Person analysis in stereoscopic movies

Guillaume Seguin

THÈSE DE DOCTORAT
de l'Université de recherche
Paris Sciences Lettres –
PSL Research University

préparée à
l'École normale supérieure

Analyse des personnes dans les films stéréoscopiques

Person analysis in stereoscopic movies

École doctorale n°386
Spécialité: Informatique
Soutenue le 29.04.2016

par Guillaume Seguin

Composition du Jury :

M Jason Corso
University of Michigan
Rapporteur

M Cristian Sminchisescu
Lund University
Rapporteur

M Ivan Laptev
Inria
Directeur de thèse

M Josef Sivic
Inria
Directeur de thèse

M Karteek Alahari
Inria
Membre du Jury

M Patrick Perez
Technicolor
Membre du Jury

M Jean Ponce
École normale supérieure
Membre du Jury

M Francis Bach
École normale supérieure
Membre du Jury

ENS ÉCOLE NORMALE SUPÉRIEURE
PSL RESEARCH UNIVERSITY PARIS

École normale supérieure
45 rue d'Ulm
75005 Paris

Inria Paris
2 rue Simone Iff
75012 Paris

UPMC
Ecole Doctorale de Sciences
Mathématiques de Paris Centre
4 place Jussieu
75252 Paris Cedex 05
Boite courrier 290

Abstract

Artificial intelligence is one of the grails of computer science, and in many cases it implies building systems which can understand the surrounding visual environment. Visual content is most often focused on people, which makes the analysis of people a challenge of great importance for computer vision. In addition, feature-length stereoscopic ("3D") movies are now widely available, providing large, varied sets of stereoscopic pairs which contain more information than standard color movies.

In this thesis, we study how we can exploit the additional information provided by 3D movies for person analysis. We first explore how to extract a notion of depth from stereo movies in the form of disparity maps. We then evaluate how person detection and human pose estimation methods perform on such data. Leveraging the relative ease of the person detection task in 3D movies, we develop a method to automatically harvest examples of persons in 3D movies and train a person detector for standard color movies using such automatically obtained training data.

We then focus on the task of segmenting multiple people in videos. We first propose a method to segment multiple people in 3D videos by combining cues derived from pose estimates with ones derived from disparity maps. We formulate the segmentation problem as an inference task in a multi-label Conditional Random Field that explicitly models occlusions between people. Our method produces a layered, multi-instance segmentation. We show the experimental effectiveness of this approach as well as its limitations.

We then propose a second model for multiple people segmentation. This model only relies on tracks of person detections and not on pose estimates. We formulate our problem as a convex optimization one, with the minimization of a quadratic cost under linear equality and inequality constraints. These constraints encode the weak localization information provided by person detections. This method does not explicitly require pose estimates or disparity maps but can integrate these additional cues. Our method can also be used for segmenting instances of other object classes from videos. We evaluate all these aspects and demonstrate the superior performance of this new method.

We demonstrate results on two newly collected datasets extracted from 3D movies, for training and testing of person detection, human pose estimation and video segmentation models. These datasets contain more than five thousand stereo pairs, one thousand person bounding boxes, five hundred person poses and one thousand person segmentation masks.

Résumé

L'intelligence artificielle est l'un des graals de l'informatique. Elle suppose dans de nombreux cas de construire des systèmes capables de comprendre l'environnement visuel qui les entoure. Les contenus visuels mettent la plupart du temps en scène des personnes, ce qui fait de l'analyse des personnes un défi d'importance majeure pour le succès de la vision par ordinateur. Par ailleurs, les films stéréoscopiques ("3D") sont maintenant largement distribués, fournissant d'énormes collections très variées de paires stéréoscopiques qui contiennent plus d'information qu'une image de film classique.

Dans cette thèse, nous étudions comment exploiter les données additionnelles issues des films 3D pour les tâches d'analyse des personnes. Nous explorons tout d'abord comment extraire une notion de profondeur à partir des films stéréoscopiques, sous la forme de cartes de disparité. Nous évaluons ensuite à quel point les méthodes de détection de personne et d'estimation de posture peuvent bénéficier de ces informations supplémentaires. En s'appuyant sur la relative facilité de la tâche de détection de personne dans les films 3D, nous développons une méthode de supervision automatique pour collecter automatiquement des exemples de personnes dans les films 3D afin d'entraîner un détecteur de personne pour les films non 3D.

Nous nous concentrons ensuite sur la segmentation de plusieurs personnes dans les vidéos. Nous proposons tout d'abord une méthode pour segmenter plusieurs personnes dans les films 3D en combinant des informations dérivées des cartes de profondeur avec des informations dérivées d'estimations de posture. Nous formulons ce problème comme un problème d'étiquetage de graphe multi-étiquettes, et nous modélisons explicitement les occlusions pour produire une segmentation multi-instance par plan. Après avoir montré l'efficacité et les limitations de cette méthode, nous proposons un second modèle, qui ne repose lui que sur des détections de personne à travers la vidéo, et pas sur des estimations de posture. Nous formulons un problème d'optimisation convexe, en tant que minimisation d'un coût quadratique sous contraintes linéaires. Ces contraintes encodent les informations de localisation fournies par les détections de personne. Cette méthode ne nécessite pas d'information de posture ou des cartes de disparité, mais peut facilement intégrer ces signaux supplémentaires. Elle peut également être utilisée pour segmenter des instances d'autres classes d'objets dans les vidéos. Nous évaluons tous ces aspects et démontrons la performance de cette nouvelle méthode.

Cette thèse présente également deux nouveaux jeux de données extraits de films 3D, permettant d'entraîner et d'évaluer les méthodes de détection de personne, d'estimation de posture humain et de segmentation vidéo. Ces jeux de données contiennent plus de 5000 paires stéréo, 1000 annotations pour la détection de personne, 500 annotations de pose et 1000 masques de segmentations fins.

Acknowledgement

I would first like to thank my PhD advisors Josef Sivic and Ivan Laptev. We had numerous projects across these years, and in both success and failures you kept being supportive and motivated which helped me a lot. Your knowledge of computer vision has been invaluable, as well as your endless will to push the boundary even further.

I thank Jason Corso and Cristian Sminchisescu for accepting the role of rapporteurs of my thesis, as well as Karteek Alahari, Francis Bach, Patrick Perez and Jean Ponce for taking part in my jury.

I also thank Jean Ponce for his help in proofreading my papers. Always iterating, always improving, you greatly helped me learn how to properly formulate my ideas and articulate a scientific paper. I will probably never reach your standards of perfection, but I will keep trying hard.

During these almost 5 years of internships and PhD studies, I have had the chance to meet, work and chat with awesome people in the WILLOW and SIERRA project teams. Our lab was a truly wonderful workplace and could not have been more stimulating. I would first like to thank Piotr and Rémi for their friendship, their advises, their help, their focus, everything. Our collaborations and discussions helped me go further, much further than I would have otherwise.

I thank Karteek Alahari for our excellent collaboration and his mentorship. You started the project which occupied me for more than one year, and together we made it a consistent, complete work. This experience has completely taught me how to do research and how to work in a team.

I would also like to thank Olivier Duchenne who I consider as a true mentor. Your endless motivation and your passion for theoretical and practical science and computer vision highly stimulated me, and near you I was able to quickly learn a large chunk of the literature of our field. More anecdotal, but your MATLAB tricks kept helping me speed up my code across the years, sparing numerous cluster CPU hours. Even today our discussions are still highly valuable to me.

I also thank my other office mates, Florent, Fajwel, Vincent and Antoine, as well as all the other members from WILLOW and SIERRA for their kindness and all our discussions. There has been so many of them during these 5 years that I will not even try listing them all for fear of forgetting someone.

I would not have been able to produce that many results without my dear collaborators `sequoia` and `meleze`. You literally saved me a lifetime of waiting, even though I often had to take great care of you.

I would like to thank "les amis d'UlmInfo": Antoine, Guillaume, Jacques-Henri, Louis, Lucas, Marc, Michäel, Nicolas, Pablo, Pierre, Stéphane. In labs all around the world we have been sharing the PhD studies experience. Sun never sets on #ulminfo, and I hope our friendship never will.

One story ends, another starts: I would like to thank my partner Arnaud for the awesome adventure we are starting. Without you I would not have dared building such a project. Your motivation gives me confidence, and you are giving me the chance to stay very close to the research world, while using our findings in extremely practical applications.

I thank my parents Catherine and François, my sister Valérie and my brother Thomas, for their endless support and love, as well as my wife's family.

Last but not least, my thoughts and thanks go to my wife Amandine. She has been supporting me during my years of PhD, even during the very chaotic, almost sleepless, deadline months. Thank you so much for accepting my weird lifestyle and my never ending busyness. Your love is my strength.

Contents

1 Introduction **1**

1.1 Context 1
1.2 Goals 7
1.3 Challenges 9
1.4 Motivation 11
1.5 Thesis outline 12
1.6 Contributions and results 13
1.6.1 Publications 14

2 Related Work **15**

2.1 3D data in computer vision 15
2.2 Person detection and pose estimation 20
2.2.1 Person detection 20
2.2.2 Human pose estimation 23
2.3 Segmentation 25
2.3.1 Semantic segmentation 26
2.3.2 Multiple object segmentation in videos 27
2.3.3 Segmentation using bounding boxes or pose estimates 30
2.3.4 Person segmentation in stereo videos 31
2.3.5 Co-segmentation 32

3 Background Theory **35**

3.1 Deformable part models for object detection with LSVM 35
3.1.1 Model and inference 37
3.1.2 Training and LSVM 38
3.2 Deformable part models for pose estimation 41
3.2.1 Model and inference 42
3.2.2 Training procedure 45
3.3 Conditional Random Fields for segmentation 46
3.4 Spectral clustering and normalized cuts 49
3.4.1 Normalized cuts 52
3.5 Discriminative clustering with the square loss 53

4 Disparity estimation, person detection and pose estimation in 3D movies 57

4.1 Disparity estimation . . . 57
4.1.1 Acquiring 3D data from stereoscopic movies . . . 57
4.1.2 Disparity maps quality . . . 61
4.2 Datasets . . . 64
4.2.1 Inria 3DMovie Dataset . . . 64
4.2.2 Inria 3DMovie Dataset v2 . . . 67
4.3 Person detection and pose estimation in 3D movies . . . 68
4.3.1 Person detection . . . 68
4.3.2 Pose estimation . . . 70
4.4 Depth-supervised training of person detection . . . 76
4.4.1 Automatic harvesting of hard positive samples . . . 77
4.5 Discussion . . . 78

5 Multiple person segmentation with pose cues 83
5.1 Introduction . . . 83
5.2 Segmentation model . . . 85
5.2.1 Occlusion-based unary costs . . . 87
5.2.2 Label likelihood β^l . . . 87
5.2.3 Smoothness cost . . . 88
5.3 Estimating an Articulated Pose Mask . . . 89
5.3.1 Person detection and tracking . . . 89
5.3.2 Pose estimation from appearance and disparity . . . 90
5.3.3 Articulated pose mask ψ_p . . . 90
5.4 Inference . . . 91
5.4.1 Obtaining disparity parameters . . . 93
5.4.2 Person segmentation . . . 93
5.5 Experiments . . . 95
5.5.1 Segmenting multiple people . . . 95
5.5.2 Sensitivity to parameters . . . 98
5.5.3 Analysis with ground truth components . . . 100
5.5.4 H2view dataset . . . 100
5.6 Discussion . . . 104

6 Multiple person segmentation under weak constraints 105
6.1 Introduction . . . 105
6.2 Problem formulation . . . 108
6.2.1 Notations and model . . . 108
6.2.2 Grouping term . . . 109
6.2.3 Discriminative term . . . 109
6.2.4 Constraints . . . 110
6.3 Optimization . . . 115
6.3.1 Continuous relaxation . . . 115

6.3.2 Frank-Wolfe algorithm . . . 115
6.3.3 Non-convex refinement . . . 117
6.3.4 Hyperparameter search . . . 118
6.4 Experiments on multiple person segmentation under weak constraints 118
6.4.1 Implementation details . . . 118
6.4.2 Baselines . . . 120
6.4.3 Results . . . 121
6.4.4 Correlation between cost and performance . . . 126
6.5 Incorporating pose cues in a weak manner for multiple person segmentation . . . 126
6.6 Handling of other object classes . . . 130
6.7 Discussion . . . 133

7 Conclusion and perspectives **135**
7.1 Contributions . . . 135
7.2 Perspectives . . . 136
7.2.1 Person detection and pose estimation . . . 137
7.2.2 Multi-person segmentation with pose cues . . . 138
7.2.3 Multiple-instance segmentation under weak constraints . . . 140

Bibliography **141**

1 Introduction

1.1 Context

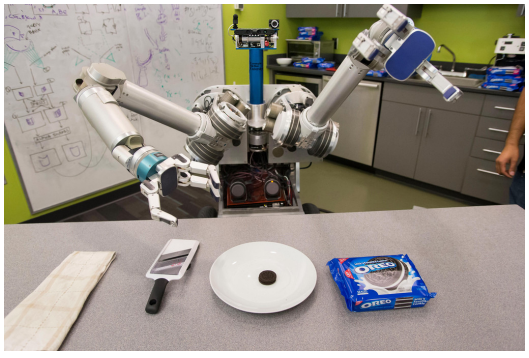
In 2016, artificial intelligence is regularly making the news. It is a hot topic, both as a scientific endeavor and as an ethical and philosophical subject. Artificial intelligence in the form of machine learning and data science is transforming many activities. For instance, online retail of digital and physical goods has already been transformed: customized suggestions are made to customers based on the products they previously bought or the ones they viewed. These suggestions are based on the behavior of the customer and on the ones of all the other customers of the platform. These techniques are also at the heart of the strategies of advertising giants such as Google: displaying the right ad for each customer, to increase click rates and then conversion rates. Natural language processing and machine learning are also being leveraged to produce speech recognition systems which are able to reply to queries expressed in plain language, such as Siri or Cortana.

More broadly, artificial intelligence aims at building systems which can autonomously understand the world they are surrounded by and correctly handle any unexpected event while performing their task. For instance, surveillance systems may need to understand and correlate visual, auditive and digital signals to properly analyze the behavior of subjects. Given visual inputs, such systems need to detect each subject as well as the action the subject is performing. This must be done in combination with an analysis of the sound cues (what the person is saying), and possibly of the digital feeds (what communications are happening). For instance, Figure 1.1 shows an example output of an outdoor video-based surveillance system, which detects and tracks people and vehicles.

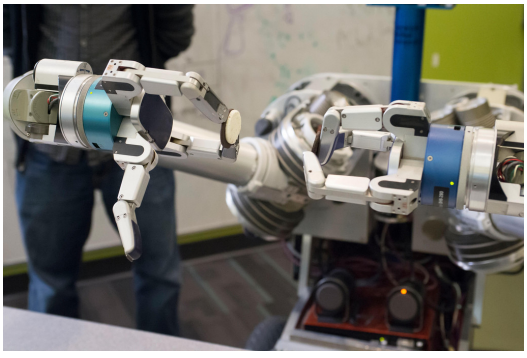
A more physical incarnation of AI is being developed in assistive technologies under the form of robots which provide physical help to elderly or disabled people. Once again, these robots must be able to feature multiple forms of intelligence in order to perform assistive tasks (lifting heavy objects, cooking, helping a person stand up...): understanding oral commands, recognizing objects and their position, shape and orientation, mapping the scene to know where it can move to and how, motion planning for both floor displacement and actuators, *etc*. An example of such a robot is shown in Figure 1.2.

Fig. 1.1: Example of surveillance system output[a]: the system detects and tracks each person and vehicle in its view. It monitors the speed of the vehicles to detect speeding ones. It also keeps tracks of persons going in and out of each store.

[a]Demo from Placemeter company `https://vimeo.com/69091237`



(a) The HERB robot



(b) Example of challenging action: picking up and tearing apart an OREO cookie

Fig. 1.2: The HERB robot[a] from the Robotics Institute of Carnegie Mellon University. This experimental assistive robot is able to automatically recognize objects and their 3D pose (position and orientation), grab them and manipulate them. This involves a combination of computer vision and robotics (in terms of mechanics and motion planning) challenges.

[a]`https://www.cmu.edu/herb-robot/photos/`

Another physical incarnation of AI can be found in self-driving cars, which face similar challenges as illustrated in Figure 1.3. They must combine traffic and routing information with an understanding of the structure of the road (lanes, traffic lights) and of where the other cars are and what they are doing. They must be able to react to sudden changes of the behavior of the other vehicles (lane changes, *etc*), and to unlikely events, such as pedestrians crossing the road at unexpected locations.

At the core of many of these applications is the ability to properly understand a visual environment. This faculty is a natural one for most living beings, which have large parts of their cognitive systems devoted to the visual perception task, systems which have evolved and adapted over millions of years. However, it remains a difficult challenge for machines. Computer vision is the scientific field which aims at enabling machines to develop a visual understanding of the world. Visual contents are an extremely rich type of data, but at the same time they are also extremely variable and noisy. Slight illumination or viewpoint changes can heavily impact the signal measured by optical acquisition devices such as cameras.

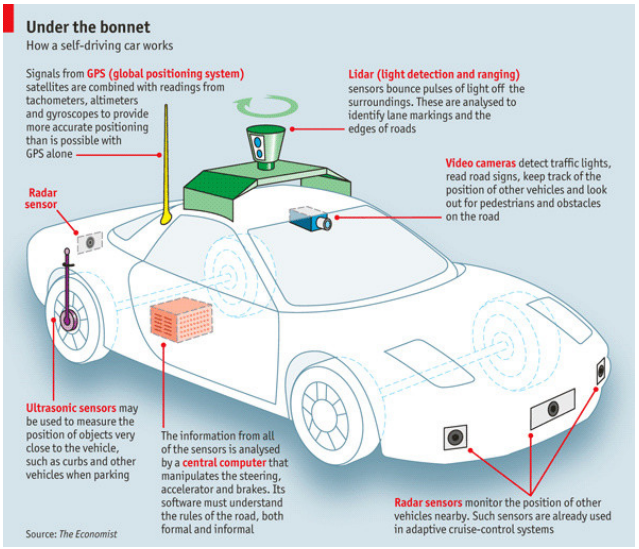
People are at the center of many practical applications, and thus at the center of many computer vision tasks, as illustrated in Figure 1.4. One of the iconic tasks of computer vision is the one of face detection. The very popular Viola-Jones method [Viola and Jones, 2004] is one of the most well-known computer vision algorithms, and a typical homework assignment for computer vision students. A more difficult people-related task is the one of person detection, which suffers from a larger range of possible deformations, occlusions and frame cropping. Given a face detection or a person detection, higher level questions can be asked: whose face is this? What is the physical pose of the person? What action is this person performing? At a finer level, a pixel-wise segmentation of the person can be a strong cue for practical applications such as image and video editing tasks.

However, people are a notably challenging class of objects. Compared to other typical object classes, people are significantly harder to analyze than rigid or mostly rigid objects, such as cars or airplanes, and even than other animal classes, which often have less appearance and pose variability.

An interesting fact is that as people are at the center of many computer vision problems, they are also at the focus of typical images and videos. Most movies, TV series and documentaries focus on the stories of human characters. In TV video footage, movies or YouTube videos, people usually occupy around 34% of the video pixels [Laptev, 2013]. Furthermore, an average feature-length movie is made of about 130000 frames. On YouTube in 2014, it was estimated that about 72 hours of videos were uploaded every minute to the platform, which at 24 frames per second

(a) Prototype of the Google Car

(b) Challenges of self-driving cars



(c) Sensors used in self-driving cars

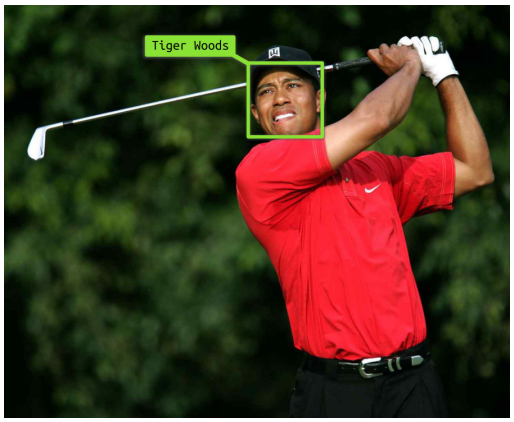
Fig. 1.3: Challenges of self-driving cars. We show in (a) a prototype of the self-driving Google Car[a]. In (b), we show an illustration of the challenges faced by self-driving cars[b]: recognizing the road layout, traffic lights and signs, *etc*. These challenges are tackled by processing and combining the output of many sensors[c]: radar and ultrasonic sensors, video cameras, lidar (light detection and ranging) sensors, *etc*.

[a] `https://www.google.com/selfdrivingcar/`

[b] Extracted from a Volvo report, photo by Henrik Ottosson `http://www.volvocars.com/SiteCollectionDocuments/TopNavigation/Corporate/Financials/FinancialReportH12013.pdf`

[c] From The Economist `http://www.economist.com/node/21560989`

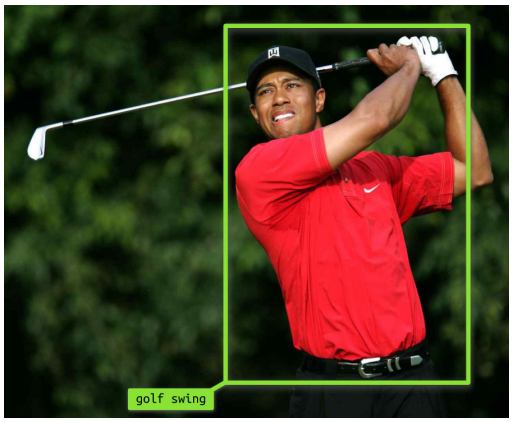
(a) Face detection



(b) Facial recognition

(c) Person detection

(d) Pose estimation



(e) Action recognition

(f) Segmentation

Fig. 1.4: Examples of people-related computer vision tasks, from high level, coarse ones such as face or person detection, to fine, low-level ones such as person segmentation.

sums to about 9 billion frames uploaded every day to the platform. This abundance of data can provide very large datasets for training models and algorithms.

Another trend is that 3D data has recently gained a worldwide commercial success. On one side, the Microsoft Kinect has been a revolution in the gaming industry. Based on an infrared emitter and camera, it is able to compute the physical depth of each point in its viewport. Combined with a powerful pose estimation algorithm [Shotton *et al.*, 2011], it allows Microsoft XBox players to play video games without holding any controller, simply by making gestures or moving their body. Combined RGB and Depth (RGB-D) sensors such as the Kinect or other depth cameras have been used to collect datasets for tasks such as pose estimation [Ionescu *et al.*, 2014] or semantic segmentation [Silberman *et al.*, 2012]. On the other side, 3D movies have been around for more than a century: the first theatrical tests of stereoscopic footage were done in 1915 by Edwin S. Porter and William E. Waddell in New York City. These tests were presented in red-green anaglyph and depicted a variety of situations, from natural landscapes to human actors and dancers. However, stereoscopic movies have only recently known a large commercial success. Screening of such movies was previously only possible in a few hundred theatres around the world, such as IMAX 3D ones. 3D movies are now available as consumer-level products: 3D movies screenings are available in more than 25000 theatres, and millions of 3D televisions have been sold to consumers. At the same time, more than 500 feature-length movies have been released in 3D, and more than 200 have been shot with a true 3D stereo rig[1]. These movies sum up to several hundreds of hours, providing millions of stereo pairs shot in a very wide ranges of scenes and situations.

(a) Anaglyph glasses **(b)** Fujifilm W3 camera **(c)** Sony HDR-TD30 camcorder

Fig. 1.5: Examples of devices for 3D movies: anaglyph glasses (a) allow watching 3D movies screened using the anaglyph technique. Consumer-level cameras (b) and camcorders (c) are available to capture stereoscopic stills and videos.

In addition to the professional hardware used to shoot 3D movies, consumer-level hardware is now available for shooting memories in 3D, as illustrated in Figure 1.5. Digital cameras such as the Fujifilm FinePix Real 3D W3 allow capturing stereoscopic pairs instead of a single image as standard cameras. 3D camcorders such as the Sony HDR-TD30 allow shooting stereoscopic videos in full HD resolution. Once again, these devices can provide large and varied amounts of stereo pairs. For in-

[1] `http://realorfake3d.com/`

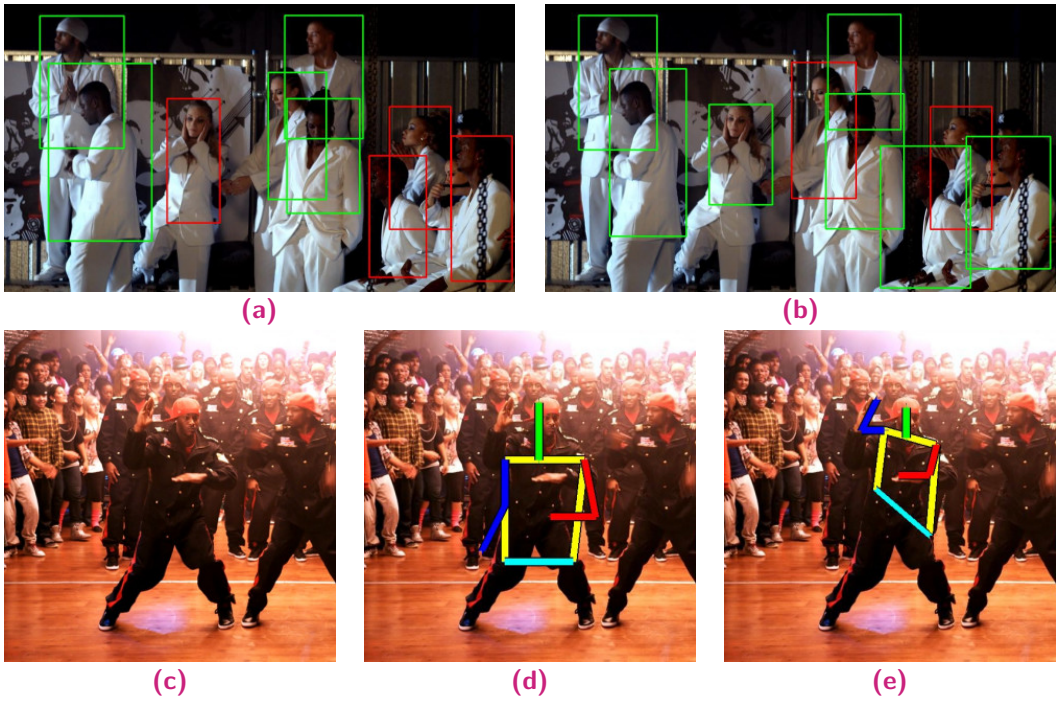


Fig. 1.6: Examples of improvements obtained by using disparity feature in person detection and pose estimation methods of Chapter 4. Top line: we compare side-by side person detection results using only color information (a) or using a combination of color and disparity information (b) for two frames. Our method that uses disparity detects two more people (green) that were missed (red) by the baseline method. Bottom line: we show the original frame (c), pose estimate using only color information (d) or using both color and disparity information (e) for two frames. Note how our method recovers more accurately the pose of the person with the extended right hand.

stance, entire Flickr groups are devoted to sharing pictures shot with the Fujifilm W3, such as `https://www.flickr.com/groups/finepix_real_3d/`.

1.2 Goals

In this thesis, we build on the success of 3D video content and study how we can exploit the additional information contained in this type of data, compared to standard color videos. We focus on person analysis in stereoscopic movies. In particular, we study three people-related tasks: person detection, pose estimation and video segmentation. For person detection and pose estimation, we analyze how disparity features can improve the output of methods based on deformable part models, as illustrated in Figure 1.6. For video segmentation, we develop two new methods to segment multiple persons. The first method combines cues derived from pose estimates with disparity cues. It not only provides a pixel-wise segmentation of each person but also outputs a layering of the persons in the scene, as illustrated in Figure 1.7. The second method only relies on tracked bounding boxes and outputs the

pixel-wise segmentation of each object instance. These bounding boxes are used to constrain the space of possible segmentations in a weak manner. As illustrated in Figure 1.8, this method not only works for people but also on any other object class, as it does not directly require any class-specific appearance model.

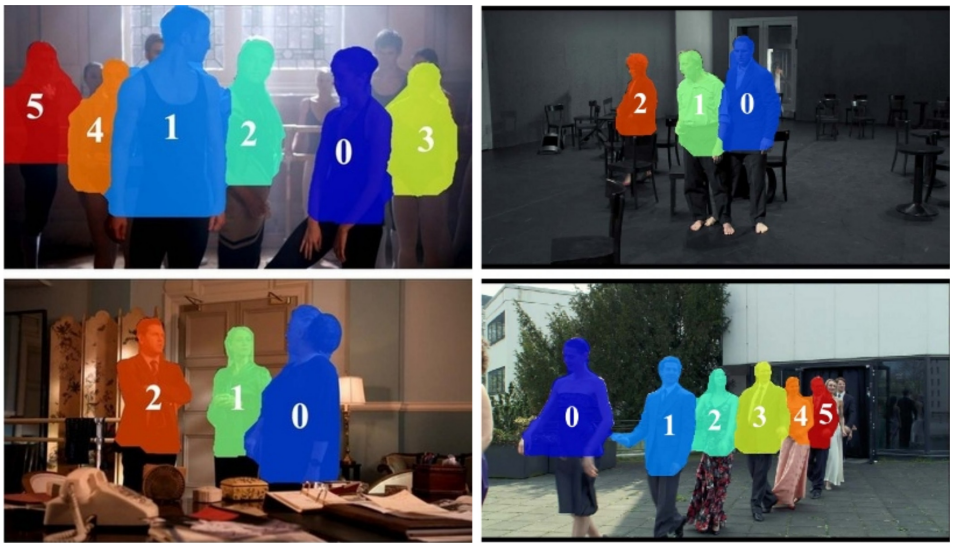


Fig. 1.7: Examples of layered multi-person segmentation results produced by our new method from Chapter 5. In addition to the pixel-wise segmentation of each person, our method outputs a layering of all the persons in the scene, shown by the numbers and the color overlays on the figure. The foremost person being identified by 0, and the color overlays use the standard jet color map, with dark blue corresponding to the foremost person and dark orange/red to the person the most in the background.

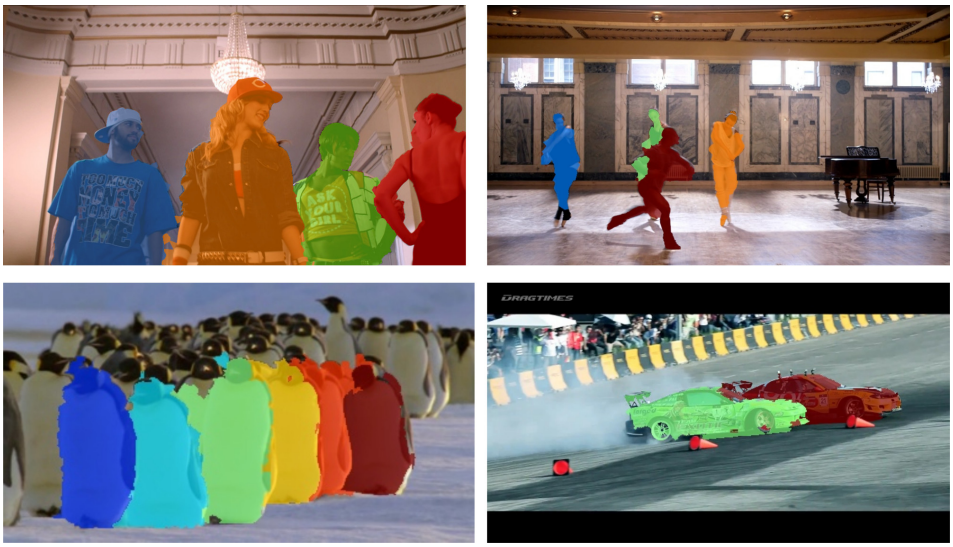


Fig. 1.8: Examples of multi-instance object segmentation results produced by our new method from Chapter 6. This method not only works on multiple persons (top line), but can also work on other object classes (bottom line) without requiring the learning of class-specific appearance models.

1.3 Challenges

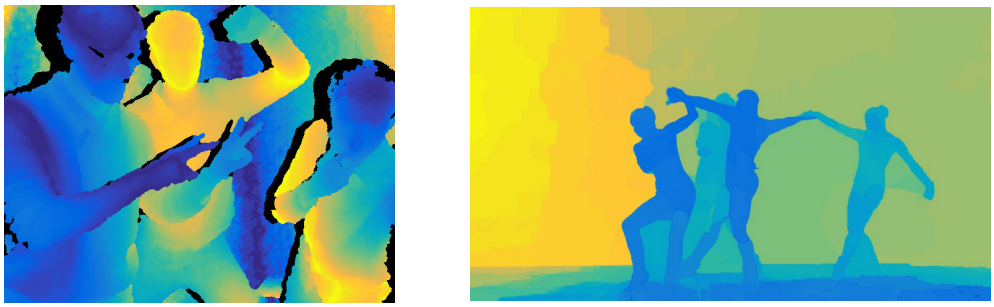
This thesis addresses the following three main challenges: modelling the heavily variable appearance of people, extracting a notion of depth from uncalibrated stereoscopic streams and combining disparity cues with other cues such as color or motion signals.

Modelling appearance of people. One of the main challenges in analyzing human appearance is the heavy variability of pose and appearance of people. Human appearance can vary considerably due to skin, body shape or clothing variability. Furthermore, the human body features many physical joints and has more than 200 degrees of freedom. As a consequence, our articulated bodies can take a very large number of poses. The fact that we often pay attention to small nuances of body postures and motions makes the problem even harder.

The combination of multiple people in a single scene creates even stronger challenges due to occlusions. For instance, if two persons are wearing similar clothes and one is partially occluding the other, segmenting apart the two persons and the background is a very difficult task. Another example is two persons holding each other's hand: knowing where the arm of the first person ends and where the arm of the second person starts is a complicated problem. We illustrate these challenges in Figure 1.9.

Fig. 1.9: Examples of challenges of person analysis: inter-person occlusions (a,b,d), challenging poses (a,c), person-background appearance similarity (c) and inter-person appearance similarity (d).

Extracting depth from stereoscopic movies. At first sight, 3D data used as an input to person analysis methods may seem like a very powerful addition. Indeed, successes such as the ones achieved in foreground-background subtraction and pose estimation by the Kinect have shown the power of depth data. However, the Kinect is a joint hardware and software projects: it uses very specific sensors developed for this purpose, which have a limited range, only work indoors, *etc*. Thus, the entire signal acquisition pipeline behind the Kinect algorithms was entirely mastered by its developers, who were able to model the behavior of the sensors and their particular noises. In the case of 3D movies, the shooting procedure and post production work are typically unknown to the consumer. In addition, 3D movies are stored and distributed as a pair of stereoscopic video streams, one meant to be seen by the left eye and another one by the right eye. This format is nowhere as explicit as the depth maps produced by active sensors such as the Kinect. Extracting a notion of depth from stereoscopic video streams often involves matching the pixels of the two streams to compute the relative displacement of each pixel from the first view to the second view. The estimated displacement field is called a disparity map. However, in uncalibrated and unrectified setups, the disparity estimates may be significantly noisy, as illustrated in Figure 1.10. The second challenge of our work thus resides in the exploitation of uncalibrated pairs of stereoscopic video streams.



(a) Kinect depth map **(b)** Disparity map extracted from a 3D movie

Fig. 1.10: Comparison of a depth map produced by the Microsoft Kinect (a) with a disparity map estimated from a stereo pair from a 3D movie (b). The Kinect depth map features fine details and explicitly outputs the regions where the depth estimation is unreliable (in black), while the disparity map only recovers layers of the scene.

Combining cues. The third challenge of our work is the use of this noisy disparity data in combination with other signals, such as color and motion signals, to solve people-related computer vision problems. While using an additional channel of information is most often beneficial, great care is needed to avoid creating new types of errors because of the weaknesses of this additional channel. For instance, as shown in Figure 1.11, segmentation methods may produce erroneous results induced by incorrect disparity estimation in texture-free areas. We thus need to properly weight the contribution of each signal in our algorithms.

(a) Left frame **(b)** Estimated disparity

Fig. 1.11: Example of disparity estimation errors which may typically lead to "leaking" problems for a disparity-based segmentation method.

1.4 Motivation

Our work aims at producing a set of tools which can be used directly or indirectly in practical applications. For instance, our person detection and pose estimation work could be used in surveillance or robotics applications. Multi-person segmentation could help editing stereo videos or movies, or be used as a mid-level representation which can be further used for tasks such as action detection and recognition.

Surveillance and robotics. The most straightforward applications of our work is the use of person detection and pose estimation methods for surveillance or robotics application. For instance, person detection and pose estimation using 3D cues can be applied directly to robots and systems already using a stereoscopic camera setup, such as the HERB robot shown in Figure 1.2 or some self-driving cars, as explained in Section 1.1. Surveillance systems which usually contain a single camera can also benefit from better person detectors trained using the proposed harvesting method which leverages the relative ease of the person detection task in 3D movies to learn better models for standard color-only contents.

Video editing. Given the people-centric nature of many movies, being able to reliably produce a pixel-wise segmentation of each person in each frame is a valuable feature for video editing softwares. With such a feature, it becomes much easier to remove a person entirely from the video, to change their clothing or swap them with another person. Given the layer ordering of the persons in the scene, which can be an output of the segmentation method, it also becomes easy to integrate additional dynamic overlays in between the different layers of the scene, such as adding a droid which weaves around the people in a Star Wars scene.

Mid-level representation for human action recognition. Reliable pixel-wise segmentation of people can also be used as a mid-level representation for other tasks, such

as action detection and recognition. The segmentation masks can either be used directly as features for the target method or can be used indirectly. For instance, many state of the art methods extract dense features from the whole video or from a tube of the video. Given per-person segmentation masks, new feature extraction strategies can be developed, such as extracting features specifically from foreground regions, or from the region associated to each person.

1.5 Thesis outline

The rest of this thesis is organized as follows.

In Chapter 2, we review the literature related to our work: uses of 3D data for computer vision tasks, person detection, human pose estimation and segmentation.

In Chapter 3, we describe the background theory and methods used in this work: deformable part models for object detection and pose estimation, Conditional Random Fields, spectral and discriminative clustering methods.

In Chapter 4, we explore the acquisition of disparity maps from feature-length stereoscopic movies. We then extract features from this additional channel of information and use these features to train person detection and human pose estimation models for 3D movies. After evaluating these models, we leverage the relative ease of the person detection task in 3D movies to perform a depth-supervised harvesting of person detection positive examples and train a better person detector for non-3D movies.

The last two chapters of this thesis focus on the task of segmenting multiple people in videos. We first introduce a model for multi-person segmentation in 3D videos in Chapter 5. We formulate the segmentation problem as a multi-label Conditional Random Field. The unary terms of the model combine cues from disparity maps with a rough segmentation mask derived from pose estimates. We also explicitly model the occlusions between the different persons in the scene, and produce a layered, multi-instance segmentation.

Finally, in Chapter 6 we propose a second model for multi-instance person segmentation in videos which takes tracked bounding boxes as an additional input. This time, we formulate the segmentation problem as a convex optimization one, the minimization of a quadratic cost under linear equality or inequality constraints. These constraints are used to encode prior knowledge about the localization of each object in a weak manner. We can for instance say that in a given region, at least

70% of the pixels belong to a given object. This model can also be used for non-person object classes, and can handle additional channels of information or prior information very easily. We show the flexibility of the model by incorporating pose estimates in a weak manner, and by performing segmentation propagation on a standard benchmark dataset.

1.6 Contributions and results

Our contributions are thus the following:

- We study how to extract disparity information from feature-length stereoscopic movies. Given that the stereoscopic pairs provided by 3D movies are neither rectified not calibrated, we resort to standard optical flow methods to match the pixels of the two views and extract a notion of disparity.
- We evaluate the impact of this additional channel of information for person detection and human pose estimation methods. We include additional disparity-based features in the feature vectors used by methods based on deformable part models and study the performance improvements on these two tasks.
- We develop a method to collect training examples from 3D movies for a person detector aimed at standard color movies. Using a small initial dataset of person bounding boxes labelled by hand, we train a powerful person detector for 3D movies, which we use to harvest a large number of person examples which can be used to train a better detector for color movies.
- We propose a model to perform multi-person layered segmentation in 3D movies. This method combines rough segmentation masks derived from pose estimates with disparity cues to jointly produce the pixel-wise segmentation of each person as well as the layering of the persons in the image.
- We propose a model to perform multi-instance object segmentation from object tracks. We cast the object tracks as constraints which shape the space of admissible segmentations in a convex optimization problem. As it involves no class-specific appearance model, this method can be easily applied to multiple instances of any object class.

To evaluate the proposed methods, we introduce two new datasets extracted from feature-length 3D movies. These datasets contain ground-truth annotations for training and testing person detection, pose estimation and video segmentation models in 3D movies.

Among the valuable results of our work, the depth-supervised training of a person detector for standard color movies exhibits a very high performance on our test

dataset as well as on a standard benchmark dataset. Our video segmentation methods produce quality segmentations in many cases, and are able to handle some very challenging sequences with heavy deformations and appearance variations.

1.6.1 Publications

This thesis has lead to the following publications:

- Karteek Alahari, Guillaume Seguin, Josef Sivic, Ivan Laptev. Pose Estimation and Segmentation of People in 3D Movies. ICCV 2013.
- Guillaume Seguin, Karteek Alahari, Josef Sivic, Ivan Laptev. Pose Estimation and Segmentation of Multiple People in Stereoscopic Movies. PAMI 2015.
- Guillaume Seguin, Piotr Bojanowski, Rémi Lajugie, Ivan Laptev. Instance-level video segmentation from object tracks. CVPR 2016.

2 Related Work

In this chapter, we review the literature related to our work. We start by describing in Section 2.1 the various ways 3D data have been involved in computer vision. We then describe the evolution of person detection and human pose estimation methods in Section 2.2. Last, in Section 2.3, we review segmentation methods for still images, sets of images and videos related to our works on multi-person and multi-object segmentation in videos.

2.1 3D data in computer vision

Given the 3D nature of our physical world, 3D is naturally at the core of many computer vision problems. The first 3D-related computer vision problem is the one of depth estimation. Indeed, while our binocular visual system allows us to perceive the world in 3D, typical cameras only capture a 2D view of the world. A stereoscopic pair of images captured by a pair of cameras can be used to estimate the depth of the scene [Marr and Poggio, 1979] has been studied since at least the 1960s [Julesz, 1962]. Stereo vision methods aim at computing a disparity map, which measures the distance between pixels in the two images which correspond to the same physical point. Each disparity value is mathematically related to depth values, usually by an inverse relationship. Based on the taxonomy from [Scharstein and Szeliski, 2002], stereo algorithms generally consist of four steps:

1. Computing a matching cost, which evaluates how similar two pixels are, for instance by measuring the mean square error between their associated colors. This step usually yields a 3D volume C where $C(i, j, d)$ encodes the similarity between the pixel at location (i, j) in the first view with the pixel $(i + d, j)$ in the second view with d being the considered disparity value.
2. For each disparity value and location, aggregating the matching cost over a local window, for instance by averaging the costs of all the pixels within the window. This step yields a new 3D volume A with similar semantic as C, but which contains a more robust information as it encodes the local similarity between the candidate regions.
3. Computing the disparity map D by performing either a local or global optimization. Local optimization approaches simply select the most likely

disparity value at each location based on the aggregated cost: $D(i,j) = \arg\max_d A(i,j,d)$, a method often referred to as "winner-take-all". Global optimization techniques combine this local reasoning with global smoothness priors. Typical global optimization techniques formulate the disparity estimation problem as an energy minimization problem over a graph corresponding to the pixels of one of the views, with unary terms reflecting the aggregated costs and binary terms encouraging the smoothness of the solution. These formulations can be efficiently and exactly optimized using methods such as graph-cuts on an appropriate graph [Ishikawa, 2003].

4. Refining the disparity map to produce a finer-grained information. For instance, many optimization procedures yield disparity maps which are integer valued. These initial maps can be refined to achieve a sub-pixel accuracy, for instance by smoothing the map or by using more advanced techniques which would have been computationally intractable to compute the initial map.

Steps 1 and 2 are often combined when the matching cost is computed by using information from an entire window, as it is done for instance for normalized cross-correlation [Hannah, 1974] where the intensities are normalized over the corresponding window by taking the average and standard deviation of the intensity in the window into account. A very large number of approaches has been proposed across the years [Scharstein and Szeliski, 2002] and differ on the type of matching cost, aggregation procedure and optimization method. Stereo vision methods are most often evaluated on the Middlebury dataset and benchmark[1] and on the KITTI stereo benchmark[2].

Most of the early works on stereo vision did not use of datasets with ground truth to learn parameters for their models and used either hand-crafted matching costs and aggregation procedures. Newer approaches are often based on machine learning, such as the method of [Kong and Tao, 2004] which learns a classifier which predicts whether the matching cost used is reliable at a given location, or unreliable due to a foreground object in the view, or unreliable due to another factor. The disparity map is then initialized to the one deduced from the matching cost and aggregation, and refined based on the probability of the location belonging to each of the three classes. More recently, a random forest classifier was trained to predict the confidence of the matching cost at each location [Spyropoulos *et al.*, 2014]. The predictions were then used to select pixels for which the matching cost has been deemed as highly reliable. The disparity map is then produced by solving a Markov Random Field optimization problem in which the assignment of these highly-reliable pixels are specified as soft constraints. The current state of the start method for stereovision [Žbontar and LeCun, 2015] trains a siamese convolutional neural network to

[1] `http://vision.middlebury.edu/stereo/`
[2] `http://www.cvlibs.net/datasets/kitti/eval_stereo.php`

compare a pair of patches from the two views. This approach trains a matching cost function which is highly fit for the disparity estimation task. This matching cost is then aggregated over an appropriate, pixel-specific region selected so that it mostly belongs to the same object [Zhang *et al.*, 2009] and smoothed using a semiglobal matching approach [Hirschmüller, 2008], before building the disparity map with the winner-take-all method.

In our work, we work with uncalibrated pairs of stereoscopic streams extracted from 3D movies. We explore how to compute disparity maps from these streams in Chapter 4.

On a slightly unrelated note, more images or viewpoints can be used to estimate the 3D location of each captured pixel. Structure-from-motion techniques [Hartley and Zisserman, 2000] jointly estimate the 3D world location of each image pixel with the 3D poses of cameras. They involve finding point correspondences between the images, and then reasoning on the underlying geometry to properly reconstruct the viewpoints and a 3D point cloud. These techniques have been used on large datasets with many views taken by different cameras and at multiple points in time: the Photo Tourism project [Snavely *et al.*, 2006][3] aimed at reconstructing 3D models of entire buildings or areas using personal photo albums or collections shared on social networks such as Flickr. At an even larger scale, the Rome in a Day project [Agarwal *et al.*, 2011][4] proposed a highly scalable system which can reconstruct entire cities by using millions of pictures, as illustrated in Figure 2.1. This represents great challenges, as it involves matching millions of images and solving very large non-linear optimization problems.

Other kinds of sensors can capture depth directly or indirectly. Direct acquisition can be achieved by using laser rangefinders such as LIDARs (contraction of "light" and "radar"), which send pulses of laser light and measures the round-trip time between a pulse emission and the reception of its reflection. Multiple forms of LIDARs exist, such as ones which can measure the depth of a single point at a time and require a mechanical scanning system to measure the depth of the entire scene, as well as time-of-flight systems which can capture the depth of the whole scene without requiring a mechanical scanner. LIDAR devices are often used in combination with standard cameras, for instance for autonomous vehicles, such as self-driving cars or unmanned aerial vehicles (UAV). Typical applications of such combinations include pedestrian detection [Premebida *et al.*, 2009] or mapping and reconstruction of the 3D world [El-Hakim *et al.*, 2004]. Time-of-flight cameras and LIDARs have also been used to capture datasets for computer vision, since they can produce high accuracy depth ground truth. For instance, the Human3.6M dataset [Ionescu

[3] http://phototour.cs.washington.edu/
[4] http://grail.cs.washington.edu/projects/rome/

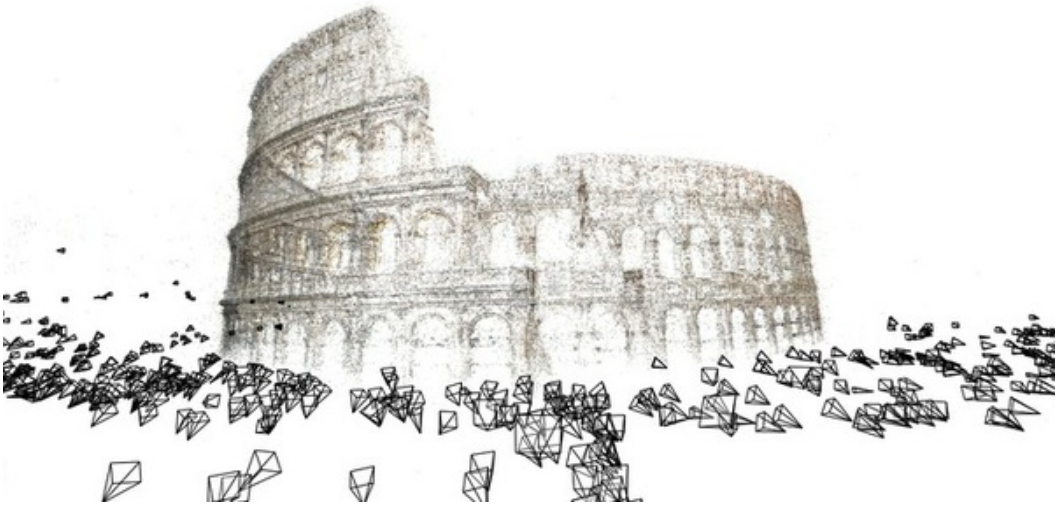
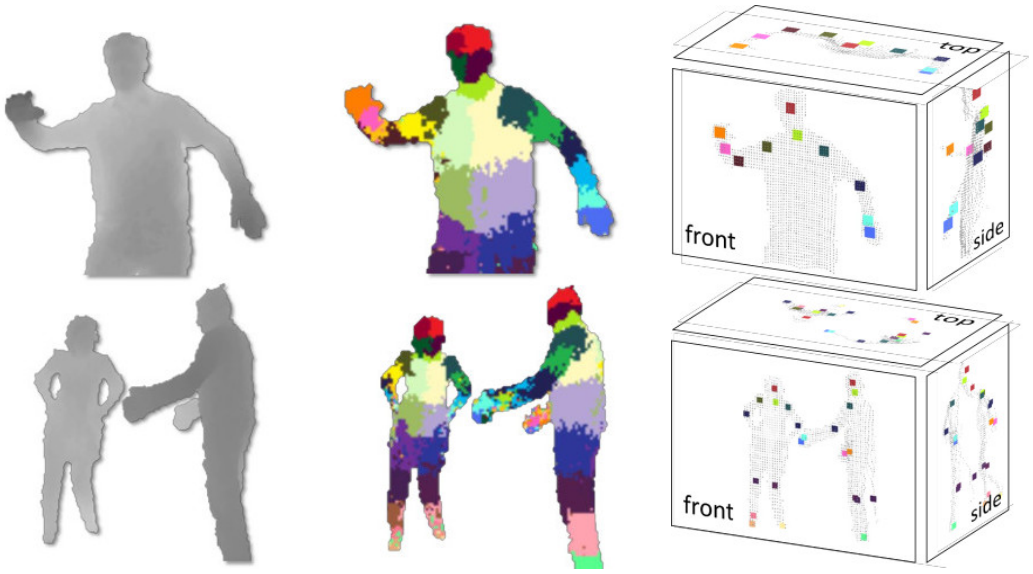


Fig. 2.1: The Rome in a Day project: millions of pictures collected from the image sharing websites are used to produce a 3D reconstruction of an entire city [Agarwal *et al.*, 2011].

et al., 2014] is composed of 3.6 million images featuring various human poses, actions and contexts, for which ground truth depth maps were captured using a combination of a time-of-flight camera with motion capture sensors. Similarly, the KITTI dataset [Geiger *et al.*, 2012] features road sequences captured from a car rigged with multiple cameras and a laser scanner, and provides ground truth depth data.

Indirect acquisition can be achieved by using active sensors, such as structured-light systems. Such systems project known patterns of light, for instance grids, horizontal lines or dots, and use a camera to capture a view of the scene. Analyzing the distortion of the projected pattern in the captured view allows to efficiently estimate the depth of each point. One example of this type of sensor is the Microsoft Kinect, which uses an infrared emitter to project a pattern of dots, along a monochrome CMOS sensor which captures the reflected infrared light. This consumer-grade device was originally developed and sold as a peripheral for the XBox 360 gaming system, which allows the player to play games without holding a physical controller, by detecting the pose of the player and specific gestures. The pose estimation algorithm developed for the Kinect [Shotton *et al.*, 2011], illustrated in Figure 2.2, relies on random forests to find the body part to which each person pixel belongs. The proposed assignments are then refined to be spatially consistent, and the position of each physical joint is estimated from the corresponding cluster of points. The differences of depth between the considered pixel and multiple neighboring pixels at various distances are used as features for the random forests. An interesting fact is that the random forests were trained using synthetic data, generated to reproduce the noises and biases of the final sensor. Such consumer-grade devices also allow making cheap 3D scanners, which can efficiently produce 3D models of objects, persons

and indoor scenes, for instance by using the KinectFusion algorithm [Newcombe *et al.*, 2011]. They can also be used for autonomous vehicles and robots to perform tasks such as Simultaneous Localization And Mapping (SLAM) [Kerl *et al.*, 2013].



(a) Input depth map **(b)** Pixel labels **(c)** Final joint predictions

Fig. 2.2: The Kinect pose estimation algorithm [Shotton *et al.*, 2011] reasons on depth maps (a) obtained by the device and for which the background has been subtracted. Each foreground pixel is classified into a body part (b), using features computing the relative difference of depth between the current pixel and neighboring ones at multiple ranges. After aggregating and refining the predictions, 3D body joint locations are computed (c). Figure from [Shotton *et al.*, 2011].

In addition to scene reconstruction and pose estimation, 3D data is also a popular medium for object recognition. For instance, in [Collet *et al.*, 2011] multiple pictures of the same object are captured from different viewpoints. Keypoints and descriptors are extracted from each image, and matched with those from the other images, and a 3D model of each object composed of a set of 3D points, each point being linked to the corresponding descriptors from the original images. At test time, features are extracted from the input image and matched to the ones in the object database. Object proposals and object 3D pose estimates are then produced using an algorithm which iteratively refines the set of features belonging to each object instance and estimates the pose aligning the keypoints with the object model.

The Kinect has also enabled researchers to collect large datasets for 3D object recognition, such as the BigBIRD dataset [Singh *et al.*, 2014], which features 600 views (RGB image + depth ground truth) for each of the 125 objects of the database, as well as 3D pose and segmentation masks ground truths. On this type of data, methods such as [Xie *et al.*, 2013] have demonstrated superior performance than on RGB-only data, by properly leveraging the depth data for preliminary segmentation as well as feature computation.

3D data has also been used for semantic segmentation. For instance, the NYU Depth Dataset V2 [Silberman *et al.*, 2012] features 464 scenes and 1449 annotated pairs of RGB image and depth map, with semantic and instance-level labels. Combining RGB and depth information enables algorithms to handle typically hard situations in RGB-only data, such as occlusions where both objects have a similar appearance. For instance, the state-of-the-art method on this dataset [Banica and Sminchisescu, 2015] combines boundaries detected on the RGB image with boundaries detected on the depth image to improve a segment proposal method. It also extracts additional features based on the 3D bounding box of each segment proposal, using the 3D point cloud of the region.

Our work is inspired by the success of the Kinect on pose estimation and the proliferation of datasets acquired by using 3D sensors. In Chapter 4, we study how to extract disparity from readily available 3D movies, introduce two datasets extracted from 3D movies and study how person detection and pose estimation methods can be improved by the additional channel of information offered by 3D movies. In Chapter 5 and Chapter 6, we study the task of instance-level segmentation and evaluate our methods on data extracted from 3D movies.

2.2 Person detection and pose estimation

Person detection and human pose estimation are two of the main people-related computer vision tasks. These tasks directly face the challenges posed by the inherent high deformability of the human body and the infinitive variation of human appearance due to body shape, skin color and clothing variations.

2.2.1 Person detection

Person detection aims at automatically finding the location of people in images and videos. In practice, the output of person detection methods is often a set of bounding boxes, rectangles inside which a person has been localized. In addition to the classical challenges of computer vision (viewpoint changes, illumination variations), person detectors face the very large variability of human shape, pose, skin and clothing combinations, as well as frequent intra-occlusions (a limb hidden behind another) or inter-occlusions (one person partially occluding another one).

Some of the first approaches to person detection targeted the simpler case of pedestrians. Indeed, the pose and shape of a standing pedestrian is usually quite simpler recognizable than when considering the set of all possible human poses. These approaches first built templates of pedestrian edges, either by manually annotating

person outlines [Gavrila, 2000] or by using a perceptron to learning 2-D filters over image contours [Felzenszwalb, 2001]. At test time, they then matched the extracted image edges to these templates.

However, edge detection is by nature a hard task, especially in the wild. Instead of extracting edges from the entire image, patches from the image can be considered using the sliding window approach. Features are extracted from each patch, and classified into person or background classes, for instance using Haar wavelets and support vector machines [Oren *et al.*, 1997; Papageorgiou and Poggio, 2000]. In [Dalal and Triggs, 2005], a new type of features, Histogram of Oriented Gradients (HOG), computed over cells on the gradients maps, was introduced and applied successfully to person detection.

To handle more types of poses than a single rigid template can, multiple templates can be used. For instance, poselets [Bourdev and Malik, 2009] are local pose-specific detectors, which are trained using examples which have a locally consistent pose, for instance people having their arms stretched would lead to a local detector, while people having their arms crossed would lead to another, and people sitting on a bench to another, as illustrated in Figure 2.3. A single example can thus be used to train multiple detectors which consider different parts of the body. Later on, a method to combine the output of these detectors into a reliable person detector was proposed by [Bourdev *et al.*, 2010].

Fig. 2.3: Examples of clusters of locally consistent poses used to learn poselets in [Bourdev and Malik, 2009].

Using a single rigid template or even multiple alternative rigid templates is too limited to handle the large deformability of the human body. To solve this forthcoming, multiple templates corresponding to different body parts can be introduced. For instance, in [Mohan *et al.*, 2001], four parts are considered: upper body, left and right arms, lower body. These parts are allowed to move slightly relatively to each other, but the extent of the moves has to be set manually. The pictorial structure model [Fischler and Elschlager, 1973] allows defining parametric relationships between parts, which can be deformed using spring-like deformation priors. The score

of a bounding box is the sum of the scores of the parts plus a deformation cost for each link between two parts. While the inference over such models was considered as hard initially, it was made tractable for certain types of deformation cost by the introduction of the generalized distance transform algorithm [Felzenszwalb and Huttenlocher, 2004a]. It was then applied to object detection by [Felzenszwalb and Huttenlocher, 2005].

These "deformable parts models" were extended in multiple directions. For instance, occlusions are handled in [Girshick *et al.*, 2011] by incorporating an *occluder* part, which can represent any object occluding part of the person. Another successful development was the person detection and pose estimation method of [Yang and Ramanan, 2011]. Instead of discovering the parts automatically as in [Felzenszwalb and Huttenlocher, 2005], the parts correspond to physical body parts and joints: wrists, elbows, shoulders, *etc*, which were fully annotated in the training set. In addition, and building upon the poselets idea, multiple components are learned for each part to handle different local configurations, such as straight elbows or folded elbows, based on a clustering of all the configurations of each part in the training set. Both the template and the deformation cost are learned relatively to the part type. This approach can successfully model a very large number of configurations and can produce very good detection results.

More recently, object detection, including person detection, has been revolutionized by the growing performance of convolutional neural networks (CNN). Using ImageNet, a very large object classification dataset, high classification performance was achieved on a 1000 class task using a deep convolutional neural network by [Krizhevsky *et al.*, 2012]. These advances were then transferred to object detection tasks and datasets by [Girshick *et al.*, 2014] by leveraging the networks trained for the classification task and adding additional adaptation layers. This way, the mid-level representations which were trained for classification are reused (and even often fine-tuned) for another task without requiring a very large training set again. Recent developments shifted away from the sliding window paradigm by using box proposal methods [Krähenbühl and Koltun, 2014; Krähenbühl and Koltun, 2015] which were later even incorporated in a single network used for both box proposal and object detection [Ren *et al.*, 2015].

In our work, we use the object detection method of [Felzenszwalb *et al.*, 2010] for person detection, which we retrain using additional features extracted from disparity maps in Chapter 4. We use this detector to perform the depth-supervised training of a person detector for color movies in Section 4.4, and to detect persons for multi-person segmentation in Chapter 5. We also use a method derived from [Girshick *et al.*, 2014] to detect person heads which are used to build our instance-specific constraints over the shape of the segmentation space in Chapter 6.

2.2.2 Human pose estimation

Human pose estimation is the task of estimating the posture of a person in visual data. The output of pose estimation methods is most often a set of sticks corresponding to limbs (lower arms, upper arms, *etc.*), or a set of keypoints corresponding to body joints (wrists, shoulders, *etc.*).

Based on the physical structure of the problem, early approaches, often using generative models, considered the human limbs as cylinders connected to each other in the 3D world [Deutscher *et al.*, 2000; Sidenbladh *et al.*, 2000; Elgammal and Lee, 2004] as illustrated in Figure 2.4. Instead of sets of cylinders, complete 3D models of the human body can be considered [Sminchisescu and Triggs, 2001]. In this setup, estimating a person pose implies estimating the relative 3D rotations between each pair of connected limbs, as well as other scaling and displacement factors. However, such approaches are typically hard, especially since the problem is quite ill-posed, given that multiple 3D poses can lead to the same human appearance in a 2D image. They often imply using advanced sampling methods [Sminchisescu and Triggs, 2002] or search techniques [Sminchisescu and Triggs, 2003].

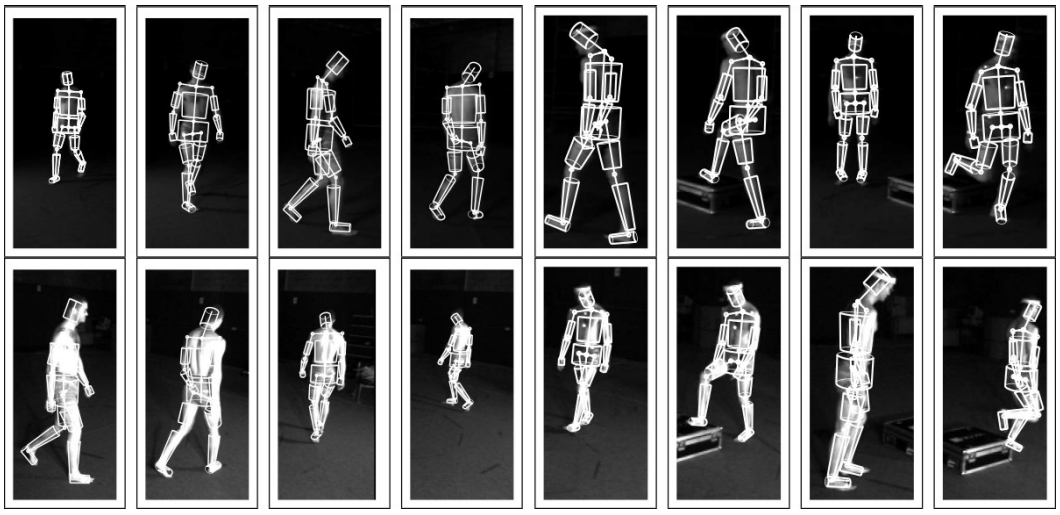


Fig. 2.4: Results of a pose estimation method [Deutscher *et al.*, 2000] which searches the space of 3D poses for matches within the image. The limbs are considered as being 3D cylinders. Figure from [Deutscher *et al.*, 2000].

Instead of building tedious generative models, pose estimation can be tackled as a learning problem. The pose estimation problem is formulated by [Ionescu *et al.*, 2011] as the selection of appropriate figure/ground proposals and the prediction of 3D joint positions, using a discriminative model. In [Agarwal and Triggs, 2004], a regressor from person silhouette to person pose is learned using a non-linear model. [Ramanan *et al.*, 2005] learned a pose-specific person detector and applied it to video frames. Once a person with the right pose was detected, an appearance model of each body part was learned and used to track the pose in the rest of

the video. [Ramanan, 2006] further extended this to drop the person detector step, and instead initialized the model using part-specific detectors based on image edges, before iteratively refining the appearance model of each part. This work was adapted to videos in [Ferrari *et al.*, 2008].

Later on, the deformable part models described in the previous section were adapted to pose estimation [Sapp *et al.*, 2010]. Instead of learning a single model, multiple detectors can be trained for different types of poses produced by clustering the training set [Johnson and Everingham, 2010; Johnson and Everingham, 2011]. Similarly, the method of [Yang and Ramanan, 2011] mentioned above in the context of person detection uses multiple part types and fully annotated part locations. Combining the two, in [Sapp and Taskar, 2013] multiple high-level pose components are used and are associated to a coarse template and to a full deformable part model like the one of [Yang and Ramanan, 2011]. Using a cascade approach to filter out unlikely pose components, this allows having more specific estimators which can produce finer results. In [Pishchulin *et al.*, 2013], an improved pose appearance model is used in combination with more expressive body part representations.

These methods, which represent a person pose as a tree of parts, were also extended to video, for instance by connecting the nodes corresponding to the same part in successive frames in a bigger pose graph [Sapp *et al.*, 2011]. Efficient inference was possible by iteratively optimizing over sub-trees of the graph. Another video extension was to use motion cues [Fragkiadaki *et al.*, 2013] to refine the pose estimation results from [Yang and Ramanan, 2011], which iteratively refines optical flow estimation and pose estimation in an alternated optimization scheme. The method of [Cherian *et al.*, 2014] samples multiple pose candidates for each person in each frame of the video and selects the best candidate by finding a track of poses which is coherent in time and with the video motion.

Deep learning advances were leveraged for pose estimation. For instance, in [Toshev and Szegedy, 2014] a multi-stage regressor outputs the position of the person joints given an image patch centered on a person. The first stage of this regressor is a deep convolutional neural network trained using the euclidean loss which outputs the position of all the joints, while the following stages refine the position of a joint given a zoomed-in patch centered on the previous stage estimate. In [Tompson *et al.*, 2014], part detectors are trained using deep networks, as well as a fully-connected spatial model. To train this model, the detectors are trained first and then used to train the spatial model, and then the two stages are combined and fine-tuned to improve performance even more. Convolutional networks are also used in [Chen and Yuille, 2014b] to train part detectors as well as to train image-dependant pairwise terms for a deformable part model similar to the one

of [Yang and Ramanan, 2011]. Combining the approaches above, [Fan *et al.*, 2015] uses multi-task learning to train a joint part detector and part localizer, taking as an input a full body patch containing the whole person and a part patch around the expected part location, as illustrated in Figure 2.5.

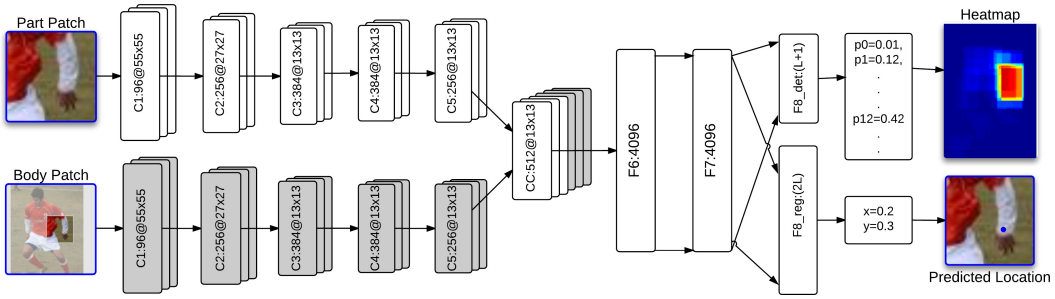


Fig. 2.5: Architecture of a deep neural network for part detection and localization [Fan *et al.*, 2015]. The network takes both a coarse patch of the whole person as well as a fine patch around the expected limb position, and outputs a heatmap for the part detection as well as the part location. Figure from [Fan *et al.*, 2015].

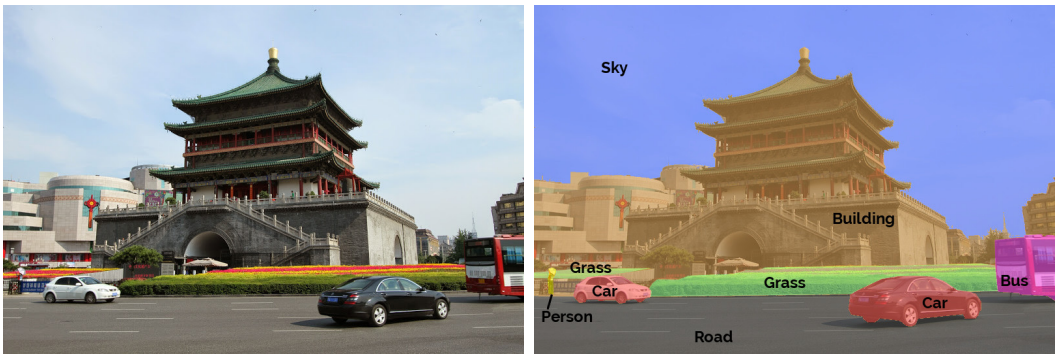
In our work, we use the pose estimator of [Yang and Ramanan, 2011], which we retrain in Chapter 4 to benefit from disparity maps. In Chapter 5, we use this method to estimate the poses of multiple persons in the image, and produce rough pose-specific segmentation masks which we include as a component of our multi-person segmentation model.

2.3 Segmentation

Segmentation is the task of grouping pixels in a coherent manner. It maps to the natural capability of our visual system to group nearby pixels into relevant regions [Wertheimer, 1923]. In computer vision, multiple types of segmentation tasks exists, from binary foreground/background segmentation to multi-class segmentation tasks, such as semantic segmentation, where each label corresponds to an object class (e.g.car, plane, person, road, sky), or instance-level segmentation, where each label maps to a different instance of an object class (e.g.multi-person segmentation). Typical difficulties for segmentation methods arise from the lack of clear boundaries between the different target segments, or at the opposite from the existence of boundaries within a target segment. In our work, we investigate the task of multi-person segmentation in videos, which is at the corner of semantic segmentation and multi-object segmentation in videos. In addition, we use cues from stereo videos as well as ideas related to co-segmentation methods. We describe works related to these problems in the following.

2.3.1 Semantic segmentation

Semantic segmentation is the task of grouping the pixels of an image which correspond to the same object class or concept. For instance, in the case of an urban environment, the classes can be cars, bicycles, road, pavement, building, sky, *etc*, as illustrated in Figure 2.6. It is often referred to as image or scene parsing, and can be seen as jointly performing recognition and segmentation.



(a) Original image **(b)** Ground-truth semantic segmentation

Fig. 2.6: Semantic segmentation aims at identifying regions of the image which correspond to a coherent object class or concept. We show here the original image (a) and a ground-truth segmentation label in the context of an urban scene, with persons, cars, roads, buildings, *etc*.

Many approaches for semantic segmentation are based on Markov Random Fields (MRFs) or Conditional Random Fields (CRFs) over pixels or superpixels [Corso *et al.*, 2008; Chen *et al.*, 2008; Russell *et al.*, 2009; Gould *et al.*, 2009; Kumar and Koller, 2010; Tighe and Lazebnik, 2010]. For instance, in the Graph-Shifts method [Corso *et al.*, 2008], a hierarchical graph is built, with nodes representing classes at the top level and nodes from a hierarchical over-segmentation below. An iterative graph manipulation procedure is applied to update the hierarchical labelling efficiently and optimize an MRF formulation for semantic segmentation. An extension of this method has also been applied to videos [Chen and Corso, 2011]. In [Kumar and Koller, 2010], the semantic segmentation problem is addressed as joint tasks of assigning pixels to regions and of assigning regions to semantic classes. The problem is formulated as an energy minimization problem, with an energy composed of two terms: one which evaluates the coherence of the pixels in each region, and one which evaluates whether each selected region label is appropriate. A similar approach is proposed by [Ion *et al.*, 2011], where *tiling sets* are considered over region proposals acquired using the Constrained Parametric Min-Cut (CPMC) algorithm [Carreira and Sminchisescu, 2012]. A tiling set is a set of region proposals which cover the image as much as possible but do not overlap each other. The proposed energy contains one term which evaluates the chosen tiling and one

term which evaluates the chosen labelling of each segment of the tiling. It is then optimized using a classic message-passing method [Kolmogorov, 2006].

Images can also be segmented in a hierarchical manner [Munoz *et al.*, 2010], by first computing a a hierarchy of over-segmentations. Then, given an image region and predictions made at the previous hierarchy levels, the method predicts the proportion of each class in the image.

Region proposals can be classified using pooled features, as shown by [Carreira *et al.*, 2012], where local features such as SIFT or local binary patterns (LBP) are pooled over region proposals using second-order operators (such as second-order max- or average-pooling). The pooled features are then mapped to an appropriate space and fed to a SVM to predict the region label.

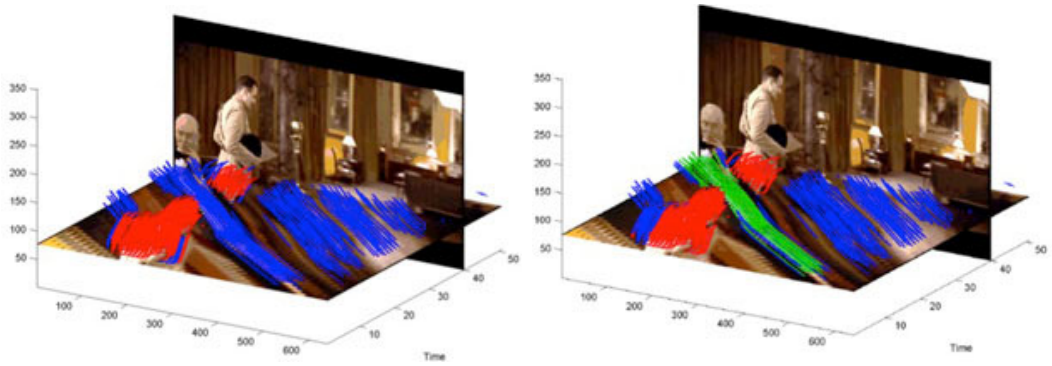
Using neural networks, convolutional features are extracted at multiple scales, and used in a segmentation tree to smooth out the prediction of semantic classes [Farabet *et al.*, 2013]. More recently, fully convolutional networks adapt state-of-the-art classification models to apply them in a sliding window fashion [Long *et al.*, 2015]. The score map produced by these methods can be further refined using a fully connected CRF [Chen *et al.*, 2015] to produce cleaner results. The CRF model can even be integrated as part of the neural network by formulating it as a Recurrent Neural Network (RNN) [Zheng *et al.*, 2015]. Doing so creates a deep network which can be trained end-to-end using back-propagation and which exhibits excellent performance and segmentation quality. Instead of using a sliding window approach to simply produce score maps, a multi-task neural network can also be trained to predict whether the input patch is centered on an instance of the considered object class and output a segmentation mask [Pinheiro *et al.*, 2015]. This approach has been shown to produce both excellent object detection performance and high quality segmentation masks.

Our work of Chapter 5 and Chapter 6 focuses on the segmentation of persons against the background, which is a form of semantic segmentation. In addition, we aim to segment each person against the rest and the background, which can be seen as multi-object segmentation, or even instance-level segmentation given that all the considered foreground objects belong to the same semantic class.

2.3.2 Multiple object segmentation in videos

The problem of object and multiple object segmentation in videos can be approached from multiple angles. Naturally, given the dynamic nature of videos, unsupervised motion-based segmentation is an efficient approach. Point tracks can

be used to group regions with a coherent motion through the video, for instance by clustering long term tracks [Ochs *et al.*, 2014]. Occlusion reasoning can be applied to long term tracks analysis to further improve the results [Lezama *et al.*, 2011], as illustrated in Figure 2.7. Using motion boundaries, the method of [Papazoglou and Ferrari, 2013] estimates an initial binary segmentation by assigning pixels inside the motion boundaries to the foreground segment. The segmentation is then refined using a spatio-temporal extension to GrabCut. Occlusion relationships can also be analyzed to recover a layered partition of each frame of a video and to segment each object of the video [Taylor *et al.*, 2015].



(a) Without occlusion reasoning **(b)** With occlusion reasoning

Fig. 2.7: Unsupervised video segmentation using long term tracks. Long term point tracks are computed and then aggregated based on the similarity of the motion (a). However, when the camera moves, static objects move similarly with the background. By reasoning on occlusions, static objects which are occluded by moving objects during the video can be identified and segmented separately.

Another approach is to track segmentation proposals through the entire video. Foreground segments can for instance be extracted using foreground/background segmentation based on motion, and then the produced blobs can be matched and connected across frames [Colombari *et al.*, 2007]. Many segmentation proposals can also be generated by methods such as CPMC, and then tracked and combined to produce the final segmentation [Li *et al.*, 2013; Banica *et al.*, 2013]. Instead of tracking proposals, clusters of coherent segments can be computed and used to compute appearance likelihoods for a space-time MRF to recover the segmentation. [Lee *et al.*, 2011].

Instead of trying to group regions having similar appearance or motion, it is possible to learn instance-specific object appearance models. For instance, in [Fathi *et al.*, 2011] soft segmentations are produced iteratively over a graph of superpixels and are used to measure the uncertainty of the segmentation. At each iteration, the labels of the most certain frame are frozen and are used to train appearance models in the form of weights for higher-order potentials. These appearance models are then used to produce the soft segmentations of the next iteration.

A slightly different but related task is the one of segmentation propagation, which has been tackled by ensuring the consistency of the segmentation over supervoxels [Jain and Grauman, 2014]. The problem is formulated as a space-time MRF over superpixels, with classical spatial and temporal neighborhood potentials as well as additional higher-order potentials encouraging the consistency of the labelling over an entire supervoxel.

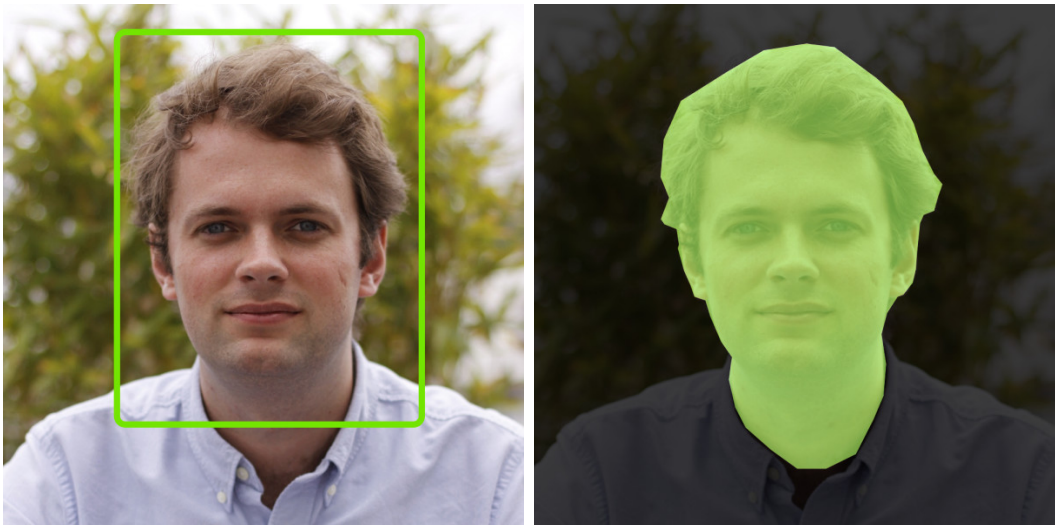
A notable work which does not focus on videos but is still relevant is a system targeted at multi-instance segmentation for cars in images [Zhang *et al.*, 2015b]. While it is restricted to a specific setup, the one of car-mounted cameras, it provides layered, instance-level segmentation for the cars in the view. A convolutional neural network is used to predict segmentation masks for at most 6 cars inside a single patch, in a layered fashion. The predictions at each location in the image are then merged by first detecting the connected components in the network outputs, and then formulating an MRF problem using priors over the ordering of the cars (the cars are the bottom of the image should be the foremost ones, while the ones at the top are likely to be behind) and short-range and long-range pairwise terms.

Other related works [Ladický *et al.*, 2013; Eichner *et al.*, 2012] have considered the case of jointly estimating the pose and segmentation of multiple people in a scene. Following previous works on jointly reasoning about poses of multiple upright people [Eichner and Ferrari, 2010], [Eichner *et al.*, 2012] proposes a pose estimation algorithm which supports multiple persons in a single scene and also outputs a soft segmentation mask for each person. The formulation proposed in [Ladický *et al.*, 2013] uses a candidate set of poses for finding a pixel-wise body part labelling of people in the scene. It combines multiple standard potentials over color and texture with potentials derived from rough segmentation masks estimated from the body parts locations of each candidate pose as well as potentials from the pose estimation itself. The solution to the pose estimation and segmentation problem is obtained by first greedily selecting high-quality candidates until adding the next best candidate degrades the quality of the initial solution. Then, the selected candidates are jointly refined until convergence.

In our work, we combine multiple cues such as appearance and motion cues to segment multiple persons or objects (potentially multiple instances from the same object class) in a video. In addition, in Chapter 5 we incorporate person detections, pose estimates and disparity cues and reason on the layering of the different persons in the scene, while in Chapter 6 we only require object tracks to perform the segmentation.

2.3.3 Segmentation using bounding boxes or pose estimates

Prior information, such as object bounding boxes or pose estimation can be used to guide the segmentation. Person and body-part detectors as well as skin color models have been used as unary potentials in CRFs [Lempitsky *et al.*, 2009; Rother *et al.*, 2004; Hernández-Vela *et al.*, 2012]. In the GrabCut framework, initial appearance models of the foreground object and of the background are computed from user inputs, such as scribbles (areas annotated by the user) or bounding boxes as in [Rother *et al.*, 2004], as illustrated in Figure 2.8. These models are then used to seed the unary potentials of a graph-cut problem, which is then solved, producing the desired segmentation. This procedure is then iterated, the appearance models being recomputed using the segmentation found at the previous iteration.



(a) Original image with bounding box **(b)** Binary segmentation result

Fig. 2.8: Given an image and a bounding box, the GrabCut algorithm [Rother *et al.*, 2004] builds an appearance model for the foreground using the pixels inside the box and one for the background using the pixels outside of the box, and solves a binary graph-cut problem to produce the segmentation. This procedure is then repeated until convergence by updating the appearance models at each iteration by using the segments found at the previous iteration.

Higher order terms of CRFs can also be used to encode priors from bounding boxes [Ladický *et al.*, 2010; Vineet *et al.*, 2011]. For instance, in [Ladický *et al.*, 2010], a higher order term of a pixel-wise CRF links all the pixels inside a bounding box to encourage their labels to be coherent. A similar term is used by [Vineet *et al.*, 2011] to perform the segmentation of multiple persons in a video, as well as terms related to body part detections.

A recently proposed task, simultaneous detection and segmentation [Hariharan *et al.*, 2014], closes the gap between object detection and segmentation. They do so by scoring region proposals produced by MCG [Arbeláez *et al.*, 2014] using a

method based on CNN features and a SVM classifier, and combine likely regions together to produce a segmentation mask and output object detections.

Recently, bounding boxes have been used to weakly supervise the training of a deep neural network for semantic segmentation [Dai *et al.*, 2015; Papandreou *et al.*, 2015]. In [Dai *et al.*, 2015], region proposals from MCG are combined with bounding boxes of an object class to learn an initial fully convolutional network for the segmentation task. The network is iteratively improved by combining the predicted masks from the network trained at the previous iteration with the most likely region proposal for each bounding box and updating the network with the produced, refined mask. In [Papandreou *et al.*, 2015], multiple strategies using bounding boxes or simply image-level tags are used to supervise the training of a fully convolutional neural network, for instance by considering the whole content of the bounding box as a positive label, or by using automatic binary segmentation within the bounding box to produce the positive segment. These methods produce results which are competitive with methods which require fully annotated object masks such as [Long *et al.*, 2015]. In addition, in both works, semi-supervised setups where pixel-wise ground truth masks are provided for a fraction of the training set and bounding boxes for the rest of the set are even more competitive with fully supervised setups.

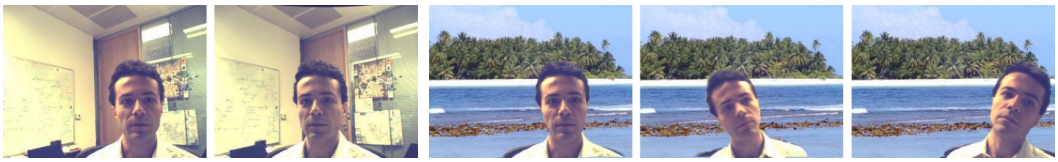
The use of object detectors as weak cues for semantic video segmentation has been explored in [Zhang *et al.*, 2015a]. In this work, given image-level object tags, object detections are produced according for the object classes specified by the tags, and region proposals are generated. The detections and proposals are tracked through the video, and then refined to only retain likely hypotheses.

In our work, bounding boxes are used in Chapter 6 to constrain the set of possible segmentations. These constraints guide the segmentation without having to explicitly build any model from their contents.

2.3.4 Person segmentation in stereo videos

The problem of segmenting a stereo video into foreground-background regions has been addressed for a teleconferencing set-up in [Kolmogorov *et al.*, 2005], with applications in graphics as illustrated in Figure 2.9. The sequences considered in this work involved only one or two people seated in front of a webcam, i.e., a restricted set of poses and at best, simple occlusions. They propose two algorithms for binary segmentation in stereo videos. The first one solves both the disparity estimation problem and the segmentation problem taking appearance matches and occlusions into account. The second one solves the segmentation problem without

explicitly involving the estimation of a disparity map. It formulates this problem as a tri-label pixel-wise graph cut and combines an appearance model with smoothness priors and a stereo consistency model.



(a) Stereo pair input **(b)** Output with automatic background substitution

Fig. 2.9: Illustration of the output and potential application of the foreground/background segmentation method for stereo videos of [Kolmogorov *et al.*, 2005]. The background can be subtracted and replaced by another one dynamically. Figure from [Kolmogorov *et al.*, 2005].

More recently, the model presented in [Sheasby *et al.*, 2012] uses disparity cues to perform human pose estimation and binary segmentation. They use the pose estimation method of [Yang and Ramanan, 2011] to recover the torso of the person and use two points from the torso as seeds for a flood fill algorithm over the disparity map. They further formulate the problems of disparity estimation, pose estimation and binary segmentation jointly, and solve them using a dual decomposition.

In our work, we integrate disparity cues either as a strong prior for segmentation in Chapter 5 or as a simple feature for a smoothness prior in Chapter 6.

2.3.5 Co-segmentation

Fig. 2.10: Co-segmentation aims at segmenting multiple images jointly by exploiting the appearance similarities of the objects present in all the pictures.

Co-segmentation is the task of simultaneously segmenting a set of images that all contain the same objects, exploiting the fact that the objects present in these images have similar appearance, as illustrated in Figure 2.10. Early works only considered pairs of images, such as [Rother *et al.*, 2006] where a classic binary segmentation MRF model is combined with a term which encourages the coherence of the color histograms of the foreground segments of the two images. Further works [Kim *et al.*, 2011; Joulin *et al.*, 2010] explored co-segmentation with many images. For instance, in [Joulin *et al.*, 2010] a global appearance model of the object class is learned jointly with the foreground/background segmentation in each image. It does so by minimizing an energy composed of a grouping term, which encourages

the smoothness of the solution in each image, and of a discriminative term, which encodes how well each segment fits the appearance model. Following the DIFFRAC discriminative clustering framework [Bach and Harchaoui, 2007], [Joulin *et al.*, 2010] uses the square loss for the discriminative term, which allows computing the appearance model in closed form when given the labels (the segmentation assignment), which in turn allows writing the whole energy as a convex function of the labels which can be optimized efficiently. The multi-class co-segmentation problem is tackled in [Joulin *et al.*, 2012] using the soft-max loss function and an expectation-maximization (EM) optimization scheme.

Note that this discriminative clustering approach has also been recently applied to other tasks: object co-localization in images [Tang *et al.*, 2014], finding actor identities in movies [Bojanowski *et al.*, 2013; Ramanathan *et al.*, 2014] and temporal action localization [Bojanowski *et al.*, 2014]. Each of these techniques is built upon a task-dependent set of constraints, modeling simple assumptions and encoding prior knowledge.

In our work, we cast the multi-instance object segmentation problem in videos as a multi-class discriminative clustering problem in Chapter 6, following the intuition that we are segmenting the same objects in all the frames of the video. We cast object tracks provided as an input into instance-specific constraints which shape the space of admissible segmentations.

3 Background Theory

In this chapter, we describe the background theory and methods used in the rest of the thesis. In particular, we focus on deformable part models for object detection (Section 3.1) and pose estimation (Section 3.2) used in Chapter 4. Conditional Random Fields (Section 3.3) is the theoretical framework behind the multi-label video segmentation method proposed in Chapter 5. Spectral (Section 3.4) and discriminative (Section 3.5) clustering methods are used in the convex multi-label segmentation framework of Chapter 6.

3.1 Deformable part models for object detection with LSVM

Given an image I, object detection is the task of predicting object locations (most often represented by bounding boxes) in I. This task can be reduced to the binary classification problem: given a bounding box p in I, the task is to predict whether the box tightly contains an object of the target category or not. Given a feature vector $\Phi(I, p)$ which represents the appearance of the image region in p and model parameters w, this classification task can be seen as thresholding a score function $f(p|I, w)$. Given such a binary classifier, detection can then be performed by classifying all bounding boxes inside image I, which can for instance be done using the sliding window approach (evaluate the classifier at all possible locations and scales) or using a cascade scheme (by skipping bounding boxes which are unlikely to contain the target object using an object proposal method or a faster classifier).

Linear classifiers where $f(p|I, w) = \langle w|\Phi(I, p)\rangle$ are a simple and efficient example of classifier for object detection, but they may not be able to capture the large space of configurations of certain object classes such as persons or cars, especially when using feature vectors $\Phi(I, p)$ which directly encode the spatial layout of the image. Indeed, with such features, the linear model w encodes a rigid template.

A more powerful approach is the one of pictorial structures [Felzenszwalb and Huttenlocher, 2005], where objects are represented by a graph (or often simply a star or a tree) of parts which can be deformed. Let us call $(\mathcal{V}, \mathcal{E})$ the vertices and edges of this graph. $i \in \mathcal{V}$ identifies one of the parts, and $(i, j) \in \mathcal{E}$ identifies one of

edge of the graph where part j is a child of part i. Parts encode local appearance information of the object and a spring-like model allows displacement of each part with respect to a subset of other parts. For instance in the case of a person class, parts can correspond to physical joints (shoulders, elbows, wrists), and the left wrist can move with respect to the left elbow, which itself can move with respect to the left shoulder. Figure 3.1 illustrates the person detection model from [Felzenszwalb *et al.*, 2013]. This model is a combination of a global object template, called the root template as illustrated in (a), and of local part templates, shown in (b), linked by spring-like connections, which behaviors are illustrated in (c).

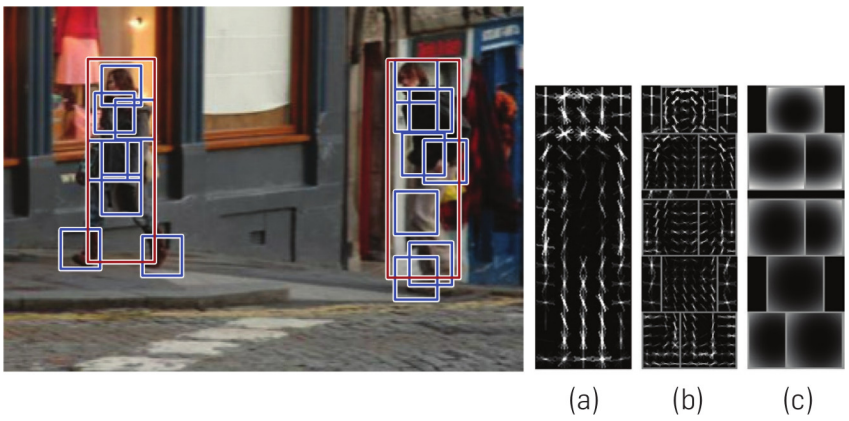


Fig. 3.1: Examples of person detections obtained with a deformable part model (left) and visualization of the global object template (a), part templates (b) and displacement costs (c). The global object template is fairly coarse, while part templates are finer, using features computed at a higher spatial resolution. The displacement costs encode the spatial model of the deformable part model, which penalizes the displacement of the part from their expected position within the bounding box. Figure from [Felzenszwalb *et al.*, 2013]

Information about part locations can be encoded as latent variables. By using the same notations as before, denoting the values of these latent variables by z and adapting our feature vector to incorporate information from z, the score function of our classifier can then be written as:

$$f(p, z|I, w) = \max_{z} \langle w|\Phi(I, p, z)\rangle \,. \tag{3.1}$$

In practice, a bias term b is usually added to the score function, so that thresholding is performed with a threshold at 0. For the sake of simplicity, we skip this term in the following as it simply offsets the threshold.

In the following, we describe the object detection method of [Felzenszwalb *et al.*, 2010]. The model, cost function and inference procedure are described in Section 3.1.1 and the training procedure in Section 3.1.2. In particular, this training procedure involves a framework called latent support vector machine (LSVM) which is able to train such latent-variable classifiers without requiring training data with part annotation.

3.1.1 Model and inference

Deformable part models for object detection can be expressed as graphical models, where parts are cast as nodes and links between connected parts are cast as edges. In these models, unary terms encode how well the part appearance correspond to the image appearance at the selected part location while binary terms encode how likely the displacement between two parts is.

Unary terms. To encode appearance information, templates which are usually linear filters are applied to a feature map computed over the whole image in a dense manner. Each element of the feature map is a d-dimensional feature vector which encodes the local appearance of the underlying image patch. With $p = (x, y, s)$, we denote by $\phi(I, p)$ the portion of the image feature map starting at image coordinates (x, y) at scale s. In practice, many practical implementations of deformable part models use histograms of oriented gradients (HOG) [Dalal and Triggs, 2005]. These features are extracted by first computing the horizontal and vertical gradients of the image and the orientation of the gradient vector at each location. These orientations are aggregated and discretized into histograms over small patches of the image. Formally, a filter F if a $w \times h \times d$ weight tensor which scores a portion of size $w \times h$ of the feature map by summing the dot products of each of the $w \times h$ feature vectors with the corresponding weight vector. We write this score as $\langle F | \phi(I, p) \rangle$.

Two levels of detail are used for the templates. A first template, usually called the root template or root filter, is used at the bounding box level and scores the global appearance of the entire bounding box. It is usually trained and evaluated on a coarse feature map, as its goal is to roughly localize the object. The rest of the templates are part templates. They are usually much smaller than the bounding box and trained and evaluated on higher resolution feature maps, as their goal is to precisely localize the object parts by capturing fine appearance details.

In the following, we will denote the root template by F_0 and the n part templates by F_1 to F_n.

Binary terms. The root template and the part templates form a star graph, with the root template being the root of the star. In the deformable part models, each edge of the star graph corresponds to a possible displacement between the expected location of the part relative to the root template. The allowed deformation is modeled using a spring-like quadratic cost. The i-th template, $1 \leq i \leq n$, is associated with a vector of deformation parameters d_i which encode the rest position and rigidity of the spring between the root template and the i-th template. If we denote by p_i

the position (location and scale) of the i-th part, we can then write the deformation cost as $\langle d_i|\psi(p_i,p_0)\rangle$ where $\psi(p_i,p_0)=(x_i-x_0,y_i-y_0,(x_i-x_0)^2,(y_i-y_0)^2)$.

Using these notations, and by noting $z=(p_1,\cdots,p_n)$, we can write the full score function:

$$f(p_0,z|I)=\sum_{i=0}^{n}\langle F_i|\phi(I,p_i)\rangle-\sum_{i=1}^{n}\langle d_i|\psi(p_i,p_0)\rangle. \tag{3.2}$$

Note that this score function can be expressed as a dot product between model parameters

$$w=(F_0,\cdots,F_n,d_1,\cdots,d_n)$$

and feature vectors

$$\Phi(I,p_0,z)=(\phi(I,p_0),\cdots,\phi(I,p_n),-\psi(p_1,p_0),\cdots,-\psi(p_n,p_0))$$

as:

$$f(p_0,z|I,w)=\langle w|\Phi(I,p_0,z)\rangle\,. \tag{3.3}$$

In turn, the score of a bounding box can be computed by finding the best parts placement:

$$\begin{aligned} f(p_0|I,w)&=\max_z f(p_0,z|I,w)=\max_z\langle w|\Phi(I,p_0,z)\rangle\\ &=\langle F_0|\phi(I,p_0)\rangle+\sum_{i=1}^{n}\max_{p_i}\langle F_i|\phi(I,p_i)\rangle-\langle d_i|\psi(p_i,p_0)\rangle. \end{aligned} \tag{3.4}$$

The score of all possible bounding boxes can be computed efficiently by using a generalized distance transform algorithm [Felzenszwalb and Huttenlocher, 2004a] as shown in Figure 3.2. This procedure involves independently applying each part template at all possible locations in the image and computing the contribution of this part to the score for all possible object bounding boxes using distance transform. The final score of each bounding box is then computed by summing the score of the root template and the score contribution of each part for this bounding box.

3.1.2 Training and LSVM

As shown above, given an image I and a bounding box p_0, our score function is:

$$f(p_0|I,w)=\max_z\langle w|\Phi(I,p_0,z)\rangle \tag{3.5}$$

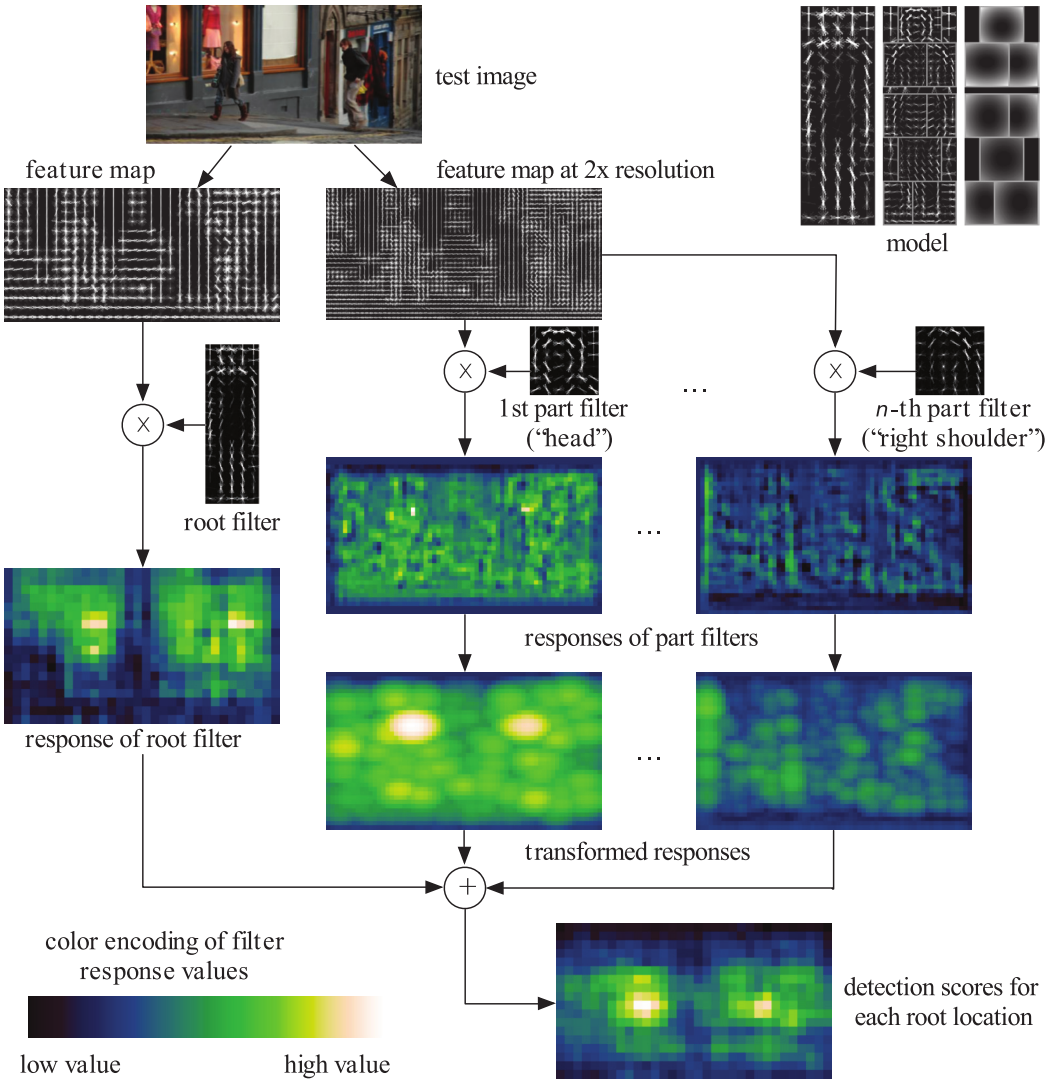


Fig. 3.2: Visualization of the efficient inference procedure for the deformable part model for object detection. Feature maps are extracted at a coarse and a fine resolution from the original image. Filter responses are computed over the entire feature maps using efficient convolution procedures. Filter response maps are then transformed using a generalized distance transform to incorporate the deformation costs. This produces maps which give the best possible part score contribution at each root part location. These maps are summed to produce the final scoremap, which gives detection scores at each possible object location and can be thresholded to produce a set of object detections. Figure from [Felzenszwalb *et al.*, 2013]

where z represents the vector of values of latent variables, w holds the model parameters and $\Phi(I, p_0, z)$ is the feature vector associated with the image I, the bounding box p_0 and the latent values z.

We add one entry to $\Phi(I, p_0, z)$ with value 1 and one entry to w to incorporate a bias in the model. Let us denote by $x_i = (I_i, p_{0i})$ a bounding box in an image and write $f(x_i|w) = f(p_{0i}|I_i, w)$. Then, given a set of N training examples $\mathcal{D} = (x_i, y_i), i \in \{1, \cdots, N\}$ with $y_i \in \{-1, 1\}$ the associated label to the example x_i, we can write the following objective function and optimization problem, analogous to the standard SVM formulation, using the hinge loss:

$$w^* = \arg\min_w \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i\in\mathcal{D}} \max(0, 1 - y_i f(x_i|w)) \tag{3.6}$$

with C the weight of the regularization term.

Given that $f(x_i|w)$ is convex (as being the maximum of a family of linear functions) but nonlinear in w, the objective function is non-convex in w. This challenge can be worked around by using an alternated optimization scheme. Indeed, when the values of the latent variables z_i are known, the score function becomes linear in w and the objective becomes the one of the linear SVM with the hinge loss. In this framework, called latent support vector machine (LSVM), at each optimization iteration:

1. The best assignment of the latent variables is recovered by performing the inference with given, fixed, model parameters w:

$$z_i = \arg\max_z \langle w|\Phi(I_i, p_{0i}, z)\rangle$$

2. The model parameters w are updated by fixing the latent variables and solving the corresponding convex optimization problem.

The other challenge of the training procedure for these deformable part models is that the training set is usually very unbalanced, with many more negative examples than positives examples. To work around this issue, a proper data mining scheme is required. A typical scheme is to use a small subset of the entire dataset while training, and regularly update it during the subset, keeping only hard positive and negative examples. This approach has been shown [Felzenszwalb *et al.*, 2013] to be guaranteed to find an optimal dataset and to terminate.

Initialization. Initial root templates are trained by learning a standard linear SVM over the feature maps computed over versions of the training bounding boxes

warped to have a common size. Parts positions are initialized in a greedy manner over salient regions of the root template, and initial part templates are derived as higher resolution versions of the corresponding root filter region, while deformation costs are initialized to penalize large displacements.

Mixtures of components and mixtures of parts. To capture even more variations, multiple object sub-types (called object mixture components) or part types (called part mixture components) can be considered in the model. Object mixtures can for instance deal with multiple object subcategories. For instance, in the case of cars, one can have one mixture component for sedan cars, one for sport cars and one for off-road vehicles.

Multiple object mixture components can be easily implemented by having a model w_k for each mixture component $k \in \{1, \cdots, K\}$ instead of a single one, and computing the best score over all components for each bounding box. The scoring function then becomes:

$$f(p_0|I, \{w_1, \cdots, w_K\}) = \max_k f(p_0|I, w_k). \tag{3.7}$$

Note that formally the model can still be written as a linear dot product, after concatenating the model weight vectors w_k into a single vector and building the feature vector appropriately. More specifically, an additional entry is added to the latent variables vector, specifying which component is selected ($z = (k, p_1, \cdots, p_n)$). If the feature vector of the single mixture case is of size D, and 0_D the vector of size D with all values being 0, the full feature vector is produced by concatenating 0_D $k-1$ times, the feature vector of the single mixture case with $z = (p_1, \cdots, p_n)$ and 0_D $K-k$ times.

At a finer level, part mixtures can help dealing with heavily deformable objects such as people. Instead of requiring many object mixture component to handle the very large variety of human poses, representing local part configurations (in the elbow case, the configurations could be straight elbow, half bent elbow, fully bent elbow, *etc*.) and pair-wise coherence between part mixture components allows representing an exponentially large space of configurations with a single model [Yang and Ramanan, 2011], as described in the next section.

3.2 Deformable part models for pose estimation

The deformable part models described in the previous section in the context of object detection do not require explicit part location annotations and automatically discovers significant object parts which may have no specific semantic meaning.

However, the task of pose estimation can be expressed as a structured prediction problem which goal is to predict the location of keypoints of the object. For instance, in the case of human pose estimation, these keypoints could be physical body joints. These keypoints can be seen as parts of the deformable part models framework, however the model needs to be extended and the training procedure adapted. In this section, we describe the pose estimation method of [Yang and Ramanan, 2011].

3.2.1 Model and inference

Using the same notations as in Section 3.1, the pose estimation problem can be formulated as:

$$z^* = \arg\max_z f(p_0, z|I) \tag{3.8}$$

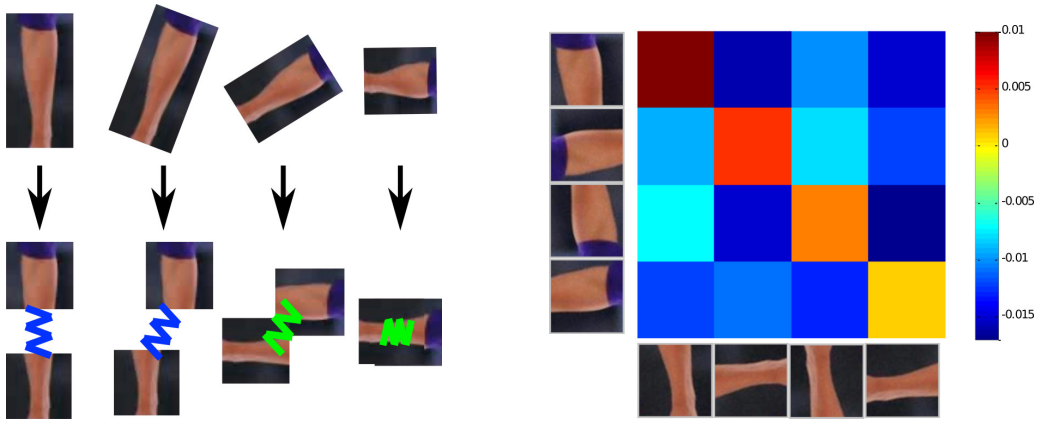
where p_0 is the root bounding box and I is the input image. Solving (3.8) implies finding the best arrangement of part positions at the root bounding box location. Note that in this context, the root bounding box does not necessarily need to contain the whole object. For instance, in the case of human pose estimation it is common place to have the root part correspond to the person head.

The first model modification is to use a tree of parts instead of a star graph. Here $\mathcal{V} = \{0, \cdots, n\}$ contains the $n+1$ parts, and $\mathcal{E}$ the n edges of the tree. This tree structure allows modeling an exponential number of pose configurations, yet inference can still be performed efficiently as shown below. With this structure, we adapt the notation of the deformation parameters for simplicity, and call d_{ij} the deformation parameters between part i and part j. The scoring function becomes:

$$f(p_0, z|I) = \sum_{i \in \mathcal{V}} \langle F_i | \phi(I, p_i) \rangle - \sum_{(i,j) \in \mathcal{E}} \langle d_{ij} | \psi(p_i, p_j) \rangle. \tag{3.9}$$

As discussed in Section 3.1, part mixtures can be incorporated in the model to deal with multiple part appearance patterns and capture an even larger number of pose configurations. Formally, this implies incorporating more latent variables, one per part, to identify the selected part mixture component. If there are T part types for each part, then we add variables $t_i \in \{1, \cdots, T\}$, $i \in \{0, \cdots, n\}$, t_i identifying the part type of part i. Instead of having 1 filter per part and $n+1$ filters in total, we now have $T \times (n+1)$ filters $F_i^{t_i}$. The deformation cost parameters also become part-types dependant, as illustrated in Figure 3.3 (a), and are now called $d_{ij}^{t_i,t_j}$. With $z = (p_1, \cdots, p_n, t_0, \cdots, t_n)$, the scoring function becomes:

$$f(p_0, z|I) = \sum_{i \in \mathcal{V}} \left\langle F_i^{t_i} | \phi(I, p_i) \right\rangle - \sum_{(i,j) \in \mathcal{E}} \left\langle d_{ij}^{t_i,t_j} | \psi(p_i, p_j) \right\rangle. \tag{3.10}$$



(a) Deformation cost **(b)** Co-occurrence prior

Fig. 3.3: Examples of part mixtures and associated deformation costs (a) and co-occurrence priors (b). Each part is associated to multiple candidate configurations, and the deformation costs between a child part and its parent part depends on the type of the parent part, as illustrated in the case of an arm in (a). In addition, a co-occurrence prior is learned from training data to encourage combinations of parent and child part types which were seen in the training data, as shown in (b). Figures from [Yang and Ramanan, 2013] and `http://www.di.ens.fr/willow/teaching/recvis14/slides/lecture07_structured_models.pdf`.

The occurrence of each part type and co-occurrence of pairs of connected part types can be taken into account in the score function as well, as illustrated in Figure 3.3 (b). By introducing new parameters $b_i^{t_i}$, $i \in \mathcal{V}$, which enables preferring certain part types, and $b_{ij}^{t_i,t_j}$, $(i,j) \in \mathcal{E}$, which enables favoring combinations of certain part types, this score can be written as:

$$\text{typesscore}(z) = \sum_{i \in \mathcal{V}} b_i^{t_i} + \sum_{(i,j) \in \mathcal{E}} b_{ij}^{t_i,t_j}. \tag{3.11}$$

Once incorporated in the score function, the latter becomes:

$$f(p_0, z|I) = \sum_{i \in \mathcal{V}} b_i^{t_i} + \left\langle F_i^{t_i} | \phi(I, p_i) \right\rangle + \sum_{(i,j) \in \mathcal{E}} b_{ij}^{t_i,t_j} - \left\langle d_{ij}^{t_i,t_j} | \psi(p_i, p_j) \right\rangle. \tag{3.12}$$

Let us note that as in Section 3.1.1, the score function $f(p_0, z|I)$ can be expressed as a dot product between an appropriately built vector of model parameters and an appropriately built feature vector.

Let us now rewrite the score function to follow the tree structure. Let $f_i(p_i, t_i|I)$ be the best score of the sub-tree starting from part i placed at position p_i with type t_i. It can be written as:

$$f_i(p_i, t_i|I) = b_i^{t_i} + \left\langle F_i^{t_i}|\phi(I, p_i)\right\rangle + \sum_{j,(i,j)\in\mathcal{E}} s_j(t_i, p_i|I), \tag{3.13}$$

$$s_j(t_i, p_i|I) = \max_{t_j} b_{ij}^{t_i,t_j} + \max_{p_j} f_j(p_j, t_j|I) - \left\langle d_{ij}^{t_i,t_j}|\psi(p_i, p_j)\right\rangle. \tag{3.14}$$

Then:

$$f(p_0|I) = \max_{t_0\in\{1,\cdots,T\}} f_0(p_0, t_0|I). \tag{3.15}$$

As before, this score function can be efficiently evaluated over the whole image using dynamic programming. Indeed, once the scores of the sub-trees of a node have been evaluated, the score of the node can be efficiently computed using the same generalized distance transform method as for the star graph case. By applying this method from the leaves to the root of the tree, the score function can be efficiently evaluated and the best part locations and types obtained by keeping track of the $\arg\max$ of each $\max$ operation, and backtracking from the best root score.

Implementation details for human upper body pose estimation

In this section we illustrate this pose estimation method by giving a few implementation details for the human upper body pose estimation model of [Yang and Ramanan, 2011]. In this model, there are 18 parts: head, neck, left and right shoulder, elbow, wrist and hips, plus several additional parts regularly spaced between the physical body joints, as shown in Figure 3.4. These additional parts aim at having a fairly dense coverage of the person body while using filters which are fairly small in terms of spatial extent, to avoid taking large parts of the background into account. The training data is produced from annotated poses with the 10 physical joints, and the positions of the additional parts is interpolated from the neighboring physical joints. Each part mixture model has $T = 6$ components. The training examples for a given part are clustered into the T components by running the K-means algorithm with $K = T$ over the set of relative position between the part example and its parent part.

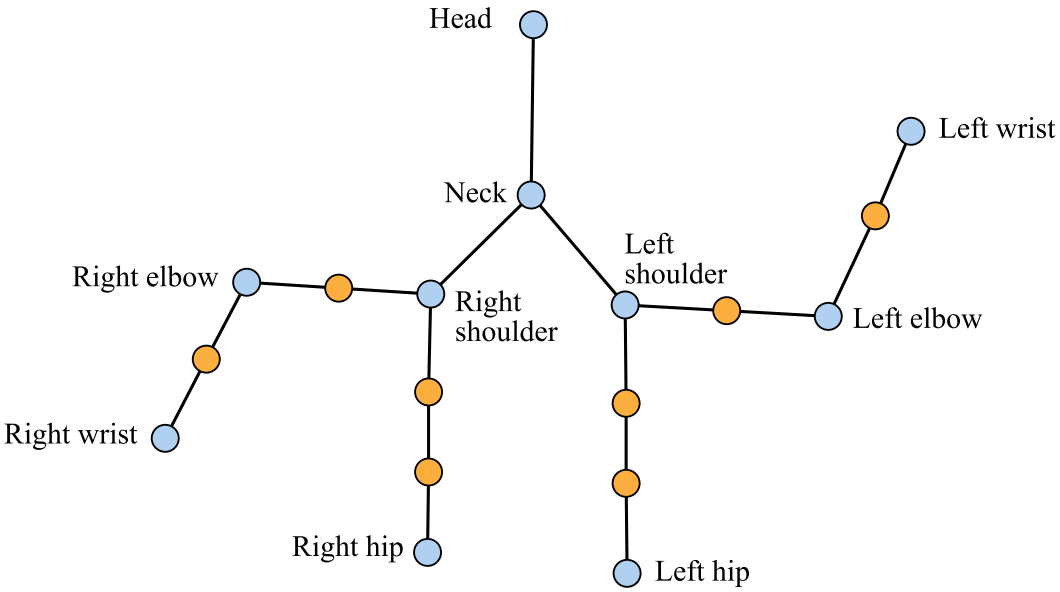


Fig. 3.4: Structure of the tree of parts for human upper body pose estimation from [Yang and Ramanan, 2011]. The nodes in blue correspond to parts that relate to physical body joints which are annotated in the training dataset. The nodes in orange correspond to additional parts which are added to have a denser coverage of the person surface.

3.2.2 Training procedure

The learning setup of this method is slightly different than the one previously used for object detection. Indeed, while the model can still perform object detection, the level of supervision for the positive examples is much higher, since both part positions (from annotations or interpolation) and part types (from part examples clustering) are known. We are thus given a dataset $\mathcal{D}$ with a set of positive pose examples pos and a set of negative examples neg, for a total of N training examples. An element i of pos corresponds to a tuple (x_i, z_i) where $x_i = (I_i, p_{0_i})$ specifies an image and root position for the pose and $z = (p_1, \cdots, p_n, t_0, \cdots, t_n)$ specifies the ground truth values of the part positions and types. An element i of neg corresponds to an x_i specifying an image and root position which does not contain the target object.

The training problem from [Yang and Ramanan, 2011] is formulated as:

$$\begin{aligned} &\arg\min_{w,\xi_i \geq 0} \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad &\forall i \in \mathrm{pos}, \quad f(x_i, z_i|w) \geq 1 - \xi_i \\ &\forall i \in \mathrm{neg}, \forall z, \quad f(x_i, z|w) \leq -1 + \xi_i \end{aligned} \tag{3.16}$$

It aims at learning a model which

1. gives high scores to the true pose of the positive examples

2. gives low scores to all possible poses in negative examples, since they contain no target object and thus no valid pose.

The constraints used to encode these desired properties of example i can be violated at the expense of paying a penalization cost, encoded in the slack variable ξ_i.

It is equivalent to the following optimization problem, which we can relate to the one for object detection:

$$\begin{aligned}\arg\min_{w} \frac{1}{2}\|w\|^2 + \frac{C}{N}\Bigg(\sum_{i\in\text{pos}} \max(0, 1 - f(x_i, z_i|w)) \\ + \sum_{i\in\text{neg}} \max_{z} \Big(\max\left(0, 1 + f(x_i, z|w)\right)\Big)\Bigg).\end{aligned} \tag{3.17}$$

This formulation is similar to the one from Equation 3.6 and could be optimized in a similar manner, by iteratively alternating between the selection of optimal latent values z for the negative examples and optimization of the model parameters w. In practice, a dual coordinate-descent solver has been used to optimize the problem from Equation 3.16 by making the negative optimization examples associated to a single negative example from the dataset share the same slack variable [Ramanan, 2013]. This is necessary as the number of constraints can be exponentially large, even though only a few of them are usually active.

3.3 Conditional Random Fields for segmentation

Foreground-background image segmentation consists in assigning a binary label $y_i \in \{0; 1\}$ to each pixel i of an image I, usually described by a descriptor $x_i \in \mathbb{R}^d$. A simple approach at solving this problem would be to model the appearance of foreground and background segments, for instance by building color histograms h_F and h_B for foreground and background respectively using prior knowledge on the content of the image or using labels given for certain pixels. However, this could lead to local errors for instance where the foreground locally looks like the background. These errors of the pixel-wise predictions could be avoided, or at least smoothed out, by adding a spatial consistency model and predicting jointly all the labels.

Going from independent predictions to structured predictions can be done by introducing graphical models. These models allow reasoning over a set of random

variables, which are in our example the pixels represented by their descriptors x_i and binary labels y_i. The descriptors x_i are observed variables, while labels y_i are hidden variables aimed at explaining the observations. Let $G = (\mathcal{V}, \mathcal{E})$ be the graph used in the model of this segmentation problem. Each of the random variables is represented by a node $v \in \mathcal{V}$ of the graph. The set of undirected edges $\mathcal{E}$ defines the connections between the variables, which can for instance map to the spatial structure of the problem. In the case of images, it can be as simple as the typical 4-neighborhood which connects each pixel to its adjacent pixels on the horizontal and vertical axes. Examples of neighborhoods are shown in Figure 3.5. For segmentation, in addition to the spatial structure of the problem, we also want to link the observation of a pixel x_i to the corresponding label y_i, resulting in a graph structure illustrated in Figure 3.6.

Once the graph structure is set, we can now formulate and reason over the joint or conditional probability distribution of the random variables. This can be done by using the framework of Markov Random Fields (MRF). This well studied framework can be used as long as the random variables satisfy the local Markov property, which requires that each variable is conditionally independent of all other variables given its neighbors. A MRF defines a joint probability distribution over the random variables as:

$$p(y, x) = \frac{1}{Z} \prod_{C \in \mathcal{G}} \Psi_C(\mathcal{V}_C) \tag{3.18}$$

where $\mathcal{C}(G)$ is the set of cliques of G, $\mathcal{V}_C$ the subset of variables in a clique C, Ψ_C a potential function for this clique and Z the normalizing factor:

$$Z = \sum_{y} \prod_{C \in \mathcal{G}} \Psi_C(\mathcal{V}_C). \tag{3.19}$$

For our example problem, this probability distribution could be written as:

$$p(y, x) = \frac{1}{Z} \prod_{y_i} \Psi(y_i, x_i) \prod_{(y_i, y_j) \in \mathcal{E}} \Phi(y_i, y_j) \tag{3.20}$$

where $\Psi(y_i, x_i)$ evaluates how well the pixel descriptor x_i fits the prior model for label y_i and $\Phi(y_i, y_j)$ encourages neighboring pixels to have the same label. In the literature, $\Psi(y_i, x_i)$ would often be called an unary potential, as it reason on the variables related to a single physical pixel, and $\Phi(y_i, y_j)$ a pairwise potential as it models the interaction between neighboring pixels. The unary potential can for instance evaluate how well the pixel color fits the estimated color models h_F and h_B of foreground and background:

$$\Psi(y_i, x_i) = h_F(x_i) y_i + h_B(x_i)(1 - y_i) \tag{3.21}$$

The pairwise potential can be derived from the classical Ising prior from [Winkler, 2003]:

$$\Phi(y_i, y_j) = \exp(-\lambda|y_i - y_j|) \tag{3.22}$$

The proposed model is a generative model, and can be used both for generating similarly looking images or to find the best labelling y^* given observations x by finding the MAP:

$$y^* = \arg\max_y p(y, x). \tag{3.23}$$

This probabilistic formulation can easily be transformed into an energy optimization problem by taking the negative log of $p(y, x)$:

$$E(y, x) = \sum_{y_i} \psi(y_i, x_i) + \sum_{(y_i, y_j) \in \mathcal{E}} \phi(y_i, y_j). \tag{3.24}$$

While the inference problem in general MRFs is NP-hard, certain types of problems can be optimized efficiently. In particular, energy functions of the form of the one of Equation 3.24 which only contain pairwise potentials and is submodular can be minimized exactly using graph cuts. More specifically, this implies building an appropriate graph, containing a pair of special nodes (s, t) linked to all other nodes of the graph, find the minimum cut of the graph by solving the maximum flow problem between nodes s and t, and recover the labeling from the edges which were cut [Ishikawa, 2003].

When exact inference is not possible, multiple methods exist for approximate inference, such tree-reweighted message passing [Kolmogorov, 2006] or α-expansion [Boykov *et al.*, 2001].

To model more complex interactions or priors, a variant of MRFs called Conditional Random Fields (CRFs) can be used. As the name implies, instead of modeling the joint distribution of all variables, it is used to model the conditional distribution of the hidden variables given the observed variables $p(y|x)$. Comparatively, CRFs avoid the need of modeling the distribution over the observations. As such, they are a form of discriminative model, while MRFs in general are expressed as generative models.

In terms of expressive power, the pairwise terms of a CRF can be made observation-dependant at little cost. For instance, in our segmentation example we can replace the pairwise term $\Phi(y_i, y_j)$ by $\Phi(y_i, y_j, x_i, x_j)$

$$\Phi(y_i, y_j, x_i, x_j) = \exp\left(-\lambda|y_i - y_j| \exp(-\gamma\|x_i - x_j\|^2)\right) \tag{3.25}$$

to take into account the difference of appearance of the neighboring pixels. This way we can encourage similarly looking neighbors to take the same label, but not penalize label mismatches in case the neighbors do not look the same.

3.4 Spectral clustering and normalized cuts

Spectral clustering is a popular unsupervised method to partition data points into a given number of clusters. Given similarity for all pairs of data points, the method aims at assigning similar points to the same clusters and dissimilar points to different clusters. It relies on the study of the eigenvectors (the spectrum) of a matrix derived from the similarity measurements to perform dimensionality reduction. It is simple to implement and can be solved using standard, efficient linear algebra tools.

Formally, let $\mathcal{V} = \{x_1, \cdots, x_n\}$ be the set of n input data points and $s_{ij} \geq 0$ be the similarity between points x_i and x_j. In this setup the similarity is symmetric, so that $s_{ij} = s_{ji}$. We denote by $\mathcal{E})$ the edges of the similarity graph, by $w_{ij} \geq 0$ the weight of the edge between x_i and x_j, and by W the weighted adjacency matrix of the graph. Typically this graph $(\mathcal{V}, \mathcal{E})$ is fully connected: each pair of points (x_i, x_j) such that $s_{ij} > 0$ leads to an edge in $\mathcal{E}$ with $w_{ij} = s_{ij}$. In many cases, the similarity is itself derived from the data points for instance using the Gaussian similarity function $s(x_i, x_j) = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$ if $x_i \in \mathbb{R}^d$.

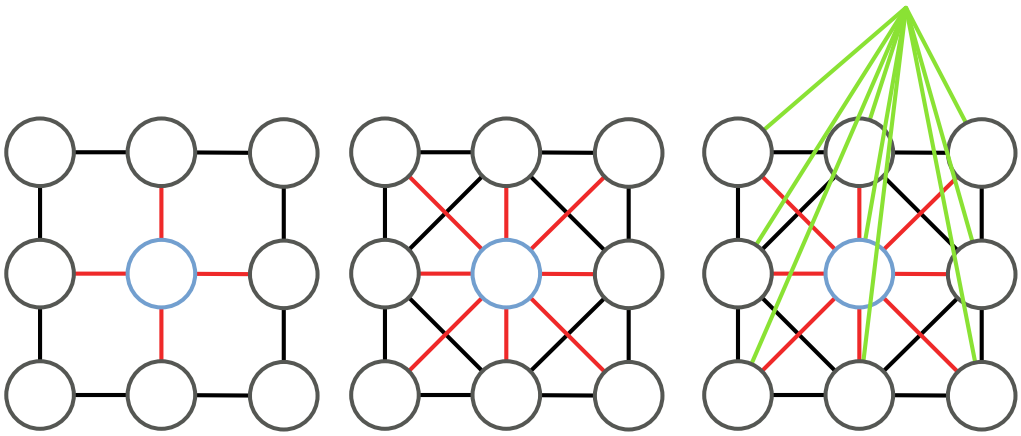
Multiple approaches have been proposed to perform spectral clustering. They all revolve around building new representations of data points by solving an eigenvector problem on a matrix L, called the Laplacian matrix, which is derived from the weighted adjacency matrix W. The new representations are usually much easier to cluster using techniques such as the k-means algorithm. There are multiple types of Laplacian matrices, and they are at the core of the spectral graph theory [Chung, 1997]. All of the formulations of these matrices involve the degree matrix D, which is a diagonal matrix such that the i-th diagonal element is:

$$d_{ii} = \sum_{j=1}^{n} w_{ij}.$$

This matrix counts the weights attached to each vertex of the graph. Typical Laplacian matrices include:

- the unnormalized Laplacian matrix:

$$L = D - W \tag{3.26}$$



(a) 4-neighborhood **(b)** 8-neighborhood **(c)** higher-order

Fig. 3.5: Examples of connectivity for graphical models. Considering the node in blue, we show (a) a simple 4-neighborhood pairwise model (with edges drawn in red), (b) a 8-neighborhood pairwise model and (c) a more complex model with an additional higher-order connection which connects all nodes to all others

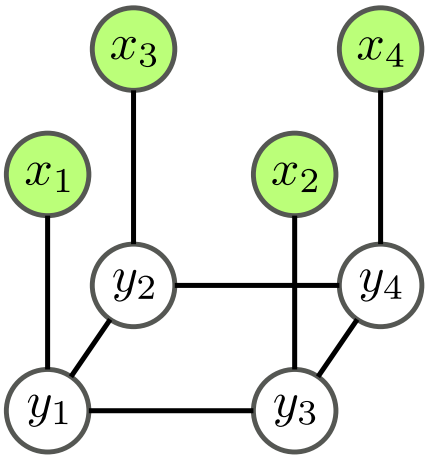


Fig. 3.6: Example of graphical model: each observation x_i is linked to the corresponding output y_i, and pairs of outputs corresponding to pairs of neighbor pixels are linked together.

- the random-walk normalized Laplacian matrix:

$$L_{rw} = I_n - D^{-1}W \tag{3.27}$$

- the symmetric normalized Laplacian matrix:

$$L_{sym} = I_n - D^{-1/2}WD^{-1/2} \tag{3.28}$$

These matrices are positive semi-definite and have n non-negative, real-valued eigenvalues. Their smallest eigenvalue is 0.

Let $y \in \{0,1\}^{n\times k}$ be an assignment matrix of our n data points into k clusters. The matrix y is such that y_{ic} is equal to one if and only if the point i is assigned to cluster c. The spectral clustering objective is defined as:

$$E(y) = \mathrm{Tr}(y^T L y). \tag{3.29}$$

For instance, when using the unnormalized Laplacian matrix, $E(y)$ translates to:

$$E(y) = \mathrm{Tr}(y^T L y) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left(\sum_{c=1}^{k}(y_{ic} - y_{jc})^2\right), \tag{3.30}$$

which intuitively means that whenever two points are in a different cluster, a cost w_{ij} is paid. In turn, this means that a low $E(y)$ corresponds to a clustering where pairs of points picked from two different clusters have low similarities. Spectral clustering techniques allow finding an assignment matrix y such that $E(y)$ is low by studying the spectrum of L and building a new representation of the data points in $\mathbb{R}^k$ using the k smallest eigenvectors of L. For instance, in Algorithm 1, we describe the normalized spectral clustering algorithm of [Shi and Malik, 2000].

Algorithm 1: Normalized spectral clustering algorithm [Shi and Malik, 2000]

Input: Similarity matrix $S = (s_{ij})_{i,j=1,\cdots,n}$ and desired number of clusters k.

Algorithm:

Build the similarity graph $(\mathcal{V}, \mathcal{E})$ and its adjacency matrix W.
Compute the random-walk normalized Laplacian L_{rw}.
Compute the k eigenvectors $u_1, \cdots, u_k$ of L_{rw} corresponding to the k smallest eigenvalues of L_{rw}.
Build the matrix $U \in \mathbb{R}^{n\times k}$ by stacking horizontally $u_1, \cdots, u_k$.
Let $v_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
Cluster the points v_i using the k-means algorithm, producing an assignment matrix $Y = (y_{ij})_{i=1,\cdots,n\ j=1,\cdots,k} \in \{0;1\}^{n\times k}$.

Output: Assignment matrix Y such that $y_{ij} = 1$ iff x_i belongs to cluster j.

3.4.1 Normalized cuts

A typical application of spectral clustering is the *normalized cut* problem [Shi and Malik, 2000]. In graph theory, a *cut* $C = (A, B)$ is the partition of the vertices of a graph $(\mathcal{V}, \mathcal{E})$ into two disjoint subsets A and B. In terms of adjacency graph, it implies removing the edges which connect vertices from one subsets to vertices of the other subset. Using the previous notations, the cost of this cut is:

$$\text{cut}(A, B) = \sum_{x_i \in A,\ x_j \in B} w_{ij}. \tag{3.31}$$

This cost evaluates the amount of similarity which has been ignored when removing the edges between the two subsets. Finding the minimum cut of a graph, which leads to the optimal bi-partitioning of the graph, is a classical and well-studied problem, with efficient solving approaches. Namely, the min-cut max-flow theorem states that finding the minimum cut of a graph is equivalent to solving a maximum flow problem over this graph. The maximum flow problem can be solved in polynomial time, for instance using the Edmonds-Karp algorithm [Edmonds and Karp, 1972].

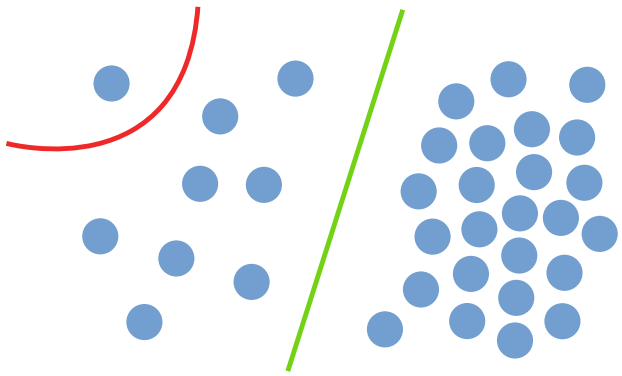


Fig. 3.7: Toy example where finding the minimum cut would lead to a cluster containing a single element (cut drawn in red) and where a normalized cut would lead to a better partitioning (cut drawn in green)

This cost cut function can be used for clustering. However, it is naturally biased towards forming unbalanced clusters, which a large one and a small one, as the cost increases with the number of edges originally connecting the two subsets, as shown by [Wu and Leahy, 1993] and illustrated in Figure 3.7. To work around this issue, a new type of cut has been proposed by Shi and Malik, 2000, the normalized cut. Instead of just taking into account the weight of the edges between the two partitions, it also takes into account all the weight of all of edges starting from each partition:

$$\text{assoc}(A, \mathcal{V}) = \sum_{x_i \in A, x_j \in \mathcal{V}} w_{ij}. \tag{3.32}$$

The cost function is formulated as:

$$\mathrm{Ncut}(A,B) = \frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(A,\mathcal{V})} + \frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(B,\mathcal{V})}. \tag{3.33}$$

This rebalances the cut problem: forming a partition with a single isolated vertex will cost 1 (as all the edges of this partition are cut), which may no longer minimize the cost as it might have been the case with the standard min-cut problem.

Using the previous notations for D, W and by introducing the assignment vector $y \in \{0,1\}^n$, the normalized cut problem:

$$\min_{A,\ B} \mathrm{Ncut}(A,B) \tag{3.34}$$

can be rewritten as:

$$\begin{aligned} &\min_{y} \frac{y^T(D-W)y}{y^T D y} \\ &\text{s.t.} \quad y^T D \mathbb{1}_n = 0. \end{aligned} \tag{3.35}$$

This is a quadratic problem under constraints, which is known to be NP hard when y takes binary values, as it is as hard as solving a max-cut problem. A continuous relaxation, by letting y take real values, can be solved as it is equivalent to solving the generalized eigenproblem

$$(D-W)y = \lambda D y. \tag{3.36}$$

It appears that normalized cut is a spectral clustering problem aiming at finding $k = 2$ clusters. The clusters can be recovered from the eigenvector corresponding to the second smallest eigenvalue of the L_{rw} matrix, for instance by thresholding the values of the eigenvector.

In our work, we use a spectral clustering term as part of the objective function of the multi-instance segmentation method described in Chapter 6.

3.5 Discriminative clustering with the square loss

Discriminative clustering is a family of unsupervised methods, such as DIFFRAC [Bach and Harchaoui, 2007] or [Guo and Schuurmans, 2007], aimed at clustering data in such a manner that if the produced clusters were provided to a supervised technique, the training error would end up as low as possible. Provided that features are discriminative enough, these techniques are proven to be more robust to noise than generative ones.

Formally, we are given N data points represented by vectors $x_1, \cdots, x_n$ in $\mathbb{R}^d$, stacked in a matrix $X \in \mathbb{R}^{N \times d}$, and our task is to partition them into K clusters. Let us define the label matrix $y \in \{0,1\}^{N \times K}$. The matrix y is such that y_{nk} is equal to one if and only if the point n is assigned to cluster k. Since each point can only belong to a single cluster, y is also such that $y\mathbb{1}_K = \mathbb{1}_N$, where $\mathbb{1}_K$ (resp. $\mathbb{1}_N$) is the constant vector of size K (resp. N) with all entries equal to one. We denote by $\mathcal{Y}$ the set of admissible label matrices y.

The supervised multi-label classification problem, given input data X and labels y is to find a predictor f^* such that:

$$f^* = \arg\min_f \; l(y, f(X)) + \mu \, r(f) \tag{3.37}$$

with l a loss function, r a regularizer and μ a weight between the two terms.

Following [Bach and Harchaoui, 2007], using the square loss $l(y, f(X)) = \frac{1}{N}\|y - f(X)\|_F^2$, a linear model $f(X) = Xw + \mathbb{1}_N b$ with $w \in \mathbb{R}^{d \times K}$ and $b \in \mathbb{R}^{1 \times K}$ and a L_2 regularization $r(f) = \|w\|_F^2$, this problem becomes:

$$w^*, \; b^* = \underset{w \in \mathbb{R}^{\times \mathbb{K}}, \; b \in \mathbb{R}^{1 \times K}}{\arg\min} \frac{1}{N}\|y - Xw - \mathbb{1}_N b\|_F^2 + \mu \, \|w\|_F^2. \tag{3.38}$$

This cost function can be minimized in closed form. Indeed, as $\|A\|_F^2 = \mathrm{Tr} A^T A$, the cost can be written as:

$$\begin{aligned}
&\frac{1}{N} \, \mathrm{Tr}((y - Xw - \mathbb{1}_N b)^T (y - Xw - \mathbb{1}_N b)) + \mu \, \mathrm{Tr} \, w^T w \\
&= \frac{1}{N} \, \mathrm{Tr}((y^T - w^T X^T - b^T \mathbb{1}_N^T)(y - Xw - \mathbb{1}_N b)) + \mu \, \mathrm{Tr} \, w^T w \\
&= \frac{1}{N} \, \mathrm{Tr}(y^T y - 2y^T Xw + w^T X^T Xw + 2b^T \mathbb{1}_N^T Xw - 2y^T \mathbb{1}_N b + N b^T b) + \mu \, \mathrm{Tr} \, w^T w.
\end{aligned} \tag{3.39}$$

After derivation with respect to b, b^* is:

$$\begin{aligned}
0 &= 2w^{*T} X^T \mathbb{1}_N - 2y^T \mathbb{1}_N + 2N b^{*T} \\
\Rightarrow N b^{*T} &= y^T \mathbb{1}_N - w^{*T} X^T \mathbb{1}_N \\
\Rightarrow N b^* &= \mathbb{1}_N^T (y - Xw^*) \\
\Rightarrow b^* &= \frac{1}{N} \mathbb{1}_N^T (y - Xw^*).
\end{aligned} \tag{3.40}$$

After derivation with respect to w, and by writing $\Pi_N = I_N - \frac{1}{N}\mathbb{1}_N\mathbb{1}_N^T$ the centering projection matrix, we get that w^* is:

$$\begin{aligned}
0 &= -2y^TX + 2w^{*T}X^TX + 2b^{*T}\mathbb{1}_N^TX + 2N\ \mu\ w^{*T}I_N \\
\Rightarrow 0 &= -y^TX + w^{*T}X^TX + \frac{1}{N}(y^T - w^{*T}X^T)\mathbb{1}_N\mathbb{1}_N^TX + N\ \mu\ w^{*T}I_N \\
\Rightarrow 0 &= -y^TX + w^{*T}X^TX + \frac{1}{N}y^T\mathbb{1}_N\mathbb{1}_N^TX - \frac{1}{N}w^{*T}X^T\mathbb{1}_N\mathbb{1}_N^TX + N\ \mu\ w^{*T}I_N \\
\Rightarrow w^{*T}(X^TX - \frac{1}{N}X^T\mathbb{1}_N\mathbb{1}_N^TX + N\ \mu\ I_N) &= y^TX - \frac{1}{N}y^T\mathbb{1}_N\mathbb{1}_N^TX \\
\Rightarrow w^{*T}(X^T\Pi_NX + N\ \mu\ I_N) &= y^T\Pi_NX \\
\Rightarrow w^* &= (X^T\Pi_NX + N\ \mu\ I_N)^{-1}X^T\Pi_Ny.
\end{aligned} \tag{3.41}$$

After reinjecting w^* and b^* into the cost function, the optimal objective value is:

$$\operatorname{Tr}\left(y^TA(X,\mu)y\right) \tag{3.42}$$

with $A(X,\mu)$ the positive, semi-definite matrix defined as:

$$A(X,\mu) = \frac{1}{N}\Pi_N(I_N - X(X^T\Pi_NX + N\mu I_N)^{-1}X^T)\Pi_N. \tag{3.43}$$

The discriminative clustering problem in itself can now be written as:

$$y^* = \operatorname*{arg\,min}_{y\in\mathcal{Y}} \operatorname{Tr}\left(y^TA(X,\mu)y\right). \tag{3.44}$$

As for the normalized cut problem, this is a quadratic problem under constraints which is NP-hard when y takes binary values. However, efficient convex relaxations can be used.

For instance, a standard technique is to reason on equivalence matrices $Y = yy^T$, perform a continuous relaxation, properly constrain the problem and solve the associated semidefinite programming problem. The resulting equivalence matrix can then be directly used to recover the cluster assignments. This approach allows incorporating prior knowledge over clusters in the form of must-link and must-not-link constraints.

Another possible technique is to perform a continuous relaxation on y, properly constrain the problem and use methods such as the Frank-Wolfe optimization algorithm [Frank and Wolfe, 1956; Jaggi, 2013] (which only relies on solving linear problems) to find a relaxed solution. This approach can handle prior knowledge directly on labels through constraints as well as must-link and must-not-link con-

straints. For instance, in an image foreground/background segmentation setup, it allows encoding different prior information for foreground and background explicitly.

Note that the quadratic cost functions has trivial solutions, which include the constant matrix (after the continuous relaxation) and the column-wise constant matrices. Proper conditioning is thus required to avoid a degenerate solutions.

In our work, a discriminative clustering term is included in the objective function of the multi-instance segmentation method described in Chapter 6. Given that the objects considered are of instances the same object class, we use this discriminative term to separate the pixels which belong to one of the instance from the rest of the image.

4 Disparity estimation, person detection and pose estimation in 3D movies

In this chapter, we first describe in Section 4.1 how we extract noisy disparity information from stereo videos. We present two new datasets we have collected from 3D movies for the tasks of person detection, pose estimation and multi person segmentation. In Section 4.3, we then describe how exploiting the additional disparity channel can help the performance of deformable part models for person detection and pose estimation. Last, in Section 4.4, we discuss how the relative ease of the person detection task in 3D movies can be exploited to automatically harvest person bounding boxes to efficiently train better person detection models for standard color images.

4.1 Disparity estimation

In this section, we explain how we acquired our stereoscopic data (i.e. pairs of videos for left and right eye views) and how we extracted depth from it. We then discuss and illustrate the quality of the produced depth maps.

4.1.1 Acquiring 3D data from stereoscopic movies

Extracting stereo pairs from BluRay movies. 3D BluRay disks are built as a combination of a main stream encoding the left eye view (which is also the fallback in case of a non-3D display), and an additional stream which stores the difference between left and right views, using both intra-view and inter-view cues to optimize compression, as specified in the H.264/MPEG4 MVC (which stands for Multi View Coding) standard [Marpe *et al.*, 2006]. We had to modify a standard decoder[1] to handle this and decode our streams, after successfully ripping (i.e. getting raw disk contents and decrypting them) and demultiplexing (i.e. separating video data from sound data or other metadata) them, as shown in Figure 4.1.

[1] `http://iphome.hhi.de/suehring/tml/`

Fig. 4.1: Pipeline used to extract stereo pairs from 3D Blu Ray disks. The first three steps, from Ripping to Demuxing, are handled by the MakeMKV software (`http://www.makemkv.com/`). The decoding is performed by a modified version of a reference MPEG4 decoder.

Computing disparity from stereo pairs

After acquiring stereoscopic pairs for each frame (Figure 4.2), we proceed with extracting depth information. Depth estimation from stereo images (or several multi-view images in general) is a standard task in computer vision, with many proposed algorithms and classical benchmarks[2]. Furthermore, there are other approaches aimed at video data which can jointly estimate the depth of the scene together with motion displacements – termed "Scene Flow" [Vedula *et al.*, 2005; Wedel *et al.*, 2008; Rabe *et al.*, 2010].

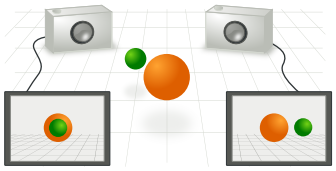


Fig. 4.2: Stereovision use-case[3].

However, in both cases cameras are assumed to be calibrated, or at least the stereo pairs to be rectified. Being calibrated means that we know which camera (or in our case, which pair of cameras) was used, with which sensor and lens. This translates into a set of parameters (called intrinsic parameters), such as focal length, pixel-to-distance scale or radial distortion parameters. In the case of stereo pairs, calibration might also include extrinsic parameters encoding the relative position of the two cameras. Knowing those parameters enables the use of epipolar geometry [Hartley and Zisserman, 2000], which provides a number of relations between 3D points and their 2D projections, as shown in Figure 4.3.

Epipolar geometry introduces the notion of epipoles (E_L and E_R in Figure 4.3), which are the intersection points between the planes of each image and the line between the optical centers of the cameras (O_L and O_R in Figure 4.3), and the notion of epipolar lines, which are the lines between a point of an image and the epipole of the same image. Epipolar geometry provides relationships between corresponding

[2] `http://vision.middlebury.edu/stereo/`
[3] `http://en.wikipedia.org/wiki/File:Aufnahme_mit_zwei_Kameras.svg`
[4] `http://en.wikipedia.org/wiki/File:Epipolar_Geometry1.svg`

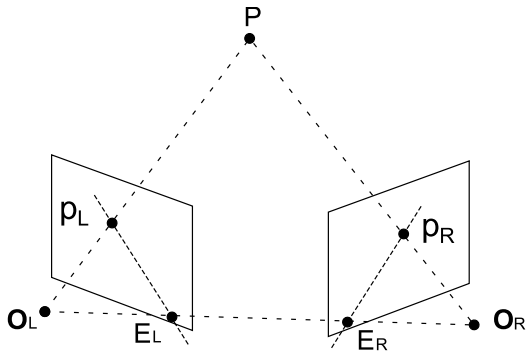


Fig. 4.3: Epipolar geometry[4].

epipolar lines in both images : given a point P_L in the left image, search for the corresponding point P_R in the right image can be reduced to the known epipolar line.

A stereo pair is said rectified when epipolar lines coincide and in practice are aligned with the horizontal axes of the two images, as shown in Figure 4.4. With such a stereo pair, the search for the matching point in the right image of a given point in the left image is much easier, since it is reduced to searching the point on the same horizontal line in the right image.

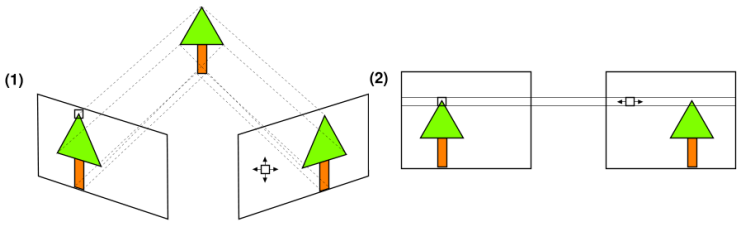


Fig. 4.4: Rectification makes epipolar lines coincide in stereo pairs[5].

This is particularly useful for stereovision, where one aims to recover the estimate of depth from a stereo pair. This can be done by evaluating the disparity of each point, which is the distance between a point in the left image and the matching point in the right image. This disparity is negatively correlated with the depth, since the larger the distance between a 3D point and the camera, the lower is the disparity value of the projected 3D point.

Unfortunately, in our attempts of working with stereo movies, we have found that standard methods of rectification and depth estimation did not produce reliable results. This might be attributed to the motion blur affecting reliable estimation of point correspondences as well as to possible geometry-violating post-production effects in movies. We have found a reliable rectification of movie frames to be important for depth-from-stereo algorithms since movie makers use dynamically

[5] http://en.wikipedia.org/wiki/File:Image_rectification.png

verged camera pairs to enhance the field of view or keep the focused object in the center of the field of view.

We have originally expected that the stereo pairs from the 3D movies would be rectified. Indeed, human vision makes use of binocular disparity just as standard stereovision algorithm to infer depth of the scene and distance to objects. Nevertheless, human eyes seem to be able to handle slightly unrectified images where the matching points are a little (i.e. by a few pixels) above or below the expected epipolar line, and we indeed found that our data was slightly unrectified. As expected, standard stereovision algorithms [Felzenszwalb and Huttenlocher, 2004b] fail when the rectification constraint is not fulfilled, as shown in Figure 4.5, producing more and more noisy results.

Fig. 4.5: Disparity maps from a rectified image pair (left) and unrectified image pairs with 1 to 4 pixels of vertical miss-alignment (four images on the right).

In our attempt to rectify the images, we used standard computer vision methods to match points in the two images (we computed SIFT (Scale-Invariant Feature Transform [Lowe, 1999]) features at Harris corners, matched them and removed outliers using RANSAC [Fischler and Bolles, 1981]) and tried to align those matches using homographies (i.e. projective transformations). Computing unconstrained homographies lead to a loss of information, since disparity would be almost suppressed by the alignment, and homographies constrained to vertical displacements did not provide satisfying results in a lot of cases, as shown in Figure 4.6.

Fig. 4.6: Left view and stereo result after tentative rectification.

To address the problem, we have tried several standard uncalibrated rectification and calibration packages [Fusiello *et al.*, 1999; Kukelova and Pajdla, 2007][6][7] (see Figure 4.7 for a sample).

Standard depth-from-stereo algorithms above rely on the strict geometric constraints which may not be satisfied in 3D movies e.g. due to the post-production effects. To address this, we have investigated a less constrained approach of dispar-

[6] `http://profs.sci.univr.it/~fusiello/demo/rect/`
[7] `http://cmp.felk.cvut.cz/minimal/8_pt_radial.php`

Fig. 4.7: Left view, ours and [Fusiello *et al.*, 1999] rectification results.

ity estimation based on the Optical Flow methods. Indeed, optical flow is commonly used to study the displacement of objects between two succeeding frames. By computing the dense optical flow between the two views and extracting the horizontal component of the displacement, dense disparity maps can be successfully estimated from the uncalibrated stereo pairs obtained from 3D movies. See Figure 4.8 and Figure 4.9 for results, using software from [Liu, 2009][8].

Fig. 4.8: Disparity maps computed using optical-flow on image examples from Figures 4.6-4.7.

4.1.2 Disparity maps quality

The quality of the disparity maps acquired using optical flow is quite satisfying, however the level of detail varies a lot from one shot to another one, as can be seen in Figure 4.9. In particular, the quality of the depth obtained from 3D movies may not be sufficient for existing approaches of human pose estimation from depth data [Grest *et al.*, 2005; Plagemann *et al.*, 2010; Shotton *et al.*, 2011], see for instance a sample Kinect depth map in Figure 4.10 (which appears of lower quality than it is due to visualization restrictions, but the disparity is almost continuous on the body).

Nevertheless, our estimated disparity maps provide quite clear object boundaries in many cases (see Figure 4.9), and the boundaries of body parts can often be visually detected, even for hard situations such as poses with crossed arms. Furthermore, disparity maps across a single shot are similar enough to be used without requiring any rescaling or normalization, which is a valuable property.

During the span of this work, we have tried multiple optical flow methods on the task of estimation horizontal disparity between unrectified stereo pairs from 3D

[8] `http://people.csail.mit.edu/celiu/OpticalFlow/`

Fig. 4.9: Left views and disparity maps computed using optical-flow.

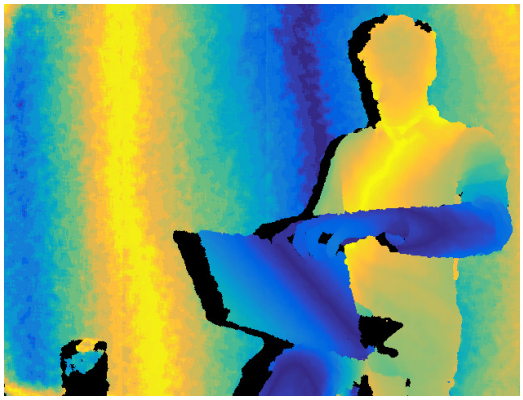


Fig. 4.10: Depth map provided by Microsoft Kinect. Fine depth information is produced by the sensor. Black areas correspond to occluded regions and occlusion boundaries.

movies. One difficulty of this study is that no ground truth disparity maps are available for this type of data, so no quantitative evaluation is possible. Visual inspection and relative performance analysis of the final tasks (e.g.person detection) are the only ways to assess the quality of the produced disparity maps. The first method we used is the one of [Liu, 2009] which solves a variational energy minimization problem in a coarse-to-fine scheme. This method runs in about 30 seconds on a standard single-core CPU for a pair of 960×540 pixel frames. The second method, from [Ayvaci *et al.*, 2012], jointly performs occlusion detection and optical flow estimation. This method runs in about 12 seconds for a pair of 960×540 pixel frames on a recent GPU (nVidia Tesla K20X). Recently a new method aimed at handling setups with large displacements and significant occlusions was proposed: EpicFlow [Revaud *et al.*, 2015]. This method first detects edges in the input before performing an edge-preserving dense matching between the two input frames. These matches are then used to initialize a standard variational energy minimization problem which solution produces the final dense optical flow map. This method runs relatively fast on a standard single-core CPU, taking about 10 seconds for a pair of 960×540 pixel frames.

A qualitative comparison between these three methods is available in Figure 4.11 and Figure 4.12. Visually, the results from [Ayvaci *et al.*, 2012] and [Revaud *et al.*, 2015] present less errors than the ones from [Liu, 2009], which tends to over-smooth the produced flow (we have tried to reduce this over-smoothing by playing with the hyperparameters of the method but could not find a better regularization trade-off). Compared to the results of [Revaud *et al.*, 2015], the results of [Ayvaci *et al.*, 2012] tend to appear flatter, discarding fine details in the disparity map. This is both a quality and a problem: it produces very clean layers in the image, but it loses valuable occlusion information.

In the rest of this work, we use the method of [Ayvaci *et al.*, 2012] to perform disparity estimation.

4.2 Datasets

We have collected and annotated two datasets from 3D movies to train and test our methods. The first dataset, the Inria 3DMovie Dataset (Section 4.2.1), contains annotated person bounding boxes and person poses in stereo movie frames as well as instance-level person segmentations in keyframes of 36 video clips. The second dataset, the Inria 3DMovie Dataset v2 (Section 4.2.2), focuses on instance-level person segmentation, with 27 annotated video clips presenting significant challenges.

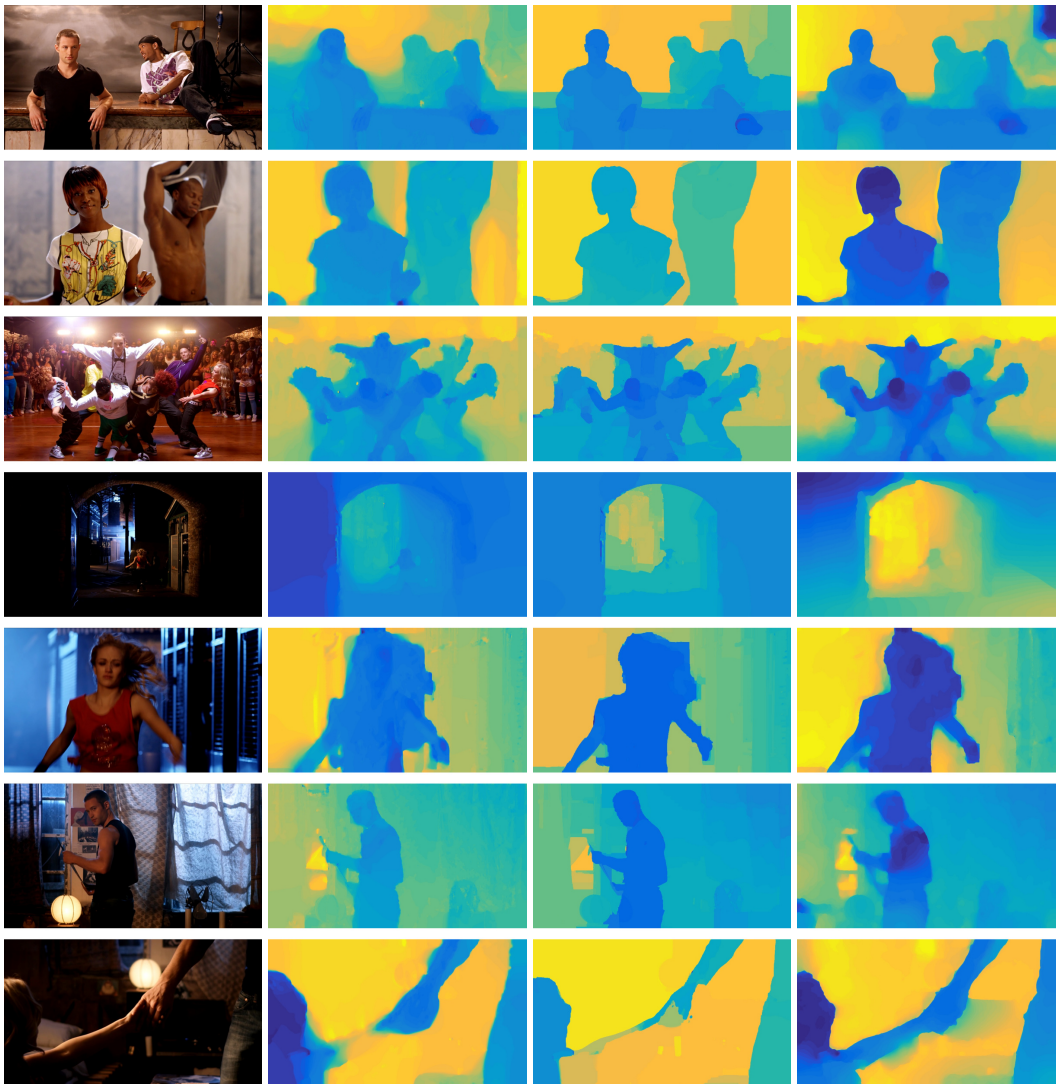
4.2.1 Inria 3DMovie Dataset

The Inria 3DMovie dataset is available on the project website[9]. Most of the frames in this dataset were obtained from the "StreetDance 3D" [Giwa and Pasquini, 2010] and "Pina" [Wenders, 2011] stereo movies. We chose these movies since they are filmed in true stereoscopic 3D, unlike others where 3D effects are added in post-production and result in inferior disparity estimation. Some of the stereo pairs used as negative examples for people-related tasks were harvested from Flickr and were originally shot with a Fuji W3 camera. The dataset includes stereo pairs (as jpegs), estimated disparity, (manually annotated) ground truth segmentations, poses and person bounding boxes.

The movie "StreetDance" was split into two parts (roughly in the middle), from which we selected the training and test frames, containing multiple people, respectively. The training set is composed of 520 annotated person bounding boxes, 438 annotated poses and 198 annotated segmentation masks from over 230 stereo pairs. Negative training data is extracted from 247 images with no people, taken from the training part of the movie, and from stereo pairs shot with a Fuji W3 camera.

The test set contains 36 stereo sequences (2727 frame pairs). For quantitative evaluation we provide 638 person bounding boxes and 149 pose annotations in 193 frames, among which a few do not contain any people. Given the cost of manually annotating pixel-wise segmentation, we provide this on a smaller set of 180 frames, containing 686 annotated person segmentations.

[9] `http://www.di.ens.fr/willow/research/stereoseg`



(a) Left frame **(b)** [Liu, 2009] **(c)** [Ayvaci *et al.*, 2012] **(d)** [Revaud *et al.*, 2015]

Fig. 4.11: Comparison between optical flow algorithms for the task of disparity estimation in stereoscopic movies.

(a) Left frame (b) [Liu, 2009] (c) [Ayvaci *et al.*, 2012] (d) [Revaud *et al.*, 2015]

Fig. 4.12: Comparison between optical flow algorithms for the task of disparity estimation in stereoscopic movies.

4.2.2 Inria 3DMovie Dataset v2

To evaluate the performance of our method on the task of instance-level video segmentation, we have collected a dataset composed of 27 video clips, corresponding to a total of 2476 frames. The video clips are taken from the 3D feature movie "StreetDance 3D" [Giwa and Pasquini, 2010]. The proposed dataset is an improved version of the *Inria 3D Movie Dataset* for the task of instance-level person segmentation, adding a substantial amount of challenges, such as longer shots, self-occlusions, inter-person occlusions, and hard poses such as dancing or jumping. Examples of frames and ground-truth annotations from our dataset are provided in Figure 4.13.

Providing ground-truth annotations for evaluation in an entire video is a highly time-consuming task. As a consequence, we have only annotated a sparse subset of 235 frames out of 2476, for all 632 person instances present in these frames. We split the dataset into a set of 7 clips for adjusting hyperparameters and a set of 20 clips for evaluation. Note that there is no training step in our method, but only a validation step to find appropriate hyperparameters, presented in Section 6.3.4.

Fig. 4.13: Examples of frames (a) and (c) and corresponding ground-truth head bounding boxes and pixel-wise segmentation labels (b) and (d). Note that the segmentation labels (indicated by color) are instance-level, i.e. we provide a segmentation mask for each individual person.

4.3 Person detection and pose estimation in 3D movies

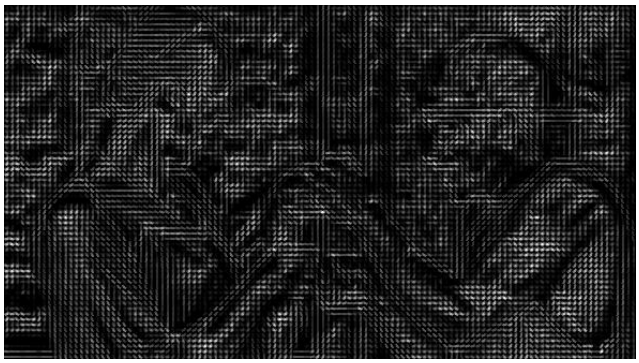
The deformable part models described in Section 3.1 for object detection and in Section 3.2 for pose estimation typically rely on feature maps built by computing histograms of oriented gradients (HOG features) over the input image converted to grayscale. To benefit from the stereo signal, we can train models jointly on appearance and disparity by concatenating appearance and disparity features into one representation. Indeed, the stereo signal that we extract as disparity maps exhibits properties which we expect to be beneficial for object detection and pose estimation. For instance, compared to color images, disparity maps are less textured and the object boundaries are often clearly visible. The feature maps computed on standard images can be easily extended to integrate features from disparity maps by computing HOG features at each location and concatenating them to the corresponding feature vector computed over the grayscale image. This is done by first converting the disparity map into a grayscale image by linearly mapping the disparity range to [0,1]. We then compute HOG on this grayscale image. Our HOG feature representation for disparity maps is similar to that used in [Spinello and Arras, 2011; Walk *et al.*, 2010] for person/pedestrian detection. The intuition is that HOG robustly captures the *changes* in the disparity rather than the actual disparity values, which can vary from scene to scene. Furthermore, compared to HOG features extracted on RGB, those extracted on the disparity map are usually less noisy and much sparser, since there are many constant areas in the disparity maps, as illustrated in Figure 4.14.

We evaluate the use of these features for person detection in Section 4.3.1 and for pose estimation in Section 4.3.2.

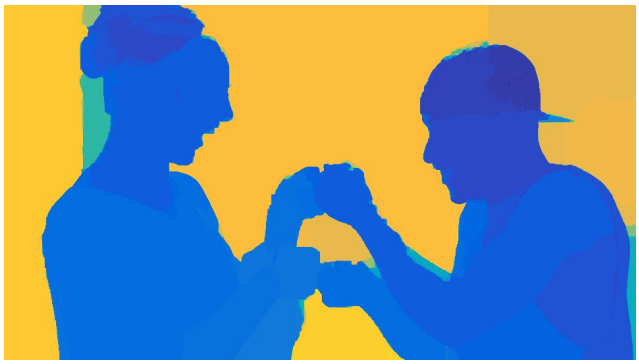
4.3.1 Person detection

In this section, we study the use of HOG features on color images and disparity maps in deformable part models (see Section 3.1) for upper body detection, using the Inria 3D Movie dataset. We use these features with the object detection method of [Felzenszwalb *et al.*, 2010] and with the person detection and pose estimation method of [Yang and Ramanan, 2011]. To produce person bounding boxes from the pose estimates produced by [Yang and Ramanan, 2011], we compute the bounding box of a subset of the estimated upper body joints: head, shoulders, elbows, hips. We have found this heuristic to perform well for the task of upper body detection.

(a) RGB image



(b) HOG features computed on the RGB image



(c) Disparity map

(d) HOG features computed on the disparity map

Fig. 4.14: Examples of HOG feature maps computed (b) on a RGB image (a) and (d) on the corresponding disparity map (c). Note that disparity maps are almost constant on object regions and highlight object boundaries. On the contrary, HOG feature maps for RGB input are sensitive to texture, which may confuse subsequent recognition.

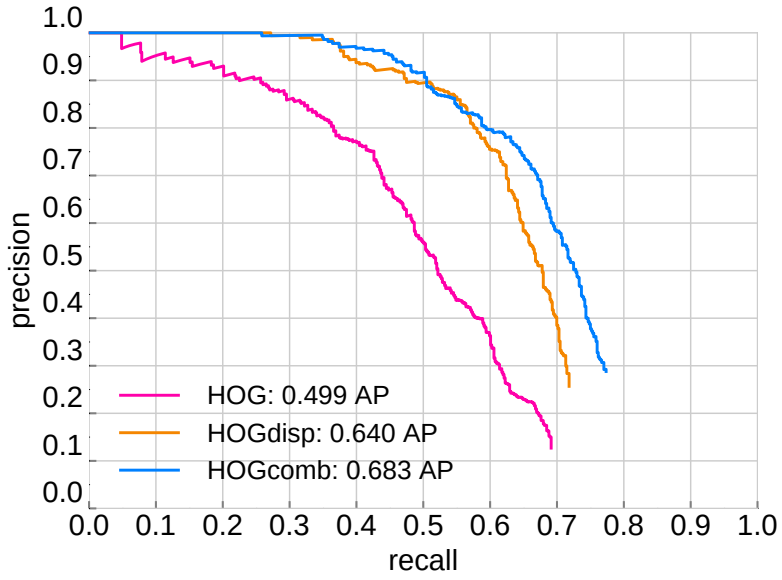
Using on the deformable part-based person detector [Felzenszwalb *et al.*, 2010], we have trained three variants using: (i) standard HOG extracted from grayscale images (*HOG*), (ii) HOG extracted from disparity maps (*HOGdisp*), and (iii) joint appearance and disparity features, using the concatenation of the two features (*HOGcomb*). We evaluated them on standard metrics from the PASCAL VOC development kit 2011 [Everingham *et al.*, 2011]. Precision-recall curves are shown in Figure 4.15 (a), with corresponding average precision (AP) values. It shows that the disparity-based detector (*HOGdisp*) improves over the appearance-based detector (*HOG*). Combining the two representations (*HOGcomb*) further increases person detection performance.

We show a qualitative comparison of these detectors in Figure 4.16. The appearance-based model *HOG* and the disparity-based model *HOGdisp* are often complementary, and the joint model *HOGcomb* outperforms both models, being able to cope well with the difficulties handled by each of the other two models. It appears that models using disparity cues are generally better at handling partially occluded persons or persons seen from the side or the back.

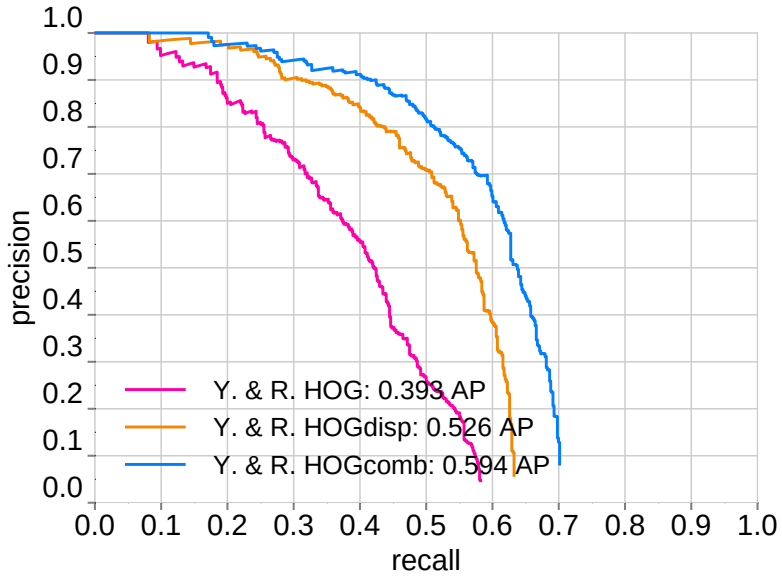
In addition, using the code from [Yang and Ramanan, 2011], we have trained pose estimators with annotated poses available for a subset of 438 person examples out of the 520 we used to train the detectors based on [Felzenszwalb *et al.*, 2010]. We have trained three models, one using appearance cues only (*Y. & R. HOG*), one using disparity cues only (*Y. & R. HOGdisp*) and one using jointly appearance and disparity cues (*Y. & R. HOGcomb*). We evaluate these models for person detection in Figure 4.15 (b) on the same test set as for the [Felzenszwalb *et al.*, 2010] detector above. As in Figure 4.15-(a), we observe considerable improvement provided by the disparity features and their combination with appearance features. However, we observe that performance is significantly lower than each of the corresponding person detector based on the model of [Felzenszwalb *et al.*, 2010]. This is likely due to [Yang and Ramanan, 2011] relying on accurate detection of all individual body parts (e.g., elbows, wrists, which are challenging to detect) to predict the location of the person, whereas [Felzenszwalb *et al.*, 2010] uses a more holistic person model. In other words, [Felzenszwalb *et al.*, 2010] is more robust to body parts being occluded or poorly detected.

4.3.2 Pose estimation

Pose estimation is typically evaluated using the percentage of correctly estimated body parts (PCP) score [Yang and Ramanan, 2011; Eichner *et al.*, 2012]. A body part is deemed to be correct if the two joints it links are within a given radius of their ground truth position, where the radius is a percentage of the ground truth

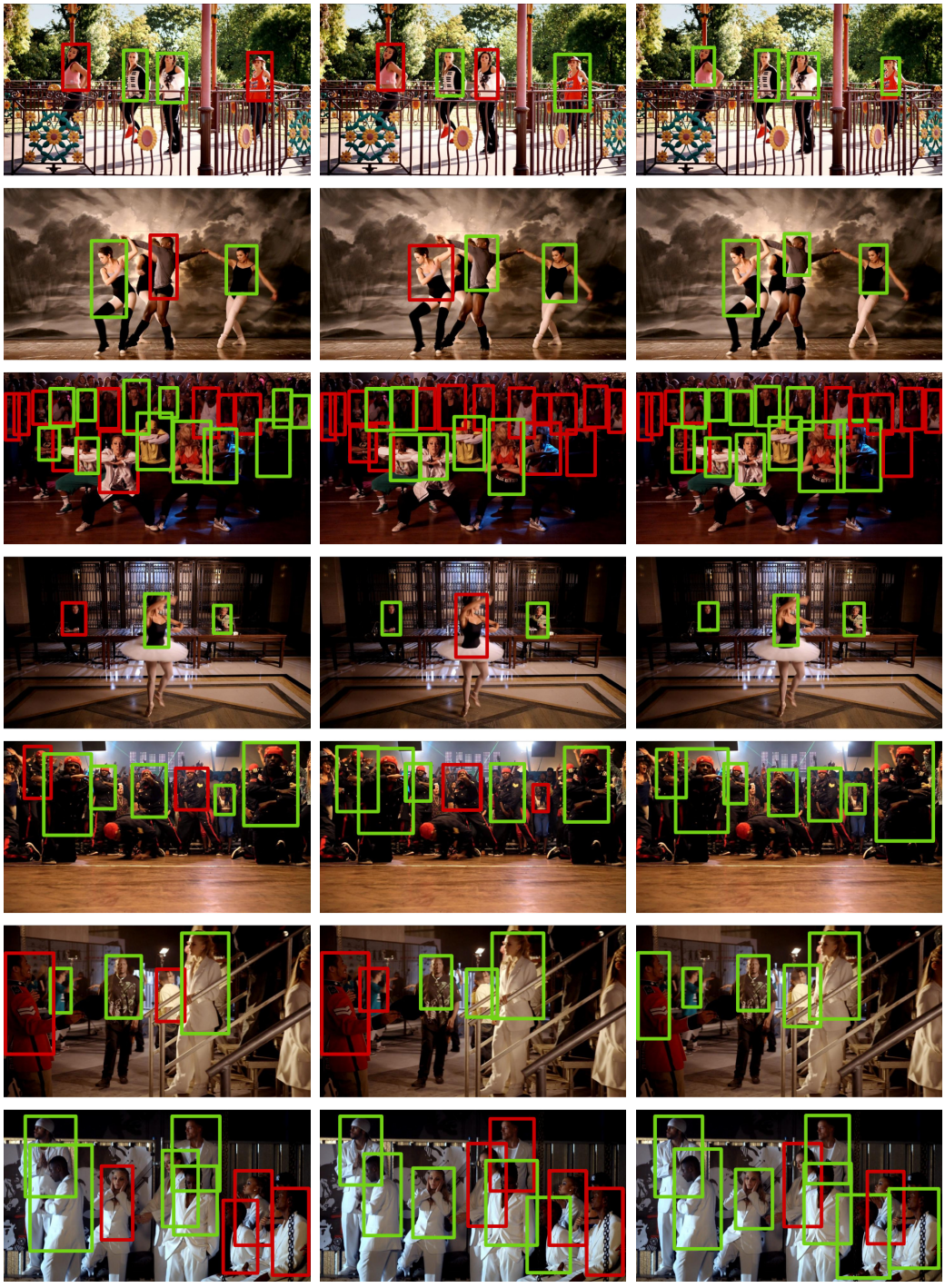


(a) Felzenszwalb *et al.* [Felzenszwalb *et al.*, 2010]



(b) Yang and Ramanan [Yang and Ramanan, 2011]

Fig. 4.15: *Precision-recall curves for person detection based on (a) Felzenszwalb* et al. *[Felzenszwalb* et al.*, 2010] and (b) Yang and Ramanan [Yang and Ramanan, 2011] methods. For both methods we report the performance of the appearance (*HOG*) and disparity (*HOGdisp*) based detectors, as well as the jointly trained appearance and disparity based detector (*HOGcomb*).* HOGcomb*, the detector based on [Felzenszwalb* et al.*, 2010] performs significantly better than the other models. **(Best viewed in colour.)***



(a) *HOG* (b) *HOGdisp* (c) *HOGcomb*

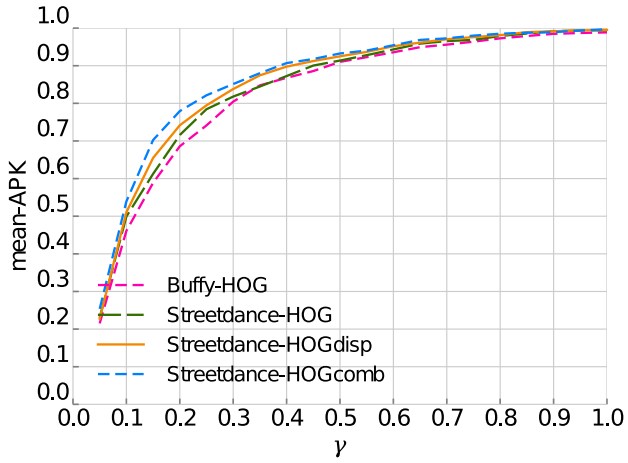
Fig. 4.16: Qualitative comparison between the appearance-based person detector *HOG*, the disparity-based detector *HOGdisp* and the joint appearance and disparity based detector *HOGdisp*. True positives are shown in green, and missed detections are shown in red. We operate in a mode where we have little to no false positives, so none are visible in these frames. See Section 4.3.1 for comments.

Tab. 4.1: *Evaluating pose estimation. We report global APK scores as well as scores for all five body parts, with $\gamma = 0.2$ as in [Yang and Ramanan, 2013]. We also evaluate the upper-body model from [Yang and Ramanan, 2011] trained on the Buffy dataset. The combination of appearance and disparity features (*HOGcomb*) outperforms the individual estimators (*HOG, HOGdisp*). Note that these scores are the average of the left and the right body parts, while those in Figure 4.17(b,c) show the scores for the left elbow and wrist only.*

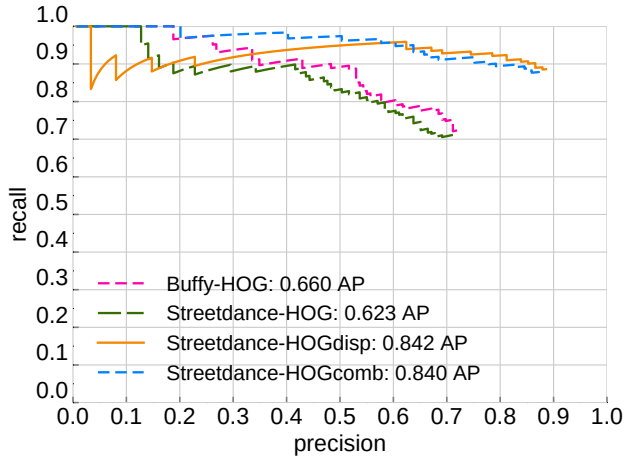
	[Yang and Ramanan, 2011]	HOG	HOGdisp	HOGcomb
Head	0.976	0.983	**0.993**	0.986
Shoulders	0.935	0.931	0.947	**0.969**
Elbows	0.658	0.665	0.759	**0.784**
Wrists	0.298	0.294	0.297	**0.400**
Hips	0.563	0.705	0.714	**0.757**
Average	0.686	0.716	0.742	**0.779**

length of the part. However, as argued in [Yang and Ramanan, 2013], a relaxed version of this definition has often been used in place of the original one, making it hard to compare published results. Furthermore, PCP requires matching the ground truth poses with the estimated ones, but there is no specification of how this matching should be done. Lastly, this measure uses the ground truth length of each part to set the radius within which the part is deemed to be correctly detected. This results in a foreshortening bias, where shorter limbs (which have a shorter radius) are penalized more severely than longer limbs. Instead of using PCP, we follow [Yang and Ramanan, 2013] and use their average precision of keypoints (APK) measure instead. In contrast to PCP, which evaluates the correctness of a part (connected to two joints/keypoints), APK measures the correctness of each keypoint. To overcome the foreshortening bias, the APK measure is based on the size of the ground truth person bounding box, rather than the individual parts. More precisely, a keypoint is considered to be correctly estimated if it lies within a radius given by the largest side of the ground truth pose bounding box, scaled by γ. Since the person detections are evaluated separately (Section 4.3.1), we use APK to only measure the pose estimation accuracy by considering the pose with the highest automatically obtained confidence score for each person detected.

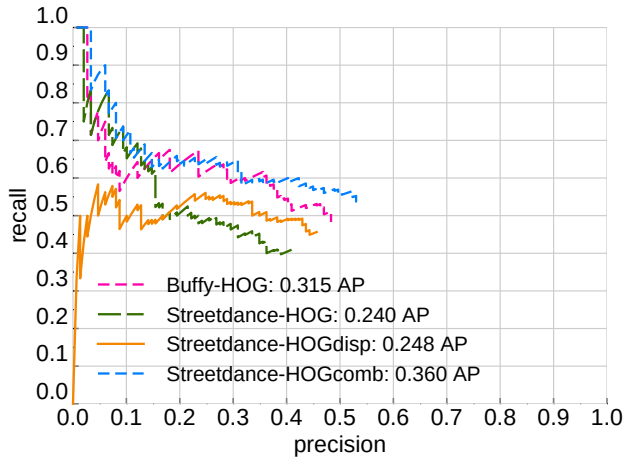
In Figure 4.17, we present mean APK curves, where we vary γ between 0 and 1, and plot APK curves for left elbow and left wrist for $\gamma = 0.2$, similar to [Yang and Ramanan, 2013]. The APK scores for all the parts are given in Table 4.1. The jointly trained pose estimator *Y. & R. HOGcomb* outperforms the individual estimators. We observe that the head and shoulder body parts are localized with high accuracy. Furthermore, combining appearance and disparity cues improves the localization of lower arms (elbows and wrists) by over 8%. We show a qualitative comparison of the three models in Figure 4.18. Visually, the joint model *Y. & R. HOGcomb* is



(a) Mean-APK



(b) Left elbow



(c) Left wrist

Fig. 4.17: *Pose estimation results.* Buffy-HOG *is the upper-body model from [Yang and Ramanan, 2011], and* Streetdance- *corresponds to our models trained on appearance or/and disparity features extracted from the 3D movie Streetdance. (a) Mean-APK curves, which are produced by varying the γ threshold. (b) & (c) Precision-recall curves for left elbow and left wrist respectively. Using disparity cues improves the recall of the pose estimator for elbows, and combining them with appearance cues shows a good initial precision performance. Estimating the wrist position remains a challenge, and the overall performance for this part is similar to [Yang and Ramanan, 2011].* (**Best viewed in colour.**)

(a) Input image (b) *Y. & R. HOG* (c) *Y. & R. HOGdisp* (d) *Y. & R. HOGcomb*

Fig. 4.18: Qualitative comparison between the appearance-based model *Y. & R. HOG*, the disparity-based model *Y. & R. HOGdisp* and the joint appearance and disparity based pose estimator *Y. & R. HOGdisp*. See Section 4.3.2 for comments.

able to handle a larger number of poses and situations, especially dark or crowded scenes as well as similar clothing and occlusions.

4.4 Depth-supervised training of person detection

Training object detectors typically require a large amount of training examples. This is especially true when considering Deep Neural Networks, which have the capacity to benefit from very large training datasets. However, bounding box annotations is a time consuming task, especially when consistency is required, and is thus expensive. In this section, we propose to leverage the fact that the detection task may be easier in other setups than standard color images. More specifically, we consider feature-length 3D movies as a source for large and diverse person bounding boxes examples. As shown in Section 4.3.1 and highlighted in Figure 4.19, the additional channel of information provided by 3D movies makes the person detection task substantially easier than in standard color images: the shape of the upper body of a person in terms of depth/disparity is fairly simple.

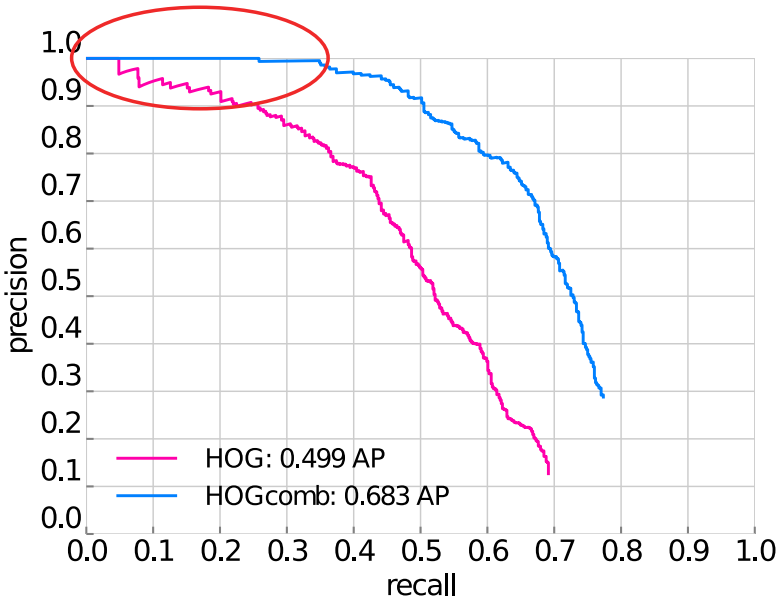


Fig. 4.19: Quantitative improvement of person detection performance when using disparity cues. By incorporating features computed on disparity maps (*HOGcomb*), we trained a model which presents a very interesting "high-precision" plateau, highlighted in red. This model, when applied on our dataset, produces perfect results up to 25% of recall. Comparatively, the model trained using only RGB information (*HOG*) only provides a very short high-precision plateau. This high-precision mode can be leveraged by applying the model to a very large amount of frames and selecting high-confidence detections.

In particular, the person detector trained using additional features computed on disparity maps presents a significant high-precision mode. This mode can be leveraged

to collect a large number of high-confidence examples. Indeed, a typical feature-length 3D movie contains on the order of 130.000 frames. Each movie is a potential source for thousands of person examples, and while the color appearance may be heavily different from the original movie, there is a high chance that the disparity appearance will be similar.

4.4.1 Automatic harvesting of hard positive samples

We develop a harvesting technique, illustrated in Figure 4.20, which only involves labelling a limited number of initial training examples from 3D movies.

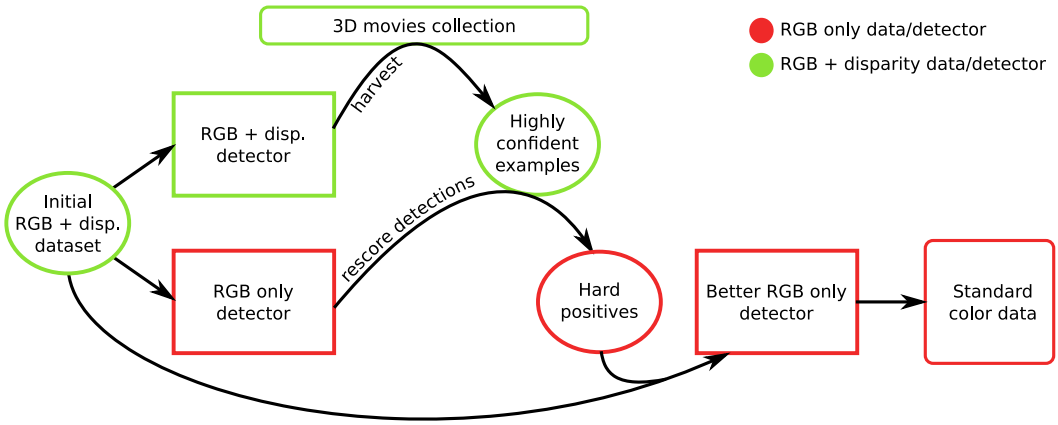


Fig. 4.20: Overview of the proposed method: from an initial training set, learn an appearance-based detector and a joint appearance and disparity-based detector. Use the joint detector to harvest many positive examples, select those which would currently be missed by the appearance-based detector, and train a better appearance-based detector by combining the initial training set with the harvested set of hard positives.

These examples are used to train two initial detectors: one over appearance channels only, and one jointly over appearance and disparity channels. In practice, we use the training examples from the Inria 3DMovie Dataset from Section 4.2.1 and the models trained in Section 4.3.1: the appearance-based model *HOG* and the joint model *HOGcomb*. The joint detector *HOGcomb* is used to harvest examples over multiple 3D movies operating in the low recall, high precision regime. The appearance-only detector *HOG* is then used to score the harvested examples, to select hard positives which were missed by the *HOG* detector and discard the ones which were already well detected by this detector. Due to this combination of detectors, one for selecting high-confidence examples using depth information and one for selecting hard examples when using appearance cues only, we call the entire method "depth-supervised training of person detection". We show examples of harvested examples in Figure 4.21. In total, we ran this harvesting procedure on 1 out of 6 frames over one movie and a half (the entire “Pina” movie and the half of the “StreetDance 3D” movie we have dedicated to training models), which sums to

more than 50000 frames. We collected a total of 118633 person detections using the *HOGcomb* detector, among which 39109 were deemed highly confident, having a detection score higher than 0.2. After filtering out examples which scored well already for the initial *HOG* detector (those having a detection score higher than 0 for this detector), we retained a total of 8902 automatically harvested bounding boxes.

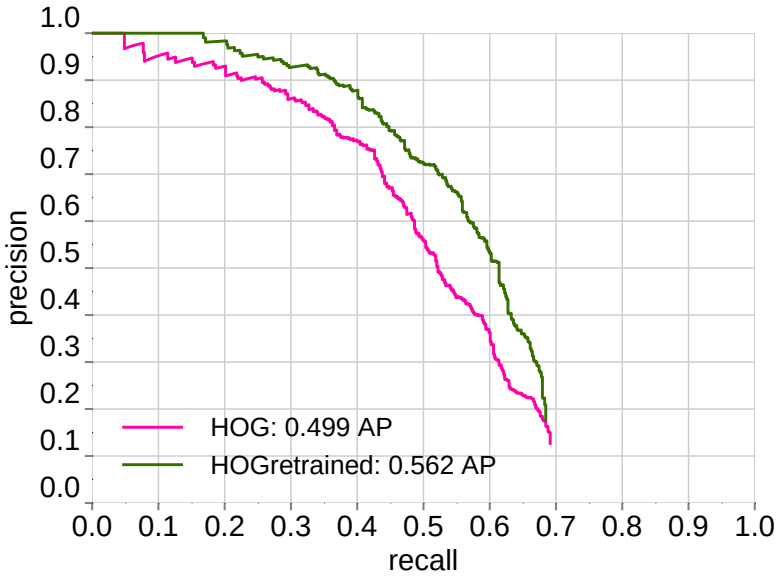
Using the initial training set combined with the newly harvested hard positives, we train a new appearance-based only detector, which we call *HOGretrained*. We evaluate this new detector on two datasets, and compare it with the original appearance-only detector *HOG*. In Figure 4.22, we show a quantitative comparison between the two models on (a) the Inria 3DMovie Dataset and (b) the Buffy dataset. The latter was extracted from the TV series Buffy the Vampire Slayer[10]. It is composed of 164 frames, out of which 79 frames contain positive examples, for a total of 101 annotated person bounding boxes. In both cases, the model trained with the additional harvested examples (*HOGretrained*) significantly outperforms the original model. The high-precision mode of the intermediate detector used to perform automatic harvesting on disparity data is partly transferred to the new models, which present significantly better high-precision plateaus. Last, we show in Figure 4.23 a qualitative comparison between the two detectors. The new detector *HOGretrained* is able to handle a larger variety of situations, such as partially occluded persons or persons seen from the side or from the back.

4.5 Discussion

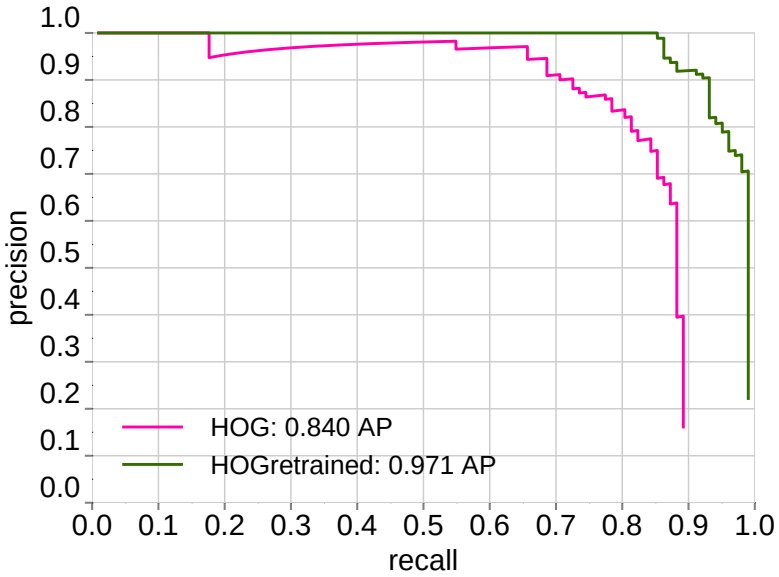
We have successfully extracted disparity maps from stereoscopic movies stored on BluRay 3D disks. We have collected two datasets which can be used to train and test methods for person detection, pose estimation and video segmentation in 3D movies. Using these datasets, we have adapted classical object detection and pose estimation methods based on the deformable part models framework to exploit the additional disparity maps provided by stereoscopic movies. We have further leveraged the relative ease of the person detection task in stereoscopic movies to use a joint appearance and disparity person detector to automatically collect person hard training samples. Using these additional examples, we have shown that the retrained appearance-only detector performs significantly better compared to the original one, trained on the same set of manually annotated samples.

[10] `http://www.robots.ox.ac.uk/~vgg/software/UpperBody/`

Fig. 4.21: Examples of automatically harvested person examples. We show side-by-side the cropped image and the cropped disparity map of each example. These examples were collected from stereoscopic movies. They were associated to a highly confident score by the joint *HOGcomb* detector, and a low score by the appearance-based *HOG* detector.

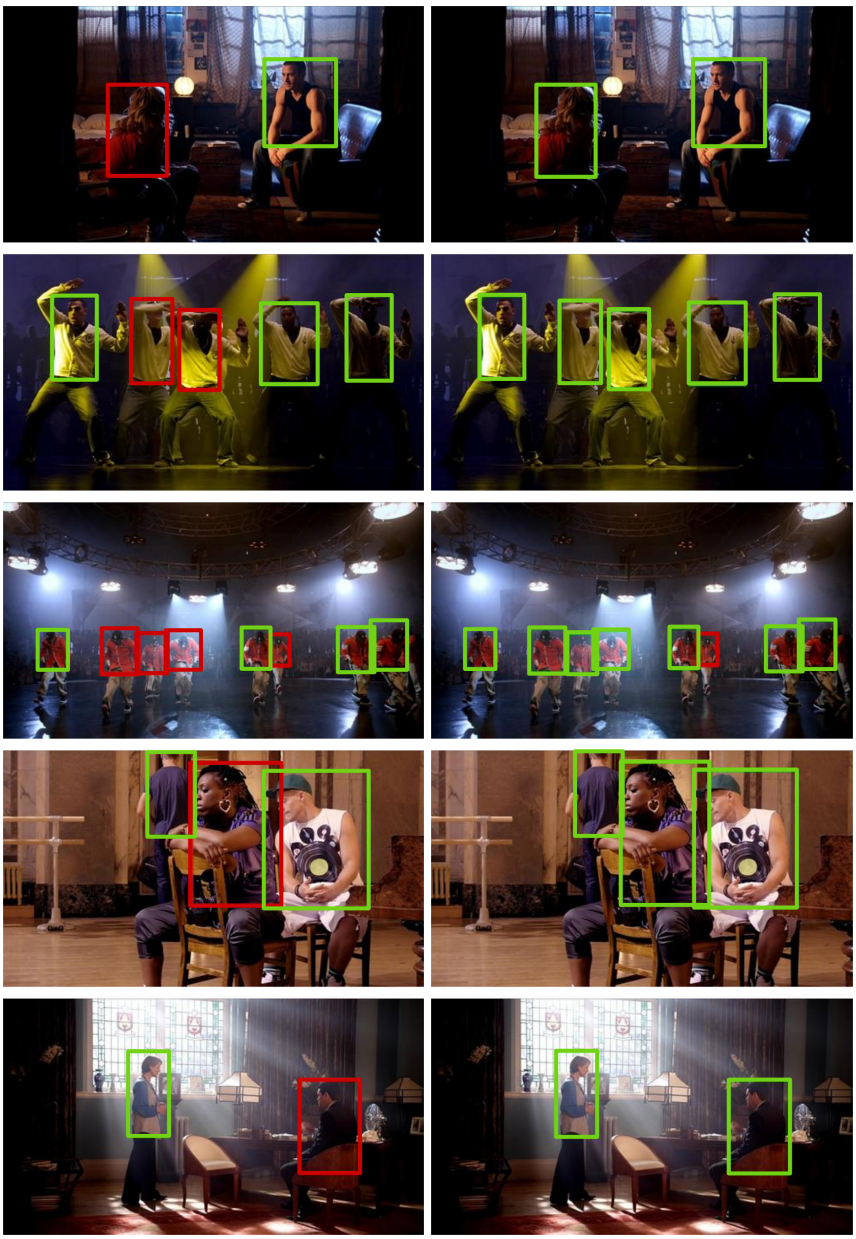


(a) Results on the Inria 3DMovie Dataset



(b) Results on the Buffy dataset

Fig. 4.22: Precision-recall curves for person detection based on Felzenszwalb *et al.* [Felzenszwalb *et al.*, 2010], using either a model trained on 520 annotated person bounding boxes from the Inria 3DMovie Dataset from Section 4.2.1 (*HOG*) or a model trained on these 520 manually annotated examples plus an additional 8902 automatically harvested examples (*HOGretrained*). We report results on the test set of the Inria 3DMovie Dataset (a) and on a test set from the Buffy dataset (b).



(a) *HOG* detector **(b)** *HOGretrained* detector

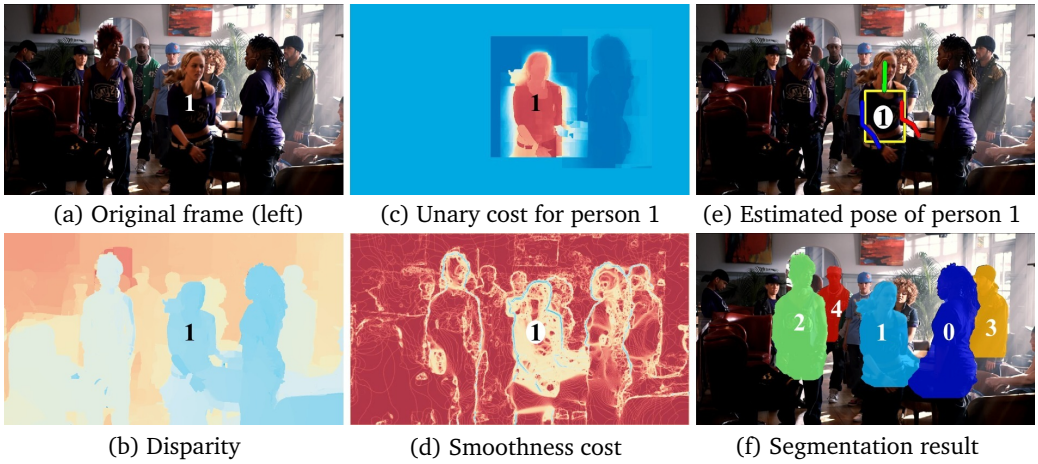
Fig. 4.23: Qualitative comparison between the original appearance-based *HOG* model (a) and the one retrained using additional depth-supervised harvested examples *HOGretrained* (b).

5 Multiple person segmentation with pose cues

In this chapter, we describe a method to obtain a pixel-wise segmentation of multiple people in stereoscopic videos. This task involves challenges such as dealing with unconstrained stereoscopic video, non-stationary cameras, and complex indoor and outdoor dynamic scenes with multiple people. We cast the problem as a discrete labelling task involving multiple person labels, devise a suitable cost function, and optimize it efficiently. We build on the improvements achieved in the Chapter 4 on person detection and pose estimation. We develop a segmentation model incorporating person detections and learned articulated pose segmentation masks, as well as colour, motion, and stereo disparity cues. The model also explicitly represents depth ordering and occlusion. We demonstrate results on the challenging Inria 3DMovie dataset from Section 4.2.1, as well as on the H2view dataset from [Sheasby *et al.*, 2012].

5.1 Introduction

Segmenting multiple people in a video is a task of great interest in computer vision. We explore this problem in the context of stereoscopic feature length movies, which provide a large amount of readily available video footage of challenging indoor and outdoor dynamic scenes. Our goal is to automatically analyze people in such challenging videos. In particular, we aim to produce a pixel-wise segmentation and recover the partial occlusions and relative depth ordering of people in each frame, as illustrated in Figure 5.1. Our motivation is three-fold. First and foremost, we wish to develop a mid-level representation of stereoscopic videos suitable for subsequent video understanding tasks such as recognition of actions and interactions of people [Yao and Fei-Fei, 2010]. Human behaviours are often distinguished only by subtle cues (e.g., a hand contact) and having a detailed and informative representation of the video signal is a useful initial step towards their interpretation. Second, disparity cues available from stereoscopic movies can improve results of person segmentation or person detection. Such results, in turn, can be used as a (noisy) supervisory signal for learning person segmentation in monocular videos or still images [Everingham *et al.*, 2011; Gulshan *et al.*, 2011; Niebles *et al.*, 2010; Yang *et al.*, 2011]. For instance, a single 90 minute feature length movie can pro-



(a) Original frame (left) (c) Unary cost for person 1 (e) Estimated pose of person 1

(b) Disparity (d) Smoothness cost (f) Segmentation result

Fig. 5.1: *Illustration of the steps of our proposed framework on a sample frame (a) from the movie "StreetDance". We compute the disparity map (b) from the stereo pair. Occlusion-aware unary costs based on disparity and articulated pose mask are computed for all the people detected in the scene. In (c) we show the unary cost for the person labelled* 1*. Pairwise smoothness costs computed from disparity, motion, and colour features are shown in (d). The range of values in (b,c,d) is denoted by the red (low) - blue (high) spectrum of colours. The estimated articulated pose for person 1 is shown in (e). (f) shows the final segmentation result, where each colour represents a unique person, and the numbers denote the front (0) to back (4) ordering of people.*

vide more than 150,000 pixel-wise segmented frames. Finally, segmentation of people will also support interactive annotation, editing, and navigation in stereo videos [Goldman *et al.*, 2008; Koppal *et al.*, 2011], which are important tasks in post-production and home video applications.

Given the recent success of analyzing people in range data from active sensors, such as Microsoft Kinect [Ren *et al.*, 2012; Shotton *et al.*, 2011], and a plethora of methods to estimate pixel-wise depth from stereo pairs [1], the task at hand may appear solved. However, depth estimates from stereo videos are much noisier than range data from active sensors, see Figure 5.1(b) for an example. Furthermore, we aim to solve sequences outside of the restricted "living-room" setup addressed by Kinect. In particular, our videos contain complex indoor and outdoor scenes with multiple people occluding each other, and are captured by a non-stationary camera.

Here, we develop a segmentation model in the context of stereoscopic videos, which addresses challenges such as: (i) handling non-stationary cameras, by incorporating explicit person detections and pose estimates; (ii) the presence of multiple people in complex indoor and outdoor scenarios, by incorporating articulated person-specific segmentation masks (Section 5.3) and explicitly modelling occlusion re-

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

lations among people; and finally (iii) the lack of accurate stereo estimates, by using other cues, such as colour and motion features. We cast the problem as a discrete labelling task involving multiple person labels, devise a suitable cost function (Section 5.2), and optimize it efficiently (Section 5.4). We evaluate the proposed model on the Inria 3DMovie dataset (Section 4.2.1) with challenging realistic dynamic scenes from two stereoscopic feature-length movies “StreetDance" [Giwa and Pasquini, 2010] and “Pina" [Wenders, 2011] (Section 5.5). Additionally, we present comparative evaluation of our method on the Humans in Two Views (H2view) dataset [Sheasby *et al.*, 2012].

5.2 Segmentation model

We aim to segment stereoscopic video sequences extracted from 3D movies into regions representing individual people. Figure 5.1 illustrates an overview of our method on a sample frame from a video. Here we consider a stereo pair (only the left image is shown in the figure), estimate the disparity for every pixel, and use it together with person detections, colour and motion features, and pose estimates, to segment individual people, as shown in Figure 5.1(f).

We initialize our model using automatically obtained person detections and assign every detection to a person, i.e., we assume a one-to-one mapping between people and detections. Each pixel i in the video takes a label from the set $\mathcal{L} = \{0, 1, \ldots, L\}$, where $\{0, 1, \ldots, L-1\}$ represents the set of person detections and the label L denotes the “background”.[2] We use the Conditional Random Field framework described in Section 3.3 with multiple labels. The cost of assigning a person (or background) label, from the set $\mathcal{L}$, to every pixel i, $E(y; \boldsymbol{\Theta}, \tau)$, is given by:

$$\begin{aligned} E(y; \boldsymbol{\Theta}, \tau) &= \sum_{i \in \mathcal{V}} \psi_i(y_i; \boldsymbol{\Theta}, \tau) + \sum_{(i,j) \in \mathcal{E}} \phi_{ij}(y_i, y_j) \\ &\quad + \sum_{(i,k) \in \mathcal{E}^t} \phi^t_{ij}(y_i, y_k), \end{aligned} \tag{5.1}$$

where $\mathcal{V} = \{1, 2, \ldots, N\}$ denotes the set of pixels in the video. The pairwise spatial and temporal neighbourhood relations among pixels are represented by the sets $\mathcal{E}$ and $\mathcal{E}^t$ respectively. The temporal neighbourhood relations are obtained from the motion field [Liu, 2009] computed for every frame. The unary potential $\psi_i(y_i; \boldsymbol{\Theta}, \tau)$ is the cost of a pixel i in $\mathcal{V}$ taking a label y_i in $\mathcal{L}$. It is characterized by pose parameters $\boldsymbol{\Theta} = \{\Theta^0, \Theta^1, \ldots, \Theta^{L-1}\}$ and disparity parameters $\tau = \{\tau^0, \tau^1, \ldots, \tau^{L-1}\}$, where Θ^l and τ^l represent the pose and disparity parameters for a person label l respectively. The disparity parameters determine the front-to-back ordering of people

[2] We refer to image regions that correspond to objects other than people as background.

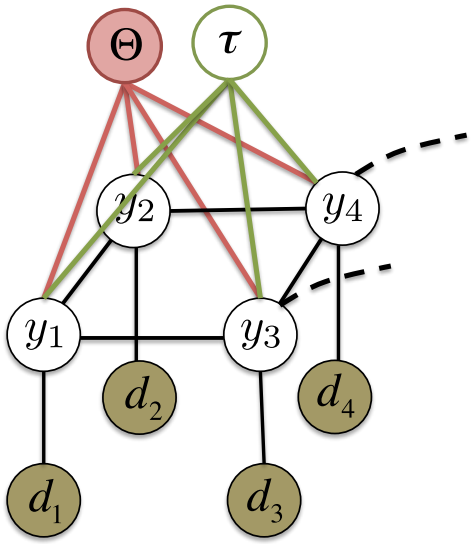


Fig. 5.2: *A graphical illustration of our model, where the observed variables are shaded. The variable d_i in the graph represents the features computed at each pixel i in the video. For clarity, we show 4 pixels from a frame, and 2 of the temporal links (dashed line), which connect pixels in one frame to the next. The person label y_i and disparity parameters τ are inferred given the image features d_i, and the pose parameters Θ.*

in the scene, as discussed in more detail in Section 5.4.1. Note that the pose and disparity parameters vary across time. However, for brevity, we drop this dependency on t in our notation.

The function $\phi_{ij}(y_i, y_j)$ is a spatial smoothness cost of assigning labels y_i and y_j to two neighbouring pixels i and j. Similarly, $\phi^t_{ij}(y_i, y_k)$ is a temporal smoothness cost. Given the parameters $\mathbf{\Theta}$ and τ, minimization of the cost (5.1) to obtain an optimal labelling $y^* = \arg\min_y E(y; \mathbf{\Theta}, \tau)$, results in segmentation of the video into regions corresponding to distinct people or background. However, in our problem, we also aim to optimize over the set of pose and disparity parameters. In other words, we address the problem of estimating y^*, the optimal segmentation labels, and $\mathbf{\Theta}^*$, τ^*, the optimal pose and disparity parameters as:

$$\{y^*, \mathbf{\Theta}^*, \tau^*\} = \arg\min_{y, \mathbf{\Theta}, \tau} E(y; \mathbf{\Theta}, \tau), \tag{5.2}$$

where $E(y; \mathbf{\Theta}, \tau)$ is the cost of label assignment y, given the pose and disparity parameters, as defined in (5.1). Given the difficulty of optimizing E over the joint parameter space, we simplify the problem and first estimate pose parameters $\mathbf{\Theta}$ independently of y and τ as described in Section 5.3. Given $\mathbf{\Theta}$, we then solve for y, τ as:

$$\{y^*, \tau^*\} = \arg\min_{y, \tau} E(y, \tau; \mathbf{\Theta}). \tag{5.3}$$

Further details are provided in Section 5.4. A graphical representation of our model is shown in Figure 5.2. The remainder of this section defines the unary costs, which are computed independently in every frame, and the spatio-temporal pairwise costs in energy (5.1).

5.2.1 Occlusion-based unary costs

Each pixel i takes one of the person or background labels from the label set $\mathcal{L}$. Building on the approach of [Yang *et al.*, 2011], we define occlusion-based costs corresponding to these labels, $\psi_i(y_i = l; \mathbf{\Theta}, \tau), l$ in $\mathcal{L}$, as a function of likelihoods β^l, computed for each label l, as follows:

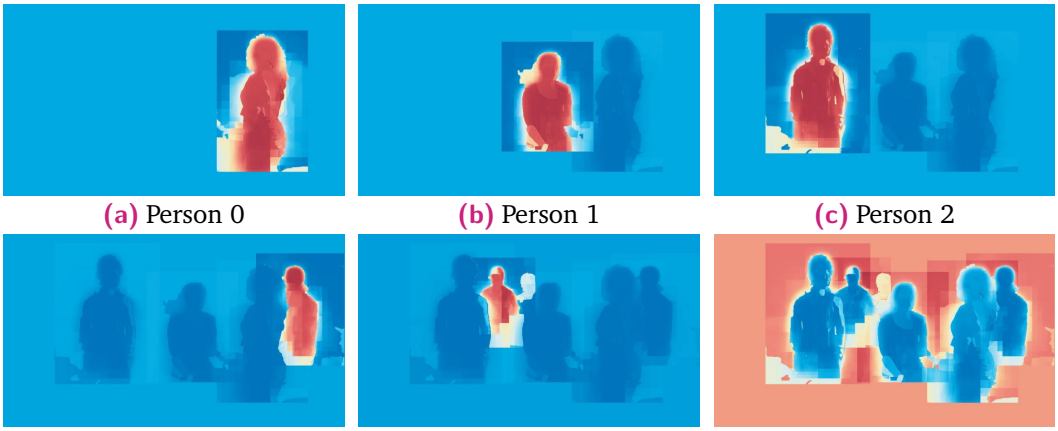
$$\psi_i(y_i = l; \mathbf{\Theta}, \tau) = -\log P(y_i = l|\mathbf{\Theta}, \tau), \tag{5.4}$$

$$\text{where} \quad P(y_i = l|\mathbf{\Theta}, \tau) = \beta_i^l \prod_{\{m|\tau^m > \tau^l\}} (1 - \beta_i^m). \tag{5.5}$$

Here, β_i^l is the likelihood of pixel i taking the person (or background) label l. Note that β_i^l's do not sum to one over the label set for any given pixel. The label likelihood over the entire image β^l is then formed by composing the likelihoods β_i^l, for all pixels $i \in \mathcal{V}$ in the image. In essence, β^l is a soft mask, which captures the likelihood for one person detection. It can be computed using the pose estimate of the person, and image features such as disparity, colour, and motion, as discussed in the following section. To account for the fact that the people in a scene may be occluding each other, we accumulate the label likelihoods in a front-to-back order as in Equation (5.5). This order is determined by the disparity parameters τ we estimate (see Section 5.4). In other words, to compute the cost of a pixel taking a person label i, we consider all the other person labels that satisfy $\tau^m > \tau^i$, i.e., are in front of person i. This makes sure that pixel i is likely to take label l, if it has sufficiently strong evidence for label l (i.e., β_i^l is high), and also has low evidence for other labels m, which correspond to people in front of person l (i.e., β_i^m is low for all labels with $\tau^m > \tau^l$). Figure 5.3 shows an illustration of these costs on an example.

5.2.2 Label likelihood β^l

Given a person detection and its corresponding pose estimate Θ^l, the problem of computing the label likelihood β^l can be viewed as that of segmenting an image into person *vs*. background. Note that we do not make a binary decision of assigning pixels to either the person or the background label. This computation is more akin



(a) Person 0 (b) Person 1 (c) Person 2

(d) Person 3 (e) Person 4 (f) Background

Fig. 5.3: *Illustration of the occlusion-based unary costs for the example in Figure 5.1. From left to right we show the unary costs for persons labelled 0 – 4 and the background. The cost for a pixel to take a label (person or background) is denoted by the red (low) - blue (high) spectrum of colours. Here we observe the effect of accumulating the label likelihoods in a front-to-back order. For example, in the illustration for Person 4, a low cost (red) for taking label 4 is observed only for the pixels that are not occluded by the other people in front.*

to generating a *soft* likelihood map for each pixel taking a particular person label. We define this using disparity and pose cues as:

$$\beta_i^l = (1 - \alpha^l)\ \psi_p(\Theta^l) + \alpha^l\ \psi_d(\tau^l), \tag{5.6}$$

where $\psi_p(\Theta^l)$ is an articulated pose mask described in Section 5.3, $\psi_d(\tau^l)$ is a disparity likelihood, and α^l is a mixing parameter that controls the relative influence of pose and disparity. The disparity potential is given by:

$$\psi_d(d_i; \tau^l, \sigma^l) = \exp\left(-\frac{(d_i - \tau^l)^2}{2(\sigma^l)^2}\right), \tag{5.7}$$

where d_i is the disparity value computed at pixel i. The disparity potential is a Gaussian characterized by mean τ^l and standard deviation σ^l, which together with the pose parameter Θ^l determines the model for person l. We set $\beta_i^L = 0.9$ for all the pixels for the background label L. The method for estimating the parameters τ^l and σ^l for person labels (i.e., $l \neq L$) is detailed in Section 5.4.

5.2.3 Smoothness cost

In some cases, the disparity cue used for computing the unary costs may not be very strong or may "leak" into the background (see examples in Figure 5.10). We introduce colour and motion features into the cost function (5.1), as part of the smoothness cost, to alleviate such issues. The smoothness cost, $\phi_{ij}(y_i, y_j)$, of as-

signing labels y_i and y_j to two neighbouring pixels i and j takes the form of a generalized Potts model [Boykov and Jolly, 2001] given by:

$$\phi_{ij}(y_i, y_j) = \begin{cases} \lambda\,(\lambda_1 \exp(\frac{-(d_i-d_j)^2}{2\sigma_c^2}) + \lambda_2 \exp(\frac{-||\mathbf{v}_i-\mathbf{v}_j||_2^2}{2\sigma_v^2}) \\ +\lambda_3 \exp(\frac{-(pb_i-pb_j)^2}{2\sigma_p^2})) \qquad \text{if } y_i \neq y_j, \\ 0 \qquad\qquad\qquad\qquad \text{otherwise,} \end{cases} \tag{5.8}$$

where λ, λ_1, λ_2, λ_3, σ_c, σ_v and σ_p are parameters of the model. The function $(d_i - d_j)^2$ measures the difference in disparity between pixels i and j. The motion vector at pixel i is denoted by $\mathbf{v}_i \in \mathbb{R}^2$, and $||\mathbf{v}_i - \mathbf{v}_j||_2$ is the norm of the motion vector difference of pixels i and j. The function $(pb_i - pb_j)^2$ measures the difference of colour features (Pb feature values [Arbelaez *et al.*, 2011]) of pixels i and j. The temporal smoothness cost $\phi_{ij}^t(y_i, y_k)$ is simply a difference of Pb features values for two pixels i and k connected temporally by the motion vector $\mathbf{v}_i$.

Thus far we have discussed the model given person detections, their pose and disparity parameters. In what follows, we will describe our method for detecting people, their poses, and the likelihood computed from them (Section 5.3). We then provide details of the inference scheme for determining the disparity parameters and the pixel-wise segmentation (Section 5.4).

5.3 Estimating an Articulated Pose Mask

The aim here is to obtain an articulated pose segmentation mask for each person in the image, which can act as a strong cue to guide the pixel-wise labelling. We wish to capture the articulation of the human pose as well as the likely shape and width of the individual limbs, torso, and head in the specific pose. We build here on the state-of-the-art pose estimator of Yang and Ramanan [Yang and Ramanan, 2011], and extend it in the following two directions. First, we incorporate disparity as input to take advantage of the available stereo signal. Second, we augment the output to provide an articulated pose-specific soft-segmentation mask learned from manually annotated training data.

5.3.1 Person detection and tracking

We obtain candidate bounding boxes of individual people and track them throughout the video. Detections are obtained from the deformable part-based person detector *HOGcomb* from Section 4.3.1. We apply this joint appearance and disparity based detector to each frame in the video sequence independently. We also compute point tracks, which start at a frame and continue until some later frame, over

the entire sequence with the Kanade-Lucas-Tomasi tracker [Shi and Tomasi, 1994]. Point tracks that lie within each detection result are used to fill-in any missing detections by interpolating the location of the bounding box and also to smooth the detections temporally [Everingham *et al.*, 2006].

5.3.2 Pose estimation from appearance and disparity

We estimate the pose of the person within each person detection bounding box. We restrict our pose estimation models to upper body poses, which are more commonly found in movie data. Again, to benefit from the stereo video, we extract both appearance and disparity features in the frame (in contrast to [Yang and Ramanan, 2011; Desai and Ramanan, 2012], which use appearance features only). The advantage is that some edges that are barely visible in the image, e.g., between people in similar clothing, can be more pronounced in the disparity map. We use HOG features for both appearance and disparity, as described above for person detection. We introduce specific mixtures for handling occlusion, as in [Desai and Ramanan, 2012], into the pose estimation framework of [Yang and Ramanan, 2011].

In this framework, the model is represented as a set of K parts, where a part refers to a patch centered on a joint or on an interpolated point on a line connecting two joints. For example, we have one part for an elbow, one for a wrist, and two parts between the elbow and the wrist, spread uniformly along the arm length. We use a model with 18 parts. The set of parts includes 10 annotated joints, *head, neck, 2 shoulders, 2 elbows, 2 wrists, 2 hips*, together with 8 interpolated parts. Further, each part is characterized by a set of mixtures. The mixture components for an elbow part, for example, can be interpreted as capturing different appearances of the elbow as the pose varies, including occlusions by other limbs or people, that are explicitly labelled in the training data. We learn up to eight mixture components, among which one or two are dedicated to handle occlusions, for each part. We refer the reader to [Yang and Ramanan, 2011] for more details on the training procedure.

5.3.3 Articulated pose mask ψ_p

The output of the pose estimator is the location of the individual parts in the frame as shown in Figure 5.5(a). To obtain a pose-specific mask we learn an average mask for each mixture component for each part. This is achieved by applying the trained pose-estimator on a training set of people with manually provided pixel-wise segmentations. All training masks, where mixture component c of part k is detected, are then rescaled to a canonical size and averaged together to obtain the mean

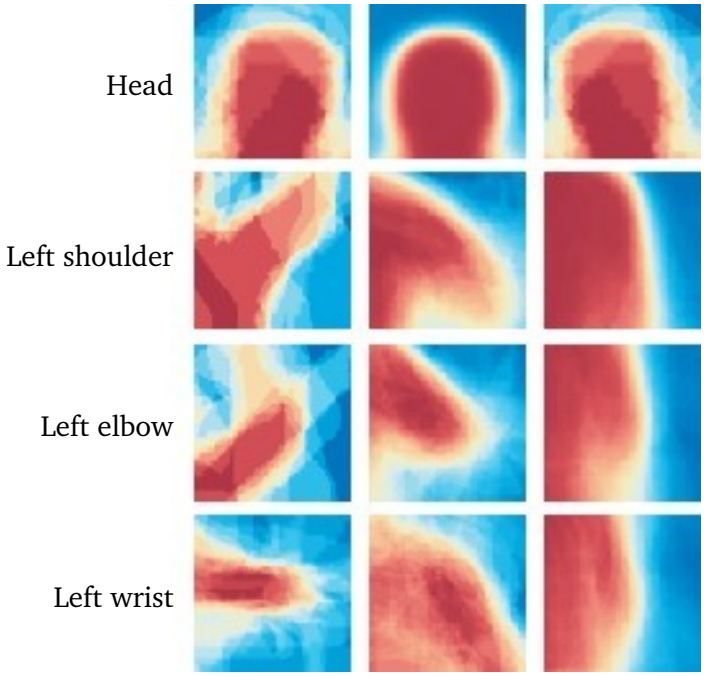


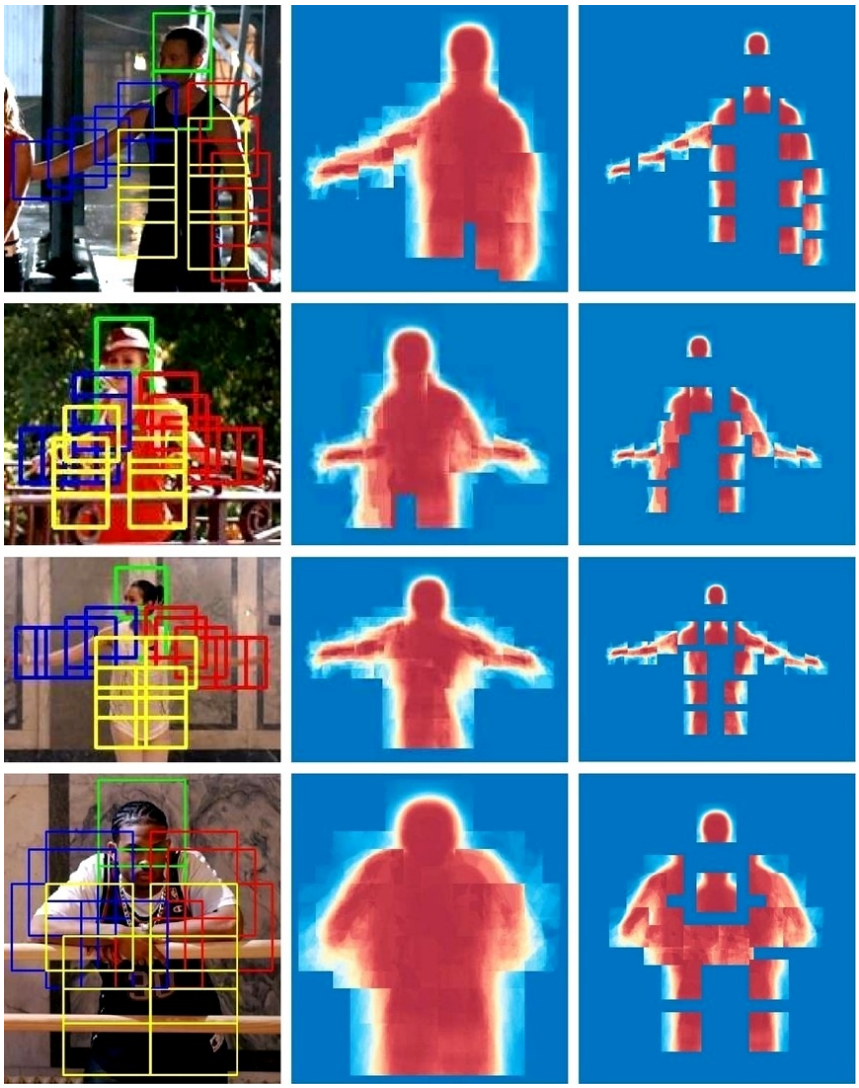
Fig. 5.4: *Articulated pose masks for three mixture components are shown for some of the body parts. The pose masks for each part capture a different configuration of the pose. For instance, the masks for "Left wrist" show three different locations of the lower arm: stretched out, partially bent over the shoulder, and lying by the torso.*

mask $m_{kc}(i)$. The value at pixel i in the mean mask counts the relative frequency that this pixel belongs to the person. An illustration of masks for individual parts and mixture components is shown in Figure 5.4.

At test time, given an estimated pose with an instantiated mixture component c^* for a part k, the likelihood for the person, $\psi_p(\Theta, i)$ at pixel i, is obtained by laying out and composing the articulated masks m_{kc^*} for all the parts. If, at pixel i, multiple masks overlap, we take the maximum as $\psi_p(\Theta, i) = \max_k m_{kc^*}(i)$. We found that taking the max was beneficial for person segmentation targeted in this chapter as it suppresses internal edges between body parts, such as a hand positioned in front of the torso. An illustration of the articulated pose masks for various examples is shown in Figure 5.5. Note how the part masks can capture fine variations in the shape and style of the pose.

5.4 Inference

In the previous section we have outlined how we compute the pose parameters Θ^l and the corresponding articulated pose mask for each person l. Poses are estimated independently for each person and fixed throughout the rest of the inference procedure described next. The aim is to compute the optimal disparity parameters τ^* and



(a) Estimated pose (b) Pose mask (c) Per-mixture masks

Fig. 5.5: *Estimated poses and masks on sample frames. Given a pose estimate (a), we compute a pose-specific mask (b) using per-mixture part masks learned from manually segmented training data. In (c) we show a scaled version of the masks, doubling the actual distances between part masks. This visually explains how each per-mixture mask is contributing to the final mask. In (b,c), the cost for a pixel to take a person label is denoted by the red (low) - blue (high) spectrum of colours.*

pixel labels $\mathbf{x}^*$ given the pose parameters Θ, as described by the minimization problem (5.3). It is well known that minimizing multi-label functions such as $E(y; \mathbf{\Theta}, \tau)$, which corresponds to the segmentation problem, given the pose and disparity parameters, is in itself NP-hard (for the type of smoothness cost we use) [Boros and Hammer, 2002]. The additional complexity of optimizing over disparity parameters τ further adds to the challenge. Methods like [Isack and Boykov, 2012] explore joint optimization solutions for such problems. Here, we propose a two-step strategy, where we first: (i) estimate the optimal disparity parameters τ^* using an approximation to (5.3), without the pairwise terms; and then (ii) obtain the pixel labels $\mathbf{x}^*$ with the estimated (and now fixed) parameters τ^* by minimizing the full cost (5.1). These two steps are detailed below.

5.4.1 Obtaining disparity parameters

The estimation of the set of disparity parameters τ for all the people in a frame can be succinctly written as:

$$\tau^* = \arg\min_{\{\tau\}} \tilde{E}(\tilde{y}; \mathbf{\Theta}, \tau), \tag{5.9}$$

where we approximate the original cost function (5.1) by only using unary and ignoring the pairwise terms[3] as $\tilde{E}(y; \mathbf{\Theta}, \tau) = \sum_{i \in \mathcal{V}} \psi_i(y_i; \mathbf{\Theta}, \tau)$. Note that for this modified cost function, the optimal pixel labelling $\tilde{y}$ for a given τ can be obtained independently for each pixel as $\tilde{y}_i = \arg\min_{m \in \mathcal{L}} \tilde{E}(y_i = m, \mathbf{\Theta}, \tau)$. Further, the disparity parameter τ is inversely related to depth, and determines the front-to-back order of people in a frame. Thus, this minimization problem (5.9) explores various combinations of the relative order of people in a frame by optimizing over $\{\tau\}$. The set of possible disparity parameter values for each person can still be large, and exploring the exponentially many combinations for all the people in the frame may not be feasible. To address this issue, we obtain and optimize over a small set of (up to 3) candidates $\{\tau^l\}$, for each person l. Using a thresholded pose mask, we compute mean disparity μ^l of all the pixels within, and set $\{\tau^l\} = \{\mu^l, \mu^l \pm \sigma^l\}$. The parameter σ^l is set according to a linear decreasing function of μ^l. Note that the disparity parameters are estimated jointly for all the people in the scene. We illustrate this on a sample image in Figure 5.6.

5.4.2 Person segmentation

With the estimated disparity (and pose) parameters, we compute the unary and smoothness costs, and use the efficient α-expansion algorithm [Boykov *et al.*, 2001]

[3] We note that this is a reasonable approximation, as τ only directly affects the unary cost ψ_i in (5.1).

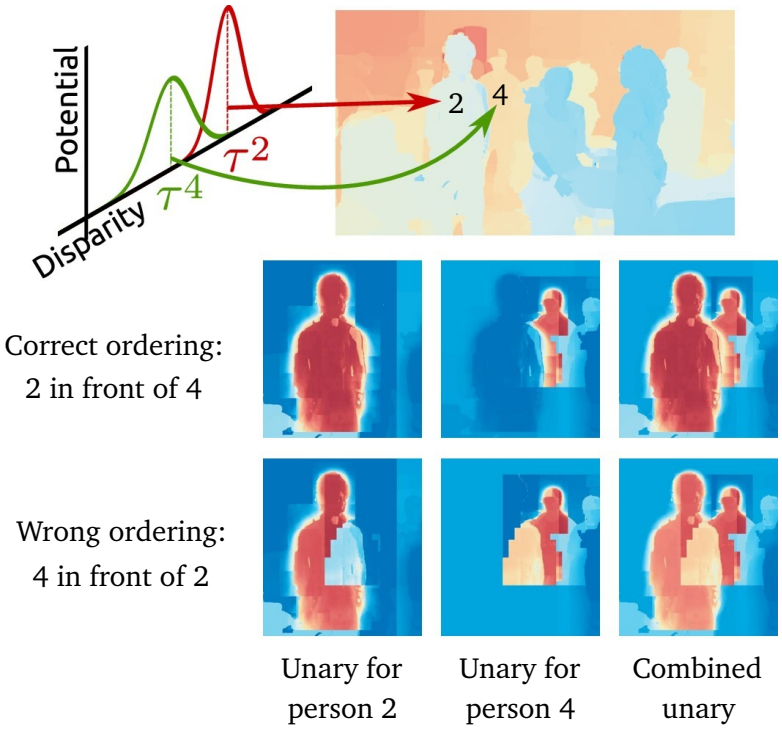


Fig. 5.6: *The front-to-back ordering of people in a scene is determined by τ^l, the disparity parameter in the potential (5.7), estimated for each person (shown at the top). The optimal set τ^* is estimated jointly for all the people by solving (5.9) over a candidate set. Here we show the effect of picking wrong τ^l for two people, which implies wrong ordering (shown at the bottom). This results in poor unary cost functions and a higher overall cost, due to the additional negative evidence in the form of $(1-\beta_i^m)$ as defined in (5.5). The colours red, yellow and blue in the unary cost figures represent low, medium and high costs respectively. Unaries (here for persons 2 and 4) are combined (third column) by taking their per-pixel minimum, as described in Section 5.4.1. Note the lower cost (more red) of the combined unary for the correct person ordering.*

to optimize (5.1). This assigns every pixel a person or background label from the set $\mathcal{L}$.

5.5 Experiments

In this section, we first report results for layered segmentation in Section 5.5.1. Next, in Section 5.5.2, we investigate the sensitivity of our algorithm to its main parameters and in Section 5.5.3, we analyze the robustness of our approach by replacing its components with ground truth results. Finally, in Section 5.5.4 we evaluate the segmentation accuracy of our method on the H2view dataset [Sheasby *et al.*, 2012].

5.5.1 Segmenting multiple people

In our experiments we used the following parameter values: $\lambda = 1.0$, $\lambda_1 = 6.3$, $\lambda_2 = 6$, $\lambda_3 = 2.7$, $\sigma_c^2 = 0.025$, $\sigma_v^2 = 0.01$, $\sigma_p^2 = 0.025$, which were set empirically, and fixed for the evaluation. A quantitative evaluation of the segmentation model using ground truth annotations is shown in Table 5.1. In this evaluation we compare three variants of our approach and two baseline methods. The first one ("No mask, single frame") refers to the case where the label likelihood $\beta_i^l = \psi_d$, i.e., there is no influence of pose on the segmentation. In other words, this method uses disparity features, but not the pose information. The second method ("Uni mask, single frame") incorporates a person location likelihood, which is computed by averaging ground truth segmentations of people from the training data (after rescaling them to a standard size) into a single non-articulated "universal" person mask – an approach inspired by the successful use of such masks in the past [Yang *et al.*, 2011]. We use this as the *person* likelihood ψ_p, and combine it with disparity likelihood ψ_d, as explained in Section 5.2. The third variant ("Pose mask, single frame") incorporates the articulated pose mask, described in Section 5.3. Our complete model ("Proposed") introduces temporal smoothness across frames.

For the "Colour only" baseline, we used a colour-based model for the unary costs without the disparity potential. These costs were computed from colour histograms for each label [Boykov and Jolly, 2001]. In other words, each label is associated with a histogram computed from a region in the image, and the unary cost of a pixel is a function of the likelihood of the pixel, given its colour, taking this label. The success of this model certainly depends on the regions used for computing the histograms. We used the result obtained by segmenting in the disparity space, i.e., "No mask, single frame", as these regions. We believe that this provides a reasonable estimate for the label potentials. The background histogram was computed with

(a) Original image (b) Segmentation result

Fig. 5.7: *Qualitative results on images from the movies "StreetDance" and "Pina". Each row shows the original image and the corresponding segmentation. Rows 1 and 2 demonstrate successful handling of occlusion between several people. The method can also handle non-trivial poses, as shown by Rows 3 and 4. The segmentation results are generally accurate, although some inaccuracies still remain on very difficult examples. For instance, in Row 1, the segmentation for the people in the background for persons 3 and 5, due to the weak disparity cue for these people far away from the camera. The numbers denote the front (low values) to back (high values) ordering of people.*

Tab. 5.1: *Evaluation of pixel-wise person segmentation on our Inria 3DMovie dataset. We used precision, recall and intersection vs. union scores to compare the methods. Our method ("Proposed"), which uses disparity, colour, and motion features, along with pose likelihoods and temporal terms shows the best performance. We also show results of variants of our approach and two baseline methods.*

Method	Precision	Recall	Int. vs Union
Proposed	**0.869**	**0.915**	**0.804**
Variants of our method:			
No mask, single frame	0.525	0.371	0.278
Uni mask, single frame	0.783	0.641	0.544
Pose mask, single frame	0.849	0.905	0.779
Baselines:			
Colour only	0.778	0.769	0.630
[Eichner *et al.*, 2012]	0.762	0.853	0.662

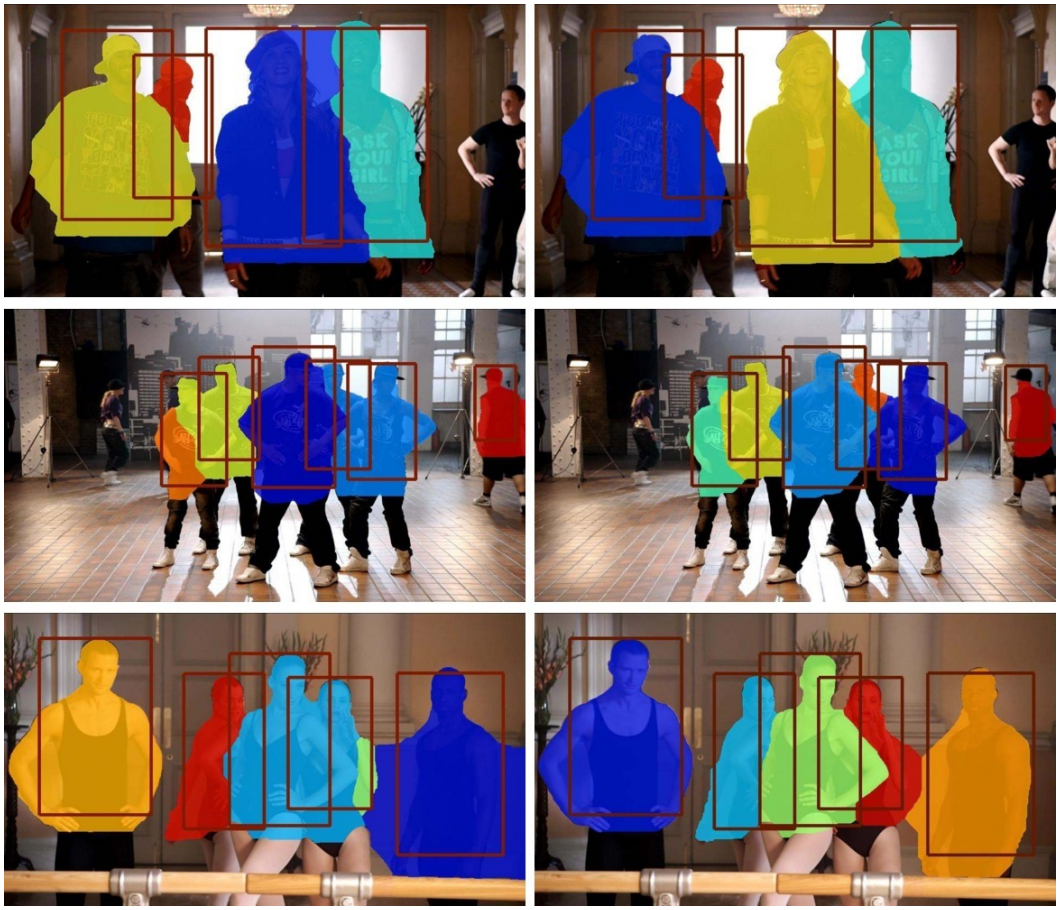
bounding boxes harvested from regions with no person detections. Another baseline we compared with, is derived from the recent work of [Eichner *et al.*, 2012], which computes the pose of a person in a scene. We evaluated the (monocular) person *vs*. background segmentation performed as part of this formulation on our dataset.

We used the precision, recall, and intersection *vs*. union [Everingham *et al.*, 2011] measures to evaluate our segmentation results. From Table 5.1, our method "Proposed" shows the best performance. The poor performance of the *Colour only* method, despite a reasonable initialization for the histograms, is perhaps an indication of the difficulty of our dataset. From Figures 5.1 and 5.7 we note that the person *vs*. background distinction is not very marked in the colour feature space. Furthermore, these images appear to be captured under challenging lighting conditions.

We then evaluated the benefits of the temporal smoothness terms in (5.1). Performing segmentation temporally shows a 2% increase in the intersection *vs*. union score (Table 5.1). We also observe that it reduces flickering artifacts, produces more consistent segments and reduces leaking in the segmentation, as shown in Figure 5.8 and the video results[4]. Other methods [Budvytis *et al.*, 2011] to propagate segmentations from a few key frames of the video onto others can also be used.

Results on a few sample frames for the "Proposed" method are shown in Figure 5.7. The influence of the articulated pose mask is analyzed in Figure 5.9. Another component of our model – the smoothness terms based on colour, motion, and depth – are analyzed in Figure 5.10.

[4] `http://www.di.ens.fr/willow/research/stereoseg`



(a) (b)

Fig. 5.8: *Comparison of segmentation performed: (a) individually on each frame; and (b) temporally on video. We overlay the result of our person detector on each image. We observe that the temporal consistency term reduces leaking (Row 1, rightmost person). It also helps segment more people in the scene accurately (Rows 2 and 3).*

The success of our approach depends on the quality of detections. Here, we operated in the high-precision mode, at the expense of missing difficult examples, e.g., heavily occluded people. Other prominent failure modes of our method are: (i) challenging poses, which are very different from the training data; and (ii) cases where the disparity signal is noisy for people far away from the camera (e.g., Figure 5.7, row 1).

5.5.2 Sensitivity to parameters

In this section we experimentally investigate the sensitivity of the proposed algorithm to its main parameters. The parameter α^l in (5.6) moderates the relative weight of the pose mask and the disparity cues for person label l. We used one single $\alpha = 0.45$ for all the labels in the results discussed thus far. In Figure 5.11(a) we show the influence of varying α on the segmentation score. We observe that

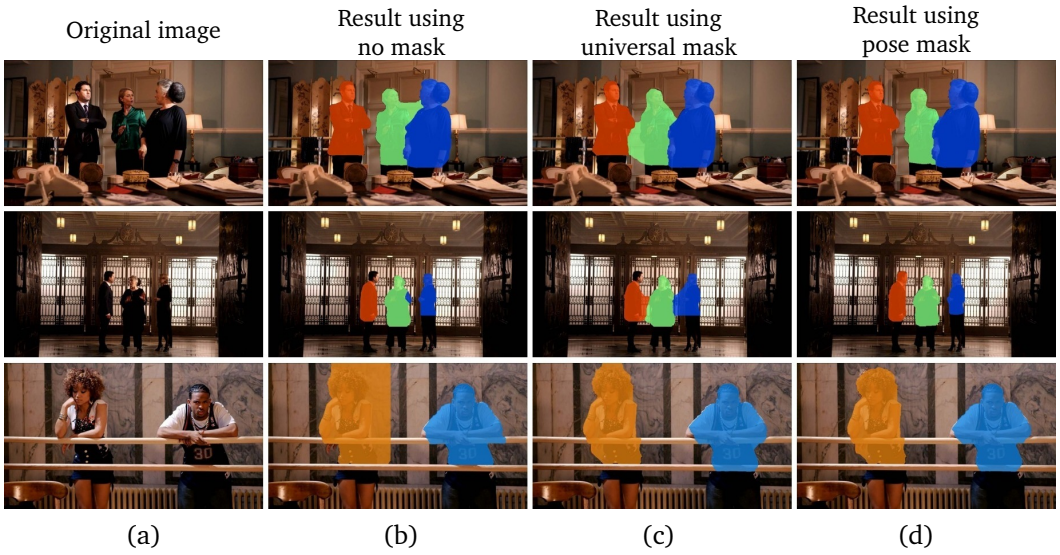


Fig. 5.9: *Benefits of the articulated pose mask. (a) Left input image. (b) Segmentation result using no mask. In this case, the disparity-based likelihoods are not combined with any pose prior. (c) Segmentation result using a single universal pose mask. The disparity-based likelihood is combined with a potential computed from the universal mask.(d) Segmentation result using articulated pose-specific masks; see Section 5.3.3. We observe that using a mask improves the segmentation, and the pose-specific masks show the best performance.*

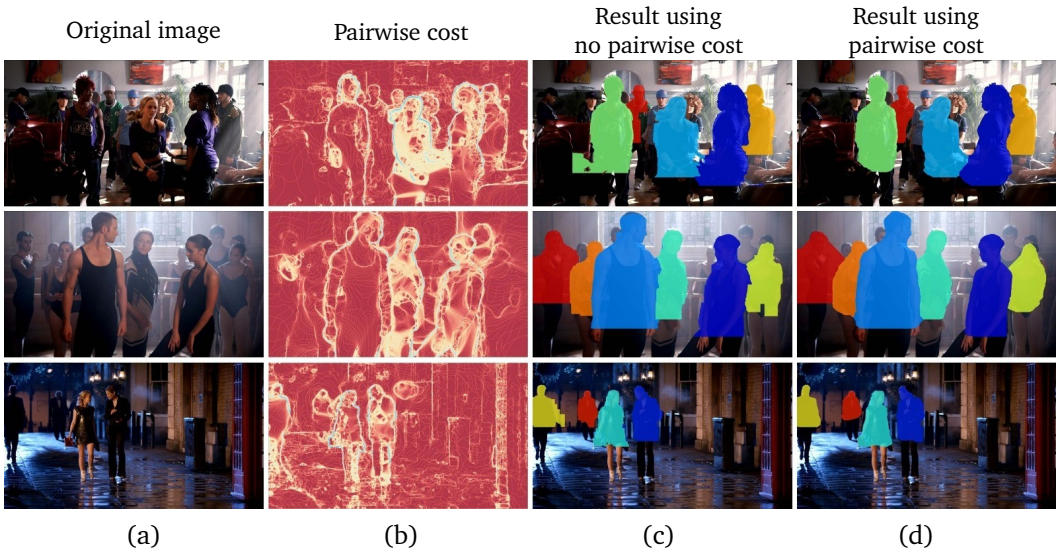


Fig. 5.10: *Influence of the motion, colour and disparity sensitive smoothness cost on segmentation results. (a) Left input image. (b) Illustration of the spatial smoothness cost. Red denotes high cost, and the yellow to blue range of colours denotes low cost. (c) Segmentation result using no smoothness cost. (d) Segmentation result using the smoothness cost. Using this pairwise term reduces person segments leaking into the background.*

using no pose cues (i.e., $\alpha = 1.0$) shows a lower average performance than giving equal importance to pose and disparity cues on the entire dataset. However, we note that increasing the influence of the disparity term segments articulated poses more accurately, as shown in Figure 5.12, at the expense of reduced precision in other situations, such as scenes with multiple people who are close to each other and at similar depth where pose estimates help. We use $\alpha = 0.45$ so that the pose and disparity terms have nearly equal influence and avoid a bias towards one of the extremes.

We also analyzed the influence of the parameters λ_1, λ_2 and λ_3 in the pairwise term (5.8). The segmentation score was fairly robust to changing these parameters. For instance, disabling any of the three terms still leads to a reasonable performance, and varying the relative influence of each term showed only minor variations in the segmentation quality. In contrast, changing the overall influence of the pairwise term, λ in (5.8), shows first a slight increase in the segmentation score but putting too much weight on the pairwise terms reduces the segmentation score as shown in Figure 5.11(b).

5.5.3 Analysis with ground truth components

We further analyze the robustness of our approach by replacing its components with ground truth results. In particular, we use ground truth person detections, pose estimates and disparity parameters. The ground truth disparity parameters are mean and standard deviation computed with the disparity values of all the pixels within each ground truth person segmentation mask. The analysis is performed on individual frames, where ground truth annotations are available, i.e., using the method "Pose mask, single frame" (see Section 5.5.1) without any temporal smoothing. The results are summarized in Table 5.2 and demonstrate that using the noisy disparity and pose estimates (rows 1-3) results in only a moderate loss in the segmentation accuracy compared to the segmentation with their ground truth values (row 4). Please note that the the segmentation results in Tables 5.1 and 5.2 are not directly comparable, since all results in Table 5.2 are based on the full set of ground truth person detections.

5.5.4 H2view dataset

The H2view dataset [Sheasby *et al.*, 2012] was acquired using a static stereo rig, in combination with a Kinect active sensor. Ground truth poses and segmentations are available for 7 test video sequences, with a total of 1598 annotated frames. It is, however, restricted to a single person setup and hence has no inter-person occlu-

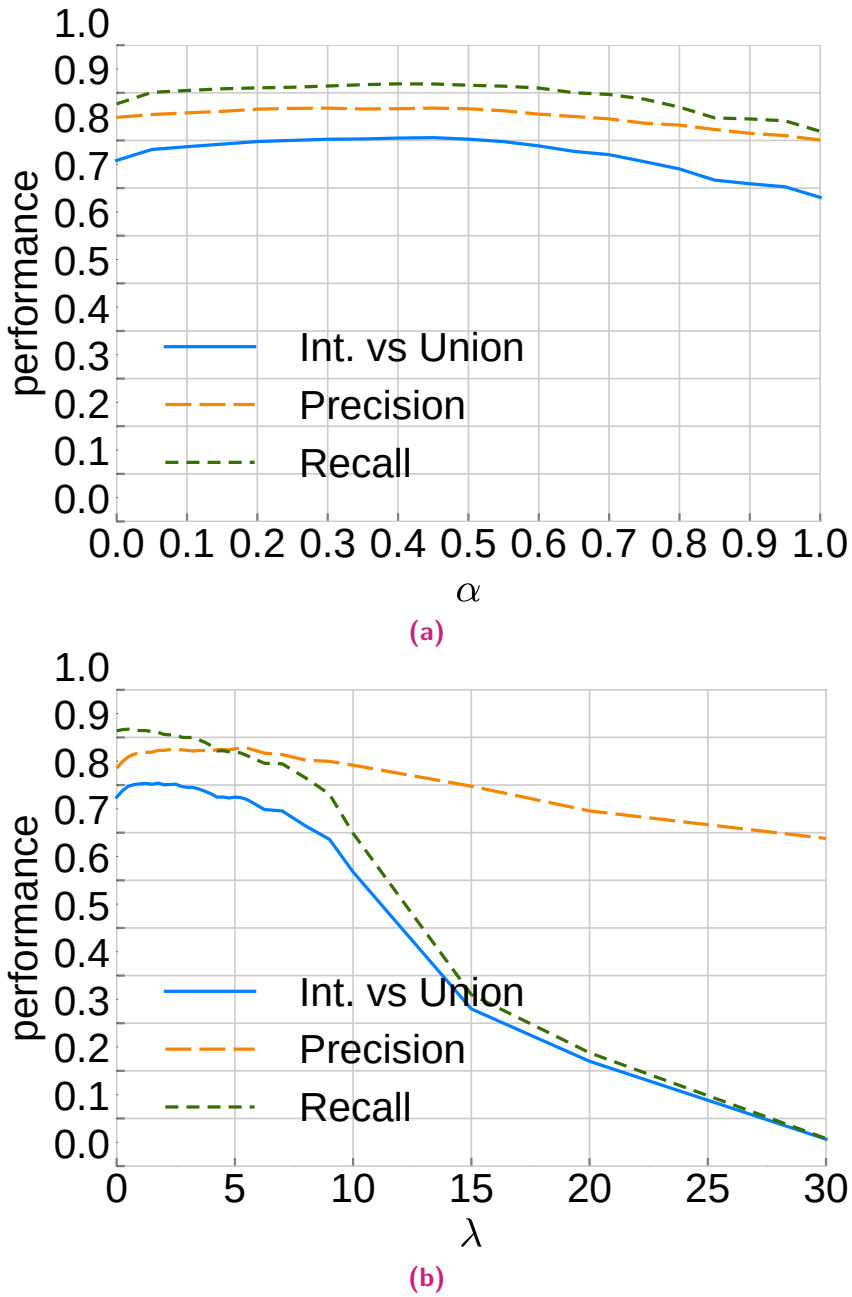


Fig. 5.11: *(a) Influence of the parameter α, specifying the relative weight of the pose mask and disparity cues. All the results in this chapter are produced with $\alpha = 0.45$. Using only disparity cues ($\alpha = 1.0$) leads to worse overall performance than using a combination of pose and disparity cues. (b) Influence of the overall weight λ of the pairwise terms. We use $\lambda = 1.0$ in all the experiments.*

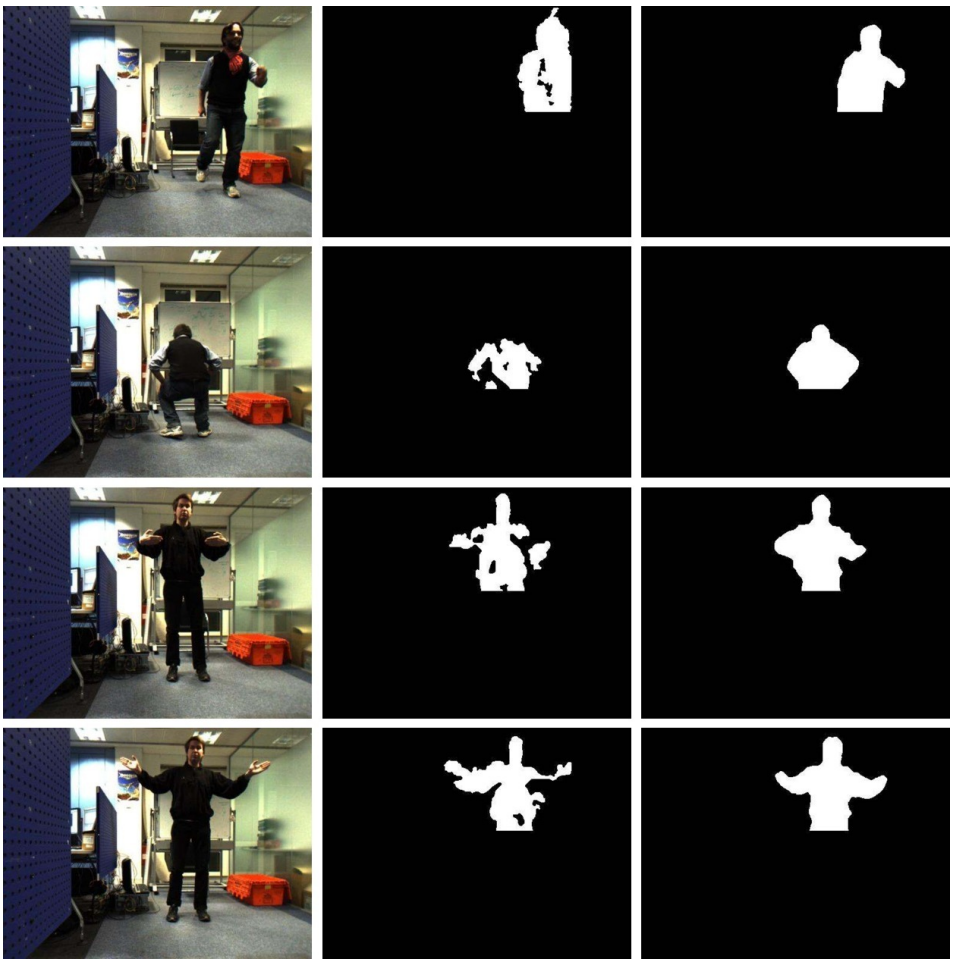
(a) $\alpha = 0.2$ (b) $\alpha = 0.45$ (c) $\alpha = 0.8$

Fig. 5.12: *Qualitative influence of the mixing parameter α, specifying the relative weight of the pose mask and disparity cues. Note that putting more weight on the disparity cues (increasing α) results in a better segmentation of people with articulated poses (Row 1), but performs worse when multiple people at a similar depth are close to each other (Row 3).*

sions. We tested our model (trained on 3D movies) directly on this dataset, without any further tuning, and analyzed the segmentation quality using the evaluation code from [Sheasby *et al.*, 2012]. As our method models only the upper body, we cropped the ground truth, our results, and those from [Sheasby *et al.*, 2012] just above the hips, and considered only upper body (rather than full body) segmentation. Our method achieves a segmentation overlap score of 0.825 compared to their 0.735 (see Table 5.3). Qualitative results on frames from different sequences in the H2view dataset are shown in Figure 5.13. Our segmentation produces cleaner, and more human-like shapes, compared to the seed-based segmentation from [Sheasby *et al.*, 2012].

An extension of our method for full body segmentation can be envisaged by expanding the bounding boxes (in which we perform the segmentation) vertically. Since our articulated pose mask does not capture the lower limbs, we only used depth cues in this setting. Although this led to some leaking in the segmentation result (due to the noisy disparity signal close to the ground), our method achieves an overall segmentation performance similar to [Sheasby *et al.*, 2012] (see Table 5.3).

Computation time: On a 960×540 frame it takes about 13s to detect and track people, 8s to estimate the pose of each person, and 30s per frame to perform the



(a) (b) (c)

Fig. 5.13: *Qualitative results on images from the H2view dataset. (a) The original image, (b) result from [Sheasby* et al.*, 2012] (upper body only), and (c) our result, are shown in each row. Note that our approach shows better performance, including cases with challenging poses (Row 2). Some of the finer details in the segmentation could be improved further, e.g. hands.*

Tab. 5.2: *Evaluation of pixel-wise person segmentation on our Inria 3DMovie dataset using ground truth components. We show results using ground truth detection (Det.), ground truth pose masks (Pose) and ground truth disparity parameters τ (Disp.). Using the noisy estimated pose and disparity parameters (rows 1-3) results in only a moderate loss in the segmentation accuracy compared to the segmentation with their ground truth values (row 4).*

Method	Precision	Recall	Int. vs Union
Variants with ground truth:			
Det.	0.862	0.864	0.759
Det. + Disp.	0.872	0.884	0.782
Det. + Pose	0.869	0.908	0.799
Det. + Pose + Disp.	**0.892**	**0.929**	**0.835**

Tab. 5.3: *Evaluation of pixel-wise person segmentation on the H2view dataset. Our method for segmenting upper bodies shows about 9% improvement in int. vs. union score over [Sheasby et al., 2012]. Note that our method for full body segmentation only uses upper body pose mask.*

Method	Precision	Recall	Int. vs Union
Upper body segmentation:			
[Sheasby *et al.*, 2012]	0.848	0.841	0.735
Proposed	**0.940**	**0.871**	**0.825**
Full body segmentation:			
[Sheasby *et al.*, 2012]	0.796	**0.832**	0.692
Proposed	**0.880**	0.789	**0.706**

segmentation with our non-optimized Matlab implementation. The time for segmentation is 6s per frame for the H2view dataset, which contains 512×384 frame sequences of a single person.

5.6 Discussion

We have developed a model for segmentation of people in stereoscopic movies. The model explicitly represents occlusions, incorporates person detections, pose estimates, and recovers the depth ordering of people in the scene. The results suggest that disparity estimates from stereo video, while noisy, can serve as a strong cue for localizing and segmenting people. The results also demonstrate that a person's pose, incorporated in the form of an articulated pose mask, provides a strong shape prior for segmentation. The developed representation presents a building block for modelling and recognition of human actions and interactions in 3D movies.

6 Multiple person segmentation under weak constraints

In this chapter, we address the problem of segmenting multiple object instances in complex videos, and in particular multiple persons. The previous chapter introduced a method for segmenting multiple persons in 3D movies which relies on person detection, pose estimates and disparity cues to perform segmentation. However, errors on pose estimates may lead to strong segmentation errors, and producing pose masks for other object classes may be impossible. Here, we aim to formulate a method which does not require manual pixel-level annotation for training, and relies instead only on readily-available object detectors and visual object tracking. Given object bounding boxes as input, we cast video segmentation as a weakly-supervised learning problem. Our proposed objective combines (a) a discriminative clustering term for background segmentation, (b) a spectral clustering one for grouping pixels of same object instances, and (c) linear constraints enabling instance-level segmentation. We propose a convex relaxation of this problem and solve it efficiently using the Frank-Wolfe algorithm. We report results and compare our method to several baselines on a challenging dataset for multi-person segmentation, *Inria 3DMovie Dataset v2*, an extension of the dataset used for evaluation in Chapter 5. We also report results and comparisons on a standard benchmark dataset for video segmentation, *SegTrack* [Tsai *et al.*, 2010].

6.1 Introduction

Semantic object segmentation in images and videos is a challenging computer vision task [Joulin *et al.*, 2010; Lempitsky *et al.*, 2009; Li *et al.*, 2013; Shi and Malik, 2000; Vineet *et al.*, 2011]. Common difficulties arise from frequent occlusions [Taylor *et al.*, 2015] and background clutter, as well as variations in object shape and appearance. Video object segmentation also requires accurate tracking of object boundaries over time in the presence of possibly fast and non-rigid motions. An additional challenge addressed by several recent works is in segmentation of individual instances of the same object class [Hariharan *et al.*, 2014; He and Gould, 2013; Vineet *et al.*, 2011; Tighe *et al.*, 2014; Zhang *et al.*, 2015b]. Indeed, while it may be easy to segment a herd of cows from a grass field, segmenting each cow separately is a much harder task.

Fig. 6.1: Results of our method applied to multi-person segmentation in a sample video from our database. Given an input video together with the tracks of object bounding boxes (left), our method finds pixel-wise segmentation for each object instance across video frames (right).

Instance-level object segmentation in video is an interesting and understudied problem at the intersection of semantic and motion-based video segmentation. Solutions to this problem can benefit from class-specific object models and motion cues. Segmentation of static and/or partially occluded objects of the same class, however, pose additional challenges, difficult to solve with existing methods of motion-based and semantic segmentation. Meanwhile, successful solutions to instance-level video segmentation can serve in several tasks such as video editing and dynamic scene understanding.

Given recent advances in object detection [Ren *et al.*, 2015] and visual object tracking [Danelljan *et al.*, 2014], coarse object localization in the form of object bounding boxes can now be used as input for solving other problems. In particular, we address in this chapter the problem of instance-level video segmentation given object tracks. We assume that prior (weak) knowledge about objects is available in the form of tracked object bounding boxes, obtained by a separate process. For instance, pre-trained object detectors or visual object tracking algorithms as the ones cited above can be used.

Segmentation methods typically optimize carefully designed objective functions combining data terms and prior knowledge. Object prior knowledge in such methods is often encoded by higher-order potentials [Ladický *et al.*, 2009; Ladický *et al.*, 2010; Seguin *et al.*, 2015], which enable richer modeling but lead to hard optimization problems. Here we take an alternative approach and build on the discriminative clustering framework [Bach and Harchaoui, 2007; Guo and Schuurmans, 2007]. Following previous work on co-segmentation [Joulin *et al.*, 2010] and weakly-supervised classification [Bojanowski *et al.*, 2013], we formulate our problem as a quadratic program under linear constraints. We use object tracks as constraints to guide segmentation, but other forms of prior knowledge could easily be integrated in our method. Our final segmentation is obtained by solving a con-

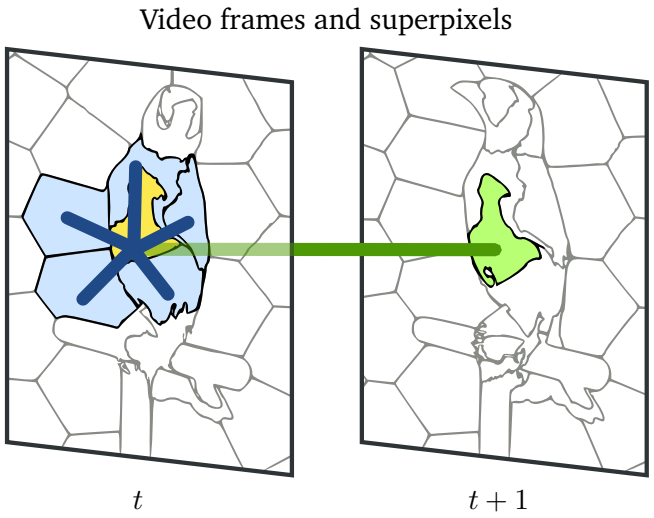


Fig. 6.2: Spatio-temporal graph of superpixels. For the yellow superpixel, spatial edges are shown in dark blue and temporal edges in dark green.

vex relaxation of our objective with the Frank-Wolfe algorithm [Frank and Wolfe, 1956].

We compare our method to the state of the art and show competitive results on a new dataset for instance-level video segmentation. In contrast to most previous methods, our approach segments multiple instances of the same object class and supports reasoning about occlusions. Figure 6.1 illustrates the data and results of our method on a sample video from our dataset.

The contributions of this chapter are two-fold. **(i)** We propose a discriminative clustering approach for instance-level video segmentation using external guidance in the form of object bounding boxes. **(ii)** We demonstrate the high accuracy and flexibility of our model on the task of multi-instance person segmentation in video.

The rest of the chapter is organized as follows. We present our problem formulation in Section 6.2. We describe the convex relaxation of our model and the optimization of the cost function with the Frank-Wolfe algorithm in Section 6.3. Section 6.4 presents our experimental setup and results. We then study how our method performs when using pixel-wise instance-specific priors in Section 6.5. Finally, in Section 6.6 we show that our method performs well when applied on other object classes and that it is very easy to adapt it to perform slightly different tasks, such as segmentation propagation.

6.2 Problem formulation

The segmentation problem we aim to solve is to assign to every pixel a label in $\{0, \dots, K\}$. To design a suitable cost function, we follow previous work on co-segmentation [Joulin *et al.*, 2010; Joulin *et al.*, 2012]. This implies using two complementary cost functions: the first one is a spectral clustering term [Shi and Malik, 2000], which enforces spatial and temporal consistencies according to some descriptors ϕ. The second term is a discriminative clustering cost based on the square loss [Bach and Harchaoui, 2007] which learns a foreground *vs*. background classifier. In order to include prior information, we propose several constraints which we detail in Section 6.2.4. The proposed constraints are linear, leading to a tractable (relaxed) optimization problem (see Section 6.3).

The intuition behind our approach is that constraints provide weak localization cues for each object instance. Discriminative clustering separates foreground objects from the background based on appearance features. Spectral clustering helps producing clean spatial boundaries, separating different instances of the same class and smoothing the segmentation in time for each object instance.

6.2.1 Notations and model

We are given a video clip composed of T frames indexed by t. Our problem is to assign a label k in $\{0, 1, \dots, K\}$ to each pixel in each frame, where label $k = 0$ corresponds to the background and all other integers in $\{1, \dots, K\}$ correspond to the K object instances in the video. Since the number of pixels in a video is usually high, we propose to work with superpixels instead. Assuming that there are N superpixels in the whole video, we index them by n in $\{1, \dots, N\}$.

Let us define a label matrix y in $\{0, 1\}^{N \times (K+1)}$. The matrix y is such that y_{nk} is equal to one if and only if the superpixel n is of label k. This matrix sums up to one along rows, since every superpixel is assigned to a single label. In Section 6.2.4, we propose several constraints that will restrain the set of admissible matrices y. We denote by $\mathcal{Y}$ this set. The constraints can be indexed by c in $\{1, \dots, C\}$. Since some of them may not be satisfied, for every constraint c, we define a slack variable ξ_c which will allow us to violate it. Let ξ be the concatenation of all the ξ_c into a single vector. We denote by $\mathcal{C}(y, \xi)$ the set of constraints over a specific y with slack ξ.

The cost we minimize is a sum of three terms: a grouping term E_G, a discriminative term E_D, and a term penalizing the slack ξ:

$$\min_{y \in \mathcal{Y},\ \xi \in \mathbb{R}^C_+} \quad E_G(y) + \alpha\, E_D(y) + \beta \|\xi\|^2, \tag{6.1}$$

under linear constraints $\mathcal{C}(y, \xi)$, where α and β allow us to weigh the different terms. We provide a detailed description of these terms in the following sections.

6.2.2 Grouping term

The grouping term E_G is a classic spectral clustering term meant to ensure spatial and temporal consistency of the segmentation, as described in Section 3.4. To this end, we define a *superpixel graph* $G = (S, \mathcal{E})$, whose nodes correspond to superpixels and edges encode spatio-temporal neighborhood information. A sample graph G is illustrated in Fig. 6.2. For two nodes n and n' from the same frame, there is an edge (n, n') in $\mathcal{E}$ if the two superpixels are spatial neighbours. For node n in frame t and node n' in frame $t+1$, we add an edge (n, n') to $\mathcal{E}$ if n and n' are temporal neighbours. The exact way we define neighbourhoods is discussed in Section 6.4.1.

For each superpixel n, we define a set of descriptors ϕ^i_n indexed by i in $\{1, \dots, I\}$. We denote by d_i the dimension of ϕ^i_n and by d the sum of all the d_i. Let us denote by ϕ_n the concatenation of all the ϕ^i_n. We then define the similarity matrix W in $\mathbb{R}^{N \times N}$ which encodes the similarities between superpixels: $W_{nn'} = \sum_{i=1}^{I} \mu_i \exp(-\lambda_i \|\phi^i_n - \phi^i_{n'}\|^2)$ if $(n, n') \in \mathcal{E}$ and 0 otherwise. μ_i and λ_i are weighting parameters for the i-th descriptor.

Following [Shi and Malik, 2000], we define the associated unnormalized Laplacian matrix $L = D - W$. D is the diagonal matrix composed of the row sums of W: $D = \mathrm{Diag}(W\mathbf{1}_N)$. Using these definitions, the grouping term can be written as the following quadratic form:

$$E_G(y) = \frac{1}{N} \mathrm{Tr}(y^T L y). \tag{6.2}$$

6.2.3 Discriminative term

E_D is a standard discriminative clustering term as described in Section 3.5. We use it to learn a foreground *vs*. background model is fit for segmenting the background from multiple instances of the same object category.

It aims to learn an affine classifier for separating foreground *vs*. background. Let M be a binary matrix in $\{0,1\}^{(K+1)\times 2}$ which maps labels to foreground and background. Let us denote by $w \in \mathbb{R}^{d\times 2}$ and b in $\mathbb{R}^2$ the parametrization of this model. We also define the matrix Φ in $\mathbb{R}^{N\times d}$ whose rows are the ϕ_n. The discriminative cost is defined as follows:

$$E_D(y) = \min_{\substack{w\in\mathbb{R}^{d\times 2}\\ b\in\mathbb{R}^2}} \frac{1}{N}\|yM - \Phi w - 1_N b^T\|_F^2 + \kappa||w||_F^2. \tag{6.3}$$

The minimization w.r.t. w in (6.3) is a ridge regression problem, whose solution can be found in closed form, and E_D is easily rewritten [Joulin *et al.*, 2010] as a quadratic form in y:

$$E_D(y) = \frac{1}{N}\mathrm{Tr}(M^T y^T A y M), \tag{6.4}$$

where $A = \frac{1}{N}\Pi_N(I_N - \Phi(\Phi^T\Pi_N\Phi + N\kappa I_d)^{-1}\Phi^T)\Pi_N$ and Π_N is the centering projection matrix $I_N - \frac{1}{N}1_N 1_N^T$.

Note that when dealing with multiple instances of multiple object categories, we could easily learn one model per object category by adapting the M matrix.

Overcoming trivial solutions. The optimization problem (6.1) is similar to the one of [Joulin *et al.*, 2010]. It has trivial solutions, which include the constant matrix and the column-wise constant matrices. These solutions are due to the symmetries of the discriminative clustering objective, as noted by [Guo and Schuurmans, 2007]. A standard technique to tackle this problem is to perform a lifting from label matrices to equivalence matrices [Bach and Harchaoui, 2007], namely, to perform the optimization in $Y = yy^T$ instead of y. In our case, we get rid of symmetries by constraining the optimization space, as done by [Bojanowski *et al.*, 2013] in the context of person identification in movies.

6.2.4 Constraints

As mentioned earlier, our model incorporates constraints on the y matrix. They allow us to encode simple priors as well as more complicated, instance-specific information. We can constrain the number of superpixels assigned to a given label in a spatio-temporal region using linear inequalities. We can also use strict equality constraints to fix the labels of some superpixels. We first provide a general form and then describe the different variants used in our experiments. Some of them are also illustrated in Fig. (6.3) for multi-instance person segmentation using head and full-body tracks.

Object tracks. We assume that we are given a track of bounding boxes for each object in the video. We denote by $\mathcal{B}$ this set and index the elements B of $\mathcal{B}$ by k in $\{1, \dots, K\}$ and t in $\{1, \dots, T\}$, such that B_k^t denotes the bounding box of the k-th object in frame t.

Inequality constraints. We want to impose linear inequality constraints on a set of superpixels in the video. In the following sections we will describe in details what these sets can correspond to. For now, let us denote by R a subset of the indices of superpixels, $R \subset \{1, \dots, N\}$. We can represent R by the indicator vector $\mathbb{1}_R$, such that the n-th entry is equal to one if the superpixel n is in R. Note that for videos, this set R can correspond to a spatio-temporal region. We use the notation $\mathbf{e}_k$ to denote the k-th vector of the canonical basis of $\mathbb{R}^{K+1}$.

For some region R and a label k, we propose to constrain the matrix y using constraints of the following form:

$$0 \geq \sigma \left(\mathbb{1}_R^T \, y \, \mathbf{e}_k - \rho \right) - \xi_c, \tag{6.5}$$

where $\sigma \in \{-1, 1\}$ controls whether this is an *at least* or an *at most* constraint, ρ a parameter and ξ_c is the slack variable allowing this constraint to be violated. Intuitively, $y \, \mathbf{e}_k$ selects the k-th vector of the label matrix y which indicates whether a superpixel is assigned to label k or not. $\mathbb{1}_R^T \, y \, \mathbf{e}_k$ then counts the number of superpixels from region R which have the label k.

The parameters R, σ, k and ρ depend on the kind of prior we want to enforce. For instance, if we want to enforce that at least $\rho = 50$ superpixels of a given region R are assigned to the label of the first object instance $k = 1$, we would add the following *at least* constraint (thus with $\sigma = -1$):

$$\begin{aligned} 0 \;&\geq\; -1 \left(\mathbb{1}_R^T \, y \, \mathbf{e}_1 - 50 \right) - \xi_c \\ \iff 50 \;&\leq\; \mathbb{1}_R^T \, y \, \mathbf{e}_1 + \xi_c. \end{aligned}$$

Replacing $\sigma = -1$ with $\sigma = 1$ would make it an *at most* constraint, enforcing that at most 50 superpixels of R are assigned to the label $k = 1$.

Note that while our notations refer to superpixels and counts of superpixels, in practice we weigh the contribution of each superpixel to the constraint by its relative area in region R. Likewise, we reason in terms of pixels when computing the ρ parameters. Let us denote by s the vector which n-th entry is equal to the number

of pixels in superpixel n, and $\mathrm{Diag}(s)$ the $N \times N$ diagonal matrix which diagonal elements are the entries of s. Our inequality constraints are actually written as:

$$0 \geq \sigma \left(\frac{1}{\mathbb{1}_R^T \, s} \, \mathbb{1}_R^T \, \mathrm{Diag}(s) \; y \; \mathbf{e}_k - \rho \right) - \xi_c. \tag{6.6}$$

As above, $y \; \mathbf{e}_k$ is a column vector which indicates whether a superpixel is assigned to label k or not. $\mathrm{Diag}(s) \; y \; \mathbf{e}_k$ then corresponds to a column vector containing the areas of the superpixels assigned to the k-th label and 0 for superpixels which are not assigned to this label. In turn, $\mathbb{1}_R^T \, \mathrm{Diag}(s) \; y \; \mathbf{e}_k$ counts the sum of areas of superpixels which are both inside the region R and assigned to the label k. Finally, $\mathbb{1}_R^T \; s$ counts the total area of the superpixels inside the region R and is used to normalize the constraint so that ρ can be expressed independently of the specific area of the region R.

As the formulation of these constraints is generic and not specific to the underlying structure of the problem, they can indifferently encode prior knowledge over bounding boxes, frame regions or even entire volumes of the video.

Equality constraints. When some supervision is available (semi-supervised setting), or when a strong cue allows us to freeze variables, we want to use equality constraints. Let us suppose that we have a set of superpixels R and a set of labels Q. We set variables for region R and labels Q to predetermined values stored in $\tilde{y}$:

$$\forall r \in R, \; \forall q \in Q, \quad y_{rq} = \tilde{y}_{rq}. \tag{6.7}$$

As for the inequality constraints, the definitions of R, Q and $\tilde{y}$ depend on the prior.

Must-link/must-not-link constraints When prior knowledge over a set of superpixels R, such as a supervoxel, specifies that these superpixels should share the same label, we can apply must-link constraints. If r_0 is one element of this set, then we can write these constraints as linear equality constraints:

$$\forall r \in R, \; \forall k \in \{1, \ldots, K+1\} \quad y_{rk} - y_{r_0 k} = 0. \tag{6.8}$$

Likewise, must-not-link constraints can be imposed. If the superpixels of the set R should not take the same label as superpixel r_0 (not in R), then:

$$\forall r \in R, \; \forall k \in \{1, \ldots, K+1\} \quad y_{rk} - y_{r_0 k} \neq 0. \tag{6.9}$$

Track constraints. Given an object bounding box B_k^t, we require that at least ρ_B superpixels inside B_k^t get assigned the label k. This can be enforced by setting R,

and σ appropriately in Eq. (6.5). We set R to the set of superpixels that lie inside B_k^t. Since this is an *at least* constraint, we set $\sigma = -1$. The amount of superpixels ρ_B is set to a ratio of the total number of superpixels in B_k^t. In Figure 6.3 (a) and (b), head tracks and object tracks are used for such constraints.

In complex videos picturing multiple objects, the bounding boxes, and thus the corresponding constraint regions, can heavily overlap. Without slack variables, our problem may be infeasible in such situations, and even with slack variables the constraints may still be misleading. To cope with occlusions, we propose a simple occlusion reasoning. In a given frame, for each pair of overlapping bounding boxes, pixels inside the region of overlap are marked as occluded. In turn, we reduce the strength of each such constraint by multiplying ρ_B by $(1 - o)$ where o is the ratio of occluded pixels in the bounding box.

Area constraints. To reduce "leaking" effects in the segmentation, we constrain the area of each object segment in each frame. For object k in frame t, we impose that at most ρ_{area} of the superpixels in frame t get assigned the label k. This can be expressed by setting R to be the set of superpixels in frame t. Since this is an *at most* constraint, we have $\sigma = +1$. We set ρ_{area} to the amount of superpixels in track B_k^t times a constant, to take object size into account.

We can also enforce a minimal amount of superpixels per label and per frame. We do so by changing σ to -1 and setting an appropriate ρ. This constraint can be used if we know the object is in the frame but lack the corresponding bounding box.

Background constraints. We request that most superpixels which are outside object bounding boxes belong to the background label. The rationale is that only a few of the superpixels outside object detections may belong to objects, as shown in Figure 6.3 (c). Typically, in the case of multiple people segmentation, these superpixels belong to lower arms. We express this constraint by setting R to the set of superpixels that do not belong to any track in frame t. This is an *at least* constraint so we set $\sigma = -1$. We set $\rho = \rho_{\text{bg}}$ to a ratio of the cardinality of R.

Non-object constraints. In our work, we make the assumption that if a pixel is far enough from an object detection, it is reasonable to assume that it does not belong to the corresponding object. We assume that when there are no detections at all, we do not apply these constraints. For a bounding box B_k^t, we build R as the set of superpixels in frame t that are further away from B_k^t than a given distance, as shown in Figure 6.3 (d). In practice, we set this minimum distance to the width of the object bounding box. R can be computed by performing a distance transform

(a) Head constraint.

(b) Body constraint.

(c) Background constraint.

(d) Non-person.

(e) Must-be background.

Fig. 6.3: Constraints (see Section 6.2.4) used in our model for multi-person segmentation. In this setup we are provided head detections, from which we derive body boxes. We require 75% of pixels inside head detections (a) and 50% of pixels inside body boxes (b) to belong to the instance. Part (c) illustrates the background constraint (96% of this surface should be background); non-person constraints which enforce superpixels far from the person to be assigned to the corresponding label (d) ; and the superpixels which can only be background (e).

and then thresholding. We then enforce an equality constraint with Q containing only the label k and $\tilde{y}$ filled with zeros.

Ground-truth supervision. Pixel-wise ground-truth segmentation can sometimes be provided for some frames, e.g.for the task of segmentation propagation. In such cases we can use equality constraints and assign $\tilde{y}$ according to the provided segmentation mask for frame t. R is the set of superpixels in frame t and Q contains all the labels. This constraint is a strong cue for both the grouping term and for the discriminative term as the fixed variables provide reliable cues to learn the discriminative model.

6.3 Optimization

6.3.1 Continuous relaxation

The quadratic problem defined in Eq. (6.1) is known to be NP hard when y takes binary values. Indeed, when the quadratic cost matrix has positive off diagonal entries, this is as hard as solving a max-cut problem. Classic relaxations of such problems [Joulin *et al.*, 2010] imply working with equivalence matrices $Y = yy^T$. Doing so in our case would be intractable due to the problem size and would prevent us from imposing constraints relating superpixels to labels. Instead, we propose a continuous relaxation of our problem by solving it over the convex hull $\overline{\mathcal{Y}}$ of the initial set $\mathcal{Y}$. Then, we aim at solving the minimization of a positive semi-definite quadratic form over a convex compact set defined by a large number of linear constraints. Due to the size of y (of the order of 10^6 entries) and the number of constraints it is not realistic to use a standard off-the-shelf quadratic programming solver based on interior point methods [Boyd and Vandenberghe, 2004]. Nevertheless it is possible to solve linear programs of such a size. This is why, following other approaches to discriminative clustering [Bojanowski *et al.*, 2014], we propose to use the Frank-Wolfe optimization algorithm [Frank and Wolfe, 1956; Jaggi, 2013] which only relies on the minimization of linear forms over $\overline{\mathcal{Y}}$.

6.3.2 Frank-Wolfe algorithm

The Frank-Wolfe algorithm, also known as the conditional gradient method, is an iterative method to optimize convex objectives over compact convex sets. Intuitively, it is an iterative optimization algorithm which considers a linear approximation of the objective function at each iteration, finds a point of the domain minimizing this linear approximation and moves a bit the current point towards this minimizer. It

is thus particularly handy when solving linear problems over the domain is possible but other operations are either impossible or too expensive. Another good feature of this algorithm is that it provides an approximation of the duality gap, which can be used as a reliable stopping criterion for the optimization. Our problem satisfies the desired properties as the optimization domain satisfies the desired properties and the objective function is convex.

Let us now briefly describe the iterations. We define our optimization variable $z = (y, \xi)$ in $\mathcal{Z} = \overline{\mathcal{Y}} \times \mathbb{R}^C_+$. For the sake of simplicity, we rewrite as $E(z)$ the sum of the three terms from Eq. (6.1). Let us denote by z_k the current point at iteration k. At iteration k, we compute the gradient $\nabla_z E(z_k)$ and minimize the following linear form: $\mathrm{Tr}(\nabla_z E(z_k)(z - z_k))$, which is the linear approximation of our quadratic objective function. This can be easily done using a generic LP solver, and yields a corner of the polytope that we will denote z_{FW}. We then update the current point as follows: $z_{k+1} = z_k + \gamma(z_{\text{FW}} - z_k)$. The optimal parameter γ^* leading to the best improvement in that direction can be found in closed form by doing an exact line search. We iterate this procedure until the duality gap d_{gap}, which is an upperbound of the difference between the objectives of the current point and of the optimal solution [Jaggi, 2013], is lower than a predefined threshold.

Algorithm 2: Frank-Wolfe algorithm to solve the problem formulated in Eq. (8).

$k \leftarrow 0$
$d_{\text{gap}} \leftarrow \inf$
while $d_{gap} > \epsilon$ **do**
  Solve $z_{FW} = \arg\min_{z \in \overline{\mathcal{Z}}} \mathrm{Tr}(\nabla_z E(z_k)^T z)$
  Set $d_{FW} = z_{FW} - z_k$
  Set $d_{\text{gap}} = -\mathrm{Tr}(\nabla_z E(z_k)^T d_{FW})$
  Find $\gamma^* = \arg\max_{\gamma \in [0,1]} E(z_k + \gamma d_{FW})$
  Update $z_{k+1} = z_k + \gamma^* d_{FW}$

Note on *away* and *pairwise* steps for the Frank-Wolfe algorithm. The standard Frank-Wolfe iterations are also called *toward* steps. Two alternative types of steps have been proposed in the literature: the *away* [Wolfe, 1970; Guelat and Marcotte, 1986; Lacoste-Julien and Jaggi, 2013] and *pairwise* steps [Allende *et al.*, 2013; Lacoste-Julien and Jaggi, 2015]. Using these alternative steps has significantly improved the speed of convergence in our experiments. This is true even though each iteration is slightly longer due to the additional evaluations. The main downside of these methods is the requirement of the explicit storage of all the points z_{FW} met during optimization with their corresponding weights. But, in our case, the points z_{FW} are sparse binary arrays. Thus this only corresponds in practice to a small increase of the memory cost.

Rounding. Using the Frank-Wolfe algorithm we obtain a solution $z^* = (y^*, \xi^*)$. The solution continuous solution we obtain needs to be rounded. We first freeze the slack variables of the constraints to the values ξ^*. We then round y^* into a binary matrix by finding the closest point to y^* in $\mathcal{Y}$ in terms of Frobenius norm $\|y - y^*\|_F^2$ which is equivalent to

$$\min_{y \in \mathcal{Y}} -2\mathrm{Tr}(y^{*T} y). \tag{6.10}$$

We solve this linear program using the LP solver.

6.3.3 Non-convex refinement

Experimentally, we observe that the convex relaxation of our problem may lead to sub-optimal rounded solutions. Indeed, our model is attracted to a degenerate solution with all constant entries of value $\frac{1}{K+1}$, which has a low objective value for the discriminative term. This is a common drawback of discriminative clustering techniques, as noted by [Joulin *et al.*, 2010; Guo and Schuurmans, 2007]. In order to push our solution away from these near-constant solution, and following the approach of graduated non-convexity [Blake and Zisserman, 1987; Zaslavskiy *et al.*, 2009], we propose to add a concave quadratic term to our objective: $\mathrm{Tr}(y^T(1-y))$, and weight it using a parameter δ. This term encourages the entries y to be close to either 0 or 1. The corresponding optimization problem is the following:

$$\min_{y \in \mathcal{Y},\ \xi \in {\mathbb{R}_+}^C} E_G(y) + \alpha E_D(y) + \beta \|\xi\|^2 + \delta\ \mathrm{Tr}(y^T(1-y)).$$

The parameter δ can be a function of the iteration count k. In practice however, choosing a scalar value is already complicated and we therefore use a piecewise constant function. We first optimize the convex relaxation of our problem with $\delta = 0$. Then we perform Frank-Wolfe steps on the non-convex objective with a non-zero δ which has been selected by parameter search. Although we are only guaranteed to converge to a local optimum of this non-convex function [Bertsekas, 1999, Section 2.2.2], we empirically observe a drastic improvement of performance as shown in Table 6.1.

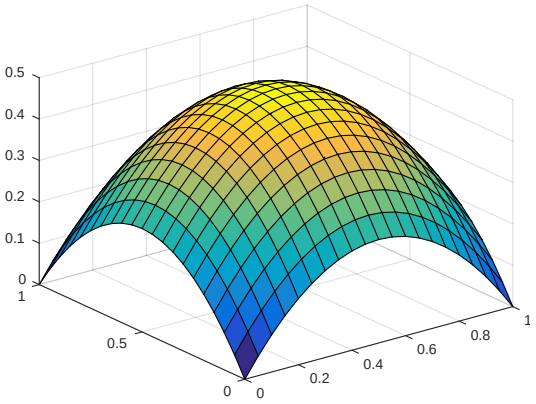


Fig. 6.4: Illustration of the non-convex cost described in Section 6.3.3. Please note that adding this cost makes our cost function non-convex. By controlling the weight of this term, we push the solution towards the extreme points of the optimization polytope.

6.3.4 Hyperparameter search

The proposed model involves several hyperparameters: two per feature channel, two for the discriminative term, one for the type of inequality constraint and others. Consequently, performing a full grid search is prohibitive. We have tried to perform coordinate descent in the parameter space, but the runtime was still too long. Instead, we have optimized hyperparameters using the method in [Snoek *et al.*, 2012], which has allowed us to optimize all hyperparameters simultaneously. The model automatically refines the search space over time by constructing a probabilistic model of the performance and exploiting it to decide where to evaluate next. We use the freely available implementation called Spearmint[1]. This method produces a better parameter set and runs 100 times faster compared to our grid search.

6.4 Experiments on multiple person segmentation under weak constraints

In this section, we describe experimental details and evaluation procedures for the proposed method. We evaluate multi-instance person segmentation in 3D movies using head tracks and full-body bounding boxes on the Inria 3DMovie Dataset v2 from Section 4.2.2.

6.4.1 Implementation details

[1] `https://github.com/HIPS/Spearmint`

Superpixels. We extract video superpixels using [Chang *et al.*, 2013]. The superpixels are evenly distributed, fairly compact, and tracked in time, as shown in Figure 6.5. We use temporal links obtained from superpixel tracks as edges in the superpixel graph (Section 6.2.2). We also add edges between superpixels from consecutive frames if sufficient pixel-wise correspondence is provided by optical flow.

Fig. 6.5: Example of superpixels produced by the method of [Chang *et al.*, 2013]. We set the method parameters to get about 2000 superpixels per frame.

Features. We first compute dense optical flow between consecutive frames using DeepFlow [Weinzaepfel *et al.*, 2013]. Then, we use two different sets of features ϕ_n for the grouping and discriminative terms. These features are computed for each superpixel based on the underlying image pixels. For the spatial edges in the similarity matrix W of the grouping term, we use: **(i)** a histogram of optical flow with 8 bins for orientations and one bin for no motion, and **(ii)** the average CIE L*a*b* color, over the superpixel. For the temporal edges of W, we use the average CIE L*a*b* color. As discriminative features in Φ, we use: **(i)** the same histogram of optical flow, **(ii)** a color histogram computed over RGB colors, with 8 bins per color channel, 512 bins in total, and **(iii)** the average SIFT descriptor over the superpixel, obtained by first computing dense SIFTs over the whole image, and then averaging the SIFTs which cover the superpixel.

We also optionally exploit recent advances in semantic segmentation by including features produced by a deep neural network trained for semantic segmentation for the PASCAL dataset [Zheng *et al.*, 2015]. We take the output of the method for each pixel and pool it (either using max-pooling or mean-pooling) over the superpixel, and use it as an additional discriminative feature in Φ. As this output represents a strong semantic cue, it should help our discriminative term to separate the foreground from the background.

For 3D movies, we also include median disparity over the superpixel in both spatial grouping and discriminative features. The method of [Ayvaci *et al.*, 2012] is used to estimate the disparity map from stereo pairs.

Person detection and tracking. We evaluate our method on ground-truth (manually annotated) head tracks as well as on tracks automatically produced by a tracking-by-detection method: we use a CNN-based detector [Girshick *et al.*, 2014] trained on heads in movies. The tracker associates these detections based on KLT tracks [Shi and Tomasi, 1994], interpolates missing detections and smooths the tracks in time [Everingham *et al.*, 2006]. Using ground-truth or automatic tracks, we extrapolate full-body bounding boxes from the head bounding boxes using a linear transformation. Note that our full-body bounding boxes start below the head, as shown in Fig. 6.1. This way, the superpixels on the sides of the head are not involved in the corresponding constraints, since they do not belong to the person in most cases.

Occlusion reasoning. We adapt the occlusion reasoning of Section 6.2.4 to stereo videos by computing a depth estimate from the median disparity inside the head box. Given two overlapping bounding boxes in the frame, we mark the pixels of the object which is behind as occluded. This procedure allows a more accurate handling of occlusions than the original reasoning, since constraint strength will only be reduced for objects which may actually be occluded.

We evaluate the proposed method on stereo videos where head (bounding boxes) tracks for multiple people are given as input to our algorithm. We use these tracks and extrapolated full-body bounding boxes, to derive two types of *track constraints* in our framework. We also integrate the corresponding *background* and *non-object* constraints from Section 6.2.4. We combine disparity, appearance and motion cues and evaluate performance on a dataset extracted from 3D movies with challenging scenes and poses.

6.4.2 Baselines

We compare our method to multiple baselines, spanning the whole range of methods from pure semantic segmentation to pure motion segmentation. Some of them are completely unsupervised: *Multi-modal motion segm.* [Ochs *et al.*, 2014], *FG/BG motion segm.* [Papazoglou and Ferrari, 2013]. Some other require pixel-wise supervision to train appearance models: *Pose & segm.* [Seguin *et al.*, 2015] (Ch. 5), *SDS* [Hariharan *et al.*, 2014], *CRF as RNN* [Zheng *et al.*, 2015]. We used the publicly available code and models for all methods.

CRF as RNN [Zheng *et al.*, 2015][2] is the state-of-the-art semantic segmentation method. It uses an end-to-end deep network combining a standard Convolutional Neural Network with a Recurrent Neural Network to perform dense CRF inference. We adapt this method to the task of instance-level segmentation for a given semantic class by assigning each pixel labelled with the said semantic class to the instance which has the closest bounding box. In practice, for people we assign the pixels to the person which spine (derived from the head bounding box) is the closest.

SDS [Hariharan *et al.*, 2014][3] is a simultaneous detection and segmentation method. It classifies region proposals by scoring CNN features extracted from the region and the corresponding bounding box. This method is inherently an instance-level segmentation method, and we evaluate it directly. Note that given the results produced by this method are using a different set of detections (which are an output of the method itself), the performances are not directly comparable with the other reported methods. This baseline is provided for reference as it is the best instance-level segmentation method available.

Pose & segm. [Seguin *et al.*, 2015] (Ch. 5)[4] is the method described in Chapter 5 of this thesis. Given person tracks, it combines pose estimates and disparity cues in an unary term after reasoning on occlusions. A binary term encodes spatio-temporal smoothness using color and motion features.

Multi-modal motion segm. [Ochs *et al.*, 2014][5] separates objects which exhibit different motions. It is a classic method for video segmentation. We adapt it to our problem by assigning the biggest segment (in terms of surface) to be the background segment, and inside each object bounding box we label the largest non-background segment as belonging to the instance.

FG/BG motion segm [Papazoglou and Ferrari, 2013][6] is a pure figure-ground motion segmentation method. We adapt it to the task of instance-level segmentation using the same method as for the first baseline, by splitting the foreground segment in multiple segments.

6.4.3 Results

We evaluate segmentation by computing per-person precision, recall, overlap (defined as the intersection over union between the ground-truth and predicted la-

[2] http://www.robots.ox.ac.uk/~szheng/CRFasRNN.html
[3] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sds/
[4] http://www.di.ens.fr/willow/research/stereoseg/
[5] http://lmb.informatik.uni-freiburg.de/resources/software.php
[6] http://groups.inf.ed.ac.uk/calvin/FastVideoSegmentation/

Tab. 6.1: Comprehensive study of the influence of each component of our method on our dataset. See Section 6.4.3 for comments.

Method	F_1	Precision	Recall	Overlap
Ours + semantic cue	**80.1%**	**81.9%**	**79.6%**	**68.6%**
Ours	78.3%	80.8%	77.3%	66.0%
No temporal smoothness	76.4%	79.2%	75.4%	63.7%
Single frames	76.4%	77.9%	76.4%	63.7%
Grouping term only	77.6%	79.4%	77.2%	65.0%
Discriminative term only	66.9%	70.7%	64.7%	52.1%
No constraint	12.8%	10.4%	40.0%	09.0%
Convex only	75.6%	78.0%	74.1%	62.4%
No disparity	74.0%	77.5%	72.6%	59.9%

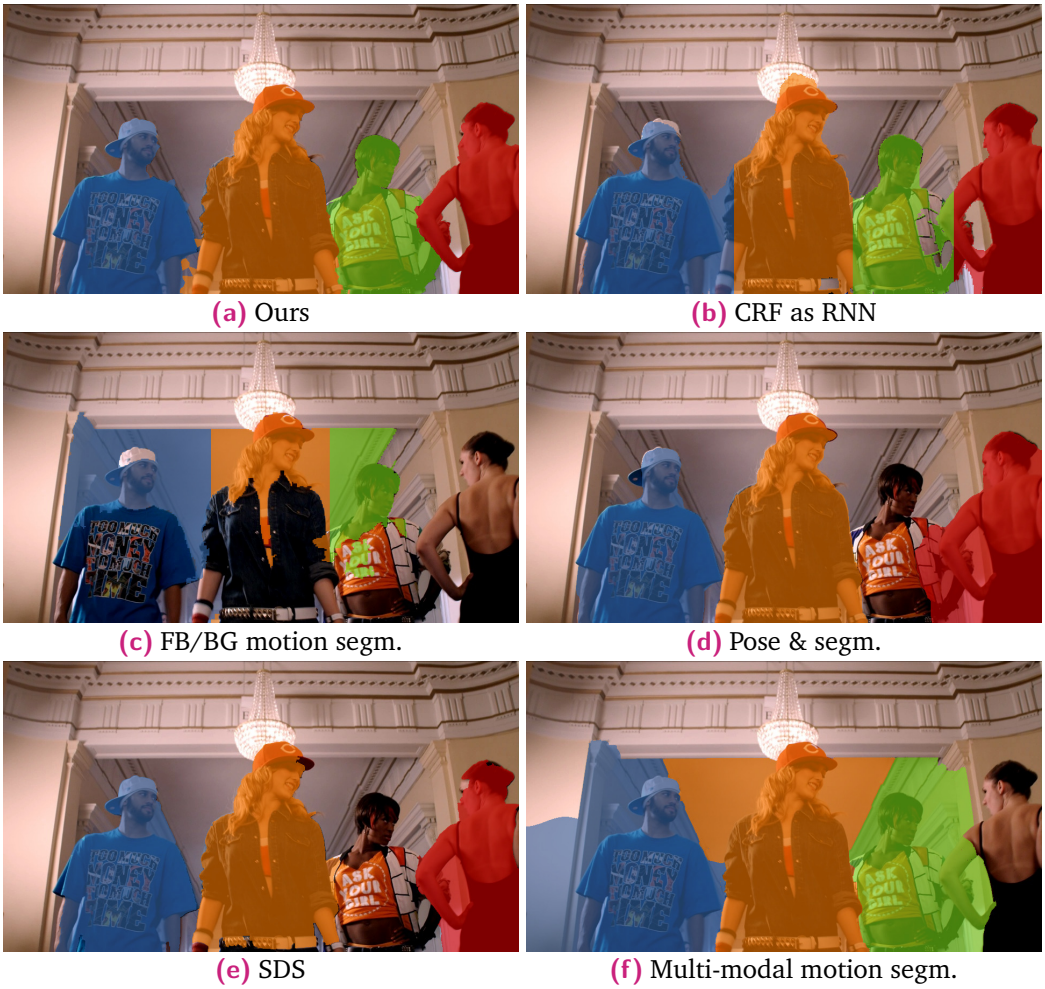
bels [Everingham *et al.*, 2010; Jaccard, 1912]) and F_1 score (the harmonic mean between precision and recall). We report the average of these measures over people and frames. We show qualitative results of our method in Figure 6.6. Video results are also available on `http://www.di.ens.fr/willow/research/instancelevel/`.

Comprehensive analysis. We first analyze each component of our method in Table 6.1. It is interesting to note that similar results are achieved when removing temporal edges from the graph (*No temporal smoothness*), or when processing frames one by one (*Single frames*). Experiments on single frames have a higher recall, while segmenting all frames at once without temporal smoothness produces higher precision, showing the influence of the discriminative term when it has access to the whole video context. Results obtained using the *Grouping term only* are quite good, whereas using the *Discriminative term only* has a lower performance since it only models foreground *vs*. background segmentation without any spatial or temporal consistency. Still, combining the two terms (*Ours*) leads to the best performance as the discriminative term helps to improve precision. Performance is pushed even further when the discriminative term contains strong semantic cues (*Ours + semantic cue*). The non-convex refinement from Section 6.3.3 used in *Full method* produces significantly better performance than using *Convex only* optimization. As discussed in [Bach and Harchaoui, 2007; Bojanowski *et al.*, 2013], using *No constraint* leads to trivial solutions and very poor results. Last, even without disparity features (*No disparity*), which are strong cues, our method produces decent results.

Baselines comparison. Quantitative and qualitative comparisons between our method and baselines are shown in Table 6.2 and Figure 6.7.

The motion segmentation baselines *Multi-modal motion segm.* and *FB/BG motion segm.* perform poorly on this challenging dataset. Both methods completely miss non-moving and almost non-moving people by nature. *Multi-modal motion segm.*

Fig. 6.6: Qualitative results of our method. Note that most of the visually unpleasant artifacts are due to the use of superpixels.



(a) Ours (b) CRF as RNN

(c) FB/BG motion segm. (d) Pose & segm.

(e) SDS (f) Multi-modal motion segm.

Fig. 6.7: Qualitative comparison between our method and the five baselines. Note that *Pose & segm.* may drop detections if the pose estimator fails, and that *SDS* is producing both detection and segmentation, so it uses its own set of detections. See Section 6.4.3 for comments.

also tends to separate the different limbs of a single person into multiple segments.

The *SDS* method performs fairly well. Its detection performance is better than the automatic detector we used (on some key sequences *SDS* detects twice more people than our detector), but it still misses a significant part of person instances. For instance, it misses most heavily occluded persons. The other main downside is that the method mostly provides upper body segmentations (due to either the region proposals or the classifier itself which has been trained on a mix of face, upper body and full body examples), in spite of the refinement procedure which is applied at the end of their method and is meant to provide more complete segmentations.

The *CRF as RNN* method is the best performing baseline. It produces a clean figure-ground segmentation for a given object class. When people are separated in the

Tab. 6.2: Quantitative performance comparison of our method with 5 baselines. Please note that the results from *Ground truth tracks*, *Automatic tracks* and *SDS detections* sections are not comparable as they use different sets of detections. We also report the upperbound of performance which can be achieved given the fact that we use superpixels. See Section 6.4.3 for comments.

Method	F_1	Precision	Recall	Overlap
Ground truth tracks:				
Ours	78.3%	80.8%	77.3%	66.0%
Ours (+ semantic cue)	**80.1%**	81.9%	**79.6%**	**68.6%**
CRF as RNN [Zheng *et al.*, 2015]	78.5%	**83.2%**	77.7%	66.5%
Pose & segm. [Seguin *et al.*, 2015] (Ch. 5)	68.5%	68.3%	76.1%	55.0%
Multi-modal motion segm. [Ochs *et al.*, 2014]	27.4%	41.0%	30.4%	19.4%
FB/BG motion segm. [Papazoglou and Ferrari, 2013]	52.2%	65.1%	49.8%	38.8%
Automatic tracks:				
Ours	63.6%	61.6%	68.6%	52.0%
CRF as RNN [Zheng *et al.*, 2015]	56.2%	58.2%	54.9%	46.5%
Pose & segm. [Seguin *et al.*, 2015] (Ch. 5)	52.7%	57.2%	59.5%	40.8%
Multi-modal motion segm. [Ochs *et al.*, 2014]	27.4%	40.6%	30.4%	19.4%
FB/BG motion segm. [Papazoglou and Ferrari, 2013]	48.4%	57.6%	50.7%	34.9%
SDS detections:				
SDS [Hariharan *et al.*, 2014]	65.1%	73.5%	62.8%	52.6%
Upperbound:				
Superpixels	94.7%	95.4%	94.1%	90.0%

image, our relabelling procedure inherently produces good instance-level segmentation results. However, when the person instances are close by or overlap, our method often outperforms the baseline. Our method, which uses only generic features (color, motion, SIFT) and ad-hoc constraints, still performs as well as this strong baseline. It successfully segments each object instance with only coarse localization cues (encoded in the constraints) and without training a pixel-level appearance model for the segmentation as does the baseline. In addition, when including semantic features in the discriminative term (extracted from the baseline), the performance of our method exceeds the one of the baseline.

Pose & segm., which uses instance-specific pose masks, performs significantly worse than the method proposed here, as it makes strong assumptions about the pose or disparity priors. For instance, it can not recover from errors from the pose estimator. In comparison, our constraints only restrict the space of possible segmentations. They can even be violated in situations which do not satisfy the implicit priors they are enforcing. However, they are strong enough to successfully guide the segmentation even for complicated poses, crowded scenes and cluttered backgrounds.

Per-video quantitative results

We provide per-video performance comparison with the baselines in Figure 6.8 and qualitative results of our method in Figure 6.9. As Figure 6.8 shows, our method handles most situations well, especially crowded scenes (videos 1, 2, 17, 18), although our method does not use a pre-trained person appearance model, as opposed to the *CRF as RNN* baseline. The two videos where our method performs less good compared to strongly-supervised baselines are #13 and #10. In #13 one person violates our assumption that most of the body pixels should be located under person's head. In #10 the body bounding box derived from the person head box is too small and our constraints are too weak to get a high recall.

Running times Our optimization procedure takes about 6 hours on a video of 200 frames with about 2000 superpixels per frame and 4 labels (3 object tracks plus the background). A better superpixels algorithm would allow having much less superpixels and heavily speeding up the optimization procedure. Comparatively, the method of [Seguin *et al.*, 2015] (Ch. 5) can only process blocks of 40 frames at a time at a resolution of 960x540, and takes a total of 15 hours to process the same total amount of frames.

6.4.4 Correlation between cost and performance

One important matter with segmentation algorithms is that they typically optimize a cost which is not directly related to segmentation performance. We therefore studied the correlation between our cost function and segmentation performance. Figure 6.10 depicts that, in most cases, lowering duality gap (which is the best optimization certificate we have when using the Frank-Wolfe algorithm) leads to a higher segmentation performance.

6.5 Incorporating pose cues in a weak manner for multiple person segmentation

The model proposed in Chapter 5 introduced instance-specific person pose masks, derived from pose estimates. While these strong pixel-wise cues could lead to segmentation mistakes when the estimated pose was incorrect, our framework can easily leverage them as weaker cues, by weighing the contribution of each pixel to the *track constraint* by the likelihood of this pixel to contain a pose.

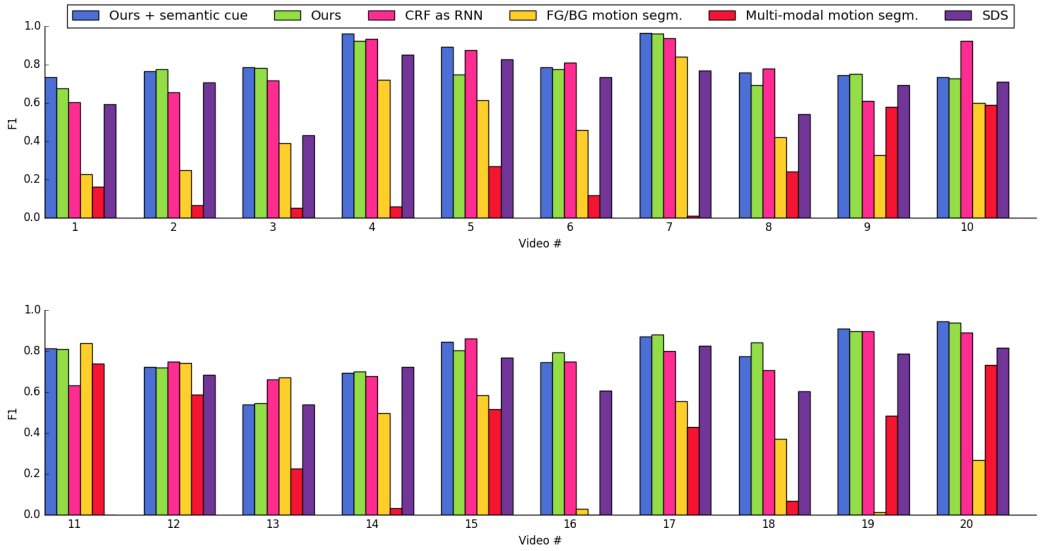


Fig. 6.8: Per-video quantitative results of our instance-level segmentation method on the Inria 3DMovie Dataset v2 (Section 4.2.2).

Fig. 6.9: Per-video qualitative results of our method (*Ours*)

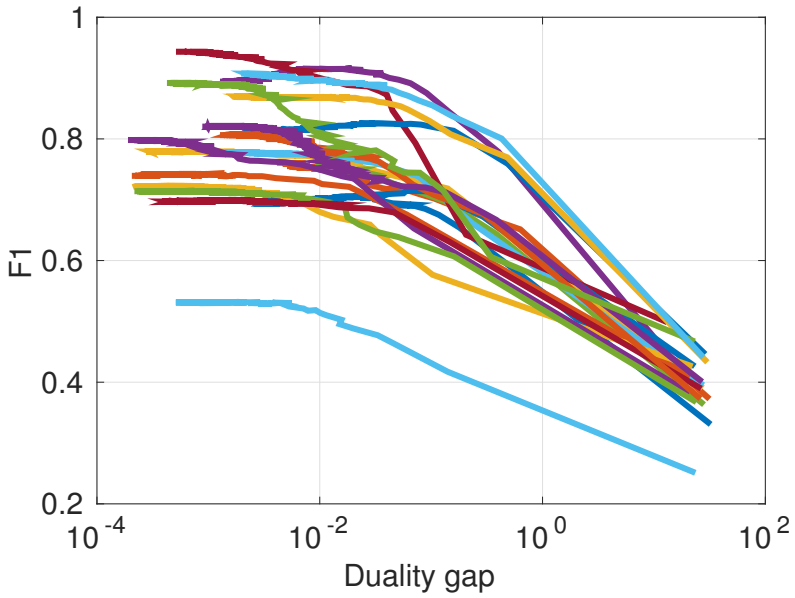


Fig. 6.10: Evaluation of F_1 score variation (y axis) w.r.t. duality gap (x axis). Each line corresponds to the convex optimization procedure for one of the 20 clips of the multi-person segmentation experiments. In most cases, the lower the duality gap is, the higher the segmentation performance is.

Using the articulated pose masks and notations from Section 5.3.3, we denote by $\psi_p(\Theta, i)$ the likelihood for a pixel i to belong to a person associated to a pose Θ. In the case of multi-person segmentation, for a given person $k \in \{1, \dots, K\}$ with associated pose Θ^k in a frame, we denote by ω the vector which n-th entry contains the average of $\psi_p(\Theta^k, \cdot)$ over the pixels of the n-th superpixel. We translate and rescale ω into $\tilde{\omega}$ so that its entries are are between 0 and 1:

$$\tilde{\omega}_i = \frac{\omega_i - \min \omega}{\max \omega - \min \omega} \tag{6.11}$$

and then compute Ω, which encodes the trade-off between the pose prior and uniform prior over the entire bounding box:

$$\Omega = \zeta \; I_N + (1 - \zeta) \text{Diag}(\tilde{\omega}). \tag{6.12}$$

with ζ a weight controlling the influence of the pose prior. We can then adapt the track constraints to incorporate this information by adjusting the form of the corresponding inequality constraints defined in Equation 6.6:

$$0 \geq \sigma \left(\frac{1}{\mathbb{1}_R^T \, \Omega \, s} \, \mathbb{1}_R^T \, \Omega \, \text{Diag}(s) \; y \; \mathbf{e}_k - \rho \right) - \xi_c. \tag{6.13}$$

By properly adjusting the influence of the pose on the contribution of each superpixel to the linear constraint through the weight ζ, we can both benefit from a

strong class-specific and instance-specific cue and avoid mistakes. Figure 6.11 illustrates the contribution of each superpixel to the constraint for various ζ values.

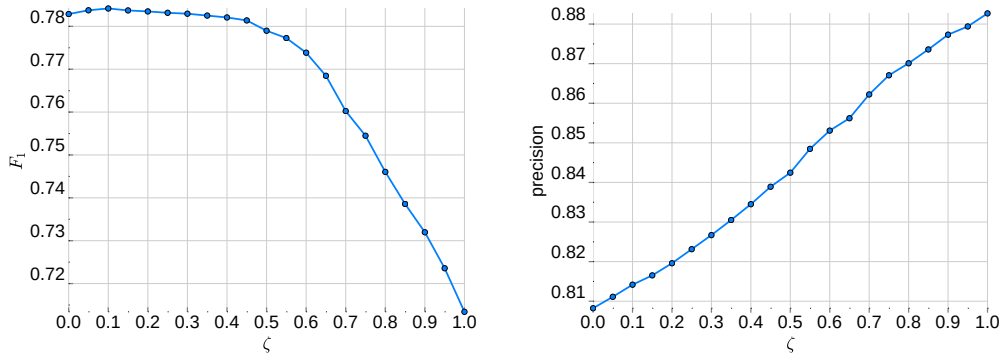
(a) Pose mask **(b)** $\zeta = 0$ **(c)** $\zeta = 0.5$ **(d)** $\zeta = 1$

Fig. 6.11: Influence of pose masks (a) on track constraints for various ζ values. When not using the pose masks ($\zeta = 0$), the contribution of each superpixel is solely based on the size of the superpixel (b). When incorporating pose cues, the superpixels inside the pose mask contribute more to the linear constraint than the ones outside (c) up to a point where only the ones inside have a contribution (d). The colormap used is the standard "jet" colormap, where low values are drawn in blue and high values are drawn in red.

To evaluate this approach, we ran the modified method with various ζ parameters, and report the variation of the performance measures according to ζ in Figure 6.12, as well as a qualitative comparison between results of the pose-agnostic method (corresponding to $\zeta = 0$) and the modified method with the best ζ ($\zeta = 0.1$), $\zeta = 0.5$ and $\zeta = 1$ in Figure 6.13.

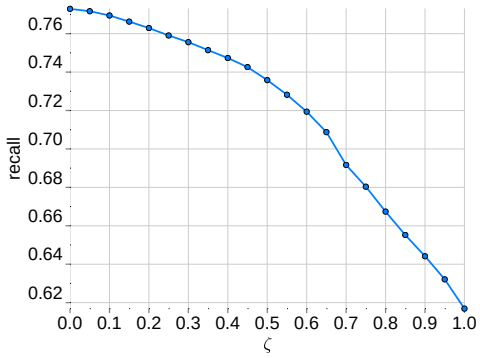
Overall, weighting the constraints using pose masks helps the precision of our method, but quickly becomes harmful to the recall of our method as ζ increases. The best quantitative trade-off is found at $\zeta = 0.1$, where the pose cues help reducing leaks but are not strong enough yet to mislead the method when the estimated pose is wrong. Qualitatively though, results are more visually pleasing at $\zeta = 0.5$, where most obvious leaks are avoided (such as leaks between legs, see columns 2 and 3 of Figure 6.13).

In terms of flexibility, we note that we can vary ζ between 0 and 0.5 and obtain stable F_1 and overlap scores. However, precision increases as ζ increases, while recall decreases. We can thus explicitly choose where we set the trade-off between precision and recall at constant F_1 performance by varying the contribution of pose information to weighting of our track constraints.

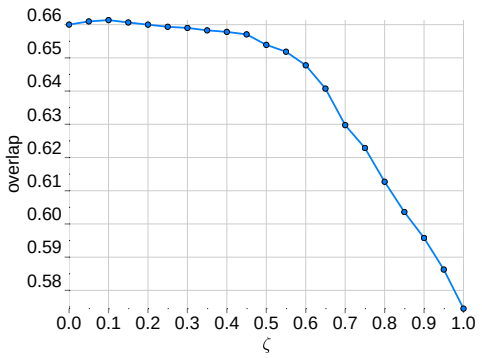


(a) F_1

(b) Precision



(c) Recall



(d) Overlap

Fig. 6.12: Quantitative evaluation of the influence of weighting the constraints based on pose masks.

6.6 Handling of other object classes

The major strength of our method is that it is mostly agnostic to the underlying object class. We provide the method with a single floating point parameter specifying which amount of each bounding box is expected to belong to the object. With this single parameter, the video input and the corresponding bounding box tracks, our method is able to properly segment the object instance from the background of the video and from the other object instances. To the best of our knowledge, there is no proper complete dataset for instance-level segmentation in videos for the moment. To show that our method can handle non-person object classes, we ran it on two videos with multiple object instances from the popular SegTrack v2 dataset [Li *et al.*, 2013]. We show two sample frames in Figure 6.14. In this section, we adapt our method to the segmentation propagation task and run it on the whole SegTrack dataset [Tsai *et al.*, 2010] using a single set of hyperparameters. We show that our method is also able to handle a variety of object classes and situations without changing the hyperparameters for this task.

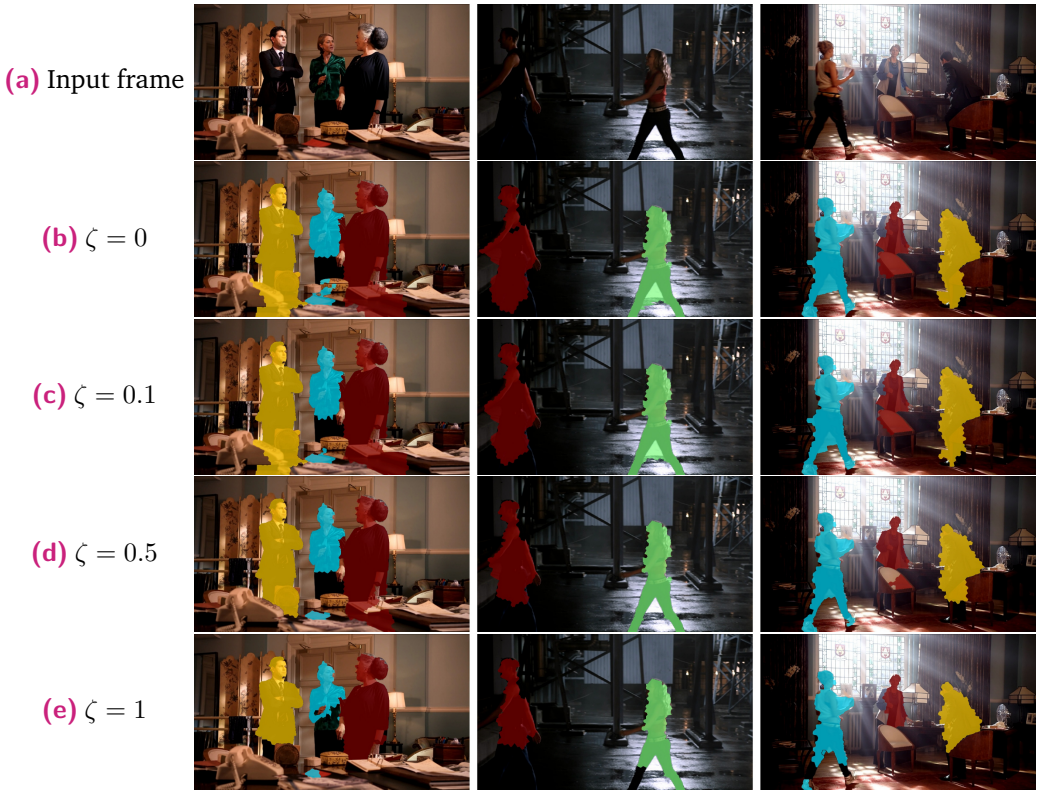


Fig. 6.13: Qualitative evaluation of the influence of weighting the constraints based on pose masks. Pose masks help reducing leaking by providing stronger anchors, however they also reduce the flexibility of the method and tend to lower the achieved recall.

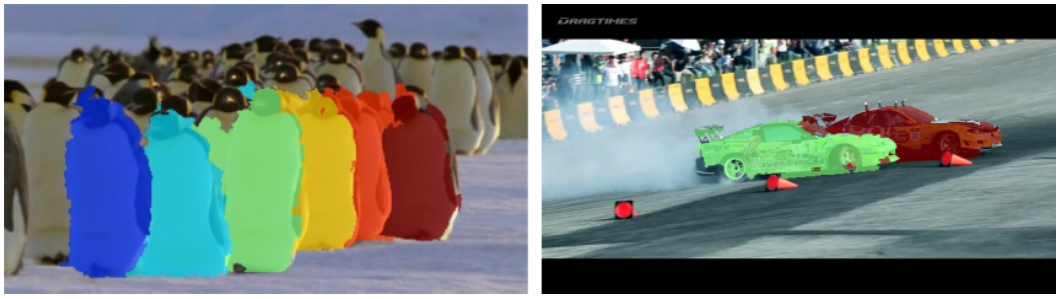


Fig. 6.14: Results of our method applied to two multi-instance videos from SegTrack v2 [Li *et al.*, 2013].

Segmentation propagation

We evaluate the performance of our approach for segmentation propagation on the SegTrack dataset [Tsai *et al.*, 2010][7]. In this setup, we are given a video and ground-truth segmentation for the first frame of the video, which we incorporate as *ground-truth supervision constraints* (Sec. 6.2.4). In addition, we derive a bounding box for each object using its segmentation label in the first frame. We then use

[7] http://cpl.cc.gatech.edu/projects/SegTrack/

an off-the-shelf tracker [Danelljan *et al.*, 2014][8] to track the object bounding box through the clip. We incorporate this information in our model as *track constraints*. We also evaluate our model with manually annotated ground-truth bounding boxes to get an upper bound if tracking was perfect. We then segment all the clips using the same set of hyperparameters.

Evaluation protocol. The standard metric used for evaluation on this dataset is the mean pixel error (MPE), which is the average number of erroneous pixels (false positives or false negatives) for each object instance for each frame.

Baselines. We compare our approach with two other methods: the first one is the full segmentation method of [Fathi *et al.*, 2011] which does not use the initial ground-truth segmentation for the first frame. Instead, it iteratively segments the video by learning appearance models from the most reliably segmented frame at the current iteration, using a measure of uncertainty. The second baseline is the segmentation propagation method of [Jain and Grauman, 2014], which uses both binary potentials and higher-order potentials based on supervoxels to propagate the segmentation.

Results. A quantitative comparison is provided in Table 6.3. Qualitative results are shown in Figure 6.15 and on the project website[9]. Our method achieves state-of-the-art performance using given ground-truth bounding box tracks, and competitive performance using automatically computed tracks, compared to [Fathi *et al.*, 2011] and [Jain and Grauman, 2014]. When motion is slow, such as for the *penguin* sequence, or when there is little deformation, as for the *birdfall* sequence, pure propagation without extra supervision produces satisfactory results. In other situations, accurate object localization is essential for our method to provide good segmentation. For instance, in sequences with fast movement and heavy deformation (*cheetah*, *monkeydog*), automatic tracks fail and the other constraints are too weak to propagate boundary-accurate segmentations. Note that we use the same set of parameters for all sequences, which indicates a good generalization of our method over different object classes, while most methods used on this dataset usually use a different set of parameter for each video to handle the different types of situations (e.g.fast *vs*. slow movement).

[8] https://github.com/gnebehay/DSST
[9] http://www.di.ens.fr/willow/research/instancelevel

Tab. 6.3: Segmentation results on the SegTrack dataset [Tsai *et al.*, 2010]. The metric is mean pixel error (lower is better). We report result for three variants of our method, as well as for two state-of-the-art methods [Fathi *et al.*, 2011; Jain and Grauman, 2014]. We test our framework with just the segmentation propagation constraints (column *No BB*) and when adding object tracks, either automatic (*BB tracks*) or ground-truth (*GT BBs*). In each row we highlight the best result (in green), the 2nd best (in yellow) and the 3rd best (in blue).

Clip	No BB	BB tracks	GT BBs	[Jain and Grauman, 2014]	[Fathi *et al.*, 2011]
birdfall	221	**169**	**168**	**189**	342
cheetah	2196	1305	**724**	**1170**	**711**
girl	2733	**1606**	**1602**	2883	**1206**
monkeydog	2405	1021	**658**	**333**	**598**
parachute	305	**251**	278	**228**	**251**
penguin	**787**	848	**830**	**443**	1367

Legend: **Best** **2nd best** **3rd best**

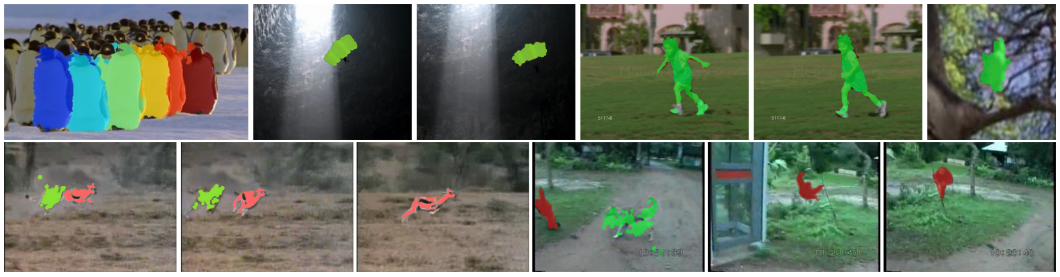


Fig. 6.15: Results on the SegTrack dataset [Tsai *et al.*, 2010], using automatic object tracks (top) or ground-truth tracks (bottom). In the first example of the top row, we notice minor leaking above the objects, as the discriminative model struggles to differentiate them from the rest of the herd. The other results on this row are visually excellent. When using ground-truth tracks, the fast moving cheetah and antelope on the left are mostly recovered. On the right, segmentation is very good for the monkey (in red), while the dog (in green), being of the same color as the road, is mostly missed. Note that these pair of animals are from different species, so we learn a background *vs*. animal model.

6.7 Discussion

We have presented a flexible and effective framework for multi-instance object segmentation. We have demonstrated its experimental performance on a challenging dataset, showing that constraining the space of segmentations is a robust way to incorporate object tracks information.

7 Conclusion and perspectives

In this chapter, we summarize the contributions of this thesis and discuss avenues for future research.

7.1 Contributions

We have shown that 3D movies can be successfully exploited for person analysis, and that it was possible to exploit the relative ease of certain tasks in 3D movies, such as person detection, to train better models for standard color data. We have also visited the problem of segmenting multiple object instances, possibly of the same object class, and in particular multiple persons, in videos. We have investigated two different approaches, either with strong semantic cues or with weak localization cues.

In this thesis:

- We have studied how to extract disparity maps from the uncalibrated, unrectified pairs of stereoscopic streams provided by 3D movies in Chapter 4. After investigating standard stereovision techniques, we resorted to using optical flow methods that can match pixels of the two views in challenging non-calibrated and non-rectified situations.
- In Section 4.3, we have studied how using disparity cues could improve methods based on deformable part models for person detection and pose estimation. We have shown that person detection models that jointly consider appearance and disparity significantly outperformed models which only consider appearance. For pose estimation, we have shown a similar trend, albeit with a smaller improvement.
- The person detector trained jointly on appearance and disparity cues features an interesting high-precision mode. In Section 4.4, we have leveraged this property to perform a depth-supervised training of a person detector for standard color movies. We harvest detections using the joint appearance and disparity-based detector in a high-precision mode, filter out the ones which were already well detected by an initial appearance-based detector, and train an improved appearance-based detector using the collected examples.

- In Chapter 5, we have proposed a method to segment multiple persons in 3D videos using a CRF which encodes disparity cues, pose estimates as well as classical color and motion cues. Using pixel-wise annotations for a subset of our training set for pose estimation, we learn soft segmentation masks for each part mixture of the pose estimation model. Using these masks, we are able to produce soft segmentation outputs which are pose-specific. We then analyze the disparity maps to build a model of each person in the disparity space, from which we derive a likelihood map for each person, which we combine with the corresponding pose-specific mask. We then reason about inter-person occlusions to produce maps which are used as the unary terms of the CRF model. The binary terms are derived from classical smoothness priors based on color and motion information in space and time.
- In Chapter 6, we have proposed an algorithm to segment multiple object instances, and in particular multiple persons, in videos with given object tracks. We formulate this problem as a convex optimization one over the space of segmentations, with a quadratic objective function combining a discriminative term to encourage a long-term coherence and a spectral clustering term to ensure local space-time consistency. We cast the object bounding boxes as linear constraints which efficiently guide the optimization of the segmentation problem in a weak manner, by shaping the space of admissible segmentations.
- To properly train our models and evaluate our work, we have collected two new datasets for person detection, pose estimation and multi-person segmentation in 3D movies. These datasets contain a total of 1158 annotated person bounding boxes, 587 annotated poses and 1318 person segmentation masks. In particular, the multi-person segmentation part of our datasets provides the segmentation mask of each person in a given frame, which makes it valuable for instance-level segmentation experiments.

7.2 Perspectives

In this section, we discuss possible directions for future research. In Section 7.2.1, we propose several avenues of research to extend our work on person detection and human pose estimation from Chapter 4. Section 7.2.2 lists two possible improvements for our multi-person segmentation method using pose cues from Chapter 5. Last, we propose two technical improvements and two future directions for the multi-instance segmentation method of Chapter 6 in Section 7.2.3.

7.2.1 Person detection and pose estimation

Iterated training for the depth-supervised person-detector. The depth-supervised training procedure for person detectors proposed in Section 4.4 is currently composed of four steps:

1. training the initial detectors from the manually labelled dataset,
2. harvesting examples using the joint appearance and disparity detector,
3. filtering the harvested examples to retain the ones which were not detected by the appearance-based detector
4. retraining the appearance-based detector with the collected examples.

A downside of this method is that among the examples we collect, many could be redundant, which means the final training set may be much bigger than necessary, unnecessarily increasing the subsequent training time. To overcome this, an iterated training scheme could be adopted, where the appearance-based detector would be retrained regularly during the filtering step to incorporate the new examples which have already been selected. It would then reduce the number of redundant examples selected while filtering the remainder of the set of harvested examples. An alternative approach would be the one of self-paced learning [Kumar *et al.*, 2010], where increasingly large subsets of the training set are presented to the learning procedure based on a notion of easiness of each example. For instance, this notion of easiness could be derived from both the score of the initial or current appearance-based detector and of the score of the initial detector trained jointly on appearance and disparity. The first examples presented to the learning procedure could be the ones which have the lowest initial appearance-based score, to start with the hardest examples, or the other way around to strictly follow the typical self-paced learning intuition. This approach would allow the algorithm to progressively incorporate the harvested examples.

Appearance models based on convolutional neural networks. The person detection and pose estimation experiments we have performed are based on deformable part models. Since then, convolutional neural networks have shown excellent performances on these tasks. For instance, the Faster R-CNN method [Ren *et al.*, 2015] is an efficient and powerful object detection method. Networks as the one of [Chen and Yuille, 2014a] can also be trained to learn part detectors and spatial relationships between parts, which can then be used in a graphical model to perform human pose estimation. These methods could be adapted to use additional disparity cues, which could lead to interesting performance improvements. In particular, our depth-supervised training pipeline could be very relevant, as it can produce a very

large number of high quality training samples which could feed the data-hungry training procedures of very deep neural networks.

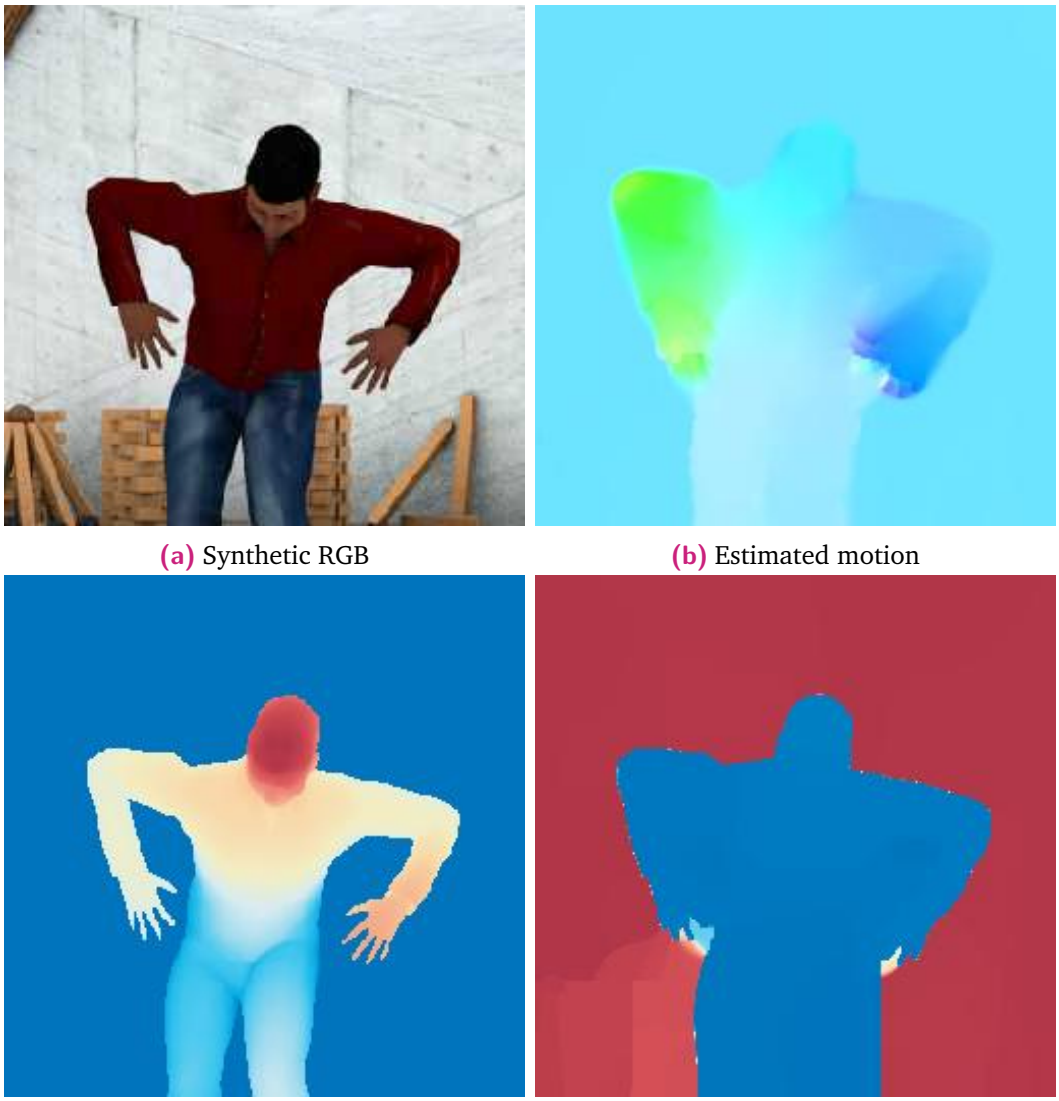
Training from synthetic data. During this PhD, we have also studied (results not shown in the thesis) the use of synthetic (computer-generated) data for pose estimation, as synthetic data provides a potentially unlimited number of examples. Using motion-capture datasets such as the CMU Mocap Dataset[1] and the Human 3.6M dataset [Ionescu *et al.*, 2014] and softwares such as Blender or Autodesk MotionBuilder, we have rendered millions of synthetic stereo pairs and depth ground truth for a varied set of poses, viewpoints and characters. We have tried to exploit this very large set of synthetic examples using multiple approaches: training deformable part models, learning classification neural networks, learning regression neural networks. We have investigated the various channels of information at hand, illustrated in Figure 7.1: ground truth depth, estimated disparity between the stereo pairs, estimated motion field, synthetic color images.

However, so far, we have only observed limited improvements, especially when it comes to transferring to non-synthetic data. We believe the estimated disparity data from real 3D movies may not be detailed enough to be able to recover fine pose information, which may be overcome by improving disparity estimation. Next, our synthesis procedure could be improved by adding more clothing and background scene variations and by rendering scenes with multiple persons interacting and occluding each other, as inter-person occlusions are a typical challenge of realistic scenarios and in particular of the dataset we used for evaluating this task on non-synthetic data.

7.2.2 Multi-person segmentation with pose cues

Principled optimization of the multi-person layering model. In Chapter 5, the choice of ordering of the persons in each frame based on the selection of disparity parameters is performed using a brute force search over a limited set of possible disparity parameters. To further improve the quality of this component, the disparity parameters could be selected using a more principled optimization method which could also search over entire intervals of disparity parameters. This may look hard at first sight, given that the disparity parameters are also involved in the occlusion model, but could be done by first enumerating the set of possible ordering, optimizing the possible disparity parameters for each ordering and then taking the global optimum over all orderings. This would lead to a finer selection of disparity parameters, which in turn could yield segmentation performance improvements.

[1] `http://mocap.cs.cmu.edu/`



(a) Synthetic RGB

(b) Estimated motion

(c) Synthetic depth

(d) Estimated disparity

Fig. 7.1: Example of synthesized upper body pose example. Stereoscopic pairs of sequences of synthetic RGB frames (a) are generated using Blender, as well as a sequence of depth ground truth (c). Motion (b) is estimated from one frame of the left view to the next, while disparity (d) is estimated from a stereoscopic pair.

Joint optimization of poses and segmentation. While our formulation from Equation 5.2 is posed as a joint optimization on the space of poses and the space of segmentations, we currently first select the best poses and then optimize the segmentation, as shown in Equation 5.3. Given a set of candidate poses for each person detection, the segmentation we obtain could be used to select a better set of poses, which in turn may lead to a better segmentation. Such an alternated optimization scheme could be explored to perform the joint optimization of the initial problem of Equation 5.2.

7.2.3 Multiple-instance segmentation under weak constraints

Speed-up optimization by working with supervoxels. The multi-instance segmentation method designed in Chapter 6 has been designed to work on tracked superpixels. With minor changes, the framework we proposed could be adapted to work with supervoxels, which could lead to large speedups. Indeed, on average on the Inria 3D Movie v2 dataset, there are about 20 times less supervoxels than superpixels on average, and up to 100 times less on some sequences. Optimizing on supervoxels could thus lead to significant speedups.

Qualitative improvements by using other superpixel methods. Similarly, using other superpixel methods which feature a stronger capability to stick to the image edges such as SLIC superpixels [Achanta *et al.*, 2012] could lead to cleaner results.

Multiple object classes. Adapting this method to handle multiple instances of multiple object classes in a single sequence would be a straightforward but valuable improvement to the method.

Part segmentation. However, we believe the most interesting next step would be to adapt our method to perform per-part segmentation of multiple people. Instead of having a single label per person, multiple labels would map to the different limbs of the person, and multiple linear constraints would be added to guide the optimization, such as one area constraint per limb, or constraints which encode the expected neighborhood of the different limbs.

Bibliography

[Achanta *et al.*, 2012] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk, 2012. „SLIC superpixels compared to state-of-the-art superpixel methods.“ In: *Trans. PAMI* 34.11, pp. 2274–2282 (cit. on p. 140).

[Agarwal and Triggs, 2004] Ankur Agarwal and Bill Triggs, 2004. „3D human pose from silhouettes by relevance vector regression.“ In: *Proc. CVPR* (cit. on p. 23).

[Agarwal *et al.*, 2011] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski, 2011. „Building Rome in a day.“ In: *Communications of the ACM* 54.10, pp. 105–112 (cit. on pp. 17, 18).

[Allende *et al.*, 2013] Héctor Allende, Emanuele Frandi, Ricardo Ñanculef, and Claudio Sartori, 2013. „Novel Frank-Wolfe Methods for SVM Learning.“ In: *arXiv* (cit. on p. 116).

[Arbelaez *et al.*, 2011] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, 2011. „Contour Detection and Hierarchical Image Segmentation.“ In: *Trans. PAMI* 33.5, pp. 898–916 (cit. on p. 89).

[Arbeláez *et al.*, 2014] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan Barron, Ferran Marques, and Jitendra Malik, 2014. „Multiscale combinatorial grouping.“ In: *Proc. CVPR* (cit. on p. 30).

[Ayvaci *et al.*, 2012] Alper Ayvaci, Michalis Raptis, and Stefano Soatto, 2012. „Sparse Occlusion Detection with Optical Flow.“ In: *IJCV* 97.3, pp. 322–338 (cit. on pp. 63–66, 120).

[Bach and Harchaoui, 2007] Francis Bach and Zaïd Harchaoui, 2007. „DIFFRAC: a discriminative and flexible framework for clustering.“ In: *Adv. NIPS* (cit. on pp. 33, 53, 54, 106, 108, 110, 122).

[Banica *et al.*, 2013] Dan Banica, Alexandru Agape, Adrian Ion, and Cristian Sminchisescu, 2013. „Video object segmentation by salient segment chain composition.“ In: *Proc. ICCV Workshops* (cit. on p. 28).

[Banica and Sminchisescu, 2015] Dan Banica and Cristian Sminchisescu, 2015. „Second-Order Constrained Parametric Proposals and Sequential Search-Based Structured Prediction for Semantic Segmentation in RGB-D Images.“ In: *Proc. CVPR* (cit. on p. 20).

[Bertsekas, 1999] Dimitri Bertsekas, 1999. *Nonlinear Programming*. Athena Scientific (cit. on p. 117).

[Blake and Zisserman, 1987] Andrew Blake and Andrew Zisserman, 1987. *Visual reconstruction*. MIT press Cambridge (cit. on p. 117).

[Bojanowski *et al.*, 2013] Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic, 2013. „Finding Actors and Actions in Movies.“ In: *Proc. ICCV* (cit. on pp. 33, 106, 110, 122).

[Bojanowski *et al.*, 2014] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic, 2014. „Weakly Supervised Action Labeling in Videos Under Ordering Constraints.“ In: *Proc. ECCV* (cit. on pp. 33, 115).

[Boros and Hammer, 2002] Endre Boros and Peter L Hammer, 2002. „Pseudo-Boolean optimization.“ In: *Discrete Applied Mathematics* (cit. on p. 93).

[Bourdev *et al.*, 2010] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik, 2010. „Detecting people using mutually consistent poselet activations.“ In: *Proc. ECCV* (cit. on p. 21).

[Bourdev and Malik, 2009] Lubomir Bourdev and Jitendra Malik, 2009. „Poselets: Body part detectors trained using 3d human pose annotations.“ In: *Proc. ICCV* (cit. on p. 21).

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe, 2004. *Convex optimization*. Cambridge university press (cit. on p. 115).

[Boykov and Jolly, 2001] Yuri Boykov and Marie-Pierre Jolly, 2001. „Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images.“ In: *Proc. ICCV* (cit. on pp. 89, 95).

[Boykov *et al.*, 2001] Yuri Boykov, Olga Veksler, and Ramin Zabih, 2001. „Fast Approximate Energy Minimization via Graph Cuts.“ In: *Trans. PAMI* 23.11, pp. 1222–1239 (cit. on pp. 48, 93).

[Budvytis *et al.*, 2011] Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla, 2011. „Semi-Supervised Video Segmentation.“ In: *Proc. CVPR*, pp. 2257–2264 (cit. on p. 97).

[Carreira *et al.*, 2012] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu, 2012. „Semantic segmentation with second-order pooling.“ In: *Proc. ECCV* (cit. on p. 27).

[Carreira and Sminchisescu, 2012] Joao Carreira and Cristian Sminchisescu, 2012. „Cpmc: Automatic object segmentation using constrained parametric min-cuts.“ In: *Trans. PAMI* 34.7, pp. 1312–1328 (cit. on p. 26).

[Chang *et al.*, 2013] Jason Chang, Donglai Wei, and John W Fisher III, 2013. „A video representation using temporal superpixels.“ In: *Proc. CVPR* (cit. on p. 119).

[Chen and Corso, 2011] Albert YC Chen and Jason J Corso, 2011. „Temporally consistent multi-class video-object segmentation with the video graph-shifts algorithm.“ In: *Proc. WACV* (cit. on p. 26).

[Chen *et al.*, 2008] Albert YC Chen, Jason J Corso, and Le Wang, 2008. „HOPS: Efficient region labeling using higher order proxy neighborhoods.“ In: *Proc. ICPR* (cit. on p. 26).

[Chen *et al.*, 2015] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, 2015. „Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs.“ In: *Proc. ICLR* (cit. on p. 27).

[Chen and Yuille, 2014a] Xianjie Chen and Alan Yuille, 2014a. „Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations.“ In: *Adv. NIPS* (cit. on p. 137).

[Chen and Yuille, 2014b] Xianjie Chen and Alan L Yuille, 2014b. „Articulated pose estimation by a graphical model with image dependent pairwise relations.“ In: *Adv. NIPS* (cit. on p. 24).

[Cherian *et al.*, 2014] Anoop Cherian, Julien Mairal, Karteek Alahari, and Cordelia Schmid, 2014. „Mixing Body-Part Sequences for Human Pose Estimation.“ In: *Proc. CVPR* (cit. on p. 24).

[Chung, 1997] Fan RK Chung, 1997. *Spectral graph theory*. Vol. 92. American Mathematical Soc. (cit. on p. 49).

[Collet *et al.*, 2011] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa, 2011. „The MOPED framework: Object recognition and pose estimation for manipulation.“ In: *IJRR* 30.10, pp. 1284–1306 (cit. on p. 19).

[Colombari *et al.*, 2007] Andrea Colombari, Andrea Fusiello, and Vittorio Murino, 2007. „Segmentation and tracking of multiple video objects.“ In: *Pattern Recognition* 40.4, pp. 1307–1317 (cit. on p. 28).

[Corso *et al.*, 2008] Jason J Corso, Alan Yuille, and Zhuowen Tu, 2008. „Graph-shifts: Natural image labeling by dynamic hierarchical computing.“ In: *Proc. CVPR* (cit. on p. 26).

[Dai *et al.*, 2015] Jifeng Dai, Kaiming He, and Jian Sun, 2015. „Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation.“ In: *Proc. ICCV* (cit. on p. 31).

[Dalal and Triggs, 2005] Navneet Dalal and Bill Triggs, 2005. „Histograms of Oriented Gradients for Human Detection.“ In: *Proc. CVPR* (cit. on pp. 21, 37).

[Danelljan *et al.*, 2014] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg, 2014. „Accurate scale estimation for robust visual tracking.“ In: *Proc. BMVC* (cit. on pp. 106, 132).

[Desai and Ramanan, 2012] Chaitanya Desai and Deva Ramanan, 2012. „Detecting actions, poses, and objects with relational phraselets.“ In: *Proc. ECCV*, pp. 158–172 (cit. on p. 90).

[Deutscher *et al.*, 2000] Jonathan Deutscher, Andrew Blake, and Ian Reid, 2000. „Articulated body motion capture by annealed particle filtering.“ In: *Proc. CVPR* (cit. on p. 23).

[Edmonds and Karp, 1972] Jack Edmonds and Richard M Karp, 1972. „Theoretical improvements in algorithmic efficiency for network flow problems.“ In: *JACM* 19.2, pp. 248–264 (cit. on p. 52).

[Eichner and Ferrari, 2010] Marcin Eichner and Vittorio Ferrari, 2010. „We are family: Joint pose estimation of multiple persons.“ In: *Proc. ECCV* (cit. on p. 29).

[Eichner *et al.*, 2012] Marcin Eichner, Manuel Marin-Jimenez, Andrew Zisserman, and Vittorio Ferrari, 2012. „2D Articulated Human Pose Estimation and Retrieval in (Almost) Unconstrained Still Images." In: *IJCV* 99.2, pp. 190–214 (cit. on pp. 29, 70, 97).

[Elgammal and Lee, 2004] Ahmed Elgammal and Chan-Su Lee, 2004. „Inferring 3D body pose from silhouettes using activity manifold learning." In: *Proc. CVPR* (cit. on p. 23).

[Everingham *et al.*, 2006] Mark Everingham, Josef Sivic, and Andrew Zisserman, 2006. „"Hello! My name is... Buffy" – Automatic Naming of Characters in TV Video." In: *Proc. BMVC* (cit. on pp. 90, 120).

[Everingham *et al.*, 2010] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, 2010. „The pascal visual object classes (voc) challenge." In: *IJCV* (cit. on p. 122).

[Everingham *et al.*, 2011] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, 2011. *The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results*. http://host.robots.ox.ac.uk/pascal/VOC/voc2011/workshop/ (cit. on pp. 70, 83, 97).

[Fan *et al.*, 2015] Xiaochuan Fan, Kang Zheng, Yuewei Lin, and Song Wang, 2015. „Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation." In: *Proc. CVPR* (cit. on p. 25).

[Farabet *et al.*, 2013] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun, 2013. „Learning hierarchical features for scene labeling." In: *Trans. PAMI* 35.8, pp. 1915–1929 (cit. on p. 27).

[Fathi *et al.*, 2011] Alireza Fathi, Maria-Florina Balcan, Xiaofeng Ren, and James M Rehg, 2011. „Combining Self Training and Active Learning for Video Segmentation." In: *Proc. BMVC* (cit. on pp. 28, 132, 133).

[Felzenszwalb, 2001] Pedro F Felzenszwalb, 2001. „Learning models for object recognition." In: *Proc. CVPR* (cit. on p. 21).

[Felzenszwalb *et al.*, 2010] Pedro F. Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan, 2010. „Object Detection with Discriminatively Trained Part Based Models." In: *Trans. PAMI* 32.9, pp. 1627–1645 (cit. on pp. 22, 36, 68, 70, 71, 80).

[Felzenszwalb *et al.*, 2013] Pedro F. Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan, 2013. „Visual object detection with deformable part models." In: *Communications of the ACM* 56.9, pp. 97–105 (cit. on pp. 36, 39, 40).

[Felzenszwalb and Huttenlocher, 2004a] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, 2004a. *Distance transforms of sampled functions*. Tech. rep. Cornell University (cit. on pp. 22, 38).

[Felzenszwalb and Huttenlocher, 2004b] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, 2004b. „Efficient belief propagation for early vision." In: *Proc. CVPR* (cit. on p. 60).

[Felzenszwalb and Huttenlocher, 2005] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, 2005. „Pictorial Structures for Object Recognition." In: *IJCV* 61.1, pp. 55–79 (cit. on pp. 22, 35).

[Ferrari *et al.*, 2008] Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman, 2008. „Progressive search space reduction for human pose estimation.“ In: *Proc. CVPR* (cit. on p. 24).

[Fischler and Bolles, 1981] Martin A Fischler and Robert C Bolles, 1981. „Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.“ In: *Communications of the ACM* 24.6, pp. 381–395 (cit. on p. 60).

[Fischler and Elschlager, 1973] Martin A Fischler and Robert A Elschlager, 1973. „The representation and matching of pictorial structures.“ In: *Transactions on Computers* 1, pp. 67–92 (cit. on p. 21).

[Fragkiadaki *et al.*, 2013] Katerina Fragkiadaki, Han Hu, and Jianbo Shi, 2013. „Pose from Flow and Flow from Pose.“ In: *Proc. CVPR* (cit. on p. 24).

[Frank and Wolfe, 1956] Marguerite Frank and Philip Wolfe, 1956. „An algorithm for quadratic programming.“ In: *Naval Research Logistics Quarterly* (cit. on pp. 55, 107, 115).

[Fusiello *et al.*, 1999] Andrea Fusiello, Emanuele Trucco, Alessandro Verri, and Ro Verri, 1999. *A Compact Algorithm for Rectification of Stereo Pairs* (cit. on pp. 60, 61).

[Gavrila, 2000] Dariu M Gavrila, 2000. „Pedestrian detection from a moving vehicle.“ In: *Proc. ECCV* (cit. on p. 21).

[Geiger *et al.*, 2012] Andreas Geiger, Philip Lenz, and Raquel Urtasun, 2012. „Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite.“ In: *Proc. CVPR* (cit. on p. 18).

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, 2014. „Rich feature hierarchies for accurate object detection and semantic segmentation.“ In: *Proc. CVPR* (cit. on pp. 22, 120).

[Girshick *et al.*, 2011] Ross Girshick, Pedro F Felzenszwalb, and David A Mcallester, 2011. „Object detection with grammar models.“ In: *Adv. NIPS* (cit. on p. 22).

[Goldman *et al.*, 2008] Dan B Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M Seitz, 2008. „Video object annotation, navigation, and composition.“ In: *Proc. UIST* (cit. on p. 84).

[Gould *et al.*, 2009] Stephen Gould, Richard Fulton, and Daphne Koller, 2009. „Decomposing a scene into geometric and semantically consistent regions.“ In: *Proc. ICCV* (cit. on p. 26).

[Grest *et al.*, 2005] Daniel Grest, Jan Woetzel, and Reinhard Koch, 2005. „Nonlinear Body Pose Estimation from Depth Images.“ In: *Pattern Recognition*. Springer, pp. 285–292 (cit. on p. 61).

[Guelat and Marcotte, 1986] Jacques Guelat and Patrice Marcotte, 1986. „Some comments on Wolfe's ”away step”.“ In: *Mathematical Programming* (cit. on p. 116).

[Gulshan *et al.*, 2011] Varun Gulshan, Victor Lempitsky, and Andrew Zisserman, 2011. „Humanising GrabCut: Learning to segment humans using the Kinect.“ In: *Proc. ICCV Workshop on Consumer Depth Cameras for Computer Vision* (cit. on p. 83).

[Guo and Schuurmans, 2007] Yuhong Guo and Dale Schuurmans, 2007. „Convex Relaxations of Latent Variable Training.“ In: *Adv. NIPS* (cit. on pp. 53, 106, 110, 117).

[El-Hakim *et al.*, 2004] Sabry F El-Hakim, J Angelo Beraldin, Michel Picard, and Guy Godin, 2004. „Detailed 3D reconstruction of large-scale heritage sites with integrated techniques.“ In: *Computer Graphics and Applications* 24.3, pp. 21–29 (cit. on p. 17).

[Hannah, 1974] Marsha J Hannah, 1974. *Computer matching of areas in stereo images*. Tech. rep. DTIC Document (cit. on p. 16).

[Hariharan *et al.*, 2014] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik, 2014. „Simultaneous Detection and Segmentation.“ In: *Proc. ECCV* (cit. on pp. 30, 105, 120, 121, 125).

[Hartley and Zisserman, 2000] Richard Hartley and Andrew Zisserman, 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press (cit. on pp. 17, 58).

[He and Gould, 2013] Xuming He and Stephen Gould, 2013. „Multi-instance object segmentation with exemplars.“ In: *Proc. ICCV Workshop* (cit. on p. 105).

[Hernández-Vela *et al.*, 2012] Antonio Hernández-Vela, Miguel Reyes, Víctor Ponce, and Sergio Escalera, 2012. „Grabcut-based human segmentation in video sequences.“ In: *Sensors* 12.11, pp. 15376–15393 (cit. on p. 30).

[Hirschmüller, 2008] Heiko Hirschmüller, 2008. „Stereo processing by semiglobal matching and mutual information.“ In: *Trans. PAMI* 30.2, pp. 328–341 (cit. on p. 17).

[Ion *et al.*, 2011] Adrian Ion, Joao Carreira, and Cristian Sminchisescu, 2011. „Probabilistic joint image segmentation and labeling.“ In: *Adv. NIPS* (cit. on p. 26).

[Ionescu *et al.*, 2011] Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu, 2011. „Latent Structured Models for Human Pose Estimation.“ In: *Proc. ICCV* (cit. on p. 23).

[Ionescu *et al.*, 2014] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu, 2014. „Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments.“ In: *Trans. PAMI* 36.7, pp. 1325–1339 (cit. on pp. 6, 17, 138).

[Isack and Boykov, 2012] Hossam Isack and Yuri Boykov, 2012. „Energy-Based Geometric Multi-model Fitting.“ In: *IJCV* 97.2, pp. 123–147 (cit. on p. 93).

[Ishikawa, 2003] Hiroshi Ishikawa, 2003. „Exact optimization for Markov random fields with convex priors.“ In: *Trans. PAMI* 25.10, pp. 1333–1336 (cit. on pp. 16, 48).

[Jaccard, 1912] Paul Jaccard, 1912. „The distribution of the flora in the alpine zone.“ In: *New Phytologist* (cit. on p. 122).

[Jaggi, 2013] Martin Jaggi, 2013. „Revisiting Frank-Wolfe: Projection-free sparse convex optimization.“ In: *Proc. ICML* (cit. on pp. 55, 115, 116).

[Jain and Grauman, 2014] Suyog Dutt Jain and Kristen Grauman, 2014. „Supervoxel-Consistent Foreground Propagation in Video.“ In: *Proc. ECCV* (cit. on pp. 29, 132, 133).

[Johnson and Everingham, 2010] Sam Johnson and Mark Everingham, 2010. „Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation.“ In: *Proc. BMVC* (cit. on p. 24).

[Johnson and Everingham, 2011] Sam Johnson and Mark Everingham, 2011. „Learning effective human pose estimation from inaccurate annotation.“ In: *Proc. CVPR* (cit. on p. 24).

[Joulin *et al.*, 2010] Armand Joulin, Francis Bach, and Jean Ponce, 2010. „Discriminative Clustering for Image Co-segmentation.“ In: *Proc. CVPR* (cit. on pp. 32, 33, 105, 106, 108, 110, 115, 117).

[Joulin *et al.*, 2012] Armand Joulin, Francis Bach, and Jean Ponce, 2012. „Multi-class cosegmentation.“ In: *Proc. CVPR* (cit. on pp. 33, 108).

[Julesz, 1962] Bela Julesz, 1962. „Towards the automation of binocular depth perception (AUTOMAP-1).“ In: *Proc. IFIPS Congress*. Vol. 1 (cit. on p. 15).

[Kerl *et al.*, 2013] Christian Kerl, Jurgen Sturm, and Daniel Cremers, 2013. „Dense visual SLAM for RGB-D cameras.“ In: *Proc. IROS*. IEEE, pp. 2100–2106 (cit. on p. 19).

[Kim *et al.*, 2011] Gunhee Kim, Eric P Xing, Li Fei-Fei, and Takeo Kanade, 2011. „Distributed cosegmentation via submodular optimization on anisotropic diffusion.“ In: *Proc. ICCV* (cit. on p. 32).

[Kolmogorov, 2006] Vladimir Kolmogorov, 2006. „Convergent Tree-Reweighted Message Passing for Energy Minimization.“ In: *Trans. PAMI* 28.10, pp. 1568–1583 (cit. on pp. 27, 48).

[Kolmogorov *et al.*, 2005] Vladimir Kolmogorov, Antonio Criminisi, Andrew Blake, Geoffrey Cross, and Carsten Rother, 2005. „Bi-layer segmentation of binocular stereo video.“ In: *Proc. CVPR* (cit. on pp. 31, 32).

[Kong and Tao, 2004] Dan Kong and Hai Tao, 2004. „A method for learning matching errors for stereo computation.“ In: *Proc. BMVC* (cit. on p. 16).

[Koppal *et al.*, 2011] Sanjeev J Koppal, C Lawrence Zitnick, Michael F Cohen, Sing Bing Kang, Bryan Ressler, and Alex Colburn, 2011. „A viewer-centric editor for 3D movies.“ In: *Computer Graphics and Applications* 31.1, pp. 20–35 (cit. on p. 84).

[Krähenbühl and Koltun, 2014] Philipp Krähenbühl and Vladlen Koltun, 2014. „Geodesic object proposals.“ In: *Proc. ECCV* (cit. on p. 22).

[Krähenbühl and Koltun, 2015] Philipp Krähenbühl and Vladlen Koltun, 2015. „Learning to propose objects.“ In: *Proc. CVPR* (cit. on p. 22).

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, 2012. „Imagenet classification with deep convolutional neural networks.“ In: *Adv. NIPS* (cit. on p. 22).

[Kukelova and Pajdla, 2007] Zuzana Kukelova and Tomas Pajdla, 2007. *A minimal solution to the autocalibration of radial distortion*. Tech. rep. (cit. on p. 60).

[Kumar *et al.*, 2010] M Pawan Kumar, Benjamin Packer, and Daphne Koller, 2010. „Self-paced learning for latent variable models." In: *Adv. NIPS* (cit. on p. 137).

[Kumar and Koller, 2010] Pawan Kumar and Daphne Koller, 2010. „Efficiently selecting regions for scene understanding." In: *Proc. CVPR* (cit. on p. 26).

[Lacoste-Julien and Jaggi, 2013] Simon Lacoste-Julien and Martin Jaggi, 2013. „An Affine Invariant Linear Convergence Analysis for Frank-Wolfe Algorithms." In: *arXiv* (cit. on p. 116).

[Lacoste-Julien and Jaggi, 2015] Simon Lacoste-Julien and Martin Jaggi, 2015. „On the global linear convergence of Frank-Wolfe optimization variants." In: *Adv. NIPS* (cit. on p. 116).

[Ladický *et al.*, 2009] Ľubor Ladický, Christopher Russell, Pushmeet Kohli, and Philip HS Torr, 2009. „Associative hierarchical CRFs for object class image segmentation." In: *Proc. ICCV* (cit. on p. 106).

[Ladický *et al.*, 2010] Ľubor Ladický, Paul Sturgess, Karteek Alahari, Chris Russell, and Philip HS Torr, 2010. „What, where and how many? combining object detectors and CRFs." In: *Proc. ECCV* (cit. on pp. 30, 106).

[Ladický *et al.*, 2013] Ľubor Ladický, Philip HS Torr, and Andrew Zisserman, 2013. „Human Pose Estimation using a Joint Pixel-wise and Part-wise Formulation." In: *Proc. CVPR* (cit. on p. 29).

[Laptev, 2013] Ivan Laptev, 2013. „Modeling and visual recognition of human actions and interactions." Habilitation à diriger des recherches. Ecole normale supérieure de Paris (cit. on p. 3).

[Lee *et al.*, 2011] Yong Jae Lee, Jaechul Kim, and Kristen Grauman, 2011. „Key-segments for video object segmentation." In: *Proc. ICCV* (cit. on p. 28).

[Lempitsky *et al.*, 2009] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp, 2009. „Image segmentation with a bounding box prior." In: *Proc. ICCV* (cit. on pp. 30, 105).

[Lezama *et al.*, 2011] José Lezama, Karteek Alahari, Josef Sivic, and Ivan Laptev, 2011. „Track to the future: Spatio-temporal video segmentation with long-range motion cues." In: *Proc. CVPR* (cit. on p. 28).

[Li *et al.*, 2013] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg, 2013. „Video Segmentation by Tracking Many Figure-Ground Segments." In: *Proc. ICCV* (cit. on pp. 28, 105, 130, 131).

[Liu, 2009] Ce Liu, 2009. „Beyond Pixels: Exploring New Representations and Applications for Motion Analysis." Doctoral dissertation. Massachusetts Institute of Technology (cit. on pp. 61, 63, 65, 66, 85).

[Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell, 2015. „Fully convolutional networks for semantic segmentation." In: *Proc. CVPR* (cit. on pp. 27, 31).

[Lowe, 1999] David Lowe, 1999. „Object recognition from local scale-invariant features." In: *Proc. ICCV* (cit. on p. 60).

[Marpe *et al.*, 2006] Detlev Marpe, Thomas Wiegand, and Gary J Sullivan, 2006. „The H.264/MPEG4 advanced video coding standard and its applications." In: *Communications Magazine* 44.8, pp. 134–143 (cit. on p. 57).

[Marr and Poggio, 1979] David Marr and Tomaso Poggio, 1979. „A computational theory of human stereo vision." In: *Proceedings of the Royal Society of London B: Biological Sciences* 204.1156, pp. 301–328 (cit. on p. 15).

[Mohan *et al.*, 2001] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio, 2001. „Example-based object detection in images by components." In: *Trans. PAMI* 23.4, pp. 349–361 (cit. on p. 21).

[Munoz *et al.*, 2010] Daniel Munoz, J Andrew Bagnell, and Martial Hebert, 2010. „Stacked hierarchical labeling." In: *Proc. ECCV* (cit. on p. 27).

[Newcombe *et al.*, 2011] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon, 2011. „KinectFusion: Real-time dense surface mapping and tracking." In: *Proc. ISMAR* (cit. on p. 19).

[Niebles *et al.*, 2010] Juan Carlos Niebles, Bohyung Han, and Li Fei-Fei, 2010. „Efficient Extraction of Human Motion Volumes by Tracking." In: *Proc. CVPR* (cit. on p. 83).

[Ochs *et al.*, 2014] Peter Ochs, Jitendra Malik, and Thomas Brox, 2014. „Segmentation of moving objects by long term video analysis." In: *Trans. PAMI* 36.6, pp. 1187–1200 (cit. on pp. 28, 120, 121, 125).

[Oren *et al.*, 1997] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio, 1997. „Pedestrian detection using wavelet templates." In: *Proc. CVPR* (cit. on p. 21).

[Papageorgiou and Poggio, 2000] Constantine Papageorgiou and Tomaso Poggio, 2000. „A trainable system for object detection." In: *IJCV* 38.1, pp. 15–33 (cit. on p. 21).

[Papandreou *et al.*, 2015] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L. Yuille, 2015. „Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation." In: *Proc. ICCV* (cit. on p. 31).

[Papazoglou and Ferrari, 2013] Anestis Papazoglou and Vittorio Ferrari, 2013. „Fast object segmentation in unconstrained video." In: *Proc. ICCV* (cit. on pp. 28, 120, 121, 125).

[Pinheiro *et al.*, 2015] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar, 2015. „Learning to segment object candidates." In: *Adv. NIPS* (cit. on p. 27).

[Pishchulin *et al.*, 2013] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele, 2013. „Strong Appearance and Expressive Spatial Models for Human Pose Estimation." In: *Proc. ICCV* (cit. on p. 24).

[Plagemann *et al.*, 2010] Christian Plagemann, Varun Ganapathi, Daphne Koller, and Sebastian Thrun, 2010. „Real-time identification and localization of body parts from depth images.“ In: *Proc. ICRA* (cit. on p. 61).

[Premebida *et al.*, 2009] Cristiano Premebida, Oswaldo Ludwig, and Urbano Nunes, 2009. „LIDAR and vision-based pedestrian detection system.“ In: *Journal of Field Robotics* 26.9, pp. 696–711 (cit. on p. 17).

[Rabe *et al.*, 2010] Clemens Rabe, Thomas Müller, Andreas Wedel, and Uwe Franke, 2010. „Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time.“ In: *Proc. ECCV* (cit. on p. 58).

[Ramanan, 2006] Deva Ramanan, 2006. „Learning to parse images of articulated bodies.“ In: *Adv. NIPS* (cit. on p. 24).

[Ramanan, 2013] Deva Ramanan, 2013. *Dual coordinate solvers for large-scale structural SVMs*. Tech. rep. (cit. on p. 46).

[Ramanan *et al.*, 2005] Deva Ramanan, David A Forsyth, and Andrew Zisserman, 2005. „Strike a pose: Tracking people by finding stylized poses.“ In: *Proc. CVPR* (cit. on p. 23).

[Ramanathan *et al.*, 2014] Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei, 2014. „Linking people with "their" names using coreference resolution.“ In: *Proc. ECCV* (cit. on p. 33).

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, 2015. „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.“ In: *Adv. NIPS* (cit. on pp. 22, 106, 137).

[Ren *et al.*, 2012] Xiaofeng Ren, Liefeng Bo, and Dieter Fox, 2012. „RGB-(D) Scene Labeling: Features and Algorithms.“ In: *Proc. CVPR* (cit. on p. 84).

[Revaud *et al.*, 2015] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid, 2015. „EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow.“ In: *Proc. CVPR* (cit. on pp. 63, 65, 66).

[Rother *et al.*, 2004] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, 2004. „"GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts.“ In: *Proc. SIGGRAPH* (cit. on p. 30).

[Rother *et al.*, 2006] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov, 2006. „Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs.“ In: *Proc. CVPR* (cit. on p. 32).

[Russell *et al.*, 2009] Chris Russell, Pushmeet Kohli, Philip HS Torr, *et al.*, 2009. „Associative hierarchical crfs for object class image segmentation.“ In: *Proc. ICCV* (cit. on p. 26).

[Sapp and Taskar, 2013] Benjamin Sapp and Ben Taskar, 2013. „MODEC: Multimodal Decomposable Models for Human Pose Estimation.“ In: *Proc. CVPR* (cit. on p. 24).

[Sapp *et al.*, 2010] Benjamin Sapp, Alexander Toshev, and Ben Taskar, 2010. „Cascaded models for articulated pose estimation.“ In: *Proc. ECCV* (cit. on p. 24).

[Sapp *et al.*, 2011] Benjamin Sapp, David Weiss, and Ben Taskar, 2011. „Parsing human motion with stretchable models.“ In: *Proc. CVPR* (cit. on p. 24).

[Scharstein and Szeliski, 2002] Daniel Scharstein and Richard Szeliski, 2002. „A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.“ In: *IJCV* 47.1-3, pp. 7–42 (cit. on pp. 15, 16).

[Seguin *et al.*, 2015] Guillaume Seguin, Karteek Alahari, Josef Sivic, and Ivan Laptev, 2015. „Pose estimation and segmentation of multiple people in stereoscopic movies.“ In: *Trans. PAMI* 37.8, pp. 1643–1655 (cit. on pp. 106, 120, 121, 125, 126).

[Sheasby *et al.*, 2012] Glenn Sheasby, Julien Valentin, Nigel Crook, and Philip HS Torr, 2012. „A Robust Stereo Prior for Human Segmentation.“ In: *Proc. ACCV* (cit. on pp. 32, 83, 85, 95, 100, 102–104).

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik, 2000. „Normalized cuts and image segmentation.“ In: *Trans. PAMI* 22.8, pp. 888–905 (cit. on pp. 51, 52, 105, 108, 109).

[Shi and Tomasi, 1994] Jianbo Shi and Carlo Tomasi, 1994. „Good Features to Track.“ In: *Proc. CVPR* (cit. on pp. 90, 120).

[Shotton *et al.*, 2011] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake, 2011. „Real-time human pose recognition in parts from single depth images.“ In: *Proc. CVPR* (cit. on pp. 6, 18, 19, 61, 84).

[Sidenbladh *et al.*, 2000] Hedvig Sidenbladh, Michael J Black, and David J Fleet, 2000. „Stochastic tracking of 3D human figures using 2D image motion.“ In: *Proc. ECCV* (cit. on p. 23).

[Silberman *et al.*, 2012] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus, 2012. „Indoor Segmentation and Support Inference from RGBD Images.“ In: *Proc. ECCV* (cit. on pp. 6, 20).

[Singh *et al.*, 2014] Ashutosh Singh, Jin Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel, 2014. „BigBIRD: A large-scale 3d database of object instances.“ In: *Proc. ICRA* (cit. on p. 19).

[Sminchisescu and Triggs, 2001] Cristian Sminchisescu and Bill Triggs, 2001. „Covariance scaled sampling for monocular 3D body tracking.“ In: *Proc. CVPR* (cit. on p. 23).

[Sminchisescu and Triggs, 2002] Cristian Sminchisescu and Bill Triggs, 2002. „Hyperdynamics importance sampling.“ In: *Proc. ECCV* (cit. on p. 23).

[Sminchisescu and Triggs, 2003] Cristian Sminchisescu and Bill Triggs, 2003. „Kinematic jump processes for monocular 3D human tracking.“ In: *Proc. CVPR* (cit. on p. 23).

[Snavely *et al.*, 2006] Noah Snavely, Steven M Seitz, and Richard Szeliski, 2006. „Photo tourism: exploring photo collections in 3D.“ In: *TOG* 25.3, pp. 835–846 (cit. on p. 17).

[Snoek *et al.*, 2012] Jasper Snoek, Hugo Larochelle, and Ryan P Adams, 2012. „Practical Bayesian optimization of machine learning algorithms.“ In: *Adv. NIPS* (cit. on p. 118).

[Spinello and Arras, 2011] Luciano Spinello and Kai O Arras, 2011. „People Detection in RGB-D Data.“ In: *Proc. IROS* (cit. on p. 68).

[Spyropoulos *et al.*, 2014] Aristotle Spyropoulos, Nikos Komodakis, and Philippos Mordohai, 2014. „Learning to detect ground control points for improving the accuracy of stereo matching.“ In: *Proc. CVPR* (cit. on p. 16).

[Tang *et al.*, 2014] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei, 2014. „Co-localization in Real-World Images.“ In: *Proc. CVPR* (cit. on p. 33).

[Taylor *et al.*, 2015] Brian Taylor, Vasiliy Karasev, and Stefano Soatto, 2015. „Causal Video Object Segmentation from Persistence of Occlusions.“ In: *Proc. CVPR* (cit. on pp. 28, 105).

[Tighe and Lazebnik, 2010] Joseph Tighe and Svetlana Lazebnik, 2010. „Superparsing: scalable nonparametric image parsing with superpixels.“ In: *Proc. ECCV* (cit. on p. 26).

[Tighe *et al.*, 2014] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik, 2014. „Scene parsing with object instances and occlusion ordering.“ In: *Proc. CVPR* (cit. on p. 105).

[Tompson *et al.*, 2014] Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler, 2014. „Joint training of a convolutional network and a graphical model for human pose estimation.“ In: *Adv. NIPS* (cit. on p. 24).

[Toshev and Szegedy, 2014] Alexander Toshev and Christian Szegedy, 2014. „Deeppose: Human pose estimation via deep neural networks.“ In: *Proc. CVPR* (cit. on p. 24).

[Tsai *et al.*, 2010] David Tsai, Matthew Flagg, and James M.Rehg, 2010. „Motion Coherent Tracking with Multi-label MRF optimization.“ In: *Proc. BMVC* (cit. on pp. 105, 130, 131, 133).

[Vedula *et al.*, 2005] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade, 2005. „Three-Dimensional Scene Flow.“ In: *Trans. PAMI* 27.1, pp. 475–480 (cit. on p. 58).

[Vineet *et al.*, 2011] Vibhav Vineet, Jonathan Warrell, Ľubor Ladický, and Philip Torr, 2011. „Human Instance Segmentation from Video using Detector-based Conditional Random Fields.“ In: *Proc. BMVC* (cit. on pp. 30, 105).

[Viola and Jones, 2004] Paul Viola and Michael J Jones, 2004. „Robust real-time face detection.“ In: *IJCV* 57.2, pp. 137–154 (cit. on p. 3).

[Walk *et al.*, 2010] Stefan Walk, Konrad Schindler, and Bernt Schiele, 2010. „Disparity statistics for pedestrian detection: Combining appearance, motion and stereo.“ In: *Proc. ECCV* (cit. on p. 68).

[Wedel *et al.*, 2008] Andreas Wedel, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers, 2008. „Efficient dense scene flow from sparse or dense stereo data.“ In: *Proc. ECCV* (cit. on p. 58).

[Weinzaepfel *et al.*, 2013] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid, 2013. „DeepFlow: Large displacement optical flow with deep matching.“ In: *Proc. ICCV* (cit. on p. 119).

[Wertheimer, 1923] Max Wertheimer, 1923. „Laws of organization in perceptual forms." In: *A source book of Gestalt Psychology* (cit. on p. 25).

[Winkler, 2003] Gerhard Winkler, 2003. *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*. Vol. 27. Springer Science & Business Media (cit. on p. 48).

[Wolfe, 1970] Philip Wolfe, 1970. „Convergence theory in nonlinear programming." In: *Integer and nonlinear programming* (cit. on p. 116).

[Wu and Leahy, 1993] Zhenyu Wu and Richard Leahy, 1993. „An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation." In: *Trans. PAMI* 15.11, pp. 1101–1113 (cit. on p. 52).

[Xie *et al.*, 2013] Ziang Xie, Ashutosh Singh, Justin Uang, Karthik S Narayan, and Pieter Abbeel, 2013. „Multimodal blending for high-accuracy instance recognition." In: *Proc. IROS* (cit. on p. 19).

[Yang *et al.*, 2011] Yi Yang, Sam Hallman, Deva Ramanan, and Charless C Fowlkes, 2011. „Layered Object Models for Image Segmentation." In: *Trans. PAMI* 34.9, pp. 1731–1743 (cit. on pp. 83, 87, 95).

[Yang and Ramanan, 2011] Yi Yang and Deva Ramanan, 2011. „Articulated Pose Estimation using Flexible Mixtures of Parts." In: *Proc. CVPR* (cit. on pp. 22, 24, 25, 32, 41, 42, 44, 45, 68, 70, 71, 73, 74, 89, 90).

[Yang and Ramanan, 2013] Yi Yang and Deva Ramanan, 2013. „Articulated Human Detection with Flexible Mixtures of Parts." In: *Trans. PAMI* 35.12, pp. 2878–2890 (cit. on pp. 43, 73).

[Yao and Fei-Fei, 2010] Bangpeng Yao and Li Fei-Fei, 2010. „Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities." In: *Proc. CVPR* (cit. on p. 83).

[Zaslavskiy *et al.*, 2009] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert, 2009. „A path following algorithm for the graph matching problem." In: *Trans. PAMI* 31.12, pp. 2227–2242 (cit. on p. 117).

[Žbontar and LeCun, 2015] Jure Žbontar and Yann LeCun, 2015. „Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches." In: *arXiv* (cit. on p. 16).

[Zhang *et al.*, 2009] Ke Zhang, Jiangbo Lu, and Gauthier Lafruit, 2009. „Cross-based local stereo matching using orthogonal integral images." In: *Trans. CSVT* 19.7, pp. 1073–1079 (cit. on p. 17).

[Zhang *et al.*, 2015a] Yu Zhang, Xiaowu Chen, Jia Li, Chen Wang, and Changqun Xia, 2015a. „Semantic Object Segmentation via Detection in Weakly Labeled Video." In: *Proc. CVPR* (cit. on p. 31).

[Zhang *et al.*, 2015b] Ziyu Zhang, Alex Schwing, Sanja Fidler, and Raquel Urtasun, 2015b. „Monocular Object Instance Segmentation and Depth Ordering with CNNs." In: *Proc. ICCV* (cit. on pp. 29, 105).

[Zheng *et al.*, 2015] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Torr, 2015. „Conditional Random Fields as Recurrent Neural Networks.“ In: *Proc. ICCV* (cit. on pp. 27, 119–121, 125).

List of Figures

1.1 Example of surveillance system output 2
1.2 Example of assistive robot 2
1.3 Artificial intelligence challenges of self-driving cars 4
1.4 Examples of people-related computer-vision tasks 5
1.5 Devices for 3D movies 6
1.6 Improvements of person detection and pose estimation methods when including disparity features 7
1.7 Examples of layered multi-person segmentation results 8
1.8 Examples of multi-instance object segmentation results 8
1.9 Challenges of person analysis 9
1.10 Challenges of working with 3D movies 10
1.11 Challenges of working with disparity estimates 11

2.1 Building Rome in a Day 18
2.2 Overview of the Kinect pose estimation algorithm 19
2.3 Examples of poselets 21
2.4 Results of a 3D pose search method 23
2.5 Multi-task neural network for part detection and localization 25
2.6 Example of desired semantic segmentation output 26
2.7 Unsupervised video segmentation using long term tracks 28
2.8 Illustration of a GrabCut result 30
2.9 Illustration of a binary segmentation method for stereo videos 32
2.10 Co-segmentation intuition 32

3.1 Examples of person detections with deformable part models and visualization of the model components 36
3.2 Inference procedure for the deformable part model for object detection 39
3.3 Illustration of the part mixtures 43
3.4 Example of tree structure for a pose estimation deformable part model 45
3.5 Examples of graphical models connectivity 50
3.6 Example of graphical model 50
3.7 Comparison of min-cut with normalized cut on a problematic example 52

4.1 Pipeline used to extract stereo pairs from 3D Blu Ray disks. The first three steps, from Ripping to Demuxing, are handled by the MakeMKV software (`http://www.makemkv.com/`). The decoding is performed by a modified version of a reference MPEG4 decoder. 58
4.2 Principle of stereovision 58
4.3 Principle of epipolar geometry 59
4.4 Principle of rectification 59
4.5 Comparison of disparity estimation performance on rectified and slightly unrectified stereo pairs 60
4.6 Rectification results 60
4.7 Rectification results 61
4.8 Examples of disparity maps 61
4.9 Left views and disparity maps computed using optical-flow. 62
4.10 Depth map provided by Microsoft Kinect. Fine depth information is produced by the sensor. Black areas correspond to occluded regions and occlusion boundaries. 63
4.11 Comparison between optical flow algorithms applied to disparity estimation 65
4.12 Comparison between optical flow algorithms applied to disparity estimation 66
4.13 Frames from the Inria 3D Movie Dataset v2 67
4.14 HOG features computed on RGB image and disparity map 69
4.15 Precision-recall curves for person detection in 3D movies 71
4.16 Qualitative comparison of the detectors trained with and without disparity features 72
4.17 Pose estimation performance evaluation in 3D movies 74
4.18 Qualitative comparison of the pose estimators trained with and without disparity features 75
4.19 Quantitative improvement of person detection performance when using disparity 76
4.20 Proposed method for automatic harvesting of person bounding boxes . 77
4.21 Examples of automatically harvested person examples 79
4.22 Precision-recall curves for person detection after automatically harvesting examples 80
4.23 Qualitative comparison of the depth-supervised person detection model 81

5.1 Proposed pipeline for segmentation of multiple people in 3D movies . . 84
5.2 Illustration of the person segmentation model 86
5.3 Occlusion-based unary costs 88
5.4 Examples of part-specific masks used to build pose masks 91
5.5 Examples of computed pose masks 92
5.6 Illustration of the disparity-based ordering scheme 94

5.7 Qualitative segmentation results for segmentation of multiple people in 3D movies . . . 96
5.8 Influence of temporal smoothness on segmentation quality . . . 98
5.9 Influence of pose masks on segmentation quality . . . 99
5.10 Influence of the smoothness cost on segmentation results . . . 99
5.11 Influence of the weight between disparity cues and pose masks and of the weight of the temporal term . . . 101
5.12 Qualitative influence of the weight between pose masks and disparity cues . . . 102
5.13 Qualitative results of our method for multi-person segmentation in 3D movies on the H2view dataset . . . 103

6.1 Sample results of our instance-level video segmentation method for multi-person scenes . . . 106
6.2 Spatio-temporal superpixels graph . . . 107
6.3 Illustration of constraints used in our instance-level segmentation method . . . 114
6.4 Shape of the non-convex cost used for refinement . . . 118
6.5 Example of superpixels map . . . 119
6.6 Qualitative results of our instance-level segmentation method . . . 123
6.7 Qualitative comparison of our instance-level segmentation with 5 baseline methods . . . 124
6.8 Per-video quantitative results of our instance-level segmentation method 127
6.9 Per-video qualitative results of our instance-level segmentation method 127
6.10 Correlation between optimization cost and segmentation performance 128
6.11 Influence of pose masks on track constraints . . . 129
6.12 Quantitative evaluation of pose-driven constraints weighting . . . 130
6.13 Qualitative evaluation of pose-driven constraints weighting . . . 131
6.14 Qualitative results of our instance-level segmentation method on two non-person videos . . . 131
6.15 Results of our instance-level segmentation method adapted for segmentation propagation on the SegTrack dataset . . . 133

7.1 Example of synthesized upper body pose example . . . 139

List of Tables

4.1 Quantitative pose estimation results in 3D movies 73

5.1 Quantitative segmentation results for segmentation of multiple people in 3D movies 97
5.2 Quantitative results of our method for multi-person segmentation in 3D movies on the Inria 3DMovie dataset using ground truth components 104
5.3 Quantitative results of our method for multi-person segmentation in 3D movies on the H2view dataset 104

6.1 Comprehensive study of our instance-level video segmentation method 122
6.2 Quantitative comparison for our instance-level video segmentation method 125
6.3 Results of our instance-level segmentation method adapted for segmentation propagation on the SegTrack dataset 133

Résumé

Les humains sont au coeur de nombreux problèmes de vision par ordinateur, tels que les systèmes de surveillance ou les voitures sans pilote. Ils sont également au centre de la plupart des contenus visuels, pouvant amener à des jeux de données très larges pour l'entraînement de modèles et d'algorithmes. Par ailleurs, si les données stéréoscopiques font l'objet d'études depuis longtemps, ce n'est que récemment que les films 3D sont devenus un succès commercial.

Dans cette thèse, nous étudions comment exploiter les données additionnelles issues des films 3D pour les tâches d'analyse des personnes. Nous explorons tout d'abord comment extraire une notion de profondeur à partir des films stéréoscopiques, sous la forme de cartes de disparité. Nous évaluons ensuite à quel point les méthodes de détection de personne et d'estimation de posture peuvent bénéficier de ces informations supplémentaires. En s'appuyant sur la relative facilité de la tâche de détection de personne dans les films 3D, nous développons une méthode pour collecter automatiquement des exemples de personnes dans les films 3D afin d'entraîner un détecteur de personne pour les films non 3D.

Nous nous concentrons ensuite sur la segmentation de plusieurs personnes dans les vidéos. Nous proposons tout d'abord une méthode pour segmenter plusieurs personnes dans les films 3D en combinant des informations dérivées des cartes de profondeur avec des informations dérivées d'estimations de posture. Nous formulons ce problème comme un problème d'étiquetage de graphe multi-étiquettes, et notre méthode intègre un modèle des occlusions pour produire une segmentation multi-instance par plan. Après avoir montré l'efficacité et les limitations de cette méthode, nous proposons un second modèle, qui ne repose lui que sur des détections de personne à travers la vidéo, et pas sur des estimations de posture. Nous formulons ce problème comme la minimisation d'un coût quadratique sous contraintes linéaires. Ces contraintes encodent les informations de localisation fournies par les détections de personne. Cette méthode ne nécessite pas d'information de posture ou des cartes de disparité, mais peut facilement intégrer ces signaux supplémentaires. Elle peut également être utilisée pour d'autres classes d'objets. Nous évaluons tous ces aspects et démontrons la performance de cette nouvelle méthode.

Mots Clés

vision par ordinateur, films 3D, détection de personne, estimation de pose, segmentation vidéo, segmentation multi-instance

Abstract

People are at the center of many computer vision tasks, such as surveillance systems or self-driving cars. They are also at the center of most visual contents, potentially providing very large datasets for training models and algorithms. While stereoscopic data has been studied for long, it is only recently that feature-length stereoscopic ("3D") movies became widely available.

In this thesis, we study how we can exploit the additional information provided by 3D movies for person analysis. We first explore how to extract a notion of depth from stereo movies in the form of disparity maps. We then evaluate how person detection and human pose estimation methods perform on such data. Leveraging the relative ease of the person detection task in 3D movies, we develop a method to automatically harvest examples of persons in 3D movies and train a person detector for standard color movies.

We then focus on the task of segmenting multiple people in videos. We first propose a method to segment multiple people in 3D videos by combining cues derived from pose estimates with ones derived from disparity maps. We formulate the segmentation problem as a multi-label Conditional Random Field problem, and our method integrates an occlusion model to produce a layered, multi-instance segmentation. After showing the effectiveness of this approach as well as its limitations, we propose a second model which only relies on tracks of person detections and not on pose estimates. We formulate our problem as a convex optimization one, with the minimization of a quadratic cost under linear equality or inequality constraints. These constraints weakly encode the localization information provided by person detections. This method does not explicitly require pose estimates or disparity maps but can integrate these additional cues. Our method can also be used for segmenting instances of other object classes from videos. We evaluate all these aspects and demonstrate the superior performance of this new method.

Keywords

computer vision, 3D movies, person detection, pose estimation, video segmentation, instance-level segmentation

Numéro national de thèse: