Visual Parsing with Weak Supervision

by

Jia Xu

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2015

Date of final oral examination: 07/30/2015

The dissertation is approved by the following members of the Final Oral Committee:

Vikas Singh, Associate Professor, Biostatistics and Medical Informatics
Charles R. Dyer, Professor, Computer Sciences
Xiaojin (Jerry) Zhu, Associate Professor, Computer Sciences
Jude W. Shavlik, Professor, Computer Sciences
Mark Craven, Professor, Biostatistics and Medical Informatics

To my family.

ACKNOWLEDGMENTS

Coming across the ocean to finish this thesis has been a surreal journey. Along this remarkable ride, I first and foremost want to thank my advisor, Vikas Singh. I began to know Vikas when I was an undergraduate (almost six years ago). His series of stellar cosegmentation papers read so elegantly to me, that I feel absolutely obsessed with solving computer vision problems in such beautiful ways. So, the first week after I landed in Wisconsin, I started to work with Vikas.

The first year was rough, and I still remember how easy it was for me to run out of vocabulary. Vikas has been extremely patient with me. As a mentor, Vikas took every opportunity to help me improve and become stronger. He spent numerous hours teaching me mathematical fundamentals, introduced me in and guided me through multiple projects, advised me to take multiple useful optimization courses, and sent me to summer schools and academic conferences when I was starting my graduate study. It was Vikas who enhanced my curiosity in computer vision, and shaped me into an independent researcher.

In spite of being a very supportive mentor, Vikas is also a creative thinker. I particularly like his philosophy of "a good idea should stand on its own". His big thinking has inspired me to keep up as best I could. During the last two years, Vikas has offered me enough freedom to pursue problems I am interested in, and cared about me as a family member. More than that, Vikas has always been a strong advocate for me. I am thankful to have Vikas as my advisor.

My deep gratitude also goes to my other thesis committee members: Chuck Dyer, Jerry Zhu, Jude Shavlik, and Mark Craven. Chuck has been super kind and encouraging to me. In the past year, we have been collaborating on the project of inferring air quality using social media images. Chuck has opened my eyes on computational photography and always

motivated me to look at computer vision problems from the physics perspective. Discussing with Chuck is also very inspiring, and I always get new ideas in our regular meetings. I am indebted for all his feedback and help, and hope to continue our long term collaboration.

Jerry is my role model in many aspects. In my first two years in graduate school, I read dozens of Jerry's papers, studied his Advanced Machine Learning class, and enjoyed all of them. Jerry's lectures and papers are super clear and precise. His presentations have been perfect examples for me to learn on how to give a good talk. Jerry also asks very insightful questions, even as a seminar audience. His questions provided me a great deal of food for thoughts. In the past year, I have been discussing a few projects with Jerry. His critical thinking has inspired me to conduct highly original and influential research. I benefited enormously from interacting with Jerry.

I would also like to thank Jude and Mark for gracefully serving on my thesis committee. Jude's enthusiasm on 'learning from advice' has motivated many problems studied in my thesis research. I highly appreciate the detailed comments Jude gave me on my dissertation. I also enjoyed many of my conversations with Mark, and I am thankful to him for allocating time for reading and giving feedback on my thesis.

I am deeply grateful to Raquel Urtasun. Working with Raquel has been a fantastic experience. Raquel's unique perspective on graphical model provided me enormous new insights on my research. During many of our scientific discussions via Skype or in person, Raquel has taught me many important aspects including time management, communication, writing, and having fun in research. Most importantly, I thank Raquel for sharing personal advice with me, and providing me support for my research and career in numerous ways. I also want to thank Alex Schwing for being an excellent collaborator and friend.

I am also lucky to have Jim Rehg as a great mentor and collaborator.

Two chapters in this thesis are from our long collaboration. Jim's sharp thinking has always made our discussion more clear and fun. His vast experience in video analysis and egocentric vision contributed tremendous novel insights into my thesis.

This thesis could not have been possible without my awesome collaborators: Maxwell Collins, Chuck Dyer, Leo Grady, Vamsi Ithapu, Hyunwoo Kim, Yin Li, Zhe Lin, Ji Liu, Lopamudra Mukherjee, Jim Rehg, Alexander Schwing, Xiaohui Shen, Vikas Singh, Raquel Urtasun, Baba Vemuri, Jamie Warner, Jerry Zhu.

I thank many senior researchers in the computer vision community who have given me encouraging advice and insightful feedback during my PhD study: Vladlen Koltun, Sebastian Nowozin, Vittorio Ferrari, Martial Hebert, Yaser Sheikh, Simon Lucey, Abhinav Gupta, Kris Kitani, Alexei Efros, Yuanqing Lin, Xiaofeng Ren, Jonathan Brandt, Zhouchen Lin, Jianxin Wu, Subhransu Maji, Greg Shakhnarovich, and many others.

I want to personally thank the wonderful administrative team in both the computer sciences department and biostatistics department who has helped me in numerous ways over the past five years. In particular, I am thankful to Angela Thorp for providing me many many letters and suggestions. It was also a pleasure to stop by her office, and Angela is always friendly and helpful. I also want to thank Cathy Whitford in the biostatistics department. Cathy is an amazing department manager, and she has helped me in multiple scenarios.

I feel lucky to have my lovely lab mates during the past five years. Deepti Pachauri has been my big sister to take care of me when I was down. She always gave me good advice and cared about me. I will miss those coffee breaks/discussions we had in WID. Maxwell Collins has been the "walking Wikipedia" in our lab and he is extremely patient and helpful to answer almost any question ranging from system issues to mathematical definitions. My first few papers were written with Maxwell,

and I learnt and benefited quite a lot from him. Hyunwoo Kim is simply a true friend. He is exceptionally smart and hard-working. Plus, Hyumwoo always stays modest. I am particularly grateful to Hyunwoo for bringing flowers to me in the graduation commencement. It is joyful to hangout with WonHwa Kim. I enjoyed sharing two common interests with him: basketball and sushi, and many related conversations. WonHwa has also been very accommodating when I was in Seoul and in the lab. Vamsi Ithapu is a fun friend to work and talk with. I thank Vamsi for the days and nights he put in when we were working on our GOSUS paper. Sathya Ravi is also a genuine friend to talk with. Jamie Warner is a super smart undergraduate and it is a pleasure to mentor him in our egocentric video summarization project. Nagesh Adluru is a great and humble friend. I thank Nagesh, Anusha, and Serena for making their home a sweet place for many get-togethers. Lopa Mukherjee is a terrific collaborator, and I enjoyed working with her. I also thank Chris Hinrichs, Kamiya Motwani, Qinyuan Sun, Greg Plumb, Chris Lindner, Seong Jae Hwang for sharing those good days in the same lab.

I also thank folks I interacted with in TTI-Chicago and University of Toronto: Sanja Fideler, Jian Yao, Shenlong Wang, Liang-Chieh Chen, Edgar Simo-Serra, Wenjie Luo, Kaustav Kundu, Gellert Mattyus, Vikas Garg, Abhishek Sen, Chen Kong, Yali Wang, Zhou Ren, Roozbeh Mottaghi, Jian Peng, Jianzhu Ma, Yuan Zhou, Qingming Tang, and many others.

The past five years in Wisconsin have been wonderful due to my lovely friends here: Junming Xu, Lanyue Lu, Wentao Wu, Xiang Peng, Jiexin Li, Jie Liu, Ji Liu, Jianqiao Zhu, Chien-Ming Huang, Ce Zhang, Yimin Tan, Xiaoming Shi, Brandon Smith, Shengqi Zhu, Yiqing Yang, Bryan R. Gibson, Shike Mei, Kwang-Sung Jun, Bo Li, Nate Fillmore, Collin Engstrom, Erdem Kaya, Tyler Adelung, Dallas Wulf, Matthew Starr, Elizabeth Soechting, Yinan Li, Yupu Zhang, Lei Kang, Jun He, Xiaodong Ma, Zhipeng Su, Xiaolei Xie, Jiankun Shao, Zexian Zeng, Feng Ju, Hua Deng, Xiaoping

Bao, Yongjia Song, Wenjiang Ma, Xin Chen, Hao Zeng, Mingwei Huang, Xiaojun Tan, Yan Wang, Fei Meng, Zufang Shan, Huifang Xu, and many others. I also enjoyed sharing offices with Jeremy Weiss, Bess Berg, Gellert Mattyus, Jialiang Wang, Uri Priel, and Kamyar Ghasemipour. I thank my awesome roommates Yang Yang, Linhai Song, Huan Wang, Yan Cui, Jianzhu Ma, Joanna Drummond, Shengyuan Huang.

Finally, I thank my parents and my sister, for their tremendous love and unconditional support.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Visual parsing is one of the most fundamental problems in computer vision. The goal of visual parsing is to decompose arbitrary images and videos into their constituent visual components, and label them with a semantic meaning when possible. It serves as the first step towards many high level applications including robotics, autonomous driving, human-computer interaction, medical image analysis, and visual surveillance. Visual parsing also provides automatic tools to make sense of and summarize massive visual data.

Despite promising performance from conventional systems, visual parsing has remained an important yet challenging problem. Traditional fully supervised parsing methods often require a large training set where each pixel is labeled. However, such full annotations are often only available at a limited size or are extremely expensive to collect. Therefore, it is essential to come up with solutions that can learn to parse from weak human supervision or weakly labeled data, as it is much cheaper to collect or readily available at much larger scale. This thesis research focuses on visual parsing under weak supervision, with a particular focus on addressing the following questions:

1. How can we utilize weakly labeled data effectively for the visual parsing task?

2. When human comes into the visual parsing loop, how can we minimize user effort while still achieving satisfactory parsing results?

This thesis provides efficient approaches for cutting out foreground objects from background in an image, semantically labeling each pixel in an image, spatio-temporally parsing a video into objects, and temporally parsing a video into informative components.

1 INTRODUCTION

We live in an age of rich artificial intelligence. An increasing number of novel technologies (as shown in Fig. 1.1) have emerged and positively impacted our quality of life. For instance, Google's self-driving cars have logged over 1 million autonomous miles[1]. Da-Jiang Innovations (DJI) ships more than 30,000 drones every month[2]. Many companies (e.g., Microsoft, Facebook, Google, DAQRI, Magic Leap) have delivered augmented reality products including Hololens, Oculus Gear VR, Google glass, DAQRI Smart Helmet.

While these technologies have made human life more convenient and safer, visual parsing plays an essential role behind them. To drive safely, the first goal is to identify where the road is, and what is in front of a vehicle. To fly stably in the air, it is crucial for drones to locate the ground and obstacles, and navigate without hitting them. It is also important to recognize and track targets (e.g., person of interest) for monitoring and rescuing. Wearable glasses (e.g., Google glass and Microsoft Hololens) have enabled life logging and augmented reality. With the help of visual parsing, it has become convenient for users to record their daily activities and obtain a compact visual summary/diary of those memorable moments. This further enables users to log their life automatically. Visual parsing systems recognize and reconstruct the surrounding real world captured by the wearable camera, and then overlay artificial objects interactively to augment the reality digitally. In short, visual parsing is a key step to advance such cutting-edge technologies to operate in realistic scenarios.

Meanwhile, these advanced technologies almost universally lead to produce massive amounts of visual data. Every day, more than 300 million images are shared on Facebook, and more than 400,000 hours videos are

[1] http://www.google.com/selfdrivingcar
[2] http://www.dji.com

Self-driving Car Unmanned Aerial Vehicle Wearable Glasses

Figure 1.1: Examples of novel technologies with visual parsing behind to provide high level applications.

uploaded to YouTube[3]. Such data captures human activities, intention, and knowledge. It is hence critical to provide automatic technologies to make sense of such visual data, summarize, and organize it in an effective way.

On the one hand, we humans have the ability to tell whether there is an object in the image and where it is by simply glancing at it. On the other hand, despite several successes in conventional vision parsing systems under restricted scenarios, current automatic visual parsing methods are far behind human capability. The ultimate goal of this thesis is to bridge this gap by effectively modeling visual data and systematically building efficient visual parsing algorithms, which can further advance high level applications like robotics, autonomous driving, augmented reality, and health-care.

To this end, we first design algorithms to learn from unlabeled (Collins et al., 2012; Mukherjee et al., 2012; Kim et al., 2015) and weakly labeled data (Xu et al., 2014; Xu, 2014; Xu et al., 2015b). This is critical in building practical visual parsing systems. Conventional visual parsing algorithms rely on learning from fully labeled data, which is often limited at a small size, due to the expensive cost for collecting those full annotations. How-

[3] https://www.youtube.com/yt/press/statistics.html

ever, unlabeled/weakly labeled data are either much easier to collect or already available at much larger scale. So, it is vital to advance visual parsing using those unlabeled/weakly labeled data.

Second, we bring humans into the visual parsing loop. This is essential for computers to overcome visual parsing ambiguity and better address the desired outcome The key challenge here is to minimize user effort, while still achieving satisfactory visual parsing results. We address this by modeling visual data with side knowledge (Xu et al., 2013b; Collins et al., 2014), and bringing in novel forms of human interactions (e.g., gaze) (Xu et al., 2013a, 2015a).

In the following, we present the context and motivation of this thesis.

1.1 A Short Overview of Visual Parsing

Visual parsing is one of the most fundamental problems in computer vision. The goal of visual parsing is to decompose arbitrary images and videos into their constituent visual components, and label them with a semantic meaning when possible. This can lead to parse at different level (e.g., objects, parts, landmarks), and we focus on parsing at object (e.g., person, car) level. We refer to cutting out foreground objects from background in an image, semantically labeling each pixel in an image, spatio-temporal parsing a video into objects, and temporal parsing a video into components. Visual parsing provides a first step for many high level computer vision applications, such as action recognition (Wang and Schmid, 2013), event detection (Fathi et al., 2011b), which directly utilize the output of a visual parsing module. This further enables other artificial intelligence applications including robotics, autonomous driving, human-computer interaction, medical image analysis, and visual surveillance.

This dissertation tackles visual parsing from the following perspectives: object segmentation, semantic segmentation, video segmentation, and

video summarization. Next, we will briefly describe each task one by one.

1.1.1 Object Segmentation

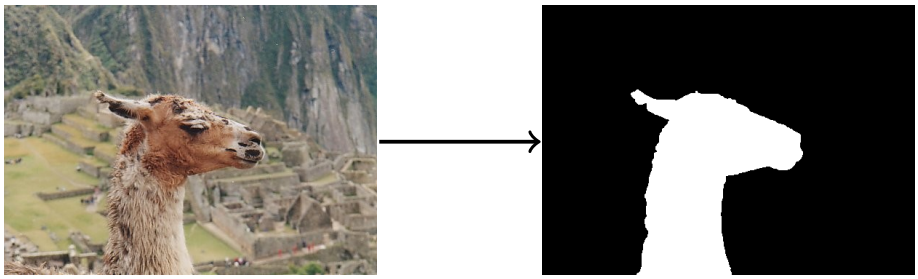

Figure 1.2: Example of object segmentation: the input image (left) is parsed into two parts (right): foreground objects and background.

This key goal here is to segment out foreground objects from background (as shown in Fig. 1.2) in a given image. The binary segmentation serves as the first step for applications like image editing (e.g., image cut-out/paste). It also enables more precise recognition, as we can focus on interesting objects while getting rid of cluttered backgrounds.

There are two typical unsupervised approaches to this problem: salient object segmentation and cosegmentation. The intuition behind the former setting is that foreground objects tend to have higher saliency than background regions (Movahedi and Elder, 2010; Borji et al., 2012; Li et al., 2014). Here, saliency can be modeled using low level image features like luminance contrast. The task of co-segmentation is to segment the shared foreground from multiple images (Rother et al., 2006; Hochbaum and Singh, 2009; Vicente et al., 2010; Mukherjee et al., 2011). This is a data driven approach based on the assumption that often are foreground objects shared across images look similar to each other, but different from the background.

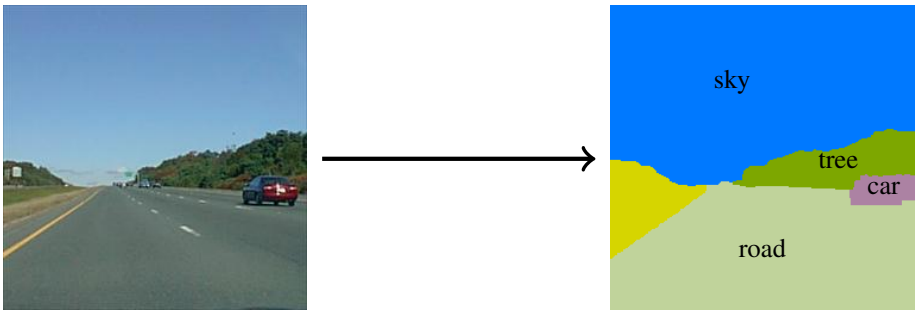

Figure 1.3: Example of semantic segmentation: the input image (left) is parsed into constituent regions (right), each of which is assigned with a semantic concept.

Another widely used setting is semi-supervised, where a user provides partial labels within an image. This is a convenient setup since users can easily put scribbles to label partial regions within an image. It is also important to eliminate ambiguity, as users can select an object of interest as foreground, while labeling cluttered objects as background. Partial labeling has been effectively utilized in interactive object segmentation with Graphcut (Boykov and Jolly, 2001a), Random Walks (Grady, 2006), Geodesic Shortest Path (Bai and Sapiro, 2009), and Geodesic Star Convexity (Gulshan et al., 2010).

1.1.2 Semantic Segmentation

Semantic segmentation extends the foreground/background labeling case to multiple semantic classes. Also known as scene parsing, it answers the question of what is where in an image (as shown in Fig. 1.3). The goal of semantic segmentation is to label each pixel with a semantic concept. This is a joint task of recognition and localization. The "things" classes (e.g., person, car) can have various poses, occlusions and shading. And the "stuff" classes (e.g., sky, grass) may have different illuminations and

no shape. These challenges have made semantically labeling an image at a pixel level particularly difficult.

A wide variety of algorithms have been developed for the fully supervised setting, where one has access to a training set where each pixel is labeled. Three types of approaches are very popular. Non-parametric methods build pixel-wise potentials using nearest neighbors (Liu et al., 2011; Eigen and Fergus, 2012; Singh and Kosecka, 2013; Tighe and Lazebnik, 2013b, 2014; Yang et al., 2014). This is based on the observation that pixels with similar semantic meaning lie close in the feature space. The second set of approaches formulate the segmentation problem as an inference task in a Markov random field (MRF) (Ladický et al., 2010; Ladicky et al., 2010; Yao et al., 2012). Supervision at different levels (tags, bounding boxes, scene types) can be easily incorporated by adding additional variables in the MRF. The final set of methods are based on object proposals (Carreira and Sminchisescu, 2010; Endres and Hoiem, 2014; Arbelaez et al., 2014; Girshick et al., 2014; Hariharan et al., 2014), where class-independent segments are generated, which are then classified into different classes by employing features defined on those segments.

1.1.3 Video Segmentation

Similar to object segmentation in images, the task in video parsing/segmentation is to extract constituent foreground objects from the background in a video (as shown in Fig. 1.4). This is a prerequisite step for a wide range of applications including video retrieval, video summarization, and action recognition (Zhang et al., 2013; Papazoglou and Ferrari, 2013; Fragkiadaki et al., 2015).

When presented with an entire sequences of frames in a video, the automatic video segmentation problem becomes very challenging. There is no given knowledge about the object appearance, scale or position, and backgrounds can also change with illumination or dynamics (e.g., waving

Figure 1.4: Example of video segmentation: a full sequence of video (top) is parsed into two layers (bottom): foreground and background.

tree shown in Fig. 1.4). Moreover, because of the sheer size of the data (e.g., 30 frames per second), a direct application of image parsing tools to tens of thousands of image frames is computationally impractical.

However, temporal relationships between image frames can make the task of video segmentation easier. First, while the background clutter may not remain constant throughout the full video sequence, the changes from one frame to the other are not abrupt. What this means is that the background can be modeled as a common subspace shared across the video frames. Second, the foreground of interest is almost never random (e.g., dispensed as salt and pepper noise) – instead, these are spatially contiguous and structured regions corresponding to people, objects, landmarks and so on. Leveraging such contiguity information as a covariate tends to make the updates of the background subspace and hence visual parsing even more accurate.

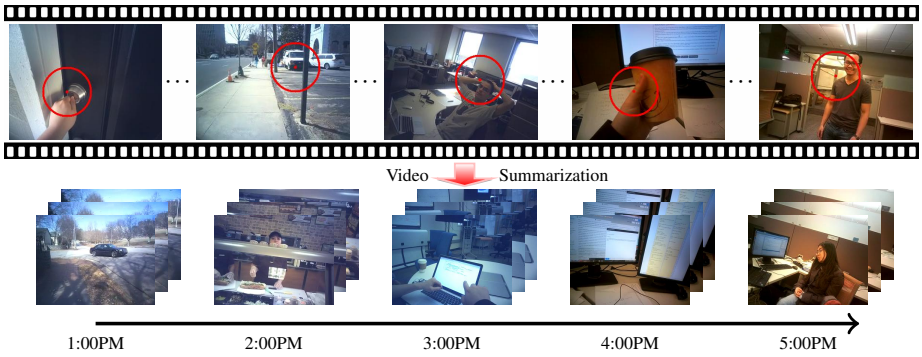

Figure 1.5: Example of video summarization: a long sequence of video (top) is parsed into a gist (bottom): a few sub-sequences from the whole video keeping the key information.

1.1.4 Video Summarization

The goal of video summarization is to extract the gist of a long video. It provides a practical functionality for users like: " I only have 2 minutes, tell me what/where to watch for this long video?" In fact, there are billions of hours videos recorded every day (e.g., surveillance videos, Youtube videos), but it is hardly possible for a human (even creators) to watch all of this content. Hence, there is an important need for mechanisms that represent the information content in a compact form (i.e., shorter videos which are more easily browsable/sharable). As illustrated in Fig. 1.5, video summarization automatically selects the most informative and interesting portion of an hours-long video, and form a visual summary by scanning through the whole volume (Wolf, 1996; Goldman et al., 2006; Khosla et al., 2013) or subshots (Lu and Grauman, 2013; Gygli et al., 2014; Gong et al., 2014). This is also one instance of visual parsing, as we select a few informative frames/sequences (as foreground) while leaving out those less relevant sequences (as background).

With the widespread availability of wearable devices (e.g., Google

glasses, GoPro cameras), we humans can now log our daily life by recording everything in a first person video. First person (also known as egocentric) video summarization is even more challenging as these videos may have poor illumination, camera shake, rapidly changing background, and a spectrum of other confounding factors. Nonetheless, given that the proliferation of wearable image-capture systems will only increase, there is a need for systems that take a long egocentric video and distill it down to its important parts. They offer the camera wearer the ability to browse/archive his/her daily activities (i.e., life logging) and review (or search) it in the future.

1.2 Why Weak Supervision?

1.2.1 Full Annotations are Expensive to Collect

Conventional fully supervised visual parsing algorithms have shown promising results. But in order to train such systems, it is typical to require full annotation of the data, which is unfortunately an onerous and expensive task. Compared with massive size of modern visual data, only a small minority of this data comes with accurate full annotations (e.g., at pixel level). Fig. 1.6 presents the modern dataset sizes with different forms of annotations. On the ImageNet dataset, 15 million images are annotated with scene categories; 1.2 million images are annotated with object bounding boxes; but only 5,000 images are annotated at a pixel level (Guillaumin et al., 2014). For video parsing, it is rare to have full frame-by-frame spatial-temporal segmentations available. The main issue here is that full annotations are much more expensive to obtain than other annotations. With such limited training data, it is extremely difficult for fully supervised algorithms to learn a reliable model.

On the other hand, image tags, bounding boxes and partial labels (user scribbles) can be easily collected or are readily available in large photo

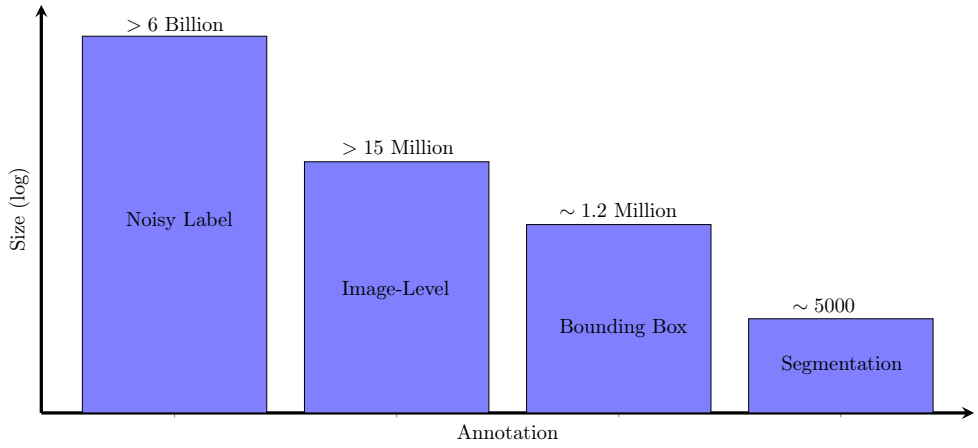

Figure 1.6: Modern dataset size with different level annotations: noisy tag, classification label, bounding box, pixel-wise segmentation.[4]

collections (e.g., tags in Flickr and Facebook). Therefore, developing visual parsing algorithms that can be learned from unlabeled/weakly labeled data is the key to push the performance of visual parsing further in the absence of massive fully annotated datasets.

1.2.2 Semantic Gap versus Human in the Loop

Despite the cost of annotating a single instance, the second question is how many annotations we need to obtain in order to train a reliable system to fully parse arbitrary images and videos? Take the object of "chair" as an example (shown in Fig. 1.7). On Ikea website, there are thousands of types of chairs with different shape, material, or color. For one single chair, there can be thousands of views taken as 2D pictures under various shading or occlusion conditions. To fill the semantic gap between human visual perception and low level visual features, millions of image annotations may be needed to learn just one visual "concept". Meanwhile, there are

[4] A earlier version appeared at http://cvn.ecp.fr/tutorials/cvpr2013/.

Input Image Instance Segmentation

Figure 1.7: Example of instance annotations in the Pascal VOC dataset (Everingham et al., 2010): the left image shows the input image, and the right image shows instance-level segmentations: one color for one instance.

more than 4 million visual articles (or concepts) on Wikipedia. Therefore, it is nearly impossible to collect these many full image annotations.

On the contrary, a human can tell whether there is an object in the image and where it is by simply glancing it immediately. Such high level visual "concept/guidance" can be easily collected. For example, we can have a human subject to look at an image, and record the gaze information. One main challenge here is how to effectively incorporate such human guidance (possibly in a noisy/weak manner) with low level visual representations.

At times, it is even necessary to have humans in the parsing loop to eliminate ambiguities. For example, when a user uploads the left image in Fig. 1.7 into a image search engine, it is not clear what exactly the user wants to search for: the table, the chairs, or the whole living room scene? Or when an user is editing the chairs in the image, does she or he want to cut out the left chair or the right one? Human intervention can make this purpose more precise.

Of course, it is hardly possible to ask humans to label every pixel with a "concept" given the massive size of modern visual data. The key question here is how to obtain good visual parsing results while minimizing human

efforts. A critical intuition which makes this possible is that visual data is not random. It presents rich structural properties (e.g., spatial contiguity, semantic context), and can be modeled using advanced mathematical/optimization tools, which then lead to effective algorithms. This has been the key theme for the research described in the following chapters.

In this dissertation, we aim to addressing such issues. In particular, we focus on addressing the following scientific questions:

- How can we utilize weakly labeled data (e.g, tags, bounding boxes, partial labels) effectively for the visual parsing task?
- When human comes into the visual parsing loop, how can we minimize user effort (e.g., user scribbles, human gaze) while still achieving satisfactory parsing results?

1.3 Structure of the Thesis

The rest of the dissertation is organized as follows:

Chapter 2 starts with the binary object segmentation where a human provides user guidance. We present an integer linear optimization model to segment out multiple objects while incorporating human interdictions as well as topological constraints.

Chapter 3 studies semantic segmentation. We formalize a latent graphical model for the problem of weakly labeled semantic segmentation, where the only source of annotation are image tags encoding which classes are present in the scene.

Chapter 4 extends the weakly labeled setting with a unified model for semantic segmentation under various forms of weak supervision (e.g., tags, bounding boxes, partial labels). We show that we can learn an effective segmentation model with such weak labels in a very efficient way.

Chapter 5 discusses a optimization model to segment foreground from background in videos on the fly, given the prior knowledge that foreground comes with a spatial contiguity. This algorithm is very expressive, and we also present applications to other binary data-layer separation problems like online tracking.

Chapter 6 establishes a submodular summarization model for first-person videos, where we capture common-sense properties of a good summary: relevance, diversity, compactness, and personalization.

Chapter 7 summarizes the contributions of this dissertation and discusses future directions.

2 OBJECT SEGMENTATION UNDER MINIMUM HUMAN INTERACTIONS

This chapter describes a new framework for interactive object segmentation(as shown in Fig. 2.1). This is an important problem in photo editing (e.g., Photoshop): the goal here to achieve a nice object cut-out from an image while minimizing user effort. We model user scribbles from a novel geometric perspective: interior indications (seeds) should lie inside object contour closures, while exterior seeds lie outside object contour closures. User scribbles are common in interactive image segmentation. What is superior of our proposed approach is that user interactions can be in the form of point. In our formulation, dense strokes are not needed to learn appearance models of foreground and background regions. As a result, we find that the segmentation results are much more robust w.r.t. seed location. We utilize concepts from discrete calculus and derive a topological constraint on how many objects are desired using the Euler characteristic. Our experiments suggest that by interpreting user indications topologically, user effort is substantially reduced. An earlier version of this chapter was published in (Xu et al., 2013a).

2.1 Problem Description and Related work

We study the problem of multiple contour completion and segmentation subject to side constraints. The types of constraints our algorithm incorporates are **(a)** those relating to inside (or outside) seed indications given via user scribbles; **(b)** global constraints on the topology, i.e., information which reflects the number of unique closed contours a user is looking for. Given the output from a boundary detector (e.g., **P**robability of **B**oundary or Pb (Maire et al., 2008)), we obtain a large set of *weighted* locally-based contours (or edgelets) as shown in Fig. 2.1. The objective then is to find k

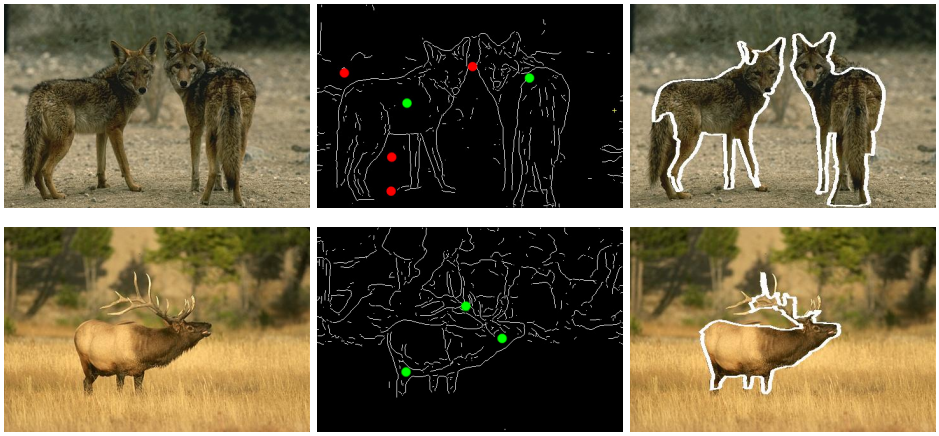

Figure 2.1: Overview of the proposed method (EulerSeg). Left to right: input images, edgelets or contours with seed indications, and final contour. Foreground is marked in green; background is marked in red; boundary is marked in white. Best viewed in color.

closed "legal" contour cycles with desirable properties (e.g., curvilinear continuity, strong edge gradient, small curvature), where legal solutions are those that satisfy the side constraints, shown in Fig. 2.1. The basic primitives in our construction are contour fragments, *not* pixels. The motivation for this choice is similar to most works on contour detection for image segmentation – by moving from predominantly region-based terms to a function that utilizes strength of edges, we seek to partly mitigate the dependence of the final segmentation on the *homogeneity* of the regions alone and the number of seeds. Additionally, in at least some circumstances, one expects benefits in terms of running time by utilizing a few hundred edges instead of a million pixels in the image. Our high level goal is the design of practical contour completion algorithms that take advice (Maclin and Shavlik, 1994, 1996; Kuhlmann et al., 2004)– which in a sense parallels a powerful suite of methods that have recently demonstrated how global knowledge can be incorporated within popular region-based image segmentation methods (Maji et al., 2011).

Related Work. The study of methods for detection of salient edges and object boundaries from images has a long history in computer vision (Ullman and Shaashua, 1988). The associated body of literature is vast – methods range from performing edge detection at the level of local patches (Shotton et al., 2005), to taking the continuity of edge contours into account (Ullman and Shaashua, 1988; Parent and Zucker, 1989), to incorporating high-level cues (Tu et al., 2005) such as those derived from shape and/or appearance (Maire et al., 2008). While the appropriateness of a specific contour detector is governed by the downstream application, developments in recent years have given a number of powerful methods that yield high quality boundary detection on a large variety of images and perform well on established benchmarks (Maire et al., 2008). Fig. 2.2 shows sample results from the globalPb (global probability of boundary) contour detector (Arbelaez et al., 2011).Broadly, this class of methods uses local measurements to estimate the likelihood of a boundary at a pixel location. To do this, the conventional approach was to identify discontinuities in the brightness channel, where as newer methods exploit significantly more information. For instance, (Martin et al., 2004) suggests a logistic regression on brightness, color, and texture, and (Dollar et al., 2006; Mairal et al., 2008) learns a classifier by operating on a large number of features derived from image patches or filter responses at multiple orientations. Contemporary to this line of research, there are also a variety of existing algorithms that integrate (or group) local edge information into a globally salient contour. Since one expects the global contour to be smooth, the well known Snakes formulation introduced an objective function based on first and second derivative of the curve. Others have proposed utilizing the ratio of two line integrals (Jermyn and Ishikawa, 2001), incorporating curvature (Schoenemann and Cremers, 2007; El-Zehiry and Grady, 2010), joining pre-extracted line segments (Wang et al., 2005; Stahl and Wang, 2008), and using CRFs to ensure the continuity of contours (Ren et al.,

Figure 2.2: Examples of boundary detection using the globalPb contour detector (Arbelaez et al., 2011).

2005). Note that despite similarities, contour detection on its own is not the same as image segmentation. In fact, even when formalized under *contour completion*, an algorithm may not always produce a closed contour. Nonetheless, from most "edge-based" methods one can obtain a partition of the image into object and background regions. Without getting into the merits of edges versus regions, one can view edge-based contours as a viable alternative to "region-based" image segmentation methods in many applications.

The success of the above developments notwithstanding, the applicability of these methods has been somewhat limited by their inability to successfully discriminate between contours of different classes of objects. To address this limitation, there has been a noticeable shift recently towards the incorporation of additional information within the contour completion process. In particular, several groups have presented frame-

GraphCut GrabCut Intelligent Scissors LabelMe

Figure 2.3: State-of-the-art of interactive segmentation methods: Graphcut (Boykov and Jolly, 2001a) and Grabcut (Rother et al., 2004) are region based; (Mortensen and Barrett, 1998) and LabelMe (Russell et al., 2008) are edge based.

works that leverage category specific (or semantic) information into the process of obtaining closed object boundaries. Specific examples of this line of work include semantic contours (Hariharan et al., 2011), the hierarchical ultrametric contour map (Arbelaez et al., 2009), and particle filtering based object detection via edges (Lu et al., 2009). The basic idea here is to achieve a balance between bottom up edge/boundary detection and top-down supervision, for simultaneous image segmentation and recognition.

While semantic knowledge based contour completion is quite powerful, its performance invariably depends on the richness of the underlying training corpus. Indeed, if the shape epitomes do not reflect the object of interest accurately enough (significant pose variations), if there is clutter/occlusion, or when a novel class is not well represented in the training data, the results may be unsatisfactory. In these circumstances, it seems natural to endow the contour completion models with the capability to leverage some form of user supervision (foreground and background seeds) (Gulshan et al., 2010). Further, knowledge provided in the form of the *number* of closed contours a user requires, can be a powerful form of user guidance as well.

Fig. 2.3 shows the-state-of-art methods for interactive segmentation.

Graphcut (Boykov and Jolly, 2001a) and Grabcut (Rother et al., 2004) are two successful region-based methods. They first learn the foreground/background appearance model from user scribbles (e.g., strokes or bounding box), and then employ max-flow to find the optimal cut-out of foreground from background. Intelligent Scissors (Mortensen and Barrett, 1998) and LabelMe (Russell et al., 2008) are two popular edge-based methods to find object boundaries. Intelligent Scissors ask user to place points a set of anchors or control points, and then produce a continuous curve (based on shortest path) passing through these control nodes. LabelMe directly asks user to draw polygon to approximate object boundaries, which has been widely used to collect segmentation datasets (Everingham et al., 2010; Liu et al., 2011; Lin et al., 2014).

While these mechanisms incorporate user interactions in region-based or edge-based segmentation separately, only a few methods take such information explicitly into account for edge-based contour completion. In this work, we leverage a discrete calculus based toolset to incorporate such topological and seed indications type supervision within a practical contour completion algorithm.

The primary contributions of this chapter are: **(i)** We present a unified optimization model for multiple contour completion/segmentation which incorporates topological constraints as well as inclusion/exclusion of foreground and background seeds. The topological knowledge is included by using the *Euler characteristic* of the edgelet graph where as inclusion/exclusion constraints utilize concepts from discrete calculus. **(ii)** On an extensive dataset, we provide strong evidence that with a small amount of user interaction, one can obtain high quality segmentations based on edge contours information alone. We give an easy to use implementation, as well as user scribble data corresponding to varying levels of interaction on this large (~ 1000) set of images.

2.2 Discrete Calculus

The tools of discrete calculus provide a powerful formalism to represent the topological information in an image (Grady and Polimeni, 2010; Kovalevsky, 1989; Chauve et al., 2010). We use conventions of discrete calculus to describe our problem of finding multiple contour closures. In this section, we introduce the idea of *cell complices* which are the fundamental building blocks of our construction. The following text also introduces the necessary notations, which will be used thoughout the rest of the text.

The domain of an image is decomposed into a set of *cells*. If the decomposition is such that (*i*) the interiors of the cells are disjoint and (*ii*) the boundary between any two p-dimensional cells is a $(p-1)$-dimensional cell, then we have a *cell complex*. As an example, consider a planar graph $G = \langle V, E, F \rangle$ with vertices V, edges E, and faces F. Such a graph has *incidence* relationship between each face and its bounding edges, and between each edge and its endpoint vertices. Similarly, each vertex is incident on two or more edges and each edge is incident on two faces. Notice that the interior of a pair of faces is disjoint, and the boundary between any two faces gives an edge, where the dimension is reduced by one. As a consequence, we get a 2D cell complex for a planar graph, and also a set of incidence relationships among simplices of different dimensions.

A cell complex may be *oriented* such that we can describe directions on each cell relative to its orientation, see Fig. 2.4(a). Each type of cell has a corresponding pair of possible orientations: a vertex (0-cell) is either a source or a sink while an edge (1-cell) may be directed toward either endpoint. Further, each cell *induces* a corresponding orientation on incident cells; for example, a directed edge has a source endpoint vertex at one end and sink at the other. The orientations of a cell and a member of its boundary are *coherent* if the induced orientations agree, an example is shown in Fig. 2.4(b).

We may represent the two-dimensional image as an oriented complex.

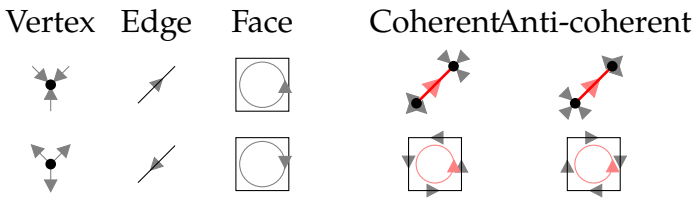

Figure 2.4: Visualization of the orientations on cells of different dimensionalities (a). In (b) we show in the left column p-cells with all of their boundary $(p-1)$-cells coherently oriented, and all boundary cells anti-coherently oriented in the right column.

All faces are given the same orientation, while edges and vertices are given arbitrary orientations. After enumerating its constituent vertices, edges and faces, a selection of some subset of faces is specified with an indicator vector $\mathbf{x} \in \{0,1\}^{|F|}$. $x_i = 1$ denotes the candidate face $F_i \in F$ is in the foreground, and $x_i = 0$ otherwise. Similarly, we represent the edge and vertex configuration of G by indicator vectors $\mathbf{y} \in \{0,1\}^{|E|}$ and $\mathbf{z} \in \{0,1\}^{|V|}$ respectively. We require that the indicator vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$ on each level of cell consistently describe a segmentation. We will overload the operator $|\cdot|$: when applied to a set (e.g., E, F, V), it will mean its cardinality; when applied to a vector (e.g., $\mathbf{x}, \mathbf{y}, \mathbf{z}$) or a matrix (e.g., A, C), it will denote to its absolute value. The key relationship is consistency between the labels on the *incident* cells. These relationships can be expressed algebraically using the notion of a dimension-appropriate *incidence matrix*. The edge-face incidence matrix (also called the boundary operator) $C_1 \in \{-1,0,1\}^{|E|\times|F|}$ is defined by

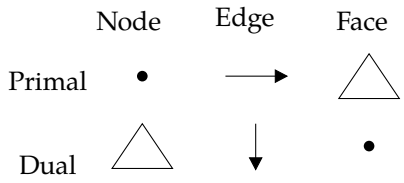

Figure 2.5: Duality relationships between 2D cell complices.

$$C_{1;ij} = \begin{cases} 1 & \text{if edge } i \text{ is incident to face } j \text{ and coherently oriented;} \\ -1 & \text{if edge } i \text{ is incident to face } j \text{ and anti-coherently oriented;} \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

Here, $C_{1;ij}$ refers to entry (i, j) in C_1. Similarly, by discarding orientation information, we can define the edge-face *corresponding matrix* $C_2 \in \{0,1\}^{|E|\times|F|}$ which labels which edges are incident to which face. It can be calculated as the element-wise absolute value of C_1, such that $C_{2;ij} = |C_{1;ij}|$.

The node-edge incident matrix $A_1 \in \{-1, 0, 1\}^{|V|\times|E|}$ is defined analogously to (2.1), where $A_{1;ij} = 1$ iff node i is incident to edge j. As with C_2, we define the node-edge corresponding matrix $A_2 = |A_1| \in \{0,1\}^{|V|\times|E|}$. We further use a node-edge *degree matrix* $A_3 \in \mathbb{R}^{|V|\times|E|}$, where $A_{3;ij} = A_{2;ij}/d_i$ where d_i denotes the degree of node i.

Discrete calculus describes the notion of *duality* between cell complices. In a p-complex, each q-cell will have a corresponding dual $(p-q)$-cell (say, $q \leqslant p$). For any given cell complex, we can construct its dual in a way that preserves incidence relationships between cells, see Fig. 2.5. Using these concepts, in the following sections, we will formalize the required constraints within a contour completion objective function.

2.3 Problem Formulation with Integer Linear Programing

As described in Section 2.2, our model works with selections of the cells constituting the foreground. Since the notion of foreground for a face is self-evident, we will describe the labeling of vertices and edges, starting from a face labeling $\mathbf{x}$. We enforce the following condition:

Condition 1. *A* p*-cell is in the foreground if and only if it is incident to a* $(p+1)$*-cell in the foreground.*

This condition ensures that each connected component of the foreground is itself a cell complex, a property we will use shortly.

First, we introduce an auxiliary indicator variable $\mathbf{w} \in \{0,1\}^{|E|}$ which selects the *boundary edges*. These edges are those which are incident to both a foreground and a background face. W.l.o.g., consider edge 1 incident to faces 1 and 2 respectively, then $w_1 = |x_1 - x_2| = \mathbb{I}(x_1 \neq x_2)$, where $\mathbb{I}(\cdot)$ is an indication function returning false or true (i.e., $0/1$). Taken together, the full set of boundary edges precisely represent the contour of the selected foreground. We can now use the *boundary operator* from Section 2.2 to derive the identity

$$\mathbf{w} = |C_1\mathbf{x}| \tag{2.2}$$

Observe that each edge is incident to exactly two faces, and we ask that all faces have identical orientation. It follows that an edge must be coherent with one face and anti-coherent with the other. Therefore, for all *internal edges* (non-boundary edges in the foreground) the C_1 operator when multiplied with $\mathbf{x}$, cancels the contribution from these two faces, leaving non-zero values only for the boundary edges.

To see how it works, Fig. 2.6 illustrates a toy example. We have three faces and seven edges, and then we can construct the edge-face incidence

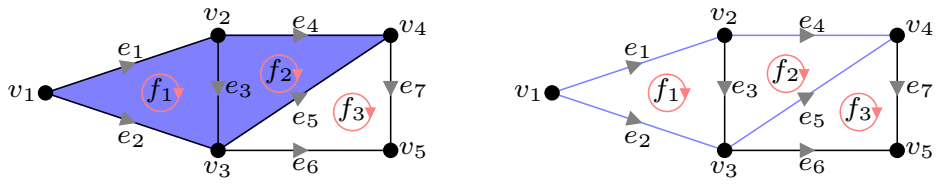

Figure 2.6: Example of boundary operator C_1 using discrete calculus.

matrix (also known as boundary operator) C_1. When selecting faces f_1, f_2, we obtain the boundary edges e_1, e_2, e_4, e_5 by $\mathbf{b} = C_1\mathbf{x}$,

$$C_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \qquad \mathbf{b} = C_1\mathbf{x} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

The internal edges (which are incident to foreground faces on both sides) can still be computed, albeit differently. The vector $C_2\mathbf{x}$ will count the inside edges twice and the boundary edges once, as we discard orientation (and thus sign information). In the preceding, w.l.o.g. $(C_2\mathbf{x})_1 = x_1 + x_2$. Thus, Condition 1 will be satisfied if the following identity holds:

$$2\mathbf{y} = \mathbf{w} + C_2\mathbf{x} \tag{2.3}$$

We use the matrices A_2, A_3 for a pair of linear inequalities which are equivalent to Condition 1 *for vertices*. Observe that the vector $A_2\mathbf{y}$ will be the number of foreground edges incident to each foreground vertex

(or node), where $(A_2\mathbf{y})_i$ is the number of foreground edges incident to vertex (or node) i. Similarily, when scaled by the degree d_i of vertex i, $(A_3\mathbf{y})_i \in [0,1]$ will be the *proportion* of edges incident to i which are in foreground. Enforcing condition 1 is equivalent to:

$$A_3\mathbf{y} \leqslant \mathbf{z} \leqslant A_2\mathbf{y} \tag{2.4}$$

Since $\mathbf{z}_i \in \{0,1\}$, the condition, $\mathbf{z}_i \geqslant (A_3\mathbf{y})_i$, will be true only for $\mathbf{z}_i = 1$ if any edge incident to i is in foreground. Conversely, if no edge incident to i is selected in the solution, then $(A_2\mathbf{y})_i = (A_3\mathbf{y})_i = 0$ and (2.4) is satisfied only for $\mathbf{z}_i = 0$.

The expressions introduced above allow the identification of whether a user provided seed falls "inside" or "outside" the contour completion given by $\mathbf{w}$, and will serve as constraints for our multiple contour completion model. Fig. 2.8 shows an illustrative example for an image, where the input to the contour completion are edgelets (or edgels) obtained from boundaries of a globalPb derived super-pixels (Levinshtein et al., 2009). We show a few examples in Fig. 2.7 for illustration.

Euler Characteristic. Our final requirement is to be able to specify the *number* of closed contours desired. The existing literature on region based image segmentation provides some ideas on how this can be accomplished for random field based models – in the form of so-called connectedness constraints. TopologyCuts is an extension of graphcuts and utilizes certain levelset ideas to preserve topology (Zeng et al., 2008). The DijkstraGC (Vicente et al., 2008) finds a segmentation where two manually indicated seed points are connected via the foreground where as Nowozin (Nowozin and Lampert, 2010) makes use of a LP relaxation. Very recently, (Chen et al., 2011a) proposed selectively perturbing the energy function to ensure topological properties. Here, we show how a much simpler form can capture the desired topological properties, as described next.

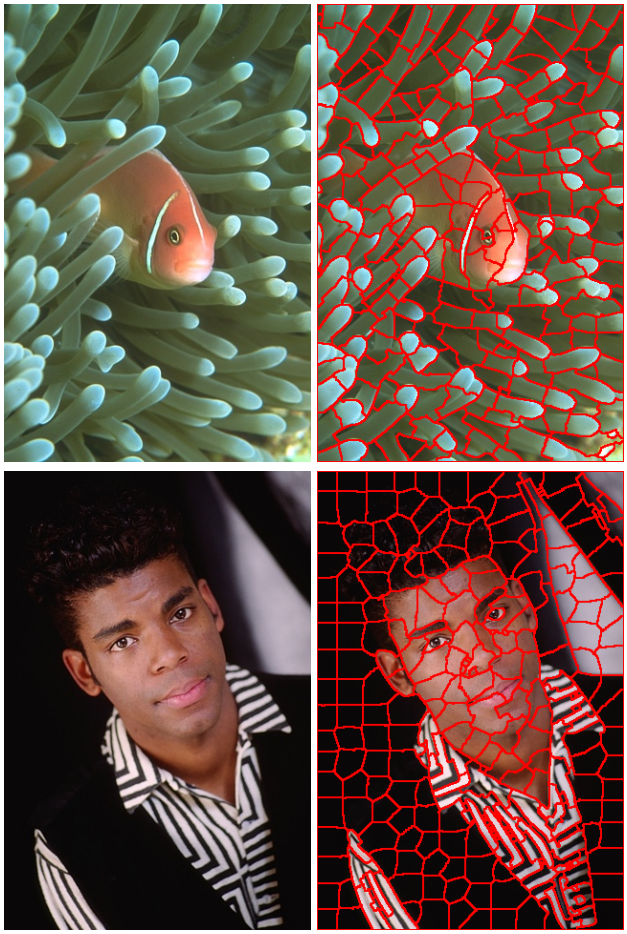

Figure 2.7: Examples of super-pixel segmentation using the TurboPixels method(Levinshtein et al., 2009).

For any graph we can define the *Euler characteristic* as

$$\chi = |V| - |E| + |F|, \tag{2.5}$$

where $\chi = 2$ for any planar embedding of a graph. If we explicitly constrain that the Euler characteristic of an induced subgraph created by selecting any given foreground is exactly two, this will give a foreground region that is connected and simple in a geometric sense. For multiple

connected regions, we can use the generalized form of this formula for arbitrary planar graphs:

$$|F| + |V| - |E| = n + 1 \tag{2.6}$$

where n is the number of connected components.)

Lemma 2.1. *Let* $\mathbf{x}, \mathbf{y}, \mathbf{z}$ *denote indicator vectors for the selection of faces, edges, and vertices for planar graph* G. *The selected subgraph will satisfy Eq. 2.6 if*

$$\sum_i x_i + \sum_k z_k - \sum_j y_j = n \tag{2.7}$$

Proof. The left-hand side of this formula counts each relevant quantity for the Euler characteristic of the selected subgraph, but it neglects to count the "outside" face. Subtract one from the RHS and derive the equality. □

This will not count the extra outside faces corresponding to any "holes". This was not a problem in our experiments, but can be explicitly avoided by requiring the background be connected using the spanning tree constraints of (Singh and Lau, 2007). Using (2.7) as a constraint in our model will guarantee that we recover n *simply connected* foregrounds.

2.3.1 Optimization Model

Before we introduce the contour completion model, we briefly describe the procedure for deriving the components of the graph from an image. This process follows existing algorithms for contour and boundary detection. First, we run the globalPb detector (Arbelaez et al., 2011) on an image which provides the probability of boundary for each image pixel. Next, we generate a set of superpixels (as shown in Fig. 2.7) from the image using the globalPb output in conjunction with TurboPixels (Levinshtein et al., 2009), which uses local information and compactness. Each superpixel

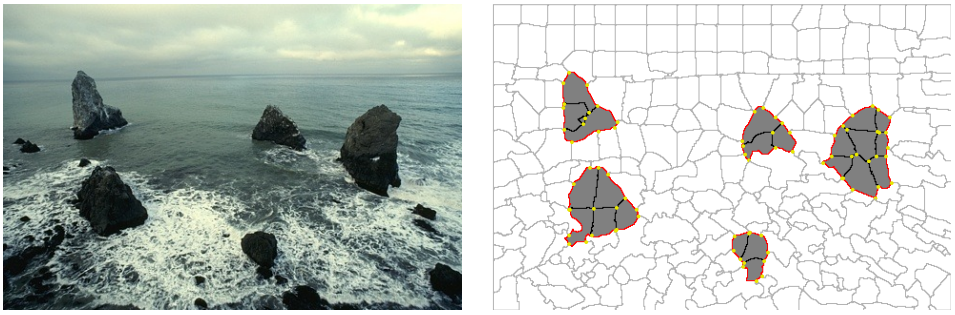

Figure 2.8: Graph representation based on superpixel segmentation with the foreground subgraph consistent under condition 1. Selected faces are shaded, foreground edges are bold and foreground vertices highlighted in yellow. Internal edges $\mathbf{y}_i \neq \mathbf{w}_i = 0$ are bold/black, boundary edges $\mathbf{y}_i = \mathbf{w}_i = 1$ are red.

corresponds to a face, and the boundary of the superpixel corresponds to edges in the graph (these are the basic primitives of the closed contours we will derive). If two edges are connected, we introduce a node in the graph. With this construction, the problem of finding multiple contour closures reduces to finding multiple cycles in the graph.

To select the cycles for the strongest contours, we want to weight the edges appropriately. For this purpose, we calculate two types of weight measures following (Levinshtein et al., 2010). The first, denoted by $\mathbf{N}$, measures the "goodness" of edges. The better edge i is, the smaller N_i will be. The second, denoted by $\mathbf{D}$, is the count of all the pixels on the edge. We use an objective function which is the ratio of these quantities, $\frac{\mathbf{N}^\top \mathbf{w}}{\mathbf{D}^\top \mathbf{w}}$. This ends up being the portion of contour w.r.t arc-length which does not lie on a true image edge. Fig. 2.9 compares the three scores with different solutions. As we can see, the ratio quantity provides a contour that has strong edge support in the image.

Finally, the user indictations are represented in terms of indicator vectors $\mathbf{x}_0, \mathbf{x}_1$, where $\mathbf{x}_{0;i} = 1$ if face i contains a background seed. With the basic components (or constraints) in hand, we now have the main

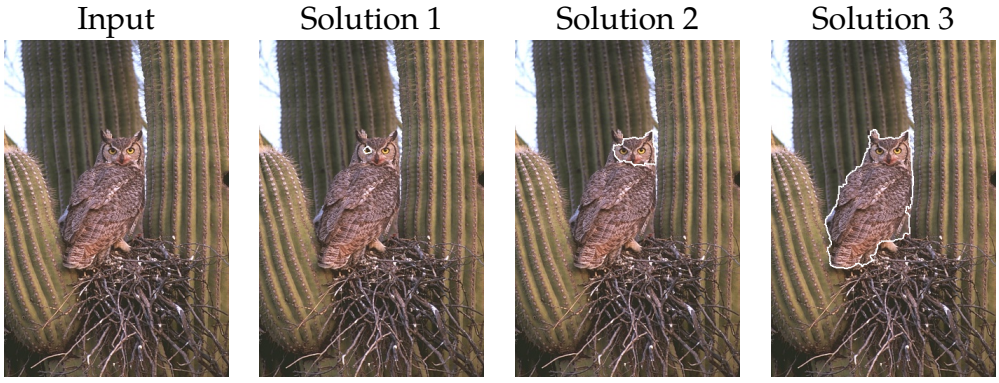

Figure 2.9: Comparison of different objective functions: numerator $\mathbf{N}^T\mathbf{w}$, denominator $\mathbf{D}^T\mathbf{w}$, and ratio score $\frac{\mathbf{N}^T\mathbf{w}}{\mathbf{D}^T\mathbf{w}}$ w.r.t. different solutions.

optimization model.

$$\begin{aligned} \min_{\mathbf{w},\mathbf{x},\mathbf{y},\mathbf{z}} \quad & \frac{\mathbf{N}^T\mathbf{w}}{\mathbf{D}^T\mathbf{w}}, \\ \text{s.t.} \quad & \mathbf{w} = |C_1\mathbf{x}|, \quad 2\mathbf{y} = \mathbf{w} + C_2\mathbf{x}, \\ & A_3\mathbf{y} \leqslant \mathbf{z} \leqslant A_2\mathbf{y}, \quad 1^T\mathbf{x} + 1^T\mathbf{z} - 1^T\mathbf{y} = n, \\ & \mathbf{x}_1 \leqslant \mathbf{x} \leqslant 1 - \mathbf{x}_0, \quad \mathbf{w},\mathbf{x},\mathbf{y},\mathbf{z} \in \{0,1\} \end{aligned} \tag{2.8}$$

2.3.2 Optimizing Ratio Objective

Since the objective in (2.8) of the main model is in ratio form, we transform it into a linear function with a free variable, t. Our linear ratio cost objective function is solved by minimizing $f(t,\mathbf{u}) = (\mathbf{N} - t\mathbf{D})^T\mathbf{u}$, over admissible $\mathbf{u}$ for a sequence of chosen values of t. Here, $\mathbf{u}$ denotes the concatenated vector of all indicator variables in the model. Assume $\mathbf{D} \geqslant 0$ and $\mathbf{D}^T\mathbf{u} \neq 0$. For an initial finite bounding interval $[t_l, t_u]$, let t_0 be the initial value. Let $\bar{\mathbf{u}} = \arg\min_{\mathbf{u}} f(t_0,\mathbf{u})$, the procedure proceeds as follows:

- $f(t_0, \bar{\mathbf{u}}) = 0$: $\mathbf{N}^T\bar{\mathbf{u}}/\mathbf{D}^T\bar{\mathbf{u}} = t_0$, stop with solution t_0
- $f(t_0, \bar{\mathbf{u}}) < 0$: $\mathbf{N}^T\bar{\mathbf{u}}/\mathbf{D}^T\bar{\mathbf{u}} < t_0$, $t_u \leftarrow \mathbf{N}^T\bar{\mathbf{u}}/\mathbf{D}^T\bar{\mathbf{u}}$
- $f(t_0, \bar{\mathbf{u}}) > 0$: $\mathbf{N}^T\bar{\mathbf{u}}/\mathbf{D}^T\bar{\mathbf{u}} > t_0$, $t_l \leftarrow t_0$

Each iteration is easily solved in a few seconds using the CPLEX IP solver on a standard workstation.

2.3.3 Spanning Tree Constraint

In Section 2.3, we stated that the $\sum_i x_i$ term in Eq. 2.7 will not count the "outside" faces. In addition, it does not count faces introduced due to holes in the selected foreground. If we denote the number of such holes by H, then the actual number of faces of the foreground subgraph is $\sum_i x_i + H + 1$. Modifying the expression to match Eq. 2.6, this takes the form

$$\begin{aligned} \left(\sum_i x_i + H + 1\right) + \sum_k z_k - \sum_j y_j &= C + 1 \\ \sum_i x_i + \sum_k z_k - \sum_j y_j &= C - H \end{aligned} \tag{2.9}$$

where C is the number of connected components. There remains a small ambiguity in the constraint, such that introducing a new connected component along with a new hole will maintain the equality. This can be easily eliminated either via a choice of objective which favors minimum arc length (and thus will avoid holes), or by *explicitly* restricting $H = 0$.

If the background faces form one connected component, there are no holes inside the selected foreground cycles. This is achievable with the constraints of (Singh and Lau, 2007) to require the existence of a spanning tree on the dual graph of unselected faces.

Denote $\bar{\mathbf{x}} = 1 - \mathbf{x}$ as the faces we did not select in our solution. We introduce auxiliary variables for a face-simultaneous-selection matrix S,

and variable T indicating which dual edges are in the spanning tree.

$$S_{ij} = \begin{cases} 1 & \text{if } \bar{x}_i = \bar{x}_j = 1, \text{ faces } i \text{ and } j \text{ adjacent;} \\ 0 & \text{otherwise.} \end{cases} \tag{2.10}$$

The spanning tree is constructed from S using the following constraints.

First, if $S_{ij} = 0$, this cannot be an edge in the tree.

$$T_{ij} \leqslant S_{ij} \quad \forall\, i, j \tag{2.11}$$

Second, if $\bar{x}_i = 1$, there must be at least one edge incident on face i

$$\sum_{j \sim i} T_{ij} \geqslant \bar{x}_i \quad \forall i \tag{2.12}$$

All the background faces should form a tree. A graph is a tree only if it has one fewer edge than faces

$$\sum_{i \sim j} T_{ij} = \sum_{i} \bar{x}_i - 1 \tag{2.13}$$

Finally, one eliminates cycles by ensuring all *subsets* of faces are no more connected than a tree

$$\sum_{i \sim j; i,j \in S} T_{ij} \leqslant \sum_{i \in S} \bar{x}_i - 1 \quad \forall\, S \subset F \tag{2.14}$$

this is enforced for *all* subsets S of faces. If a feasible T exists, then the background must be connected and there are no holes in the foreground. We do not use (2.14) in our solver, and present it for completeness. Instead, we rely in practice on the tendency of our model's *objective* to prefer short, simple boundaries which do not introduce unnecessary holes.

Relation to genus: The idea of counting holes H introduced in this

section is highly related to the notion of non-orientable genus in topology (Massey, 1991). For a non-orientable surface, the relationship between the genus k and Euler characteristic χ is

$$\chi = 2 - g \tag{2.15}$$

This suggests that some other ideas from topology may be helpful towards imposing richer priors on the desired segmentation.

2.4 Beyond Superpixel-derived Edgelets

Recall that the model in Section 2.3.1 constructs a cell complex using a superpixel decomposition of the image domain. While fast algorithms for finding this decomposition are available (Levinshtein et al., 2009), it is known that superpixels are not robust for all types of images. Occlusion or weak boundaries give cases where the set of superpixel boundary primitives (the input to our optimization) do *not* include some valid edgelets (ones which have not been picked up by either the contour detector or superpixel method). The natural solution to this is to *supplement* the basic set of edgelet primitives with additional contour pieces that bridge the 'gaps' and allow a more accurate contour closure even in the presence of very weak signal variations. Next, we present such an extension to find completions using a base set of disconnected edgelets. But introducing completions between all pairs of edgelets is prohibitive and leads to a problem with a large number of variables (especially for multiple contours). The following model, while applicable to the multiple contour setting, is most effective for finding *a single contour* which encloses a simply connected foreground region.

Euler Spiral. A key subcomponent of this problem is how to join two edgelets which will follow each other on the contour. This is the problem solved by (Kimia et al., 2003) which proposes to use segments of

the *Euler spiral*. This spiral can be shown to be the curve $\mathcal{C}$ with minimal *total curvature*,

$$\mathrm{TC}_2 = \int_{\mathcal{C}} \kappa(s)^2 \, ds$$

where $\kappa(s)$ is the curvature at a given point on the curve parameterized by arc-length. For any pair of points along with tangents we can construct a segment of an euler spiral which connects these points with consistent tangents. (Kimia et al., 2003) show that these completions satisfy the conditions given by (Horn, 1983) for a "pleasing" curve (invariance to similarity transformations, symmetry, extensibility, smoothness, roundness).

We parameterize the spiral by the turning angle as in (Walton and Meek, 2009). To form a completion, we consider the Euler spiral under a similarity transformation determined by the position and Frenet frame $(\mathbf{P}_0, \mathbf{T}_0, \mathbf{N}_0)$ at the spiral's *inflection point*, and a scaling factor a. The transformed spiral is

$$Q(\theta) = \begin{cases} \mathbf{P}_0 + aC(\theta)\mathbf{T}_0 + aS(\theta)\mathbf{N}_0 & \theta \geqslant 0 \\ \mathbf{P}_0 - aC(-\theta)\mathbf{T}0 - aS(-\theta)\mathbf{N}_0 & \theta < 0 \end{cases}$$

where S and C are the *Fresnel integrals*. A choice of interval $[\theta_1, \theta_2]$ selects a given segment. (Walton and Meek, 2009) gives a set of equations to determine these free variables, given segment endpoints $\mathbf{P}_1, \mathbf{P}_2$ and their tangents $\mathbf{T}_1, \mathbf{T}_2$. We solve these equations using a modified Newton's method. The most expensive step, the computation of the Fresnel integrals, is sped up considerably using (Fleckner, 1968), but augmented with precomputed tables. We can compute an average completion in 30μs, versus 1ms for (Kimia et al., 2003) on the same machine, making it an attractive option to calculate a large number of completions, quickly, within the core contour completion engine.

Euler Spiral for One Contour Completion. We are given a set of

image edgelets derived from an edge detector as before, as well as user-provided foreground and background seeds. The core objective considered by the algorithm is an *alternating path* p which consists of a sequence of edgelets joined by Euler Spiral segments. The goal is to find a closed contour that minimizes an objective function that increases with the addition of each contour segment.

Our solution strategy is to iteratively build upon the current *partial path*, until we get a cycle that encloses a feasible region. To do this, we adopt a specialized branch and bound procedure. Here, each node v of the branch-and-bound tree corresponds to some alternating path p. If p is a cycle, then v is a leaf node and thus a candidate solution. In this case, we check p is checked for feasiblity w.r.t. the seed constraints. If p is *not* a cycle, we may construct the children of this node by considering each image edglet in sequence and calculating the euler completion, on the fly. The path for the a child is then p plus the current completion and edgelet appended to the end. Children are discarded if they give rise to a self-intersecting partial path; therefore, entire subtrees can be discarded directly. Any partial path with objective worse than the best candidate solution found so far may be ignored. Otherwise, we descend the tree to each child in turn, ordered by the cost of their partial contour.

This algorithm implicitly solves a model of the form in (2.8), with a linear objective function on $\mathbf{w}$ and smoothness constraints on the solution contour. We can construct a planar graph for this model using the *extensibility* property of Euler spirals and splitting any two intersecting segments.

2.4.1 Branch-and-Bound Method

We give details on the solver which solves our model without explicitly constructing the full cell complex.

Construction. The solver is given a set of image *edgels* E and seeds. The

branch-and-bound algorithm considers partial solutions p to a contour completion problem. p is an alternating path which consists of a sequence of edgels joined by Euler Spiral segments $\mathcal{C}$. The *children* of a branch-and-bound node are simply those contours which extend p by a single completion and edgel

$$\begin{aligned}\text{children}(p) =&\{\text{concatenate}(p, \mathcal{C}, e) \mid \\ &\mathcal{C} \text{ joins tail}(p) \text{ and } e \; \forall e \in E\}\end{aligned}$$

Cost Function. We seek a closed p which minimizes some integral cost over the contour, for instance the elastica energy

$$C(p) = \int_p \alpha \kappa_p(s)^2 + \beta \; ds \tag{2.16}$$

Note that $\beta > 0$ suggests using completions based on general elastica (Horn, 1983), though our experiments suggest using negligible $\beta \ll \alpha$. Note that any cost of this form will satisfy $C(q) \leqslant C(p)$ for any $q \in \text{children}(p)$.

Overview. In order to allow a flexible node visit order, we use a *priority queue* over partial solutions. This provides an `enqueue` operation which places an contour in the queue, and a `dequeue` operation which removes the queue element with minimum cost and removes it. If every time we `dequeue` a contour we `enqueue` its children this will iterate over all possible contours in order of increasing cost. As soon as we find a feasible closed contour this is the solution. Fig. 2.10 present sample results from our branch-bound contour completion method.

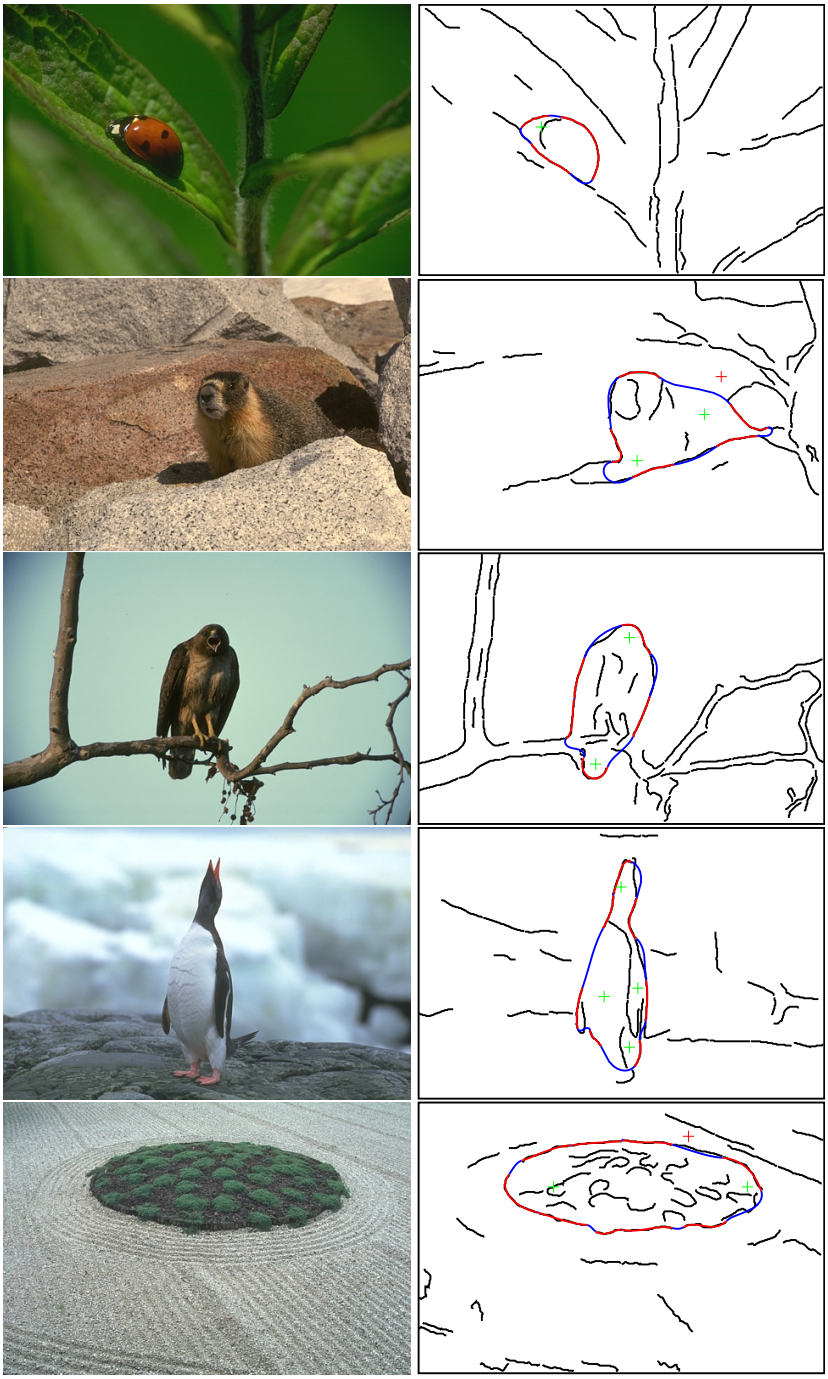

Figure 2.10: Results from the branch-and-bound algorithm for BSD images. The input edgelets are shown in black, Euler completions in blue. The red segments are input edgelets, smoothed for accurate derivative estimation.

2.5 Experiments

2.5.1 Dataset and Experiments Setting

We first provide evaluations of the model from Section 2.3 on images from the Weizmann Horse Database (WHD) (Borenstein and Ullman, 2002), the Weizmann Segmentation Database (WSD) (Alpert et al., 2007), and the Berkeley Segmentation Data Set (BSDS500) (Arbelaez et al., 2011). We then continue describing our experiments with a *robot user* on the ISEG dataset. These experiments will show that the combination of interaction with a contour-based method can achieve high levels of accuracy with a minimum of user effort.

We compare our approach (which we refer as EulerSeg) with three other contour grouping methods: (*i*) Ratio Region Cut (RRC) from (Stahl and Wang, 2007), (*ii*) Superpixel Closure (SC) from (Levinshtein et al., 2010), and an adaptive grouping method (EJ) (Estrada and Jepson, 2006). We note that these are unsupervised whereas our algorithm incorporates user interaction, but SC and EJ produce multiple segmentations of which we select the most favorable. We compute the F-measure by the region overlapping and report quantitative results in Fig. 2.14.

The cell complex is generated from superpixels via (Levinshtein et al., 2009) and the same number of superpixels as SC in all our experiment. We typically indicate $1 \sim 2$ interior seeds for the sought objects, but in the presence of $\geqslant 2$ objects, we may need $3 - 7$ points including both interior and exterior seeds. The indicated seeds are shown in the images: green marks are foreground and red marks are background.

RRC was run using the default parameters $\lambda = 0, \alpha = 1$. That method has an additional parameter to indicate an arbitrary number of objects. However, it frequently fails to get a second boundary even when the image includes 2 objects. For SC, we use their reported best parameters with the number of superpixels set to 200 and $T_e = 0.05$. That algorithm generates

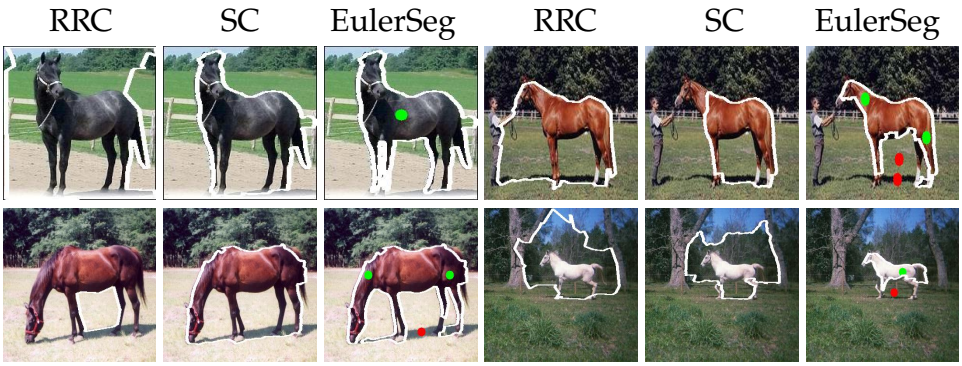

Figure 2.11: Sample results from WHD. **Best viewed in color.**

K = 10 possible solutions, here we report results for the best one.

2.5.2 Qualitative Evaluation on Contour Completion

WHD Results: WHD consists of 328 side-view images of horses, with exactly one horse in each image. Fig. 2.11 shows both RRC and SC select large regions of ground between the horses' legs due to their large-region bias. As the examples show, our objective function minimizes gaps in the closure and leverages user seeds to handle slender objects better and outperforms both with $\leqslant 5$ seeds.

WSD Results: WSD contains 200 images and is divided into 2 subsets of images with one or two foreground objects. As shown in Fig. 2.12, our algorithm is comparable to RRC and SC when there is one object with only one seed. However, when the image contains 2 objects, our Euler characteristic constraint fires in and we correctly segment both objects of interest, while RRC and SC either selects one of the objects or segments one large region which includes both.

BSDS500 Results: Compared with WSD and WHD, images in this dataset are more complicated. We note that in some images of BSDS500, there are no salient objects or closed contours (e.g., images of sky or street).

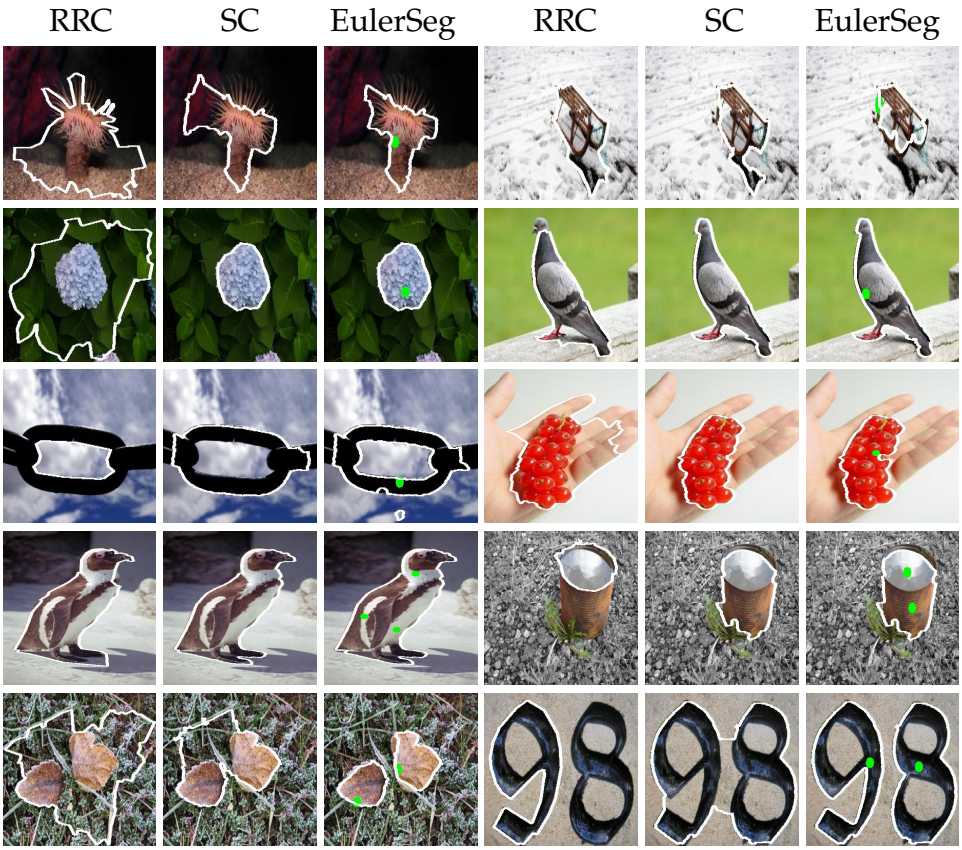

Figure 2.12: Sample results from WSD. **Best viewed in color.**

In these cases our algorithm cannot find a meaningful closed contour, but where one is present our model performs at least as well as any of the compared methods. However, another challenging class of images in BSD are those that depict a large number of foreground objects, here our algorithm significantly improves upon previous results with a small amount of user guideline and the topological constraint. An example of this can be seen in the bottom row of Fig. 2.13, where RRC and SC fail whereas our method is able to find the correct solution easily.

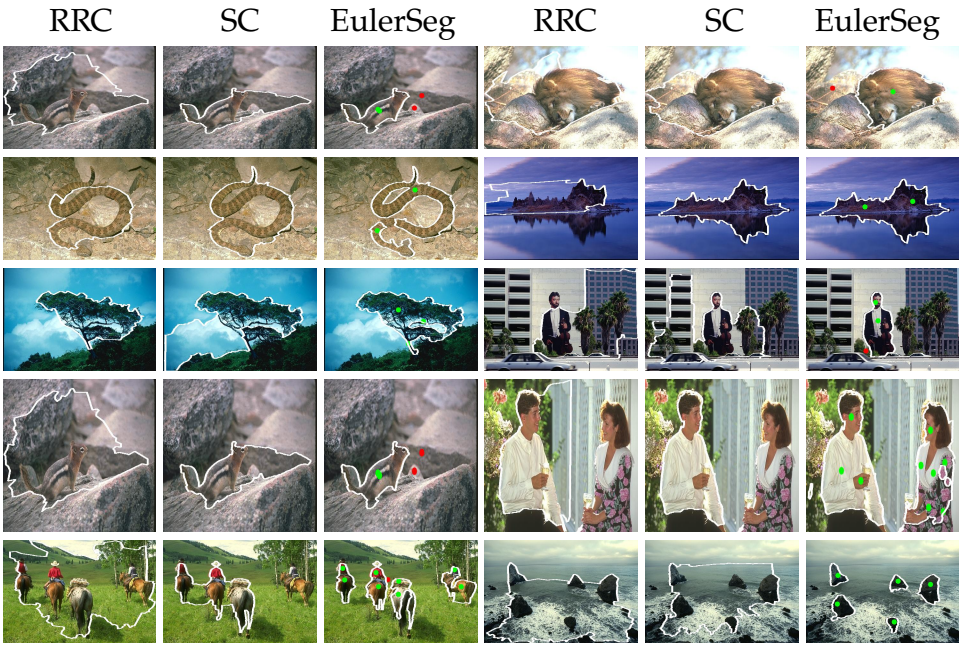

Figure 2.13: Sample results from BSDS500. **Best viewed in color.**

2.5.3 Quantitative Evaluation on Contour Completion

For a region A from an algorithm and a region B from the ground truth, we define the precision as the ratio of true points on A:

$$P = \frac{|\text{Matched}(A, B)|}{|A|} \tag{2.17}$$

and recall as the proportion of detected points on B:

$$R = \frac{|\text{Matched}(B, A)|}{|B|} \tag{2.18}$$

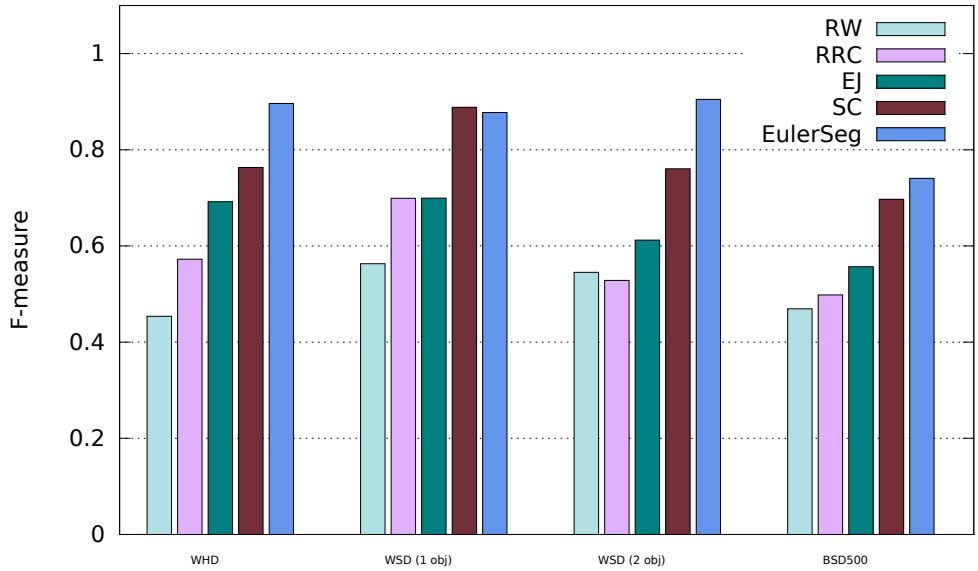

Figure 2.14: F-measure scores on datasets described in Section 2.5.

where $|\text{Matched}(A, B)|$ is the intersected pixels of the segmented region and ground truth. We define our F-Measure as

$$F = \frac{2PR}{P + R} \tag{2.19}$$

The average performance of the four algorithms (RRC, EJ, SC, and ours) is shown in Fig. 2.14. In the BSD500 truth, as the images are parsed into a few number of regions ($\geqslant 5$), we use our seed points to extract a binary ground truth, with any regions marked with a foreground seed placed in the foreground. We also compare here to a basic supervised method from the region/graph-based setting, Random Walker (RW) (Grady, 2006) using the same seeds. Fig. 2.14 shows EulerSeg (our algorithm) performs comparably with SC on the WSD with one object while on the WHD, WSD with 2 objects and BSD500, our algorithm performs significantly better than the four baseline algorithms.

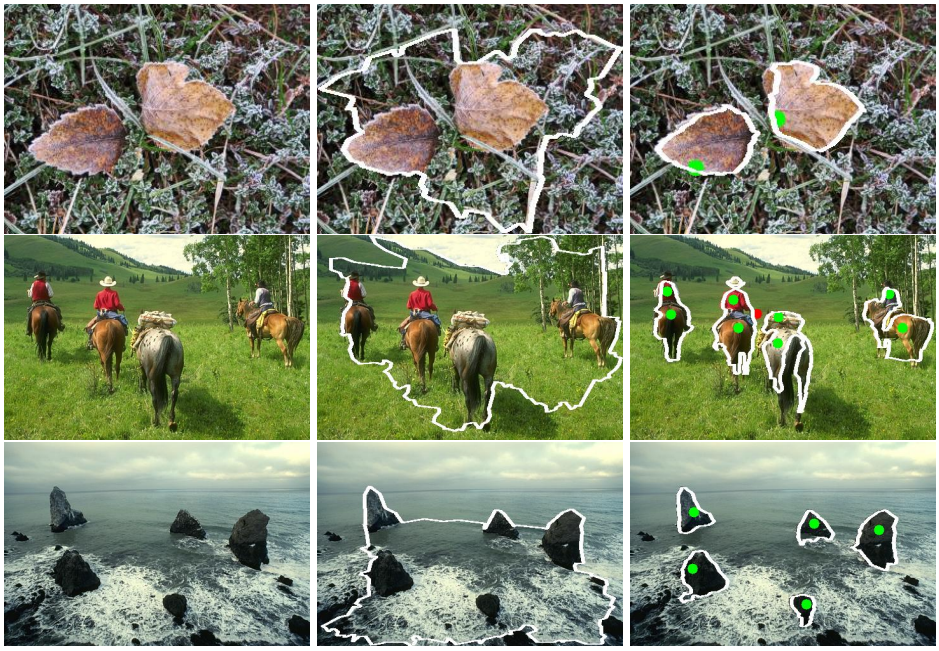

Figure 2.15: Example of multiple closures. Middle row shows the failure cases from RRC, where only one single object closure is found. Right row shows our results, which successfully find multiple objects.

2.5.4 Multiple Closures

As mentioned in (Stahl and Wang, 2007), the authors attempt to solve multiple contour closures by removing all the edges associated with the detected one and repeating their single-detection method. However, this approach is problematic as shown in Fig. 2.15. If the single-detection method select two closures at the very beginning (shown in the middle of Fig. 2.15) and removes all the edges related to these two, It is not possible to get these two closures back in subsequent step of their algorithm. However, our algorithm can select all five closures in one shot as shown on the right of Fig. 2.15.

2.5.5 Results on Interactive Segmentation

Table 2.1: Average interaction efforts required to reach an F=0.95.

Method	BJ	RW	SP	GSCseq	EulerSeg
Avg. Effort	5.51	6.48	4.54	2.30	**2.06**

ISEG Results: We compare our algorithm with the state-of-art interactive segmentation methods on the ISEG dataset (Gulshan et al., 2010). These include Boykov & Jolly (BJ) with no shape constraints (Boykov and Jolly, 2001b), shortest paths method (SP) (Bai and Sapiro, 2009), Random Walker (RW) (Grady, 2006), and Geodesic Star Convexity sequential system (GSCseq) (Gulshan et al., 2010). We measure the effects of user interactions using a robot user setting. All the algorithms are set up with the default setting using the robot engine from (Gulshan et al., 2010). The question we ask is how much user interaction is required to get a region F-measure score of 0.95 for the ISEG dataset (restricted to cases where all algorithms can achieve F=0.95 within 20 strokes). Table 2.1 demonstrates that EulerSeg requires the fewest stokes to reach a reasonable segmentation. On the other hand, as ISEG already provides a good initialization, which benefits the rest of the methods in building an appearance model, the extra effort needed for a good segmentation is reduced. It is important to note that seeds in EulerSeg act as a pure geometric role and enable segmentation with fewer stroked pixels. These results are shown in Fig. 2.16.

ISEG Results without Initialization: As ISEG already provides a good initialization, which benefits the other methods for building up an appearance model, the extra effort needed for a good segmentation is reduced. It is important to note that seeds in EulerSeg serve a purely geometric role and enable segmentation with fewer indications. When starting with no initialization, EulerSeg is still able to segment the object(s). Here we provide additional results on our algorithm without initialization (which we refer as EulerSeg-0). In EulerSeg-0, we start our segmenta-

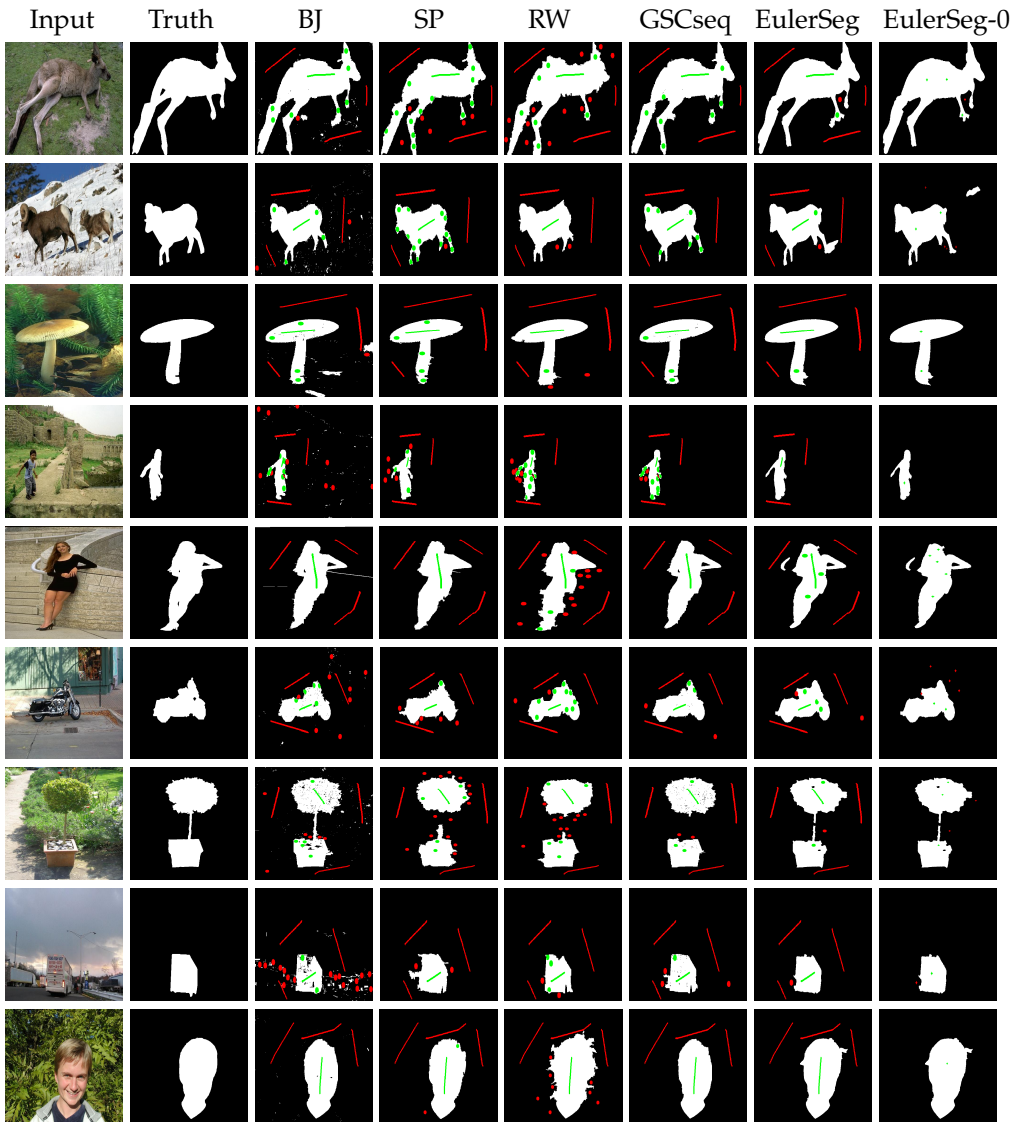

Figure 2.16: Sample results from ISEG. Red strokes are background seeds while green strokes are foreground seeds. Strokes for column 3-7 are the default setting in the robot engine Gulshan et al. (2010) with brush radius equal to 8 pixels, while strokes feed in EulerSeg-0 are simple point seeds, whose radius is one pixel. We marked seeds for EulerSeg-0 as crosses just for noticeability. **Best viewed in color**.

tion without any seeds, then we use the robot engine to add seed points iteratively.

Fig. 2.16 shows the first segmentation found by the robot user which has a region F-Measure of at least 0.95. The varying number of strokes seen between different algorithms on the same image shows the amount of additional input necessary to achieve this level of accuracy. Fig. 2.16 along with Tab. 2.1 demonstrates that by interpreting the seeds topologically, the user interactions needed to get high-accuracy segmentation can be significantly reduced.

Running Time: The preprocessing to generate superpixels is the primary computational cost, and is the only resolution-dependent component of our method. The total number of variables in our ILP typically is about 2000 (with residuals); on a 3GHz i7 CPU, each iteration of the linear ratio objective solver takes < 1s. Given superpixels, our implementation creates a segmentation usually within 15 iterations, though for some exceptionally textured images or those with a large number of components our algorithm may take more than 1 minute to solve.

2.6 Summary

This chapter presents a framework based on discrete calculus which unifies the contour completion and segmentation settings. This is augmented with a Euler characteristic constraint which allows us to specify the topology of the segmented foreground. Our model easily accommodates user indications and multiple foreground regions. Two solvers specialized toward different aspects of the problem are derived: one based on an ILP over superpixels and the other a branch-and-bound using completions with spirals to join edgelets. We demonstrate our model finds salient contours across a large dataset, showing significant improvement over similar methods.

3 SCENE PARSING WITH IMAGE LEVEL TAGS

This chapter extends the binary segmentation setting to multiple classes. Here, we would like to associate foreground objects with a semantic concept (e.g., person, car, dog), as well as actually classify background regions (e.g., sky, grass, road). Specifically, we tackle the problem of weakly labeled scene parsing, where the only source of annotation are image tags encoding which classes are present in the scene. This is an interesting setting since tags are either readily available within most online photo collections or they can be easily obtained at a much less cost than annotating entire images (at a pixel level). However, this is a difficult problem as no pixel-wise labelings are available, not even at training time. This chapter shows that this problem can be formalized as an instance of learning in a latent structured prediction framework, where the graphical model encodes the presence and absence of a class as well as the assignments of semantic labels to super-pixels. As a consequence, we are able to leverage standard algorithms with well understood properties. We demonstrate the effectiveness of our approach using the challenging SIFT-flow dataset and show average per-class accuracy improvements of 7% over state-of-the-art methods. A preliminary version of this chapter was published in (Xu et al., 2014).

3.1 Problem Description

Traditional approaches to semantic segmentation require a large collection of training images labeled at the pixel level. Most approaches annotate a object with a polygon by launching tools like LabelMe (Russell et al., 2008) with crowd-sourcing systems such as Amazon Mechanical Turk (MTurk). Fig. 3.1 shows a few examples of such annotations. Despite the availability of such systems, densely labeling images is still a very expensive process,

Figure 3.1: Example annotations from LabelMe[1]. The first row shows good annotations of building, tree, ship and person. The second tow shows bad object labels: the building/tree is not entirely labeled (in the first two images), and there are three people inside the center polygon for the bottom right image.

particularly since multiple annotators are typically employed to label each image. Furthermore, a quality control process is frequently required in order to sanitize the annotations. For instance, the second row in Fig. 3.1 shows a few cases, where users provide poor annotations. Any algorithm utilizing such training date will invariably perform poorly.

Here, we are interested in leveraging weak annotations in order to reduce the labeling cost. In particular, we exploit image tags capturing which classes are present in the scene as our sole source of annotation (see Fig. 3.2 for an illustration). This is an interesting setting as tags are either readily available within most online photo collections or they can be easily obtained at a much cheaper cost than annotating each pixel. This task is,

[1] http://labelme.csail.mit.edu/guidelines.html.

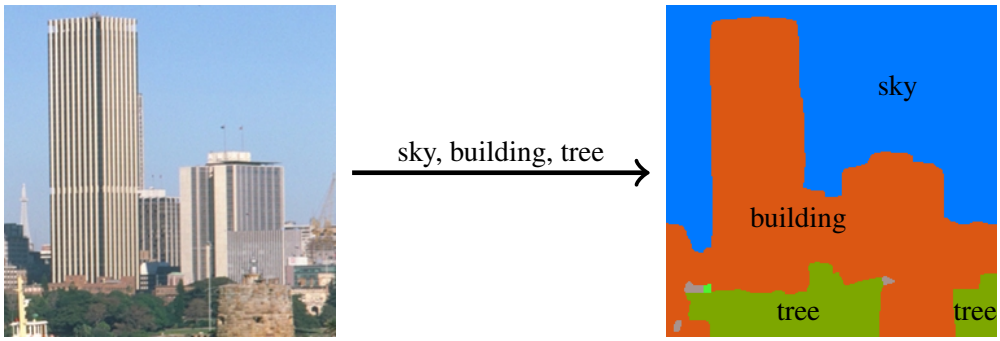

Figure 3.2: Our approach takes labels in the form of which classes are present in the scene during training, and learns a segmentation model, even though no annotations at the pixel-wise are available.

however, very challenging, partly because an appearance model cannot be trained since the assignment of superpixels to semantic labels is unknown, even at training time.

Several approaches in the literature have investigated this setting. In early work, Verbeek and Triggs proposed the latent aspect model (Verbeek and Triggs, 2007). They employ probabilistic latent semantic analysis (PLSA) to model each image as a finite mixture of latent semantic classes, and label regions in images with these classes. The authors extended PLSA to capture spatial relationship via a Markov random field. This model was further extended in a series of papers by (Vezhnevets and Buhmann, 2010; Vezhnevets et al., 2011, 2012), for example, to leverage information between multiple images. However, the resulting optimization problem is very complex and non-smooth, making learning a very difficult task. As a result, several heuristics were employed to make the problem computationally tractable.

In this chapter, we show that this problem can be formalized as the one of learning in a latent structured prediction framework, where the graphical model encodes the presence/absence of a class as well as the assignments of semantic labels to superpixels. As a result, we are able

to leverage algorithms with good theoretical properties which have been developed for this more general setting. Under our model, different levels of supervision can be simply expressed by specifying which variables are latent and which are observed, without changing the learning and inference algorithms. We demonstrate the effectiveness of our approach using the challenging SIFT-flow dataset (Liu et al., 2011), showing improvements of 7% in terms of mean class accuracy over the state-of-the-art. In the next section, we first review related work. We then present our weakly label segmentation framework, followed by an experimental evaluation and conclusions.

3.2 Related Work

Many different techniques have been proposed to handle the fully supervised setting where pixel-wise labels are available at training time. Amongst the most successful techniques are approaches based on object proposals. They first extract bottom-up regions, and then learn a classifer (e.g., linear SVM) for each semantic class. A greedy method is then followed to obtain the final segmentation by picking the class with maximum likelihood (Gu et al., 2009; Carreira et al., 2012). Another popular approach is to formulate segmentation as inference in a (conditional) Markov random field (Cardinal et al., 2010), particularly when seeking a full holistic scene interpretation (Yao et al., 2012; Ladický et al., 2010).

It is relatively easy for an annotator to provide information about which objects/classes are present in the scene. It is, however, significantly more tedious to carefully outline all visible objects. As a consequence, annotation time and cost can be significantly reduced by leveraging image tags, particularly as these annotations are readily available in many image collections.

There has been, however, little work in the weakly labeled setting due

to the fact that it is significantly more challenging than the fully supervised task. One of the first approaches to learn a segmentation model given only image tags is the latent aspect model of (Verbeek and Triggs, 2007), which leverages several appearance descriptors and the image location to learn a probabilistic latent semantic analysis (PLSA) model. The name 'aspect model' originates from the famous topic models for document classification. Here, the 'aspects' refer to pixel class labels. Since these models do not capture the spatial 2D relationships commonly observed in images, PLSA was used as unary features in a Markov random field. Generalizations were subsequently introduced in a series of papers (Vezhnevets and Buhmann, 2010; Vezhnevets et al., 2011, 2012). Different from the latent aspect model, these new approaches leverage label correlations between different images. However, they result in complex optimization problems which are non-convex, non-smooth and thus very difficult to optimize.

Another form of weak supervision are 2D bounding boxes. Grab-cut and its extensions have been widely used for interactive figure/ground segmentation (Boykov and Jolly, 2001a; Rother et al., 2004). These methods learn Gaussian mixture models for the foreground and background, and a binary MRF encoding both appearance and smoothness is employed to perform the segmentation. Strokes are another popular way to provide weak annotations and are typically used with a human in the loop to correct mistakes. In (Pandey and Lazebnik, 2011), the deformable part-based model (Felzenszwalb et al., 2010) is used with latent structured support vector machines to exploit weak labels in the form of bounding boxes. Recently, (Chen et al., 2014) extended the form of weak supervision in 3D for object segmentation given annotated 3D bounding boxes. They formulate the segmentation problem with a binary Markov random field which exploits appearance models, stereo, point clouds (might be noisy), 3D CAD models, and topological constraints. The resulted method auto-

matically generates very accurate object segmentations which even match human labeling performance.

A related problem is cosegmentation, where one is interested in segmenting objects which concurrently appear in a set of images (Rother et al., 2006; Mukherjee et al., 2009). Most previous methods focus on the setting where a single foreground object is present in all images (Vicente et al., 2010; Mukherjee et al., 2011; Collins et al., 2012). This setting has been extended to segment multiple objects by analyzing the subspace structure of multiple foreground objects (Mukherjee et al., 2012), using a greedy procedure with submodular optimization (Kim and Xing, 2012), or by grouping image regions via spectral discriminative clustering (Joulin et al., 2012).

The work most related to ours is (Vezhnevets et al., 2012). They formulate the problem of weakly labeled semantic segmentation using a conditional random field (CRF), where nodes represent semantic classes at the superpixel level, unary potentials encode appearance and pairwise potentials encode smoothness. Their key contribution is a three step algorithm to learn the appearance model and the CRF weights. In particular, after every update of the CRF weights, an alternating optimization iterates between finding the pixel-wise labeling given the current model and updating the appearance model given an estimated labeling. The authors view optimization of the feature weights as a model selection procedure where every possible weight vector defines a different model. The optimization criteria employed is expected agreement, which is computed by partitioning the data into two parts which are encouraged to agree in their predictions. As the objective function in (Vezhnevets et al., 2012) is non-differentiable, they resort to Bayesian optimization to select the next set of parameters. This makes learning extremely difficult and computationally expensive.

In contrast, in this chapter we show that the problem of semantic seg-

mentation from weakly labeled data (in the form of image-level tags indicating whether a class is present or absent in the image) can be formulated as learning in a structured prediction framework with latent variables. As a consequence, well studied algorithms such as hidden conditional random fields (HCRFs) (Quattoni et al., 2007) or latent structured support vector machines (LSSVMs) (Yu and Joachims, 2009) as well as efficient extensions (Schwing et al., 2012) can be leveraged. This results in simpler optimization problems that can be optimized by algorithms possessing good theoretical guarantees.

3.3 Weakly Labeled Semantic Segmentation

In this section, we investigate how weak supervision can be used in order to perform semantic segmentation. In particular, we focus on the case where the supervision is given by means of a set of tags, describing which classes are present in the image. Towards this goal, we frame the weakly supervised semantic segmentation problem as a learning problem in a graphical model encoding the presence and absence of each class as well as the semantic class of each superpixel.

3.3.1 Semantic segmentation from tags

More formally, let $y_i \in \{0,1\}$ be a random variable describing whether the i-th class is present in the image, with $i \in \{1,\dots,C\}$ indexing the semantic classes. Further, let $h_j \in \{1,\dots,C\}$ be a random variable denoting the semantic label associated with the j-th superpixel, and let x be the image evidence. We define $\mathbf{h} = (h_1,\dots,h_N)$ to be the set of segmentation variables for all superpixels within one image, and $\mathbf{y} = (y_1,\dots,y_C)$ the set of binary variables indicating for all classes their presence/absence. Note that we assume no training examples to be available for superpixel labels $\mathbf{h}$, and only image level tags $\mathbf{y}$ to be labeled. Employing the aforementioned

notation, we define the probability for a given configuration $(\mathbf{y}, \mathbf{h})$ given an image x to be

$$p_\epsilon(\mathbf{y}, \mathbf{h} \mid x) = \frac{1}{Z_\epsilon(w)} \exp \frac{w^\top \phi(\mathbf{y}, \mathbf{h}, x)}{\epsilon}, \tag{3.1}$$

where $Z_\epsilon(w)$ is the normalizing constant also known as the partition function. Note that the weights w are the parameters of the model and ϵ is a temperature parameter.

Fig. 3.3 shows the graphical model encoding the dependencies introduced by this probabilistic model, with gray-colored nodes depicting observed variables. We note that this architecture is similar to the first two layers in the holistic model of (Yao et al., 2012), but we use different potentials and perform semantic segmentation in the weakly labeled setting.

Next, we define the potentials $\phi(\mathbf{y}, \mathbf{h}, x)$ to be the sum of unary terms encoding the likelihood of the tags $\phi^{pres}(x, y_i)$, unary potentials encoding the appearance model for segmentation $\phi^{ap}(x, h_j)$ and pairwise potentials ensuring compatibility between both types of variables $\phi^{co}(y_i, h_j)$ as,

$$\begin{aligned} w^\top \phi(\mathbf{y}, \mathbf{h}, x) = & \sum_i w_i^{pres\top} \phi^{pres}(x, y_i) \\ & + \sum_j w_j^{ap\top} \phi^{ap}(x, h_j) + \sum_{i,j} w_{i,j}^{co\top} \phi^{co}(y_i, h_j). \end{aligned} \tag{3.2}$$

We now discuss the potentials employed in more details.

Presence/Absence potential $\phi^{pres}(x, y_i)$: We construct a 2D vector to encode the presence of each class. During training, this potential is built from the ground truth, i.e., $\phi^{pres}(x, y_i) = [1; -1]$ if class i is absent, while $\phi^{pres}(y_i, x) = [-1; 1]$ if class i is present. At test time, when this information is latent, this potential comes from an image level tag classifier.

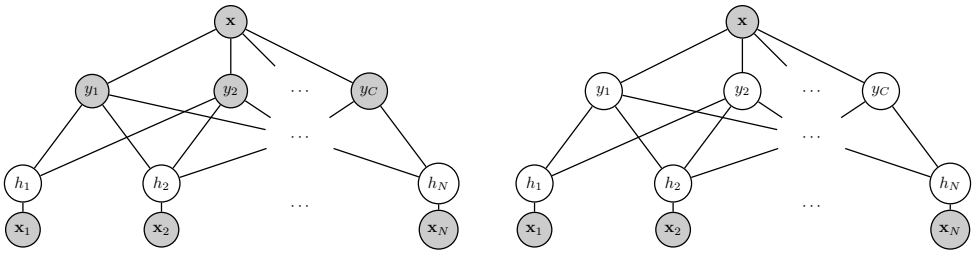

Figure 3.3: **Graphical Model:** (Left) Graphical model for learning as well as inference when the tags are provided at test time. (Right) Graphical model for inference when the tags are not provided at test time. Gray nodes are observed, and whilt nodes are hidden variables.

We refer the reader to the experimental section for more details about the specific form of this predictor. Note that typically one will use a predictor both at training and testing time, however, we have found the use of the oracle (i.e., truth) predictor at training to yield better results in practice. We hypothesize that this is due to the fact that in this setting the supervision is very weak.

Appearance model $\phi^{ap}(x, h_j)$: We utilize the superpixel features of (Tighe and Lazebnik, 2013b), which include texture/SIFT, color, shape, location and GIST. This results in a 1690 dimensional feature vector, which we reduce to a 100 dimensional vector using PCA. To form the final feature, we append the superpixel location (i.e., y-coordinate of its center) to form our final feature. Note that we learn a different set of weights for each class, yielding a $101 \times C$ dimensional feature vector.

Compatibility $\phi^{co}(y_i, h_j)$: The compatibility term encourages the consistency between the class presence variables and the superpixels, such that the information is propagated all the way to the segmentation. In particular, it penalizes configurations where a superpixel is labeled with a

class that is inferred to be absent. Thus

$$\phi^{co}(y_i, h_j) = \begin{cases} -\eta & \text{if } y_i = 0 \text{ and } h_j = i \\ 0 & \text{otherwise} \end{cases}$$

where η is a big number (10^5 in our experiments).

3.3.2 Learning in the Weakly Labeled Setting

During learning, we are interested in estimating a linear combination of features such that the distribution in Eq. 3.2 is able to discriminate between 'good' and 'bad' assignments for variables $\mathbf{y}$ and $\mathbf{h}$. To define 'good' we are given a training set of data samples. Contrasting the fully supervised setting where the training samples contain fully labeled configurations $(\mathbf{y}, \mathbf{h})$, the available data is only partly labeled. In particular, the training set $\mathcal{D}$ consists of $|\mathcal{D}|$ image-tag pairs $(\mathbf{y}, x)$, i.e., $\mathcal{D} = \{(\mathbf{y}, x)_i\}_{i=1}^{|\mathcal{D}|}$.

During learning, a loss function $\ell(\hat{\mathbf{y}}, \mathbf{y})$ is commonly included to bias the algorithm. So we augment the scoring function defined in Eq. 3.1 as,

$$p_\epsilon^\ell(\hat{\mathbf{y}}, \hat{\mathbf{h}}, x) = \frac{1}{Z_\epsilon^\ell(w)} \exp \frac{w^\top \phi(\hat{\mathbf{y}}, \hat{\mathbf{h}}, x) + \ell(\hat{\mathbf{y}}, \mathbf{y})}{\epsilon} \tag{3.3}$$

Here, we want to find a weight vector w, which minimizes the sum of the negative (loss-augmented) marginal log-posterior of the training data $\mathcal{D}$ and a regularization term which originates from a prior distribution on w. The resulting model reads as follows

$$\min_w \frac{1}{2}\|w\|_2^2 - \sum_{(\mathbf{y},x)\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{h}}} p_\epsilon^\ell(\mathbf{y}, \hat{\mathbf{h}} \mid x) \tag{3.4}$$

Note that we marginalize over the unobserved superpixel variables $\mathbf{h}$ to obtain the likelihood of the observed data (i.e., class labels).

The aforementioned program generalizes a few well-known settings.

Letting $\epsilon = 0$, we obtain structured support vector machines with latent variables as introduced by (Yu and Joachims, 2009), while setting $\epsilon = 1$ yields the hidden conditional random field of (Quattoni et al., 2007). In case of fully observed data we obtain the conditional random field framework of (Lafferty et al., 2001) or the structured support vector machine of (Taskar et al., 2003; Tsochantaridis et al., 2005) when employing $\epsilon = 1$ and $\epsilon = 0$ respectively.

The weakly labeled setting is significantly more difficult to solve than for general graphical models. The additional challenge besides summations over exponentially sized sets $\mathbf{h}$ and $\mathbf{y}$, is the non-convexity of the objective given in (3.4) resulting from the partition function. We note, however, that the cost function of the program given in (3.4) is a difference of terms, each being convex in the parameters w. We exploit this fact and employ the concave-convex procedure (CCCP) (Yuille and Rangarajan, 2003), which is a generalization of expectation maximization (EM) to minimize (3.4).

CCCP is an iterative approach. At each iteration we linearize the concave part at the current iterate w and solve the remaining convex objective augmented by a linear term to update the weight vector w. Importantly, this approach is guaranteed to converge to a stationary point (Sriperumbudur and Lanckriet, 2009). To linearize the concave part, we are required to compute an expectation of the feature vector $\phi(\mathbf{y}, \mathbf{h}, x)$ w.r.t.a distribution over the unobserved variables $\mathbf{h}$. More formally this expectation is defined as

$$\mathsf{E}_{p(\hat{\mathbf{h}}|x)}\left[\phi(\mathbf{y}, \hat{\mathbf{h}}, x)\right] = \sum_{\hat{\mathbf{h}}} p(\hat{\mathbf{h}} \mid x)\phi(\mathbf{y}, \hat{\mathbf{h}}, x).$$

Given this expectation, we solve a fully supervised objective with modified empirical means. Note that the derivation naturally results in a two-step approach where we first compute a distribution over the unobserved variables $\mathbf{h}$ to obtain the expectation, before using this information to solve

Structured prediction with latent variables
Iterate between

1. The latent variable prediction problem:

$$\forall x \text{ compute} \quad \mathrm{E}_{p(\hat{\mathbf{h}}|x)}\left[\phi(\mathbf{y}, \hat{\mathbf{h}}, x)\right]$$

2. Solving the parameter update task

$$\min_{w} \frac{1}{2}\|w\|_2^2 + \sum_{(\mathbf{y},x)\in\mathcal{D}} \left(\epsilon \ln Z_\epsilon^\ell(w) - w^\top \mathrm{E}_{p(\hat{\mathbf{h}}|x)}\left[\phi(\mathbf{y}, \hat{\mathbf{h}}, x)\right]\right)$$

Figure 3.4: Latent Structured Prediction via CCCP.

the fully supervised learning problem. The procedure is summarized in Fig. 3.4.

For the first step it is crucial to notice that in our graphical model we can trivially solve the 'latent variable prediction problem' given the bi-partite model of the weakly labeled segmentation task. Assuming the ground truth tags $\mathbf{y}$ to be known (see Fig. 3.3), the model decomposes into unary potentials over superpixels, and inference can be efficiently and exactly solved to yield a distribution $p(\hat{\mathbf{h}} \mid x)$. For the second step we need to solve a fully supervised learning task. This task is identical to the one encountered during inference, i.e., we are required to predict a maximizer for the joint model. We therefore defer discussion to the inference section 3.3.4.

3.3.3 Loss function

The distribution of class presence as well as the distribution of pixel-wise labelings follows a power law distribution (i.e., many classes occur very rarely). In order to take this into account we derive a loss function

which employs the statistics of class presence at the image level. As the segmentation metric is average per-class accuracy, our loss gives more importance for mistakes in classes that appear very rarely. In particular, for each class i, we count how many training images contain this class, and then normalize this frequency vector $\mathbf{t}$ to sum to 1. The loss function $\ell(\hat{\mathbf{y}}, \mathbf{y})$ is then defined to decompose into a sum of unary terms, i.e., $\ell(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i \in \{1,\dots,C\}} \ell_i(\hat{y}_i, y_i)$ with

$$\ell_i(\hat{y}_i, y_i) = \begin{cases} \frac{1}{t_i} & \text{if } y_i \neq \hat{y}_i \text{ and } y_i = 0 \\ t_i & \text{if } y_i \neq \hat{y}_i \text{ and } y_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

where y_i is the ground truth label, and $\hat{y}_i$ is the prediction for the i-th class. Note that our loss function is only defined on the class presence variables $\mathbf{y}$ that are observed during training.

3.3.4 Inference

The configuration with the minimum energy or the highest probability $p(\mathbf{y}, \mathbf{h} \mid x)$, also known as the maximum a posteriori (MAP) estimate, can be computed by solving the following problem

$$(\mathbf{y}^*, \mathbf{h}^*) = \arg\max_{\mathbf{y},\mathbf{h}} w^\top \phi(\mathbf{y}, \mathbf{h}, x) \tag{3.6}$$

given an image x. This is an NP-hard task since the optimization is equivalent to an integer linear program. Fortunately, linear programming (LP) relaxations have proven very effective. We employ a message passing approach for leveraging the graphical model structure. In particular, we use distributed convex belief propagation (dcBP) (Schwing et al., 2011), which partitions the graph and imposes agreement between the beliefs in the boundaries. The full inference problem is splited into several local

optimization problems (one per machine), and they are solved in parallel. Additional Lagrange multipliers are then sent as messages between machines to ensure belief agreement. The resulted distributed message-passing algorithm hence preserves the convergence guarantees of existing methods. Note that this is not the case for other message passing algorithms such as loopy belief propagation.

3.4 Experimental Evaluation

We perform our experiments using the SIFT-flow segmentation dataset (Liu et al., 2011), which contains 2688 images and $C = 33$ classes. This dataset is very challenging due to the large number of classes (4.43 classes per image) as well as the fact that their frequency is distributed with a power-law. As shown in the first line of Table 3.2, a few 'stuff' classes like sky, sea and tree are very common, while the 'object' classes like person, bus and sun are very rare. We use the standard dataset split (2488 training images and 200 testing images) provided by (Liu et al., 2011).

Following (Vezhnevets et al., 2012) we report mean per-class accuracy as our metric. This metric gives the same importance to each class, independently of their frequency. We construct our superpixels using the ultrametric contour map of (Arbelaez et al., 2011), which respects boundaries well even when a small number of superpixels is used. In our experiments, we set the boundary probability threshold to be 0.14, which results in 19 segments per image on average.

In our experiments we exploit two settings. In the first case we follow the standard weakly labeled setting, in which only image level tags are given for training and no annotations are given at the pixel-level. During testing, no source of annotation is provided. Learning in this setting corresponds to the graphical model in Fig. 3.3 (left), while inference is shown on Fig. 3.3 (right). In the second setting we assume that tags are

Method	Supervision	Per-class accuracy (%)
(Tighe and Lazebnik, 2013a)	full	39.2
(Tighe and Lazebnik, 2013b)	full	30.1
(Liu et al., 2011)	full	24
(Vezhnevets et al., 2011)	weak	14
(Vezhnevets et al., 2012)	weak	21
Ours (CNN-Tag)	weak	**27.9**
Ours (Truth-Tag)	weak	**44.7**

Table 3.1: Comparison to state-of-the-art on the SIFT-flow dataset. We outperformed the state-of-the-art in the weakly supervised setting by 7%.

given both at training and test time, and thus the graphical model in Fig. 3.3 (left) depicts both learning and inference. This is a natural setting when employing image collections where tags are readily available.

Our first experiment utilizes tags only during training. We utilize an image-tag classifier which leverages deep learning in order to construct presence/absence potential $\phi^{pres}(x, y_i)$ at test time. In particular, we first extract a 4096 dimensional feature vector for each image from the second to last layer of a convolutional neural network (CNN) pre-trained on ImageNet (Deng et al., 2009). We use the publicly available implementation of (Jia, 2013) to compute the features, and a linear SVM per class to form the final potential. We refer to this setting as "Ours (CNN-Tag)."

Comparison to the state-of-the-art: Tab. 3.1 compares our approach to state-of-the-art weakly labeled approaches. For reference, we also include the state-of-the-art when pixel-wise labels are available at training time (fully labeled setting). We would like to emphasize that our approach outperforms significantly (7% higher) all weakly labeled approaches. Furthermore, we even outperform the fully supervised method developed by (Liu et al., 2011). The per-class rates for each class are provided in Tab. 3.2. We observe that our approach performs well for classes which have

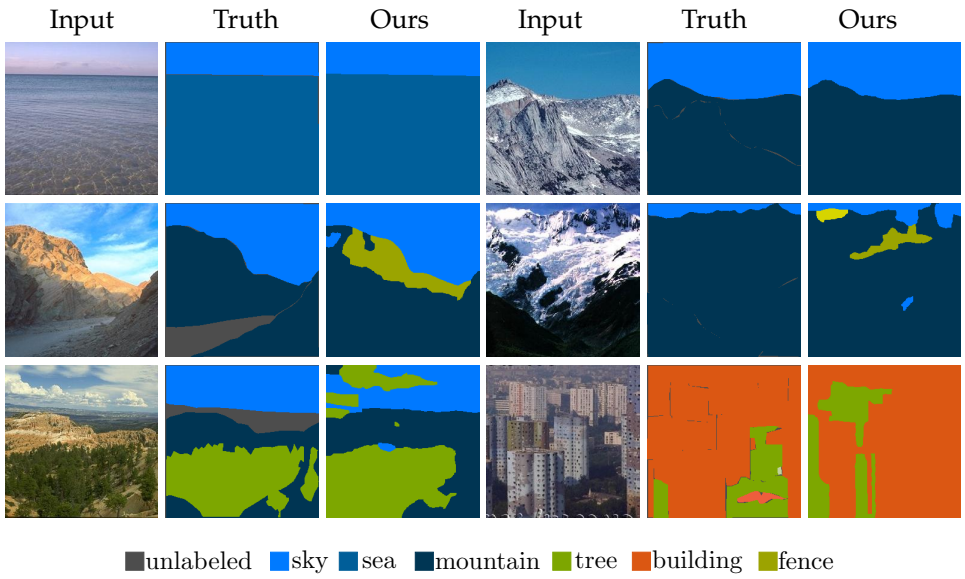

Figure 3.5: Sample results when tags are predicted at test time using a convolutional net. **Best viewed in color.**

very distinctive and consistent appearance, e.g., sand, sun, staircases. We missed a few classes, e.g., bus, crosswalk, bird, due to their largely varying appearance and small training set size.

Quality of image-tag prediction: Our CNN-Tag predictor predicts tags with an accuracy of 93.7%, which is measured as the mean of the diagonal of the confusion matrix. The last row of Table 3.2 shows the performance of the tag predictor for each class. Interestingly, tag prediction errors do not correlate well with segmentation errors, e.g., crosswalk and bird tags are predicted with very high accuracy, but segmentation accuracy is very low for both classes.

Qualitative results: Fig. 3.5 and Fig. 3.6 show success and failure cases respectively. Typical failure modes are due to under-segmentation when creating the superpixels as well as dealing with classes where different

Figure 3.6: Failure cases when tags are predicted using a convolutional net at test time. **Best viewed in color.**

instances have very different appearance, e.g., due to viewpoint changes.

Tags given at both training and testing: In our second setting, tags are given both at training and testing time. Note that the training procedure here is identical to the previous setting. However, at test time our image level class potentials are built from observed ground truth tags. We denote this setting as "Ours (Truth-Tag)." As shown in Tab. 3.1, we almost double the per-class accuracy of the previous setting. Surprisingly, we outperformed all fully labeled approaches while not requiring any example to be labeled at the pixel-level. Fig. 3.7 depicts qualitative results for this setting. When image level tags are given, our approach is able to identify more challenging classes, e.g., buildings.

Partial tags given at test time: We further evaluate our model when only a subset of the tags are provided. For each run, we randomly sample a small portion of ground truth (GT) tags, and predict the remaining ones via our CNN tag classifier. The combined potentials are then fed into our model for inference. We conduct our experiments using four different sample ratios $\{0.1, 0.2, 0.3, 0.5\}$. For each setting, we repeat our procedure

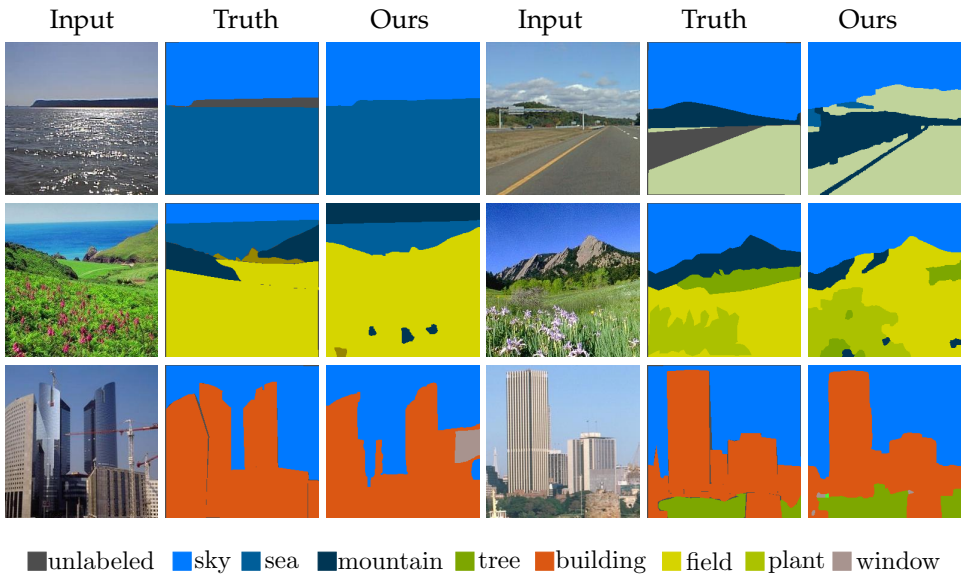

Figure 3.7: Sample results when ground truth tags are given at test time. **Best viewed in color.**

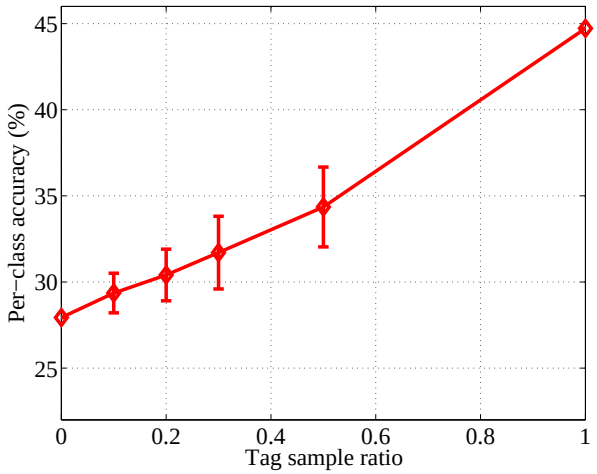

Figure 3.8: Per-Class accuracy as a function of the percentage of ground-truth tags available at test time.

10 times and report the mean and standard deviation. As shown in Fig. 3.8, our approach gradually improves when more GT tags are given.

3.5 Summary

We have presented an approach to semantic segmentation which is able to exploit weak labels in the form of image tags when no pixel-wise labeling are available. We have shown that this problem can be formulated as structured prediction in a graphical model with latent variables. Unlike existing approaches, this allowed us to leverage standard algorithms with good theoretical guarantees. We have demonstrated the effectiveness of our approach and showed improvements of 7% over the state-of-the-art in this task. Our novel view of the problem can be used to incorporate other types of supervision without changing the learning or inference algorithms. In the future we plan to exploit other annotations such as the type of scene or bounding boxes as well as other forms of learning such as active learning (Luo et al., 2013) to further reduce the need of supervision.

Class	Tag Frequency	CNN-Tag	Truth-Tag	CNN-ILT
sky	85.37	8.23	12.34	93.00
tree	50.08	30.19	27.90	81.50
building	45.78	17.15	23.33	86.50
mountain	37.86	35.27	33.00	82.50
road	31.71	5.52	10.03	91.00
car	23.83	8.79	14.16	94.50
sidewalk	16.96	1.98	4.52	90.00
sea	14.15	15.93	18.75	97.50
window	13.38	9.02	10.81	93.00
person	12.66	18.46	22.04	89.50
plant	12.42	4.01	37.07	86.00
rock	10.17	39.84	83.05	91.00
river	9.81	46.39	64.57	92.50
grass	9.32	55.21	63.13	89.50
door	9.28	53.21	49.29	93.50
field	8.88	57.97	81.40	92.50
sign	8.08	32.29	41.72	91.50
streetlight	8.08	15.20	22.04	93.00
sand	5.83	86.09	87.58	95.50
fence	5.47	48.12	81.25	93.50
pole	3.46	1.75	36.86	95.50
bridge	3.38	13.78	39.54	94.50
boat	3.18	28.23	74.89	96.50
awning	2.85	20.27	44.45	95.00
staircase	2.41	77.41	79.51	98.00
sun	2.17	100.00	100.00	100.00
balcony	1.81	0.00	37.65	99.00
crosswalk	1.45	7.78	23.71	99.00
bus	1.21	0.00	58.52	98.00
bird	0.28	0.00	58.49	99.00
mAp		27.94	44.72	93.07

Table 3.2: **Accuracy for each class:** First column shows tag frequency (percentage of images) for each class. Column 2 and 3 show segmentation accuracy for each class when a CNN tag predictor or the ground truth tags are used respectively. The last column shows the accuracy of our image tag predictor for each class.

4 SEMANTIC SEGMENTATION UNDER VARIOUS FORMS OF WEAK SUPERVISION

This chapter further expands the range of weak supervision from image level tags to various forms of weak labels for scene parsing. We present a unified approach that can learn from multiple types of weak supervision (e.g, image level tags, bounding boxes, partial labels) and produce pixel-wise labelings. We formulate the weakly supervised semantic segmentation problem as a max-margin clustering, where supervision comes as additional constraints on the assignments of pixels to class labels. This allows us to have a unified formulation that can exploit arbitrary combinations of different types of supervision. We then build an efficient learning algorithm, which is parallelizable and scalable for large image sets. Our approach when compared to existing weakly labeled methods, is very efficient, taking only 20 minutes for learning (one order of magnitude faster) and a fraction of a second for inference. In our recent work (Xu et al., 2015b), we conduct a rigorous evaluation of the proposed method on standard benchmarking datasets for various weakly labeled settings and show that our approach outperforms the state-of-the-art by 12% on per-class accuracy, while maintaining comparable per-pixel accuracy.

4.1 Problem Description

Semantic segmentation is one of the most fundamental challenges in computer vision, and conventional fully supervised algorithms have demonstrated promising results (Liu et al., 2011; Farabet et al., 2012; Eigen and Fergus, 2012; Singh and Kosecka, 2013; Tighe and Lazebnik, 2013b, 2014; Yang et al., 2014; Ladický et al., 2010; Ladicky et al., 2010; Yao et al., 2012; Schwing and Urtasun, 2015). However, in order to train fully supervised systems, a set of training examples with semantic labels for each pixel in

an image is required. Considering the recent performance improvements obtained when employing millions of data points, it is obvious that the size of the training data is one of the bottlenecks for semantic segmentation. This is because labeling each pixel with a semantic category is a very expensive and time-consuming process, even when utilizing crowd-sourcing platforms such as Amazon Mechanical Turk.

Compared to the massive size of modern visual data – everyday, more than 300 million images are uploaded to Facebook – only a tiny fraction is assigned accurate pixel-wise annotations. For instance, in the ImageNet dataset (Deng et al., 2009), 14 million images are assigned scene categories and 500,000 images are annotated with object bounding boxes but only $4,460$ images are segmented at the pixel level (Guillaumin et al., 2014). The reason for this disparity is rather obvious: for almost any given image we are able to quickly decide whether an object is present in a scene, but careful delineation is tedious. Therefore, certainly image tags, but sometimes even bounding boxes, and occasionally partial labels in the form of user scribbles are either easily collected or are even readily available in large photo collections of websites such as Flickr and Facebook.

In the absence of large pixel-wise annotated datasets, the development of visual parsing algorithms that benefit from weakly labeled data is key to further improving the performance of semantic segmentation. Supervision in the form of partial labels has been effectively utilized in interactive object segmentation via graph-cuts (Boykov and Jolly, 2001a), random walks (Grady, 2006), geodesic shortest path (Bai and Sapiro, 2009), and geodesic star convexity (Gulshan et al., 2010). Recursive propagation of segmentations from labeled masks to unlabeled images has also been investigated (Guillaumin et al., 2014). An alternative form of weak supervision are bounding boxes. Grabcut (Rother et al., 2004) has been highly successful for binary object segmentation, taking advantage of a bounding box around the foreground object. Recent research has extended this

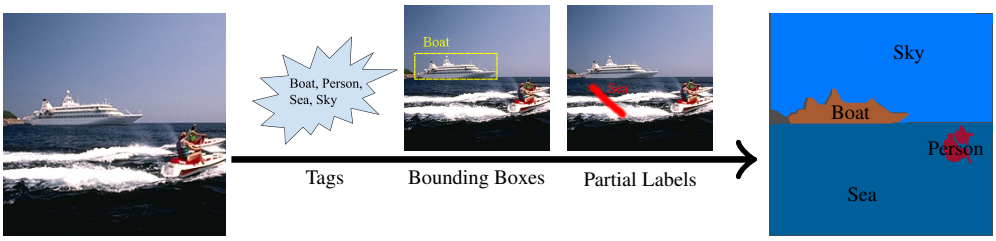

Figure 4.1: Our semantic segmentation algorithm learns from various forms of weak supervision (image level tags, bounding boxes, partial labels), and produces pixel-wise labels.

idea to semantic segmentation by building object detectors from bounding boxes (Xia et al., 2013). A more challenging setting is inference of a pixel-wise labeling, given only image level tags. Encouraging results have been presented for this weakly supervised semantic segmentation task by connecting super-pixels across images, and jointly inferring pixel labels for all images (Vezhnevets et al., 2011, 2012; Vezhnevets, 2012). An alternative are Markov random fields with latent variables denoting super-pixel labels, and observed variables representing image level tags (Xu et al., 2014; Schwing et al., 2012).

In this chapter, we propose a unified approach that takes any form of weak supervision, e.g., tags, bounding boxes, and/or partial labels, and learns to semantically segment images. We refer the reader to Fig. 4.1 for an illustration of the problem. When compared to existing weakly labeled methods (Vezhnevets et al., 2012; Xu et al., 2014), our approach is very efficient, taking only 20 minutes for learning, and a fraction of a second for inference. We conduct a rigorous evaluation on the challenging Siftflow dataset for various weakly labeled settings, and show that our method outperforms the state-of-the-art by 12% in per-class accuracy (Rubinstein et al., 2012), while maintaining a result comparable in the per-pixel metric.

4.2 Related Work

In Chapter 3, we briefly reviewed several works related to our ideas. Here, we systematically describe different segmentation methods proposed for different forms of supervision.

Fully supervised semantic segmentation: Semantic segmentation, sometimes called scene parsing, is widely studied in computer vision. A large variety of algorithms have been developed for the fully supervised setting, requiring access to a fully labeled training set. Three types of approaches are very popular. Non-parametric methods (Liu et al., 2011; Eigen and Fergus, 2012; Singh and Kosecka, 2013; Tighe and Lazebnik, 2013b, 2014; Yang et al., 2014) build pixel-wise potentials using nearest neighbors. These methods are motivated by the observation that similar semantic pixels lie close in some feature space. The second set of approaches frames semantic segmentation as an inference task using Markov random fields (MRF) (Ladický et al., 2010; Ladicky et al., 2010; Yao et al., 2012). These methods handle supervision at different levels (tags, bounding boxes, scene types) by adding variables to the energy function. The final set of methods are based on object proposal (Carreira and Sminchisescu, 2010; Endres and Hoiem, 2014; Arbelaez et al., 2014; Girshick et al., 2014; Hariharan et al., 2014), where class-independent segments are generated, and subsequently classified into different categories using features computed on those segments. These conventional methods tend to work well, given a sufficient amount of fully labeled data. Unfortunately, such pixel-wise labelings are very expensive and challenging to obtain.

Co-segmentation: A large number of researchers have therefore explored ways to make use of unlabeled or weakly labeled data. One possibility is co-segmentation (Rother et al., 2006; Mukherjee et al., 2009; Vicente et al., 2010; Mukherjee et al., 2011), where the task is to segment the shared fore-

ground from multiple images. This is a data driven approach based on the assumption that common foreground objects look alike, while differing significantly from the background. Co-segmentation has been further extended to the multi-class case via discriminative clustering (Kim and Xing, 2012; Joulin et al., 2012), and the multi-object case using subspace analysis (Mukherjee et al., 2012).

Segmentation with tags: Weakly supervised semantic segmentation and co-segmentation share the same motivation. Consider a case where one image is tagged with labels for cow and grass, and another one is assigned the categories for cow and road. It is reasonable to assume that pixels which are similar in both images take the class label for cow, while the remaining image content may take the label of other assigned categories. Researchers have attempted to tackle this challenge by connecting super-pixels across images, and jointly inferring pixel labels for all images (Vezhnevets et al., 2011, 2012; Vezhnevets, 2012; Liu et al., 2012). Alternatively, propagation via dense image correspondences (Rubinstein et al., 2012), or learning a latent graphical model between tags and super-pixel labels (Xu et al., 2014) has been considered. Promising results have also been demonstrated using deep learning (Pathak et al., 2014; Pinheiro and Collobert, 2015; Papandreou et al., 2015; Pathak et al., 2015), even though training is expensive. A recent work (Song et al., 2014) also presented a one-shot object detection algorithm with object tags.

Segmentation with semi-supervision: Another form of weak annotation is semi-supervision. Here, a user provides partial labels for some pixels within an image. This is a convenient setting, as it is reasonably easy for annotators to perform strokes which partially label an image. Such a form of supervision has been effectively utilized in interactive object segmentation with graph-cuts (Boykov and Jolly, 2001a), random walks (Grady, 2006), geodesic shortest path (Bai and Sapiro, 2009), geodesic

star convexity (Gulshan et al., 2010). Recently, (Xu et al., 2013a) and (Russakovsky et al., 2015) have employed user interactions in the form of seed points (e.g., only several pixels are annotated) to further reduce human effort.

Segmentation with bounding boxes: Also of interest for weak supervision are bounding boxes. Among the biggest successes is GrabCut (Rother et al., 2004), where a user-provided bounding box is employed to learn a Gaussian Mixture Model (GMM) differentiating between foreground and background. Recent work has extended this idea to semantic segmentation by building object detectors from multiple bounding boxes (Xia et al., 2013). (Pandey and Lazebnik, 2011) utilizes bounding boxes to locate objects of interest, within a latent structured SVM framework. 3D bounding boxes as a form of weak supervision have been shown to produce human-level segmentation results (Chen et al., 2014).

In this chapter, we describe a first attempt to employ all of the aforementioned forms of weak supervision within a single unified model. We then build an efficient learning algorithm, which is parallelizable and scalable to large image sets.

4.3 Unified Model for Various Forms of Weak Supervision

In this section, we address the problem of semantic segmentation using various forms of weak supervision, like image level tags, strokes (i.e., partial labels) as well as bounding boxes. More specifically, we are interested in inferring pixel-level semantic labels for all the images, as well as learning an appearance model for each semantic class. The latter component permits prediction in previously unseen test examples. Note that we never observe even a single labeled pixel in most of our settings. We formulate

this task using a max-margin clustering framework, where knowledge from supervision is included via constraints, restricting the assignment of pixels to class labels. We obtain a unifying formulation that is able to exploit arbitrary combinations of supervision.

4.3.1 Unified Model

Following recent research (Vezhnevets et al., 2012; Xu et al., 2014), we first over-segment all images into a total of n super-pixels. For each super-pixel $p \in \{1, \dots, n\}$, we then extract a d dimensional feature vector $\mathbf{x}_p \in \mathbb{R}^d$. Let the matrix $H = [\mathbf{h}_1, \dots, \mathbf{h}_n]^T \in \{0,1\}^{n \times C}$ contain the hidden semantic labels for all super-pixels. We use a 1-of-C encoding, and thus a C-dimensional column vector $\mathbf{h}_p \in \{0,1\}^C$, with C denoting the number of semantic classes and h_p^c referring to the c-th entry of the vector $\mathbf{h}_p$.

Our objective is motivated by the fully supervised setting and the success of max-margin classifiers. As the assignments of super-pixels to semantic labels is not known, not even for the training set, supervised learning is not possible. Instead, we take advantage of max-margin clustering (MMC) (Zhao et al., 2008, 2009) which searches for those assignments that maximize the margin. We therefore aim at minimizing the regularized margin violation

$$\frac{1}{2}\mathrm{tr}(W^T W) + \lambda \sum_{p=1}^{n} \sum_{c=1}^{C} \xi(\mathbf{w}_c; \mathbf{x}_p, h_p^c), \tag{4.1}$$

where $W = [\mathbf{w}_1, \dots, \mathbf{w}_C] \in \mathbb{R}^{d \times C}$ is a weight matrix encoding the learned appearance model, $\mathbf{w}_c \in \mathbb{R}^d$ is the c-th column of matrix W, and λ is a hyper-parameter of our framework.

Note that in most semantic segmentation tasks the class categories are distributed according to a power law. Hence, instead of using a standard hinge loss for the margin violation $\xi(\mathbf{w}_c; \mathbf{x}_p, h_p^c)$, we want to take into

account the fact that class labels typically occur in a very unbalanced way. Therefore, we let

$$\xi(\mathbf{w}_c; \mathbf{x}_p, h_p^c) = \begin{cases} \max(0, 1 + (\mathbf{w}_c^T \mathbf{x}_p)), & h_p^c = 0 \\ \mu^c \max(0, 1 - (\mathbf{w}_c^T \mathbf{x}_p)), & h_p^c = 1 \end{cases} \tag{4.2}$$

where $\mu^c = \frac{\sum_{p=1}^{n} 1(h_p^c == 0)}{\sum_{p=1}^{n} 1(h_p^c == 1)}$. If the number of negative examples ($h_p^c = 0$) is bigger than the positive ones, this asymmetric loss penalizes more if we incorrectly label a positive instance. Note that if the matrix of super-pixel class assignments H was known, the cost function given in (4.1) is identical to a one-vs-all support vector machine (SVM).

We now show how to incorporate different forms of weak supervision, i.e., tags, partial labels and bounding boxes. To this end we add constraints to the program given in (4.1). Thus, our learning algorithm reads as follows:

$$\begin{aligned} \min_{W,H} \quad & \frac{1}{2} \mathrm{tr}(W^T W) + \lambda \sum_{p=1}^{n} \xi(W; \mathbf{x}_p, \mathbf{h}_p) \\ \text{s.t.} \quad & H\mathbf{1}_C = \mathbf{1}_n, \quad H \in \{0,1\}^{n \times C}, \quad H \in \mathcal{S}, \end{aligned} \tag{4.3}$$

where $\mathbf{1}_C$ is an all ones vector of length C, and $\mathbf{1}_n$ is an all ones vector of length n. The parameter λ balances the regularization term $\mathrm{tr}(W^T W)$ and the loss contribution $\xi(W; \mathbf{x}_p, \mathbf{h}_p)$.

4.3.2 Incorporating Weak Supervision

Next we describe the constrained space $\mathcal{S}$ for each form of weak annotation.

Image level tags (ILT): Considering image level tags (ILT), each image $i \in \{1, \ldots, m\}$ is assigned a set of categories, indicating which classes are present. However the specific location of the class, i.e., the super-pixel is not specified in any way. Let the binary matrix $Z \in \{0,1\}^{m \times C}$ denote the image-level tag supervision: $Z_{ic} = 1$ if class $c \in \{1, \ldots, C\}$ is present in

image $i \in \{1, \ldots, m\}$, and $Z_{ic} = 0$ otherwise. Let the binary matrix B be a super-pixel-image incidence matrix of size $n \times m$: $B_{pi} = 1$, if super-pixel $p \in \{1, \ldots, n\}$ belongs to image $i \in \{1, \ldots, m\}$, and $B_{pi} = 0$ otherwise. Given the binary matrices B and Z, we incorporate tag-level supervision by adding two sets of constraints. The first set expresses the fact that if a tag is not assigned to an image, super-pixels in that image can not be assigned to that class. This fact is encoded via

$$H \leqslant BZ. \tag{4.4}$$

The second set of constraints encodes the fact that if an image tag is present, at least one super-pixel should take that class as its label. Such a statement is described with

$$B^T H \geqslant Z. \tag{4.5}$$

To explain how these constraints work in practice, let us demonstrate the details using a toy example. Suppose we have $m = 2$ images, each partitioned into 2 super-pixels, i.e., $n = 4$. Let the number of classes of interest $C = 3$. We further assume the first image is tagged with categories $\{1, 2\}$, while the second one is assigned labels $\{2, 3\}$. Then our matrices look as follows:

$$B = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad BZ = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Note that the product BZ 'copies' the image-level tags for super-pixels belonging to that particular image. Due to the less than or equal to constraint and the restriction to a binary matrix H, these super-pixels can not take classes which are not assigned to an image. Similarly, suppose we are given the following class assignment matrix H, then $B^T H$ counts how

many instances are labeled for each class within each image, making sure that at least one super-pixel takes on a required class-label:

$$H = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B^T H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Semi-supervision: In the semi-supervised setting, we are given class labels for only a subset of the super-pixels. This is a useful setting, as users can simply scribble on an image, and label a subset of the super-pixels. This form of supervision is easily framed using an equality constraint for each super-pixel that is labeled, i.e.,

$$H_\Omega = \hat{H}_\Omega, \tag{4.6}$$

where $\Omega \subseteq \{1, \ldots, n\}$ corresponds to the set of annotated super-pixels and the matrix $\hat{H}_\Omega \in \{0, 1\}^{|\Omega| \times C}$ refers to the user-specification.

Bounding boxes: Including the bounding box annotation follows the image level tag case and adds additional restrictions. Note that the ILT setting, is equivalently phrased using a single bounding box of size corresponding to the image dimensions. Several tags are given for this bounding box. Here, we extend this setting to allow for smaller bounding boxes, each of which is assigned a single tag. Following the constraint given in (4.5) we can treat each box as a sub-image and enforce

$$(B^{pos})^T H^{pos} \geqslant Z^{box}, \tag{4.7}$$

where H^{pos} corresponds to label variables for super-pixels entirely inside the bounding boxes that were provided. In addition, $B^{pos}_{p,j} = 1$ if super-pixel p is entirely inside box j, and 0 otherwise. Further, Z^{box} is the

binary label matrix for bounding boxes: $Z^{pos}_{j,c} = 1$, if c is the class of bounding box j and 0 otherwise. Note that $(B^{pos})^{\top} H^{pos} \geqslant Z^{box}$ forces the model to assign at least one super-pixel to the bounding box class c. The matrix H^{neg}_c refers to label variables (for class c) of super-pixels which are partially inside the bounding boxes. A constraint of the form $H^{neg}_c \leqslant 0$ asks that 'negative' super-pixels should not take the bounding box class c. This is typically a reasonable constraint, as we assume our bounding box to be tight. However, due to the fact that our super-pixels suffer from under-segmentation, we do not use negative constraints $H^{neg}_c \leqslant 0$ in our experimental evaluation. To make it robust to under segmentation, in practice we use super-pixels which are 80% inside the bounding boxes to define B^{pos}.

Unlabeled examples: We make use of unlabeled examples by simply incorporating them in the objective. Note that no constraints are added as no supervision is available.

Importantly, for all forms of weak supervision discussed above, the set of constraints subsumed within $\mathcal{S}$ turns out to be linear.

4.3.3 Learning via Alternate Optimization

During learning, we jointly optimize for the feature weight matrix W encoding the appearance model, and the semantic labels H for all n super-pixels as specified by the program given in (4.3). Note that all forms of supervision considered in this chapter can be incorporated via linear constraints. Nonetheless, (4.3) is generally a non-convex mixed integer programing problem, which is challenging to optimize. Investigating the model given in (4.3) more closely, we observe that our optimization problem is however bi-convex, i.e., it is convex w.r.t. W if H is fixed, and convex w.r.t. H if W is fixed. Further, our constraints are linear and they only involve the super-pixel assignment matrix H. For optimization we

Algorithm 1 Learning to Segment

1: **Input:** $X, \mathcal{S}$
2: Initialize: compute $Z(\mathcal{S})$, $B(\mathcal{S})$, $H \leftarrow BZ$;
3: **for** iter$= 1 \rightarrow$ max_iter **do**
4: Fix H solve for W independent of classes (1-vs-all linear SVM);
5: Fix W infer super-pixel labels H in parallel w.r.t images (small LP instances);
6: **end**
7: **return** W, H;

therefore employ an alternating procedure, where we iterate the two steps of optimizing H and W for fixed values of W and H respectively. We refer the reader to Alg. 1 for an outline of the proposed learning algorithm.

It is easy to see that for a fixed class assignment matrix H, the resulting optimization task is equivalent to the fully supervised setting, where the labels are obtained from the current estimate of H. In our formulation, this decomposes into C different 1-vs-all SVMs which can be trained in parallel.

When optimizing w.r.t. the assignment matrix H for a fixed appearance model W, we need to solve a constrained optimization problem where both the objective and the constraints are linear. In addition, H is required to be binary, resulting in an integer linear program (ILP). Such optimization problems are generally NP-hard. However, we will show that in our case we can decompose the problem into smaller tasks that can be optimally solved in parallel via an LP relaxation. This LP relaxation is guaranteed to retrieve an integer solution, and thus an optimal integral point.

Our objective in (4.3) is a min-max function with respect to H. Due to the dependence of μ^c, defined in the loss function given in Eq. 4.2, on the assignment matrix H, this problem is extremely challenging. However we found the solution obtained by simply dropping μ^c during learning to work very well in practice, as shown in the experimental section. In addition it drastically simplifies the loss-augmented inference required

during learning. Solving for the assignment matrix H after dropping μ^c during learning is then equivalent to finding the maximum a-posteriori (MAP) prediction within the label space. To see this, note that picking the class with maximum $\mathbf{w}_c^T \mathbf{x}_p$ yields the smallest hinge-loss ξ. For instance, if $\mathbf{w}_c^T \mathbf{x}_p < 0$, setting $h_p^c = 0$ returns the smaller hinge-loss ξ. Similarly, if $\mathbf{w}_c^T \mathbf{x}_p > 0$, letting $h_p^c = 1$ yields a smaller loss ξ. Thus, we want to pick a super-pixel class assignment H which maximizes the score while remaining feasible:

$$\begin{aligned} \max_{H} \quad & \operatorname{tr}((X^T W)^T H) \\ \text{s.t.} \quad & H\mathbf{1}_C = \mathbf{1}_n, \quad H \in \{0,1\}^{n \times C}, \quad H \in \mathcal{S}. \end{aligned} \tag{4.8}$$

We combine the data into the matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$. The trace objective function is equivalent to a linear sum, hence we can divide it into a sum of super-pixel instances. The constraint $H\mathbf{1}_C = \mathbf{1}_n$ is also separable by super-pixel instances. But the weak supervision (including tags, semi-supervision, bounding boxes) constraints act within a single image. Hence, taking it all together, the entire constraint space is separable by images. This nice property permits to separate the ILP into much smaller sub-programs, which reason about each image independently and can be solved in parallel. We can further reduce the size of the individual ILPs by removing the variables referring to tags not relevant for a particular image.

Importantly, our program has the additional property that the coefficient matrix for the constraints is totally uni-modular. As a consequence we can solve each ILP exactly using a linear programming relaxation which is tight. This tightness is reflected in the following proposition.

Proposition 1. *Relaxing the integrality constraints in (4.8) and using a linear programming solver gives the integral optimal solution for our constraint set $\mathcal{S}$.*

Proof. We start by presenting some preliminaries that are necessary for

the proof. We refer the reader to (Nemhauser and Wolsey, 1988) for more details.

Definition 4.1. *(Nemhauser and Wolsey, 1988) A matrix* A *is totally unimodular (TU), iff the determinants of all square submatrices of* A *are either* -1, 0, *or* 1.

Theorem 4.2. *(Nemhauser and Wolsey, 1988) A* $(0, +1, -1)$ *matrix* A *is totally unimodular if both of the following conditions are satisfied:*

- *Each column contains at most two nonzero elements*
- *The rows of* A *can be partitioned into two sets* A_1 *and* A_2 *such that two nonzero entries in a column are in the same set of rows if they have different signs and in different sets of rows if they have the same sign.*

Corollary 4.3. *(Nemhauser and Wolsey, 1988) A* $(0, +1, -1)$ *matrix* A *is totally unimodular if it contains no more than one* $+1$ *and no more than one* -1 *in each column.*

Theorem 4.4. *(Truemper, 1978; Grady, 2010; Grady and Polimeni, 2010) If* A *is totally unimodular and* b *is integral, then solving linear programs of form* $\{\min \mathbf{c}^{\mathsf{T}}\mathbf{x} \mid \mathrm{A}\mathbf{x} = \mathbf{B}, \mathbf{x} \geqslant 0\}$ *have integral optima, for any* $\mathbf{c}$.

The main idea of our proof is to show that the matrix describing our linear constraints is totally unimodular. Employing Theorem 4.4 we then know that the LP relaxation gives integral optima since the right hand side is integral in our optimization problem. Given that our inference problem is fully decomposable with respect to images, we first decompose it into

small LPs, one for each image. More formally, for each image i, we have,

$$\begin{aligned} \max_{H^i} \quad & \operatorname{tr}\left((X^i W)^\top H^i\right) \\ \text{s.t.} \quad & H^i \mathbf{1}_C = \mathbf{1}_{n^i} \\ & B'^\top H^i \geqslant \mathbf{z}'^i \\ & 0 \leqslant H^i \leqslant B^i \mathbf{z}^i, \end{aligned} \tag{4.9}$$

where n^i is the number of super-pixels in image i and H^i is a binary label matrix for n^i super-pixels in image i. $B'^\top H^i \geqslant \mathbf{z}'^i$ are the constraints from both the bounding boxes as well as tags. Note that for the semi-supervised case we remove the labeled super-pixels in the above LP. Additionally, the corresponding row c in $B'^\top H^i \geqslant \mathbf{z}'^i$ is already satisfied (one instance is labeled) for class c by default, hence it is no longer required to be considered.

Given $0 \leqslant H^i \leqslant B^i \mathbf{z}^i$, those not-tagged classes will be filtered out in the final solution, i.e., $H^i_{pc} = 0$ if $z^i_c = 0$. We remove these classes and obtain the following LP:

$$\begin{aligned} \max_{H^i} \quad & \operatorname{tr}\left((X^i W_{C'})^\top H^i_{C'}\right) \\ \text{s.t.} \quad & H^i_{C'} \mathbf{1}_{C'} = \mathbf{1}_{n^i} \\ & \mathbf{1}^\top_{n^i} H^i_{C'} \geqslant \mathbf{1}_{C'} \\ & H^i_{C'} \geqslant 0, \end{aligned} \tag{4.10}$$

where C' is the number of potential classes for the super-pixels in image i.

Next, let us rephrase our LP into the canonical form. We vectorize $H^i_{C'}$

by stacking each row into $\mathbf{x} \in \{0,1\}^{n^i C' \times 1}$: $\mathbf{x} = [\mathbf{h}_1; \mathbf{h}_2; \cdots; \mathbf{h}_{n^i}]$ to obtain

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A_1 \mathbf{x} = \mathbf{1}_{n^i} \\ & A_2 \mathbf{x} \geqslant \mathbf{1}_{C'} \\ & \mathbf{x} \geqslant 0, \end{aligned} \tag{4.11}$$

with

$$A_1 = I_{C'} \otimes \mathbf{1}_{n^i}^T = \begin{bmatrix} 1 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ & & & & & & \cdots & & & \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 1 \end{bmatrix},$$

$$A_2 = \mathbf{1}_{C'} \otimes I_{n^i} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & \end{bmatrix}.$$

Following Corollary 4.3, A_1, A_2 are both total unimodular. Next we introduce slack variables $\mathbf{y}$ to rephrase our LP (4.11) into the following form,

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A_1 \mathbf{x} = \mathbf{1}_{n^i} \\ & A_2 \mathbf{x} - \mathbf{y} = \mathbf{1}_{C'} \\ & \mathbf{x}, \mathbf{y} \geqslant 0. \end{aligned} \tag{4.12}$$

Consequently the coefficient matrix reads as

$$A = \begin{bmatrix} A_1 & 0 \\ A_2 & -I \end{bmatrix}. \tag{4.13}$$

It is obvious that A has at most two non-zero entries in every column. Further, if we partition matrix A into $\begin{bmatrix} A_1 & 0 \end{bmatrix}$ and $\begin{bmatrix} A_2 & -I \end{bmatrix}$, two nonzero entries are in different sets of rows if they have the same sign. Hence, A is total unimodular by Theorem 4.2. Finally, employing Theorem 4.4, the LP relaxation gives integral optima. This concludes our proof.

□

Note that this result only implies that the second step (4.8) in the alternating algorithm is solved to an integral optimal, but not the original full model in (4.3).

4.3.4 Inference

We experiment with two inference strategies. Given a learned appearance model W, our first strategy predicts using the standard 1-vs-all rule. We refer to this setting as "Ours (1-vs-all)." Our second strategy makes use of a tag predictor to create additional constraints for the test images. Give the tag predictions we perform inference on the test set by minimizing (4.8) with the ILT constraints described in (4.4). Note that we do not employ the constraints provided in (4.5) as our tag classifier might be wrong. We refer to this setting as "Ours (ILT)."

4.3.5 Transductive Learning

In the standard setting, we learn the weights of the appearance model matrix W using the training set images. We also experiment with a transductive setting (Zhu, 2005) which exploits the test images as well. Note that the test set can be used by incorporating the images as unlabeled examples or by using a tag classifier and adding the constraints detailed in (4.4) for the test images. We refer to this setting as "Ours (1-vs-all + transductive)" and "Ours (ILT+transductive)" respectively.

Method	Supervision	per-class	per-pixel
(Liu et al., 2011)	full	24	76.7
(Farabet et al., 2012)	full	29.5	78.5
(Farabet et al., 2012) balanced	full	46.0	74.2
(Eigen and Fergus, 2012)	full	32.5	77.1
(Singh and Kosecka, 2013)	full	33.8	79.2
(Tighe and Lazebnik, 2013b)	full	30.1	77.0
(Tighe and Lazebnik, 2014)	full	39.3	78.6
(Yang et al., 2014)	full	48.7	79.8
(Vezhnevets et al., 2011)	weak (tags)	14	N/A
(Vezhnevets et al., 2012)	weak (tags)	22	51
(Rubinstein et al., 2012)	weak (tags)	29.5	63.3
(Xu et al., 2014)	weak (tags)	27.9	N/A
Ours (1-vs-all)	weak (tags)	**32.0**	**64.4**
Ours (ILT)	weak (tags)	**35.0**	**65.0**
Ours (1-vs-all + transductive)	weak (tags)	**40.0**	59.0
Ours (ILT + transductive)	weak (tags)	**41.4**	**62.7**

Table 4.1: Per-class and per-pixel accuracy comparison to state-of-the-art on the SIFT-flow dataset. For (Vezhnevets et al., 2012), we report the per-pixel number from (Vezhnevets, 2012). Note that our approach while only using tags as supervision and thus never observing a single pixel labeled, is able to perform almost as well as the state-of-the-art in the fully supervised setting. Furthermore, we outperform the state-of-the-art in the weakly label case by more than 10%.

4.4 Experimental Evaluation

4.4.1 Dataset and Super-pixel/feature Extraction

Dataset: To illustrate the performance of our model, we conduct a rigorous evaluation on the Siftflow data set (Liu et al., 2011), which has been widely studied (Liu et al., 2011; Farabet et al., 2012; Eigen and Fergus, 2012; Singh and Kosecka, 2013; Tighe and Lazebnik, 2013b, 2014; Yang et al., 2014; Vezhnevets et al., 2011, 2012; Xu et al., 2014). The Sift-Flow data

Method	Supervision	per-class	per-pixel
(Shotton et al., 2008)	full	67	72
(Yao et al., 2012)	full	79	86
(Vezhnevets et al., 2011)	weak (tags)	67	67
(Liu et al., 2012)	weak (tags)	N/A	**71**
Ours (ILT + transductive)	weak (tags)	**73**	70

Table 4.2: Per-class and per-pixel accuracy comparison to state-of-the-art on the MSRC dataset.

contains $m = 2688$ images and $C = 33$ classes in total. The data set is very challenging due to its large scale and its heavily tailed class distribution. A few 'stuff' classes like 'sky,' 'road,' 'sea,' and 'tree' are very common, while the 'things' classes like 'sun,' 'person,' and 'bus' are very rare. We use the standard split of 2488 training images and 200 testing images provided by (Liu et al., 2011), and randomly sampled 20% of the training set to tune our parameter λ. To further evaluate the capacity of our algorithm, we also test it on the MSRC dataset (Shotton et al., 2008), which has $m = 591$ images and $C = 21$ classes. We report both the per-class and per-pixel accuracy.

Super-pixel segmentation: For each image, we compute the Ultrametric Contour Map (UCM) (Arbelaez et al., 2014), and threshold it at 0.4 to extract super-pixels. UCM produces a few tiny super-pixels, which create noise during learning. To alleviate this issue we adopt a local search algorithm (Rantalankila et al., 2014) which merges similar adjacent tiny super-pixels. On average, this procedure results in 14 regions per image. We use the same super-pixel segmentation for all the reported experiments.

Super-pixel feature extraction: For each super-pixel, we first extract R-CNN (Girshick et al., 2014) features within the bounding box as well as within the masked box. These two sets of features – 8192 dimensional

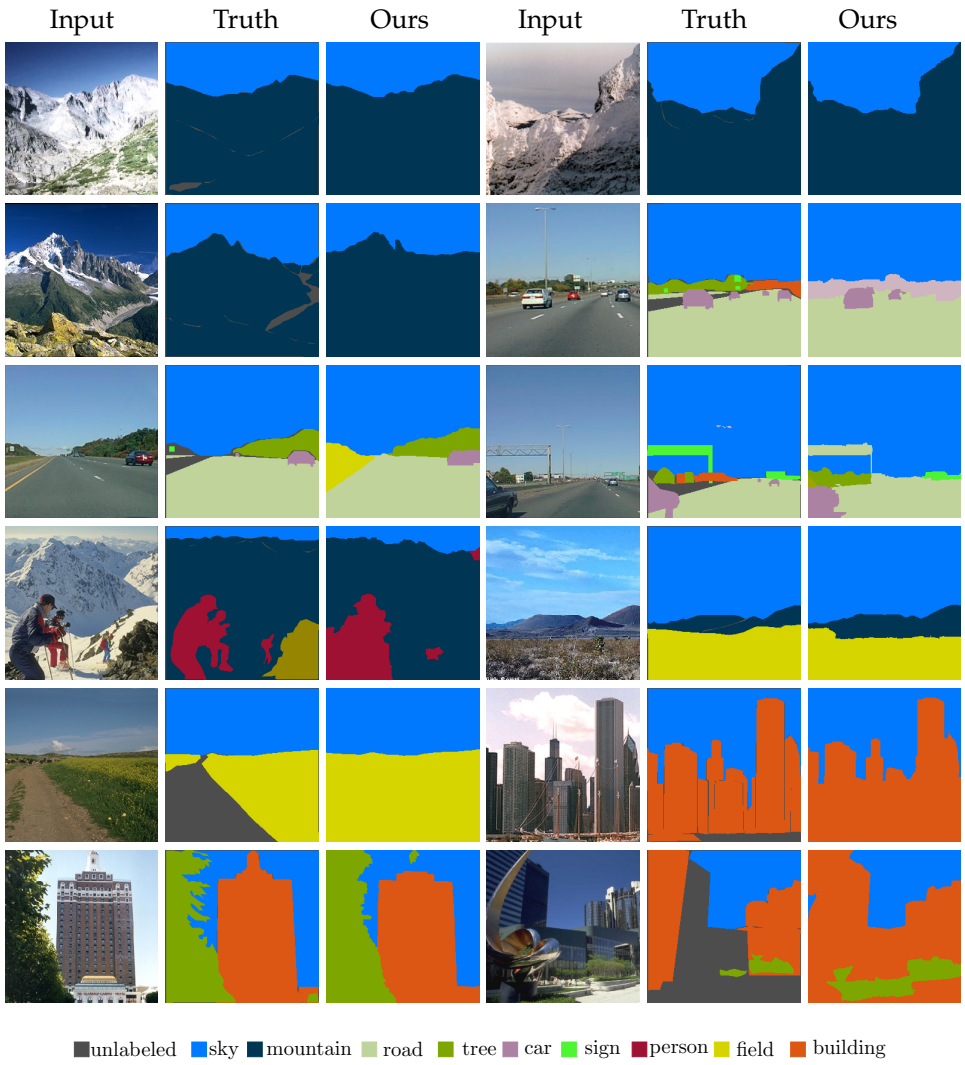

Figure 4.2: Sample results from "Ours(ILT+transductive)". Note gray regions in the second and fifth column are not labeled in ground truth. **Best viewed in color.**

in total – capture the local context and the shape of the super-pixel. To take into account global context and super-pixel size/location, we further replace the bounding box with the whole image to compute additional features. That is, we compute R-CNN features for the whole image, as

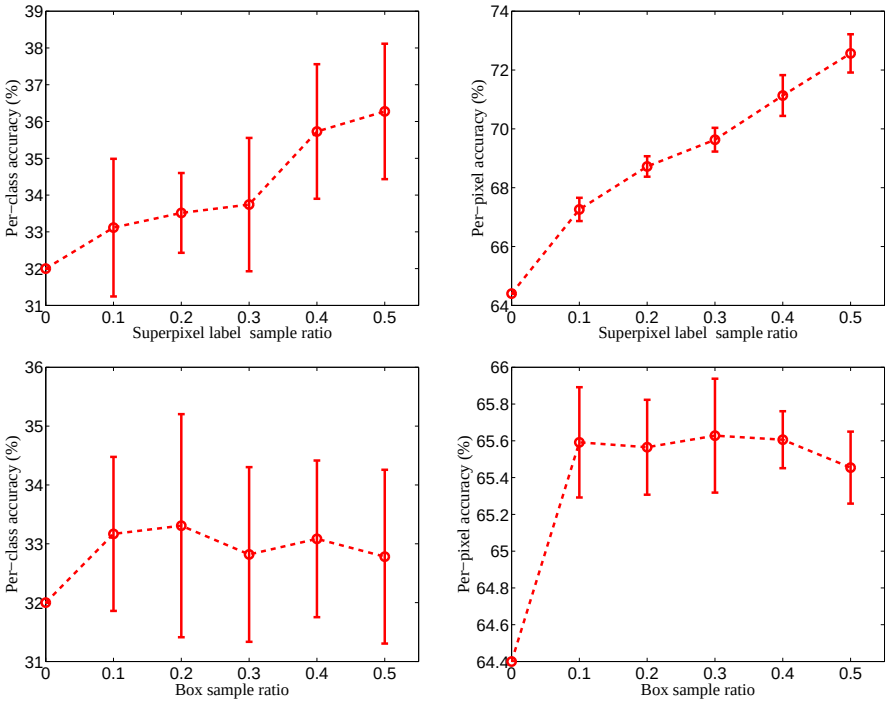

Figure 4.3: Per-class (first column) and per-pixel accuracy (second column) with respect to super-pixel label (first row) and bounding box (last row) sample ratio.

well as the masked image. This gives another 8192 dimensional feature vector. After concatenation, we obtained a $d = 16884$ dimensional feature vector $\mathbf{x}_p$ for each super-pixel p.

4.4.2 Evaluation on Various Weak Superpvisons

Training with tags only: We first evaluate our algorithm on the standard weakly supervised semantic segmentation setting. During training, we are only given image level tags, and at test time, we infer pixel-wise labels without tags. We first investigate the feature weights we learned using Alg. 1 via the 1-vs-all inference approach. For each super-pixel, we simply

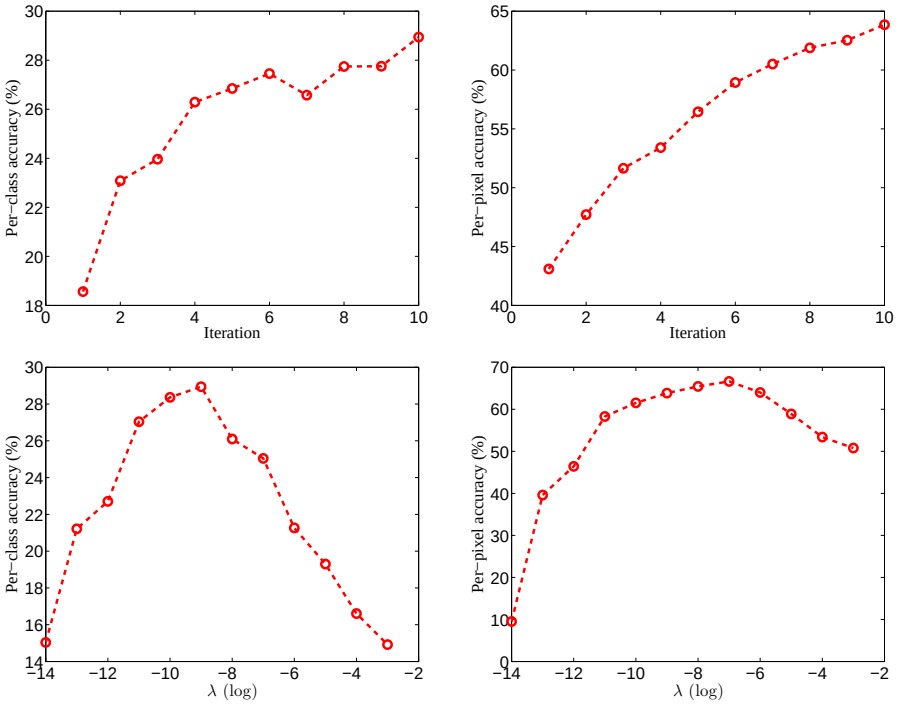

Figure 4.4: Per-class (first column) and per-pixel accuracy (second column) from "Ours(1-vs-all)" with respect to number of iterations (first row) and λ (second row). The accuracy is reported on the validation set.

pick the class with maximum potential from $\mathbf{x}^{\top}W$. As shown in Tab. 4.1, this simple approach tagged "Ours(1-vs-all)" achieves 32.0% per-class accuracy and a 64.4% per-pixel accuracy, outperforming the state-of-the-art. Motivated by recent work (Vezhnevets et al., 2011; Xu et al., 2014), we trained a 1-vs-all linear SVM ILT classifier with R-CNN features extracted from the whole image. We then feed the classifier output into our inference detailed in Eq. 4.8, and predict the labels for super-pixels for the testing images. We refer to this setting via "Ours (ILT)." It further improves the per-class accuracy to 35.0%, and the per-pixel metric to 65.0%.

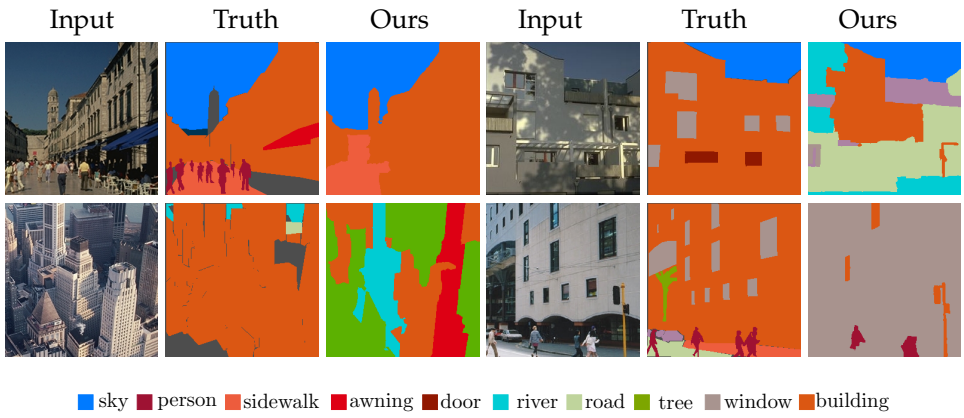

Figure 4.5: Failure cases. **Best viewed in color.**

Training with tags (transductive): As provided in Tab. 4.1, using the transductive setting we achieved a 41.4% per-class accuracy, which outperforms the state-of-the-art by 11.9%. We also note that using this transductive setting without the tag classifier achieves 40.0% per-class accuracy, which further demonstrates that ILT are very helpful in inferring pixel-wise labels. As shown in Tab. 4.2, the resulting performance on MSRC in per-class/per-pixel accuracy is 73%/70%. In contrast (Vezhnevets et al., 2011) reports 67%/67%. We present qualitative results in Fig. 4.2. The presented approach performs well for 'stuff' segments like 'sky,' 'mountain,' and 'road' which have a fairly reliable super-pixel segmentation and a discriminative appearance model. We are also able to obtain correct labels for 'things' classes (e.g., 'cars' and 'person').

Training with semi-supervision: We next evaluate the semi-supervised setting where a subset of the super-pixels is labeled in addition. We focus on the 1-vs-all inference setting. Strokes are simulated by randomly labeling superpixes from the ground truth using a sampling ratio of $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Due to randomness, we repeat our experiments 10 times and report the mean and standard deviation in the first two plots

of Fig. 4.3. Note that both the per-class and the per-pixel accuracies improve consistently when more super-pixels are observed. Furthermore, the per-pixel accuracy increases almost linearly with the sampling ratio.

Training with bounding boxes: Next we evaluate the bounding box setting. Again we focus on the 1-vs-all inference setting. We consider bounding boxes for 'things' classes: {bird, boat, bus, car, cow, door, moon, person, pole, sign, streetlight, sun, window} and simulate this setting by randomly picking bounding boxes around connected segments from ground truth annotations. Due to the randomness we repeat our experiments 10 times and report the mean and standard deviation in the last two plots of Fig. 4.3. We improve the per-class accuracy by 1% even with only 10% of 'things' boxes, which translates to approximately 0.05 boxes per image. We also observe the improvement is less significant, when more boxes are added. This is due to the quality of our super pixels, which suffer from under segmentation.

4.4.3 Model Behavior

Number of iterations: During training, we iterate between learning the feature weights and inferring super-pixel labels. To study how the performance changes with respect to the number of iterations, we evaluate the weights learned after each iteration using the 1-vs-all rule, and plot the accuracy on the validation set in Fig. 4.4. We observe that the per-class/per-pixel accuracy quickly improves for the first 4 iterations, and starts to converge after 8 iterations. In all our experiments, we report the performance after 10 iterations.

Performance with respect to λ: To evaluate how our algorithm behaves w.r.t. λ, we plot the accuracy of "Ours(1-vs-all)" for all λ in $\{2^{-14}, ..., 2^{-3}\}$ in Fig. 4.4. We used a weighted sum of the per-pixel and per-class accuracy

to find the best lambda on the validation set. All experiments use this fixed value of $\lambda = 2^{-9}$.

Running time: As we discussed in the optimization, both tasks of learning W and inferring H can be parallelized. On our machine with 12 threads, each iteration takes about $1 \sim 2$ minutes, resulting in less than 20 minutes for training on the full Siftfow dataset. Inference takes < 0.01s per image after super-pixel segmentation and feature extraction.

Failure cases: We present failure cases in Fig. 4.5. Super-pixel under segmentation is a common failure mode, where small 'things' segments (top left in Fig. 4.5) are challenging to obtain by UCM. Extreme shading changes (top right in Fig. 4.5) pose challenges just like cluttered textures (bottom left in Fig. 4.5). As shown on the right hand side of Fig. 4.5 our model may also get confuse classes that co-occur frequently. For instance, we accidentally labeled building segments as window, and vice versa. This is expected as only tags are used for learning.

4.5 Summary

In this chapter, we introduced a unified semantic segmentation approach to handle weak supervision in the form of tags, partial labels and bounding boxes. Our approach is efficient in both training and testing. We demonstrated the effectiveness of our approach on the challenging Siftflow dataset and show that the presented method outperforms the state-of-the-art by more than 10%. Our method provides a natural way to make use of readily available weak labeled data at a large scale, and hence offers a potential to build a base of visual knowledge (Chen et al., 2013) using for example data from the Internet.

5 ONLINE FOREGROUND AND BACKGROUND VIDEO SEGMENTATION

This chapter extends the scope of our visual parsing methods from images to videos. The parsing goal here is to segment out foreground objects from background in videos. Observing the background in videos is usually stable and does not change dramatically, we model the background as a common subspace shared across the video frames. Meanwhile, leveraging contiguity information of the foreground tends to make the updates of the background subspace even more accurate. Our main technical contribution in this chapter is to show that the corresponding online estimation of the background layer procedure while assuring that the residual foreground has good spatial support can be formulated as an approximate optimization process on a Grassmannian manifold. We propose an efficient numerical solution, GOSUS, Grassmannian Online Subspace Updates with Structured-sparsity, for the online foreground/background video segmentation problem. GOSUS is expressive enough in modeling both homogeneous perturbations of the subspace and structural contiguities of outliers, and solvable via an optimization scheme known as alternating direction method of multipliers (ADMM). In our preliminary work (Xu et al., 2013b), we perform an empirical evaluation of this algorithm on two problems of interest: online background/foreground estimation and online multiple face tracking, and demonstrate that it achieves superior performance relative to the state-of-the-art; further, in near real time.

5.1 Problem Description

The problem we want to tackle is online foreground and background spatio-temporal segmentation in videos. In spite of the success of image parsing, the sheer size of video data creates unique computational and

efficiency issues for video parsing. Nonetheless, video segmentation may be considered to be easier in a certain sense as the temporal sequence provides unique information about motion. Take the video in Fig. 5.1 as an example. While the background clutter may not remain constant throughout the full video sequence, the changes from one frame to the other are not abrupt. What this means is that the background can be modeled as a common subspace shared across the video frames, a strategy deployed in many purpose in computer vision, especially in the batch setting (De La Torre and Black, 2003; Lin et al., 2010; Wang et al., 2012b). We also see that foreground of interest is almost never random — instead, these are spatially contiguous and structured regions corresponding to people, objects and landmarks. Leveraging such contiguity information as a covariate should make the foreground segmentation much more accurate. Our key idea is to employ subspace learning for adapting to an evolving background, and segment out foreground as structured spatially contiguous regions.

Subspace learning methods have been extensively studied in computer vision with applications spanning motion analysis, clustering, background estimation, and deriving semantic representations of scenes (De La Torre and Black, 2003; Cai et al., 2007; Bucak and Gunsel, 2009; Favaro et al., 2011). Within the last few years, new developments in matrix factorization (Wang et al., 2012b; Arora et al., 2012) and sparse modeling (Mairal et al., 2010; Yu et al., 2011) have led to significant renewed interest in this construct, and has provided a suite of new models and optimization schemes for many variants of the problem. These methods have shown very promising performance on video segmentation, with the key assumption that background does not change rapidly, and can be modeled using a low dimension subspace. An interesting version that several authors have proposed recently is *Online Subspace Learning* (Wang et al., 2008; Balzano et al., 2010; He et al., 2012). Here, observations are presented sequentially, in the

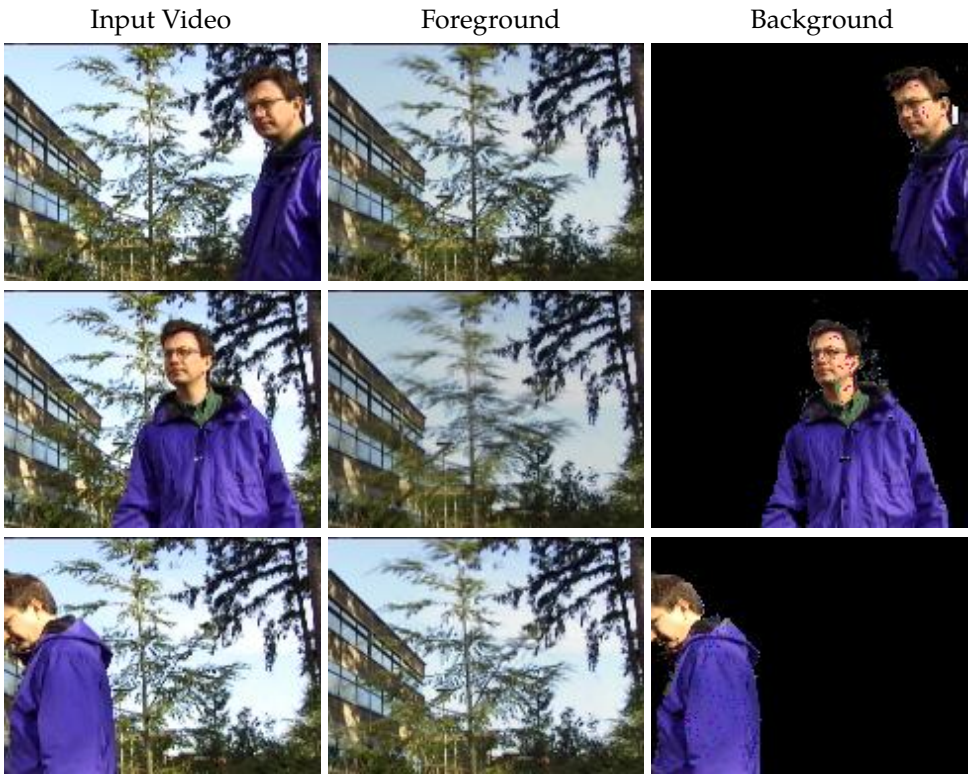

Figure 5.1: Example of foreground and background video segmentation: given a video (first column), we segment out the foreground (third column) out as structured sparsity while recovering the background (second column) using online subspace learning.

form of an unknown mixture of the primary subspace(s) plus a residual component. The objective is to keep an estimate of the contributing subspace(s) updated as the observations continually present themselves. This is a natural setting for video segmentation and parsing, as video frames are collected in a sequential manner. More importantly, online processing may even bring more efficiency, as each time, we only need to process one frame, instead of the whole video volume.

The standard strategy of modeling the foregoing online estimation question is to assume that the observation (i.e., video) is an unknown

mixture of two components: background and foreground. The first relates to the *subspace* terms comprising one or multiple subspaces (and with or without regularization). Statistically, one may regard this term as the *main effect* (i.e., background) which explains *most* of the measurement. But fitting the signal to high fidelity will necessarily involve a large degree of freedom in the subspace term, and so the model allows for a small amount of compensatory residual error — this corresponds to the second term contributing to the observed signal. To encourage the residual quantity to be small, most proposals impose a sparsity penalty on its norm (Lin et al., 2010; He et al., 2012). Therefore, the main technical concern, both in the "batch" and online settings, is to efficiently estimate the subspace and if possible provide probabilistic guarantees of correct recovery.

Within the last year, a particularly relevant application of online subspace learning is in the context of keeping updated estimates of background and foreground layers for video data (as shown in Fig. 5.1). Here, one exploits concepts from matrix completion for subspace estimation, by drawing i.i.d. samples from each incoming frame, and adjusting the current subspace parameters using *only* the sub-sampled data (He et al., 2012). The mass of the signal outside the support of the subspace may then be labeled as foreground. This strategy works quite well when the background is completely static: essentially, the model has seen several hundred frames and has converged to a good estimate already. However, for more challenging video parsing settings, such as when there are small but continual variations in the background (e.g., a swaying tree) and/or it is undergoing changes due to camera motion, zoom or illumination differences, it takes time for the subspace estimates to stabilize. Here, the residual must then compensate for a less than ideal estimate of the main effect, which leads to salt-pepper isolated foreground regions, scattered over the image. One reason for this unsatisfactory behavior is that the model does not enforce spatial homogeneity in the foreground region.

Imposing 'structure' on the secondary term, such as asking for contiguity, has the highly beneficial effect that the residual serves a more important role than merely accounting for the error/corruption. From a statistical modeling perspective, the residual structure acts as a regularizer that improves the estimate of the main effect (the background reconstruction via subspace modeling). Consequently, in the background/foreground setting, we see that the estimated foreground regions are far more meaningful. The resultant improvements in performance are quite significant, compared to alternative approaches. For several other interesting visual parsing applications which we discuss later in the chapter, the benefits are clear, though the notion of structure (i.e., structured sparsity operator) is different and better reflects the needs of that domain.

This chapter. Consider a regression model, $Y = f(W) + \epsilon$. If the distributional properties of the second term is known (e.g., Rician, Poisson), it must improve the estimation of $f(\cdot)$. We seek to translate this simple idea to the problem of Online Subspace Learning, by incorporating structure (i.e., via a group norm) on the secondary term. The **key** contributions of this chapter are: **1)** Show how group sparsity based structural homogeneity can be incorporated within estimation problems defined on Grassmannian manifolds; **2)** Present an efficient online optimization scheme where most constituent steps reduce to simple matrix operations; **3)** Demonstrate for two example applications (online background subtraction and online multiple face tracking) using a variety of datasets, that the method gives competitive empirical performance in near real time.

5.2 Related Work

Subspace learning, and more generally, learning low dimensional multilinear models has a long and rich history in Computer Vision. The contemporary suite of algorithms for this problem may be classified into a

few separate categories, which nonetheless share important similarities. Models inspired from dimensionality reduction techniques build upon the traditional principal component analysis (PCA) framework. For instance, *Robust subspace learning* (De La Torre and Black, 2003; Favaro et al., 2011) and *Generalized Principal Component Analysis* (GPCA) (Vidal et al., 2005) take a hybrid geometric/statistical view of separating heterogeneous 'mixed' data drawn from one or more subspaces. Building upon classical approaches based on factor analysis, independent component analysis (ICA) and its variants (Li et al., 2005) parameterize the subspace as a combination of a small set of sources (Hyvärinen and Oja, 2000), and work well for subspace estimation applications such as action recognition (Le et al., 2011), segmentation (Mukherjee et al., 2012) and facial pose analysis (Li et al., 2005). More recently, theory from compressive sensing (also, matrix completion) (Candès and Recht, 2009), and matrix factorization (Arora et al., 2012) have been successfully translated into new models and optimization schemes for this problem. An important representative from this group, which has found a multitude of vision applications, is Robust Principal Components Analysis (RPCA) which expresses the measurement as a combination of a low rank matrix and a ℓ_1-regularized noise component (Lin et al., 2010; Candès et al., 2011). Separately, several authors express subspace estimation as a non-negative matrix factorization (NMF) (Wang et al., 2012b; Bucak and Gunsel, 2009; Arora et al., 2012) and give rigorous recovery guarantees. While the literature devoted to the batch setting above is interesting, there is also some research activity in vision, especially in the last two years, focused on the *online* version of this problem. This has led to a set of powerful online subspace learning methods (Wang et al., 2008; Balzano et al., 2010; He et al., 2012), which are related to the above ideas as well as a parallel body of work in manifold learning (Hamm and Lee, 2008; Turaga et al., 2008) — they leverage the fact that the to-be-estimated signal lies on a Grassmannian (Turaga et al.,

2008). In particular, GROUSE (Balzano et al., 2010) and GRASTA (He et al., 2012) (an online variant of RPCA) show how the subspace updates can be accurately maintained in real time by using sub-sampling ideas. Our framework leverages this body of work, and we will point out similarities to known results in the presentation that follows.

5.3 Grassmannian Online Subspace Updates with Structured-sparsity

Notations. We denote matrices by non–bold upper case letters (e.g., V), vectors by bold lower case (e.g., $\mathbf{x}$) and scalars by non-bold lower case letters (e.g., μ). Subscripts and superscripts will denote frame numbers, iterations, indices, etc., which will be explained as needed.

This section describes the various sub-components that make up the main model studied in this chapter. The data V is a composition of a main effect (or signal) B and a secondary term (or outlier) X. That is, $V = B + X$ where $V, B, X \in \mathbb{R}^{n \times m}$, n is the data dimensionality and m is the number of observations. The signal B is given as a linear combination of d sources (subspace basis) in n dimensions, denoted by $U = [\mathbf{u}_d]$. This assumption is reasonable since the variation in signal across consecutive frames is small enough that it allows the few ($d \ll n$) degrees of freedom to recover most changes. The orthogonal structure of U implies that it lies on a Grassmannian manifold $\mathcal{G}_{n,d}$ embedded in a n-dimensional Euclidean space. Let the coefficient matrix be W. In the absence of any error, we have $B = UW$. Now, if $\mathbf{v} \in \mathbb{R}^n$ is an observation and $\mathbf{x} \in \mathbb{R}^n$ is the corresponding outlier vector (lies outside the support of the subspace given by U), then $\mathbf{v} = U\mathbf{w} + \mathbf{x}$, where $\mathbf{w}$ is the coefficients vector for the current observation. This expression is under constrained when both the signal and the outlier are unknown. To drive the estimation procedure, we impose a regularization constraint expressing what constitutes a 'good' outlier,

for instance, contiguity. That is, we may ask that the outlier be spatial coherent ensuring that isolated detections scattered across the image are strongly discouraged. The implicit expectation is that this makes $\mathbf{x}$ more meaningful in the context of the application, and so usefully biases the estimation of the subspace. We elaborate on the notion of structure next.

5.3.1 Structured sparsity

For the background estimation example, the texture/color of the foreground objects (i.e., outliers) is homogeneous and so the outliers should be contiguous in an image. For multiple face tracking (which we elaborate later), we need to *track* a set of faces in a given video where the subspace constitutes the faces themselves. But the outliers created by occlusions are *not* pixel sparse, instead, constitute contiguous regions distributed at different face positions (Jia et al., 2012). As an example, consider a person wearing sunglasses or if a shadow or irregular illumination is distorting a part of the face. We do not want such occlusions to cause large changes in the online updates and destroy the notion of a face subspace. Instead, we must allow the $\mathbf{x}$ term to subsume and accommodate such structured deviations from a 'face' subspace.

To formalize this prior on the outlier, we use structured (or group) sparsity (Yuan and Lin, 2006; Huang et al., 2009; Huang and Zhang, 2010). For one image frame, the groups may correspond to sets of sliding windows on the image, super-pixels generated via a pre-processing method (which encourages perceptually meaningful groups), or potential face sub-regions (as shown in Fig. 5.2). A $n \times n$ (n is the dimensionality of each observation) diagonal matrix D^i is used to denote a "group" i. Each diagonal element of D^i corresponds to the presence/absence of a pixel in the i^{th} group, as

$$D^i_{jj} = \begin{cases} 1 & \text{if pixel } j \text{ is in group } i; \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

1	1	
1	1 2	2
	2	2

Figure 5.2: Examples of structures in visual data. The first column shows the spatial contiguity of objects, the second column shows a structure between landmark regions, and the third column shows a toy example of overlapping groups 1 and 2.

where D^i_{jj} is the j^{th} diagonal element of D^i. A penalty function is then defined as,

$$h(\mathbf{x}) = \sum_{i=1}^{l} \mu_i \|D^i \mathbf{x}\| \tag{5.2}$$

where μ_i gives the weight for group i and l is the number of such groups. D^i is sparse and allows overlap with other D^js ($i \neq j$), so that we can form groups from overlapping homogeneous regions (groups may also be disjoint, if desired). Our group sparsity function $h(\cdot)$ in (5.2) has a mixed norm structure. The inner norm is either l_2 or l_∞ (we use l_2) forcing pixels in the corresponding group to be similarly weighted, and the outer norm is l_1 which encourages sparsity (i.e., only few groups are selected). In general, the design of D^is depends on the needs of the application. We will give specific examples shortly.

5.3.2 Model

With these components in hand, we can now present our main model. Given an input data $V \in \mathbb{R}^{n \times m}$, our model estimates the subspace matrix U, the coefficient vector $\mathbf{w}$, and the outlier $\mathbf{x}$, at a given time point (where $\mathbf{v}$ denotes the given current observation) by the following minimization,

(λ is a positive regularization parameter)

$$\min_{U^T U = I_d, \mathbf{w}, \mathbf{x}} \quad \sum_{i=1}^{l} \mu_i \|D^i \mathbf{x}\|_2 + \frac{\lambda}{2} \|U\mathbf{w} + \mathbf{x} - \mathbf{v}\|_2^2 \tag{5.3}$$

5.4 Online Optimization

While model (5.3) faithfully models our requirements, optimizing it can be challenging. This is due to the non-smoothness of the mixed norm and non-convexity arising due to the orthogonal structure of U. In fact, several recent papers (Mairal et al., 2011; Chen et al., 2011b; Qin and Goldfarb, 2012) are devoted to ideas for optimizing the structured sparsity norm objectives *alone*, and even by itself, it gets complicated due to overlapping groups. Specifically, one may require the design of specialized proximal operators, and the running time of many existing schemes ($\sim$ 30 minutes, (Qin and Goldfarb, 2012)) is impractical for problem sizes encountered in our application.

Observe that at any given time point, the model has already processed many frames before it, and has obtained a reasonable estimate of U. Because the changes in U are not drastic from one frame to the other, local updates of the variables in (5.3) are sufficient in practice. This is a compromise since obtaining a global optimum for the nonconvex U is unlikely anyway. We adopt a block-wise approach which solves for a subset of variables keeping the others fixed. In particular, we observe (5.3) is convex for $(\mathbf{w}, \mathbf{x})$ when U is fixed (denoted as U^*), which can be computed efficiently. A sequential update scheme (Nocedal and Wright, 2006) is used when optimizing for U, while still preserving its orthogonality. Below, we give a detailed analysis of these sub-procedures and outline methods to optimize each component and the overall model.

5.4.1 Solve for tuple $(\mathbf{w}, \mathbf{x})$ at fixed U^*

As $\mathbf{x}$ is shared across the two terms in the objective in (5.3), we introduce a set of slack variables $\{\mathbf{z}^i\}$ for each $\mathbf{D}^i\mathbf{x}$. This gives the following sub-problem

$$\begin{aligned} \min_{\mathbf{w},\mathbf{x}} \quad & \sum_{i=1}^{l} \mu_i \|\mathbf{z}^i\|_2 + \frac{\lambda}{2}\|U^*\mathbf{w} + \mathbf{x} - \mathbf{v}\|_2^2 \\ \text{s.t.} \quad & \mathbf{z}^i = D^i\mathbf{x}, \quad i = 1, \cdots, l. \end{aligned} \tag{5.4}$$

Model (5.13) is convex over $\{\mathbf{z}^i\}$ and $(\mathbf{w}, \mathbf{x})$, while the constraints are affine. A natural choice to solve such a problem efficiently is the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011), assuming we can show that each resultant sub-calculations can be performed cheaply. Next, we demonstrate that this is indeed the case here.

The augmented Lagrangian (Nocedal and Wright, 2006) of (5.13) is given by

$$\begin{aligned} \mathcal{L}(\mathbf{w}, \mathbf{x}, \{\mathbf{z}^i\}, \{\mathbf{y}^i\}) = & \sum_{i=1}^{l} \mu_i \|\mathbf{z}^i\|_2 + \frac{\lambda}{2}\|U^*\mathbf{w} + \mathbf{x} - \mathbf{v}\|_2^2 \\ & + \sum_{i=1}^{l} \mathbf{y}^{i\,T}(D^i\mathbf{x} - \mathbf{z}^i) + \sum_{i=1}^{l} \frac{\rho_i}{2}\|D^i\mathbf{x} - \mathbf{z}^i\|_2^2 \end{aligned} \tag{5.5}$$

Here ρ^i are predefined positive parameters, and $\mathbf{y}^i$ are the dual variables associated with the constraints. Our update scheme proceeds as follows. Given the current observation $\mathbf{v}$ and the tuple $(\mathbf{w}_k, \mathbf{x}_k, \{\mathbf{z}_k^i\}, \{\mathbf{y}_k^i\})$ at k^{th} iteration, the step-by-step updating of the tuple at $(k+1)^{th}$ iteration is:
(w,x)-minimization: To minimize (5.5) with respect to (**w**,**x**) alone, keep-

ing all the other parameters fixed, we have

$$\begin{aligned}\min_{\mathbf{w},\mathbf{x}} \quad & \frac{\lambda}{2}\|U^*\mathbf{w}+\mathbf{x}-\mathbf{v}\|_2^2+\sum_{i=1}^{l}{\mathbf{y}_k^i}^T D^i\mathbf{x} \\ & +\sum_{i=1}^{l}\frac{\rho_i}{2}\|D^i\mathbf{x}-\mathbf{z}_k^i\|_2^2\end{aligned} \tag{5.6}$$

(5.6) takes the form of a convex quadratic problem in $(\mathbf{w},\mathbf{x})$ and the closed form solution comes from the linear system, $A\begin{bmatrix}\mathbf{w} & \mathbf{x}\end{bmatrix}^T=\mathbf{B}$. Note that ${D^i}^T D^i = {D^i}^T = D^i$, and $A, \mathbf{B}$ are computed as line 2 and 3 in Algorithm 2. Solving this linear system directly can be computational expensive when n is large. However, observing the structure of A, we have the following result.

Observation 1. *For* $\lambda > 0, {U^*}^T U^* = I_d, \rho_i > 0, \forall i \in \{1,\cdots,l\}$, *we have* $A \succ 0$.

Proof. In (5.6), the closed form solution comes from the linear system, $A\begin{bmatrix}\mathbf{w} & \mathbf{x}\end{bmatrix}^T=\mathbf{B}$. A can be computed as

$$A=\begin{bmatrix}\lambda I_d & \lambda {U^*}^T \\ \lambda U^* & \lambda I_n+\sum_{i=1}^{l}\rho_i D^i\end{bmatrix} \tag{5.7}$$

and denoting $Q=\sum_{i=1}^{l}\rho_i D^i$, we have (for any $(\mathbf{w},\mathbf{x})$),

$$\begin{aligned}\begin{bmatrix}\mathbf{w}\\ \mathbf{x}\end{bmatrix}^T A\begin{bmatrix}\mathbf{w}\\ \mathbf{x}\end{bmatrix} &= \begin{bmatrix}\lambda(\mathbf{w}^T+\mathbf{x}^T U^*) & \lambda(\mathbf{w}^T{U^*}^T+\mathbf{x}^T)+\mathbf{x}^T Q\end{bmatrix}\begin{bmatrix}\mathbf{w}\\ \mathbf{x}\end{bmatrix} \\ &= \lambda(\mathbf{w}^T\mathbf{w}+\mathbf{x}^T U^*\mathbf{w}+\mathbf{w}^T{U^*}^T\mathbf{x}+\mathbf{x}^T\mathbf{x})+\mathbf{x}^T Q\mathbf{x} \\ &= \lambda\|\mathbf{x}+U^*\mathbf{w}\|_2^2+\mathbf{x}^T Q\mathbf{x}\end{aligned} \tag{5.8}$$

Let us check both terms in (5.8). Observe that $\forall \mathbf{x}$, $\mathbf{x}^T Q \mathbf{x} \geqslant 0$. Next, as $\lambda \|\mathbf{x} + U^* \mathbf{w}\|_2^2 \geqslant 0$, for the LHS of the identity in (5.8), we have

$$\begin{bmatrix} \mathbf{w} \\ \mathbf{x} \end{bmatrix}^T A \begin{bmatrix} \mathbf{w} \\ \mathbf{x} \end{bmatrix} \geqslant 0 \tag{5.9}$$

For the equality to hold in (5.9), we need to have both the terms, $\|\mathbf{x} + U^* \mathbf{w}\|_2^2$ and $\mathbf{x}^T Q \mathbf{x}$ equal to 0. But $Q \succ 0$ because $\mathbf{x}^T Q \mathbf{x} = 0$ only when $\mathbf{x} = 0$. Further, $\mathbf{w} = 0$ whenever $\mathbf{x} = 0$, for $\|\mathbf{x} + U^* \mathbf{w}\|_2^2$ to be zero. Hence equality in (5.9) holds only when $\mathbf{w}$ and $\mathbf{x}$ are zero.

□

Together with the fact that A is sparse, we use a GPU solver using preconditioned conjugate gradient method, which reduces the running time significantly (Nocedal and Wright, 2006).

$\mathbf{z}^i$-minimization: Minimizing a specific $\mathbf{z}^i$ for group i, is independent of the other $\mathbf{z}^{j \neq i}$ and hence can be solved in parallel. The objective w.r.t $\mathbf{z}^i$ takes the form,

$$\min_{\mathbf{z}^i} \quad \mu_i \|\mathbf{z}^i\|_2 - {\mathbf{y}_k^i}^T \mathbf{z}^i + \frac{\rho_i}{2} \|D^i \mathbf{x}_{k+1} - \mathbf{z}^i\|_2^2 \tag{5.10}$$

Denoting $\mathbf{r}_k^i = \mathbf{D}^i \mathbf{x}_{k+1} + \frac{{\mathbf{y}^i}_k}{\rho_i}$, (5.10) has a closed form solution by the block soft thresholding formula (Yuan and Lin, 2006) given as,

$$\mathbf{z}_{k+1}^i = \max\{\|\mathbf{r}_k^i\|_2 - \frac{\mu_i}{\rho_i}, 0\} \frac{\mathbf{r}_k^i}{\|\mathbf{r}_k^i\|_2} \tag{5.11}$$

$\mathbf{y}^i$-updating: We can now update $y^i, \forall i \in \{1, \cdots, l\}$ along the gradient direction by,

$${\mathbf{y}^i}_{k+1} = \mathbf{y}_k^i + \rho_i (D^i \mathbf{x}_{k+1} - \mathbf{z}_{k+1}^i) \tag{5.12}$$

The above analysis shows that the key update steps (summarized in

Algorithm 2) within a ADMM procedure can all be performed efficiently. In our implementation, we alternatively solve for $(\mathbf{w}^*, \mathbf{x}^*, \mathbf{z}^{i*}, \mathbf{y}^*)$ until the changes in $\mathbf{x}$ and the objective value reaches a desired level of tolerance. Given the convexity of each item in the tuple, we have the following convergence theorem.

Theorem 5.1. *For* $\lambda > 0, \mu_i > 0, \rho_i > 0, \forall i \in \{1, \cdots, l\}$, *the sequence* $\{(\mathbf{w}_k, \mathbf{x}_k, \{\mathbf{z}_k^i\}, \{\mathbf{y}^i\})\}$ *generated by Alg. 2 from any initial point* $(\mathbf{w}_0, \mathbf{x}_0, \{\mathbf{z}_0^i\}, \{\mathbf{y}_0^i\})$ *converges to* $(\mathbf{w}^*, \mathbf{x}^*, \{\mathbf{z}^{i*}\}, \{\mathbf{y}^{i*}\})$, *which minimizes (5.5) at fixed* U^*.

Proof. Our proof emulates the convergence proof in (Boyd et al., 2011). We first show that model with a fixed U agrees with the standard ADMM formulation in (Boyd et al., 2011).

$$\begin{aligned} \min_{\mathbf{w},\mathbf{x},\mathbf{z}^i} \quad & \sum_{i=1}^{l} \mu_i \|\mathbf{z}^i\|_2 + \frac{\lambda}{2}\|U\mathbf{w} + \mathbf{x} - \mathbf{v}\|_2^2 \\ \text{s.t.} \quad & \mathbf{z}^i = D^i\mathbf{x}, \end{aligned} \tag{5.13}$$

If we denote $f(\mathbf{w}, \mathbf{x}) = \frac{\lambda}{2}\|U\mathbf{w} + \mathbf{x} - \mathbf{v}\|_2^2$, $g(\{\mathbf{z}^i\}) = \sum_{i=1}^{l} \mu_i \|\mathbf{z}^i\|_2$, our problem (5.13) is really a special case of (3.1) in (Boyd et al., 2011).

We next need to show that (5.13) satisfies the two main assumptions made in the convergence proof given in (Boyd et al., 2011).

Assumption (i) $f(\mathbf{w}, \mathbf{x})$ and $g(\{\mathbf{z}^i\})$ are both convex, proper and closed.

Assumption (ii) $\mathcal{L}(\mathbf{w}^*, \mathbf{x}^*, \{\mathbf{z}^{i*}\}, \{\mathbf{y}^{i*}\})$ has a saddle point.

For notational simplicity, denote $\mathbf{c} = \begin{bmatrix} \mathbf{w} & \mathbf{x} \end{bmatrix}^T$, $\hat{U} = \begin{bmatrix} U & I_n \end{bmatrix}$ and $\hat{D}^i = \begin{bmatrix} 0_{n\times d} & D^i \end{bmatrix}$ ($0_{n\times d}$ is a $n \times d$ zero matrix). Using this notation, $f(\mathbf{w}, \mathbf{x}) = f(\mathbf{c}) = \frac{\lambda}{2}\|\hat{U}\mathbf{c} - \mathbf{v}\|_2^2$. Here, $f(\mathbf{c})$ is convex. By non-negativity of the norm squared function, $f(\mathbf{c}) \geqslant 0 > -\infty$, and taking $\mathbf{c} = 0$, we have $f(\mathbf{c}) = \|v\|_2^2 < \infty$. Hence, $f(\mathbf{c})$ is proper. Further, the domain of $\mathbf{c}$ is $\mathcal{R}^{n+d}$ and $f(\mathbf{c})$ is continuous on that domain. Following the closure property of proper convex functions (Rockafellar, 1970), we see that $f(\mathbf{c})$ is closed.

Following similar arguments as above, consider $g_1(\mathbf{z}^i) = \|\mathbf{z}^i\|_2$. Since $\|.\|_2$ is convex and increasing and as $\mathbf{z}^i = D^i x$, using the composition rule, $g_1(\mathbf{z}^i)$ is convex. Using the non-negativity of the norm and by taking $\mathbf{x} = 0$ which gives $g_1(\mathbf{z}^i) < \infty$, we have $g_1(\mathbf{z}^i)$ is proper. Finally, observe that $g_1(\mathbf{z}^i)$ is a continuous function of $\mathbf{x}$, and the domain on $\mathbf{x}$ ($\mathcal{R}^n$) is closed. Hence, $g_1(\mathbf{z}^i)$ a closed proper convex function. The non-negative sum of closed proper convex functions is also closed proper convex when the domain of summation remains unchanged. This concludes the proof of the first assumption.

For the second part, using the new notation, the augmented Lagrangian is,

$$\begin{aligned} &\mathcal{L}_0(\mathbf{w}, \mathbf{x}, \{\mathbf{z}^i\}, \{\mathbf{y}^i\}) \sim \mathcal{L}_0(\mathbf{c}, \{\mathbf{z}^i\}, \{\mathbf{y}^i\}) \\ &= \sum_{i=1}^{l} \mu_i \|\mathbf{z}^i\|_2 + \frac{\lambda}{2}\|\hat{U}\mathbf{c} - \mathbf{v}\|_2^2 + \sum_{i=1}^{l} \mathbf{y}^{i^T}(\mathbf{z}^i - \hat{D}^i\mathbf{c}) \end{aligned} \tag{5.14}$$

First, observe that the domain of $\mathbf{c}$, $\mathbf{z}^i$, and $\mathbf{y}^i$ is $\mathcal{R}^{n+d}$, $\mathcal{R}^n$, and $\mathcal{R}^n_+$ respectively, which are compact and convex sets. Fixing $\mathbf{y}^i$s for $i = 1, \ldots, l$, $\mathcal{L}_0$ is a convex function of $\mathbf{c}$ and $\mathbf{z}^i$. This follows from the fact that the first two terms in (5.14) are convex and the last term is affine in the primal parameters (when $\mathbf{y}^i$'s are fixed). So, there exists a triple, $(\mathbf{c}^*, \{\mathbf{z}^{i^*}\}, \{\mathbf{y}^{i^*}\})$ such that

$$\mathcal{L}_0(\mathbf{c}^*, \{\mathbf{z}^{i^*}\}, \{\mathbf{y}^{i^*}\}) \leqslant \mathcal{L}_0(\mathbf{c}, \{\mathbf{z}^i\}, \{\mathbf{y}^{i^*}\}).$$

Further, for a fixed ($\mathbf{c}$,$\mathbf{z}^i$), $\mathcal{L}_0$ is an linear combination of affine functions in $\mathbf{y}^i$s. Hence it is concave. So, there exists a triple, $(\mathbf{c}^*, \{\mathbf{z}^{i^*}\}, \{\mathbf{y}^{i^*}\})$

$$\mathcal{L}_0(\mathbf{c}^*, \{\mathbf{z}^{i^*}\}, \{\mathbf{y}^i\}) \leqslant \mathcal{L}_0(\mathbf{c}^*, \{\mathbf{z}^{i^*}\}, \{\mathbf{y}^{i^*}\}).$$

Thus, $\mathcal{L}_0$ has a saddle point $(\mathbf{c}^*, \{\mathbf{z}^{i^*}\}, \{\mathbf{y}^{i^*}\})$ in the primal–dual domain.

□

Algorithm 2 ADMM for solving $(\mathbf{w}^*, \mathbf{x}^*)$

In: Subspace matrix: U^*, observation: $\mathbf{v}$, initial: $\mathbf{x}_0$, $\mathbf{z}_0^i, \mathbf{y}_0^i$, group operator: D^i, hyper-parameters: λ, μ, ρ
Out: Subspace coefficient: $\mathbf{w}^*$, structured outliers: $\mathbf{x}^*$
Procedure:

for $k = 0 \rightarrow K$ **do**
  $A \leftarrow \begin{bmatrix} \lambda I_d & \lambda U^{*T} \\ \lambda U^* & \lambda I_n + \sum_{i=1}^{l} \rho_i D^i \end{bmatrix}$;
  $\mathbf{B} \leftarrow \begin{bmatrix} \lambda U^{*T}\mathbf{v} \\ \lambda \mathbf{v} - \sum_{i=1}^{l} D^i \mathbf{y}_k^i + \sum_{i=1}^{l} \rho_i D^i \mathbf{z}_k^i \end{bmatrix}$
  $(\mathbf{w}_{k+1}, \mathbf{x}_{k+1}) \leftarrow \min_{\mathbf{w},\mathbf{x}} \|(A[\mathbf{w} \quad \mathbf{x}]^T - \mathbf{B})\|^2$ using GPU solver
  $\mathbf{r}_k^i \leftarrow \mathbf{D}^i \mathbf{x}_{k+1} + \frac{\mathbf{y}^i{}_k}{\rho_i}$
  $\mathbf{z}_{k+1}^i \leftarrow \max\{\|\mathbf{r}_k^i\|_2 - \frac{\mu_i}{\rho_i}, 0\} \frac{\mathbf{r}_k^i}{\|\mathbf{r}_k^i\|_2}$
  $\mathbf{y}^i{}_{k+1} \leftarrow \mathbf{y}_k^i + \rho_i (D^i \mathbf{x}_{k+1} - \mathbf{z}_{k+1}^i)$
  Stop if tolerance conditions satisfied.
end

5.4.2 Update of U with estimated $(\mathbf{w}^*, \mathbf{x}^*)$

The key idea to update U is to refine it from the estimation $(\mathbf{w}^*, \mathbf{x}^*)$ derived from the current observation $\mathbf{v}$ on the Grassmannian. Given the estimated tuple $(\mathbf{w}^*, \mathbf{x}^*)$, the derivative of $\mathcal{L}(.)$ in (5.5) with respect to the components of U and the gradient are given by

$$\frac{\partial \mathcal{L}}{\partial U} = \lambda(U\mathbf{w}^* + \mathbf{x}^* - \mathbf{v})\mathbf{w}^{*T} = \mathbf{s}\mathbf{w}^{*T} \tag{5.15}$$

where $\mathbf{s} = \lambda(U\mathbf{w}^* + \mathbf{x}^* - \mathbf{v})$ denotes the residual vector. Using identity (2.70) in (Arias et al., 1998), the gradient on the Grassmannian can be computed by

$$\nabla \mathcal{L} = (I - UU^T)\frac{\partial \mathcal{L}}{\partial U} = (I - UU^T)\mathbf{s}\mathbf{w}^{*T} = \mathbf{s}\mathbf{w}^{*T} \tag{5.16}$$

(5.16) is valid because the residual vector $\mathbf{s}$ is orthogonal to all of the columns of U. It is obvious that $\nabla \mathcal{L}$ is a rank one matrix, since $\mathbf{s}$ and $\mathbf{w}^*$ are both vectors. Hence, we can compute the compact SVD of $\nabla \mathcal{L}$ by

$\nabla\mathcal{L} = \mathbf{p}\sigma\mathbf{q}$, where $\mathbf{p} = \frac{\mathbf{s}}{\|\mathbf{s}\|}$, $\sigma = \|\mathbf{s}\|\|\mathbf{w}^*\|$ and $\mathbf{q} = \frac{\mathbf{w}^*}{\|\mathbf{w}^*\|}$.

Following (Arias et al., 1998; He et al., 2012), we update U with a gradient stepsize η in the direction $-\nabla\mathcal{L}$.

$$\nabla\mathcal{L} = \frac{\mathbf{s}}{\|\mathbf{s}\|} \times \|\mathbf{s}\|\|\mathbf{w}^*\| \times \left(\frac{\mathbf{w}^*}{\|\mathbf{w}^*\|}\right)^T = \mathbf{p}\sigma\mathbf{q}^T$$

Here, we approximate the full SVD with

$$\nabla\mathcal{L} = S\Sigma V^T = \begin{bmatrix}\mathbf{p} & \mathbf{p}_2 & \cdots & \mathbf{p}_d\end{bmatrix} \times \mathrm{diag}(\sigma, 0, \cdots, 0) \times \begin{bmatrix}\mathbf{q} & \mathbf{q}_2 & \cdots & \mathbf{q}_d\end{bmatrix}^T$$

where $S = \begin{bmatrix}\mathbf{p} & \mathbf{p}_2 & \cdots & \mathbf{p}_d\end{bmatrix}$, $\Sigma = \mathrm{diag}(\sigma, 0, \cdots, 0)$, $V = \begin{bmatrix}\mathbf{q} & \mathbf{q}_2 & \cdots & \mathbf{q}_d\end{bmatrix}$, $\mathbf{p}_2, \cdots, \mathbf{p}_d$ and $\mathbf{q}_2, \cdots, \mathbf{q}_d$ are slack orthonormal basis, which will be omitted by the zero singular values.

By Thm. 2.65 in (Arias et al., 1998), we can update our subspace U with stepsize η by

$$U(\eta) = \begin{bmatrix}UV & -S\end{bmatrix} \begin{bmatrix}\cos(\Sigma\eta) \\ \sin(\Sigma\eta)\end{bmatrix} V^T$$

Observe that the above update involves full matrix operations. It can

be simplified as,

$$\begin{aligned}
U(\eta) =& UV\cos(\Sigma\eta)V^T - S\sin(\Sigma\eta)V^T \\
=& UV\cos(\mathrm{diag}(\sigma\eta, 0, \cdots, 0))V^T - S\sin(\mathrm{diag}(\sigma\eta, 0, \cdots, 0))V^T \\
=& UV\mathrm{diag}(\cos(\sigma\eta), 1, \cdots, 1)V^T - S\mathrm{diag}(\sin(\sigma\eta), 0, \cdots, 0)V^T \\
=& UV\mathrm{diag}(1, 1, 1, \cdots, 1)V^T + UV\mathrm{diag}(\cos(\sigma\eta) - 1, 0, \cdots, 0)V^T \\
& - S\mathrm{diag}(\sin(\sigma\eta), 0, \cdots, 0)V^T \\
=& UVI_d V^T + UV\mathrm{diag}(\cos(\sigma\eta) - 1, 0, \cdots, 0)V^T \\
& - S\mathrm{diag}(\sin(\sigma\eta), 0, \cdots, 0)V^T \\
=& U + (\cos(\sigma\eta) - 1)U\mathbf{q}\mathbf{q}^T - \sin(\sigma\eta)\mathbf{p}\mathbf{q}^T
\end{aligned} \tag{5.17}$$

where η is the stepsize to update the subspace U on the Grassmann manifold. We incorporate an adaptive stepsize η using the updating scheme by (Klein et al., 2009) but in the experiments, a constant stepsize works well also. To show the validity of (5.17), we give the following lemma,

Lemma 5.1. *The subspace updating procedure (5.17) preserves the column-wise orthogonality of* U *(He et al., 2012).*

Proof. The residual vector $\mathbf{s}$ is orthogonal to all the columns of U, thus we have

$$U^T\mathbf{p} = U^T\frac{\mathbf{s}}{\|\mathbf{s}\|} = 0$$

Also, $\mathbf{p}, \mathbf{q}$ are unary vectors, hence $\mathbf{q}^T\mathbf{q} = 1, \mathbf{p}^T\mathbf{p} = 1$. Now we show

$U(\eta)^T U(\eta) = I_d$:

$$
\begin{aligned}
& U(\eta)^T U(\eta) \\
=& \left(U + (\cos(\sigma\eta) - 1)UU\mathbf{q}\mathbf{q}^T - \sin(\sigma\eta)\mathbf{p}\mathbf{q}^T\right)^T (U + \\
& (\cos(\sigma\eta) - 1)UU\mathbf{q}\mathbf{q}^T - \sin(\sigma\eta)\mathbf{p}\mathbf{q}^T) \\
=& U^T U + (\cos(\sigma\eta) - 1)U^T UU\mathbf{q}\mathbf{q}^T - \sin(\sigma\eta)U^T\mathbf{p}\mathbf{q}^T + (\cos(\sigma\eta) - 1)\mathbf{q}\mathbf{q}^T U^T U \\
& + (\cos(\sigma\eta) - 1)^2\mathbf{q}\mathbf{q}^T U^T UU\mathbf{q}\mathbf{q}^T - (\cos(\sigma\eta) - 1)\sin(\sigma\eta)\mathbf{q}\mathbf{q}^T U^T\mathbf{p}\mathbf{q}^T \\
& - \sin(\sigma\eta)\mathbf{q}\mathbf{p}^T U - (\cos(\sigma\eta) - 1)\sin(\sigma\eta)\mathbf{q}\mathbf{p}^T UU\mathbf{q}\mathbf{q}^T + \sin^2(\sigma\eta)\mathbf{q}\mathbf{p}^T\mathbf{p}\mathbf{q}^T \\
=& I_d + (\cos(\sigma\eta) - 1)\mathbf{q}\mathbf{q}^T - 0 + (\cos(\sigma\eta) - 1)\mathbf{q}\mathbf{q}^T + (\cos(\sigma\eta) - 1)^2\mathbf{q}\mathbf{q}^T \\
& - 0 - 0 - 0 + \sin^2(\sigma\eta)\mathbf{q}\mathbf{q}^T \\
=& I_d + (2\cos(\sigma\eta) - 2 + \cos^2(\sigma\eta) - 2\cos(\sigma\eta) + 1 + \sin^2(\sigma\eta))\mathbf{q}\mathbf{q}^T \\
& \quad /* \text{ (} 2\cos(\sigma\eta) \text{ cancels out and } \cos^2(\sigma\eta) + \sin^2(\sigma\eta) = 1 \text{) } */ \\
=& I_d
\end{aligned}
$$

Thus, the subspace updating procedure preserves the column-wise orthogonality of U.

□

Notice that (5.17) is related to a stochastic gradient updating procedure, where at each iteration, we draw an example in a sequential manner, instead of random sampling. We compute the gradient from each example, and use this gradient to improve the subspace. The optimal subspace is not computed fully, and is instead updated by analyzing successive observations. At this point, we are ready to summarize our optimization pipeline in Algorithm 3.

5.5 Applications

We apply GOSUS to the problem of foreground/background separation and multiple face tracking/identity management. Our implementation

Algorithm 3 Main Procedure of GOSUS

In: Observation: V, subspace initialization: U_0, hyperparameters: λ, μ, ρ
Out: Approximated signal: B, structured outliers: X
Procedure:

1: **for** $t = 1 \rightarrow T$ **do**
2: Solve $(\mathbf{w}^*, \mathbf{x}^*, \{\mathbf{z}^{i*}\}, \{\mathbf{y}^{i*}\})$ by Algorithm 2;
3: (Optional) Update stepsize η_t ;
4: Update U_t by (5.17);
5: **end**

and experiments are publicly available in http://pages.cs.wisc.edu/~jiaxu/projects/gosus/.

5.5.1 Background Subtraction

Datasets. We used two benchmark datasets: Perception Test Images Sequences (Li et al., 2004) and Wallflower Test Images Sequences (Toyama et al., 1999), which are heavily used in recent work (Mairal et al., 2011; Qin and Goldfarb, 2012; He et al., 2012; Wang et al., 2012b). The data includes 12 video sequences, with a variety of characteristics, such as changing foreground with static (Bootstrap, Shopping Mall, Hall) and dynamic (Fountain, Escalator, Waving Trees, Water Surface, Curtain, Campus) backgrounds as well as illumination changes (Lobby, Time of Day, Light switch).

Experiments setup. GOSUS is compared to three different models: **(i)** Batch model: (RPCA) Robust PCA using Inexact Augmented Lagrange Multiplier Method (Lin et al., 2010) **(ii)** Batch model: (RPMF) Robust Probabilistic Matrix Factorization (Wang et al., 2012b), **(iii)** Online model: (GRASTA) Grassmannian Robust Adaptive Subspace Tracking He et al. (2012). For these baseline methods, we use code from the corresponding authors' websites. For RPCA, the maximum number of iterations was set to 1000 and the regularization parameter was $\frac{1}{\gamma}$ (γ is the number of pixels in the image frame). The regularization parameters (one for each of the two

factorizing matrices) in RPMF were set to 1. To obtain best possible results from GRASTA, sub-sampling was turned off and the code was initialized with the suggested default settings. In GOSUS, for each color frame, we extract a vector $\mathbf{v}$ with size n (i.e, # of pixels times 3 for the RGB channels). The ADMM hyperparameters used were $\rho^i = 0.3/\text{mean}(\mathbf{v}), \forall i = 1, \cdots, l$ and stepsize η was 0.01. λ was set using cross–validation and all μ_is were set to 1. An initial estimate of the background subspace was set as a random orthonormal matrix $n \times d$ (where $d = 5$, n is equal to three times # of pixels in each frame). The tolerance level for all methods was set at 10^{-6}. Note that RPCA and RPMF see all the data at once which gives them an inherent advantage over GRASTA. Receiver Operating Characteristic (ROC) curves, and the corresponding area under curve (AUC) values are used as performance evaluation measures.

Group Construction. Together with a 3×3 grid group structure (patches) and hierarchical tree group structure (Jia et al., 2012), we also use a coarse-to-fine superpixel group construction. Pixels belonging to each superpixel form a group which can overlap with others. We employ the SLIC superpixel algorithm (Achanta et al., 2012), with region sizes $\{80, 40, 20, 10\}$ in order to generate coarse-to-fine groups. The group construction captures the boundary information of objects and our evaluations show this setting works well.

Quantitative Evaluations. Figure 5.3 summarizes the ROC plots for 6 videos, representative examples from the three different data categories that constitute our data. Table 5.1 presents the AUC values for all 12 videos. The results indicate that GOSUS performs better than all baseline methods (except on the 'Light Switch' video where RPMF was the best). In particular, from Table 5.1 we see that GOSUS competes very favorably with GRASTA, both being online methods. This is particularly clear in data with dynamic background (Fountain, Campus) and illumination changes (Light Switch, Lobby). Also note that RPCA and RPMF are batch models,

	Models			
Video Datasets	RPCA (Lin et al., 2010)	RPMF (Wang et al., 2012b)	GRASTA (He et al., 2012)	**GOSUS**
Fountain	0.94	0.94	0.69	**0.99**
Escalator	0.91	0.90	0.90	**0.96**
WavingTrees	0.74	0.84	0.87	**0.98**
Campus	0.90	0.86	0.77	**0.98**
Bootstrap	0.87	0.91	0.87	**0.93**
WaterSurface	0.73	0.84	0.87	**0.97**
Hall	0.82	0.90	0.76	**0.93**
Time of Day	0.80	0.85	0.84	**0.89**
LightSwitch	0.87	**0.92**	0.62	0.88
Curtain	0.87	0.90	0.88	**0.96**
Lobby	0.89	0.94	0.70	**0.95**
ShoppingMall	0.92	0.93	0.90	**0.94**

Table 5.1: Area under ROC curves for RPCA, RPMF, GRASTA, GOSUS.

and GOSUS attains better performance than either in almost all categories, which supports the intuition that imposing structure (spatial homogeneity) on the outliers enables it to improve estimating the subspace.

Qualitative Evaluations. Figure 5.4 shows the effectiveness of GOSUS in adapting to intermittent object motion in the background. GOSUS starts with a random subspace and finds the correct background after 200 frames. At frame $t_0 + 645$, a person comes in, sits for a while, and leaves on frame $t_0 + 882$. GOSUS successfully learn the new background (notice the pose of the red chair) as early as frame $t_0 + 930$.

Figure 5.5 shows example detections for four different videos (one frame for each) of our algorithm and several baselines. The first row corresponds to an example with static background, and GOSUS performs comparably with others. The last three videos have dynamic background, where the water surface is moving, trees are swaying, etc. Observe that outputs of GOSUS contain very few isolated foreground regions, unlike GRASTA and the other batch models RPCA and RPMF, which do not regularize the secondary term at all. Further, the foreground object by itself is better segmented (very few pixels missing along the boundaries)

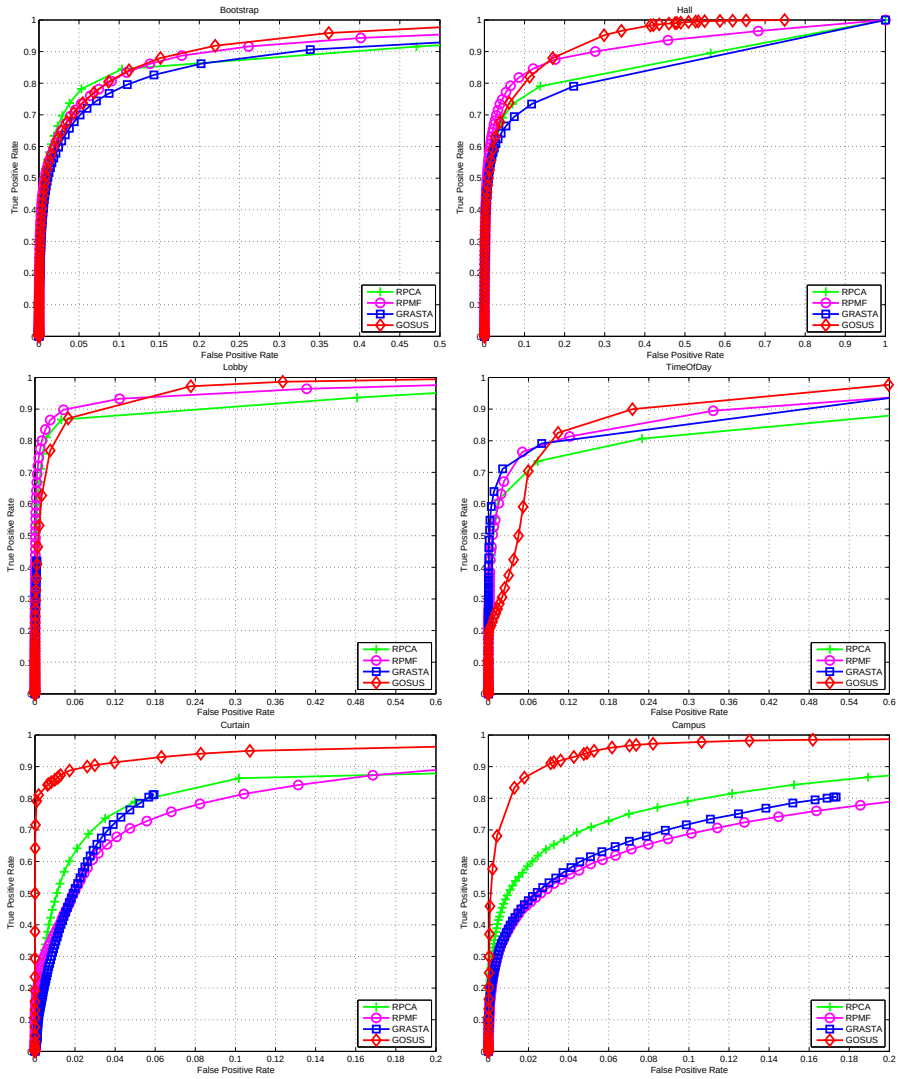

Figure 5.3: ROC curves of 6 datasets for 3 different categories (row 1: static background; row 2: illumination changes; row 3: dynamic background) showing the performance of RPCA, RPMF, GRASTA and GOSUS.

in GOSUS. This shows that the structured sparsity used in GOSUS, is not only acting as a noise removal filter (on salt-and-pepper like foreground detections) but also improves the estimation of the perturbed (dynamic/moving) subspace. Further note that GOSUS outperforms both batch

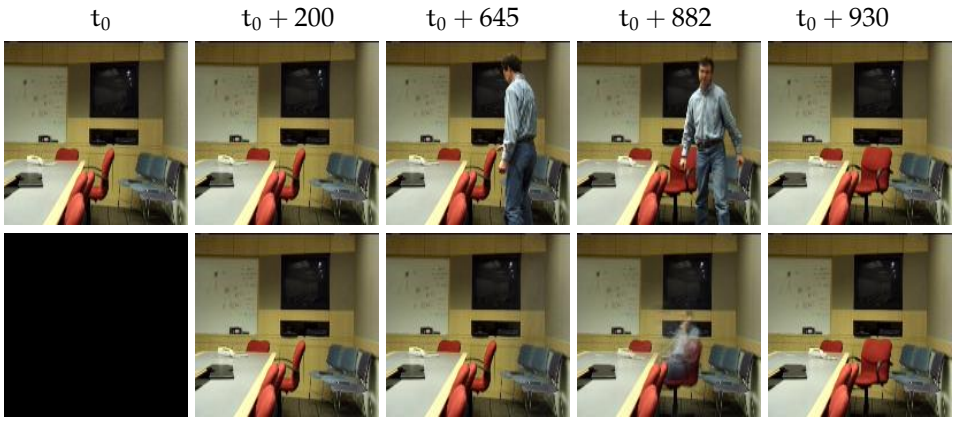

Figure 5.4: Effectiveness on adapting to intermittent object motion in the background. The first row are the original frames, and the second row are the background learned by GOSUS.

models (RPCA and RPMF), since the latter do not use any form of spatial contiguity. Overall, both Table 5.1 and Figure 5.5 indicate that GOSUS improves background subtraction in various categories, and offers substantial improvements when the background is dynamic.

We also compare GOSUS with sparse coding based methods. As shown in Figure 5.6, our method is competitive with (Mairal et al., 2011), except there are some grid artifacts from (Mairal et al., 2011) due to their group construction. However, our algorithm achieves $1 \sim 2$ frames per second given the original image size (no resizing). This is significantly faster than the bi-level process used in (Mairal et al., 2011), and several orders of magnitude faster than speed reported in (Qin and Goldfarb, 2012), a method devoted to optimizing structured sparsity norm.

5.5.2 Multiple Face Tracking/Identity Management

Our second application is to track multiple faces (keeping track of the identities) in real world videos, e.g., TV shows and movies. This problem is extremely challenging due to the dramatic variation in the appearance of each person's face, and the dynamics of characters coming in and out.

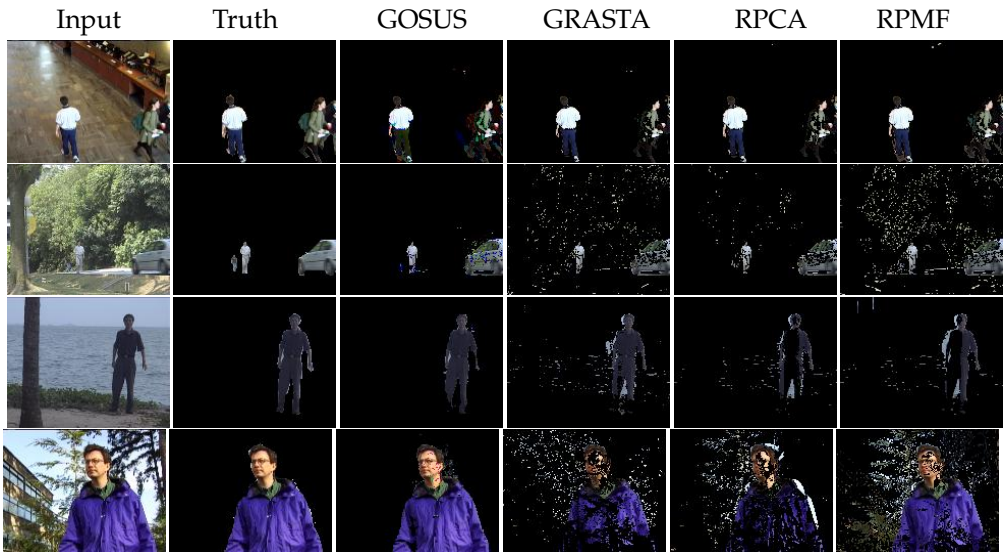

Figure 5.5: Example results on Bootstrap, Campus, and Water Surface comparing GOSUS to ground truth followed by GRASTA, RPCA and RPMF.

Existing work has achieved the state of art by utilizing all visual frames, audio, aligned subtitle and script texts (Everingham et al., 2006; Sivic et al., 2009). We aim to tackle this problem using *only* the visual data, and in an efficient manner.

We first run Viola-Jones detector (Viola and Jones, 2004) on all image frames. For robustness to pose/expression variation, lighting, and partial occlusion, we use a parts-based descriptor extracted around detected facial features (Everingham et al., 2006; Sivic et al., 2009). We detect 13 facial feature points (the left and right corners, center of each eye and mouth, the two nostrils, tip of the nose, center of the eyes) and simply extract a pixel-wise descriptor of the circular region around each feature point (which we transform on to a canonical face). This gives us a 1937 dimensional feature vector $\mathbf{v}$ for each face. The structured sparsity prior refers to each circular region as a group. This setting can capture the occlusion created by glasses/shadows as well as self-occlusions due to pose variations.

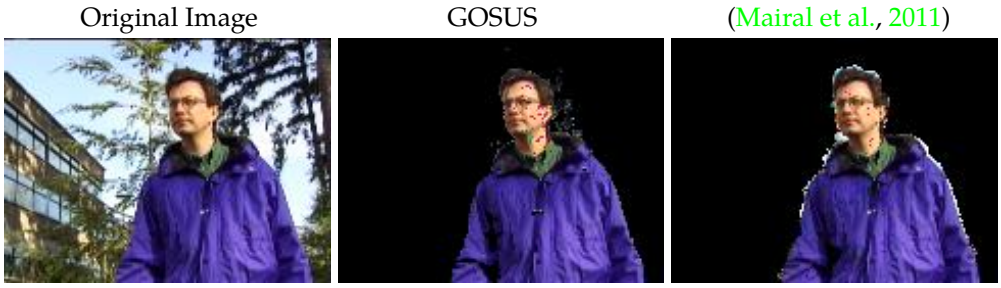

Figure 5.6: Comparison with (Mairal et al., 2011) using overlapping groups.

The tracking and identity management procedure is related to face recognition approaches reported in (Turk and Pentland, 1991; Jia et al., 2012). We consider $\mathbf{U}$ as a face subspace, with each column representing an 'eigenface'. The observed face vector is described by a combination of eigenfaces using $\mathbf{w}$ and structured outliers $\mathbf{x}$, created by occlusion/disguise. $\mathbf{w}$ acts as a signature for each face. False positives from the face detector are rejected by thresholding the norm of $\mathbf{x}$. We maintain a window (size 400) for tracked faces. The label for each face (i.e., identity) comes from a majority nearest neighbor votes from this window, along with temporal consistency. When a new face is found, we add a new label/identity to our signature window.

We demonstrate the effectiveness of GOSUS on several real world videos from the TV show: 'The Big Bang Theory'. Sample results are shown in Figure 5.7. Faces marked with the same number are from the same track. Firstly observe that Amy in frame 151 and frame 1009, is tracked correctly even with significant changes in camera shot. The person marked 7 (Penny) is also correctly tracked over a long time (frame 1297 through 2012 to 3693). However, different tracks for the same person may be introduced if the person (Rajesh/Sheldon marked as 3/4) disappears in the video for a long time or has dramatic facial expressions.

Though our preliminary application on multiple face tracking shows

Figure 5.7: Examples of multiple face tracking in videos from the Big Bang Theory. Faces marked with the same number are from the same track. Frame number is shown on the left top corner. Complete video results are provided on the project website.

promising results for real videos, the current pipeline is limited (in terms of efficiency) to the output from the face detector. On these videos (720×1280), it takes about 2 seconds to detect all possible faces (for each frame), whereas GOSUS on its own can process all 6000 frames with all detected faces in ~ 20 seconds. Also note that the face detector can only detect frontal faces (the face of the male in frame 151 is missing), and can introduce a sizable number of false positives for real world videos. Improvements to these modules will seamlessly yield improvements in the empirical performance of GOSUS.

5.6 Summary

The main contribution of this chapter is an intuitive yet expressive model, GOSUS, which exploits a meaningful structured sparsity term to significantly improve the accuracy of online subspace updates. We discuss the modeling and optimization aspects in detail. Our solution is based on ADMM, where most key steps in the update procedure reduce to simple matrix operations yielding real-time performance for several interesting problems in video analysis.

6 GAZE-ENABLED EGOCENTRIC VIDEO SUMMARIZATION

The final visual parsing task we tackle is egocentric (a.k.a, first person) video summarization. The goal here is to temporally parse multiple hours long first person video into interesting/informative components (i.e., foreground sequences to be picked out) and less important video frames (i.e., background to leave out). This is an important parsing problem, as the proliferation of wearable cameras and the number of videos of users documenting their personal lives are rapidly increasing. Since such videos may span hours, there is an important need for mechanisms that represent the information content in a compact form (i.e., shorter videos which are more easily browsable/sharable). Motivated by these applications, we tackle the problem of egocentric video summarization. In this chapter, we present a first study on the utility of gaze in egocentric video summarization and demonstrate that using gaze tracking information (such as fixation and saccade) significantly helps the summarization task. It allows meaningful comparison of different image frames, and enables deriving personalized summaries (gaze provides a sense of the camera wearer's intent). We formulate a summarization model which captures common-sense properties of a good summary, and show that it can be solved as a submodular function maximization with partition matroid constraints, opening the door to a rich body of work from combinatorial optimization. We evaluate our approach on a new gaze-enabled egocentric video dataset (over 15 hours), and demonstrate the superiority of our method compared with previous baselines. An earlier version of this chapter was published in (Xu et al., 2015a).

6.1 Problem Description

The advent of wearable cameras and the ability to record visual data from a first person point of view (namely, egocentric video) has opened the door to a rich trove of computer vision problems. These range from socio-behavioral modeling to analyzing recurring patterns in a person's daily life. Such a wealth of data poses an interesting scientific question — how should one compactly summarize continuous video streams acquired over many hours? A mature body of work on video summarization provides a meaningful starting point, but egocentric videos still pose unique challenges. We want to support continuous egocentric video capture, which will result in long segments, only a few subsets of which will actually contain 'memorable' or 'interesting' content. Further, simple measures of diversity among frames and low-level appearance or flow cues which are useful modules of a classical approach to video summarization may not be informative at all, in fact, even misleading. For example, strong motion cues and potentially strong differences among frames due to background clutter will show up prominently in a sequence of a long walk back from campus. The ideal solution would be to compress such redundant periods but also not leave out anomalies or shorter segments that may be interesting to the camera wearer.

The description above suggests that egocentric video summarization is an ill-posed problem. Indeed, these videos may have poor illumination, camera shake, rapidly changing background, and a spectrum of other confounding factors. Nonetheless, given that the proliferation of wearable image-capture systems will only increase, there is a need for systems that take a long egocentric video and distill it down to its informative parts. They offer the camera wearer the ability to browse and archive his/her daily activities (life log), and review (or search) it in the future. The last two years have seen a number of interesting strategies for this problem. For instance, (Khosla et al., 2013) observed that canonical viewpoints of

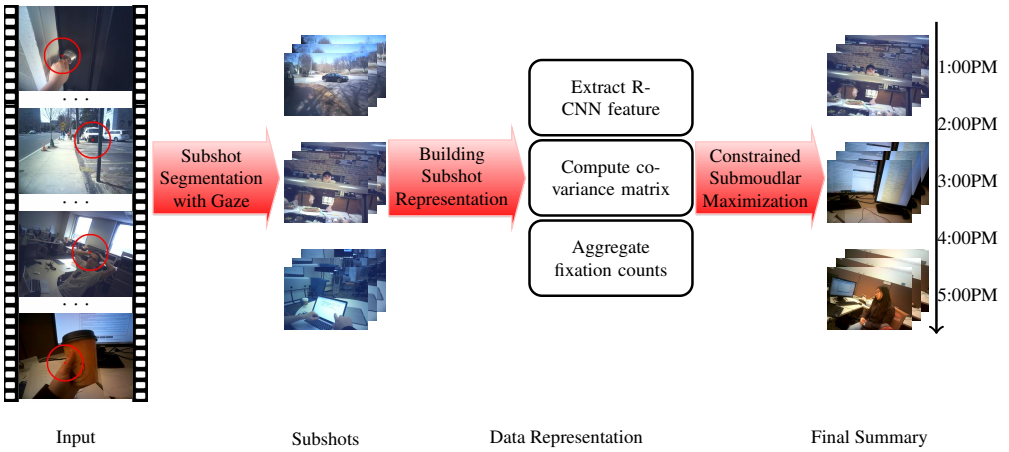

Figure 6.1: Overview of our submodular summarization algorithm: our approach takes an egocentric video with gaze tracking as input (first column), time windows (last column) as a partition matriod constraint, and produces a compact personalized visual summary: getting lunch, working in an office, and conversation with a colleague.

objects that are relevant for representation in a egocentric summary can be identified by mining large collections of images on the Internet. Very recently, (Lu and Grauman, 2013) proposed regularizing the summarization process with a so-called "storyline narrative": a coherent (chronological) set of video subshots. Both approaches have been shown to work well but need a nominal amount of training data, which can be very expensive to collect and limited at scale.

Despite the advances described above, the literature on this problem is still in its developmental phase. Approaches so far have not attempted to *personalize* the summary. But, egocentric video summarization *is* subjective and its utility depends greatly on its relevance to the camera wearer. The challenge is that personalization cannot be accomplished without close involvement of the user. This chapter makes the case that a powerful surrogate to personalization is *egocentric gaze*. In fact, how the person views the world through a sequence of gaze measurements conveys a strong

sense of his/her intent and interest. In a recent study (Yun et al., 2013), eye movements were found to inform visual understanding in different but complementary ways. For instance, relative importance of content in an image correlates to how a person's attention was spatially and temporally distributed (the patterns of saccades and fixations). We contend that egocentric gaze measurements are a key missing ingredient in egocentric video summarization – they serve to make the problem well posed, enable comparisons across frames (even with clutter) and provide guidance on which content the user would like to leave out. It turns out that such gaze measurements are now available as wearable devices, as small form-factor attachments (e.g., Pupil Labs (Kassner and Patera, 2012)) and/or can be predicted in a egocentric sequence via a combination of saliency and machine learning techniques (Li et al., 2013).

In this chapter, we address the issue of incorporating gaze information to efficiently summarize egocentric videos (Fig. 6.1 outlines the overview of our algorithm). Our main contributions are: **(i)** We make the first attempt to study the role of gaze in summarizing egocentric videos. To our knowledge, our results are the first to demonstrate that gaze gives the means to 'personalize' the synopsis of a long egocentric sequence which leads to results that are more *relevant* to the camera wearer — arguably, the primary measure of a summary's utility. **(ii)** On the modeling side, gaze helps make the problem well-posed. This leads to a property that is taken as granted in a standard computer vision problem but difficult to achieve with summarization objectives – that a better evaluation of the objective function indeed corresponds to a more meaningful summary. We formulate a summarization model which captures common-sense properties of a good summary: relevance, diversity, fidelity with the full egocentric sequence, and compactness. The optimization scheme is an adaptation of recent work on non-monotone submodular maximization with matroid constraints and comes with approximation guarantees. **(iii)** We introduce

a new dataset with 21 egocentric videos. Each video comes with calibrated gaze information, a summary annotation from the wearer as well as human experts.

6.2 Related Work

We first provide a brief review of literature from a few different lines of work that are related to this chapter.

Video Summarization. The problem of video summarization has been studied from various perspectives (Wolf, 1996; Laganiere et al., 2008; Ngo et al., 2003). Most methods select a sequence of keyframes (Wolf, 1996; Goldman et al., 2006; Khosla et al., 2013) or subshots (Lu and Grauman, 2013; Gygli et al., 2014; Gong et al., 2014) to form a visual summary of the most informative parts of a video. Previous summarization techniques are designed for professionally produced videos and rely on low-level features (Laganiere et al., 2008) and motion cues (Wolf, 1996). Some recent approaches extract scenes of interest by training a supervised model of important objects (Liu et al., 2010; Khosla et al., 2013), attention models (Ma et al., 2005), user preferences (Almeida et al., 2012), events (Wang et al., 2012a), multi-view (Fu et al., 2010), and user interactions (Goldman et al., 2006; Pongnumkul et al., 2008; Ellouze et al., 2010). These methods are general and usually do not perform well for user-shot videos or egocentric sequences, and so recent works (Khosla et al., 2013; Lee et al., 2012; Lu and Grauman, 2013) have investigated and offered specialized solutions. Other recent works offering various interesting improvements an/or directions include (Zhao and Xing, 2014; Potapov et al., 2014; Yeung et al., 2014).

Egocentric Video Analysis. Egocentric vision has attracted a great deal of interest in the last few years for applications such as activity detection and recognition (Fathi et al., 2011a; Pirsiavash and Ramanan, 2012; Ryoo and Matthies, 2013), object detection and segmentation (Li and Kitani,

2013; Ren and Gu, 2010), temporal segmentation and activity classification (Spriggs et al., 2009), and novel event detection (Aghazadeh et al., 2011). While several works have discussed potential uses of such sequences as a daily-log, summarization strategies have only appeared recently (Lee et al., 2012; Lu and Grauman, 2013). This chapter complements these developments by introducing gaze measurements as an alternative to direct user supervision.

Gaze in Computer Vision. Attention is an integral part of the human visual system and has been widely studied (Yun et al., 2013). Previous works have demonstrated the utility of gaze in object segmentation (Mishra et al., 2012; Xu et al., 2013a; Li et al., 2013), action recognition (Fathi et al., 2012) and action localization (Shapovalova et al., 2013). Gaze measurements contain importance cues regarding the most salient objects in the scene (Li et al., 2013) and the intent of the camera-wearer. These cues help the task of video analysis and can help overcome poor illumination and background clutter.

Submodular Optimization. Submodular function optimization is a well studied topic in theoretical computer science (Lee et al., 2010; Chekuri et al., 2011; Filmus and Ward, 2012). It has also been heavily explored, albeit in a specialized form for labeling (energy minimization) problems in vision (Boykov et al., 2001). Maximization of submodular functions has not found many applications in vision, although it has received much interest recently in machine learning (Krause et al., 2008; Krause and Guestrin, 2011; Liu et al., 2013; Iyer and Bilmes, 2013; Lin and Bilmes, 2011, 2012). A small but interesting body of papers has shown how submodular function optimization (either unconstrained or with knapsack constraints) can be used to model problems like sensor placement (Krause et al., 2008; Krause and Guestrin, 2011), feature selection (Liu et al., 2013) and document summarization (Lin and Bilmes, 2011, 2012).

6.3 Submodular Video Summarization

We now introduce our approach to gaze-enabled video summarization via submodular maximization. The starting point is to decompose a continuous video record into subshots which will form the basis for our optimization approach. We perform gaze-enabled subshot selection and extract feature representations for each subshot (we discuss this procedure in detail in our experimental section). Let the set of all subshots be $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$. Our objective is to choose a subset $\mathcal{S} \subseteq \mathcal{V}$, denoted as the summary of $\mathcal{V}$. In the following section, we formulate the video summarization problem as the maximization of a submodular function subject to some constraints.

As with any summarization task (Lin and Bilmes, 2012), we expect a summary to be a good representative of the video — informative but compact. These goals can be achieved by choosing a subset that maximizes two key properties, namely *relevance* and *diversity*, where the relevance term encourages the inclusion of important events from the larger video (i.e., coverage of the sequence), while diversity reduces redundancy in the summary. We define a few concepts and then give the precise forms of these terms.

Definition 6.1. *A set function* F *is monotone nondecreasing if* $\mathcal{A} \subseteq \mathcal{S}$, $F(\mathcal{A}) \leqslant F(\mathcal{S})$.

Definition 6.2. *For any* $\mathcal{A} \subseteq \mathcal{S} \subseteq \mathcal{V}$ *and* $i \in \mathcal{V}, i \notin \mathcal{S}$. $F(\mathcal{S})$ *is submodular if* $F(\mathcal{S} + i) - F(\mathcal{S}) \leqslant F(\mathcal{A} + i) - F(\mathcal{A})$.

6.3.1 Relevance and Diversity Measurement with Mutual Information

Intuitively, we want to select subshots which are most informative with respect to the entire video, that is, if given an ideal summary $\mathcal{S}$, the knowl-

edge of $\mathcal{V}$ is maximized, compared to any other subset of $\mathcal{V}$. A natural notion to quantify this is to minimize the conditional entropy function $H(\mathcal{V}\backslash\mathcal{S}|\mathcal{S})$. Unfortunately, several works (Krause et al., 2008) have shown that it can sometimes lead to suboptimal results. The reason is that conditional entropy is defined as $H(\mathcal{V}\backslash\mathcal{S}|\mathcal{S}) = H(\mathcal{V}) - H(\mathcal{S})$, therefore optimizing such a function is equivalent to maximizing $H(\mathcal{S})$. So, it only considers the entropy of the selected subshots, rather than taking the coverage over the entire video into account.

Instead, we want a criterion that helps identify the subset of subshots that most significantly reduces the uncertainty about the remainder of the sequence. Mutual information offers precisely this behavior. Specifically, we define our first objective as the mutual information between the sets $\mathcal{S}$ and $\mathcal{V}\backslash\mathcal{S}$,

$$\begin{aligned} M(\mathcal{V}\backslash\mathcal{S};\mathcal{S}) &= H(\mathcal{V}\backslash\mathcal{S}) - H(\mathcal{V}\backslash\mathcal{S}|\mathcal{S}) \\ &= H(\mathcal{V}\backslash\mathcal{S}) + H(\mathcal{S}) - H(\mathcal{V}) \end{aligned} \tag{6.1}$$

The optimal solution $\mathcal{S}^* = \text{argmax}_{\mathcal{S}} M$ obtains the maximum entropy over both the selected sequence $\mathcal{S}^*$ and the remaining sequence $\mathcal{V}\backslash\mathcal{S}^*$, as desired.

Next, we discuss how to compute this score for a summary $\mathcal{S}$. Let L be the $n \times n$ covariance matrix of the set of subshots $\mathcal{V}$, which we assume are Gaussian random variables. For $\mathcal{S} \subseteq \mathcal{V}$, let $L_{\mathcal{S}}$ be the principal submatrix of L indexed by $\mathcal{S}$. It is well known ((Lee et al., 2010)) that the entropy $H(\mathcal{S})$ of the random variables indexed by $\mathcal{S}$ can be computed as

$$H(\mathcal{S}) = \frac{1 + \log(2\pi)}{2}|\mathcal{S}| + \frac{1}{2}\log(\det(L_{\mathcal{S}})) \tag{6.2}$$

Then maximizing the mutual information is equivalent to maximizing

$$M(\mathcal{S}) = \frac{1}{2}\log(\det(L_{\mathcal{V}\backslash\mathcal{S}})) + \frac{1}{2}\log(\det(L_{\mathcal{S}})) \tag{6.3}$$

as $|\mathcal{S}| + |\mathcal{V}\backslash\mathcal{S}| = n$, and $H(\mathcal{V})$ is constant. Here, the first term of M mea-

sures the information we have for the subshots we do *not* select, which is equivalent to the relevance we want to measure.

Relation to Determinantal Point Process. In the limit, it might seem that the relevance function will encourage coverage of the entire video in the summary because it does not necessarily preclude inclusion of subshots which are very similar. If this happens, the summary will contain identical or very similar (redundant) segments, which are not indicative of a good summary. However, it turns out because of a special property of our relevance function, inclusion of such redundant frames will be discouraged in the summary. Note that the second term of our objective M has the same functional form as the well-known determinantal point processes (DPPs)(Kulesza and Taskar, 2012). Proposed by Kulesza and Taskar (Kulesza and Taskar, 2012), DPPs use the log determinant function to measure the volume spanned by columns of of a subset $\mathcal{S}$ in $\mathcal{V}$. Recent research independent of our work has also shown encouraging results on standard video summarization tasks (Gong et al., 2014). It is often used as a means to devise tractable algorithms to measure (and also optimize) diversity in a given set, because maximizing the determinant encourages a bigger volume which in turn implies that the columns of $\mathcal{S}$ are close to orthogonal or uncorrelated which encourages diversity in the elements of $\mathcal{S}$. As a result, our objective function M not only measures relevance but also implicitly encourages diversity in the obtained summary.

6.3.2 Attention Measurement using Gaze Fixations

For egocentric videos, another important source of information is the user's point of interest in the video. Past research (Lee et al., 2012; Lu and Grauman, 2013; Gygli et al., 2014) has developed sophisticated strategies to estimate which regions are important and which frames convey useful information towards a good summary. However, if we have access to gaze information, we have an alternative to such complex preprocessing steps.

We explore how the pattern of a subject's gaze can inform the generation of meaningful summarizations. Here, for each subshot, we compute the attention score c_i by counting the number of frames containing fixations (after the subshot extraction pipline shown in Fig. 6.2). This is similar to the interestingness (Gygli et al., 2014) and importance (Lee et al., 2012) measure from recent work, though, our proposal is a more natural measurement of interest characterizing how much attention this subshot attracted from the user. We use this to define an additional term $I(\mathcal{S})$ in the objective as follows.

$$I(\mathcal{S}) = \sum_{i \in \mathcal{S}} c_i \tag{6.4}$$

6.3.3 Partition Matroid Constraint

In practical settings, we want our summary to reflect human preference in terms of subshot allocation. For instance, users usually want to allocate more subshots in a summary when more interesting things happen (e.g., in Disneyland) than when less interesting ones happen (e.g., in an office). An ideal summary should respect the compactness while maintaining a user-preferred distribution, if available.

To achieve this goal, we incorporate a partition matroid constraint into our model. First, we partition the video into b disjoint blocks $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_b$. Given the amount of user preference in each block, we specify an upper bound on the number of sub-shots that can be included from that block. Now, since each block pertains to a subset of subshots, and blocks are mutually disjoint, such a partitioning can be denoted as a matroid, $\mathcal{M} = (\mathcal{V}, \mathcal{P})$ where the set of subshots $\mathcal{V}$ is the ground set, and the blocks are 'independent' subsets of $\mathcal{V}$. We can associate each of the b blocks with an integer, $\{f_1, \cdots, f_b\}$ and ask that no more than f_i subshots be selected from each block. This requirement can be imposed by combinatorial structure known as the partition matroid (Fujishige, 2005), $\mathcal{I} = \{\mathcal{A} : \quad |\mathcal{A} \cap \mathcal{P}_m| \leqslant$

$f_m, m = 1, 2, \cdots, b\}$.

6.3.4 Full Model

Putting all the above pieces together, we can set up a simple model for summarization,

$$\begin{aligned} \max_{\mathcal{S}} \quad & \log(\det(L_{\mathcal{V}\setminus\mathcal{S}})) + \log(\det(L_{\mathcal{S}})) + \lambda \sum_{i\in\mathcal{S}} c_i \\ \text{s.t.} \quad & \mathcal{S} \in \mathcal{I} \end{aligned} \tag{6.5}$$

where λ is a positive trade-off coefficient and the feasible set $\mathcal{I}$ corresponds to the partition matroid introduced above. We denote our full objective function as $F(\mathcal{S})$.

6.4 Constrained Submodular Optimization

Our model in (6.5) captures various desirable properties of a good summary, but this constrained combinatorial problem is in general difficult to optimize directly. There are three pertinent issues: objective function, monotonicity, and constraints.

First, we analyze our objective function. The mutual information criteria in (6.1) is difficult to optimize globally. Fortunately, as shown in various recent works on submodular optimization (Krause et al., 2008), $M(\mathcal{S})$ in (6.3) is submodular. The other term is a linear sum over positive scalars, hence is also submodular. Since the sum of two submodular functions is also submodular, our objective function in (6.5) is submodular. Second, when we analyze the monotonicity properties, we see that the mutual information term makes the objective non-monotone. While monotone objectives have better approximation guarantees, non-monotone objectives allow us to provide a richer model. For instance, with a monotone

objective, we see that an upper bound constraint will always become tight (even if parts of the summary are redundant) because the model incurs no penalty for including an additional subshot. In contrast, our model prevents adding redundant subshots when the key information is already there (as shown in Fig. 6.5). Overall, our model is a constrained optimization model. Maximizing submodular functions subject to arbitrary linear constraints is very difficult. However, if the constraints are expressible as a knapsack constraint or a constraint over matroids, recent work (Lee et al., 2010; Filmus and Ward, 2012) from combinatorial optimization provides a variety of strategies.

Motivated by ideas from (Lee et al., 2010; Filmus and Ward, 2012), we propose a local search algorithm, which requires no rounding. All intermediate solutions are integral. There are three key local operations: add, swap, and delete. The key idea is to iteratively search over these possible operations until no improvement can be made. Alg. 4 outlines our algorithm. Our algorithm achieves an approximation factor with respect to the unknown optimal solution.

Proposition 2. *Alg. 4 achieves a $\frac{1}{4}$-approximation factor for our constrained submodular maximization problem (6.5) as long as $F(S) \geqslant 0$.*

Proof. Recall our objective $F(S) = M(S) + \lambda I(S)$ is submodular. Pick λ such that $F(S)$ is non-negative.

By the submodularity(Fujishige, 2005), we first have

$$F(\mathcal{A}) + F(\mathcal{B}) \geqslant F(\mathcal{A} \cup \mathcal{B}) + F(\mathcal{A} \cap \mathcal{B}), \quad \forall \mathcal{A} \subseteq \mathcal{V}, \mathcal{B} \subseteq \mathcal{V} \tag{6.6}$$

Following (Lee et al., 2010), we apply our Alg. 1 twice to get two local optimal solutions $S_1 = \text{argmax}_{local}\{F(S_1) : S_1 \in \mathcal{I}, S_1 \subseteq \mathcal{V}_1 = \mathcal{V}\}$, $S_2 = \text{argmax}_{local}\{F(S_2) : S_2 \in \mathcal{I}, S_2 \subseteq \mathcal{V}_2 = \mathcal{V} \backslash S_1\}$. And return the maximum from these two as our final solution: $S = \text{argmax}\{F(S_1), F(S_2)$. Given the

local optimality and by Lemma 2.5 from (Lee et al., 2010), we then have

$$2(1+\epsilon)F(\mathcal{S}_i) \geqslant F(\mathcal{S}_i \cup \mathcal{C}) + F(\mathcal{S}_i \cap \mathcal{C}), \quad \forall \mathcal{C} \subseteq \mathcal{I}, \quad |\mathcal{S}_i| = |\mathcal{C}|, \quad i = 1, 2 \tag{6.7}$$

Let $\mathcal{O}$ denote the unknown global optimal solution to the original problem $\max\{F(\mathcal{S}) : \mathcal{S} \in \mathcal{I}, \mathcal{S} \subseteq \mathcal{V}\}$. Let $\mathcal{O}_i = \mathcal{O} \cap \mathcal{V}_i, i = 1, 2$. We note $\mathcal{O}_1 = \mathcal{O} \cap \mathcal{V}_1 = \mathcal{O} \cap \mathcal{V} = \mathcal{O}$. With (6.7), we have

$$2(1+\epsilon)(F(\mathcal{S}_1) + F(\mathcal{S}_2)) \geqslant F(\mathcal{S}_1 \cup \mathcal{O}_1) + F(\mathcal{S}_1 \cap \mathcal{O}_1) + F(\mathcal{S}_2 \cup \mathcal{O}_2) + F(\mathcal{S}_2 \cap \mathcal{O}_2) \tag{6.8}$$

Since $F(\mathcal{S}) \geqslant F(\mathcal{S}_1), F(\mathcal{S}) \geqslant F(\mathcal{S}_2)$, we have

$$4(1+\epsilon)F(\mathcal{S}) \geqslant F(\mathcal{S}_1 \cup \mathcal{O}_1) + F(\mathcal{S}_1 \cap \mathcal{O}_1) + F(\mathcal{S}_2 \cup \mathcal{O}_2) + F(\mathcal{S}_2 \cap \mathcal{O}_2) \tag{6.9}$$

Using submodularity, we have

$$\begin{aligned}
&F(\mathcal{S}_1 \cup \mathcal{O}_1) + F(\mathcal{S}_2 \cup \mathcal{O}_2) + F(\mathcal{S}_1 \cap \mathcal{O}_1) \\
\geqslant &F(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{O}_1 \cup \mathcal{O}_2) + F((\mathcal{S}_1 \cup \mathcal{O}_1) \cap (\mathcal{S}_2 \cup \mathcal{O}_2)) + F(\mathcal{S}_1 \cap \mathcal{O}_1) \\
= &F(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{O}) + F(\mathcal{O}_2) + F(\mathcal{S}_1 \cap \mathcal{O}_1) \\
\geqslant &F(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{O}) + F(\mathcal{O}_2 \cup (\mathcal{S}_1 \cap \mathcal{O}_1)) + F(\mathcal{O}_2 \cap \mathcal{S}_1 \cap \mathcal{O}_1) \\
= &F(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{O}) + F(\mathcal{O}) + F(\emptyset) \\
= &F(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{O}) + F(\mathcal{O})
\end{aligned} \tag{6.10}$$

Putting (6.10) back into (6.9), we get

$$4(1+\epsilon)F(\mathcal{S}) \geqslant F(\mathcal{O}) + F(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{O}) + F(\mathcal{S}_2 \cap \mathcal{O}_2) \geqslant F(\mathcal{O}) \tag{6.11}$$

This concludes our proof. □

It is useful to point out that in (Gillenwater et al., 2012), an approximation algorithm for maximizing an unconstrained DPP (also non-monotone) was proposed. While the approximation factor in that work was looser

Algorithm 4 Local Search for Constrained Submodular Maximization

1: **Input:** $\mathcal{M} = (\mathcal{V}, \mathcal{I}), F, \epsilon \geqslant 0$
2: Initialize $\mathcal{S} \leftarrow \emptyset$;
3: **while** (Any of the following local operations applies, update $\mathcal{S}$ accordingly) **do**
4: **Add operation.** If $e \in \mathcal{V} \backslash \mathcal{S}$ such that $\mathcal{S} \cup \{e\}) \in \mathcal{I}$ and $F(\mathcal{S} \cup \{e\}) - F(\mathcal{S}) > \epsilon$, then $\mathcal{S} = \mathcal{S} \cup \{e\}$.
5: **Swap operation.** If $e_i \in \mathcal{S}$ and $e_j \in \mathcal{V} \backslash \mathcal{S}$ such that $(\mathcal{S} \backslash \{e_i\}) \cup \{e_j\} \in \mathcal{I}$ and $F((\mathcal{S} \backslash \{e_i\}) \cup \{e_j\}) - F(\mathcal{S}) > \epsilon$, then $\mathcal{S} = (\mathcal{S} \backslash \{e_i\}) \cup \{e_j\}$.
6: **Delete operation.** If $e \in \mathcal{S}$ such that $F(\mathcal{S} \backslash \{e\}) - F(\mathcal{S}) > \epsilon$, then $\mathcal{S} = \mathcal{S} \backslash \{e\}$.
7: **end while**
8: **return** $\mathcal{S}$;

than other known results, a salient property was that the algorithm was simple to implement and avoided the usual reduction to a multi-linear relaxation of the submodular function (Chekuri et al., 2011). Algorithm 4 can be viewed as a general case of the method in (Gillenwater et al., 2012). It is equally simple to implement and optimizes a similar (DPP type) objective function but permits inclusion of additional constraints (which is frequently difficult in submodular maximization).

6.5 Experimental Evaluations

We will begin by describing the datasets used in our experiments and follow it with a discussion of the other baselines used for comparison. Then, we will delve into the qualitative and quantitative evaluations of our approach.

6.5.1 Dataset Collection and Annotation

Given the type of data (egocentric videos + gaze) we require, there are very few existing publicly available benchmarks that can be used directly

for our purposes. Here, we make use of the GTEA-gaze+ dataset which is the only public dataset with video and gaze. In addition, we will also present results on a newly collected dataset (~ 15 hours).

GTEA-gaze+ dataset. This dataset is designed for action recognition, though it can be used for summarization as well. It consists of 30 videos, each of which includes a meal preparation recording and lasts 12 ~ 20 minutes. There are fine-grained action annotations available with this dataset. In addition, we ask human experts to generate summaries by grouping those action annotations, and asking them to select 5 ~ 15 group of consequent segments (referred to as events or blocks), which they think are appropriate to summarize each video. These group level annotations serve as our ground truth summary T. We present sample results on this dataset in Fig. 6.4.

EgoSum+gaze dataset. We collected a new dataset of videos, acquired by 5 subjects wearing our eye tracking devices to record their daily lives in an uncontrolled setting, along with associated gaze information. We used a pair of SMI eye-tracking glasses[1] and a Pupil eye-tracking device[2] to collect gaze information. Our collection has 21 videos, each lasting 15 min ~ 1.5 hour. To facilitate evaluations using this dataset, we obtained human annotations for the summary. To this end, we asked our subjects to select a set of events (blocks) in our videos. Each event constituted a sequence of similar subshots, and we assume that any one of them is an equally good representative of the event. This avoids unnecessarily penalizing an otherwise good summary which differs from the ground truth in the precise frame 'indexes' but is perfectly consistent in a semantic sense. We asked each wearer to select events (5 ~ 15) which in their opinion should be included in a good machine generated summary. These annotations serve as the gold standard for our evaluation.We present sample results on this dataset in Fig. 6.5 and Fig. 6.6.

[1] http://www.eyetracking-glasses.com
[2] http://pupil-labs.com

6.5.2 Subshot extraction and representation

A natural way to represent videos prior to summarization, is to extract subshots and compute their feature descriptors. We found that in general, this is challenging for egocentric videos since such videos are mostly continuous and therefore may not have a clear 'boundary' between shots. Fortunately, gaze turns out to be very useful for egocentric video temporal segmentation (see the first row of Fig. 6.2) which is used as follows.

We first extracted gaze tracking information (fixation, saccade or blink) for each frame (using our eye tracking device). Next, we removed frames with bad eye tracking data. This procedure provides 6000 ~ 9000 segments with fixations per one hour of video. We picked the centroid frame as the *key-frame* in each segment, and extracted a feature descriptor around the gaze region (100×100) on this frame using R-CNN (Girshick et al., 2014). We then computed the cosine similarity between each key-frame using, $\kappa(i,j) = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}$ where $\mathbf{x}_i$ refers to the R-CNN feature vector of keyframe i. Next, we grouped consequent segments into subshots (Fig. 6.2, second row) by thresholding the neighborhood similarity distance at 0.5, which yields around 800 subshots per hour of video. Next, we picked the center key-frame from each subshot, and computed a R-CNN feature descriptor on the whole frame. This is the final descriptor $\mathbf{v}_k$ for subshot k used for our summarization algorithm.

6.5.3 Baselines

We adopted four baseline algorithms that do not require training summaries but are widely used for comparison purposes in video summarization (Almeida et al., 2012; Khosla et al., 2013). The first two are uniform sampling and k-means without gaze information. Our next two baselines incorporate *our specific* subshot segmentation and feature extraction processes into the previous two methods, which we refer as uniform (our

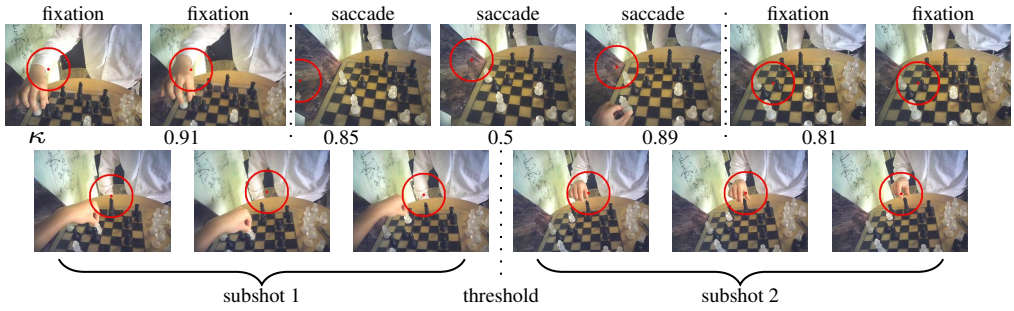

Figure 6.2: Illustration of our two-stage subshot extraction pipeline. First row: we extract gaze information for each frame, and filter frames with saccades/blinks and isolated fixations. Second row: we group similar neighbor key-frames by thresholding at 0.5 the similarities $\kappa(i, i+1)$ obtained from the gaze region.

subshots) and k-means (our subshots). In both cases, a (k subshot) summary is generated by selecting k equally separated subshots in uniform sampling and selecting the subshot closest to each of the k cluster centroid by k-means (reported by average of 20 runs). Unfortunately, direct comparisons to (Lee et al., 2012) and (Lu and Grauman, 2013) were not possible due to the lack of available training data.

6.5.4 Evaluation

For our experiments, we set $\epsilon = 1e^{-6}$ and $\lambda = 0.001$. To enforce the compactness criteria, we divided our video into equal-sized blocks by time (the number of blocks depends on the length of the video), and set $f_i = 1$ for the ith block. Though it worked well in practice, this step can be substituted by more sophisticated procedures if desired. To perform quantitative evaluations, we computed F-measure scores on all summaries by comparing it with ground truth summaries. If a subshot in an output summary lies in a block/event of ground truth subshots, we count it as correct for that algorithm. This allows us to compute precision (P) values

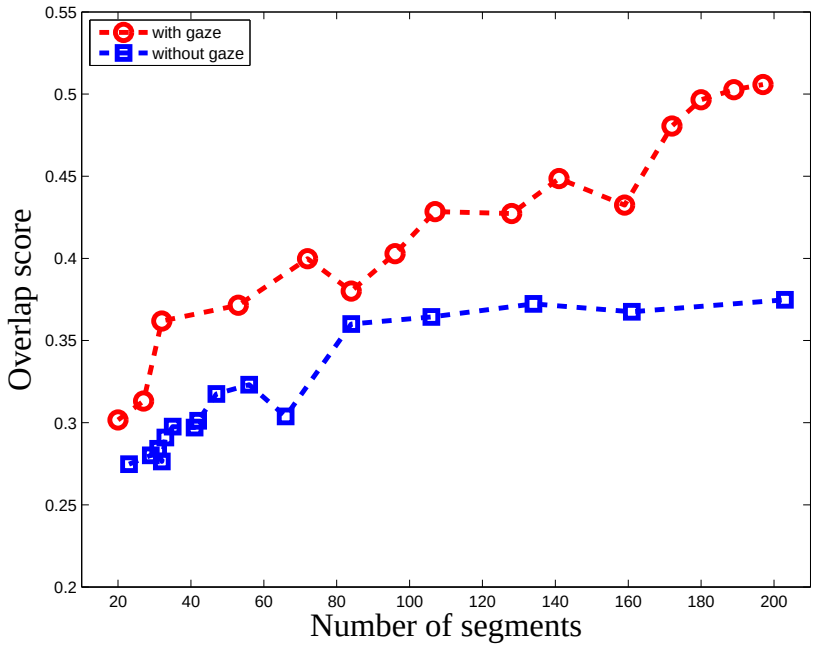

Figure 6.3: Comparison of temporal segmentation with/without gaze.

as $P = \frac{|S \cap T|}{|S|}$, where $S \cap T$ is the set of subshots from the summary which can be found in a ground truth annotation. Similarly, we computed the recall as $R = \frac{|S \cap T|}{|T|}$, where $|T|$ is equal to the number of events/blocks in the ground truth. The final F-measure is computed by $F = \frac{2PR}{P+R}$. Finally, note that the running time of our algorithm is 12 frames per second (fps), which is much faster than the time reported in (Lu and Grauman, 2013). Next, we analyze the goodness of the summary and the utility of gaze information, using our experimental results.

Is Gaze useful? We evaluated the utility of gaze in two different contexts, which is described next.

Temporal Segmentation. This is a critical preprocessing step, and is facilitated by the gaze information. To see how the quality of temporal segmentation is affected when gaze is *not* used, we performed temporal segmenta-

Figure 6.4: Results from GTEA-gaze+ data comparing the four baselines to our method (bottom row). This is from a pizza preparation video: cutting vegetables and meat (rows $1-2, 4$), frying (row 3), and adding sauce and toppings (rows $5-8$). Our algorithm picks these important subshots as they are with heavy attention.

tion on the GTEA-gaze+ dataset simply by partitioning the video sequence into subshots whenever similarity (using R-CNN from the full frame) between two consecutive frames falls above a threshold. We compared our gaze-based (see Fig. 6.2) approach with this method, by computing the overlap with ground truth action segmentation provided with this dataset. The results in Fig. 6.3 show that our gaze-enabled temporal segmentation dominates the baseline which suffers due to the fact that egocentric videos are continuous. In other words, if we agree with the premise that a good temporal segmentation will help *any* egocentric video summarization method, our experiments provide empirical evidence that gaze information will almost certainly improve summarization results (by generating better temporal segments).

Relevance of Summary with and without Gaze. We analyze this issue both qualitatively and quantitatively using GTEA-gaze+ dataset. Fig. 6.4 shows results for a pizza preparation video summary. As we can see, without

Method	uniform	kmeans	uniform (our subshots)	kmeans (our subshots)	ours
F-measure	0.161	0.215 ± 0.016	0.526	0.475 ± 0.026	0.621

Table 6.1: Comparisions of average F-measure on GTEA-GAZE+.

Method	uniform	kmeans	uniform (our subshots)	kmeans (our subshots)	ours
F-measure	0.080	0.095 ± 0.030	0.476	0.509 ± 0.025	0.585

Table 6.2: Comparisions of average F-measure on our new EgoSum+gaze dataset.

gaze information, uniform sampling and k-means pick many saccade frames (column 6 in the first two rows), which do not carry much content at all. However, when using gaze-enabled subshot segmentation and representation, these baselines benefit significantly. As we can see in rows 3 and 4, each subshot captures a useful amount of information. For a more quantitative measurement, we looked at the F-measure scores in Tab. 6.1 and 6.2. Cols 2/4 and 3/5, which show these values for both baselines, and our method. There is a significant improvement in the F-measure score whenever gaze is utilized, which is further improved using our proposed algorithm discussed next.

Quality of Our Summarization Algorithm. On the pizza preparation video 6.4, we observed that our method outperforms the other baselines, even when all methods utilize gaze information. As is evident, both uniform sampling and k-means, still include irrelevant subshots (row $1, 5$). Our summary, on the other hand, constitutes the key stages in the meal preparation procedure: cutting vegetables and meat (rows $1-2, 4$), frying (row 3), and adding sauce and toppings (rows $5-8$). These subshots are selected mainly due to the fact that subjects focused on them, and our objective $\mathrm{I}(\mathcal{S})$ picks these important subshots which contained substantial

Figure 6.5: Results from our new EgoSum+gaze dataset comparing the four baselines to our method (bottom row). In this video, our subject mixes a shake, drinks it, washes his cup, plays chess and texts a friend.

attention. Similar results can be seen with our new dataset (Fig. 6.5 and 6.6). As shown in row 2 and 4 in Fig. 6.5, in this uncontrolled setting, k-means ends up selecting many outliers. Also, row 3 in Fig. 6.6 shows uniform sampling fails when there are repeated scenes (e.g., washing dishes). Interestingly, our algorithm also achieves better compactness (fewer subshots in the summary) since our objective is non-monotone. The F-measures in Tab. 6.1 and 6.2 also provide evidence that our method using gaze outperforms the other baselines in both datasets.

Comparisons on other measures of quality. Note that we reported on F-measures here since they are widely used in computer vision. Separately, we also performed evaluations using summarization measures developed in the NLP literature such as ROUGE (Lin, 2004). The main sequence of steps here is for a human to (a) annotate the full video in text and then separately (b) write a summary of the video. Our algorithm generates a sub-shot summary which can be mapped to an English summary using the corresponding sentences in the human's full annotation of the video, i.e., (a) above. Here, we found that if the human's sentences/words are nearly similar between his/her summary and the full annotation (e.g.,

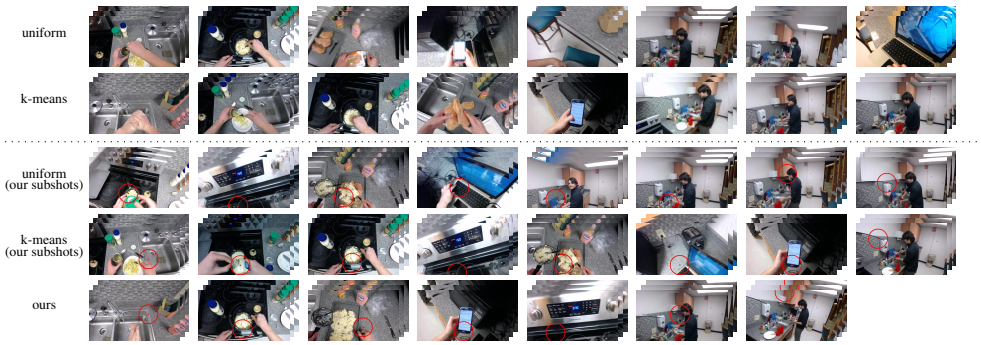

Figure 6.6: Results from our new EgoSum+gaze dataset comparing the four baselines to our method (bottom row). In this video, our subject is cooking chicken and have a conversation with his roommate.

as between (a) and (b)), then the system generated summary can indeed be meaningfully compared to the human's summary. In this case, the conclusions we can derive from ROUGE scores are similar to the ones we reported here using F-measure in Tab. 6.1 and 6.2. On the other hand, if the human's language usage in the full annotation and the summary written by him/her are different (in terms of word usage), then these scores are not very informative.

6.6 Summary

This chapter introduced a new approach for egocentric video summarization, utilizing gaze information. We give strong results showing that gaze provides the means to "personalize" the summary, by focusing on what is important to the camera wearer. We formulate the corresponding summarization objective as a submodular function maximization that captures desirable and common-sense requirements of a summary such as relevance and diversity. The compactness property is enforced as a partition matroid constraint, and solved using a simple to implement local search

method which offers approximation guarantees. Our experiments show that gaze information universally improves the relevance of a summary. For our experiments, we acquired a large set of gaze enabled egocentric video sequences, which is potentially valuable for future work on this topic.

7 DISCUSSION AND FUTURE DIRECTIONS

7.1 Summary of Contributions

The ultimate goal of this dissertation is to advance the state-of-the-art in reliably and efficiently parsing arbitrary images and videos and labeling them with a semantic meaning when possible. To achieve this objective, we make use of weak human supervision and weakly labeled data, and tackle the visual parsing problem from five perspectives in a bottom-up manner:

- We established a framework based on discrete calculus which unifies the contour completion and segmentation settings. Our method allows segmenting multiple objects with user seeds and topological constraints. We demonstrated via experiments that our model finds salient contours across a large dataset, offering significant improvement over similar methods in terms of both speed and human effort to achieve a satisfactory segmentation.

- We presented an approach for semantic segmentation which is able to exploit weak labels in the form of image level tags when no pixel-wise labeling is available for training. We showed that this problem can be formulated as structured prediction in a graphical model with latent variables, which allows us to leverage standard algorithms with good theoretical guarantees. We demonstrated the superiority of our approach and showed improvements of 7% over the state-of-the-art for this task.

- We later extended the above setting with a unified model, which learns from various forms of weak supervision including image level tags, bounding boxes, and partial labels. Our approach is efficient in both training and testing. We demonstrated the effectiveness of our

approach on the challenging Sift-flow dataset and showed that our method outperforms the state-of-the-art by more than 10%.

- We developed an intuitive yet expressive online subspace updating scheme, GOSUS, for video segmentation. We showed that imposing priors on the residuals (i.e., foreground) improves subspace estimation and is robust to noisy observations. The key steps of our solution reduce to simple matrix operations, yielding real-time performance for several interesting problems in video analysis.

- We studied the role of gaze in egocentric video summarization, and created a submodular summarization model for parsing and distilling the content of egocentric videos. Our model captures commonsense properties of a good summary: relevance, diversity, compactness, and personalization. We acquired a large set of gaze enabled egocentric video sequences, which is potentially valuable for future work on this topic. Our experiments showed that gaze information universally improves the relevance of a summary.

7.2 Future Directions

My long term research goal is to build a system which enables a human subject to wear a ego-centric camera as shown in Fig. 7.1, and our algorithm reliably and efficiently parses all of the visual data captured by that camera. Towards this goal, I would like to make use of larger quantities of weakly labeled data by joint visual and textual parsing on social media. Meanwhile, I want to continue my research on ego-centric vision for social good: e.g, disease diagnoses, assistance to the aging and disabled segments of the population.

7.2.1 Joint Visual and Textual Parsing

Social media data provides much richer information than merely tags. Visual data and textual data complement each other in knowledge extraction, when they are presented simultaneously. Text accompanying with images and videos provides high level visual "concept/guidance", which naturally bridges the semantic gap in visual parsing. Meanwhile, structural content exhibited in images and videos makes the intended sense of text much clearer. This motivates the problem of parsing visual and textual data jointly.

Concepts, as well as structural and geometric information from these concepts are presented in visual and textual data. An important question here is to build a correspondence of these concepts in images (videos) and sentences. We may also want to find and exploit structural and geometric properties between these concepts. In an ongoing collaboration, we are planning to harvest the shared knowledge on social media and build a massive knowledge base using such data from Flickr and YouTube.

With such a knowledge base in hand, I will be positioned to tackle more challenging problems. In my previous research, we have developed video summarization algorithms (Xu et al., 2015a). In my future research, I want to develop algorithms which automatically generate video trailers while preserving key plots. I would also like to design methods to produce text summaries for daily life videos.

7.2.2 Egocentric Vision for Social Good

As part of the central nervous system, the eye is among the most important sensors in our daily lives. Eye movements (fixation/saccade/blink) provide a natural interface for human-computer interaction. Such interaction can be easily incorporated into interactive object segmentation (Xu et al., 2013a). Further, eye movements convey human intention, which can be

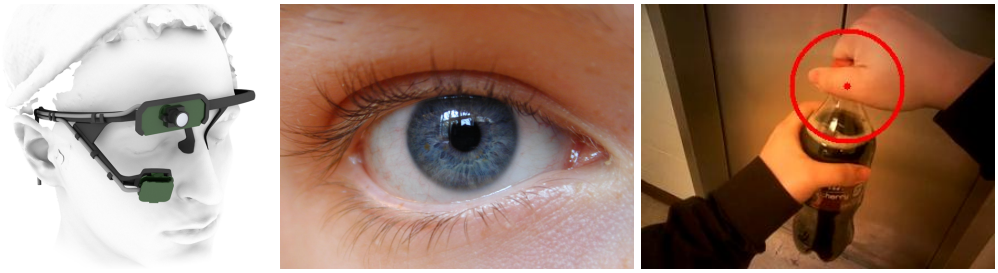

Figure 7.1: Overview of our eye tracking device (left) with the inside camera recording eye movements (middle), and outside camera recoding world scene with gaze mapping (right).

further used for more advanced mind reading applications.

In my thesis research, I have been developing user interfaces for our eye tracking device shown in Fig .7.1. This device uses two cameras on a eye-glass frame: one facing outside to record the world scene and one facing inside to record eye movements. This hardware together with our software system provides highly reliable gaze estimation, with great mobility. Hence, we are able to deploy our system in a practical setting. We aim to find early visual problems that can predict incipient brain disease (e.g., dementia).

Moving forward, a severe difficulty suffered by the aging population (e.g., Alzheimer's Disease (AD) patients) as well as their family members is memory loss. With our egocentric vision techniques, we may be able to provide a memory aid for the aging population. We can imagine a hardware-software approach to help them recall what they have seen/done in the past days, by summarizing their daily life events, and presenting these summaries of these scene to them at a daily base. We further want to help them to recognize their family members, when their memory fails. Our visual parsing technique can also be employed on wearable camera as an 'artificial eye'. This way, the blind can sense the world, with the help of such a system.

REFERENCES

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

Aghazadeh, Omid, Josephine Sullivan, and Stefan Carlsson. 2011. Novelty detection from an ego-centric perspective. In *Proc. CVPR*.

Almeida, Jurandy, Neucimar J. Leite, and Ricardo da Silva Torres. 2012. VISON: VIdeo Summarization for ONline applications. *Pattern Recognition Letters* 33(4):397–409.

Alpert, Sharon, Meirav Galun, Ronen Basri, and Achi Brandt. 2007. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *Proc. CVPR*.

Arbelaez, P., M. Maire, C. Fowlkes, and J. Malik. 2009. From contours to regions: An empirical evaluation. In *Proc. CVPR*.

Arbelaez, P., J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. 2014. Multiscale Combinatorial Grouping. In *Proc. CVPR*.

Arbelaez, Pablo, Michael Maire, Charless Fowlkes, and Jitendra Malik. 2011. Contour detection and hierarchical image segmentation. *PAMI* 33(5):898–916.

Arias, T., A. Edelman, and S. Smith. 1998. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications* 20:303–353.

Arora, Sanjeev, Rong Ge, Ravi Kannan, and Ankur Moitra. 2012. Computing a nonnegative matrix factorization – provably. In *Proc. STOC*.

Bai, Xue, and Guillermo Sapiro. 2009. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV* 82(2):113–132.

Balzano, Laura, Robert Nowak, and Benjamin Recht. 2010. Online identification and tracking of subspaces from highly incomplete information. In *Proceedings of the allerton conference on communication*.

Borenstein, Eran, and Shimon Ullman. 2002. Class-specific, top-down segmentation. In *Proc. ECCV*.

Borji, Ali, Dicky N Sihite, and Laurent Itti. 2012. Salient object detection: A benchmark. In *Proc. ECCV*.

Boyd, Stephen, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3(1):1–122.

Boykov, Y., and M.-P. Jolly. 2001a. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *Proc. ICCV*.

Boykov, Yuri, Olga Veksler, and Ramin Zabih. 2001. Fast approximate energy minimization via graph cuts. *TPAMI* 23(11):1222–1239.

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

Bucak, S, and Bilge Gunsel. 2009. Incremental subspace learning via non-negative matrix factorization. *Pattern Recog.* 42(5):788–797.

Cai, Deng, Xiaofei He, and Jiawei Han. 2007. Spectral regression for efficient regularized subspace learning. In *Proc. ICCV*.

Candès, Emmanuel J., Xiaodong Li, Yi Ma, and John Wright. 2011. Robust principal component analysis? *J. ACM* 58(3):11.

Candès, Emmanuel J., and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* 9(6): 717–772.

Cardinal, G., X. Boix, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez. 2010. Harmony Potentials for Joint Classification and Segmentation. In *Proc. CVPR*.

Carreira, J., F. Li, and C. Sminchisescu. 2012. Object Recognition by Sequential Figure-Ground Ranking. *IJCV* 98(3):243–262.

Carreira, Joao, and Cristian Sminchisescu. 2010. Constrained parametric min-cuts for automatic object segmentation. In *Proc. CVPR*.

Chauve, Anne-Laure, Patrick Labatut, and Jean-Philippe Pons. 2010. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *Proc. CVPR*.

Chekuri, Chandra, Jan Vondrák, and Rico Zenklusen. 2011. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proc. STOC*.

Chen, C., D. Freedman, and C.H. Lampert. 2011a. Enforcing topological constraints in random field image segmentation. In *Proc. CVPR*.

Chen, L. C., S. Fidler, A. Yuille, and R. Urtasun. 2014. Beat the MTurkers: Automatic Image Labeling from Weak 3D Supervision. In *Proc. CVPR*.

Chen, Xi, Qihang Lin, Seyoung Kim, Jaime G. Carbonell, and Eric P. Xing. 2011b. Smoothing proximal gradient method for general structured sparse learning. In *Proc. UAI*.

Chen, Xinlei, Abhinav Shrivastava, and Abhinav Gupta. 2013. Neil: Extracting visual knowledge from web data. In *Proc. ICCV*.

Collins, Maxwell D., Ji Liu, Jia Xu, Lopamudra Mukherjee, and Vikas Singh. 2014. Spectral Clustering with a Convex Regularizer on Millions of Images. In *Proc. ECCV*.

Collins, Maxwell D., Jia Xu, Leo Grady, and Vikas Singh. 2012. Random Walks based Multi-Image Segmentation: Quasiconvexity Results and GPU-based Solutions. In *Proc. CVPR*.

De La Torre, F., and M.J. Black. 2003. A framework for robust subspace learning. *IJCV* 54(1):117–142.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. CVPR*.

Dollar, P., Z. Tu, and S. Belongie. 2006. Supervised learning of edges and object boundaries. In *Proc. CVPR*.

Eigen, David, and Rob Fergus. 2012. Nonparametric image parsing using adaptive neighbor sets. In *Proc. CVPR*.

El-Zehiry, Noha Youssry, and Leo Grady. 2010. Fast global optimization of curvature. In *Proc. CVPR*.

Ellouze, Mehdi, Nozha Boujemaa, and Adel M. Alimi. 2010. Im(s)^2: Interactive movie summarization system. *JVCIR* 21(4):283–294.

Endres, Ian, and Derek Hoiem. 2014. Category-independent object proposals with diverse ranking. *PAMI* 36(2).

Estrada, Francisco J., and Allan D. Jepson. 2006. Robust boundary detetion with adaptive grouping. In *Proc. POCV*.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2010. The pascal visual object classes (voc) challenge. *IJCV* 88(2):303–338.

Everingham, M., J. Sivic, and A. Zisserman. 2006. "Hello! My name is Buffy" – automatic naming of characters in TV video. In *Proc. BMVC*.

Farabet, Clément, Camille Couprie, Laurent Najman, and Yann LeCun. 2012. Scene parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers. In *Proc. ICML*.

Fathi, Alireza, Ali Farhadi, and James M. Rehg. 2011a. Understanding egocentric activities. In *Proc. ICCV*.

Fathi, Alireza, Yin Li, and James M. Rehg. 2012. Learning to recognize daily actions using gaze. In *Proc. ECCV*.

Fathi, Alireza, Xiaofeng Ren, and James M Rehg. 2011b. Learning to recognize objects in egocentric activities. In *Proc. CVPR*.

Favaro, Paolo, René Vidal, and Avinash Ravichandran. 2011. A closed form solution to robust subspace estimation and clustering. In *Proc. CVPR*.

Felzenszwalb, P. F., R. B. Girshick, D. McAllester, and D. Ramanan. 2010. Object detection with discriminatively trained part-based models. *PAMI* 32(9):1627–1645.

Filmus, Yuval, and Justin Ward. 2012. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *Proc. FOCS*.

Fleckner, Oscar L. 1968. A method for the computation of the fresnel integrals and related functions. *Mathematics of Computation* 22(103):635–640.

Fragkiadaki, Katerina, Pablo Arbeláez, Panna Felsen, and Jitendra Malik. 2015. Learning to segment moving objects in videos. In *Proc. CVPR*.

Fu, Yanwei, Yanwen Guo, Yanshu Zhu, Feng Liu, Chuanming Song, and Zhi-Hua Zhou. 2010. Multi-view video summarization. *IEEE Transactions on Multimedia* 12(7):717–729.

Fujishige, Satoru. 2005. *Submodular functions and optimization*, vol. 58. Elsevier.

Gillenwater, Jennifer, Alex Kulesza, and Ben Taskar. 2012. Near-optimal map inference for determinantal point processes. In *Proc. NIPS*.

Girshick, Ross B., Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic segmentation. In *Proc. CVPR*.

Goldman, Dan B., Brian Curless, David Salesin, and Steven M. Seitz. 2006. Schematic storyboarding for video visualization and editing. *SIGGRAPH* 25(3):862–871.

Gong, Boqing, Wei-Lun Chao, Kristen Grauman, and Fei Sha. 2014. Diverse sequential subset selection for supervised video summarization. In *Proc. NIPS*.

Grady, Leo. 2006. Random walks for image segmentation. *PAMI* 28(11): 1768–1783.

———. 2010. Minimal Surfaces Extend Shortest Path Segmentation Methods to 3D. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(2):321–334.

Grady, Leo, and Jonathan R. Polimeni. 2010. *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer.

Gu, C., J. J. Lim, P. Arbelaez, and J. Malik. 2009. Recognition using region. In *Proc. CVPR*.

Guillaumin, Matthieu, Daniel Kuttel, and Vittorio Ferrari. 2014. Imagenet auto-annotation with segmentation propagation. *IJCV* 110(3):328–348.

Gulshan, Varun, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. 2010. Geodesic star convexity for interactive image segmentation. In *Proc. CVPR*.

Gygli, Michael, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. Creating summaries from user videos. In *Proc. ECCV*.

Hamm, Jihun, and Daniel D Lee. 2008. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proc. ICML*.

Hariharan, B., P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. 2011. Semantic contours from inverse detectors. In *Proc. ICCV*.

Hariharan, Bharath, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. 2014. Simultaneous detection and segmentation. In *Proc. ECCV*.

He, Jun, Laura Balzano, and Arthur Szlam. 2012. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *Proc. CVPR*.

Hochbaum, Dorit S, and Vikas Singh. 2009. An efficient algorithm for co-segmentation. In *Proc. ICCV*.

Horn, B. 1983. The curve of least energy. *ACM Trans. Math. Soft.* 9(4): 441–460.

Huang, Junzhou, Xiaolei Huang, and Dimitris N. Metaxas. 2009. Learning with dynamic group sparsity. In *Proc. ICCV*.

Huang, Junzhou, and Tong Zhang. 2010. The benefit of group sparsity. *Annals of Statistics* 38:1978–2004.

Hyvärinen, Aapo, and Erkki Oja. 2000. Independent component analysis: algorithms and applications. *Neural Networks* 13(4):411–430.

Iyer, Rishabh, and Jeff Bilmes. 2013. Submodular optimization with submodular cover and submodular knapsack constraints. In *Proc. NIPS*.

Jermyn, I.H., and H. Ishikawa. 2001. Globally optimal regions and boundaries as minimum ratio weight cycles. *PAMI* 23(10):1075–1088.

Jia, Kui, Tsung-Han Chan, and Yi Ma. 2012. Robust and practical face recognition via structured sparsity. In *Proc. ECCV*.

Jia, Yangqing. 2013. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/.

Joulin, Armand, Francis Bach, and Jean Ponce. 2012. Multi-class cosegmentation. In *Proc. CVPR*.

Kassner, Moritz, and William Patera. 2012. Pupil: Constructing the space of visual attention. Master thesis, Massachusetts Institute of Technology, Cambridge, MA.

Khosla, Aditya, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. 2013. Large-scale video summarization using web-image priors. In *Proc. CVPR*.

Kim, Gunhee, and Eric P. Xing. 2012. On multiple foreground cosegmentation. In *Proc. CVPR*.

Kim, Hyunwoo, Jia Xu, Baba C. Vemuri, and Vikas Singh. 2015. Manifold-valued Dirichlet Process. In *Proc. ICML*.

Kimia, Benjamin B., Ilana Frankel, and Ana-Maria Popescu. 2003. Euler spiral for shape completion. *IJCV* 54(1-3):159–182.

Klein, Stefan, Josien P. W. Pluim, Marius Staring, and Max A. Viergever. 2009. Adaptive stochastic gradient descent optimisation for image registration. *IJCV* 81(3):227–239.

Kovalevsky, V. A. 1989. Finite topology as applied to image analysis. *Computer Vision, Graphics, Image Processing* 46(2):141–161.

Krause, Andreas, and Carlos Guestrin. 2011. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology* 2(4):32.

Krause, Andreas, Ajit Paul Singh, and Carlos Guestrin. 2008. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR* 9:235–284.

Kuhlmann, Gregory, Peter Stone, Raymond Mooney, and Jude Shavlik. 2004. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. *The AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*.

Kulesza, A., and B. Taskar. 2012. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning* 5(2–3).

Ladický, L., C. Russell, P. Kohli, and P. H. S. Torr. 2010. Graph Cut based Inference with Co-occurrence Statistics. In *Proc. ECCV*.

Ladicky, Lubor, Paul Sturgess, Karteek Alahari, Chris Russell, and P. H. S. Torr. 2010. What, where and how many? combining object detectors and crfs. In *Proc. ECCV*.

Lafferty, J., A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for segmenting and labeling sequence data. In *Proc. ICML*.

Laganiere, Robert, Raphael Bacco, Arnaud Hocevar, Patrick Lambert, GrÃ©gory PaÃ¯s, and Bogdan Ionescu. 2008. Video summarization from spatio-temporal features. In *ACM TRECVid Video Summarization workshop*.

Le, Quoc V., Will Y. Zou, Serena Y. Yeung, and Andrew Y. Ng. 2011. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proc. CVPR*.

Lee, Jon, Vahab S. Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. 2010. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM J. Discrete Math.* 23(4):2053–2078.

Lee, Yong Jae, Joydeep Ghosh, and Kristen Grauman. 2012. Discovering important people and objects for egocentric video summarization. In *Proc. CVPR*.

Levinshtein, Alex, Cristian Sminchisescu, and Sven Dickinson. 2010. Optimal contour closure by superpixel grouping. In *Proc. ECCV*.

Levinshtein, Alex, Adrian Stere, Kiriakos Kutulakos, David Fleet, Sven Dickinson, and Kaleem Siddiqi. 2009. Turbopixels: Fast superpixels using geometric flows. *PAMI* 31(12):2290–2297.

Li, Cheng, and Kris M. Kitani. 2013. Model recommendation with virtual probes for ego-centric hand detection. In *Proc. ICCV*.

Li, Liyuan, Weimin Huang, Irene Y. H. Gu, and Qi Tian. 2004. Statistical modeling of complex backgrounds for foreground object detection. *TIP* 13(11):1459–1472.

Li, Stan Z., Xiaoguang Lu, XinWen Hou, Xianhua Peng, and QianSheng Cheng. 2005. Learning multiview face subspaces and facial pose estimation using independent component analysis. *TIP* 14(6):705–712.

Li, Yin, Alireza Fathi, and James M. Rehg. 2013. Learning to predict gaze in egocentric video. In *Proc. ICCV*.

Li, Yin, Xiaodi Hou, Christian Koch, James M Rehg, and Alan L Yuille. 2014. The secrets of salient object segmentation. In *Proc. CVPR*.

Lin, Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

Lin, Hui, and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proc. ACL*.

———. 2012. Learning mixtures of submodular shells with application to document summarization. In *Proc. UAI*.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proc. ECCV*.

Lin, Zhouchen, Minming Chen, Leqin Wu, and Yi Ma. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv* 1009.5055.

Liu, C., J. Yuen, and A. Torralba. 2011. Nonparametric Scene Parsing via Label Transfer. *PAMI* 33(12):2368–2382.

Liu, David, Gang Hua, and Tsuhan Chen. 2010. A hierarchical visual model for video object summarization. *TPAMI* 32(12):2178–2190.

Liu, Si, Shuicheng Yan, Tianzhu Zhang, Changsheng Xu, Jing Liu, and Hanqing Lu. 2012. Weakly supervised graph propagation towards collective image parsing. *IEEE Transactions on Multimedia* 14(2):361–373.

Liu, Yuzong, Kai Wei, Katrin Kirchhoff, Yisong Song, and Jeff Bilmes. 2013. Submodular feature selection for high-dimensional acoustic score spaces. In *Proc. ICASSP*.

Lu, C., L.J. Latecki, N. Adluru, X. Yang, and H. Ling. 2009. Shape guided contour grouping with particle filters. In *Proc. ICCV*.

Lu, Zheng, and Kristen Grauman. 2013. Story-driven summarization for egocentric video. In *Proc. CVPR*.

Luo, W., A. Schwing, and R. Urtasun. 2013. Latent structured active learning. In *Proc. NIPS*.

Ma, Yu-Fei, Xian-Sheng Hua, Lie Lu, and HongJiang Zhang. 2005. A generic framework of user attention model and its application in video summarization. *IEEE Transactions on Multimedia* 7(5):907–919.

Maclin, Richard, and Jude W Shavlik. 1994. Incorporating advice into agents that learn from reinforcements. *Proceedings of the National Conference on Artificial Intelligence* 694–694.

———. 1996. Creating advice-taking reinforcement learners. *Machine Learning* 22(1):251–281.

Mairal, J., M. Leordeanu, F. Bach, M. Hebert, et al. 2008. Discriminative sparse image models for class-specific edge detection and image interpretation. In *Proc. ECCV*.

Mairal, Julien, Francis Bach, J. Ponce, and G. Sapiro. 2010. Online learning for matrix factorization and sparse coding. *JMLR* 11:19–60.

Mairal, Julien, Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. 2011. Convex and network flow optimization for structured sparsity. *JMLR* 12:2681–2720.

Maire, M., P. Arbeláez, C. Fowlkes, and J. Malik. 2008. Using contours to detect and localize junctions in natural images. In *Proc. CVPR*.

Maji, S., N.K. Vishnoi, and J. Malik. 2011. Biased normalized cuts. In *Proc. CVPR*.

Martin, David R., Charless Fowlkes, and Jitendra Malik. 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI* 26(5):530–549.

Massey, William S. 1991. *A basic course in algebraic topology*, vol. 127. Springer Science & Business Media.

Mishra, Ajay K., Yiannis Aloimonos, Loong Fah Cheong, and Ashraf A. Kassim. 2012. Active visual segmentation. *TPAMI* 34(4):639–653.

Mortensen, Eric N, and William A Barrett. 1998. Interactive segmentation with intelligent scissors. *Graphical models and image processing* 60(5):349–384.

Movahedi, Vida, and James H. Elder. 2010. Design and perceptual validation of performance measures for salient object segmentation. In *Proc. POCV*.

Mukherjee, L., V. Singh, and J. Peng. 2011. Scale invariant cosegmentation for image groups. In *Proc. CVPR*.

Mukherjee, Lopamudra, Vikas Singh, and Charles R Dyer. 2009. Half-integrality based algorithms for cosegmentation of images. In *Proc. CVPR*.

Mukherjee, Lopamudra, Vikas Singh, Jia Xu, and Maxwell D. Collins. 2012. Analyzing the subspace structure of related images: Concurrent segmentation of image sets. In *Proc. ECCV*.

Nemhauser, George L, and Laurence A Wolsey. 1988. *Integer and combinatorial optimization*, vol. 18. Wiley New York.

Ngo, Chong-Wah, Yu-Fei Ma, and HongJiang Zhang. 2003. Automatic video summarization by graph modeling. In *Proc. ICCV*.

Nocedal, Jorge, and Stephen Wright. 2006. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering.

Nowozin, Sebastian, and Christoph Lampert. 2010. Global interactions in random field models: A potential function ensuring connectedness. *SIAM J. Imag. Sci.* 3(4):1048–1074.

Pandey, M., and S. Lazebnik. 2011. Scene Recognition and Weakly Supervised Object Localization with Deformable Part-Based Models. In *Proc. ICCV*.

Papandreou, George, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. 2015. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*.

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

Parent, P., and S.W. Zucker. 1989. Trace inference, curvature consistency, and curve detection. *PAMI* 11(8):823–839.

Pathak, Deepak, Philipp Krähenbühl, and Trevor Darrell. 2015. Constrained convolutional neural networks for weakly supervised segmentation. *arXiv preprint arXiv:1506.03648*.

Pathak, Deepak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. 2014. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144*.

Pinheiro, Pedro O, and Ronan Collobert. 2015. From image-level to pixel-level labeling with convolutional networks. In *Proc. CVPR*.

Pirsiavash, Hamed, and Deva Ramanan. 2012. Detecting activities of daily living in first-person camera views. In *Proc. CVPR*.

Pongnumkul, Suporn, Jue Wang, and Michael F. Cohen. 2008. Creating map-based storyboards for browsing tour videos. In *UIST*.

Potapov, Danila, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. 2014. Category-specific video summarization. In *Proc. ECCV*.

Qin, Zhiwei, and Donald Goldfarb. 2012. Structured sparsity via alternating directions methods. *JMLR* 13:1373–1406.

Quattoni, A., S. Wang, L.-P. Morency, M. Collins, and T. Darrell. 2007. Hidden-state Conditional Random Fields. *PAMI* 29(10):1848–1852.

Rantalankila, Pekka, Juho Kannala, and Esa Rahtu. 2014. Generating object segmentation proposals using global and local search. In *Proc. CVPR*.

Ren, X., C.C. Fowlkes, and J. Malik. 2005. Scale-invariant contour completion using conditional random fields. In *Proc. ICCV*.

Ren, Xiaofeng, and Chunhui Gu. 2010. Figure-ground segmentation improves handled object recognition in egocentric video. In *Proc. CVPR*.

Rockafellar, R. Tyrrell. 1970. *Convex analysis*. Princeton, NJ: Princeton University Press.

Rother, C., V. Kolmogorov, and A. Blake. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. *SIGGRAPH*.

Rother, Carsten, Thomas P. Minka, Andrew Blake, and Vladimir Kolmogorov. 2006. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *Proc. CVPR*.

Rubinstein, Michael, Ce Liu, and William T. Freeman. 2012. Annotation propagation in large image databases via dense image correspondence. In *Proc. ECCV*.

Russakovsky, Olga, Amy L Bearman, Vittorio Ferrari, and Fei-Fei Li. 2015. What's the point: Semantic segmentation with point supervision. *arXiv preprint arXiv:1506.02106*.

Russell, Bryan C, Antonio Torralba, Kevin P Murphy, and William T Freeman. 2008. Labelme: a database and web-based tool for image annotation. *IJCV* 77(1-3):157–173.

Ryoo, Michael S., and Larry Matthies. 2013. First-person activity recognition: What are they doing to me? In *Proc. CVPR*.

Schoenemann, Thomas, and Daniel Cremers. 2007. Introducing curvature into globally optimal image segmentation: Minimum ratio cycles on product graphs. In *Proc. ICCV*.

Schwing, A. G., T. Hazan, M. Pollefeys, and R. Urtasun. 2011. Distributed Message Passing for Large Scale Graphical Models. In *Proc. CVPR*.

———. 2012. Efficient Structured Prediction with Latent Variables for General Graphical Models. In *Proc. ICML*.

Schwing, A. G., and R. Urtasun. 2015. Fully Connected Deep Structured Networks. Http://arxiv.org/abs/1503.02351.

Shapovalova, Nataliya, Michalis Raptis, Leonid Sigal, and Greg Mori. 2013. Action is in the eye of the beholder: Eye-gaze driven model for spatio-temporal action localization. In *Proc. NIPS*.

Shotton, J., A. Blake, and R. Cipolla. 2005. Contour-based learning for object detection. In *Proc. ICCV*.

Shotton, J., M. Johnson, and R. Cipolla. 2008. Semantic texton forests for image categorization and segmentation. In *Proc. CVPR*.

Singh, Gautam, and Jana Kosecka. 2013. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *Proc. CVPR*.

Singh, Mohit, and Lap Chi Lau. 2007. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proc. STOC*.

Sivic, J., M. Everingham, and A. Zisserman. 2009. "Who are you?" – learning person specific classifiers from video. In *Proc. CVPR*.

Song, Hyun Oh, Ross B. Girshick, Stefanie Jegelka, Julien Mairal, Zaïd Harchaoui, and Trevor Darrell. 2014. One-bit object detection: On learning to localize objects with minimal supervision. In *Proc. ICML*.

Spriggs, Ekaterina H., Fernando De La Torre, and Martial Hebert. 2009. Temporal segmentation and activity classiï¬'cation from first-person sensing. In *IEEE Workshop on Egocentric Vision, CVPR*.

Sriperumbudur, B., and G. Lanckriet. 2009. On the convergence of the concave-convex procedure. In *Proc. NIPS*.

Stahl, Joachim S., and Song Wang. 2007. Edge grouping combining boundary and region information. *TIP* 16(10):2590–2606.

———. 2008. Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. *PAMI* 30(3):395–411.

Taskar, B., C. Guestrin, and D. Koller. 2003. Max-Margin Markov Networks. In *Proc. NIPS*.

Tighe, J., and S. Lazebnik. 2013a. Finding Things: Image Parsing with Regions and Per-Exemplar Detectors. In *Proc. CVPR*.

———. 2013b. Superparsing - Scalable Nonparametric Image Parsing with Superpixels. *IJCV* 101(2):329–349.

———. 2014. Scene Parsing with Object Instances and Occlusion Ordering. In *Proc. CVPR*.

Toyama, Kentaro, John Krumm, Barry Brumitt, and Brian Meyers. 1999. Wallflower: Principles and practice of background maintenance. In *Proc. ICCV*.

Truemper, Klaus. 1978. Algebraic characterizations of unimodular matrices. *SIAM Journal on Applied Mathematics* 35(2):328–332.

Tsochantaridis, I., T. Joachims, T. Hofmann, and Y. Altun. 2005. Large Margin Methods for Structured and Interdependent Output Variables. *JMLR*.

Tu, Z., X. Chen, A.L. Yuille, and S.C. Zhu. 2005. Image parsing: Unifying segmentation, detection, and recognition. *IJCV* 63(2):113–140.

Turaga, Pavan, Ashok Veeraraghavan, and Rama Chellappa. 2008. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *Proc. CVPR*.

Turk, M. A., and A. P. Pentland. 1991. Face recognition using eigenfaces. In *Proc. CVPR*, 586–591.

Ullman, Shimon, and Amnon Shaashua. 1988. Structural saliency: The detection of globally salient structures using a locally connected network. Tech. Rep., MIT.

Verbeek, J., and B. Triggs. 2007. Region classification with Markov field aspect models. In *Proc. CVPR*.

Vezhnevets, A. 2012. Weakly supervised semantic segmentation of natural images. Ph.D. thesis, ETH Zurich.

Vezhnevets, A., and J. M. Buhmann. 2010. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *Proc. CVPR*.

Vezhnevets, A., V. Ferrari, and J. M. Buhmann. 2011. Weakly supervised semantic segmentation with a multi image model. In *Proc. ICCV*.

———. 2012. Weakly Supervised Structured Output Learning for Semantic Segmentation. In *Proc. CVPR*.

Vicente, S., V. Kolmogorov, and C. Rother. 2008. Graph cut based image segmentation with connectivity priors. In *Proc. CVPR*.

———. 2010. Cosegmentation Revisited: Models and Optimization. In *Proc. ECCV*.

Vidal, René, Yi Ma, and Shankar Sastry. 2005. Generalized principal component analysis (GPCA). *PAMI* 27(12):1945–1959.

Viola, Paul A., and Michael J. Jones. 2004. Robust real-time face detection. *IJCV* 57(2):137–154.

Walton, D. J., and D. S. Meek. 2009. G1 interpolation with a single cornu spiral segment. *Journal of Computational and Applied Mathematics* 223(1): 86–96.

Wang, Heng, and Cordelia Schmid. 2013. Action recognition with improved trajectories. In *Proc. ICCV*.

Wang, Meng, Richang Hong, Guangda Li, Zheng-Jun Zha, Shuicheng Yan, and Tat-Seng Chua. 2012a. Event driven web video summarization by tag localization and key-shot identification. *IEEE Transactions on Multimedia* 14(4):975–985.

Wang, Naiyan, Tiansheng Yao, Jingdong Wang, and Dit-Yan Yeung. 2012b. A probabilistic approach to robust matrix factorization. In *Proc. ECCV*.

Wang, Song, Toshiro Kubota, Jeffrey Mark Siskind, and Jun Wang. 2005. Salient closed boundary extraction with ratio contour. *PAMI* 27(4):546–561.

Wang, T., AG Backhouse, and I.Y.H. Gu. 2008. Online subspace learning on grassmann manifold for moving object tracking in video. In *Int. conf. acoustics, speech, and signal processing*.

Wolf, Wayne. 1996. Key frame selection by motion analysis. In *Proc. ICASSP*.

Xia, Wei, Csaba Domokos, Jian Dong, Loong-Fah Cheong, and Shuicheng Yan. 2013. Semantic segmentation without annotating segments. In *Proc. ICCV*.

Xu, Jia. 2014. Joint Visual and Textual Mining on Social Media. In *International Conference on Data Mining PhD Forum*.

Xu, Jia, Maxwell D. Collins, and Vikas Singh. 2013a. Incorporating User Interaction and Topological Constraints within Contour Completion via Discrete Calculus. In *Proc. CVPR*.

Xu, Jia, Vamsi K. Ithapu, Lopamudra Mukherjee, James M. Rehg, and Vikas Singh. 2013b. GOSUS: Grassmannian Online Subspace Updates with Structured-sparsity. In *Proc. ICCV*.

Xu, Jia, Lopamudra Mukherjee, Yin Li, Jamieson Warner, James M. Rehg, and Vikas Singh. 2015a. Gaze-enabled Egocentric Video Summarization via Constrained Submodular Maximization. In *Proc. CVPR*.

Xu, Jia, Alexander G. Schwing, and Raquel Urtasun. 2014. Tell me what you see and i will show you where it is. In *Proc. CVPR*.

———. 2015b. Learning to Segment with Various Forms of Weak Supervision. In *Proc. CVPR*.

Yang, Jimei, Brian L. Price, Scott Cohen, and Ming-Hsuan Yang. 2014. Context Driven Scene Parsing with Attention to Rare Classes. In *Proc. CVPR*.

Yao, J., S. Fidler, and R. Urtasun. 2012. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Proc. CVPR*.

Yeung, S., A. Fathi, and L. Fei-Fei. 2014. Videoset: Video summary evaluation through text. *arXiv:1406.5824*.

Yu, C.-N., and T. Joachims. 2009. Learning structural SVMs with latent variables. In *Proc. ICML*.

Yu, Kai, Yuanqing Lin, and John Lafferty. 2011. Learning image representations from the pixel level via hierarchical sparse coding. In *Proc. CVPR*.

Yuan, Ming, and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B* 68:49–67.

Yuille, A. L., and A. Rangarajan. 2003. The Concave-Convex Procedure (CCCP). *Neural Computation*.

Yun, Kiwon, Yifan Peng, Dimitris Samaras, Gregory J. Zelinsky, and Tamara L. Berg. 2013. Studying relationships between human gaze, description, and computer vision. In *Proc. CVPR*.

Zeng, Yun, Dimitris Samaras, Wei Chen, et al. 2008. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation. *CVIU* 112:81–90.

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

Zhao, Bin, James Tin-Yau Kwok, and Changshui Zhang. 2009. Maximum Margin Clustering with Multivariate Loss Function. In *Proc. ICDM*.

Zhao, Bin, Fei Wang, and Changshui Zhang. 2008. Efficient multiclass maximum margin clustering. In *Proc. ICML*.

Zhao, Bin, and Eric P Xing. 2014. Quasi real-time summarization for consumer videos. In *Proc. CVPR*.

Zhu, Xiaojin. 2005. Semi-supervised learning literature survey.