame numbers of proposals s_i and s_j, which reduces penalty for assigning different labels to proposals that are distant in time. The *maximum a posteriori* (MAP) labeling of the random field $Y^* = argmax_{Y \in \mathcal{L}} P(Y|\mathcal{X}, \mathcal{F})$ minimizing the Gibbs energy $E(Y|\mathcal{X}, \mathcal{F})$ produces the segmentation of the video.

To efficiently recover Y^* we use the framework of Krähenbühl and Koltun [2011], which provides a linear time $O(N)$ algorithm for the inference of N variables on a fully-connected graph based on a mean field approximation to the CRF distribution. The efficiency of the method comes with the limitation that the pairwise potential must be expressed as a linear combination of Gaussian kernels having the form:

$$\psi_p(y_i, y_j, \mathcal{F}) = \mu(y_i, y_j) \sum_{m=1}^{K} w_m k_m(\mathbf{f}_i, \mathbf{f}_j) \tag{4.4}$$

where each Gaussian kernel defined as:

$$k_m(\mathbf{f}_i, \mathbf{f}_j) = \exp\left(-\frac{1}{2}(\mathbf{f}_i - \mathbf{f}_j)^T \Lambda_m (\mathbf{f}_i - \mathbf{f}_j)\right). \tag{4.5}$$

We now describe the embedding techniques we employ to project $\mathcal{F}$ into Euclidean space in order to overcome this limitation.

4.1.2.2 Euclidean Embedding

To enable the use of arbitrary pairwise potentials we seek a new representation of the data in which the l_2-norm is a good approximation to the distance of the original nonlinear space. In practice, given the original set of features $\mathcal{F}$ we seek a new embedding $\hat{\mathcal{F}}$ into the Euclidean space $\mathbb{R}^d$ s.t.:

$$\mathcal{D}(\mathbf{f}_i, \mathbf{f}_j) \approx \left|\left|\hat{\mathbf{f}}_i - \hat{\mathbf{f}}_j\right|\right|_2. \tag{4.6}$$

Campbell et al. have demonstrated the effectiveness of Landmark Multidimensional Scaling (LMDS) [de Silva and Tenenbaum, 2002] in a context similar to ours. LMDS is an efficient variant of Multidimensional Scaling [Cox and Cox, 1994] that uses the Nystrom approximation [Belongie et al., 2002a] to reduce the complexity from $O(N^3)$ to $O(Nmk + m^3)$ where N is the number of points, m is the number of landmarks and k the dimensionality of the new space. We refer the reader to [Campbell et al., 2013; Platt, 2005] for more details.

Stage	Time
Optical flow	113.1
Object Proposals	55.6
Feature Extraction	541.7
SVM Classification	42.7
MDS Embedding	78.4
CRF Inference	260.0
Video Segmentation	1091.5

Table 4.2: *Running time in seconds for each individual stage to segment a video of 75 frames and spatial resolution 960x540.*

We use LMDS to conform the pairwise potential to Eq. (4.4). We express pairwise potentials ψ_p in Eq. (4.3) as a linear combination of several terms. For better control of the resulting embedding error, we separately embed each of the components. For each $\mathcal{D}_*$ term of Eq. (4.3), we empirically determine the dimensionality of the embedding space from the analysis of their dissimilarity matrix eigenvalues. The resulting pairwise potential conforming to Eq. (4.4) is:

$$\psi_p(y_i, y_j; \hat{\mathcal{F}}) = [y_i \neq y_j](\omega_c k_c(\hat{c}_i, \hat{c}_j) + \omega_s k_s(\hat{s}_i, \hat{s}_j) + \omega_p k_p(\hat{p}_i, \hat{p}_j) + \omega_t k_t(t_i, t_j)). \quad (4.7)$$

The features $\hat{c}$, $\hat{s}$, $\hat{p}$ are Euclidean vectors of 10, 20 and 50 dimensions respectively. Note that the temporal term t is already Euclidean, and so it does not require embedding.

4.1.2.3 Segmentation

The final video segmentation is computed as the sum of the proposals weighted by the conditional probability $P(y = \text{fg}|\mathcal{X}, \hat{\mathcal{F}})$ and scaled to range $[0, 1]$ on a per-frame basis. As a final post-processing step, we refine the segmentation with a median filter of width 3 applied along the direction of the optical flow [Brox and Malik, 2010]. This has the effect of removing temporal instability that arises from different per-frame object proposal configurations. The final segmentation can then be thresholded by β to achieve a binary mask.

4.2 Implementation Details

We conducted all experiments on a machine with 2 Intel Xeon 2.20 GHz processors with 8 cores each. The algorithm has been implemented in Python. For the SVM-based pruning we employ the implementation of *scikit-learn* [Pedregosa et al., 2011]. Most of the components of our algorithm are parallelizable. Those that are not, such as MDS and the CRF, are relatively efficient. In Table 4.2 we report the time consumption of each individual component for a sample video of 75 frames and resolution of 960x540. It takes about 20 minutes to complete the segmentation which is about 16 seconds per frame. The running time performance of our algorithm is comparable to the fastest existing methods such as [Ramakanth and Babu, 2014; Papazoglou and Ferrari, 2013; Faktor and Irani, 2014]. The weights of the CRF pairwise potential ψ_p of Eq. (4.3) are specific to the dataset. For FBMS we used $\omega_c = 1.0$, $\omega_s = 0.15$, $\omega_p = 0.3$ and $\omega_t = 0.2$, while for SegTrack we reduced the impact of spatial-temporal relationships between proposals setting $\omega_s = \omega_t = 0.01$. The proposal generation step uses 200 seeds, 200 level sets, with the rejection overlap set to 0.95. The only necessary modification of parameters was a reduction of the number of proposals for the evaluation of the CRF step only (without proposal pruning), which we discuss in detail below. For that experiment, we reduced the number of proposals using 30 seeds, 30 level sets and rejection threshold of 0.88.The parameter β that binarizes the final segmentation is set empirically to 0.03 for FBMS and 0.07 for SegTrack.

4.3 Results

We quantitatively evaluate our approach and its components with respect to various state-of-the-art techniques on the *Freiburg-Berkeley Motion Segmentation Dataset* (FBMS [Brox and Malik, 2010]), see Figure 4.2 for qualitative results. Further evaluation of the proposed approach is provided in Chapter 7.

FBMS Results. The FBMS dataset consists of 59 sequences featuring typical challenges of unconstrained videos such as fast motion, motion blur, occlusions, and object appearance changes. The dataset is split into a training and testing set. Since none of the methods we compare with requires a training phase, we measure performance on both sets. Due to running-time and memory constraints of the prior approaches that we compare to, we limit the length of the videos to 75 frames. For the purpose of testing

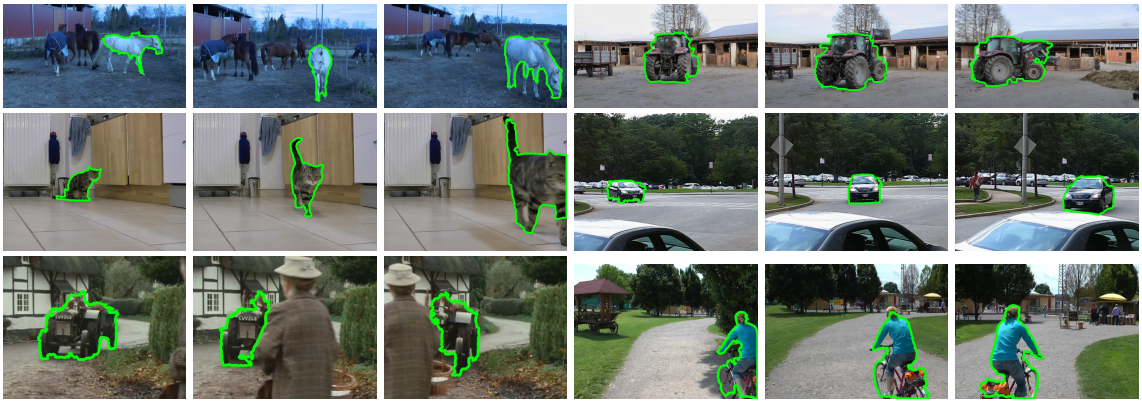


Figure 4.2: *Top to bottom, left to right: qualitative video object segmentation results on six sequences (horses05, farm01, cats01, cars4, marple8 and people5) from the FBMS dataset. Our method demonstrates reasonable segmentation quality for challenging cases, e.g., non-rigid motion and considerable appearance changes (horse05, cats01). The rich set of features of the SVM and the pairwise potentials of the CRF make our method robust to cluttered background (farm1, cars4), while the fully connected graph on which we perform inference provides robustness to partial and full occlusions (marple8). The aggregation of object proposals is also effective for complex, multi-colored objects (people05).*

segmentation quality in the presence of fast motion we temporally subsample frames from videos exhibiting slow motion. In videos that have multiple objects we manually selected the one with dominant motion. Similar to previous works [Li et al., 2013; Faktor and Irani, 2014] we measure the segmentation quality in terms of *intersection-over-union*, which is invariant to image resolution and to the size of the foreground object. We compare our method (FCP) with several recent state-of-the-art approaches [Papazoglou and Ferrari, 2013; Ramakanth and Babu, 2014; Zhang et al., 2013; Faktor and Irani, 2014]. These methods have been selected based on their quality of results, underlying approaches, and availability of their source code. *SeamSeg* (SEA), seeks connected paths of low energy to track the object boundaries. Zhang et al. [2013] (DAG) integrate objectness and appearance similarity in a directly acyclic graph whose shortest-path corresponds to a video segmentation. Under the assumption that the object moves differently from the background Papazoglou and Ferrari [2013] (FST) find the closed motion boundary and propagate the initial estimate using a spatio-temporal optimization. Finally, Faktor and Irani [2014] (NLC) consolidate an initial foreground estimate based on saliency using a Markov chain.

Our method and SEA are semi-supervised while the others are unsuper-

	FCP	CRF	SVM	SEA	FST	DAG	NLC
cars1	69.0	80.0	68.0	**83.0**	82.0	10.0	27.0
cats01	**83.0**	68.0	76.0	62.0	80.0	34.0	71.0
cats03	39.0	00.0	11.0	17.0	**53.0**	32.0	12.0
dogs01	55.0	22.0	39.0	38.0	53.0	**56.0**	54.0
goats01	82.0	**84.0**	78.0	53.0	**84.0**	79.0	58.0
horses05	**77.0**	66.0	47.0	69.0	34.0	44.0	38.0
lion01	**84.0**	74.0	80.0	73.0	80.0	77.0	67.0
marple2	59.0	57.0	71.0	**78.0**	65.0	56.0	60.0
marple4	**88.0**	73.0	87.0	69.0	15.0	45.0	19.0
marple6	77.0	64.0	77.0	**86.0**	24.0	18.0	48.0
people1	68.0	64.0	22.0	58.0	54.0	69.0	**85.0**
people2	81.0	78.0	76.0	77.0	**92.0**	48.0	77.0
rabbits02	66.0	11.0	33.0	42.0	65.0	32.0	**71.0**
rabbits03	43.0	40.0	23.0	42.0	41.0	22.0	**44.0**
rabbits04	29.0	00.0	12.0	23.0	**38.0**	12.0	20.0
tennis	48.0	27.0	41.0	55.0	30.0	51.0	**64.0**
Avg. Test	**65.0**	51.0	53.0	58.0	56.0	43.0	51.0
Avg. Training	**77.0**	62.0	61.0	71.0	68.0	60.0	56.0

Table 4.3: *Intersection-over-union comparisons on a subset of the FBMS dataset. The columns SVM and CRF correspond to the results obtained using either only our SVM-based classification, or only our CRF-based labeling, respectively. Our full approach (FCP) is generally (close to) the best performing one (highlighted in bold), and achieves the highest average values of all methods.*

vised. For a more informative and fairer comparison, we therefore removed any of the videos from the comparison in Table 4.3, for which at least one of these unsupervised methods did not detect the object. We report detailed sequence evaluation for the test set and the average for the training set. We separately evaluate the steps of our algorithm: SVM only, CRF only, and the full approach FCP. Corresponding precision, recall, and f-measure plots are shown in Figure 4.3. As discussed in the implementation section, in the CRF experiment we modified the parameters generating object proposals to produce roughly the same number of proposals that are retained during the pruning step.

Results in Table 4.3 demonstrates that our method consistently produces a good segmentation yielding roughly a 10% improvement over the current state-of-the-art in terms of average performance. The importance of combining both the SVM and CRF steps is also apparent.

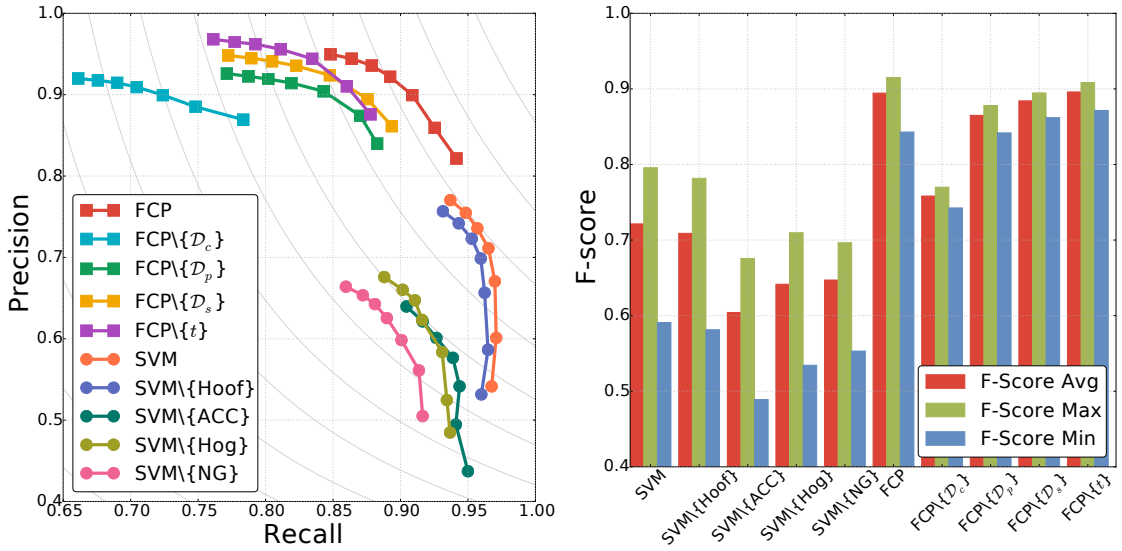


Figure 4.3: *Left: Precision-Recall curves and F-score isolines ($\beta^2 = 0.3$) for the SVM and CRF classification of object proposals into foreground and background, obtained by varying the minimum amount of overlap τ required for a proposal to be considered foreground. The SVM classification (SVM) is less precise but has better recall, preventing the removal of foreground proposals, with which the CRF can perform the final classification (FCP). The plot also evaluates the individual importance of the SVM features of Table 4.1 and the CRF potentials of Eq. (4.3) in terms of the resulting loss if they were removed during the classification. Right: Average, maximum and minimum F-score. Our solution FCP outperforms the SVM only classification. Note that the best scores, respectively SVM and FCP, are obtained when all features and potentials are employed.*

4.4 Discussion

We presented a novel approach to segment objects in unconstrained videos, which provides state-of-the-art performance on challenging video data. During these studies we realized that, due to the constant increase in terms of video resolution and quality, more complex benchmarks than FBMS are required to provide real-world application scenarios for evaluating video segmentation algorithms. Therefore, in Chapter 7 we propose a new dataset comprising 50 densely annotated, high-resolution video sequences. Additional evaluation on the new benchmark demonstrated that methods based on object proposals appear to be a great candidate for addressing the computational challenges arising from higher resolution video data, since the use of proposals greatly reduces computational complexity, allowing us to

	FCP	NLC	FST	DAG	TMF	KEY	HVS
birdfall	25.0	**74.0**	59.0	71.0	62.0	49.0	57.0
cheetah	49.0	**69.0**	28.0	40.0	37.0	44.0	19.0
girl	54.0	**91.0**	73.0	82.0	89.0	88.0	32.0
monkeydog	64.0	78.0	**79.0**	75.0	71.0	74.0	68.0
parachute	91.0	94.0	91.0	94.0	93.0	**96.0**	69.0
Average	57.0	**81.0**	66.0	72.0	70.0	70.0	49.0

Table 4.4: *Intersection-over-union computed on the SegTrack dataset. On low resolution video, there are insufficient foreground proposals generated for our method to work well.*

employ a fully connected CRF over a *complete* video sequence. A similar, fully connected formulation at the pixel level would be infeasible.

Limitations and Future Works. Our approach is designed to work with real-world video sequences with fast object motion, and occlusions. In particular, since our method is based on object proposals, it requires a sufficiently high video resolution such that the computation of proposals using existing techniques produces meaningful results. This becomes clear in Table 4.4, when running our approach on lower resolution video such as the SegTrack benchmark[1] [Lee et al., 2011; Papazoglou and Ferrari, 2013]. We additionally compare with Li et al. [2013] (TMF), Lee et al. [2011] (KEY) and Grundmann et al. [2010] (HVS). In this dataset, very few proposals overlap with the foreground object due to the limited image resolution (highest is 414x320) and the small size of the objects, so our approach, which is based on aggregating multiple object hypothesis works less well. For example the training set of the *birdfall* video has a ratio of 1:4000 foreground and background proposals, with only 13 proposals on the object. The lack of positive samples weakens the self-training of the SVM and, as consequence, the effectiveness of the CRF is severely limited. While some of the results are comparable with other approaches, these limitations are the reason why our method performs significantly better on the FBMS dataset.

There exist several further opportunities for followup work. For example, to improve the final segmentation accuracy, it would be interesting to investigate approaches to combine the prediction of the CRF in a more principled manner (*e.g.*, incorporating higher-order potentials), or to employ bilateral filtering techniques to refine the proposal-based segmentation to the pixel level.

[1] In accordance with prior works we do not evaluate *'penguin'*

CHAPTER 5

Learning Video Segmentation from Static Images

In the previous chapter we explored the usage of object proposals and hand-crafted features to segment foreground objects. We demonstrated that solving a global energy minimization problem over the entire video volume have several advantages, such as robustness to occlusion, fast-motion and appearance changes, and overall it shows lower performance decay and good temporal stability. However processing the entire video at once might not always be feasible due to the memory footprint required by the algorithm. In contrast to Chapter 4, in this part of the thesis we investigate how accurate video object segmentation can be enabled by training a Convolutional Neural Network (ConvNet) on static images only and processing the video sequence on a frame-by-frame basis.

Many fundamental areas of computer vision have recently witnessed dramatic progresses thanks to the rise of deep learning techniques coupled with the availability of large-scale annotated data. However, in the domain of video segmentation, densely annotating a large-scale dataset is almost prohibitive. Motivated by the lack of per-pixel labeled video data, we demonstrate that highly accurate video object segmentation can be enabled using a ConvNet trained with *static images* only. This is one of key insights and contribution of this chapter. Furthermore, our method is efficient due to its feed-forward architecture and can handle different type of initialization, ranging from precise segmentation masks, to weaker object localizations such as a bounding-boxes. These two characteristics render our approach well suited to parse large-scale datasets.

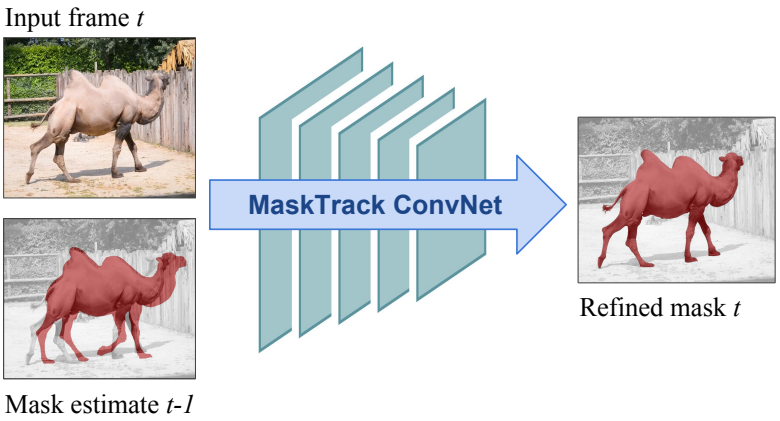


Figure 5.1: *Given a rough mask estimate from the previous frame $t-1$, we train a ConvNet to provide a refined mask output for the current frame t.*

Inspired by recent advances of deep learning in instance segmentation and object tracking in this chapter we present a novel approach that adapt a semantic segmentation network to class agnostic video object segmentation. To this end, we introduce the concept of guidance, to steer the model to focus on the desired instance of the object. The propose approach is semi-supervised but can handle different type of input annotations, such as segments or bounding boxes, and therefore is suitable for a variety of application that require different manual effort.

To the best of our knowledge our approach is the first to use a pixel labelling network, *e.g.* DeepLabv2 [Chen et al., 2016], for the task of video object segmentation. We name our approach `MaskTrack`.

5.1 Method

We tackle the video objects segmentation problem from a new perspective. We build upon an existing architecture designed for semantic pixel labeling and adapt it to generate per-frame segments of generic object instances. Specifically, we choose DeepLabv2 [Chen et al., 2016] as our baseline ConvNet, as at the time of these studies it was one of the best performer on Pascal VOC [Everingham et al., 2012] and the code was publicly released. Nevertheless, our approach can be easily built around any other architectures. Given a ConvNet trained for instance segmentation, the challenge is then how to inform the network to segment a particular object instance. We tackle this problem with two complementary strategies. One is guiding the network towards the instance of interest, stacking a rough segmentation into the RGB input image. The imperfect input segmentation can be either

a manual annotation, *e.g.* in the case of the first video frame of a sequence, or the previous frame mask. The network learns to refine the segmentation mask during the offline training (§5.1.1). The second strategy learns the appearance of the specific instance to be segmented by fine-tuning the model with an online training procedure (§5.1.2). In the ablation study (§5.3.2) we demonstrate that the combination of online and offline training yields highly accurate results. A visual representation of the proposed approach is shown in Figure 5.1.

5.1.1 Offline Training

To provide the ConvNet with guidance towards the object of interest to be segmented, we extend the data layer of the network from 3-channels RGB to 4-channels RGB+mask. The augment channel has the purpose of providing the network with a rough shape and location of the object instance. The network refines this prior estimate into a high-quality segmentation. Therefore we can interpret our model as a "mask-refinement" network (Figure 5.1).

Two key observations make the approach we propose appealing from a practical point of view. First, it doesn't require highly accurate segmentation masks to be initialized. In our experiments, we found that a coarse input masks and even simple bounding boxes, as demonstrated in the ablation study (§5.3.2) are enough for the trained network to produce sensible output results. Second, our approach does not require densely annotated video data and therefore we are able to exploit a large set of diverse images and avoids having to use existing video segmentation benchmarks for training.

A simplified architecture of our system is shown in Figure 5.1. In order to make the ConvNet robust to inaccurate masks priors during offline training, we generated novel input masks deforming the ground-truth mask. We employ affine and non-rigid transformations via thin-plate splines [Bookstein, 1989] to simulate the motion the object might undergoes in the successive frames. Furthermore we coarsen the input mask throughout morphological dilation to remove small details of the object contour. The coarsening generates a mask that is a better representative of the test time data, simulating the blobby shape of the output ConvNet mask. The aforementioned sets are necessary to increase robustness to noisy input and to reduce error accumulation from the preceding frames.

An exemplar deformed mask is shown in Figure 5.2. Note that we apply this procedure during the offline training, over an ensemble dataset of $\sim 10^4$ images. During test-time we only coarsen output mask estimate at time $t-1$,

Annotated Image Example training masks

Figure 5.2: *Examples of training mask generation. From one annotated image, multiple training masks are generated. The generated masks mimic plausible object shapes on the preceding frame.*

with dilation and use that as rough mask for frame t, but we do not apply any transformation.

The outcome of the offline training is a ConvNet similar to DeepMask [Pinheiro et al., 2015] and Hypercolumns [Hariharan et al., 2015], that takes a coarse input mask as guidance instead of a bounding box. We provide further training details such as parameters in Section 5.2. The offline trained network achieve competitive performance compared to state-of-the-art on several benchmarks (§5.3.2), however in our studies we found possible to improve the results encoding into the network the appearance knowledge of the specific object instance to be segmented. We call this online training strategy and we discuss it in the next section.

5.1.2 Online Training

For further boosting the video segmentation quality, we borrow and extend ideas that were originally proposed for tracking.

Inspired by current to performing tracking techniques [Danelljan et al., 2016; Nam and Han, 2016] that all exploits different nuances of online training, we adopt the same strategy to improve the segmentation quality. In practice, in a semi-automatic setting test-time, we are given at least an annotated frame that indicates to our method which object is to be segmented. The idea is to use this information as additional training data and to fine-tune the network to incorporate the appearance of the specific object instance.

In order to fine-tune the network, we augment the annotate data available with the same strategies we used to improve robustness to noisy input during the offline training (§5.1.1). Having obtained multiple variants of the same annotation we proceed fine-tuning our model.

While fine-tuning, in theory, could be applied recursively frame after frame,

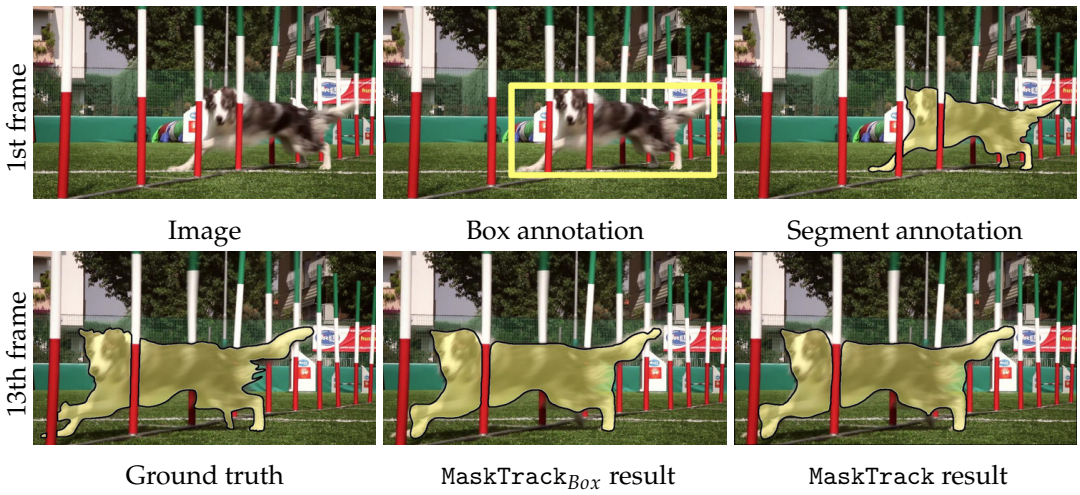


Figure 5.3: *By propagating annotation from the 1st frame, either from segment or just bounding box annotations, our system generates results comparable to ground truth.*

this, would not only drastically increase the running time, but might lead to over-fitting and drifting from the foreground object to background. Therefore, in our experiments we only do fine-tuning using the manually annotated frame(s). The details of the online fine-tuning are provided in Section 5.2.

5.1.3 Variants

In this section we consider variances of the proposed model. We demonstrate the flexibility of our approach to handle different different type and levels input annotations. Furthermore we show, how motion information could be seamless integrated, improving the quality of the segmentation. In particular motion helps to disambiguate foreground moving objects from static background.

Box annotation. Our system can simply handle bounding boxes as initialization, by replacing the manual segmentation of the first frame with a rectangle having the bounding box coordinates of the object. To improve performance we re-train a model named MaskTrack$_{Box}$, that takes a bounding box as input and output a segmentation mask. Therefore to segment videos annotated with a bounding box, we use MaskTrack$_{Box}$ on the first frame, and then proceed from the second frame onward with the standard MaskTrack model.

Motion. Inter-frame motion is an additional source of information that

RGB Images

Optical flow magnitude

Figure 5.4: *Examples of optical flow magnitude images.*

can be used to guide the segmentation. Given a video sequence, we compute the optical flow using EpicFlow [Revaud et al., 2015] with Flow Fields matches [Bailer et al., 2015] and convolutional boundaries [Maninis et al., 2016]. Therefore, a second segmentation using as input the motion magnitude, is computed using the same model trained on images. To match the tensor dimension of the input, the motion magnitude is replicated along the depth dimension, into the three channels. Despite the model being trained on RGB images, it can be employed to process motion magnitude as it resembles a gray-scale object (Figure 5.4) and thus, it captures the object shape. The RGB and motion magnitude output are later fused by averaging the probability scores. We name this variant `MaskTrack+Flow`. As demonstrate by the results of Section 5.3, inter-frame motion provides a complementary information to `MaskTrack` with RGB images.

5.2 Network implementation and training

In all experiments, we employ the publicly available implementation of DeepLabv2-VGG network [Chen et al., 2016]. DeepLabv2-VGG was originally designed for semantic segmentation of static images and at the time of this research it was one of the best performing approaches, on Pascal VOC [Everingham et al., 2012]. The model is initialized from the VGG16 architecture, pre-trained on ImageNet [Simonyan and Zisserman, 2015]. In order to accommodate the input mask, we extend the first convolutional layer by increasing the dimensionality along the depth. Weights for the extra-channel

are randomly initialized sampling from a gaussian distribution with zero mean and unit variance.

Offline training. As discussed in Section 5.1 our method does not employ any pixel-labeled video dataset for training. Instead we use annotated images from an ad-hoc ensemble of salient object segmentation datasets. Specifically we combine the following datasets: ECSSD [Shi et al., 2016], MSRA10K [Cheng et al., 2015], SOD [Movahedi and Elder, 2010], and PASCAL-S [Li et al., 2014], yielding an aggregated total of 11 282 training images. To deform the input masks we use affine transformation with random scaling ($\pm$5% of object size) and translation ($\pm$10% shift). Furthermore, we apply non-rigid deformations via thin-plate splines [Bookstein, 1989] placing 5 control points along the annotated shape contours and randomly shifting the points in xy direction within sampling from a uniform distribution of in range $\pm$10%. Eventually, we coarsened the mask using morphological operations such as dilation and erosion with a 5 pixel radius. We keep the training parameters reported in [Chen et al., 2016], specifically we use SGD with mini-batches of 10 images and a polynomial learning policy with initial learning rate of 0.001. The momentum and weight decay are set to 0.9 and 0.0005. The network is trained for 20k iterations.

Online training. The online adaptation is performed fine-turning, on the first-frame, the model previously trained offline an the saliency dataset. The online training runs for a total of 200 iterations with training samples obtained from the first frame annotation. We introduce diversity in the samples by performing the same type of augmentations and the same parameters, applied in during the offline training, for a total of $\sim 10^3$ variations of the first annotated image. Amortizing the online training time over the entire video sequence, the proposed approach runtime is 12 seconds per frame, which is a magnitude faster of compared to other state-of-the-art approaches such as `ObjFlow` [Tsai et al., 2016].

5.3 Results

In this paragraph, we investigate the relevance of the different components of our techniques (§5.3.2) and report both a quantitative and qualitative comparison with respect to the state-of-the-art over three well established datasets (§5.3.3).

5.3.1 Experimental setup

Datasets. To evaluate the proposed approach we employ three different datasets: DAVIS (§7), YoutubeObjects [Prest et al., 2012], and SegTrackv2 [Li et al., 2013]. The union of these datasets, provide us with a diverse set of challenges such as fast-motion, occlusions, small-resolution and multiple objects instances.

DAVIS is the dataset and benchmark we propose in Chapter 7. Briefly, it comprises a total of 50 high quality videos. The dataset comes with binary, spatio-temporally dense per-pixel annotations separating foreground object(s) from background. Further evaluation of `MaskTrack` on this dataset is bundled together with the evaluation of the approaches proposed in Chapter 4 and Chapter 6 and detailed in Section 7.4.

YoutubeObjects [Prest et al., 2012] was designed around 10 object categories. Following previous evaluation protocols we evaluate our approach on a subset of 126 videos with more than 20 000 frames, for which temporally sparse, pixel-level ground truth segmentation masks are provided by [Jain and Grauman, 2014].

SegTrackv2 [Tsai et al., 2010] is a smaller dataset, containing only 14 videos for a total of 24 target objects to segment. Similarly to DAVIS, each frame comes with a per-pixel annotation. In the case of sequences containing multiple target objects, each instance is assigned a different label. We process each instance separately.

Evaluation. Following the procedure of Section 4.3, to measure the quality of our results, we employ the *intersection-over-union metric* or *Jaccard Index*, computed, on a per-frame basis, over the estimated segmentation and the provided ground-truth. The per-frame results are first averaged per-sequence and then over the entire dataset. Consistently with previous approaches, we exclude the first frame from the evaluation of YoutubeObjects and SegTrackv2 [Tsai et al., 2010], while on DAVIS the last frame is also dismissed.

Since previous works were not coherent with the evaluation protocols previously described, we re-computed the scores ourself. Specifically, we collected new results for `ObjFlow` [Tsai et al., 2016] and `BVS` [Maerki et al., 2016] to ensure consistency in the results.

5.3.2 Ablation study

In this section we investigate the performance gain produced by the different components of our approach. Experiments are performed over the

DAVIS dataset and measured using the intersection-over-the-union measure (mIoU). In Table 5.1 we report the contribution of each component described in Section 5.1. Results demonstrate that each individual component is beneficial and improves the accuracy of the `MaskTrack` model.

Add-ons. Combining our base model (`MaskTrack`) with complementary information such as inter-frame motion (§5.1.3) improves the results from 74.8% to 78.4% mIoU. Despite optical flow producing a sensible boost for the DAVIS dataset we found it to provide inconsistent gains across all the three datasets and therefore, in order to provide a single solution with fixed parameters, we do not include a per-dataset optimized optical flow in the results of Section 5.3.3.

Besides optical flow, post-processing the output of the ConvNet with a CRF [Krähenbühl and Koltun, 2011] improves the performance by $\sim$2%, reaching $\sim$80% mIoU on DAVIS.

Training. In this paragraph we study the contribution of training the network online and offline. In our experiments we found that disabling online fine-tuning substantially reduce the performance of $\sim$5% mIoU. If instead we skip offline training and only rely on online fine-tuning performance drops drastically, albeit the absolute quality (57.6 mIoU) is surprisingly high for a system trained on ImageNet for classification and on a single frame for class agnostic segmentation.

In our experiments we note that the amount of data is not critical and decreasing the number of training images from 11k to 5k yields only a minor decrease in terms of mIoU.

Finally we test offline training on video data. We train our model on SegTrack and YoutubeObjects and evaluate the results on DAVIS. We report a minor descrease in mIoU which could explained with by insufficient diversity and the domain shift between different datasets, validating once more the usage of static images for training video object segmentation tasks.

Mask deformation. Exploring the contribution of coarsening and non-rigidly deforming the input masks, we found that both strategies provide a decent gain. However, as demonstrated in Table 5.1, it is the absence of any form of deformation that is mostly critical for our approach. Therefore, deforming input masks is essential to make our model robust to noisy input segmentation at test time.

Input channels. As discussed in Chapter 1, accurate manual segmentation are expensive to obtain. Therefore we study the effect of varying the extra-channel input with a form of weaker annotation *i.e.* bounding boxes and

Aspect	System variant	mIoU	ΔmIoU
Add-ons	`MaskTrack+Flow+CRF`	**80.3**	+1.9
	`MaskTrack+Flow`	78.4	+3.6
	`MaskTrack`	*74.8*	-
Training	No online fine-tuning	69.9	−4.9
	No offline training	57.6	−17.2
	Reduced offline training	73.2	−1.6
	Training on video	72.0	−2.8
Mask deformation	No dilation	72.4	−2.4
	No deformation	17.1	−57.7
	No non-rigid deformation	73.3	−1.5
Input channel	Boxes	69.6	−5.2
	No input	72.5	−2.3

Table 5.1: *Ablation study of our* `MaskTrack` *method on DAVIS. Given our full system, we remove one component at a time, to understand each individual contribution. See §5.3.2 for discussion.*

found the performance to be comparable. Most interestingly, however, we tested our approach without the additional input channel, therefore operating in the modality of a salient object detector. The competitive results indicate that the fine-tuning is able to capture the appearance of the object.

5.3.3 Evaluation

As demonstrated in Table 5.2, `MaskTrack` obtains competitive performance across all three datasets. The results are obtained using purely our feed-forward network trained on the same data and sharing the same parameters across the three datasets. `JOTS` [Wen et al., 2015] achieved a higher score of (71.3 mIoU) on SegTrackv2. However they tune the parameters per video and therefore their results are not comparable with our fix-parameters setup.

In Table 5.2 we report the results for the variant of our base model that is initialized with a bounding-box annotation instead of a segmentation mask. We refer to this variant as $\texttt{MaskTrack}_{Box}$ (§5.1.3). Despites the annotation being weaker, the performance loss is only minor and our method ranks among the top three best results in all datasets.

We note that, adding components specifically trained for different datasets, we can further boost the performance of our technique. Specifically, adding

	DAVIS	YoutbObjs	SegTrackv2
Box oracle	45.1	55.3	56.1
Grabcut oracle	67.3	67.6	74.2
`ObjFlow` [Tsai et al., 2016]	71.4	70.1	67.5
`BVS` [Maerki et al., 2016]	66.5	59.7	58.4
`NLC` [Faktor and Irani, 2014]	64.1	-	-
`FCP` [Perazzi et al., 2015b]	63.1	-	-
`W16` [Wang et al., 2016]	-	59.2	-
`Z15` [Zhang et al., 2015]	-	52.6	-
`TRS` [Xiao and Lee, 2016]	-	-	**69.1**
`MaskTrack`	**74.8**	**71.7**	67.4
$\texttt{MaskTrack}_{Box}$	73.7	69.3	62.4

Table 5.2: *Video object segmentation results on three datasets. Compared to related state-of-the-art, our approach provides consistently good results. On DAVIS the extended version of our system* `MaskTrack+Flow+CRF` *reaches* 80.0 *mIoU. See §5.3.3 for details.*

optical flow and CRF post-processing (§5.1.3) we obtain a score of 80.0 mIoU on DAVIS, 72.6 on YoutubeObjects and 70.3 on SegTrackv2

Qualitative evaluation of our approach is provided in Figure 5.5

Attribute-based analysis. Table 5.3 presents an attribute-based evaluation on DAVIS (§7). Video attributes represent challenging factors and allow us to identify groups of videos with a dominant feature *e.g.*, presence of occlusions, which is key to explaining the algorithms' performance. The attribute based analysis shows that our generic model, `MaskTrack`, is robust to various video challenges present in this dataset. It compares favourably on any subset of videos sharing the same attribute, except camera-shake, where `ObjFlow` [Tsai et al., 2016] marginally outperforms our approach. We observe that `MaskTrack` handles fast-motion and motion-blur well, which are typical failure cases for methods relying on spatio-temporal connections [Tsai et al., 2016].

Due to the online fine-tuning on the first frame annotation of a new video, our system is able to capture the appearance of the specific object of interest. This allows it to better recover from occlusions, out-of-view scenarios and appearance changes, which usually affect methods that strongly rely on propagating segmentations on a per-frame basis.

Incorporating optical flow information into `MaskTrack` substantially increases robustness on all categories. As one could expect,

	Method, mIoU				
Attribute	BVS	ObjFlow	MSK	MSK+Flow	MSK+Flow+CRF
Appearance change	46.0	54.0	**65.0**	75.0	76.0
Fast-motion	53.0	55.0	**66.0**	74.0	75.0
Background clutter	63.0	68.0	**77.0**	78.0	79.0
Camera-shake	62.0	**72.0**	71.0	77.0	78.0
Dynamic background	60.0	67.0	**69.0**	75.0	76.0
Deformation	70.0	77.0	**77.0**	78.0	80.0
Edge ambiguity	58.0	65.0	**68.0**	74.0	74.0
Heterogeneous object	63.0	66.0	**71.0**	77.0	79.0
Interacting objects	63.0	68.0	**74.0**	75.0	77.0
Low resolution	59.0	58.0	**60.0**	75.0	77.0
Motion blur	58.0	60.0	**66.0**	72.0	74.0
Occlusion	68.0	66.0	**74.0**	75.0	77.0
Out-of-view	43.0	53.0	**66.0**	71.0	71.0
Scale variation	49.0	56.0	**62.0**	72.0	73.0
Shape complexity	67.0	69.0	**71.0**	72.0	75.0

Table 5.3: *Attribute based evaluation on DAVIS.*

MaskTrack+Flow+CRF better discriminates cases involving color ambiguity and salient motion. However, we also observed less-obvious improvements in cases with scale-variation and low-resolution objects.

5.4 Conclusion

In this chapter we presented a ConvNet based approach to video object segmentation. Introducing the concept of guidance, we adapted an architecture designed for semantic image segmentation into a class agnostic object segmenter.

We demonstrated that a combination of offline and online trained on static images yields highly accurate results on three heterogeneous datasets, while sharing the same parameters among all video sequences. Our approach is efficient in terms of running time and versatile enough to employ different type of annotations such as bounding boxes. The ablation study reveals the effect of each components.

This is one of the first approaches to use ConvNet for video object segmentation. The state-of-the-art performance leads us to believe more sophisticated

Figure 5.5: *Qualitative results of three different datasets. Our algorithm is robust to challenging situations such as occlussions, fast motion, multiple instances of the same semantic class, object shape deformation, camera view change and motion blur.*

networks should be investigated, especially, those such as LSTM, that could better leverage the previous frames' information.

CHAPTER 6

Interactive Segmentation in Bilateral Space

In Chapter 5 we describe a method that produces a fairly precise segmentation given minimal user input. However the quality of the results does not reach yet the level of accuracy that is required for specific applications such as movie post-production. Therefore for such application scenarios video segmentation must be addressed interactively, with the user providing feedback in order to correct erroneous estimates of the underlying algorithm.

A crucial aspect of interactive video segmentation methods is *responsiveness*. A user expects instant feedback, and any computation delay present significant challenges to the adoption of these technologies. This is one of the key reasons that segmentation related tasks, such as rotoscoping, form the bulk of manual labor, and therefore associated costs, of video effects. In this chapter, we present a highly efficient method for user-guided video segmentation that is able provide iterative feedback in a fraction of the time of previous approaches, while still generating high quality results in semi-supervised applications, as demonstrated on multiple benchmarks.

We accomplish this by performing the segmentation in "bilateral space", which is a high dimensional feature space, originally proposed for accelerated bilateral filtering [Chen et al., 2007], and recently extended to computing depth from stereo triangulation [Barron et al., 2015]. We describe a novel energy on a "bilateral grid" [Chen et al., 2007], a regular lattice in bilateral space, and infer labels for these vertices by minimizing an energy using graph cuts. Processing on the bilateral grid has several advantages over other approaches. First, the regular and data-independent structure al-

lows for a more efficient mapping from image to bilateral space (and vice versa) than super-pixels or k-means clustering approaches. Second, it allows for flexible interpolation schemes that lead to soft assignments of pixels to multiple intermediate variables. And finally, a bilateral representation allows us to infer labels on a simple, locally connected graph, while still enforcing large spatio-temporal neighborhood regularization, which would be intractable to solve directly. We show that the combination of these advantages significantly improves segmentation quality, and importantly, allows us to segment video data, generating temporally consistent results with robustness to object and camera motion.

6.1 Method Overview

Let $\mathcal{V} : \Omega \rightarrow \mathbb{R}^3$ be a color video, defined on a finite discrete domain $\Omega \subset \mathbb{R}^3$. Given some user input as a set of known foreground and background pixels, $FG, BG \subset \Omega$, we seek a binary mask $\mathcal{M} : \Omega \rightarrow \{0, 1\}$ that labels each pixel of the video either as background or foreground.

Our approach makes use of a bilateral grid [Chen et al., 2007], Γ, consisting of regularly sampled vertices $\mathbf{v} \in \Gamma$. The mask $\mathcal{M}$ is computed in four main stages, Figure 6.1: by *lifting* pixels into a higher dimensional feature space (§6.1.1), *splatting* them onto regularly sampled vertices (§6.1.2), computing a *graph cut* label assignment (§6.1.3), and *slicing* vertex labels back into pixel space (§6.1.4).

6.1.1 Lifting

The first step is to embed each pixel $\mathbf{p} = [x, y, t]^\mathsf{T}$ in a higher d-dimensional feature space, for example by concatenating YUV pixel color and spatial and temporal coordinates:

$$\mathbf{b}(\mathbf{p}) = \left[c_y,\, c_u,\, c_v,\, x,\, y,\, t\right]^\mathsf{T} \in \mathbb{R}^6 \tag{6.1}$$

In this bilateral space, Euclidean distance encodes both spatial proximity and appearance similarity. We evaluated a number of feature spaces, generalized as the concatenation of appearance features $\mathcal{A}(\mathbf{p})$ and position features $\mathcal{P}(\mathbf{p})$, and interestingly found that state-of-the-art results can be achieved by simply extending traditional 5D bilateral features with a temporal dimension, which is very efficient due to the low dimensionality.

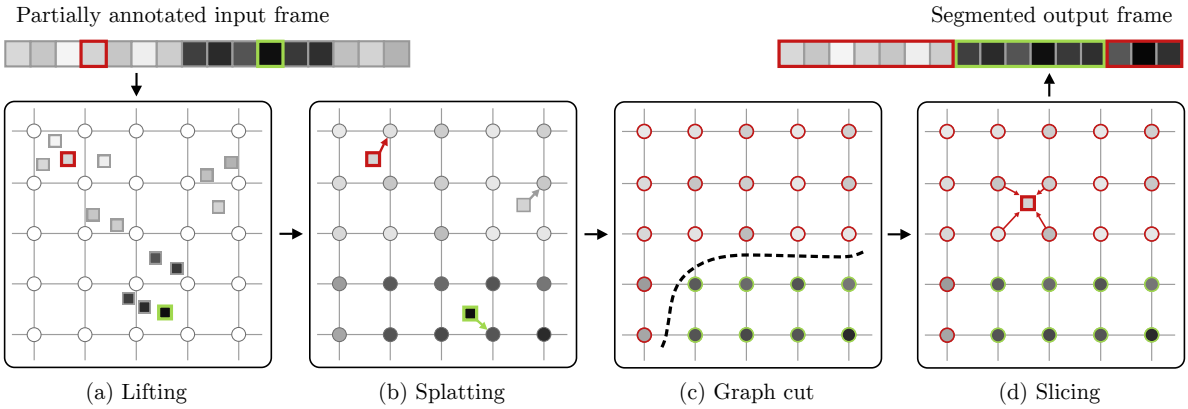


Figure 6.1: *Our pipeline, demonstrated on a 1D example. Pixels are lifted into a 2D feature space (a), with two user assigned labels (red and green highlighted pixels). Values are accumulated on the vertices of a regular grid (b), a graph cut label assignment is computed on these vertices (c), and finally pixel values are sliced at at their original locations (c), showing the final segmentation (again, red and green boundaries).*

6.1.2 Splatting

Instead of labeling each lifted pixel $\mathbf{b}(\mathbf{p})$ directly, we resample the bilateral space using a regular grid [Chen et al., 2007; Barron et al., 2015] and compute labels on the vertices of this grid. The process of accumulating values on the bilateral space vertices is known as "splatting". For each vertex $\mathbf{v} \in \Gamma$, a weighted sum of lifted pixels $\mathbf{b}(\mathbf{p})$ is computed as:

$$S(\mathbf{v}) = \sum w(\mathbf{v}, \mathbf{b}(\mathbf{p})) \cdot (\hat{\mathbf{p}}) \tag{6.2}$$

where

$$\hat{\mathbf{p}} = (\mathbb{1}_{FG}(\mathbf{p}), \mathbb{1}_{BG}(\mathbf{p}), 1) \tag{6.3}$$

and $\mathbb{1}_{\times}(\mathbf{p})$ is an indicator function that is 1 iff $\mathbf{p} \in \times$.

The weight function $w(\mathbf{v}, \mathbf{b}(\mathbf{p}))$, determines the range and influence that each lifted pixel $\mathbf{b}(\mathbf{p})$ has on the vertices of Γ. Prior work has used a nearest neighbor (NN) indicator [Barron et al., 2015] or multi-linear interpolation weights [Chen et al., 2007]. Importantly, these approaches have limited support, (1 nonzero vertex for each pixel using NN, and 2^{d-1} for multi-linear), which is necessary for computation and memory efficiency. The NN approach is the fastest, but can lead to blocky artifacts, while the multi-linear interpolation is slower, but generates higher quality results. We propose an *adjacent* interpolation that provides a good compromise between the two, yielding high quality results, but with a linear growth in the number of non-

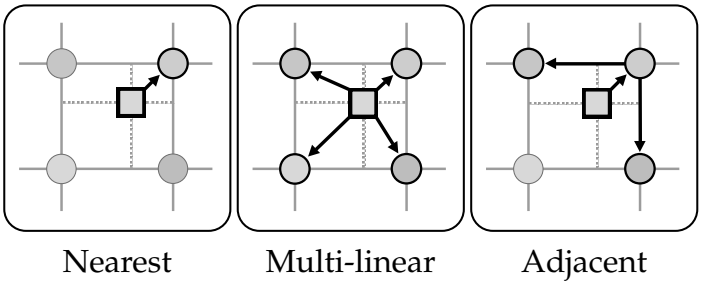


Figure 6.2: *Different interpolation schemes. Adjacent interpolation scales significantly better to higher dimensionality when compared to multi-linear interpolation, with only a small reduction in quality.*

zero weights as a function of feature space dimension, as opposed to the exponential growth of the multi-linear case (Figure 6.2).

The idea behind adjacent weighting is that with multi-linear interpolation, weights quickly decrease for vertices that differ from the nearest neighbor $N_{\mathbf{b}(\mathbf{p})}$ in many dimensions. More precisely,

$$w_l(\mathbf{v}, \mathbf{b}(\mathbf{p})) \leq 0.5^{\left|\mathbf{v} - N_{\mathbf{b}(\mathbf{p})}\right|_0} \tag{6.4}$$

presents an upper bound for the weight, because each factor of the linear interpolation is smaller than 0.5 if for that dimension $\mathbf{v}_i$ is not the integer value that $\mathbf{b}_i(\mathbf{p})$ was rounded to. We use this bound to skip weight computation where the result would have been small anyway:

$$w_a(\mathbf{v}, \mathbf{b}(\mathbf{p})) = \begin{cases} \prod_{i=1}^{d} \left|\mathbf{v}_i - N_{\mathbf{b}(\mathbf{p})}\right| & \text{if } \mathbf{v} \in A_{\mathbf{b}(\mathbf{p})} \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

We found that interpolation between the nearest neighbor and vertices that differ in only one dimension (the set of adjacent vertices $A_{\mathbf{b}(\mathbf{p})}$) already produces significantly better results than hard nearest neighbor assignments with only a minor increase in runtime.

6.1.3 Graph Cut

We now seek binary labels $\boldsymbol{\alpha}$, that mark each vertex $\mathbf{v}$ as foreground, $\alpha_{\mathbf{v}} = 1$, or background, $\alpha_{\mathbf{v}} = 0$.

We compute these labels by constructing a graph $\mathcal{G} = (\Gamma, \mathcal{E})$ where the vertices are the vertices in the bilateral grid, and edges connect immediate neighbors (e.g., 4 neighbors when $d = 2$, 6 neighbors when $d = 3, \ldots$). We

then define an energy based on the assumption that the label assignment is smooth in bilateral space:

$$E(\boldsymbol{\alpha}) = \sum_{\mathbf{v} \in \Gamma} \theta_{\mathbf{v}}(\mathbf{v}, \alpha_{\mathbf{v}}) + \lambda \sum_{(\mathbf{u},\mathbf{v}) \in \mathcal{E}} \theta_{\mathbf{uv}}(\mathbf{u}, \alpha_{\mathbf{u}}, \mathbf{v}, \alpha_{\mathbf{v}}) \tag{6.6}$$

$\theta_{\mathbf{v}}$ is the unary term, $\theta_{\mathbf{uv}}$ is the pairwise term, and λ is a weight that balances the two.

The unary term $\theta_{\mathbf{v}}$ models deviations from the supplied user input. As we invert the splatting step to retrieve final pixel labels, the splatted value $S_{BG}(\mathbf{v})$ expresses the total cost of assigning $\mathbf{v}$ to foreground, $\alpha_{\mathbf{v}} = 1$, and $S_{FG}(\mathbf{v})$ the cost of assigning it to background, $\alpha_{\mathbf{v}} = 0$, respectively.

$$\theta_{\mathbf{v}}(\mathbf{v}, \alpha_{\mathbf{v}}) = (1 - \alpha_{\mathbf{v}}) \cdot S_{FG}(\mathbf{v}) + \alpha_{\mathbf{v}} \cdot S_{BG}(\mathbf{v}) \tag{6.7}$$

The pairwise term $\theta_{\mathbf{uv}}$ attempts to ensure that neighboring vertices are assigned the same label. In order to derive $\theta_{\mathbf{uv}}$, we consider that the bilateral space graph $\mathcal{G}$ is equivalent to a densely connected pixel graph, where edge weights between pixels assigned to the same vertex are set to infinity (as it is impossible to assign them different labels in bilateral space). The edge weight between other pixels is then approximated by the distance of their respective vertices. With that in mind, it becomes clear that the weights between vertices need to be scaled by the total number of points $S_{\#}(\mathbf{u})$ and $S_{\#}(\mathbf{v})$ that have been assigned to the two vertices (we can retrieve $S_{\#}(\mathbf{u})$ and $S_{\#}(\mathbf{v})$ from the homogeneous (3rd) coordinate in Eq. (6.4)). That way, assigning different labels to two vertices is (approximately) equivalent to assigning the labels to all the original points and our pairwise term can be written as:

$$\theta_{\mathbf{uv}}(\mathbf{u}, \alpha_{\mathbf{u}}, \mathbf{v}, \alpha_{\mathbf{v}}) = g(\mathbf{u}, \mathbf{v}) \cdot S_{\#}(\mathbf{u}) \cdot S_{\#}(\mathbf{v}) \cdot [\alpha_{\mathbf{u}} \neq \alpha_{\mathbf{v}}] \tag{6.8}$$

where $g(\mathbf{u}, \mathbf{v})$ is a high-dimensional Gaussian kernel where the diagonal matrix Σ scales each dimension to balance color, spatial and temporal dimensions:

$$g(\mathbf{u}, \mathbf{v}) = e^{-\frac{1}{2}(\mathbf{u}-\mathbf{v})^{\mathsf{T}} \Sigma^{-1} (\mathbf{u}-\mathbf{v})} \tag{6.9}$$

This formulation also reduces the complexity of the graph cut due to the fact that all vertices without any assigned pixels, $S_{\#}(\mathbf{v}) = 0$, are now completely excluded from any computation and thus need no representation in the graph. We can now efficiently apply a max-flow computation to find the vertex labeling with minimal energy [Boykov et al., 1999].

Connectivity analysis. So far we have assumed that increased connectivity leads to higher quality results. We validate this by conducting experiments where we compute a graph cut segmentation on a *per-pixel* (not

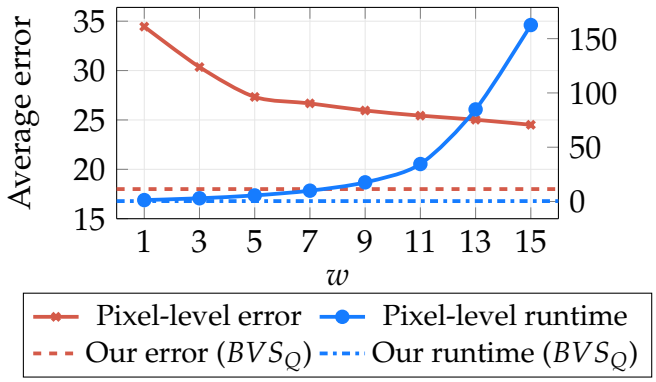


Figure 6.3: *Mask propagation on a pixel-level graph with increasing neighborhood sizes w. Error decreases with larger neighborhoods at the expense of larger runtimes. Our approach (BVS_Q) is shown for comparison. We obtain lower error than even large window sizes, while being much faster as well.*

bilateral) graph, as in [Boykov and Jolly, 2001]. We begin with just local neighbor edges (4 neighbors on a 2D graph), and increase the connectivity by connecting all points in an $n \times n$ window (Figure 6.3). This plot clearly shows that increasing connectivity leads to better results, but at an increased running time [Faktor and Irani, 2014; Choi et al., 2012; Li et al., 2013].

6.1.4 Slicing

Given the foreground and background labels of the bilateral vertices, the final mask $\mathcal{M}$ is retrieved by slicing, i.e. interpolating grid labels at the positions of the lifted pixels in the output frame. We generally use the same interpolation scheme for both splatting and slicing, although a even more precise adjustment of the quality/speed trade-off is possible by choosing different interpolations.

$$\mathcal{M}(\mathbf{p}) = \sum_{\mathbf{v} \in \Gamma} w(\mathbf{v}, \mathbf{b}(\mathbf{p})) \cdot L(\mathbf{v}) \tag{6.10}$$

Finally, we post-process each frame with a simple 3×3 median filter in order to remove minor high frequency artifacts that arise due to the solution being smooth in bilateral space, but not necessarily pixel space, however we note that a more sophisticated method like the geodesic active contours of [Fan et al., 2015] could also be applied.

	BVS_Q (quality)	BVS_S (speed)
Feature space	YUV XY T	YUV XY T
Intensity grid size	35	15
Chroma grid size	30	10
Spatial grid size	$w/35, h/35$	$w/50, h/50$
Temporal grid size	2	2
Interpolation	Linear	Adjacent
Runtime	$0.37s$	$0.15s$

Table 6.1: *The parameters for two different configurations used for the evaluation.*

6.2 Results

Implementation. Our approach is implemented in Matlab, with C++ bindings for most time consuming routines. All our experiments were performed on a Mac Pro with a 3.5 GHz 6-Core Intel Xeon E5 CPU and 16 GB RAM. The measured timings include the complete pipeline except for IO-operations. Unlike many other approaches, we do not rely on pre-computed flows, edge maps or other information.

Parameters. We evaluate two different sets of settings, one tuned for quality, BVS_Q, and the other for speed, BVS_S, parameters are listed in Table 6.1. In the remaining part of this thesis, for simplicity, we will reference to BVS_Q as BVS. Our method can predict temporally global segmentations, and higher temporal resolutions allow for compensating for large degrees of object motion. However, this did not improve result quality on the benchmarks due to limited object motion, and the testing strategy of Fan et al. [2015], where a single keyframe is propagated forward by multiple frames. In cases where user input is distributed temporally, e.g., in the interactive interface, we use a higher temporal grid size of $N = 5, \ldots, 15$.

We set the pairwise weight to $\lambda = 0.001$ for all results. The lifting stage also allows for different feature dimensions to be scaled independently of each other (Σ in Equation 6.9). For all results, we scale by 0.01, 0.5, 1.3, 1.5 the temporal (t), spatial (xy), the intensity (c_y) and the chroma ($c_u c_v$) dimensions respectively, but we didn't notice any particular dependency on the unary edge factor or the dimension scaling. All parameters could be tuned to achieve better results per benchmark or even per video, but we leave them fixed in all tests to represent a more real-world scenario.

Runtime. Comparing runtime is difficult, with different code bases and levels of optimization, however, we give some average runtimes from our ob-

	BVS_Q	BVS_S	SEA	JMP	NLC	HVS
480p	0.37*s*	0.15*s*	6*s*	12*s*	20*s*	5*s*
1080p	1.5*s*	0.8*s*	30*s*	49*s*	20*s*	24*s*

Table 6.2: *Approximate running time per frame for a number of fast methods with code available. Ours is roughly an order of magnitude faster than prior methods, and scales linearly with image size. NLC has mostly constant running time because it uses a fixed number of superpixels.*

servations as a rough idea of the expected computational complexity. As many existing video segmentation methods take even up to one hour for a single frame, we compare only with the following fastest state-of-the-art methods: SEA: SeamSeg [Ramakanth and Babu, 2014], JMP: JumpCut [Fan et al., 2015], NLC: Non-Local Consensus Voting [Faktor and Irani, 2014], and HVS: Efficient Hierarchical Graph-Based Video Segmentation [Grundmann et al., 2010].

Our method computes 480p masks in as little as 0.15 seconds (Table 6.2) which is roughly an order of magnitude faster than all other approaches. Even if we trade speed for quality, our method still takes significantly less time than the second-fastest approach. Furthermore, the two most expensive steps, i.e. lifting and slicing, can be trivially parallelized since their output values only depends on color and position of individual pixels. Splatting can also be performed on concurrent threads, simply augmenting the grid with a small number of accumulators at bilateral vertices. The only stage that is not easily parallelizable is graph-cut, which anyway has small runtime due to the size and sparsity of the bilateral grid. Therefore we would expect a tuned GPU implementation to report substantial performance gains.

6.2.1 Quantitative Evaluation

In order to evaluate our approach with respect to existing methods, we focus on the task of *mask propagation*, which has been widely used by previous work. Given a manual segmentation of the first frame, each method predicts subsequent frames, without any additional user input. Using this approach, we measured the performance on three different benchmark datasets. In this section we discuss results on *JumpCut* and *SegTrack* datasets, while in Chapter 7 we provide the evaluation based on our proposed dataset and evaluation protocol DAVIS.

JumpCut. The recent method of Fan et al. [2015] includes a dataset consist-

	BVS_Q	BVS_S	RB	DA	SEA	JMP
animation	**0.78**	1.77	1.98	1.26	1.83	1.59
bball	**1.36**	3.29	1.55	1.71	1.90	1.61
bear	1.34	1.56	1.82	**1.07**	1.84	1.36
car	1.01	5.48	1.35	1.38	0.73	**0.54**
cheetah	**2.72**	3.56	7.17	3.99	5.07	4.41
couple	2.65	6.43	4.09	3.54	3.78	**2.27**
cup	**0.99**	4.54	3.72	1.34	1.19	1.16
dance	**5.19**	23.96	6.65	9.19	7.55	6.62
fish	**1.78**	4.06	2.80	1.97	2.54	1.80
giraffe	4.06	9.89	8.49	6.99	4.77	**3.83**
goat	2.68	4.87	3.68	2.57	3.30	**2.00**
hiphop	**3.21**	8.08	8.02	4.62	6.94	3.37
horse	3.60	16.32	3.99	4.14	3.00	**2.62**
kongfu	**1.97**	2.51	5.42	3.71	5.78	3.28
park	**2.35**	5.89	3.95	3.49	3.33	2.93
pig	2.15	3.18	3.86	**2.08**	3.39	2.97
pot	**0.62**	1.25	0.94	1.49	0.80	0.70
skater	**4.72**	11.23	6.33	5.33	5.09	4.89
station	2.07	8.55	2.53	2.01	2.37	**1.53**
supertramp	9.68	9.76	14.70	8.99	17.40	**6.17**
toy	0.66	7.16	1.02	1.32	0.70	**0.58**
tricking	**4.23**	5.57	42.20	9.71	11.90	5.02
Average	**2.72**	6.77	6.19	3.72	4.33	2.78

Table 6.3: *Errors (lower is better) on the JumpCut benchmark for two transfer distances and several different methods as reported by Fan et al. [2015].*

ing of 22 videos with medium resolution and good per-frame ground truth masks. In addition to the methods mentioned above, we compare to RB: RotoBrush, based on SnapCut [Bai et al., 2009], and DA: Discontinuity-aware video object cutout [Zhong et al., 2012]. As we do not have access to implementations for all methods reported on this dataset, we instead adapt our method to conform to the same testing strategy and error metric used in [Fan et al., 2015]. That is, propagating masks from multiple keyframes $0, 16, \ldots, 96$, over different transfer distances $(1, 4, 8, 16$ frames), and reporting error as follows:

$$Err = \frac{100}{n} \sum_{i=1}^{n} \frac{\text{\# error pixels in i-th frame}}{\text{\# foreground pixels in i-th frame}} \tag{6.11}$$

Overall, our method performs best on this benchmark, closely followed by JumpCut (Table 6.3).

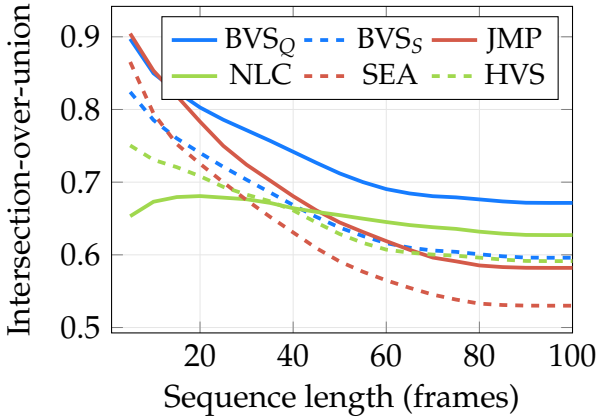


Figure 6.4: *This plot shows how IoU (higher is better) decreases when a* single *mask is propagated over increasing numbers of frames. Our method degrades favorably when compared to other approaches. The NLC approach stays constant as it is an automatic method that doesn't depend on the input of the first frame.*

We note that our approach uses a simple refinement step (3x3 median filter). However, we conducted an experiment using an active contour refinement, similar to JumpCut, and our result improved to 2.45 on average, with a running time of only 1*s* per frame. We additionally observe that many methods degrade in quality over long sequences, as errors accumulate over time. In contrast, our method scores better on long videos, experiencing less drift of the object region than other approaches (Figure 6.4).

SegTrack. For the sake of completeness we also present an evaluation on the popular benchmark of Tsai et al. 2010. We additionally compare to: FST: Fast Object Segmentation in Unconstrained Video [Papazoglou and Ferrari, 2013], DAG: Video object segmentation through spatially accurate and temporally dense extraction of primary object regions [Zhang et al., 2013], TMF: Video segmentation by tracking many figure-ground segments [Li et al., 2013], and KEY: Key-segments for video object segmentation [Lee et al., 2011]. In this case, it can be seen from Table 6.4 that our method clearly struggles to compete with existing approaches. This is most likely due to a combination of factors related to the low quality and resolution of the input videos, which lead to many mixed pixels that confuse the bilateral model. We also note that many of these methods were optimized with this dataset in mind, using different parameter settings per *sequence*. Instead, we use the same parameter settings for all three datasets. We also believe that the more recent datasets from JumpCut and our additional videos provide a more contemporary representation of video segmentation tasks.

	BVS_Q	BVS_S	NLC	FST	DAG	TMF	KEY	HVS
birdfall	66.0	40.0	**74.0**	59.0	71.0	62.0	49.0	57.0
cheetah	10.0	14.0	**69.0**	28.0	40.0	37.0	44.0	19.0
girl	89.0	87.0	**91.0**	73.0	82.0	89.0	88.0	32.0
monkeydog	41.0	38.0	78.0	**79.0**	75.0	71.0	74.0	68.0
parachute	94.0	92.0	94.0	91.0	94.0	93.0	**96.0**	69.0
Average	60.0	54.0	**81.0**	66.0	72.0	70.0	70.0	49.0

Table 6.4: *Comparison of our method on the SegTrack dataset, using the IoU metric (higher is better).*

Figure 6.5: *Qualitative video segmentation results from three sequences of* DAVIS *(horsejump, stroller and soapbox). The segmentation is computed non-interactively, given the first frame as initialization. Our method demonstrates robustness to challenging scenarios such as complex objects, fast-motion, and occlusions.*

6.2.2 Interactive Segmentation

It is important to note that while our method scores well on these two higher-resolution benchmarks, the real advantage is the fast running time, when used in an interactive framework. To demonstrate this, we built a simple prototype editor (Figure 6.6) in Matlab that allows a user to draw strokes on an image to mark foreground or background regions. After every stroke, the newly marked pixels are splatted to the bilateral grid and a global spatio-temporally solution is computed. Finally, the mask is sliced from the current frame and its outline is overlaid on the image.

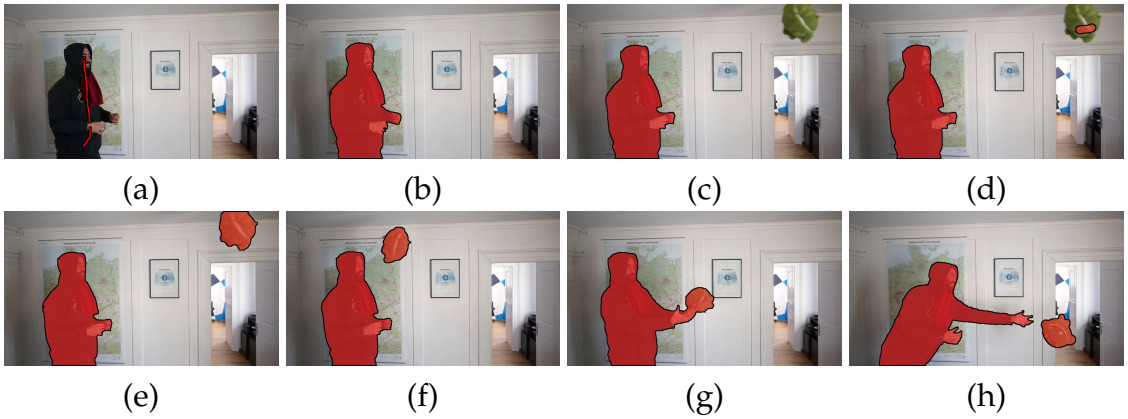


Figure 6.6: *Our interactive segmentation editor. Very simple input (a) is sufficient to infer an accurate foreground mask (b) and track it over time. As a new object enters the scene (c), the user can choose to add it to the foreground with an additional input stroke (d). The mask is then automatically propagated to to the other frames (e-h) without further corrections.*

6.3 Discussion

In this chapter, we have shown how simple and well-understood video segmentation techniques leveraging graph cuts can yield state-of-the-art results when performed in bilateral space.

There are many exciting avenues for extending the research in this area. For example, one could consider alternate, more descriptive feature spaces in the lifting step. We made some initial experiments with using patches, and obtained marginally better results, but at the expense of higher running time. Additionally, while the bilateral representation can handle some degree of motion, it does not explicitly account for camera or object motion. One possibility is to warp pixels using their optical flow before splatting. Our initial experiments indicated that due to the instability of flow, such methods were unreliable; sometimes leading to large improvements in quality, but in other times made the results worse. These methods also rely on precomputing optical flow, which is costly. Nonetheless, explicitly exploring scene motion is a promising venue to future work.

Despite this, we believe that the method as presented here has many attractive qualities. It is simple to implement, parallelizable, and fast, all without sacrificing quality. This efficiency gain is not only vital to providing faster feedback to users, but is also important for extending to low computational power (mobile) devices, or large scale (cloud-based) problems, which will hopefully enable new applications.

CHAPTER 7

Dataset and Evaluation Methodology

In the previous chapters we presented novel approaches aiming to discover and separate foreground objects from the background region of a video. The resulting pixel-level, spatio-temporal bipartition of the video is instrumental to several applications including, action recognition, object tracking, video summarization, or rotoscoping for video editing. Despite remarkable progress in recent years, video object segmentation still remains a challenging problem and most existing approaches still exhibit severe limitations in terms of quality and efficiency to be applicable in practical applications, *e.g.*for processing large datasets, or video post-production and editing in the visual effects industry.

What is most striking is the performance gap among state-of-the-art video object segmentation algorithms and closely related methods focusing on image segmentation and object recognition, which have experienced remarkable progress in the recent years. A key factor bootstrapping this progress has been the availability of large scale datasets and benchmarks [Russakovsky et al., 2014; Martin et al., 2001; Everingham et al., 2010]. This is in stark contrast to video object segmentation. While several datasets exists for various different video segmentation tasks [Tsai et al., 2010; Li et al., 2013; Tron and Vidal, 2007; Brostow et al., 2009; Badrinarayanan et al., 2010; Gorelick et al., 2007; Fathi et al., 2011; Ren and Philipose, 2009; Grundmann et al., 2010; Prest et al., 2012; Brox and Malik, 2010; Sundberg et al., 2011], none of them targets the specific task of video *object* segmentation.

To date, the most widely adopted dataset is SegTrack [Tsai et al., 2010], which, however, was originally proposed for joint segmentation and track-

Figure 7.1: *Sample sequences from our dataset, with ground truth segmentation masks overlayed. Please refer to Figure 7.6 for the complete dataset.*

ing and only contains six low-resolution video sequences, which are not representative anymore for the image quality and resolution encountered in today's video processing applications. As a consequence, evaluations performed on such datasets are likely to be overfitted, without reliable indicators regarding the differences between individual video segmentation approaches, and the real performance on unseen, more contemporary data becomes difficult to determine [Butler et al., 2012]. Despite the effort of some authors to augment their evaluation with additional datasets, a standardized and widely adopted evaluation methodology for video object segmentation does not yet exists.

To this end, we introduce a new dataset DAVIS (Densely Annotated VIdeo Segmentation) specifically designed for the task of video object segmentation. The dataset is publicly available and contains fifty densely and professionally annotated high-resolution Full HD video sequences, with pixel-accurate ground-truth data provided for every video frame. The sequences have been carefully captured to cover multiple instances of major challenges typically faced in video object segmentation. The dataset is accompanied with a comprehensive evaluation of the techniques proposed in this thesis and several other state-of-the-art approaches [Papazoglou and Ferrari, 2013; Ramakanth and Babu, 2014; Brox and Malik, 2010; Fragkiadaki et al., 2012; Shen et al., 2015; Faktor and Irani, 2014; Lee et al., 2011; Taylor et al., 2015; Chang et al., 2013; Fan et al., 2015; Grundmann et al., 2010]. To evaluate the performance we employ three complementary metrics measuring the spatial accuracy of the segmentation, the quality of the silhouette and its temporal coherence. Furthermore, we annotated each video with specific attributes

Dataset	SIZE	HD-Q	VARY	DENSE-GT	OBJ
DAVIS	✓	✓	✓	✓	✓
MoSeg [Brox and Malik, 2010]	✓				✓
BVSD [Sundberg et al., 2011]	✓	✓	✓		
SegTrack [Tsai et al., 2010]				✓	✓
SegTrack v2 [Li et al., 2013]			✓	✓	✓

Table 7.1: *Summary of requirements fulfilled by datasets most relevant to video object segmentation. From left: large overall size of the dataset (SIZE), high-resolution videos (HD-Q), variety of content and challenges (VARY), pixel-accurate, per-frame ground-truth (DENSE-GT) and object presence (OBJ). A detailed overview of the requirements is described in Section 7.1. Our dataset is the only one meeting all requirements.*

such as *occlusions*, *fast-motion*, *non-linear deformation* and *motion-blur*. Correlated with the performance of the tested approaches, these attributes enable a deeper understanding of the results and point towards promising avenues for future research. The components described above represent a complete benchmark suite, providing researchers with the necessary tools to facilitate the evaluation of their methods and advance the field of video object segmentation.

7.1 Dataset Description

In this section we describe our new dataset DAVIS (Densely Annotated VIdeo Segmentation) specifically geared towards the task of video object segmentation. Example frames of some of the sequences are shown in Figure 7.1, refer to Figure 7.6 for the complete dataset. Based on experiences with existing datasets we first identify four key aspects we adhere to, in order create a balanced and comprehensive dataset. A summary of the requirements detailed below can be found in Table 7.1.

Data Amount and Quality. A sufficiently large amount of data is necessary to ensure content diversity and to provide a uniformly distributed set of challenges. Furthermore, having enough data is crucial to avoid overfitting and to delay performance saturation, hence guaranteeing a longer lifespan of the dataset [Butler et al., 2012]. The quality of the data also plays a crucial role, as it should be representative of the current state of technology. To this end, DAVIS comprises a total of 50 sequences, 3455 annotated frames, all captured at 24*fps* and Full HD 1080p spatial resolution. Due to

ID	Description
BC	*Background Clutter*. The back- and foreground regions around the object boundaries have similar colors (χ^2 over histograms).
DEF	*Deformation*. Object undergoes complex, non-rigid deformations.
MB	*Motion Blur*. Object has fuzzy boundaries due to fast motion.
FM	*Fast-Motion*. The average, per-frame object motion, computed as centroids Euclidean distance, is larger than $\tau_{fm} = 20$ pixels.
LR	*Low Resolution*. The ratio between the average object bounding-box area and the image area is smaller than $t_{lr} = 0.1$.
OCC	*Occlusion*. Object becomes partially or fully occluded.
OV	*Out-of-view*. Object is partially clipped by the image boundaries.
SV	*Scale-Variation*. The area ratio among any pair of bounding-boxes enclosing the target object is smaller than $\tau_{sv} = 0.5$.
AC	*Appearance Change*. Noticeable appearance variation, due to illumination changes and relative camera-object rotation.
EA	*Edge Ambiguity*. Unreliable edge detection. The average ground-truth edge probability (using [Dollár and Zitnick, 2013]) is smaller than $\tau_e = 0.5$.
CS	*Camera-Shake*. Footage displays non-negligible vibrations.
HO	*Heterogeneus Object*. Object regions have distinct colors.
IO	*Interacting Objects*. The target object is an ensemble of multiple, spatially-connected objects (*e.g.*mother with stroller).
DB	*Dynamic Background*. Background regions move or deform.
SC	*Shape Complexity*. The object has complex boundaries such as thin parts and holes.

Table 7.2: *List of video attributes and corresponding description. We extend the annotations of We et al. [2013] (top) with a complementary set of attributes relevant to video object segmentation (bottom). We refer the reader to Table 7.6 for the list of attributes for each in video in the dataset.*

the computational complexity being a major bottleneck in video processing, the sequences have a short temporal extent (about 2-4 seconds), but include all major challenges typically found in longer video sequences, see Table 7.2.

Experimental Validation. For each video frame, we provide pixel-accurate, manually created segmentation in the form of a binary mask. While we subdivide DAVIS into training- and a test-set to provide guidelines for future works, in our evaluation, we do not make use of the partition, and instead consider the dataset as a whole, since most of the evaluated approaches are not trained and a grid-search estimation of the optimal parameters would be infeasible due to the involved computational complexity.

Object Presence. Intuitively each sequence should contain at least one target foreground-object to be separated from the background regions. The clips in DAVIS contain either one single object or two spatially connected objects. We choose not to have multiple distinct objects with significant motion in order to be able to fairly compare segmentation approaches operating on individual objects against those that jointly segment multiple objects. Moreover, having a single object per sequence disambiguates the detection performed by methods which are fully automatic. A similar design choice made in [Liu et al., 2011] has been successfully steering research in salient object detection from its beginnings to the current state-of-the-art. To ensure sufficient content diversity, which is necessary to comprehensively assess the performance of different algorithms, the dataset spans four evenly distributed classes (*humans*, *animals*, *vehicles*, *objects*) and several actions.

Unconstrained Video Challenges. To enable a deeper analysis and understanding of the performance of an algorithm, it is fundamentally important to identify the key factors and circumstances which might have influenced it. Thus, inspired by Wu et al. [2013] we define an extensive set of video attributes representing specific situations, such as fast-motion, occlusion and cluttered background, that typically pose challenges to video segmentation algorithms. Attributes are summarized in Table 7.2. They are not exclusive, therefore a sequence can be annotated with multiple attributes. Their distribution over the dataset, *i.e.*number of occurrences, and their pairwise dependencies are shown in Figure 7.2. The annotations enable us to decouple the analysis of the performance into different groups with dominant characteristics (e.g. occlusion), yielding a better understanding of each methods' strengths and weaknesses.

7.2 Evaluated Algorithms

Besides the methods proposed in this thesis, we evaluate a total of twelve video segmentation algorithms, which we selected based on their demonstrated state-of-the-art performance and source code availability, and one technique commonly used for preprocessing. The source code was either publicly available or it was shared by the authors upon request. We now mantain a webpage (*davischallenge.org*) with up-to-date state-of-the-arts results of the new techniques published at top-tier computer vision conferences.

Within the unsupervised category we evaluate the performance of NLC [Faktor and Irani, 2014], FST [Papazoglou and Ferrari, 2013], SAL [Shen et al., 2015], TRC [Fragkiadaki et al., 2012], MSG [Brox and Malik, 2010] and

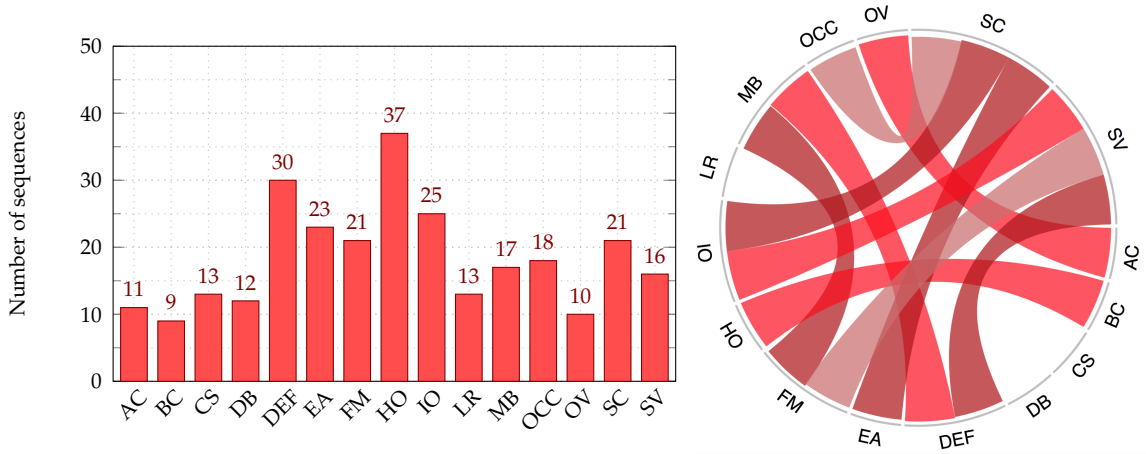


Figure 7.2: *Left: Attributes distribution over the dataset. Each bin indicates the number of occurrences. Right: Mutual dependencies among attributes. The presence of a link indicates high probability of an attribute to appear in a sequence, if the one on the other end is also present.*

CVOS [Taylor et al., 2015]. The three latter approaches generates multiple segments per-frame, and therefore, as suggested in [Brox and Malik, 2010], we solve the bipartite graph matching that maximizes region similarity in terms of $\mathcal{J}$ to select the most similar to the target object. Among the semi-automatic techniques, we compare the approaches proposed in this thesis FCP (§4), MSK (§5) against SEA [Ramakanth and Babu, 2014], JMP [Fan et al., 2015], TSP [Chang et al., 2013] and HVS [Grundmann et al., 2010]. HVS is meant for hierarchical over-segmentation, hence we search the hierarchy level and the corresponding segments that maximizes $\mathcal{J}$ of the first frame, keeping the annotation fixed throughout the entire video. FCP (§4) uses a pair of annotated object proposals to initialize the classifiers. In our evaluation KEY [Lee et al., 2011] is deemed to be semi-automatic since we override their abjectness score and instead use the ground-truth to select the optimal hypotheses which is then refined solving a series of spatio-temporal graph-cuts. The other methods are initialized using the first-frame, ground-truth segmentation.

The selected algorithms span the categories devised in Chapter 2 based on the level of supervision. However, interactive approaches with manual feedback could theoretically yield optimal results, and are not directly comparable with un- and semi-automatic approaches, since the number of user edits, *e.g.*strokes, should be also taken into account. Therefore we cast JMP [Fan et al., 2015] and BVS (§6) into semi-automatic methods that propagates masks to consecutive frames similar to SEA [Ramakanth and Babu, 2014]. We reduce the number of categories in Table 7.4 and Table 7.5 accordingly.

Additionally we evaluate the performance of the salient object detector proposed in Section 3.1 and the performance of an object proposal generator, as their output is a useful indicator with respect to the various video segmentation algorithms that are built upon them. We extract per-frame saliency from CIE-Lab images SF-LAB (§3.1) and from inter-frame motion SF-MOT (§3.1), while we use ground-truth to select the hypotheses of the object proposal generator MCG [Pont-Tuset et al., 2016] maximizing the per-frame Jaccard region similarity $\mathcal{J}$.

7.3 Experimental Validation

In order to judge the quality of a segmentation, the choice of a suitable metric is largely dependent on the end goal of the final application [Csurka et al., 2013]. Intuitively, when video segmentation is used primarily a classifier within a larger processing pipeline, *e.g.*for parsing large scale datasets, it makes sense to seek the lowest amount of mislabeled pixels. On the other hand, in video editing applications the accuracy of the contours and their temporal stability is of highest importance, as these properties usually require the most painstaking and time-consuming manual input. In order to exhaustively cover the aforementioned aspects we evaluate the video segmentation results using three complementary error metrics. We describe the metrics in Section 7.3.1 and we empirically validate their complementary properties on the proposed dataset in Section 7.3.2.

7.3.1 Metrics Selection

In a supervised evaluation framework, given a ground-truth mask G on a particular frame and an output segmentation M, any evaluation measure ultimately has to answer the question how well M *fits* G. As justified in [Pont-Tuset and Marques, 2015], for images one can use two complementary points of view, region-based and contour-based measures. As videos extends the dimensionality of still images to time, the temporal stability of the results must also be considered. Our evaluation is therefore based on the following measures.

Region Similarity $\mathcal{J}$. To measure the region-based segmentation similarity, *i.e.*the number of mislabeled pixels, we employ the Jaccard index $\mathcal{J}$ defined as the *intersection-over-union* of the estimated segmentation and the ground-truth mask. The Jaccard index has been widely adopted since its first appearance in PASCAL VOC2008 [Everingham et al., 2010], as it provides intuitive, scale-invariant information on the number of mislabeled pixels. Given an

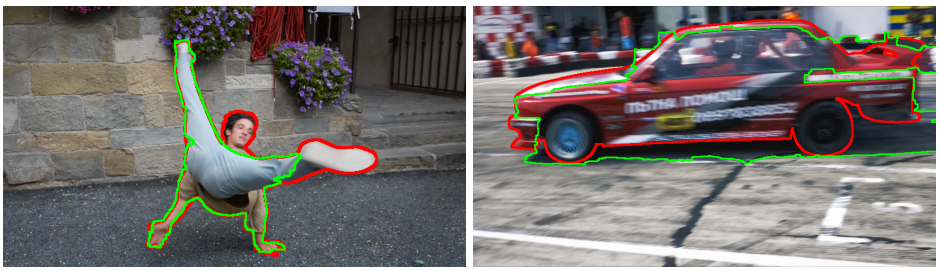


Figure 7.3: *Discrepancy between metrics. Ground truth in red and an example segmentation result in green. On the left, the result is penalized by $\mathcal{J}$ because in terms of number of pixels there is a significant amount of false negatives (head and foot), while with respect to the boundary measure $\mathcal{F}$ the missed percentage is lower. On the right the response of both measures is switched. The discrepancy in terms of pixels is low because the erroneous area is small, but the boundaries are highly inaccurate.*

output segmentation M and the corresponding ground-truth mask G it is defined as $\mathcal{J} = \frac{|M \cap G|}{|M \cup G|}$.

Contour Accuracy $\mathcal{F}$. From a contour-based perspective, one can interpret M as a set of closed contours $c(M)$ delimiting the spatial extent of the mask. Therefore, one can compute the contour-based precision and recall P_c and R_c between the contour points of $c(M)$ and $c(G)$, via a bipartite graph matching in order to be robust to small inaccuracies, as proposed in [Martin et al., 2004]. We consider the so called F-measure $\mathcal{F}$ as a good trade-off between the two, defined as $\mathcal{F} = \frac{2P_c R_c}{P_c + R_c}$. For efficiency, in our experiments, we approximate the bipartite matching via morphology operators.

Temporal stability $\mathcal{T}$. Intuitively, $\mathcal{J}$ measures how well the pixels of the two masks match, while $\mathcal{F}$ measures the accuracy of the contours. However, temporal stability of the results is a relevant aspect in video object segmentation since the evolution of object shapes is an important cue for recognition and jittery, unstable boundaries are unacceptable in video editing applications. Therefore, we additionally introduce a temporal stability measure which penalizes such undesired effects.

The key challenge is to distinguish the *acceptable* motion of the objects from the undesired instability and jitter. To do so, we estimate the deformation needed to transform the mask at one frame to the next one. Intuitively, if the transformation is smooth and precise, the result can be considered stable.

Formally, we transform mask M_t of frame t into polygons representing its contours $P(M_t)$. We then describe each point $p_t^i \in P(M_t)$ using the Shape Context Descriptor (SCD) [Belongie et al., 2002b]. Next, we pose the match-

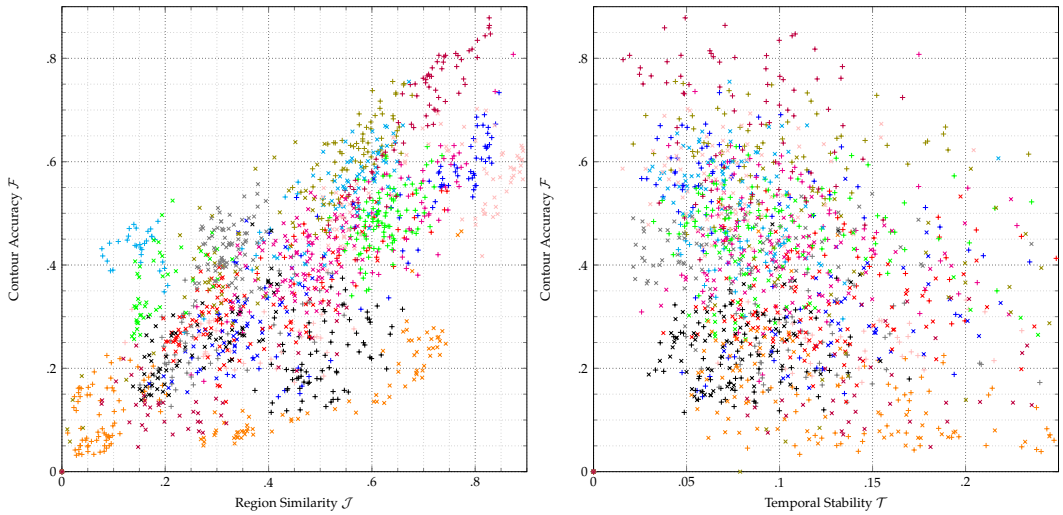


Figure 7.4: *Correlation between the proposed metrics. Markers correspond to video frames. Colors encode membership to a specific video sequence. The contour accuracy measure $\mathcal{F}$ exhibits a slight linear dependency with respect to the region similarity $\mathcal{J}$ (left), while it appears uncorrelated to the temporal stability $\mathcal{T}$ (right).*

ing as a Dynamic Time Warping (DTW) [Rabiner and Juang, 1993] problem, were we look for the matching between p_t^i and p_{t+1}^j that minimizes the SCD distances between the matched points while preserving the order in which the points are present in the shapes.

The resulting mean cost per matched point is used as the measure of temporal stability $\mathcal{T}$. Intuitively, the matching will compensate motion and small deformations, but it will not compensate the oscillations and inaccuracies of the contours, which is what we want to measure. Occlusions and very strong deformations would be misinterpreted as contour instability, so we compute the measure on a subset of sequences without such effects.

7.3.2 Metrics Validation

To verify that the use of these measures produces meaningful results on our dataset, we compute the pairwise correlation between the region similarity $\mathcal{J}$ and the contour accuracy $\mathcal{F}$ and between $\mathcal{F}$ and the temporal stability measure $\mathcal{T}$. The degree of correlation is visualized in Figure 7.4. As can be expected, there is a tendency towards linear correlation between $\mathcal{J}$ and $\mathcal{F}$ (Figure 7.4, *left*), which can be explained by the observation that higher quality segmentations usually also result in more accurate contours. We note, however, that the level of independence is enough to justify the use of both

	Preprocessing			Unsupervised							Semi-Supervised						
	MCG	SF-LAB	SF-MOT	NLC	CVOS	TRC	MSG	KEY	SAL	FST	TSP	SEA	HVS	JMP	FCP	BVS	MSK
Time	1498s	85s	84s	880s	6190s	16612	2614s	55191s	149s	2319s	2870s	299s	634s	576s	2261s	29s	960s

Table 7.3: *Running times. Estimated running times (in seconds) for each of the evaluated approaches on a video sequence of 80 frames. Due to the substantial processing power required to carry out this large scale evaluation, we used multiple machines and a cluster with thousands nodes and different CPU, therefore while computing times have been normalized to comparable processing power, they should be considered an aproximate estimate.*

measures. To get a qualitative idea of the differences between the two measures, Figure 7.3 shows two results of discrepant judgments between $\mathcal{J}$ and $\mathcal{F}$. The temporal stability measure $\mathcal{T}$ and the contour accuracy $\mathcal{F}$ instead are nearly uncorrelated (Figure 7.4, *right*), which is also expected since temporal instability does not necessarily impact the per-frame performance.

7.4 Quantitative Evaluation

In this section we report the results of the fifteen evaluated approaches. We first provide different statistics evaluated for each of the three error measures (regions, contours, temporal), and then discuss evaluation results at the attribute level (*e.g.*, performance with respect to appearance changes).

For each of the methods we kept the default parameters fixed throughout the entire dataset. Despite a considerable effort to speed-up the computation (parallelizing preprocessing steps such as motion estimation or extraction of boundary preserving regions) and to reduce the memory footprint (caching intermediate steps), several methods based on global optimization routines cannot be easily accelerated. Therefore, in order to be able to evaluate all methods with respect to each other, we were forced to down-sample the videos to 480p resolution. Due to the enormous processing power required, we performed experiments on different machines and partly on a cluster with thousands of nodes and heterogeneous CPU cores. Indicative runtimes are reported in Table 7.3.

The evaluation scripts, the input data, and the output results are made publicly available.

We exclude from the evaluation the first frame, which is used as ground-truth by semi-automatic methods, and the last frame which is not processed by some of the approaches. The overall results and considerations are re-

		Preprocessing			Unsupervised							Semi-Supervised						
	Measure	MCG	SF-LAB	SF-MOT	NLC	CVOS	TRC	MSG	KEY	SAL	FST	TSP	SEA	HVS	JMP	FCP	BVS	MSK
$\mathcal{J}$	Mean $\mathcal{M}\uparrow$	**72.4**	17.3	53.2	**64.1**	51.4	50.1	54.3	56.9	42.6	57.5	35.8	55.6	59.6	60.7	63.1	66.5	**80.3**
	Recall $\mathcal{O}\uparrow$	**91.2**	7.5	67.2	**73.1**	58.1	56.0	63.6	67.1	38.6	65.2	38.8	60.6	69.8	69.3	77.8	76.4	**93.5**
	Decay $\mathcal{D}\downarrow$	2.6	**-2.0**	5.0	8.6	12.7	5.0	**2.8**	7.5	8.4	4.4	38.5	35.5	19.7	37.2	**3.1**	26.0	8.9
$\mathcal{F}$	Mean $\mathcal{M}\uparrow$	**65.4**	21.8	45.2	**59.3**	49.0	47.8	52.5	50.3	38.3	53.6	34.6	53.3	57.6	58.6	54.6	65.6	**75.8**
	Recall $\mathcal{O}\uparrow$	**78.1**	5.2	44.0	**65.8**	57.8	51.9	61.3	53.4	26.4	57.9	32.9	55.9	**71.2**	65.6	60.4	77.4	**88.2**
	Decay $\mathcal{D}\downarrow$	4.6	**-1.6**	5.2	8.6	13.8	6.6	**5.7**	7.9	7.2	6.5	38.8	33.9	20.2	37.3	**3.9**	23.6	9.5
$\mathcal{T}$	Mean $\mathcal{M}\downarrow$	65.2	75.8	**63.7**	35.6	24.3	32.7	25.0	**19.0**	60.0	27.6	32.9	13.7	29.6	**13.1**	28.5	31.6	18.6

Table 7.4: *Overall results of region similarity ($\mathcal{J}$), contour accuracy ($\mathcal{F}$) and temporal (in-)stability ($\mathcal{T}$) for each of the tested algorithm. For rows with an upward pointing arrow higher numbers are better (e.g., mean), and vice versa for rows with downward pointing arrows (e.g., decay, instability).The per-sequence evaluation of each of the aforementioned approaches can be found in Tables 7.7, 7.8, 7.9.*

ported in Section 7.4.1 and summarized in Table 7.4, while the attributes-based evaluation is discussed in Section 7.4.2 and summarized in Table 7.5.

In Figure 7.5 we visualize the mean performance of all evaluated approaches, based the per-sequence region-similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$. The results are an estimator of the expected segmentation difficulty for a specific sequence. Sequences are sorted with respect to the estimated difficulty.

7.4.1 Error Measure Statistics

For a given error measure $\mathcal{C}$ we consider three different statistics. Let $R = \{S_i\}$ be the dataset of video sequences S_i and let $\bar{\mathcal{C}}(S_i)$ be the error measure average on S_i. The *mean* is the average dataset error defined as $\mathcal{M}_{\mathcal{C}}(R) = \frac{1}{|R|}\sum_{S\in R}\bar{\mathcal{C}}(S_i)$. The *decay* quantifies the performance loss (or gain) over time. Let $Q_i = \{Q_i^1, .., Q_i^4\}$ be a partition of S_i in quartiles, we define the *decay* as $\mathcal{D}_{\mathcal{C}}(R) = \frac{1}{|R|}\sum_{Q_i\in R}\bar{\mathcal{C}}(Q_i^1) - \bar{\mathcal{C}}(Q_i^4)$. The *object recall* measures the fraction of sequences scoring higher than a threshold, defined as $\mathcal{O}_{\mathcal{C}}(R) = \frac{1}{|R|}\sum_{S\in R}\mathbb{1}_{\bar{\mathcal{C}}(S_i)>\tau}$, with $\tau = 0.5$ in our experiments.

The region-based evaluation for all methods is summarized in Table 7.4. The best performing approach in terms of mean *intersection-over-union* is MSK (§5, $\mathcal{M}_{\mathcal{J}} = 80.3$). It outperforms by an ample margin of $\sim$10 points BVS (§6, $\mathcal{M}_{\mathcal{J}} = 66.5$), which is closely followed by NLC [Faktor and Irani, 2014] and FCP (§4, $\mathcal{M}_{\mathcal{J}} = 63.1$)

With the exception of FCP (§4), which solves a global optimization problem over a fully connected graph and MSK (§5) that uses the previous estimate as

a rough guidance, all the others semi-automatic approaches such as BVS (§6), propagate the initial manual segmentation iteratively to consecutive frames and thus exhibit higher temporal performance decay as reflected in the results. To alleviate this problem, propagating using bigger steps and interpolating the results in-between can reduce the drift and improve the overall results [Fan et al., 2015]. TRC [Fragkiadaki et al., 2012] and MSG [Brox and Malik, 2010] belong to a class of methods that uses motion segmentation as a prior, but the resulting over-segmentation of the object reflects negatively on the average performance. CVOS [Taylor et al., 2015] uses occlusion boundaries, but still encounters similar issues. Differently from TRC and MSG, CVOS performs online segmentation. It scales better to longer sequences in terms of efficiency but experiences higher decay.

Aiming at detecting per-frame indicators of potential foreground object locations, KEY [Lee et al., 2011], SAL [Shen et al., 2015], and FST [Papazoglou and Ferrari, 2013] try to determine prior information sparsely distributed over the video sequence. The prior is consolidated enforcing spatio-temporal coherence and stability by minimizing an energy function over a locally connected graph. While the local connectivity enables propagation of the segmentation similar to those of the semi-automatic approaches listed above, these methods suffer less decay as annotations are available at multiple different time frames.

Within the *preprocessing* category, the oracle MCG [Pont-Tuset et al., 2016] is an informative upper-bound for methods seeking the best possible proposal per-frame. It has the highest region-based performance $\mathcal{J}$ and superior object recall $\mathcal{M}_{\mathcal{J}}$. The performance of MCG, also supported by the good performance of FCP and KEY that use concurrent object proposal generators, indicates that this could be a promising direction for more future research. As expected, in video sequences motion is a stronger low-level cue for object presence than color. Consequently salient motion detection SF-MOT (§3.1) hows a significantly better performance than SF-LAB.

In terms of contour accuracy the best performing approaches are MSK (§5) and BVS (§6). Our deep-learning based approach MSK, exploits the expressiveness of convnet features which, coupled with a per-pixel CRF, consistently yields accurate contours. The good performance of BVS, demonstrates that the *adjacent* interpolation scheme we propose in Section 6.1.2 is a good compromise between speed and accuracy. The two aforementioned approaches are followed by NLC and JMP. The former uses a large number of superpixels per-frame ($\sim$2000) and a discriminative ensemble of features to represent them. In contrast, JMP exploits geodesic active contours to refine the object boundaries. The motion clusters of TRC and MSG, as well

	Unsupervised							Semi-Supervised						
Attr	NLC	CVOS	TRC	MSG	KEY	SAL	FST	TSP	SEA	HVS	JMP	FCP	BVS	MSK
AC	54 *+13*	42 *+12*	37 *+17*	48 *+8*	42 *+19*	33 *+12*	**55 *+4***	17 *+23*	46 *+12*	42 *+23*	58 *+3*	51 *+16*	46 *+26*	**77 *+4***
DB	**53 *+15***	37 *+18*	39 *+15*	43 *+15*	52 *+7*	35 *+10*	53 *+6*	40 *-6*	58 *-3*	60 *-1*	60 *+1*	62 *+1*	60 *+3*	**76 *+5***
FM	**64 *+0***	37 *+24*	41 *+16*	46 *+14*	50 *+12*	35 *+13*	50 *+12*	18 *+31*	40 *+28*	42 *+31*	50 *+18*	55 *+13*	54 *+22*	**76 *+8***
MB	**61 *+4***	36 *+23*	32 *+27*	35 *+29*	51 *+8*	33 *+15*	48 *+14*	15 *+32*	39 *+24*	44 *+24*	51 *+15*	53 *+15*	58 *+13*	**74 *+9***
OCC	**70 *-9***	43 *+13*	44 *+10*	48 *+10*	52 *+8*	44 *-2*	53 *+7*	27 *+14*	47 *+13*	53 *+11*	47 *+21*	59 *+7*	68 *-12*	**77 *+4***

Table 7.5: *Attribute-based aggregate performance. For each method, the respective left column corresponds to the average region similarity $\mathcal{J}$ over all sequences with that specific attribute (e.g., AC), while the right column indicates the performance gain (or loss) for that method for the remaining sequences without that respective attribute.*

as the occlusion boundaries of CVOS generate sub-optimal results along the boundaries.

The top ranked methods in terms of temporal stability are those that propagate segmentation on consecutive frames (JMP, SEA). Despite processing the video sequence on a per-frame basis, MSK is refined with a CRF over a temporal window of three frames that reduces the temporal instability, yielding overall good performance. Similarly the temporal stability of BVS could be improved at the cost of efficiency, by embedding more frames on the bilateral grid. As expected those that are used on a per-frame basis and cannot enforce continuity over time, such as MCG and SF-(*) generate considerably higher temporal instability. As a sanity check, we evaluate the temporal stability of the ground truth and we get $\mathcal{T} = 9.3$, which is lower than any of the sequences. The per-sequence evaluation of each of the aforementioned approaches can be found in Tables 7.7, 7.8, 7.9.

7.4.2 Attributes-based Evaluation

As discussed in Section 7.1 and Table 7.2 we annotated the video sequences with attributes each representing a different challenging factor. These attributes allow us to identify groups of videos with a dominant feature *e.g.*, presence of occlusions, which is key to explaining the algorithms' performance. However, since multiple attributes are assigned to each sequence (Table 7.6), there might exists hidden dependencies among them which could potentially affect an objective analysis of the results. Therefore, we first conduct a statistical analysis to establish these relationship, and then detail the corresponding evaluation results.

Attributes Dependencies. We consider the presence or absence of each attribute in a video sequence to be represented as a binary random variable, the dependencies between which can be modelled by a pairwise Markov random field (MRF) defined on a graph G with vertex set $V \in \{1, \ldots, 16\}$ and (unknown) edge set E. The absence of an edge between two attributes denotes that they are *independent* conditioned on the remaining attributes. Given a collection of $n = 50$ binary vectors denoting the presence of attributes in each video sequence, we estimate E via ℓ_1 penalized logistic regression. To ensure robustness in the estimated graph we employ *stability selection* [Meinshausen and Bühlmann, 2010]. Briefly, this amounts to performing the above procedure on $n/2$-sized subsamples of the data multiple times and computing the proportion of times each edge is selected. Setting an appropriate threshold on this selection probability allows us to control the number of wrongly estimated edges according to Theorem 1 in [Meinshausen and Bühlmann, 2010]. For example, for a threshold value of 0.6 and choosing a value of λ which on average selects neighbourhoods of size 4, the number of wrongly selected edges is at most 4 (out of $16^2 = 256$ possible edges). The estimated dependencies are visualized in Figure 7.2 (*right*). As expected there is a mutual dependency between attributes such as *fast-mostion* (FM) and *motion-blur* (MB), or *interacting-object* (IO) and *shape-complexity* (SC). We refer the reader to Section 7.5 for further details.

Results. In Table 7.5 we report the performance on subsets of the datasets characterized by a particular attribute. We reduce the analysis to the most informative and recurrent attributes. The full attribute-based evaluation is reported in Table 7.10.

Appearance changes (AC) poses a challenge to several approaches, in particular for those methods strongly relying on color appearance similarity such as HVS and TCP. For example, TSP performance drops almost 50% as a consequence of the Gaussian process it uses to update the appearance model and therefore not being robust enough to strong appearance variations. Despite the dense connectivity of its conditional random field, FCP also experiences a considerable loss of performance. The reason resides in a sub-optimal automatic choice of the annotated proposals. Likely the proposals did have enough variety to span the entire object appearances causing the classifiers to overfit. Similarly BVS is not robust to appearance changes. The performance loss is due to the efficient, but simple, bilateral features that are too sensitive to color changes. In contrast, MSK captures a robust, global representation of the object which is less subject to appearance changes. This is, in part thanks to the invariance of the convnet based features, but mostly due to the fine-tuning step and in particular to the data augmentation that mimics unseen instances of the object.

Dynamic background (DB) scenes, *e.g.*flowing water, represent a major difficulty to the class of unsupervised methods, such as NLC and SAL, which adopt distinctive motion saliency as the underlying assumption to predict the object location. Interestingly the assumption of a completely closed motion boundary curve coinciding with the object contours can robustly accommodate background deformations (FST). Finally, MSG and TRC experience a considerable performance degradation as the motion clusters they rely on [Brox and Malik, 2010] are constructed from dissimilarities of point-trajectories, under the assumption that translational models are a good approximation for nearby points, which is not true on deforming image regions. None of the methods we propose in this thesis explicitly employ intra-frame motion and therefore they are not subject to scenes exhibiting dynamic background, therefore the loss of all three approaches is only marginal.

Fast motion (FM) is a problem for any of the algorithms exploiting motion information as the condition is a major challenge to reliable optical-flow computation. Note that there is a strong dependency between fast motion and motion-blur (MB) (Figure 7.2, *right*), yielding fuzzy object boundaries almost impossible to separate from the background region. Methods such as TRC and MSG use point-tracks for increased robustness towards fast motion, but are still susceptible with respect to motion-blur due to the sensitivity of the underlying variational approach used for densification of the results. NLC is the only method which has none or negligible loss of performance in both circumstances, possibly because the saliency computation is still reliable on a subset of the frames, and their random-walk matrix being non-locally connected is robust to fast motion. Fast motion is also a major challenge for BVS which uses pixel coordinates to splat the data into the bilateral grid. The grid discretization plays a major role in this case. Under fast motion a fine grid causes corresponding pixels to end-up in distant grid cells. In the case of fast-motion a coarse grid along the pixel-coordinates dimensions might help. MSK also experience a loss due to fast motion, this is likely to the guiding segmentation from the previous frame, being less accurate due to the substantial change of the object position. Similarly for FCP the point-tracks are less reliable causing a degrade of accuracy.

Occlusions (OCC) being one of the well known challenges in video segmentation, only a small subset of the algorithms, which propagate sequentially manually annotated frames such as SEA and JMP, struggle with this type of situation. Despite processing the video sequence on a per-frame bases, MSK holds a specific representation of the object instance to segment and therefore is robust to occlusions. As expected, methods that exploit large spatio-temporal connectivity such as BVS,FCP and NLC are quite robust to these challenges.

7.5 Attributes Dependency

We consider the presence or absence of each attribute in a video sequence to be represented as a binary random variable $X = (X_1, X_2, \ldots, X_d)$. The dependencies between the attributes can be modelled by a pairwise Markov random field (MRF) defined on the undirected graph $G = (V, E)$ where $V = \{1, \ldots, d\}$ is the set of vertices and E is the (unknown) set of edges. Each variable X_s is associated with a vertex $s \in V$. The pairwise MRF associated with G is the family of distributions which factorise as

$$\mathbb{P}_\theta(x) \propto \exp\left\{ \sum_{(s,t)\in E} \theta_{s,t}\, x_s x_t \right\}.$$

The absence of an edge between s and t means that X_s and X_t are *independent* conditioned on their respective Markov blankets[1]. In other words, given the state of the neighbours of s and t, knowing t gives us no information about s and *vice-versa*.

Equivalently, θ can be viewed as a $\binom{d}{2}$-dimensional vector which indexes all distinct pairs of vertices but is non-zero only when the vertex pair (s, t) belongs to the edge set E of the graph. Recovering E is equivalent to recovering the neighbourhood set $\mathcal{N}(r) := \{t \in V | (r, t) \in E\}$ for each $r \in V$. Estimating the neighbourhood set $\mathcal{N}(r)$ is equivalent to estimating the support (i.e. location of non-zero entries) of the $(d-1)$ dimensional sub-vector $\theta_{\setminus r} := \{\theta_u, u \in V \setminus r\}$.

Following [Ravikumar et al., 2010], given a collection of n observations $\mathcal{X}^n = \{x^{(1)}, \ldots, x^{(n)}\}$ of d-dimensional binary vectors $x^{(i)}$, the support of each $\theta_{\setminus r}$ can be estimated by solving the following minimization problem

$$\min_{\theta_{\setminus r} \in \mathbb{R}^{d-1}} -\frac{1}{n} \sum_{i=1}^{n} \log \mathbb{P}_\theta(x_r^{(i)} | x_{\setminus r}^{(i)}) + \lambda \|\theta_{\setminus r}\|. \tag{7.1}$$

Since the random variables are binary, minimizing the penalised negative log likelihood above corresponds to solving ℓ_1 penalised logistic regression treating $x_{\setminus r}^{(i)} \in \mathbb{R}^{n \times (d-1)}$ as covariates and $x_r^{(i)}$ as the response.

The solution to (7.1) can be highly sensitive to the regularization strength λ which controls the sparsity of the solution. In order to determine the correct degree of sparsity we employ *stability selection* [Meinshausen and Bühlmann, 2010]. Briefly, this amounts to performing the above procedure on multiple

[1] For a MRF this consists of the neighbours of s and t, respectively

$n/2$-sized subsamples of the data and computing the proportion of times each edge is selected. Setting an appropriate threshold on this selection probability allows us to control the number of wrongly estimated edges according to Theorem 1 in [Meinshausen and Bühlmann, 2010]. For example, for a threshold value of 0.6 and choosing a value of λ which on average selects neighbourhoods of size 4, the number of wrongly selected edges is at most 4 (out of $16^2 = 256$ possible edges).

7.6 Discussion

To the best of our knowledge, this work represents the currently largest scale performance evaluation of video object segmentation algorithms. One of course has to consider that the evaluated approaches have been developed using different amounts and types of input data and ground-truth, or were partially even designed for different problems and only later adapted to the task of video object segmentation. However, the primary aim of our evaluation is not to determine a winner, but to provide researchers with high-quality, contemporary data, a solid standardized evaluation procedure, and valuable comparisons with the current state-of-the-art.

Currently, running time efficiency and memory requirements are a major bottleneck for the usability of several video segmentation algorithms. In our experiments we observed that a substantial amount of time is spent preprocessing images to extract boundary preserving regions, object proposals and motion estimates. We encourage future research to carefully select those components bearing in mind they could compromise the practical utility of their work. Efficient algorithms will be able to take advantage of the Full HD videos and accurate segmentation masks made available with this dataset. Leveraging high resolution might not produce better results in terms of region-similarity, but it is essential to improve the segmentation of complex object contours and tiny object region.

Workshop Supported by the growing interests of the video segmentation community towards the DAVIS dataset and benchmark we organized the first *DAVIS Challenge on Video Object Segmentation 2017*. The workshop, co-located with CVPR, has the objective to promote and facilitate the development of research techniques aiming to separate foreground objects from background regions in video sequences. Motivated by the substantial amount of feedback we collected, we (i) extended the original dataset with 100 extra sequences, totaling around 10K of pixel-accurate annotated frames; (ii) hosted a public challenge and competition to further engage researchers

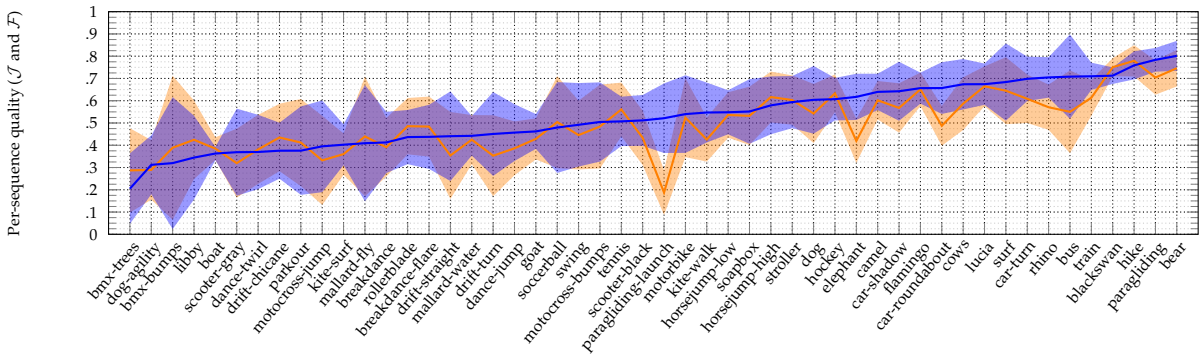


Figure 7.5: *Per-sequence mean performance. Mean and variance of region Jaccard $\mathcal{J}$ (blue) and boundary F measure $\mathcal{F}$ (orange). Sequences are sorted by difficulty, i.e. mean performance of $\mathcal{J}$ over all techniques.*

around the increasingly popular topic of video object segmentation. Several speakers were invited to present the current methods and future trends encouraging constructive discussion among participants.

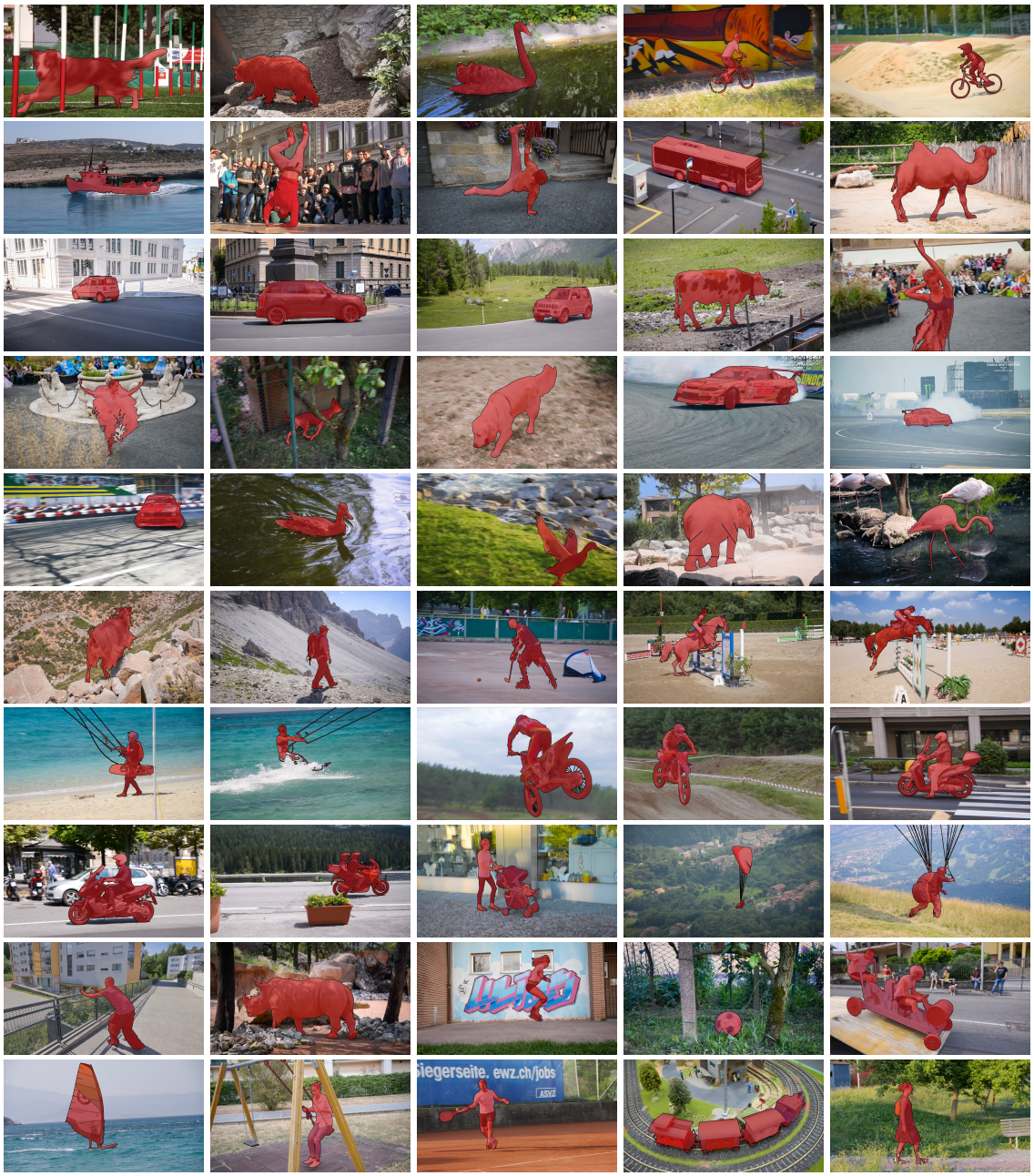


Figure 7.6: *Sample sequences from our dataset DAVIS, with ground truth segmentation masks overlayed.*

Sequence	AC	BC	CS	DB	DEF	EA	FM	HO	IO	LR	MB	OCC	OV	SC	SV
bear					✓										
blackswan															
bmx-bumps			✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bmx-trees		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
boat		✓		✓		✓								✓	
breakdance	✓			✓	✓		✓	✓			✓		✓		
breakdance-flare			✓		✓		✓	✓			✓				
bus						✓		✓				✓		✓	
camel			✓		✓				✓						
car-roundabout		✓													
car-shadow	✓	✓				✓				✓					
car-turn		✓													✓
cows			✓		✓			✓	✓			✓			
dance-jump				✓	✓	✓		✓			✓	✓		✓	
dance-twirl			✓		✓			✓	✓		✓		✓	✓	
dog			✓		✓	✓	✓				✓				
dog-agility	✓				✓	✓	✓	✓			✓	✓	✓		
drift-chicane	✓			✓		✓	✓	✓		✓					✓
drift-straight	✓		✓			✓	✓	✓		✓	✓		✓		✓
drift-turn	✓			✓			✓	✓	✓				✓		✓
elephant			✓	✓	✓	✓									
flamingo				✓	✓			✓	✓					✓	
goat		✓	✓		✓	✓									
hike					✓			✓		✓					
hockey					✓			✓	✓					✓	
horsejump-high					✓			✓	✓			✓		✓	
horsejump-low					✓	✓		✓	✓			✓		✓	
kite-surf				✓		✓		✓	✓		✓	✓		✓	✓
kite-walk				✓	✓			✓	✓			✓		✓	
libby					✓	✓		✓			✓	✓		✓	
lucia					✓			✓				✓			
mallard-fly	✓			✓	✓	✓	✓			✓	✓		✓		✓
mallard-water				✓		✓			✓	✓					
motocross-bumps	✓	✓					✓	✓	✓				✓		✓
motocross-jump	✓				✓	✓	✓	✓	✓		✓		✓	✓	✓
motorbike						✓	✓	✓	✓	✓		✓		✓	✓
paragliding								✓	✓	✓				✓	
paragliding-launch					✓	✓		✓	✓					✓	
parkour	✓				✓		✓	✓		✓		✓			✓
rhino		✓			✓							✓			
rollerblade			✓		✓		✓	✓		✓	✓				
scooter-black						✓		✓	✓						✓
scooter-gray		✓				✓	✓	✓	✓			✓		✓	
soapbox	✓				✓			✓	✓		✓				✓
soccerball							✓	✓		✓	✓	✓			
stroller			✓		✓		✓	✓	✓					✓	
surf			✓	✓			✓	✓	✓				✓		✓
swing					✓		✓	✓	✓			✓		✓	
tennis					✓		✓	✓	✓		✓				✓
train						✓		✓						✓	

Table 7.6: *List of attributes for each video in the dataset. Left to right: appearance changes (AC), background clutter (BC), camera shake (CS), dynamic background (DB), non-linear deformation (DEF), edge ambiguity (EA), fast-motion (FM), heterogeneus object (HO), interacting objects (IO), low resolution (LR), motion blur (MB), occlusions (OCC), out-of-view (OV), shape complexity (SC), scale variation (SV). See Table 1 in the paper for the description of each attribute.*

	Preprocessing			Unsupervised							Semi-Supervised						
Sequence	MCG	SF-LAB	SF-MOT	NLC	CVOS	TRC	MSG	KEY	SAL	FST	TSP	SEA	HVS	JMP	FCP	BVS	MSK
bear	93.7	12.6	55.6	90.6	86.4	87.3	85.1	89.1	65.7	89.8	77.8	91.2	93.8	92.9	90.6	95.5	93.1
blackswan	87.1	52.4	54.7	87.4	42.2	56.9	52.6	84.2	22.2	73.2	87.2	93.3	91.7	93.0	90.8	94.3	90.3
bmx-bumps	49.0	3.0	28.1	63.5	36.8	35.0	35.3	30.9	18.8	24.1	29.0	19.8	42.8	33.6	30.0	43.4	57.1
bmx-trees	47.3	2.1	46.8	21.2	12.1	16.2	18.8	19.3	19.4	18.0	9.5	11.3	17.8	22.9	24.8	38.2	57.5
boat	61.9	6.8	17.1	0.7	5.6	13.0	14.4	06.5	27.1	36.1	65.6	79.3	78.2	70.5	61.3	64.4	54.8
breakdance	71.3	20.4	64.9	67.3	18.3	11.4	23.7	54.9	42.2	46.7	5.6	32.9	55.0	47.8	56.7	50.0	76.2
breakdance-flare	73.3	20.6	62.9	80.4	31.7	24.5	15.7	55.9	47.6	61.6	4.0	13.1	49.9	43.0	72.3	72.7	77.6
bus	74.9	29.0	79.7	62.9	66.4	68.4	88.5	78.5	73.9	82.5	51.5	75.2	80.9	66.8	83.2	86.3	89.1
camel	79.5	1.5	62.0	76.8	85.0	77.8	75.6	57.9	32.0	56.2	65.4	64.9	87.6	64.0	73.4	66.9	80.1
car-roundabout	78.6	30.6	70.8	50.9	87.1	55.2	63.0	64.0	50.0	80.8	61.4	70.8	77.7	72.6	71.7	85.1	96.0
car-shadow	70.1	13.5	76.5	64.5	75.9	44.9	88.0	58.9	53.8	69.8	63.6	77.5	69.9	64.5	72.3	57.8	93.5
car-turn	86.5	7.1	59.3	83.3	82.0	80.5	62.1	80.6	61.1	85.1	32.3	90.9	81.0	83.4	72.4	84.4	88.6
cows	81.1	13.9	68.4	88.3	56.2	83.3	79.9	33.7	62.3	79.1	59.5	70.7	77.9	75.6	81.2	89.5	88.2
dance-jump	47.0	25.1	58.2	71.8	34.1	30.3	06.5	74.8	29.1	59.8	13.2	66.2	68.0	49.0	52.2	74.5	78.8
dance-twirl	64.4	9.3	62.4	34.7	45.2	36.6	36.6	38.0	37.2	45.3	9.9	11.7	31.8	44.4	47.1	49.2	84.4
dog	62.1	19.5	53.2	80.9	75.3	78.6	33.1	69.2	56.6	70.8	31.3	58.1	72.2	67.3	77.4	72.3	90.9
dog-agility	66.3	6.0	35.4	65.2	19.3	13.8	11.0	13.2	5.5	28.0	07.9	35.4	45.7	69.9	45.3	34.5	78.9
drift-chicane	80.6	4.6	39.6	32.4	31.3	72.2	75.8	18.8	24.4	66.7	01.8	11.9	33.1	24.3	45.7	3.3	86.2
drift-straight	75.3	17.1	42.7	47.3	34.4	43.1	57.5	19.4	26.8	68.3	19.7	51.3	29.5	61.8	66.8	40.2	56.0
drift-turn	85.6	16.1	35.9	15.4	61.5	41.2	63.8	25.5	34.9	53.3	16.2	66.7	27.6	71.7	60.6	29.9	85.9
elephant	68.6	9.9	64.0	51.8	49.4	76.0	68.9	67.5	51.0	82.4	66.6	55.3	74.2	75.0	65.5	84.9	87.2
flamingo	85.0	24.7	51.7	53.9	78.3	73.1	79.4	69.2	57.0	81.7	66.6	58.3	81.1	53.0	71.7	88.1	79.0
goat	64.1	5.7	13.8	01.0	7.4	79.3	73.6	70.5	25.7	55.4	44.4	53.5	58.0	73.1	67.7	66.1	84.5
hike	90.0	33.5	65.7	91.8	87.8	75.6	60.3	89.5	68.3	88.9	67.9	77.6	87.7	66.4	87.4	75.5	93.1
hockey	77.5	8.0	53.8	81.0	81.7	67.4	71.3	51.5	56.6	46.8	41.3	71.4	69.8	67.7	64.7	82.9	83.4
horsejump-high	64.9	26.3	59.6	83.4	83.0	36.4	73.4	37.0	56.8	57.8	23.6	63.7	76.5	58.6	67.6	80.1	81.7
horsejump-low	54.5	12.5	61.8	65.1	74.3	70.5	68.2	63.0	38.8	52.6	29.1	49.8	55.1	66.3	60.7	60.1	80.6
kite-surf	65.4	5.9	20.8	45.3	35.7	50.1	41.9	58.5	19.3	27.2	36.6	48.7	40.5	50.0	57.7	42.5	60.0
kite-walk	73.6	66.8	42.0	81.3	44.7	5.2	59.7	19.7	72.5	64.9	44.7	49.8	76.5	50.9	68.2	87.0	64.5
libby	65.5	9.7	44.3	63.5	16.9	7.3	5.0	61.1	47.0	50.7	7.0	22.6	55.3	29.5	31.6	77.6	77.5
lucia	82.0	11.9	76.0	87.6	84.0	66.9	41.7	84.7	70.6	64.4	37.7	62.6	77.6	83.6	80.1	90.1	91.1
mallard-fly	79.9	2.2	31.0	61.7	38.0	29.3	03.3	58.5	22.7	60.1	20.0	55.7	43.6	53.6	54.1	60.6	57.3
mallard-water	75.5	3.5	1.7	76.1	24.5	19.0	4.5	78.5	8.5	8.7	62.3	86.5	70.4	75.1	68.7	90.7	90.4
motocross-bumps	82.7	23.6	46.0	61.4	60.3	50.2	46.6	68.9	35.1	61.7	13.3	47.0	53.4	76.1	30.6	40.1	59.9
motocross-jump	76.0	20.4	42.8	25.1	24.5	33.8	61.8	28.8	49.1	60.2	12.3	38.6	9.9	58.3	51.1	34.1	68.5
motorbike	68.8	11.6	57.2	71.4	38.7	72.3	73.7	57.2	33.5	55.8	34.0	45.1	68.7	50.6	71.3	56.3	56.6
paragliding	87.7	14.0	74.3	88.0	89.0	81.6	93.3	86.1	56.8	72.5	73.5	86.3	90.7	95.1	86.6	87.5	95.9
paragliding-launch	59.9	25.5	50.1	62.8	59.1	55.5	51.3	55.9	53.9	50.6	30.1	57.7	53.7	58.9	57.1	64.0	62.1
parkour	81.5	28.8	49.1	90.1	14.6	34.5	29.5	41.0	39.2	45.8	7.0	12.1	24.0	34.2	32.2	75.6	88.2
rhino	86.4	25.3	61.5	68.2	52.0	84.6	90.2	67.5	68.5	77.6	69.4	73.6	81.2	71.6	79.4	78.2	91.1
rollerblade	55.4	0.4	56.4	81.4	40.6	56.6	80.1	51.0	14.1	31.8	9.8	13.8	46.1	72.6	44.9	58.8	78.7
scooter-black	70.4	12.5	61.3	16.2	75.9	43.5	57.9	50.2	34.8	52.2	37.8	79.3	62.4	62.6	50.4	33.7	82.5
scooter-gray	65.3	3.9	70.3	58.6	32.7	35.7	34.5	36.3	42.1	32.5	13.3	24.1	43.3	12.3	48.3	50.8	82.9
soapbox	68.0	16.3	57.9	63.4	83.2	29.4	67.2	75.7	33.2	41.0	24.7	78.3	68.4	75.9	44.9	78.9	89.9
soccerball	85.6	5.2	73.2	82.9	24.2	35.0	37.0	87.8	37.8	84.3	02.9	65.3	6.5	9.6	82.0	84.4	89.0
stroller	60.0	31.3	57.3	85.0	61.9	72.0	67.8	75.9	46.6	58.0	36.9	46.4	66.2	65.6	59.7	76.7	85.4
surf	94.4	52.1	65.3	77.5	27.3	46.4	77.0	89.3	31.2	47.5	81.4	82.1	75.9	94.1	84.3	49.2	92.8
swing	70.9	38.7	67.7	85.1	53.3	41.3	62.2	71.0	56.9	43.1	09.8	51.1	10.4	11.5	64.8	78.4	81.9
tennis	71.4	3.2	56.2	87.1	49.4	19.6	59.0	76.2	48.0	38.8	7.4	48.2	57.6	76.5	62.3	73.7	86.1
train	53.5	20.5	54.8	72.9	90.3	87.6	88.7	45.0	62.0	83.1	64.8	85.4	84.6	87.3	84.1	87.2	90.4
Mean	72.4	17.3	53.2	64.1	51.4	50.1	54.3	56.9	42.6	57.5	35.8	55.6	59.6	60.7	63.1	66.5	80.3

Table 7.7: *Results of region similarity ($\mathcal{J}$) for each video sequence in the dataset. The best performing method of each category is highlighted in bold.*

	Preprocessing			Unsupervised							Semi-Supervised						
Sequence	MCG	SF-LAB	SF-MOT	NLC	CVOS	TRC	MSG	KEY	SAL	FST	TSP	SEA	HVS	JMP	FCP	BVS	MSK
bear	93.4	18.0	45.1	85.0	84.5	83.2	78.1	77.5	49.5	86.0	63.5	89.9	90.5	90.4	84.5	94.5	90.6
blackswan	87.3	49.1	50.0	82.0	69.5	65.4	70.0	78.7	43.0	73.6	85.7	95.7	91.0	94.5	90.5	96.5	89.4
bmx-bumps	59.7	10.6	28.3	73.4	40.9	32.5	41.0	45.3	31.3	34.9	33.8	25.4	52.5	39.7	34.0	49.3	67.8
bmx-trees	60.5	14.5	55.7	33.0	11.8	18.9	26.3	36.6	20.6	34.8	13.8	12.5	28.2	30.9	32.4	65.2	73.6
boat	53.6	28.9	13.5	3.6	10.8	40.3	48.5	0.0	26.4	19.7	68.2	76.4	80.7	60.7	46.0	64.8	50.3
breakdance	67.0	26.5	63.5	66.1	19.1	12.1	23.1	46.3	30.0	41.1	7.0	38.9	47.3	51.1	47.3	48.8	72.5
breakdance-flare	78.1	21.5	62.6	80.8	33.5	30.1	23.0	58.5	51.2	69.4	11.6	16.7	62.5	52.3	73.8	77.5	78.4
bus	49.7	29.6	50.5	40.6	53.5	54.2	65.7	63.5	57.0	58.4	47.7	72.4	68.2	60.4	53.9	84.4	65.3
camel	72.8	8.5	49.8	71.9	87.3	69.8	62.9	43.7	43.2	59.0	52.9	61.4	87.1	71.1	61.7	70.5	73.5
car-roundabout	51.2	24.6	49.1	25.0	67.8	45.1	60.2	36.2	30.1	62.5	43.5	71.0	55.1	61.9	47.8	62.4	92.6
car-shadow	58.8	23.3	62.8	54.6	61.7	47.4	85.8	45.9	44.1	54.0	51.3	75.5	59.4	62.5	64.2	47.4	94.7
car-turn	76.0	11.3	43.1	63.4	70.3	74.1	67.7	63.2	48.5	73.1	37.9	88.3	60.5	74.2	61.4	68.9	78.2
cows	73.6	16.8	55.4	80.7	49.9	72.1	62.1	29.3	49.9	68.1	54.4	67.7	63.2	70.0	66.7	85.1	81.2
dance-jump	37.2	20.4	49.7	56.7	28.2	27.2	3.8	56.9	26.2	46.2	18.6	56.7	57.1	52.6	41.8	64.5	62.9
dance-twirl	60.4	12.1	58.9	36.5	44.4	37.6	32.5	31.7	30.1	47.1	12.8	21.3	51.6	52.0	42.7	48.1	80.9
dog	58.2	14.0	49.5	70.7	76.1	69.5	30.4	63.3	41.8	65.9	29.5	54.3	63.5	59.6	67.2	59.4	88.5
dog-agility	51.7	15.3	28.8	55.1	26.2	12.2	7.6	9.5	10.2	26.5	8.3	41.0	44.6	65.4	31.5	34.6	68.4
drift-chicane	90.6	15.4	36.4	31.2	39.7	82.3	88.6	19.2	20.6	73.1	3.3	15.9	54.7	33.8	47.7	07.6	96.4
drift-straight	59.3	15.2	31.2	38.5	33.0	40.8	50.9	5.3	16.7	47.0	21.3	50.0	26.6	47.3	47.9	41.9	55.0
drift-turn	70.0	20.7	19.9	18.5	48.0	31.0	45.9	1.8	23.1	44.2	21.7	51.2	21.6	63.1	48.8	37.1	80.9
elephant	50.6	8.0	44.6	25.1	35.9	54.6	50.5	32.4	23.1	56.9	52.3	39.9	57.9	54.2	43.0	63.2	64.3
flamingo	87.6	25.3	62.2	61.0	80.6	66.3	77.6	58.9	62.1	76.3	54.4	56.3	79.0	65.0	64.1	93.3	72.4
goat	52.6	17.1	22.5	13.3	24.1	72.4	65.7	55.2	18.7	40.0	40.4	47.0	54.6	61.7	57.6	58.4	81.4
hike	92.6	31.2	54.0	94.3	92.2	80.4	70.2	92.5	69.1	91.8	67.5	79.6	87.8	74.4	91.2	76.4	96.0
hockey	74.2	19.8	43.0	80.8	78.9	65.1	76.1	56.0	55.9	58.4	57.9	72.1	77.8	72.6	61.2	85.0	79.1
horsejump-high	70.3	32.6	56.1	88.1	84.1	40.5	74.8	39.2	61.3	62.1	34.3	65.5	80.7	65.3	69.9	80.4	85.1
horsejump-low	54.8	15.2	51.6	65.9	70.9	67.2	63.7	53.3	41.9	49.0	35.6	54.8	57.2	69.6	53.3	56.5	81.2
kite-surf	44.7	22.1	28.6	44.8	24.1	42.2	52.1	50.4	36.8	34.6	26.8	28.5	37.5	30.9	36.2	64.5	43.8
kite-walk	48.6	52.4	28.5	66.2	43.8	1.4	57.7	12.8	52.6	56.1	43.5	35.5	62.4	35.9	41.1	72.8	44.1
libby	73.3	24.4	58.1	74.8	18.5	8.6	11.8	73.0	52.9	71.8	9.1	20.9	64.1	36.5	38.9	83.9	85.6
lucia	74.3	23.3	73.6	87.2	80.1	66.3	49.1	81.9	69.1	56.8	45.3	54.2	78.2	87.0	70.8	90.0	89.5
mallard-fly	82.4	7.1	33.8	66.0	39.1	33.2	1.9	63.1	29.3	63.3	23.5	60.7	44.1	57.9	53.9	64.5	60.1
mallard-water	70.1	11.5	3.4	69.2	25.4	22.5	0.0	73.3	11.5	7.9	58.5	88.6	64.6	75.5	55.7	91.4	93.9
motocross-bumps	71.0	24.2	33.8	56.0	56.7	49.7	46.6	67.4	30.0	61.0	18.4	52.0	54.8	74.3	30.2	49.0	55.4
motocross-jump	56.8	27.4	29.0	30.3	18.6	30.7	39.3	23.7	38.8	45.3	11.6	40.4	13.7	53.9	38.6	37.6	52.7
motorbike	64.2	24.7	26.4	57.1	38.0	54.1	59.4	72.6	39.1	58.5	40.6	48.1	82.3	57.8	63.2	69.6	59.7
paragliding	74.2	13.3	79.8	74.4	74.4	72.4	90.9	68.1	54.1	67.5	63.4	74.4	85.7	90.7	72.7	77.3	93.3
paragliding-launch	20.0	18.8	17.1	24.3	18.2	15.7	19.6	25.3	16.9	18.5	12.2	18.0	20.6	17.6	18.3	32.4	22.9
parkour	81.1	30.4	55.3	91.6	15.8	42.1	40.1	37.4	35.9	47.8	9.4	27.8	32.3	41.8	29.2	67.8	87.4
rhino	76.7	19.3	47.2	43.1	46.9	73.9	82.6	42.9	48.7	63.4	49.9	65.8	65.8	65.3	64.7	59.0	82.6
rollerblade	57.7	11.4	58.6	86.8	47.5	68.7	82.2	35.1	21.1	41.1	14.3	15.5	55.2	75.9	57.6	64.5	85.0
scooter-black	54.8	17.7	45.1	22.8	55.7	30.4	56.5	42.0	25.7	39.5	41.1	72.2	57.4	52.9	36.3	40.7	66.1
scooter-gray	49.4	12.6	43.4	46.7	21.2	26.6	27.2	36.7	33.3	32.1	12.2	27.5	54.5	12.3	43.7	60.2	65.9
soapbox	64.7	19.0	45.3	65.8	75.4	38.9	63.3	71.9	30.7	35.5	33.6	75.0	69.0	67.7	42.3	76.2	84.2
soccerball	89.8	4.3	75.4	85.5	26.2	37.7	40.1	92.4	35.5	90.0	5.9	69.7	7.4	12.7	83.6	84.9	92.8
stroller	63.5	42.6	48.8	87.4	60.6	69.1	66.2	75.1	41.7	55.8	40.4	52.5	70.8	71.8	58.1	79.0	85.5
surf	88.1	52.4	37.2	67.3	51.5	63.7	80.4	82.0	39.5	44.5	64.1	73.2	65.2	87.2	71.3	53.1	91.1
swing	59.9	37.3	57.1	77.8	49.3	41.7	61.1	61.4	50.2	49.1	8.7	40.9	9.1	10.9	53.8	74.6	74.5
tennis	75.1	15.1	45.0	92.7	54.7	30.1	67.0	81.8	53.0	56.7	11.4	53.7	57.9	81.8	65.2	84.5	91.1
train	40.3	34.7	50.4	52.1	83.1	76.6	77.0	46.4	44.0	66.0	58.9	71.3	68.8	77.0	73.6	79.2	83.5
Mean	65.4	21.8	45.2	59.3	49.0	47.8	52.5	50.3	38.3	53.6	34.6	53.3	57.6	58.6	54.6	65.6	75.8

Table 7.8: *Results of boundary precision ($\mathcal{F}$) for each video sequence in the dataset. The best performing method of each category is highlighted in bold.*

	Preprocessing			Unsupervised							Semi-Supervised						
Sequence	MCG	SF-LAB	SF-MOT	NLC	CVOS	TRC	MSG	KEY	SAL	FST	TSP	SEA	HVS	JMP	FCP	BVS	MSK
bear	0.2	63.3	84.0	15.1	5.9	27.2	15.6	6.8	44.8	22.7	7.7	4.7	8.6	5.1	11.4	7.6	8.3
blackswan	30.9	33.2	32.2	11.0	5.8	21.9	14.5	4.9	66.0	22.5	4.9	3.2	6.0	2.9	6.4	3.3	6.1
boat	81.3	43.7	97.1	55.9	1.6	35.0	16.3	1.5	38.2	17.7	06.7	5.5	12.5	6.2	13.6	56.1	18.6
bus	74.1	75.7	39.7	17.8	14.6	19.4	15.4	14.3	36.9	27.0	29.3	10.9	30.6	19.3	15.6	19.9	15.8
camel	59.3	43.8	34.1	23.2	12.3	17.2	12.9	13.8	38.0	16.1	08.4	5.5	11.7	6.2	21.2	9.5	16.8
car-roundabout	54.5	1.2	42.8	35.2	6.4	38.2	29.1	16.1	53.6	24.2	15.8	7.1	25.5	7.8	28.3	14.8	9.7
car-shadow	1.2	1.3	22.7	36.1	18.0	45.2	20.6	31.3	79.3	35.3	20.6	23.0	35.1	27.4	33.9	51.4	11.3
car-turn	27.4	71.4	47.7	23.5	11.8	20.2	20.4	10.8	56.6	21.4	79.6	11.7	13.5	6.5	25.6	9.8	10.8
cows	49.4	1.0	45.7	14.7	13.3	14.8	19.6	41.2	51.1	28.2	17.9	4.4	16.4	5.5	16.3	12.7	12.6
dance-jump	1.1	63.0	87.0	31.6	45.9	57.6	11.0	21.4	58.6	24.2	27.2	28.6	32.4	17.3	50.6	25.8	23.4
drift-straight	96.9	1.2	77.5	59.9	90.0	63.8	54.3	29.2	95.0	48.2	82.6	39.6	82.3	31.7	59.7	52.8	48.6
drift-turn	39.1	98.7	98.4	85.0	33.4	47.5	40.2	15.0	1.0	25.8	63.3	16.8	70.3	12.8	32.8	70.7	23.5
elephant	1.1	96.0	55.7	31.5	11.8	23.6	23.6	8.5	42.6	13.9	9.7	7.6	21.3	7.5	40.4	14.0	14.4
flamingo	52.3	31.9	76.9	13.8	17.3	21.5	38.2	11.3	48.6	17.5	11.8	6.9	13.3	8.9	18.2	13.4	13.7
hike	33.1	87.3	28.2	15.8	12.5	23.0	25.1	11.7	41.2	24.7	14.1	12.2	12.0	9.2	16.4	12.8	13.2
hockey	51.9	1.0	37.8	22.7	15.9	22.8	21.1	16.2	37.7	27.6	40.3	10.3	25.8	10.2	22.8	24.5	13.1
kite-surf	42.0	46.4	88.8	94.4	24.8	43.2	50.7	23.3	56.8	40.4	27.8	12.5	49.7	11.7	21.2	70.1	41.7
kite-walk	40.9	32.2	82.9	22.1	12.7	00.2	32.8	36.6	35.6	30.1	24.1	17.3	18.5	15.4	16.6	15.7	26.3
mallard-water	81.5	57.9	1.6	24.2	39.4	64.1	0.0	18.4	1.7	23.0	28.7	12.3	29.5	21.9	31.7	14.1	16.2
motocross-bumps	60.6	75.6	91.9	54.1	32.7	56.6	48.1	34.4	90.3	32.9	62.8	28.9	76.7	21.1	48.6	66.3	39.1
paragliding-launch	49.8	59.8	32.4	25.9	27.3	34.7	33.1	21.3	60.2	70.3	66.0	20.8	31.6	18.0	32.9	23.0	22.6
rhino	30.8	39.6	70.0	18.8	6.4	15.3	9.3	5.6	39.0	13.8	6.6	3.7	9.3	3.7	15.1	8.8	8.4
scooter-black	1.0	91.0	41.9	76.0	32.0	57.7	36.4	55.8	79.0	47.5	96.0	21.6	59.9	28.2	42.3	83.7	22.6
soapbox	98.3	61.3	50.4	39.0	15.4	41.3	21.4	16.0	61.3	15.8	76.3	11.2	31.4	12.6	37.9	23.5	21.8
stroller	1.2	96.8	86.4	20.5	11.6	23.5	36.6	12.7	54.6	18.4	13.4	13.0	36.3	18.9	37.6	25.1	21.4
surf	23.4	1.3	72.8	36.4	16.9	37.5	22.3	8.6	1.1	39.8	20.7	23.8	29.1	12.7	27.6	87.1	15.5
train	1.0	73.9	37.7	57.6	5.6	11.0	7.0	27.0	39.6	15.9	24.9	6.9	10.6	4.7	44.8	31.9	6.2
Mean	65.2	75.8	63.7	35.6	24.3	32.7	25.0	19.0	60.0	27.6	32.9	13.7	29.6	13.1	28.5	31.4	18.6

Table 7.9: *Results of temporal stability (T) for each video sequence in the dataset. The best performing method of each category is highlighted in bold. Please note that this measure is only computed on those sequences without occlusions and strong deformations.*

	Unsupervised							Semi-Supervised						
Attr	NLC	CVOS	TRC	MSG	KEY	SAL	FST	TSP	SEA	HVS	JMP	FCOP	BVS	MSK
AC	53 +13	41 +12	36 +17	47 +8	41 +19	33 +12	54 +4	17 +23	45 +12	41 +22	57 +4	50 +16	46 +26	76 +5
BC	45 +22	45 +6	50 -2	54 -1	52 +5	42 +0	57 -0	41 -7	58 -4	61 -4	60 -0	58 +5	63 +5	79 +2
CS	59 +6	41 +12	54 -6	53 -0	51 +6	36 +9	53 +5	34 +1	42 +17	54 +6	60 +0	60 +3	62 +6	78 +2
DB	52 +14	37 +18	38 +14	43 +14	51 +7	34 +1	52 +6	39 -6	57 -3	59 -1	59 +1	61 +1	60 +8	76 +6
DEF	67 -11	51 -1	48 +1	51 +5	57 -1	45 -8	56 +1	31 +1	49 +14	58 +0	58 +3	60 +4	70 -10	81 -1
EA	50 +24	40 +19	45 +7	45 +16	48 +15	35 +12	51 +1	31 +6	51 +7	53 +1	54 +9	57 +1	58 +17	75 +10
FM	62 +1	36 +25	40 +16	44 +15	49 +12	34 +13	49 +13	17 +31	39 +28	40 +31	49 +18	54 +13	54 +22	76 +8
HO	64 -4	49 +5	45 +14	53 +1	53 +12	42 -1	54 +1	27 +29	49 +23	53 +21	55 +17	59 +13	63 +14	79 +5
IO	61 +3	54 -6	46 +5	57 -7	53 +5	41 +2	48 +16	34 +3	53 +3	55 +6	58 +4	59 +7	63 +7	78 +6
LR	65 -3	40 +14	45 +5	48 +6	53 +5	32 +13	52 +6	29 +8	46 +11	46 +16	50 +13	58 +6	59 +10	77 +5
MB	59 +5	35 +23	31 +28	33 +3	50 +8	32 +15	47 +15	14 +32	39 +24	42 +25	50 +15	52 +15	58 +13	74 +9
OCC	68 -8	42 +13	42 +11	46 +11	51 +8	43 -2	52 +8	26 +14	46 +13	51 +12	46 +21	58 +7	68 -3	78 +4
OV	49 +17	34 +2	32 +21	39 +17	42 +18	30 +15	48 +1	20 +19	43 +14	39 +24	59 +0	51 +13	43 +29	72 +11
SC	59 +6	49 +2	46 +6	52 +1	50 +11	45 -5	51 +9	32 +5	50 +9	56 +5	51 +14	58 +7	67 -1	75 +9
SV	52 +16	42 +12	42 +11	50 +5	49 +1	33 +14	49 +11	23 +18	48 +9	44 +21	57 +4	51 +16	49 +26	73 +10

Table 7.10: *Attribute-based aggregate performance. For each method, the respective left column corresponds to the average region similarity $\mathcal{J}$ over all sequences with that specific attribute (e.g., AC), while the right column indicates the performance gain (or loss) for that method for the remaining sequences without that respective attribute. The best performing method of each category is highlighted in bold.*

CHAPTER 8

Conclusion

We began our studies in Chapter 3 revising the notion of color contrast within the domain of salient object detection. Perceptual studies suggest color contrast to play a key role steering human gaze. Previous approaches implemented contrast based on different type of images features such as color histograms, multi-scale descriptors, structure and repetitivity of image patches, edges gradients and spatial frequencies or combinations thereof. The varying performance of similar approaches led us to conclude that the relevance of each features was unclear. To this end, we reconsidered some of the design choices of previous methods and proposed a conceptually clear and intuitive algorithm for contrast-based saliency estimation. Our first contribution was a novel derivation of saliency based on color uniqueness and spatial distribution computed over perceptually homogeneous element that abstract away unnecessary details. Our second contribution a was unified way to handle the definition of color measures using high-dimensional Gaussian filters. We demonstrated that the entire algorithm can be implemented within a single high-dimensional Gaussian filtering framework yielding linear time complexity.

While the concept of saliency based on color contrast has proven to be overall effective, some of the assumptions underlying its definition are in practice often violated in natural images. In particular, methods based on color contrast appeared to be less suitable for circumstances such as the case of multiple unique colors or multicolored objects. To this end we proposed a new method with orthogonal properties to the aforementioned approach based on the assumption that most of the image boundaries are covered by non-salient background. In our exploration we found the Fiedler vector of

the Laplacian graph built over image superpixels to be a robust saliency indicator. The image graph augmented with a dummy node representing the image background conveniently encodes the background prior and local perceptual similarities. As a result, we demonstrated that simple eigen decomposition is sufficient to reliably estimate saliency.

In the second part of the thesis we focused our attention towards developing novel semi-automatic techniques aimed to segment foreground objects in videos. We developed two novel approaches reaching state-of-the-art performance, while using different type of annotations such as segments, object proposals and bounding boxes. Motivated by the successes that object proposals enabled in the domain of object detection, several video object segmentation methods investigated techniques to select a set of proposals, one for each frame. These proposals are generally employed as a rough initial segmentation and later refined via different nuances of graph-cuts. Despite these methods having achieved good performance, we observed that the sparsity of the graph upon which they minimize the energy function selecting proposals, is brittle in challenging scenarios such as in the presence of occlusions or fast motion. Therefore, differently from previous methods that select the best proposal per-frame, we study the problem of grouping multiple, possibly incomplete proposals, that overlap with the foreground object(s) to be segmented. To this end, our main contribution was a novel approach that fully connects the entire set of object proposals and select a subset of those overlapping with the foreground object based on the maximum a posteriori of the resulting conditional random field. We map our similarity term into a Euclidean space, which is computationally efficient to optimize and well suited for modeling long-range connections. Message passing throughout the entire set of proposals coupled with sparse but confident point-tracks based features ensure robustness to occlusions and other difficult situations as demonstrated in the evaluation of Section 7.4.2. Later, motivated by the unprecedented advances of deep-learning we investigated its application in the field of video object segmentation. Our contribution is a novel approach which incorporates the concept of guidance in fully convolutional networks designed for semantic segmentation. Processing a video frame by frame, our network is directed toward the object of interest through either a bounding box or the previous frame segmentation mask. The architecture is derived from a well known per-pixel semantic segmentation convnet, and adapted to be class agnostic. The network is conveniently trained only on static images, therefore overcoming the scarcity of densely annotated video data.

In the third part of the thesis we investigated a novel approach suitable for interactive video segmentation. During our studies on semi-automatic

video segmentation, we observed that the fully connected object proposal approach is robust to occlusion and exhibited minor performance decay over time, however, depending on the size of the pool of proposals and the length of the video, it's usage might not be practical due to heavy memory footprint and running time. While the problem could be overcome with a naive sliding window approach, we study a more principled way to incorporate local and global connections efficiently. To this end we proposed a novel approach that implicitly approximates long-range, spatio-temporal connections between pixels while limiting the number of graph nodes and edges. Our main contribution was a novel energy function that is defined on the vertices of a regularly sampled spatiotemporal bilateral grid and has only local edge connectivity. The formulation yield an efficient minimization problem that can be solved using a standard graph cut label assignment. To further improve run-time efficiency while preserving the segmentation quality, we proposed a fast "adjacent" interpolation scheme for high-dimensional grids. As a result, our method is highly efficient and scales linearly with image resolution, allowing the user to interact with the algorithm to correct possible inaccuracies.

In the fourth and last part of the thesis we address the lack of densely annotated video data. When we began exploring the topic of video object segmentation the most commonly used benchmark in this domain only included about fifteen sequences of mixed resolution videos. While this dataset has been a driving force for research on this topic, several years of progress led to saturation of the results making difficult to appreciate the real performance of the new approaches. In addition, a clear evaluation protocol was not yet defined and different measures were being used. To this end we proposed a new dataset and evaluation methodology specifically designed for the topic of video object segmentation. Furthermore we analyzed and provided an extensive analysis of several state-of-the-art approach, including those we proposed in this thesis. Besides establishing three measures to evaluate region accuracy, boundary and temporal stability, we proposed an attributed-based evaluation, where these attributes, manually assigned to each video sequence, allowed us to identify groups of videos with a dominant feature e.g., presence of occlusions, which is key to explaining the algorithms' performance. This type of analysis lead us to get a deeper understanding of the strength and weaknesses of the benchmarked approaches. The attention gained by our latest work on video object segmentation and the experience collected during these studies motivated us to organized the first workshop on video object segmentation.

8.1 Future Works.

In this section we summarize potential avenues for future for research. Several ideas have been already outlined in the concluding paragraphs of the respective chapters.

Salient Object Detection. The field of salient object detection has recently witnessed substantial advances due to the rise of deep learning and several approaches have been proposed that demonstrated superior performance on more challenging datasets, thanks to the power of extracted deep features. While theoretically the network should implicitly learn high level assumptions, *e.g*. color-contrast and non-salient boundaries, it might be beneficial and certainly interesting to investigate how these observation could be explicitly incorporated into the networks.

Video Object Segmentation. In this domain we proposed one of the first approaches that use and end-to-end trained convnet. Our method demonstrated high performance but only partially leverages the entire information present in the video signal. Therefore we believe future research in this domain should exploit different architectures such as LSTM Recurrent Neural Networks that naturally incorporate and exploit previous information. This could help to better disambiguate challenging scenarios such as fast motion, or appearance changes. Another open question which remained unresolved in our research is whether the pairwise-frame motion information helps, or given the extreme discriminative power of deep-features, it is redundant or misleading when inaccurate. Another important information that is currently not exploited is a prior knowledge of the object shape. Learning a manifold embedding for specific classes and navigating this manifold to perform the segmentation is certainly an interesting topic for future research. While the per-pixel classification style of segmentation has lot of potential for future research we also believe that novel approaches that operate on parametric shapes should be investigate. This type of approaches will have immediate application in the visual effects industry as they would seamlessly merging into the industry workflow helping the artists to get better results faster.

Dataset and Benchmarks. Producing per-pixel accurate and temporally stable segmentation masks for videos is a tedious and costly job. Due to budget limitations our first data releases comprised a total of fifty videos for a total of about 3.5K frames. Despite the large amount of data the high correlation between frames of the same video sequence prevents the training of deep networks. Furthermore, to reduce costs we constrained the footage to have a single foreground object, yielding a slight bias towards salient mo-

tion being well aligned with the foreground object. As mentioned in the introduction and the respective chapter, this dataset became part of a bigger project and a it resulted in a Workshop co-located with CVPR 2017. We have collected more funding from academic and corporate sponsors and we are planning to release another set of 100 annotated videos with multiple objects annotated and novel challenges such as multi-object instance segmentation. Future works comprises even mores video and a fully automated evaluation platform.

References

[Achanta et al., 2008] Radhakrishna Achanta, Francisco J. Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. In *ICVS*, pages 66–75, 2008.

[Achanta et al., 2009] Radhakrishna Achanta, Sheila S. Hemami, Francisco J. Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.

[Achanta et al., 2012] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE PAMI*, 34(11):2274–2282, 2012.

[Adams et al., 2010] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. *Comput. Graph. Forum*, 29(2):753–762, 2010.

[Agarwala et al., 2004] Aseem Agarwala, Aaron Hertzmann, David H Salesin, and Steven M Seitz. Keyframe-based tracking for rotoscoping and animation. *SIGGRAPH*, 2004.

[Arbeláez et al., 2014] Pablo Andrés Arbeláez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marqués, and Jitendra Malik. Multiscale combinatorial grouping. In *Proc. CVPR*, 2014.

[Avidan and Shamir, 2007] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3):10, 2007.

[Badrinarayanan et al., 2010] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label propagation in video sequences. In *CVPR*, 2010.

[Bai et al., 2009] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Trans. Graph.*, 28(3), 2009.

[Bailer et al., 2015] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015.

[Barron et al., 2015] Jonathan T Barron, Andrew Adams, YiChang Shih, and Carlos Hernández. Fast bilateral-space stereo for synthetic defocus. In *CVPR*, 2015.

[Belongie et al., 2002a] Serge Belongie, Charless Fowlkes, Fan R. K. Chung, and Jitendra Malik. Spectral partitioning with indefinite kernels using the nyström extension. In *Proc. ECCV*, 2002.

[Belongie et al., 2002b] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[Bookstein, 1989] Fred Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *PAMI*, 1989.

[Borji et al., 2012] Ali Borji, Dicky N. Sihite, and Laurent Itti. Salient object detection: A benchmark. In *ECCV (2)*, pages 414–429, 2012.

[Boykov and Jolly, 2001] Y Y Boykov and M P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, 2001.

[Boykov and Kolmogorov, 2004] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE TPAMI*, 26(9):1124–1137, 2004.

[Boykov et al., 1999] Y Boykov, O Veksler, and R Zabih. Fast approximate energy minimization via graph cuts. In *ICCV*, 1999.

[Brendel and Todorovic, 2009] William Brendel and Sinisa Todorovic. Video object segmentation by tracking regions. In *Proc. ICCV*, 2009.

[Brostow et al., 2009] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2), 2009.

[Brox and Malik, 2010] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Proc. ECCV*, 2010.

[Butler et al., 2012] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[Campbell et al., 2013] Neill D. F. Campbell, Kartic Subr, and Jan Kautz. Fully-connected crfs with non-parametric pairwise potential. In *Proc. CVPR*, 2013.

[Carreira and Sminchisescu, 2012] João Carreira and Cristian Sminchisescu. CPMC: automatic object segmentation using constrained parametric min-cuts. *IEEE TPAMI*, 34(7):1312–1328, 2012.

[Carreira et al., 2012] João Carreira, Fuxin Li, and Cristian Sminchisescu. Object recognition by sequential figure-ground ranking. *IJCV*, 98(3):243–262, 2012.

[Chang et al., 2011] Kai-Yueh Chang, Tyng-Luh Liu, and Shang-Hong Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *CVPR*, 2011.

[Chang et al., 2013] Jason Chang, Donglai Wei, and John W. Fisher III. A video representation using temporal superpixels. In *CVPR*, 2013.

[Chaudhry et al., 2009] Rizwan Chaudhry, Avinash Ravichandran, Gregory D. Hager, and René Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Proc. CVPR*, 2009.

[Chen and Corso, 2010] A Y C Chen and J J Corso. Propagating multi-class pixel labels throughout video frames. In *WNYIPW*, 2010.

[Chen et al., 2007] Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. *SIGGRAPH*, 2007.

[Chen et al., 2016] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016.

[Cheng et al., 2011] Ming-Ming Cheng, Guo-Xin Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416, 2011.

[Cheng et al., 2014] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip H. S. Torr. BING: binarized normed gradients for objectness estimation at 300fps. In *Proc. CVPR*, 2014.

[Cheng et al., 2015] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *TPAMI*, 2015.

[Chockalingam et al., 2009] Prakash Chockalingam, S. Nalin Pradeep, and Stan Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, 2009.

[Choi et al., 2012] Inchang Choi, Minhaeng Lee, and Yu-Wing Tai. Video Matting Using Multi-frame Nonlocal Matting Laplacian. In *ECCV*, 2012.

[Chuang et al., 2002] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H Salesin, and Richard Szeliski. Video matting of complex scenes. In *SIGGRAPH*, 2002.

[Collins et al., 2005] Robert Collins, Xuhui Zhou, and Seng Keat Teh. An open source tracking testbed and evaluation web site. In *PETS 2005*, January 2005.

[Cordts et al., 2016] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Cox and Cox, 1994] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994.

[Criminisi et al., 2010] Antonio Criminisi, Toby Sharp, Carsten Rother, and Patrick Pérez. Geodesic image and video editing. *ACM Trans. Graph.*, 29(5):134, 2010.

[Csurka et al., 2013] Gabriela Csurka, Diane Larlus, and Florent Perronnin. What is a good evaluation measure for semantic segmentation? In *BMVC*, 2013.

[Danelljan et al., 2015] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCV workshop*, pages 58–66, 2015.

[Danelljan et al., 2016] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*. Springer, 2016.

[de Silva and Tenenbaum, 2002] Vin de Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Proc. NIPS*, 2002.

[Doersch et al., 2013] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *Advances in Neural Information Processing Systems (NIPS)*, pages 494–502, 2013.

[Dollár and Zitnick, 2013] Piotr Dollár and C. Lawrence Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[Dolson et al., 2010] Jennifer Dolson, Jongmin Baek, Christian Plagemann, and Sebastian Thrun. Upsampling range data in dynamic environments. In *CVPR*, pages 1141–1148, 2010.

[Dondera et al., 2014] Radu Dondera, Vlad Morariu, Yulu Wang, and Larry Davis. Interactive video segmentation using occlusion boundaries and temporally coherent superpixels. In *2014 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 784–791. IEEE, 2014.

[Duan et al., 2011] Lijuan Duan, Chunpeng Wu, Jun Miao, Laiyun Qing, and Yu Fu. Visual saliency detection by spatially weighted dissimilarity. In *CVPR*, pages 473–480, 2011.

[Duffner and Garcia, 2013] Stefan Duffner and Christophe Garcia. Pixeltrack: A fast adaptive algorithm for tracking non-rigid objects. In *ICCV*, 2013.

[Einhauser et al., 2003] W. Einhauser, P. Konig, and P. Konig. Does luminance-contrast contribute to a saliency map for overt visual attention? *European Journal of Neuroscience*, 17(5), 2003.

[Endres and Hoiem, 2014] Ian Endres and Derek Hoiem. Category-independent object proposals with diverse ranking. *IEEE TPAMI*, 36(2):222–234, 2014.

[Everingham et al., 2010] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2), 2010.

[Everingham et al., 2012] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[Faktor and Irani, 2014] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014.

[Fan et al., 2015] Qingnan Fan, Fan Zhong, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Jumpcut: Non-successive mask transfer and interpolation for video cutout. *SIGGRAPH Asia*, 2015.

[Fathi et al., 2011] Alireza Fathi, Xiaofeng Ren, and James M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011.

[Feng et al., 2011] Jie Feng, Yichen Wei, Litian Tao, Chao Zhang, and Jian Sun. Salient object detection by composition. In *ICCV*, 2011.

[Fisher, 2004] R. B. Fisher. The pets04 surveillance ground-truth data sets. 2004.

[Fragkiadaki and Shi, 2011] Katerina Fragkiadaki and Jianbo Shi. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *CVPR*, 2011.

[Fragkiadaki et al., 2012] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012.

[Galasso et al., 2013] Fabio Galasso, Naveen Shankar Nagaraja, Tatiana Jimenez Cardenas, Thomas Brox, and Bernt Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013.

[Girshick et al., 2014] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014.

[Goferman et al., 2010] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. In *CVPR*, pages 2376–2383, 2010.

[Gorelick et al., 2007] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *TPAMI*, 29(12), 2007.

[Grundmann et al., 2010] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan A. Essa. Efficient hierarchical graph-based video segmentation. In *Proc. CVPR*, 2010.

[Guo et al., 2008] Chenlei Guo, Qi Ma, and Liming Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In *CVPR*, 2008.

[Hagen and Kahng, 1992] Lars W. Hagen and Andrew B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[Harel et al., 2006] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *NIPS*, pages 545–552, 2006.

[Hariharan et al., 2015] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[He et al., 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[Held et al., 2016] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016.

[Hickson et al., 2014] Steven Hickson, Stan Birchfield, Irfan A. Essa, and Henrik I. Christensen. Efficient hierarchical graph-based segmentation of RGBD videos. In *Proc. CVPR*, 2014.

[Hou and Zhang, 2007] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *CVPR*, 2007.

[Itti and Baldi, 2005] Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. In *NIPS*, 2005.

[Itti et al., 1998] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 20(11):1254–1259, 1998.

[Jain and Grauman, 2014] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014.

[Kadir and Brady, 2001] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[Koch and Ullman, 1985] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985.

[Kohli and Torr, 2007] P Kohli and P H S Torr. Dynamic Graph Cuts for Efficient Inference in Markov Random Fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2079–2088, 2007.

[Krähenbühl and Koltun, 2011] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Proc. NIPS*, 2011.

[Krähenbühl and Koltun, 2014] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *Proc. ECCV*, 2014.

[Lee et al., 2011] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *Proc. ICCV*, 2011.

[Li and Yu, 2016a] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. *CoRR*, abs/1603.01976, 2016.

[Li and Yu, 2016b] Guanbin Li and Yizhou Yu. Visual saliency detection based on multiscale deep CNN features. *IEEE Trans. Image Processing*, 25(11):5012–5024, 2016.

[Li et al., 2005] Yin Li, Jian Sun, Heung-Yeung Shum, Yin Li, and Jian Sun. Video object cut and paste. *ACM Transactions on Graphics (TOG)*, 24(3):595–600, July 2005.

[Li et al., 2013] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

[Li et al., 2014] Yin Li, Xiaodi Hou, Christof Koch, James M. Rehg, and Alan L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014.

[Lin et al., 2014] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[Liu and Han, 2016] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Liu et al., 2007] Tie Liu, Jian Sun, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. In *CVPR*, 2007.

[Liu et al., 2011] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *TPAMI*, 33(2), 2011.

[Lowe, 2004] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[Lu et al., 2011] Yao Lu, Wei Zhang, Hong Lu, and Xiangyang Xue. Salient object detection using concavity context. In *ICCV*, pages 233–240, 2011.

[Ma and Latecki, 2012] Tianyang Ma and Longin Jan Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *Proc. CVPR*, 2012.

[Ma and Zhang, 2003] Yu-Fei Ma and HongJiang Zhang. Contrast-based image attention analysis by using fuzzy growing. In *ACM Multimedia*, pages 374–381, 2003.

[Maerki et al., 2016] Nicolas Maerki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016.

[Maninis et al., 2016] K.K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool. Convolutional oriented boundaries. In *ECCV*, 2016.

[Martin et al., 2001] David R. Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[Martin et al., 2004] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *TPAMI*, 26(5), 2004.

[Meinshausen and Bühlmann, 2010] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[ming Cheng et al.,] Ming ming Cheng, Jonathan Warrell, Wen yan Lin, Shuai Zheng, Vibhav Vineet, and Nigel Crook. Efficient salient region detection with soft image abstraction. In *in IEEE ICCV*, pages 1529–1536.

[Movahedi and Elder, 2010] Vida Movahedi and James H. Elder. Design and perceptual validation of performance measures for salient object segmentation. In *CVPR Workshops*, 2010.

[Nam and Han, 2016] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[Ng et al., 2002] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[Ochs and Brox, 2011] Peter Ochs and Thomas Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *Proc. ICCV*, 2011.

[Oh et al., 2011] Sangmin Oh, Anthony Hoogs, A. G. Amitha Perera, and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011.

[Pan et al., 2016] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E. O'Connor. Shallow and deep convolutional networks for saliency prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Papazoglou and Ferrari, 2013] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.

[Parkhurst et al., 2002] D. Parkhurst, K. Law, and E. Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, 42(1):107–123, 2002.

[Pathak et al., 2016] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[Pedregosa et al., 2011] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay.

Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Perazzi et al., 2012] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Proc. CVPR*, 2012.

[Perazzi et al., 2015a] Federico Perazzi, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. Efficient Salient Foreground Detection for Images and Video using Fiedler Vectors. In W. Bares, M. Christie, and R. Ronfard, editors, *Eurographics Workshop on Intelligent Cinematography and Editing*. The Eurographics Association, 2015.

[Perazzi et al., 2015b] Federico Perazzi, Oliver Wang, Markus Gross, and Alexander Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, 2015.

[Pinheiro et al., 2015] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In *NIPS*, 2015.

[Platt, 1999] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, pages 61–74, 1999.

[Platt, 2005] John C. Platt. Fastmap, metricmap, and landmark mds are all nystrom algorithms. In *Proc. Workshop on Artificial Intelligence and Statistics*, pages 261–268, 2005.

[Pont-Tuset and Marques, 2015] Jordi Pont-Tuset and Ferran Marques. Supervised evaluation of image segmentation and object proposal techniques. *TPAMI*, 2015.

[Pont-Tuset et al., 2016] Jordi Pont-Tuset, Pablo Arbeláez, Jonathan T. Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *TPAMI*, 2016.

[Prest et al., 2012] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[Price et al., 2009] Brian L Price, Bryan S Morse, and Scott Cohen. LIVEcut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, 2009.

[Rabiner and Juang, 1993] Lawrence Rabiner and Biing-Hwang Juang. Fundamentals of speech recognition. 1993.

[Ramakanth and Babu, 2014] S. Avinash Ramakanth and R. Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *CVPR*, 2014.

[Ravikumar et al., 2010] Pradeep Ravikumar, Martin J Wainwright, John D Lafferty, et al. High-dimensional ising model selection using l1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

[Reinagel and Zador, 1999] Pamela Reinagel and Anthony M Zador. Natural scene statistics at the centre of gaze. In *Network: Computation in Neural Systems*, pages 341–350, 1999.

[Ren and Malik, 2007] Xiaofeng Ren and Jitendra Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, 2007.

[Ren and Philipose, 2009] Xiaofeng Ren and M. Philipose. Egocentric recognition of handled objects: Benchmark and analysis. In *CVPR Workshops*, 2009.

[Ren et al., 2010] Zhixiang Ren, Yiqun Hu, Liang-Tien Chia, and Deepu Rajan. Improved saliency detection based on superpixel clustering and saliency propagation. In *ACM Multimedia*, pages 1099–1102, 2010.

[Ren et al., 2015] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[Reso et al., 2014a] Matthias Reso, Jörn Jachalsky, Bodo Rosenhahn, and Jörn Ostermann. Superpixels for Video Content Using a Contour-Based EM Optimization. *ACCV*, 9006(Chapter 45):692–707, 2014.

[Reso et al., 2014b] Matthias Reso, Björn Scheuermann, Jörn Jachalsky, Bodo Rosenhahn, and Jörn Ostermann. Interactive Segmentation of High-Resolution Video Content Using Temporally Coherent Superpixels and Graph Cut. In *Advances in Visual Computing*. Springer International Publishing, Cham, December 2014.

[Revaud et al., 2015] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.

[Rother et al., 2004] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2004.

[Russakovsky et al., 2014] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[Shen et al., 2015] Jianbing Shen, Wenguan Wenguan, and Fatih Porikli. Saliency-Aware geodesic video object segmentation. In *CVPR*, 2015.

[Shi and Malik, 1997] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *CVPR*, 1997.

[Shi et al., 2016] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended CSSD. *TPAMI*, 2016.

[Simonyan and Zisserman, 2015] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Singh et al., 2012] Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, 2012.

[Sundberg et al., 2011] Patrik Sundberg, Thomas Brox, Michael Maire, Pablo Arbelaez, and Jitendra Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011.

[Taylor et al., 2015] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, 2015.

[Torralba and Efros, 2011] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR'11*, June 2011.

[Tron and Vidal, 2007] Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, 2007.

[Tsai et al., 2010] David Tsai, Matthew Flagg, and James M. Rehg. Motion coherent tracking with multi-label MRF optimization. In *British Machine Vision Conference, BMVC 2010, Aberystwyth, UK, August 31 - September 3, 2010. Proceedings*, pages 1–11, 2010.

[Tsai et al., 2016] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *CVPR*, 2016.

[Van den Bergh et al., 2013] Michael Van den Bergh, Gemma Roig, Xavier Boix, Santiago Manen, and Luc J Van Gool. Online Video SEEDS for Temporal Window Objectness. 2013.

[Varas and Marqués, 2014] David Varas and Ferran Marqués. Region-based particle filter for video object segmentation. In *Proc. CVPR*, 2014.

[Vijayanarasimhan and Grauman, 2012] Sudheendra Vijayanarasimhan and Kristen Grauman. Active frame selection for label propagation in videos. In *ECCV*, 2012.

[von Luxburg, 2007] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[Wang and Japkowicz, 2008] Benjamin X. Wang and Nathalie Japkowicz. Boosting support vector machines for imbalanced data sets. In *Proc. ISMIS*, 2008.

[Wang et al., 2005] Jue Wang, Pravin Bhat, R Alex Colburn, Maneesh Agrawala, and Michael F Cohen. Interactive video cutout. *ACM Transactions on Graphics (TOG)*, 24(3):585–594, July 2005.

[Wang et al., 2011] Meng Wang, Janusz Konrad, Prakash Ishwar, Kevin Jing, and Henry A. Rowley. Image saliency: From intrinsic to extrinsic context. In *CVPR*, pages 417–424, 2011.

[Wang et al., 2014] Tinghuai Wang, Bo Han, and John Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *CVIU*, 2014.

[Wang et al., 2015] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015.

[Wang et al., 2016] Huiling Wang, Tapani Raiko, Lasse Lensu, Tinghuai Wang, and Juha Karhunen. Semi-supervised domain adaptation for weakly labeled semantic video object segmentation. In *ACCV*, 2016.

[Wei et al., 2012] Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun. Geodesic saliency using background priors. In *ECCV (3)*, pages 29–42, 2012.

[Wen et al., 2015] Longyin Wen, Dawei Du, Zhen Lei, Stan Z Li, and Ming-Hsuan Yang. Jots: Joint online tracking and segmentation. In *CVPR*, 2015.

[Wu et al., 2013] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[Xiao and Lee, 2016] Fanyi Xiao and Yong Jae Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016.

[Xu and Corso, 2012] Chenliang Xu and Jason J. Corso. Evaluation of super-voxel methods for early video processing. In *Proc. CVPR*, 2012.

[Zhang et al., 2013] Dong Zhang, Omar Javed, and Mubarak Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *Proc. CVPR*, 2013.

[Zhang et al., 2015] Yu Zhang, Xiaowu Chen, Jia Li, Chen Wang, and Changqun Xia. Semantic object segmentation via detection in weakly labeled video. In *CVPR*, 2015.

[Zhang et al., 2016] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. *ECCV*, 2016.

[Zhao et al., 2015] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274. IEEE Computer Society, 2015.

[Zhong et al., 2012] Fan Zhong, Xueying Qin, Qunsheng Peng, and Xiangxu Meng. Discontinuity-aware video object cutout. *ACM Transactions on Graphics (TOG)*, 31(6):175, November 2012.