

UNIVERSITY OF GENOA

Doctoral School in

Sciences and Technologies for Information and Knowledge

Ph.D Course in

COMPUTATIONAL INTELLIGENCE

INFORMATION TECHNOLOGY FOR INTERACTING COGNITIVE ENVIRONMENTS

CYCLE XXVIII

DOCTOR OF PHILOSOPHY THESIS

HAND-RELATED METHODS IN EGOCENTRIC VISION

PIETRO MORERIO

Ph.D. Candidate

PROFESSOR CARLO S. REGAZZONI

Advisor

PROFESSOR SILVANO CINCOTTI

Chairperson

Abstract

The emergence of new pervasive wearable technologies such as action cameras and smart glasses, brings the focus of Computer Vision research to the so called First Person Vision (FPV), or Egocentric Vision. Nowadays, more and more everyday-life videos are being shot from a first-person point of view, overturning the classical fixed-camera understanding of Vision, specializing the existing knowledge of video processing from moving cameras and bringing new challenges in the field. The trend in research is already oriented towards a new type of Computer Vision, centred on moving sensors and driven by the need for new applications for smart wearable devices. More in detail, the simple realization that we often look at our hand, even while performing the simplest tasks in everyday life, motivates recent studies in hand-related inference methods.

Indeed, this thesis investigates hand-related methods, as a way for providing new functionalities to wearable devices. Inspired by a detailed state-of-the-art investigation, a unified hierarchical structure is proposed, that optimally organizes processing levels to reduce the computational cost of the system and improve its performance. Such structure borrows some concepts from the theory of Cognitive Dynamic Systems. The central body of the thesis consists then in a proposed approach for most of the levels sketched in the global framework proposed¹.

¹This thesis summarizes the effort of a three-year PhD and the results presented are the outcome of a deep collaboration with my research group. For this reason I am mostly using plural statements ("we"), since much of my work has strongly relied on my colleagues' cooperation.

Contents

1	Introduction	1
1.1	Motivation and context	1
1.2	Research contribution	3
1.3	Thesis outline	4
2	Background and Related Work	6
2.1	First Person Vision (FPV) video analysis	7
2.1.1	Objectives	13
2.1.2	Subtasks	23
2.1.3	Video and image features	25
2.1.4	Methods and algorithms	27
2.2	Public datasets	30
2.3	Conclusion and future research	32
3	Global Framework	34
3.1	Context and motivation	34
3.2	A unified hierarchical framework for hand-related methods	38
3.2.1	Levels structure	41
3.3	Cognitive framework	49
3.3.1	Functional model	50
3.3.2	Discussion	54
4	Hand Detection	56
4.1	State of the art	59
4.2	UNIGE-HANDS dataset	61
4.3	Hand-detection DBN	64
4.4	Results	69

4.5	Conclusions and future research	72
5	Hand Segmentation	74
5.1	Pixel-wise colour-based segmentation	74
5.2	Supapixel-based segmentation	80
5.2.1	A Generative Supapixel Method	82
5.2.2	Video optimization of Supapixel algorithms	95
6	Left/Right Identification	106
6.1	Introduction and related work	106
6.2	Hands-Identity	107
6.2.1	Building the L/R model	108
6.2.2	Hands occlusions	111
6.2.3	Segmentation disambiguation	113
6.3	Results	113
6.3.1	Perfect segmentation	113
6.3.2	Disambiguating occlusions	114
6.4	Conclusions and future research	115
7	Hand Pose	117
7.1	Pose and gesture recognition	117
7.2	Pose recognition framework	119
7.2.1	Colour segmentation	119
7.2.2	Graph construction	121
7.2.3	Graph spectral analysis	122
7.2.4	Classification	124
7.3	Major findings	125
7.3.1	Preliminary investigation (E1)	125
7.3.2	UTC dataset (E2)	127
7.4	Conclusion	128
8	Conclusion and Future Work	129
8.1	Summary of contribution and major findings	129
8.2	Future developments	130
	Bibliography	131

List of Figures

1-1	Number of articles per year directly related to FPV video analysis. This plot contains the articles published until 2014, to the best of our knowledge	2
1-2	Examples of the commercial smart patents. (a) Google patent of the smart-glasses; (b) Microsoft patent of an augmented reality wearable device.	3
2-1	Some of the more important works and commercial announcements in FPV.	8
2-2	Hierarchical structure to explain the state of the art in FPV video analysis.	12
2-3	Some of the more important works in <i>object recognition and tracking</i> . .	15
2-4	Some of the more important works in activity recognition.	18
2-5	Some of the more important works and commercial announcements in FPV.	20
3-1	Hand detection in a human silhouette [133]	36
3-2	Some of the most relevant papers on hand-based methods in FPV	39
3-3	Examples of hand-segmentation.	41
3-4	Hierarchical levels for hand-based FPV methods.	43
3-5	Optimized superpixels of a frame with hands [161]	45
3-6	Hand-Identification.	46
3-7	Hand-Tracking.	47
3-8	RGB and RGB-D wearable devices.	48
3-9	Cognitive cycle of a human being.	50
3-10	Haykin's hierarchy for Cognitive Dynamic Systems. Cognitive Perceptor (CP) unit; Cognitive Controller (CC) unit; Probabilistic Reasoning Machine (PRM) [94].	51
3-11	Perception-action cycle: details.	53

4-1	Dynamic Bayesian Network for smoothing the decision process.	65
4-2	Performance of the DBN in each of the locations in the UNIGE-HANDS dataset.	71
5-1	Experiment: hand colour characterization.	75
5-2	Histogram of hand pixels' colour (relative to a single frame, ROI is shown in Figure 5-1). Blue line is Y channel, green is Cr and red is Cb.	76
5-3	Average hand pixels' colour frame by frame, changing illumination in the scene	77
5-4	Colour based segmentation. (a) Clustering of Cb and Cr features (b) Clustering of Cb/Y and Cr/Y features.	78
5-5	Colour based segmentation. (a) Sample frame (b) Colour-based segmentation (c) Optic flow-improved segmentation.	79
5-6	Optic flow	79
5-7	Trend for the interest in Superpixels along the last 10 years (Web searches). 80	
5-8	Superpixel methods provide a higher level representation of an image. Such a representation is particularly suitable for segmentation purposes and in fact often used to this end.	82
5-9	ITM segmentation with $r_{max} = 60$, $\varepsilon = 0.005$ and spatial distance only. 88	
5-10	ITM segmentation with $r_{max} = 60$, $\varepsilon = 0.005$ and Euclidean distance. . 88	
5-11	ITM segmentation with $r_{max} = 60$, $\varepsilon = 0.005$ and 2D colour distance. . 89	
5-12	Segmentation without postprocessing step: isolated pixels are a common issue, especially along superpixel's borders.	91
5-13	Parameter comparison between GSP and SLIC superpixel methods. We fixed $r_{max} = 50$ and $a = 2$ for the Generative Superpixel method. This result in $\varepsilon = 0.013$. The algorithm then generates 30 superpixels, which are set as a parameter in SLIC. Setting $N_c = 35$ in slic is then equivalent to setting $a = 2$ in our method.	95
5-14	SLIC breaks for small N_{sp} (setting $a = 10$ in our method, is equivalent to having $N_c = 14$).	96
5-15	For a high number of superpixels, the SLIC method provides a better superpixel representation, although execution time it high above GSP's. In particular, many borders in the green region are extremely irregular. . 97	
5-16	Unlike standard k -means algorithms, SLIC searches a limited $2S \times 2S$ region only. The expected superpixels' size is $S \times S$, as the initialization grid cells. S is derived from the image dimensions and the number of desired superpixels	98

5-17	Execution time [ms] against average number of iterations required for convergence: the relation is approximately linear. Data are provided in table 5.3.	99
5-18	Graphical model (Dynamic Bayesian Network) for a Bayes filter. . . .	100
5-19	Rectangular patterns appear after a while. This drifting phenomenon is due to the fact that SLIC does not implement an exact k -means clustering, but searches in a $2S \times 2S$ window, which can leave gaps in case of divergent flows. The absence of noise in dynamic model makes the phenomenon even more evident	104
5-20	The issue of rectangular patterns is less predominant when injecting Gaussian noise in the dynamic model.	104
6-1	Block diagram of the proposed approach.	108
6-2	Manually segmented hands	109
6-3	Fitting segmentation blobs with ellipses	109
6-4	Empirical (Top) and theoretical (Bottom) hand distribution function given the distance to relative distance to the sides of the image. For the left(right) the relative distance to the left(right) side is used.	110
6-5	Hand-to-hand occlusion: a single blob is created and thus a single ellipse is generated.	112
7-1	(a,b,c) Typical poses corresponding to the three different activities considered in our preliminary results (E1). (d,e,f) Three grasps of the UTC dataset [101] (masks are provided), corresponding to three different taxonomic categories [34] used in E2.	118
7-2	Work-flow diagram of the proposed method.	120
7-3	Segmentation step	121
7-4	Graph construction trough modified ITM algorithm (typing pose)	124
7-5	The effect of the number of eigenvalues on the accuracy of the classifiers in the two experiments.	126
7-6	Execution time	127

List of Tables

2.1	Commercial approaches to wearable devices with FPV video recording capabilities	9
2.2	Summary of the articles reviewed in FPV video analysis according to the main objective	14
2.3	Number of times that a subtask is performed to accomplish a specific objective	24
2.4	Number of times that each feature is used in to solve an objective or subtask	26
2.5	Mathematical and computational methods used in objective or each subtask	27
2.6	Current datasets and sensors data availability	31
3.1	Comparision of the performance of the HOG-SVM and the proposed DBN.	44
4.1	Current datasets and sensors availability [22].	60
4.2	Examples of the dataset frames.	62
4.3	Performance of the proposed <i>hand-detectors</i>	63
4.4	Comparision of the performance of the HOG-SVM and the proposed DBN.	72
5.1	Parameters appearing in the Generative Superpixel method	92
5.2	Execution time (milliseconds)	94
5.3	Performances	102
6.1	Left and right hand identification at contour level	114
6.2	Evaluation of hand segmentation when split is required	115
6.3	Confusion matrices with and without occlusion disambiguation.	115

6.4	Detailed segmentation results for each video. The "Coffe" sequence is used for training. Confusion matrix for each testing video and the overall result are provided at a pixel level. The results include occlusion detection, segmentation disambiguation (split) and id-competition. . . .	116
7.1	Pairwise confusion of the classification accuracy with 10 eigenvalues (E1).	125
7.2	Confusion matrices of the classification accuracy (E2).	128

Chapter 1

Introduction

First Person Vision (Egocentric) video analysis stands nowadays as one of the emerging fields in computer vision. The availability of wearable devices, recording exactly what the user is looking at, is ineluctable and the opportunities and challenges carried by this kind of devices are broad. Particularly, for the first time a device is so intimate with the user to be able to record the movements of his hands, making hand-based applications for First Person Vision one the most promising area in the field.

1.1 Motivation and context

Portable head-mounted cameras, able to record dynamic high quality first-person videos, have become a common item among sportsmen over the last five years. These devices represent the first commercial attempts to record experiences from a first-person perspective. This technological trend is a follow-up of the academic results obtained in the late 1990s, combined with the growing interest of the people to record their daily activities. Up to now, no consensus has yet been reached in literature with respect to naming this video perspective. *First Person Vision* (FPV) is arguably the most commonly used, but other names, like *Egocentric Vision* or *Ego-vision* has also recently grown in popularity. The idea of recording and analyzing videos from this perspective is not new in fact, several such devices have been developed for research purposes over the last 15 years [145, 156, 154, 97, 29]. Figure 1-1 shows the growth in the number

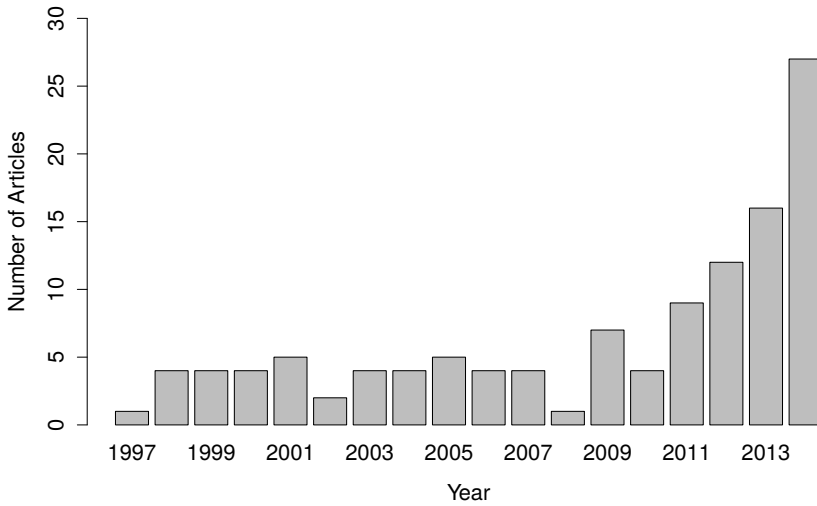
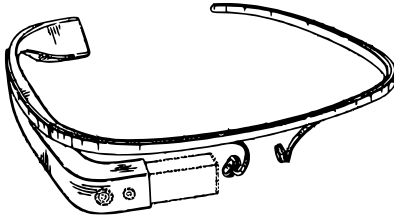


Figure 1-1: Number of articles per year directly related to FPV video analysis. This plot contains the articles published until 2014, to the best of our knowledge

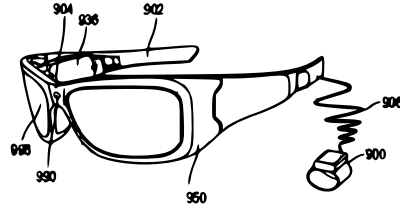
of articles related to FPV video analysis between 1997 and 2014. Quite remarkable is the seminal work carried out by the Media lab (MIT) in the late 1990s and early 2000s [214, 215, 196, 197, 212, 9], and the multiple devices proposed by Steve Mann who, back in 1997 [144], described the field with these words:

“Let’s imagine a new approach to computing in which the apparatus is always ready for use because it is worn like clothing. The computer screen, which also serves as a viewfinder, is visible at all times and performs multi-modal computing (text and images)”.

Recently, in the awakening of this technological trend, several companies have been showing interest in this kind of devices (mainly *smart-glasses*), and multiple patents have been presented. Figures 1-2(a) and 1-2(b) shows the devices patented in 2012 by Google and Microsoft. Together with its patent, Google also announced *Project Glass*, as a strategy to test its device among a exploratory group of people. The project was introduced by showing short previews of the Glasses’ FPV recording capabilities, and its ability to show relevant information to the user through the head-up display.



(a) Google glasses (U.S. Patent D659,741 - May 15, 2012).



(b) Microsoft augmented reality glasses (U.S. Patent Application 20120293548 - Nov 22, 2012).

Figure 1-2: Examples of the commercial smart patents. (a) Google patent of the smart-glasses; (b) Microsoft patent of an augmented reality wearable device.

Remarkably, the impact of the Glass Project (wich the most significant attempt to commercialize wearable technology up to date) is to be ascribed not only to its hardware, but also to the appeal of its underlying operating system. The latter continues to bring a large group of skilled developers, thus in turn making a significant boost in the number of prospective applications for smart-glasses, a phenomenon that has happened with smart-phones several years ago. On one hand, the range of application fields that could benefit from smart-glasses is wide and applications are expected in areas like military strategy, enterprise applications, tourist services [200], massive surveillance [148], medicine [111], driving assistance [135], among others. On the other hand, what was until now considered as a consolidated research field, needs to be re-evaluated and restated under the light of this technological trend: wearable technology and the first person perspective rise important issues, such as privacy and battery life, in addition to new algorithmic challenges [170].

1.2 Research contribution

The major research contributions of this work can be summarized as follows:

- A detailed and comprehensive *review of the evolution of First Person Vision methods*, together with a categorization of the latters and a discussion of challenges and opportunity within the field.
- A *global framework* for hand-related ego-vision methods: we propose a basic *hierarchical structure*, and how to extend it in order to provide wearable systems'

inference with *cognitive functionalities*.

- Some levels of the proposed framework are fully investigated at algorithmic level, namely:
 - *Hand detection* level: a classifier for dynamically detecting hands' presence in frames.
 - *Hand segmentation* level: a pixel-wise segmenter based on colour; a superpixel algorithm for segmentation; a framework for optimizing superpixel methods in videos.
 - *Hand identification* level (left-right): an identification algorithm based on a position-angle model.
 - *Hand pose recognition* level: a pose classification algorithm based on a graph representation of hands.

1.3 Thesis outline

This thesis is comprised of eight chapters, most of which are based on a number of peer-reviewed journals, conference and workshop papers. Each chapter is intended to serve as a stand-alone technical textbook, and, although being not completely detached from the rest of the thesis, sometimes re-introduces relevant concepts needed to gain a comprehensive insight into the topics it discusses, along with the corresponding bibliography. The main exception is represented by chapter 3, which provides the big picture of the global framework we introduce and thus includes several references to the other chapters.

The thesis is organized as follows:

Chapter 2 gives a background on First Person Vision and a detailed review of literature, which forms the base for the proposed framework. Particular stress is given to the evolution of technology, namely wearable devices. A categorization of algorithms, tasks and objectives is proposed in order to provide the reader with a well focused insight on the field. Eventually, a thorough investigation of existing datasets is presented.

Chapter 3 shifts the focus over hand-related First Person Vision. The justification of this interest lies in the fact that hands are almost always in our field of view, and are involved in the majority of everyday task. They are in fact one of the fundamental means we

employ to interact with the surrounding world and may thus represent a goods starting point for implementing context-aware functionalities and human-computer interfaces on wearable devices. A global hierarchical structure for hand-related computer vision methods is devised and justified. In addition, the framework of Cognitive Dynamic Systems is introduced and fused with the proposed approach as a way to provide a flavour of cognitive functionalities to wearable devices.

Chapter 4 addresses the lower level of the proposed hierarchy i.e. the detection step. This levels is intended to answer the question: *are there hands in the current frame?* The yes/no nature of the question makes a binary classification approach the most suitable to give an answer. The algorithm proposed relies on a SVM classifier which is fed with HOG features extracted by the current frame. A temporal smoothing process is also applied at the decision level by exploiting the mathematical formulation of the SVM.

Chapter 5 describes the second level of the hierarchy, namely the segmentation level. Once the answer is yes at the lower level, this module is asked the question: *where are hands located?*. This step is often (con)fused with the previous one but in fact performs a different task. The problem can be seen as a *local* (pixel-level) binary classification problem, and as such is first addressed in section 5.1. Section 5.2 revises the notion of *local* by introducing the concept of Superpixel, and proposes a novel algorithm together with a video optimization method for this class of methods.

Chapter 6 address the problem of giving segmented blobs a left-right id. This is accomplished by fitting the segmentation results with ellipses and by making them compete against a left and a right model. In case hand-to-hand occlusion is detected, this module is able to cope with it by splitting the fused blob using past segmentation results (disambiguation). Hand identification is performed after segmentation, however, since disambiguation is sometimes needed, the two levels are exchanging valuable information.

Chapter 7 investigates the hand pose problem, as the base for gesture recognition tasks. Typical hand poses can be recognized in a multi-class classification framework. The proposed algorithm relies on a graph representation of hands, which is in turn made possible by the fitting capabilities of a particular Neural Gas. A Laplacian representation of the graph provides discriminative features for hand pose classification.

Eventually, chapter 8 concludes the thesis, summarizing the most important concepts, highlighting major findings, drawing conclusions and sketching limitation of the work done that could be explored in the future.

Chapter 2

Background and Related Work

This chapter summarizes the state of the art in FPV video analysis and its temporal evolution between 1997 and 2014, analyzing the challenges and opportunities of this video perspective. It reviews the main characteristics of previous studies using tables of references, and the main events and relevant works using timelines. As an example, Figure 2-1 presents some of the most important papers and commercial announcements in the general evolution of FPV. We direct interested readers to the must read papers presented in this timeline. In the following sections, more detailed timelines are presented according to the objective addressed in the summarized papers. The categories and conceptual groups presented in this chapter reflects a schematic perception of the field coming from a detailed study of the existent literature. We are confident that the proposed categories are wide enough to conceptualize existent methods, however due to the growing speed of the field they could require future updates. As will be shown in the coming sections, the strategies used during the last 20 years are very heterogeneous. Therefore, rather than providing a comparative structure between existing methods and features, the objective of this chapter is to highlight common points of interest and relevant future lines of research. The bibliography presented in this chapter is mainly in FPV. However, some particular works in classic video analysis are also mentioned to support the analysis. The latter are cited using italic font as a visual cue¹.

To the best of our knowledge, the only work summarizing the general ideas of the FPV is [110], which presents a wearable device and several possible applications. Other

¹The literature overview presented in this chapter has been published as a survey paper in [22]

related reviews include the following: [91] reviews the activity recognition methods with multiple sensors; [62] analyzes the use of wearable cameras for medical applications; [154] presents some challenges of an active wearable device.

In the remainder of this chapter, existent methods in FPV are summarized according to a hierarchical structure proposed, highlighting the more relevant works and the temporal evolution of the field. Section 2.1 introduces general characteristics of FPV and the hierarchical structure, which is later used to summarize the current methods according to their final objective, the subtasks performed and the features used. In section 2.2 we briefly present the publicly-available FPV datasets. Finally, section 2.3 discusses some future challenges and research opportunities in this field.

2.1 First Person Vision (FPV) video analysis

During the late 1990s and early 2000s, the advances in FPV analysis were mainly performed using highly elaborated devices, typically proprietarily developed by different research groups. The list of devices proposed is wide, where each device was usually presented in conjunction with their potential applications and a large array of sensors which only envy from modern devices in their design, size and commercial capabilities. The column “Hardware” in Table 2.2 summarizes these devices. The remaining columns of this table are explained in section 2.1.1. Nowadays, current devices could be considered as the embodiment of the futuristic perspective of the already mentioned pioneering studies. Table 2.1 shows the currently available commercial projects and their embedded sensors. Such devices are grouped in three categories:

- **Smart-glasses:** Smart-glasses have multiple sensors, processing capabilities and a head-up display, making them ideal to develop real time methods and to improve the interaction between the user and its device. Besides, smart-glasses are nowadays seen as the starting point of an augmented reality system. However, they cannot be considered a mature product until major challenges, such as battery life, price and target market, are solved. The future of these devices is promising, but it is still not clear if they will be adopted by the users on a daily basis like smartphones, or whether they will become specialized task-oriented devices like industrial glasses, smart-helmets, sport devices, etc.
- **Action cameras:** commonly used by sportsmen and lifeloggers. However, the research community has been using them as a tool to develop methods and algorithms while anticipating the commercial availability of the smart-glasses during

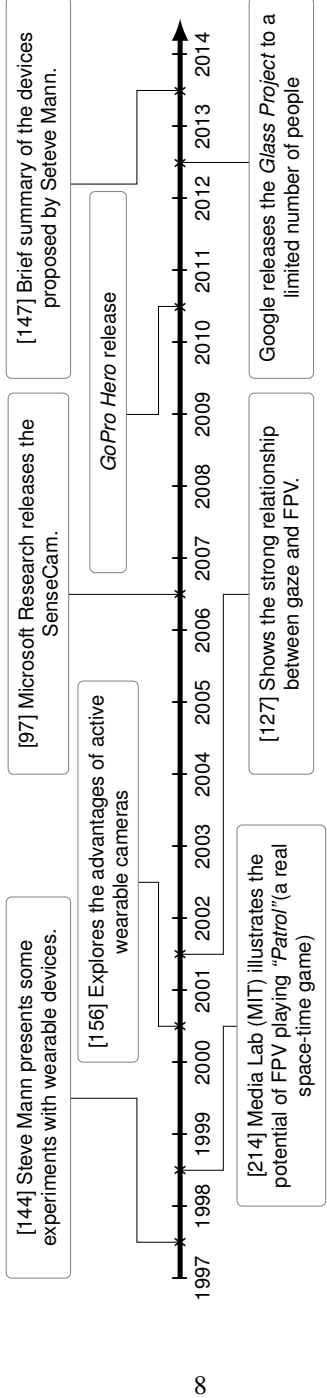


Figure 2-1: Some of the more important works and commercial announcements in FPV.

the coming years. Action cameras are becoming cheaper, and are starting to exhibit (albeit still somewhat limited) processing capabilities.

- **Eye trackers:** have been successfully applied to analyze consumer behaviors in commercial environments. Prototypes are available mainly for research purposes, where multiple applications have been proposed in conjunction with FPV. Despite the potential of these devices, their popularity is highly affected by the price of their components and the obtrusiveness of the eye tracker sensors, which is commonly carried out using an eye pointing camera.

Table 2.1: Commercial approaches to wearable devices with FPV video recording capabilities

	Camera	Eye Tracking	Microphone	GPS	Accelerometer	Gyroscope	Magnetometer	Altitude	Light Sensor	Proximity Sensor	Body-Heat Detector	Temperature Sensor	Head-Up Display
Google Glasses	✓		✓	✓	✓	✓	✓		✓	✓			✓
Epson Moverio	✓		✓	✓	✓	✓	✓						✓
Recon Jet	✓		✓	✓	✓	✓	✓					✓	✓
Vuzix M100	✓		✓		✓	✓	✓		✓	✓			✓
GlassUp	✓		✓		✓	✓	✓		✓				✓
Meta	✓		✓	✓	✓	✓							✓
Optinvent Ora-s	✓		✓	✓	✓	✓	✓		✓				✓
SenseCam	✓		✓		✓			✓	✓		✓	✓	
Lumus	✓		✓		✓	✓	✓						✓
Pivothead	✓		✓										
GoPro	✓		✓						✓				
Looxcie camera	✓		✓										
Epiphany Eyewear	✓		✓										
SMI Eye tracking Glasses	✓	✓	✓										
Tobii	✓	✓	✓										

¹ Other projects such as *Orcam*, *Nissan*, *Telepathy*, *Olympus MEG4.0*, *Oculon* and *Atheer* have been officially announced by their producers but no technical specifications have been already presented.

² According to unofficial online sources, other companies like *Apple*, *Samsung*, *Sony*, *Oakley* could be working on their own versions of similar devices, however no information has been officially announced up to date. *Microsoft* recently announced the Hololens but not technical specifications have been officially presented.

² This data is created on January 2015.

³ In [110] one multi-sensor device is presented for research purposes.

FPV video analysis gives some methodological and practical advantages, but also inherently brings a set of challenges that need to be addressed [110]. On one hand, FPV solves some problems of the classical video analysis and offers extra information:

- *Videos of the main part of the scene:* Wearable devices allow the user to (even unknowingly) record the most relevant parts of the scene for the analysis, thus reducing the necessity for complex controlled multi-camera systems [71].
- *Variability of the datasets:* Due to the increasing commercial interest of the technology companies, a large number of FPV videos is expected in the future, making it possible for the researchers to obtain large datasets that differ among themselves significantly, as discussed in section 2.2.
- *Illumination and scene configuration:* Changes in the illumination and global scene characteristics could be used as an important feature to detect the scene in which the user is involved, e.g. detecting changes in the place where the activity is taking place, as in [139].
- *Internal state inference:* According to [243], eye and head movements are directly influenced by the person's emotional state. As already done with smartphones [27], this fact can be exploited to infer the user's emotional state, and provide services accordingly.
- *Object positions:* Because users tend to see the objects while interacting with them, it is possible to take advantage of the prior knowledge of the hands' and objects' positions, e.g. active objects tend to be closer to the center, whereas hands tend to appear in the bottom left and bottom right part of the frames [180, 19].

On the other hand, FPV itself also presents multiple challenges, which particularly affect the choice of the features to be extracted by low level processing modules (feature selection is discussed in details in section 2.1.3):

- *Non static cameras:* One of the main characteristics of FPV videos is that cameras are always in movement. This fact makes it difficult to differentiate between the background and the foreground [87]. Camera calibration is not possible and often scale, rotation and/or translation-invariant features are required in higher level modules.
- *Illumination conditions:* The locations of the videos are highly variable and uncontrollable (e.g. visiting a touristic place during a sunny day, driving a car at night, brewing coffee in the kitchen). This makes it necessary to deploy robust

methods for dealing with the variability in illumination. Here shape descriptors may be preferred to color-based features [19].

- *Real time requirements:* One of the motivations for FPV video analysis is its potential of being used for real time activities. This implies the need for the real time processing capabilities [160].
- *Video processing:* Due to the embedded processing capabilities (for smart-glasses), it is important to define efficient computational strategies to optimize battery life, processing power and communication limits among the processing units. At this point, cloud computing could be seen as the most promising candidate tool to turn the FPV video analysis into an applicable framework for daily use. However, a real time cloud processing strategy requires further development in video compressing methods and communication protocols between the device and the cloud processing units.

The rest of this chapter summarizes FPV video analysis methods according to a hierarchical structure, as shown in Figure 2-2, starting from the raw video sequence (bottom) to the desired objectives (top). Section 2.1.1 summarizes the existent approaches according to 6 general objectives (Level 1). Section 2.1.2 divides these objectives in 15 weakly dependent subtasks (Level 2). Section briefly introduces the most commonly used image features, presenting their advantages and disadvantages, and relating them with objectives. Finally, section 2.1.4 summarizes the quantitative and computational tools used to process data, moving from one level to the other. In the literature review carried out, we found that existing methods are commonly presented as combinations of the aforementioned levels. However, no standard structure is presented, making it difficult for other researchers to replicate existing methods or improve the state of the art. We propose this hierarchical structure as an attempt to cope with this issue.

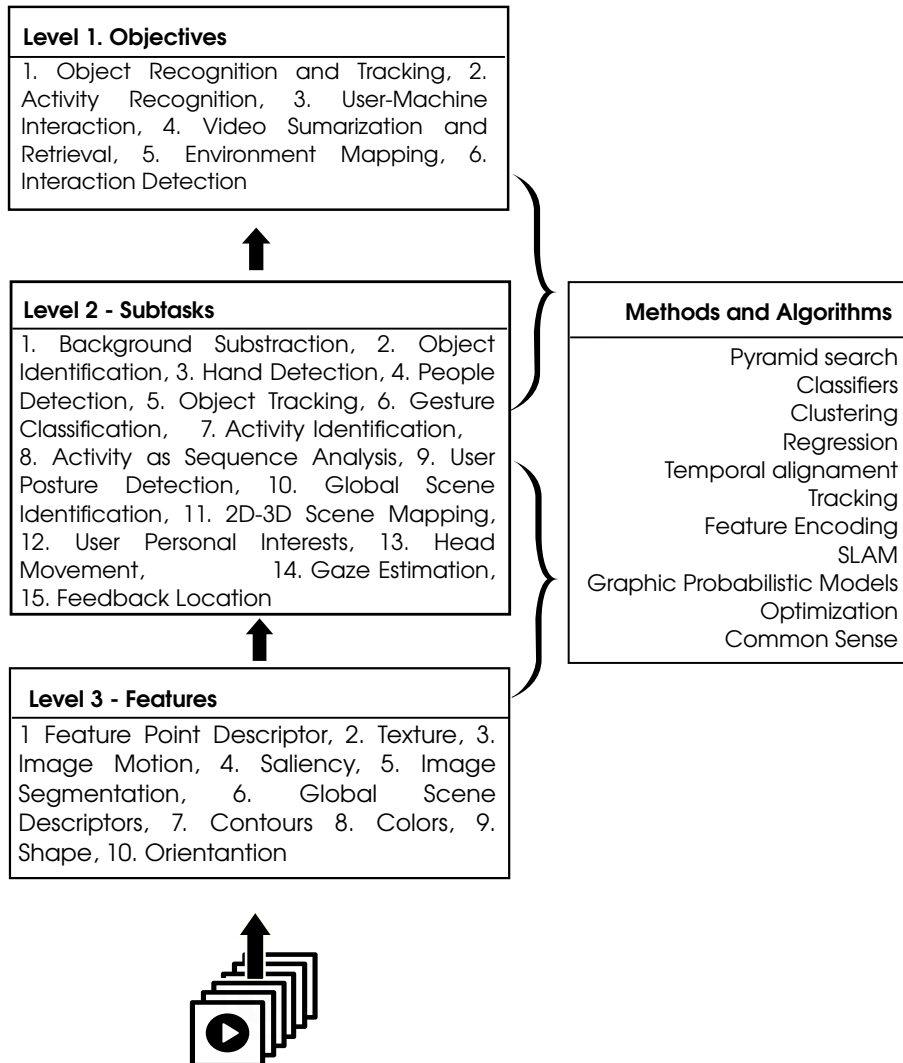


Figure 2-2: Hierarchical structure to explain the state of the art in FPV video analysis.

2.1.1 Objectives

Table 2.2 summarizes a total of 117 articles. The articles are divided in six objectives according to the main goal addressed in each of them. The left side of the table contains the six objectives described in this section, and on the right side, extra groups related to hardware, software, related surveys and conceptual articles, are given. The category named "Particular Subtasks" is used for articles focused on one of the subtasks presented in section 2.1.2. The last column shows the positive trend in the number of articles per year, and is plotted in Figure 1-1.

Note from the table that the most commonly explored objective is *Object Recognition and Tracking*. We identify it as the base of more advanced objectives such as *Activity Recognition*, *Video Summarization and Retrieval* and *Environment Mapping*. Another often studied objective is *User-Machine Interaction* because of its potential in Augmented Reality. Finally, a recent research line denoted as *Interaction Detection* allows the devices to infer situations in which the user is involved. Along with this section, some details are presented of how existent methods have addressed each of these 6 objectives. One important aspect is that some methods use multiple sensors within a data-fusion framework. For each objective, several examples of data-fusion and multi-sensor approaches are mentioned.

Object recognition and tracking

Object recognition and tracking is the most explored objective in FPV, and its results are commonly used as a starting point for more advanced tasks, such as activity recognition. Figure 2-3 summarizes some of the most important papers that focused on this objective.

In addition to the general opportunities and challenges of the FPV perspective, this objective introduces important aspects to be considered: i) Because of the uncontrolled characteristics of the videos, the number of objects, as well as their type, scale and point of view, are unknown [180, 182]. ii) Active objects, as well as user's hands, are frequently occluded. iii) Because of the mobile nature of the wearable cameras, it is not easy to create background-foreground models. iv) The camera location makes it possible to build a priori information about the objects' position [180, 19].

Hands are among the most common objects in the user's field of view, and a proper detection, localization, and tracking could be a main input for other objectives. The authors in [19] highlight the difference between hand-detection and hand-segmentation,

Table 2.2: Summary of the articles reviewed in FPV video analysis according to the main objective

Year	Objective		Extra Categories					# Articles Reviewed			
	Object Recognition and Tracking	Activity Recognition	User-Machine Interaction	Video Summarization and Retrieval	Environment Mapping	Interaction Detection	Particular Subtasks		Related Software Design	Hardware	Conceptual Academic Articles
1997									[144]		1
1998	[215, 214, 33]								[145]		4
1999	[196, 51]	[214]	[197]						[197]		4
2000	[68]	[50]	[126]						[156]		4
2001	[125]		[124, 117]	[5]			[127]				5
2002				[169, 85]			[85]				2
2003			[59]	[194, 98]						[148]	4
2004			[12, 119]	[86]			[86]		[146]		4
2005	[155, 217]	[155]	[223]	[221, 4]			[4]		[154]		5
2006		[29]	[13, 118]						[97, 29]		4
2007	[42, 41]	[235]			[42, 56, 41]						4
2008	[43]				[43]						1
2009	[180, 217]	[244, 210, 218]	[142]							[170]	7
2010	[188]	[111, 66]		[66, 37]							4
2011	[96, 74]	[61, 70]		[3, 61, 115]	[175]				[60]		9
2012		[31, 182, 72, 87, 114, 171]	[90]			[71]	[240]			[91]	12
2013	[131, 132, 160, 231, 246]	[134, 88, 73]	[200, 248]	[139, 73]	[193]		[152]		[110]	[213]	16
2014	*	[167, 153, 54, 184, 250, 247]		**	[7, 6]		[19, 128]	[92]		[195]	27

* [232, 136, 76, 183, 191, 192, 222, 35, 135]

** [238, 158, 30, 15, 172, 11]

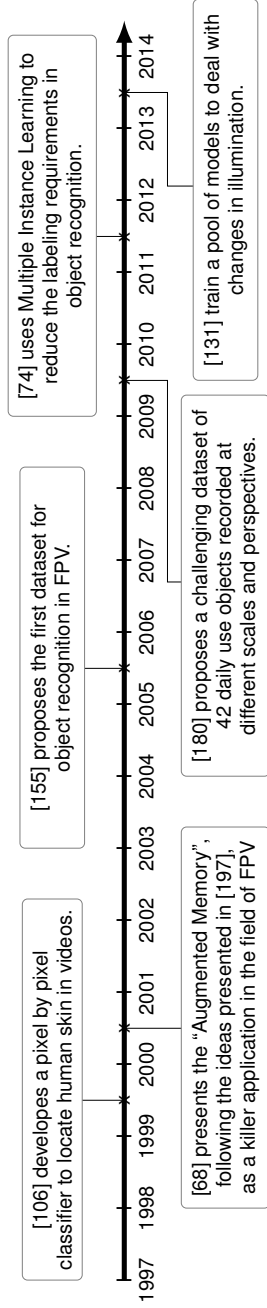


Figure 2-3: Some of the more important works in *object recognition and tracking*.

particularly in the framework of wearable devices where the number of deployed computational resources, directly influences the battery life of the devices. In general, due to the hardware availability and price, hand-detection and tracking is usually carried out using RGB videos. However, [191, 192] uses a chest-mounted RGB-D camera to improve the hand-detection and tracking performance in realistic scenarios. According to [155], hand detection could be divided into model-driven and data-driven methods.

Model-driven methods search for the best matching configuration of a computational hand model (2D or 3D) to recreate the image that is being shown in the video [199, 198, 186, 217, 191, 192]. These methods are able to infer detailed information of the hands, such as the posture, but in exchange large computational resources, highly controlled environments or extra sensors (e.g. Depth Cameras) could be required.

Data-driven methods use image features to detect and segment users' hands. The most commonly used features for this purpose are the color histograms looking to exploit the particular chromaticism of human skin, especially in suitable color spaces like HSV and YCbCr [160, 200, 131, 132]. Color-based methods can be considered as the evolution of the pixel-by-pixel skin classifiers proposed in [106], in which color histograms are used to decide whether a pixel represents human skin. Despite their advantages, the color-based methods are far from being an optimal solution. Two of their more important restrictions are: i) The computational cost, because in each frame they have to solve the $O(n^2)$ problem implied by the pixel-by-pixel classification. ii) Their results highly influenced by significant changes in illumination, for example indoor and outdoor videos[19]. To reduce the computational cost, some authors suggest the use of super-pixels [200, 160, 132], however, an exhaustive comparison of the computational times of both approaches is still pending, and computationally efficient superpixel methods applied to video (especially FPV video) are still at an early stage [161]. Regarding the noisy results, the authors in [131, 200] train a pool of models and automatically select the most appropriate depending on the current environmental conditions.

In addition to hands, there is an uncountable number of objects that could appear in front of the user, whose proper identification could lead to development of some of the most promising applications of FPV. An example is "The Virtual Augmented Memory(VAM)" proposed by [68], where the device is able to identify objects, and to subsequently relate them to previous information, experiences or common knowledge available online. An interesting extension of the VAM is presented in [24], where the user is spatially located using his video, and is shown relevant information about the place or a particular event. In the same line of research, recent approaches have been trying to fuse information from multiple wearable cameras to recognize when the users are being recorded by a

third person without permission. This is accomplished in [183, 99] using the motion of the wearable camera as the identity signature, which is subsequently matched in the third person videos without disclosing private information such as the face or the identity of the user.

The augmented memory is not the only application of object recognition. The authors in [182] develop an *activity recognition method* which based only a list of the used objects in the recorded video . Despite the importance of these applications, the problem of recognition is far from being solved due to the large amount of objects to be identified as well as the multiple positions and scales from which they could be observed. It is here that machine learning starts playing a key role in the field, offering tools to reduce the required knowledge about the objects [74] or exploiting web services (such as Amazon Turk) and automatic mining for training purposes [208, 87, 18, 235].

Once the objects are detected, it is possible to track their movements. In the case of the hands, some authors use the coordinates of center as the reference point [160], while others go a step further and use dynamic models [119, 118]. Dynamic models are widely studied and are successfully used to track hands, external objects [56, 42, 56, 43, 41], or faces of other people [33].

Activity recognition

An intuitive step in the hierarchy of objectives is *Activity Recognition*, aimed at identifying what the user is doing in a particular video sequence. Figure 2-4 presents some of the most relevant papers on this topic. A common approach in activity recognition is to consider an activity as a sequence of events that can be modeled as Markov Chains or as Dynamic Bayesian Networks (DBNs) [214, 196, 50, 29, 218]. Despite the promising results of this approach, the main challenge to be solved is the scalability to multiple user and multiple strategies to solve a similar task.

Recently, two major methodological approaches for *activity recognition* are becoming popular: object based and motion based recognition. Object based methods aim to infer the activity using the objects appearing in video sequence [218, 70, 182], assuming of course that the activities can be described by the required group of objects(e.g. prepare a cup of coffee requires coffee, water and a spoon). This approach opens the door to highly scalable strategies based on web mining to know the objects usually required for different activities. However, after all, this approach depends on a proper *Object Recognition* step and on its own challenges (Section 2.1.1). Following an alternative

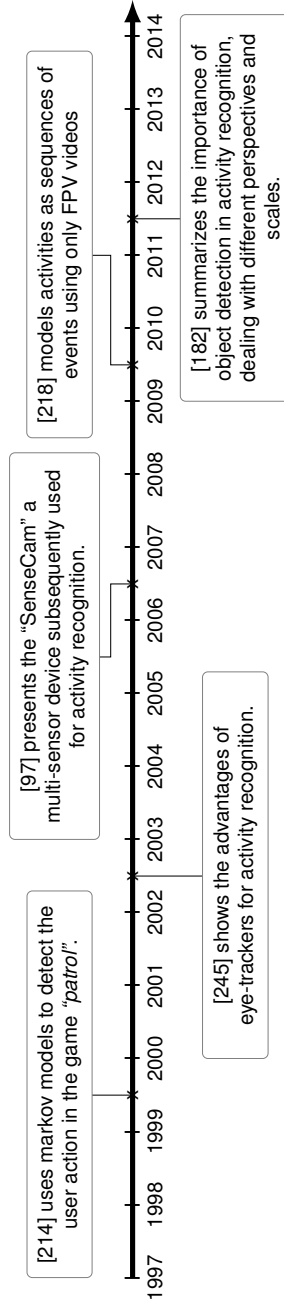


Figure 2-4: Some of the more important works in activity recognition.

path, during the last 3 years, some authors have been using the fact that different kind of activities create different body motions and as consequence different motion patterns in the video, for example: walking, running, jumping, skiing, reading, watching movies, among others [115, 171, 184]. It is remarkable the discriminative power of motion features for this kind of activities and the robustness to deal with the illumination and the color skin challenges.

Activity recognition is one of the fields that has drawn most benefits from the use of multiple sensors. This strategy started growing in popularity with the seminal work of Clarkson et al. [50, 51] where basic activities are identified using FPV video jointly with audio signals. An intuitive realization of the multi-sensor strategy allows to reduce the dependency between *Activity Recognition* and *Object Recognition*, by using Radio-Frequency Identification (RFID) tags in the objects [235, 123, 176, 181]. However, the use of RFIDs reduces the applicability to environments previously tagged. The list of multiple sensors does not end with audio and RFIDs, it also contains Inertial Measurement Units [210], multiple sensors of the “SenseCam²” [61, 37], GPS [87], and eye-trackers [244, 72, 241, 240, 134].

User-machine interaction

As already mentioned, smart-glasses open the door to new ways for interaction between the user and his device. The device, being able to give feedback to the user, allows to close the interaction loop originated by the visual information captured and interpreted by the camera. Only approaches related to FPV video analysis are presented (other sensors are omitted, such as audio and touch panels), categorizing them based on two approaches: i) the user sends information to the device, and ii) the device uses the information of the video to show the feedback to the user. Figure 2-5 shows some of the most important works concerning *User-machine interaction*.

In general, the interaction between the user and his device starts with intentional or unintentional command. An *intentional command* is a signal sent by the user using his hands through his camera. This kind of interaction is not a recent idea and several approaches have been proposed, particularly using static cameras [190, 84], which, as mentioned in section 2.1.1, can not be straightforwardly applied to FPV due to the mobile nature of wearable cameras. A traditional approach is to emulate the mouse of computers with the hands [186, 126, 124], allowing the user to point and click at virtual objects created in

²Wearable device developed by Microsoft Research in Cambridge with accelerometers, thermometer, infrared and light sensor

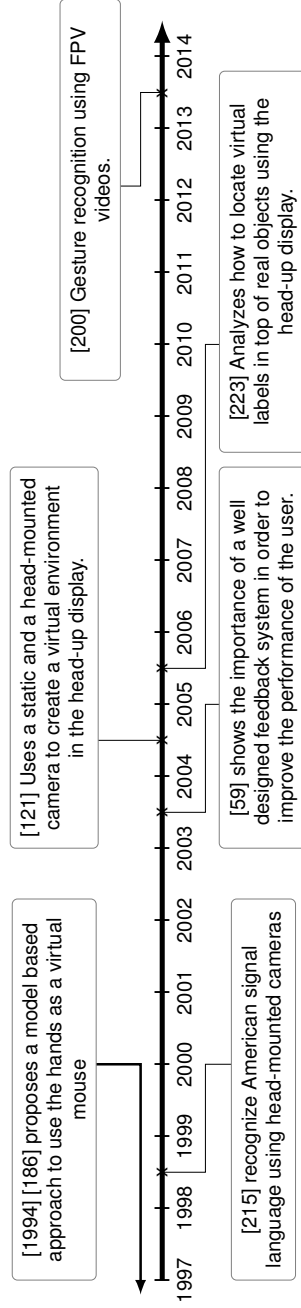


Figure 2-5: Some of the more important works and commercial announcements in FPV.

the head-up display. Other approaches look for more intuitive and technology focused ways of interaction. For example, the authors in [200] develop a gesture recognition algorithm to be used in an interactive museum using 5 different gestures: “point out”, “like”, “dislike”, “OK” and “victory”. In [248], the head movements of the user are used to assist a robot in the task of finding a hidden object in a controlled environment. Under this perspective some authors combine static and wearable cameras [121, 215]. Quite remarkable are the results of Starner in 1998, being able to recognize American sign language with an efficiency of 98% with a static camera and head mounted camera. As is evident, hand-tracking methods can give important cues in this objective [164, 211, 201, 119], and make it possible to use features such as position, speed or acceleration of the users’ hands.

Unintentional commands are triggers activated by the device using information about the user without his conscious intervention, for example: i) the user is cooking by following a particular recipe (Activity Recognition), and the device could monitor the time of different steps without the user previously asking for it. ii) The user is looking at a particular item [Object Recognition] in a store [GPS or Scene Recognition] then the device could show price comparisons and reviews. *Unintentional commands* could be detected using the results of other FPV objectives, the measurements of its sensors, or behavioural routines learned from the user while previously using his device, among others. From our point of view, these kinds of commands could be the next step of user-machine interaction for smart-glasses, and a main enabler to reduce the required time to interact with the device [213].

Regarding the second part of the interaction loop, it is important to properly design the feedback system to know when, where, how, and which information should be shown to the user. In order to accomplish this, several issues must be considered in order to avoid misbehaviour of the system that could work against the user’s performance in addressing relevant tasks [59]. In this line, multiple studies develop methods to optimally locate virtual labels in the user’s visual field, without occluding the important parts of the scene [142, 223, 90].

Video summarization and retrieval

The main task of *Video summarization and retrieval* is to create tools to explore and visualize the most important parts of large FPV video sequences [139]. The objective and main issue is perfectly summarized in [5] with the following sentence: “*We want to record our entire life by video. However, the problem is how to handle such a huge*

data". In general, existing methods define importance functions to select the more relevant subsequences or frames of the video, and later cut or accelerate the less important ones [172]. Recent studies define the importance function using the objects appearing in the video [87], their temporal relationships and causalities [139], or as a similarity function, in terms of its composition, between them and intentional pictures taken with a traditional cameras [238]. A remarkable result is achieved in [115, 184] using motion features to segment videos according to the activity performed by the user. This work is a good example of how to take advantage of the camera movements in FPV, usually considered as a challenge, to achieve good classification rates.

The use of multiple sensors is common within this objective, and remarkable fusions have been made using brain measurements in [5, 169], gyroscopes, accelerometers, GPS, weather information and skin temperature in [194, 98, 221], and online available pictures in [238]. An alternative approach to video summarization is presented in [175] and [11], where multiple FPV videos of the same scene are unified using the collective attention of the wearable cameras as an importance function. In order to define whether the two videos recorded from different cameras are pointing at the same scene, the authors in [16] use superpixels and motion features to propose a similarity measurement. Finally, it is significant to mention that "Video summarization and retrieval" has led to important improvements in the design of the databases and visualization methods to store and explore the recorded videos [85, 86]. In particular, this kind of developments can be considered an important tool for reducing computational requirements in the devices, as well as alleviate privacy issues related with the place where videos are stored.

Environment Mapping

Environment Mapping aims at the construction of a 2D or 3D virtual representation of the environment surrounding the user. In general, the of variables to be mapped can be divided in two categories: physical variables, such as walls and object locations, and intangible variables, such as attention points. Physical mapping is the more explored of the two groups. It started to grow in popularity with [56], which showed how, by using multiple sensors, Kalman Filters and monoSLAM, it is possible to elaborate a virtual map of the environment. Subsequently, this method was improved by adding object identification and location as a preliminary stage [42, 41]. Physical mapping is one of the more complex tasks in FPV, particularly when 3D maps are required due to the calibration restrictions. This problem can be partially alleviated by using a multi-camera approach to infer the depth [43, 110]. Research on *intangible variables*, can be

considered an emerging field in FPV. Existent approaches define attention points and attraction fields, mapping them in rooms with multiple people interacting [11].

Interaction detection

The objectives described above are mainly focused on the user of the device as the only person that matters in the scene. However, they hardly take into account the general situation in which the user is involved. We label the group of methods aiming to recognize the types of interaction that the user is having with other people as *Interaction Detection*. One of the main purposes in this objective is *social interaction detection*, as proposed by [71]. In their paper, the authors inferred the gaze of the other people and used it to recognize human interactions as monologues, discussions or dialogues. Another approach in this field was proposed by [193], which detected different behaviors of the people surrounding the user (e.g. hugging, punching, throwing objects, among others). Despite not being widely explored yet, this objective can be considered one of the most promising and innovative ones in FPV due to the mobility and personalization capabilities of the coming devices.

2.1.2 Subtasks

As explained before, the proposed structure is based on objectives which are highly co-dependent. Moreover, it is common to find that the output of one objective is subsequently used as the input for the other (e.g. activity recognition usually depends on object recognition). For this reason, a common practice is to first address small subtasks, and later merge them to accomplish main objectives. Based on the literature review, a total of 15 subtasks are proposed. Table 2.3 shows the number of articles analyzed in this survey that use a subtask (columns) in order to address a particular objective (rows). It is important to highlight the many-to-many relationship among objectives and subtasks, which means that a subtask could be used to address different objectives, and one objective could require multiple subtasks. To mention some: i) hand detection, as a subtask, could be the objective itself in object recognition, [160], but could also give important cues in activity recognition [72]; moreover, it could be the main input in the user-machine interaction [200]. ii) The authors in [182] performed object recognition to subsequently infer the performed activity. As their names are self-explanatory, separate explanation of each of the subtasks is omitted, with the possible exceptions of the following: i) *Activity as a Sequence* analyzes an activity as a set of ordered steps; ii) *2D-*

3D Scene Mapping builds a 2D or 3D virtual representation of the scene recorded; iii) *User Personal Interests* identifies the parts in the video sequence potentially interesting for the user using physiological signals such as brainwaves[169]; iv) *Feedback location* identifies the optimal place in the head-up display to locate the virtual feedback without interfering with the user’s visual field.

Table 2.3: Number of times that a subtask is performed to accomplish a specific objective

Objective	Background Subtraction	Object Identification	Hand Detection and Segmentation	People Detection	Object Tracking	Gesture Identification	Activity Identification	Activity as Sequence Analysis	User Posture Detection	Global Scene Identification	2D-3D Scene Mapping	User Personal Interests	Head Movement	Gaze Estimation	Feedback Location
Object Recognition and Tracking	4	15	13	3	10					2			1	1	
Activity Recognition	3	8	3	1			13	2	1	8	1		6	6	
User-Machine Interaction			6		3	2							1		3
Video Summarization	1	4	1	4			5	1		4		1	2	1	
Environment Mapping		3			4						5			1	
Interaction Detection				2	1		2				1		2	2	

As can be deduced from table 2.3, *Hand detection* plays an important role as the base for advanced objectives such as *Object Recognition* and *User-Machine interaction*. *Global scene identification*, as well as *Object Identification*, stand out as two important subtasks for activity recognition. More in detail, the tight bound between the *Activity Recognition* and the *Object Recognition* supports the idea of [182], which states that Activity Recognition is “all about objects”. Moreover, the use of gaze estimation in multiple objectives confirms the advantages of the recent trend of using eye-trackers in conjunction with FPV videos. Finally, it can be noted that *Background Subtraction* has lost some of its reputation if compared with fixed camera scenarios, due to the highly unstable nature of the backgrounds when observed from the First-person perspective.

2.1.3 Video and image features

As mentioned before, FPV implies highly dynamic changes in the attributes and characteristics of the scene. Due to these changes, an appropriate selection of the features becomes critical in order to alleviate the challenges and exploit the advantages presented in section 2.1. As is well known, feature selection is not a trivial task, and usually implies an exhaustive search in the literature and extensive testing to identify which method leads to optimal results.

The process of feature extraction is carried out at different levels, starting from the pixel level, with color channels of the image, and subsequently extracting more elaborated indicators at the frame level, such as *saliency*, *texture*, *superpixels*, *gradients*, etc. As expected, these features can be used to address some of the subtasks, such as object recognition or scene identification. However, they do not include any kind of dynamic information. To add dynamic information in the analysis, different approaches can be followed, for example analyzing the geometrical transformation between two frames to obtain image *Motion* features such as *optical flow*, or aggregating frame level features in temporal windows. Usually, dynamic features tend to be computationally expensive, and are therefore usually applied to objectives in which the video is processed once the activities have finished. Particularly interesting is the method presented in [161], which uses the information of the superpixels of the previous frame to initialize and compute the current frame superpixels, thus reducing the computational complexity of the algorithm by 60%.

Table 2.4 shows the most commonly used features in FPV to address a particular subtask. The features are listed in the rows and the subtasks in the columns. Note that *color histograms* are by far the most commonly used feature for almost all the subtasks, despite being highly criticized due to their dependence on illumination changes. Another group of features frequently used for several subtasks is *Image Motion*. Some of its most remarkable results are for *Activity Recognition* in [115, 184], for *Video Summarization* in [172], and recently as the input of a Convolutional Neural Network (CNN) to create a biometric sensor that is able to identify the user recording the video in [99]. The use of *Feature Point Descriptors* (FPD) is also worth noting. As expected, they are popular for object identification, but it is also remarkable their application to identify relevant places such as touristic hotspots [3, 111, 66]. Note from the table that the “dynamic objectives” like *Activity Recognition* and *Video Summarization* are the ones which take the most advantage of the *Motion* features, while *Object Recognition* is mainly based on frame features such as *FPD* and *Color histograms*.

Table 2.4: Number of times that each feature is used in to solve an objective or subtask

		Objectives						Subtasks														
		Object Recognition and Tracking	Activity Recognition	User Machine interaction	Video Summarization and Retrieval	Environment Mapping	Interaction Detection	Background Subtraction	Object Identification	Hand Detection and segmentation	People detection	Object Tracking	Gesture Classification	Activity Identification	Activity as Sequence Analysis	User Posture Detection	Global Scene Identification	2D-3D mapping	User Personal Interests	Head Movement	Gaze estimation	Feedback Location
FPD	SIFT	9	5	1	4	3		2	14		1			1		2						
	GFTT	1	1									1										
	BRIEF	2								1							1					
	FAST	1															1					
	SURF	2	6		1				2		1	1		2		2						
	Diff. of Gaussians				1				1													
	ORB	1								1												
	STIP		2					1	1													
Texture	Wiccest				1									1								
	Laplacian Transform	1								1												
	Edge Histogram		1	1	1				1										1			1
	Wavlets	1									1											
	Other		1		1									1								
Saliency	GBVS	1	1																			4
	Other		1						1												1	
	MSSS			1																		1
Motion	Optical Flow	5	14	2	5		1	5	1	2		2		6		1				4	5	
	Motion Vectors		1	1	3							1				1		1	3	1		
	Temporal Templates		1							1												
Glob. Scene	CRFH		1													1						
	GIST	1	2							1						2					1	
Img. Segment.	Superpixels	2	2	1				2	3													
	Blobs	2								1	1											
Contour	OWT-UCM	1	3					2	2													
Color	Histograms	21	20	11	10			3	8	20	4	4	1	5		7		1			2	
Shapes	HOG	6	4		3		1	2	5	1	1		3			1				1		
Orientation	Gabor				1									1								

From personal previous studies in *Hand-detection* and *Hand-segmentation* using multiple features and *superpixels*, *Color* features are a good approach, particularly if a suitable color space is exploited [160]. We found that low level features such as *Color Histograms* could help to reduce the computational complexity of the methods and get close to real time applications. On the other side, under large illumination changes, in [19] we highlight how *Color-based* hand-segmentators could introduce and disseminate in the system noise created by hands misdetections. To alleviate this problem, we used shape features, such as HOG, in order to pre-filter wrong measurements and improve the classification rate of the overall system.

The two empty columns in table 2.4 can be explained as follows: *Activity as a sequence* is usually chained with the output of a short activity identification [9, 244, 3], whereas identification of the *User Posture* is accomplished in [29] without employing visual features, but using GPS and accelerometers.

2.1.4 Methods and algorithms

Once that features are selected and estimated, the next step is to use them as inputs to reach the objective (outputs). At this point, quantitative methods start playing the main role, and as expected, an appropriate selection directly influences the quality of the results, ultimately showing whether the advantages of the FPV perspective are being exploited, or whether the FPV-related challenges are impacting the objectives negatively. Table 2.5 shows the number of occurrences of each method (rows) being used to accomplish a particular objective or a subtask (columns).

Table 2.5: Mathematical and computational methods used in objective or each subtask

	Objective						Subtasks																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
	Object Recognition and Tracking						Subtasks																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
	Activity Recognition		User Machine interaction		Video Summarization and Retrieval		Environment Mapping		Interaction Detection		Background Subtraction		Object Identification		Hand Detection and segmentation		People detection		Object Tracking		Gesture Classification		Activity Identification		Activity as Sequence Analysis		User Posture Detection		Global Scene Identification		2D-3D Scene Mapping		User Personal Interests		Head Movement		Gaze estimation		Feedback Location																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
3D Mapping	3	1				4																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							

^{PGM} Probabilistic Graphical Models.

^{DPMM} Dirichlet Process Mixture Models.

The table highlights *classifiers* as the most popular tool in FPV, which is commonly used to assign a category to an array of characteristics (see [138] for a more detailed survey on classifiers). The use of classifiers is wide and varies from general applications, such as scene recognition [74], to more specific, such as activity recognition given a set of objects [72]. Particularly, we found that the most used are the Support Vector Machines (SVM) due to their capability to deal with non-separable non-linear multi-label problems using low computational resources. On the other hand, SVMs require large labeled training sets which restricts the range of potential applications.

In our previous works we performed a comparison of the performance of multiple features (HOG, GIST, Color) and classifiers (SVM, Random Forest, Random Trees) to solve the hand-detection problem [19]. Our conclusion was that HOG-SVM was the best performing combination, achieving a classification rate of 90% and 93% of true positives and true negatives respectively. Another group of methods commonly used are *clustering* algorithms due to its simplicity, computational cost, and small requirements in the training datasets. Despite their advantages, *clustering* algorithms could require post-processing analysis of the results in order to endow them with human interpretation.

Another promising group of tools are the *Probabilistic Graphical Models* (PGMs), which can be interpreted as a framework to combine multiple sensors and chain results from different methods in a unique probabilistic hierarchical structure (e.g. to recognize the object and subsequently use it to infer the activity). *Dynamic Bayesian Networks* (DBNs) are a particular type of PGMs which include time in their structure, in turn making them suitable for application in video analysis [47]. As an example, DBNs are frequently used to represent activities as sequences of events [214, 196, 50, 29, 218]. It is common to find that particular methods, such as Dirichlet Process Mixture Models (DPMM), are presented in their PGM notation, however given the promising recent results achieved in *Activity Recognition* and *Video Segmentation*, they were grouped separately.

As stated in section 2.1.3, there is a large number of features that can be extracted for FPV applications. A common practice is to mix or chain multiple features before using them as input of a particular algorithm (table 2.5). This practice usually results in extremely large vectors of features that can lead to computationally expensive algorithms. In this context, the role of *Feature Encoding* methods, such as Bag-of-Words, is crucial to control the size of the inputs. We highlight the importance that some authors are giving to this tool, which, despite not being an automatic strategy like Linear Discriminant Analysis (LDA) and Principal Components Analysis (PCA), can nevertheless help to include human intuition in the analysis. As an example, the authors in [153] use BoW

in *Activity Recognition* taking into account the presence, level of attention, and the role of the objects in the video.

The use of machine learning methods (e.g. classifiers, clustering, regressors) introduces an important question to the analysis: how to train the algorithms on realistic data without restricting their applicability? This question is widely studied in the field of Artificial Intelligence, and two different approaches are commonly followed, namely unsupervised and supervised learning [40]. Unsupervised learning requires less human interaction in training steps, but requires human interpretation of the results. Additionally, unsupervised methods have the advantage of being easily adaptable to changes in the video (e.g. new objects in the scene or uncontrolled environments [210]). The most commonly used unsupervised method in FPV are the clustering algorithms, such as k-means. In fact, the best performing superpixels are the result of an unsupervised clustering procedure applied over a raw image[2]. In [161] we proposed an optimization of the SLIC superpixels, and latter in [162] we introduced a new superpixel method based on Neural Networks. The proposed algorithm is a self-growing map that adapts its topology to the frame structure taking advantage of the dynamic information available in the previous frames.

Regarding the supervised methods, their results are easily interpretable but commonly imply higher requirements in the training stage. As an example, at the beginning of this section some of the applications of SVMs were highlighted. Supervised methods use a set of inputs, previously labeled, to parametrize the models. Once the method is trained, it can be used on new instances without any additional human supervision. In general, supervised methods are more dependent on the training data, fact which could work against their performance when used on newly-introduced cases [182, 139, 210, 71, 235, 87, 129]. In order to reduce the training requirements, and take advantage of the useful information available on Internet, some authors create their datasets using services like Amazon Mechanical Turk [208, 87], automatic web mining [18, 235], or image repositories [238]. We named this practice in table 2.5 as *Common Sense*.

Weakly supervised learning is another commonly used strategy, considered as a middle point between supervised and unsupervised learning. This strategy is used to improve the supervised methods in two aspects: i) extending the capability of the method to deal with unexpected data; and ii) reducing the necessity for large training datasets. Following this trend, the authors of [66, 111] used Bag of Features (BoF) to monitor the activity of people with dementia. Later, [74, 70] used Multiple Instance Learning (MIL) to recognize objects using general categories. Afterwards, [3] used BoF and Vector of Locally Aggregated Descriptors (VLAD) to temporally align a sequence of videos.

Eventually, let us mention *Deep learning*, a relatively recent approach which combines supervised and unsupervised learning techniques in a unified framework, where low level significant features are learned in an unsupervised fashion [116].

2.2 Public datasets

In order to support their results and create benchmarks in FPV video analysis, some authors have provided their datasets for public use to the academic community. The first publicly available FPV dataset is released by [155]. It consists of a video containing 600 frames recorded in a controlled office environment using a camera on the left shoulder, while the user interacts with five different objects. Later, [180] proposed a larger dataset with two people interacting with 42 object instances. The latter one is commonly considered as the first challenging FPV dataset because it guaranteed the requirements identified by [197]: i) Scale and texture variations, ii) Frame resolution, iii) Motion blur, and iv) Occlusion by hand.

Implicitly, previous sections explain some of the main characteristics of FPV videos. In [220], these characteristics are compared for several FPV and Third Person Vision (TPV) datasets and their classification capabilities are evaluated. The authors reach a classification accuracy of 80.9% using blur, illumination changes, and optical flow as input features. In their study they also found a considerable difference in the classification rate explained by the camera position. The authors concluded that the more stable the camera, the less blur and motion and then the less discriminative power of these features. We highlight this difference as an important finding because it opens the door to an interesting discussion concerning which kind of videos, based on quantitative measurements, should be considered as FPV. Extra evidence about the role of the non-wearable cameras, such as hand-held devices when they are used to record from a first person perspective, is still pending. Our intuition points that, despite having some of the challenging characteristics of wearable cameras like mobile backgrounds and unstable motion patterns, hand-held videos would drastically differ in terms of features compared in [220].

Table 2.6 presents a list of the publicly-available datasets, along with their characteristics. Of particular interest are the changes in the camera location, which have evolved from shoulder-based to the head-based. These changes are clearly explained by the trend of the smart-glasses and action cameras (see Table 2.1). Also noticeable are the changes in the objectives of the datasets, moving from low level, such as object recognition,

Table 2.6: Current datasets and sensors data availability

Sensors												# Objects		Cam. Location			
		Year	Location	Controlled Conditions	Objective	Video	Depth-Sensor	IMUs	Body Media	eWatch	Eye Tracking	Activities	Objects	Num. of people	Shoulder	Chest	Head
Mayol05	[155]	2005	Desktop	✓	O1	✓							5	1	✓		
Intel	[180]	2009	Multiple locations		O1	✓							42	2	✓		
Kitchen.	[210]	2009	Kitchen Recipes	✓	O2	✓		✓	✓	✓		3	18				✓
GTEA11	[70]	2011	Kitchen Recipes	✓	O2	✓						7	4				✓
VINST	[3]	2011	Going to the work		O2	✓								1		✓	
UEC Dataset	[115]	2011	Park		O2	✓						29	1				✓
ADL	[182]	2012	Daily activities		O2	✓						18	20			✓	
UTE	[87]	2012	Daily activities		O4	✓							4				✓
Disney	[71]	2012	Thematic Park		O6	✓							8				✓
GTEA gaze	[72]	2012	Kitchen Recipes	✓	O2	✓					✓	7	10				✓
EDSH	[132]	2013	Multiple locations		O1	✓						-	-	-			✓
JPL	[193]	2013	Office Building		O6	✓						7	1				✓
EGO-HSGR	[200]	2013	Library Exhibition		O3	✓						5	1				✓
BEOID	[54]	2014	Multiple locations		O2	✓					✓	6	5				✓
EGO-GROUP	[7]	2014	Multiple locations		O6	✓							19				✓
EGO-HPE	[6]	2014	Multiple locations		O1	✓							4				✓
EgoSeg	[184]	2014	Multiple locations		O2	✓						7	2				✓
Egocentric Intel/Creative	[191]	2014	Multiple locations		O1	✓	✓						2			✓	

* **Objectives:** [O1] Object Recognition and Tracking. [O2] Activity Recognition. [O3] User-Machine Interaction. [O4] Video Summarization. [O5] Physical Scene Reconstruction. [O6] Interaction Detection.

** The table summarizes the characteristic described in the technical reports or the papers proposing the datasets.

to more complex objectives, such as social interaction and user-machine interaction. It should also be noted that less controlled environments have recently been proposed to improve the robustness of the methods in realistic situations. In order to highlight the robustness of their methods, several authors evaluated them on Youtube sequences recorded using goPro cameras [115].

Another aspect to highlight from the table is the availability of multiple sensors in some of the datasets. For instance, the Kitchen dataset [210] includes four sensors, the GTEA approach [72] includes eye tracking measurements, and the Egocentric Intel/Creative [191] was recorded with a RGBD camera.

2.3 Conclusion and future research

Wearable devices such as smart-glasses will presumably constitute a significant share of the technology market during the coming years, bringing new challenges and opportunities in video analytics. The interest in the academic world has been growing in order to satisfy the methodological requirements of this emerging technology. This survey provides a summary of the state of the art from the academic and commercial point of view, and summarizes the hierarchical structure of the existent methods. This paper shows the large number of developments in the field during the last 20 years, highlighting main achievements and some of the up-coming lines of study.

From the commercial and regulatory point of view, important issues must be faced before the proper commercialization of this new technology can take place. Nowadays, the privacy of the recorded people is one of the most discussed ones, as these kinds of devices are commonly perceived as intruders [170]. Other important aspects are the legal regulations depending on the country, , and the intention of the user to avoid recording private places or activities[222]. Another hot topic is the real applicability of smart-glasses as a massive consumption device or as a task-oriented tool to be worn only in particular scenarios. In this field, the technological companies are designing their strategies in order to reach out to specific markets. As an illustration, recent turn of events has seen Google move out of the glass project (originally intended to end with a massively commercialized product), in order to target the enterprise market. Microsoft, on the other hand, recently announced its task-oriented holographic device “HoloLens” embodied with a larger array of sensors.

From the academic point of view, the research opportunities in FPV are still wide. Under the light of this bibliographic review and our personal experience, we identify 4 main hot topics:

- Existing methods are proposed and executed in previously recorded videos. However, none of them seems to be able to work in a *closed-loop* fashion, by continuously learning from users’ experiences and adapt to the highly variable and uncontrollable surrounding environment. From our previous studies [48, 49], we believe that a cognitive perspective could give important cues to this aspect and could aid the development of the self-adaptive devices.
- The *personalization* capabilities of smart-glasses open the door to new learning strategies. Incoming methods should be able to receive personalized training from the owner of the device. We have found out, for instance, that this kind of ap-

proach can help alleviate problems, such as changes in the color skin models from different users [160] in a hand detection application. Indeed, color features, as stressed in 2.4, has proven to be extremely suitable to be exploited in this field.

- This thesis focuses on methods for addressing tasks accomplished mainly by one user coupled with a single wearable device. However, *cooperative devices* would be useful to increase the number of applications in areas such as environment mapping, military applications, cooperative games, sports, etc.
- Finally, regarding the *real time requirements*, important developments should be made in order to optimally compute FPV methods without draining the battery. This must be accomplished both from the hardware and the software side. On the one hand, progress still needs to be made on the processing units of the devices. On the other, lighter, faster and better optimized methods are yet to be designed and tested. Our personal experience lead us to explore fast machine learning methods [19] for hand detection, in the trend highlighted by table 2.5, and to discard standard features such as optic flow [160] because of computational restrictions. Promising methods in standard computer vision research, such as superpixel methods, were built from scratch in [162] in order to make them faster and better suited for video analysis [161]. Eventually, important cues to the problem of computational power optimization may also be found in cloud computing and high performance computing.

Chapter 3

Global Framework

This chapter justifies the interest in hand-related methods in First Person Vision (section 3.1) and fit them in a global structure. Starting from a hierarchical taxonomy of algorithms (section 3.2), the framework is extended to include cognitive features in the design (section 3.3)¹.

3.1 Context and motivation

Based on the considerations done in the previous chapters, we see the emergence of a new field of research in computer vision, focused on the users' point of view. Such a *First Person Computer Vision* (FPCV) is *de facto* solicited, and at the same time made possible, by the above mentioned brand-new available technology, namely a wearable computer equipped with a first-person camera framing everyday life. Noticeably, FPV is somehow more specific than simple vision from moving cameras, due to the constraints the device has with the subject and his sub-parts. This specificity might represent an exploitable added value in processing the information gathered, especially for what concerns interactions between the user and the outside world.

¹This chapter is mainly based on two peer reviewed publications: the overview presented in section 3.1 can be found in [160], while the analysis provided in section 3.2 has been published in [20]

As already mentioned, FPV from wearable devices involves information fusion at various complexity and abstraction levels ranging from pure image processing to inference over patterns. We stress here three points:

- At a low level, image processing must be exploited for object detection, localization, tracking and recognition. Here the problem of information fusion [28] appears in a manifold of aspects [58] [209] [64].
- Wearable devices such as the above described glasses are going to be equipped with other sensors than a camera. These may include an accelerometer, a gyroscope, a magnetometer, wifi, bluetooth, gps barometer, microphone and many others. Fusing data from this variety of sensor will become an issue in designing applications as for smartphones. Multisensor data fusion is a well known issue and has always drawn the attention of researchers in many fields [79] [78].
- At a higher level of abstraction, scene and behaviour understanding is required for cognitive applications. Here context-based information fusion plays a central role. More in detail, the modelling of the interaction between a the user and the outside world often relies on the fusion of "external" and "internal" information (e.g. [150] and [46]).

We present here a global framework, focusing on inference over hands. The reason we focus on hands is exceptionally simple: hands are almost always in our field of view, and are involved in the majority of everyday task (just think about writing, lacing shoes, driving, eating ...). Hands are maybe the principal means we employ to actively interact with the surrounding world, things and persons. So we claim that hands are the best starting point for implementing context-aware functionalities on wearable devices. In addition, new technologies as for instance the Microsoft Kinect and the ever-new Leap Motion controllers [207], seem to be pushing towards hands-free and even device-free interfaces for an enhanced human/computer interaction. In this context, gesture recognition will play a central role.

Hand detection, segmentation, tracking and extended tracking (i.e. tracking of hand sub-parts) are problems which have been widely addressed in computer vision. At present, a perfect hand segmentation or accurate hand localization are hardly reached especially under complex conditions. Past approaches are mainly focused on detecting hands from a fixed camera framing a whole person and thus relied on prior knowledge of human silhouettes [122]. The credit for the best degree of accuracy surely goes to depth sensors, which allow for a 3D reconstruction of the scene and a more accurate segmentation of hand shapes [133], thanks to the additional information carried by the depth channel.

Yet, we focus on old fashioned digital cameras, as for the moment no such depth sensor is expected to be wearable, although there exist prototypes implementing stereoscopic vision.



Figure 3-1: Hand detection in a human silhouette [133]

Most 2D approaches for hand tracking rely on skin colour features, which looks natural. However, colour features are sensitive to variations in illumination and shadows. A colour correction method is proposed in [236] to overcome this difficulty. It is here claimed that a Gaussian Mixture Model (GMM) can well capture complex variations caused by the difference of human races, gender, age etc. In [157] a Skin HOG model (SHOG) is proposed to construct a robust and efficient hand detector. However, a hybrid approach seems preferable, as also claimed in [26], where the Viola-Jones-like object detection scheme (originally applied to face detection [227]) is combined with a colour based detector, giving satisfactory results for the set of postures considered.

As already pointed out, in the near future more and more videos will be shot by wearable devices, from a first-person point of view. FPV video processing will be more and more requested (raising, among other things, considerable privacy issues). We address here some points of this problems and make some considerations.

Good news is that first person perspective can be somehow exploited. Besides, hand tracking is a very peculiar problem. It is common understanding that the more specific a problem is, the more the constraints from the problem itself can be taken advantage of. General issues are often more difficult to be addressed.

- *Number of targets.* Although FPV hand tracking is not a problem with a fixed number of targets, priors on it can be guessed. No more than two targets are allowed, and an equal probability of having zero, one or two targets in the scene can be conjectured.
- *Shape.* Hands from a first person point of view are often framed together with the naked arm. A typical oblong silhouette is shown in Figure 7-3.

- *Occlusions*. Hands from a first person point of view are hardly occluded. Hand-to-hand or hand-to-object occlusions may of course occur, however hand-to-body occlusions hardly happen.
- *Interactions*. Hands often interact one with the other while performing basic tasks. It has been showed [150] that target interactions can be exploited for improving object tracking.
- *Geometrical constraints*. Hands are linked through arms to the trunk, on which our head is mounted. Degrees of freedom in moving them are then limited by geometrical constraints, which can be then exploited in hand tracking.
- *Personalization* Skin colour features vary a lot from person to person and it is difficult to capture such complex variations, caused by the difference of human races, gender, etc. However, wearable devices are personal (as mobile phones and smartphones), and customized colour models learned from a single user are simpler and more reliable.

On the other hand, some bad news comes out in considering first person perspective from a wearable device.

- *Moving camera*. The problem is a typical moving camera issue, which has been widely addressed in literature especially for what concerns moving vehicles. However constant speed or at least certain amount of regularity in the motion must often be hypothesised (see e.g. [55]). Anyway many standard methods such as old fashioned background subtraction for change detection cannot be employed in these circumstances, unless an accurate off-line training phase is performed [151].
- *Framing*. As depicted in Figure 3-1, hand detection is often performed on well framed *ad hoc* images. This hardly happens in shooting first person videos, but for specific gestures that could be required for a hypothetical device-free interface.
- *Camera motion estimation* is complicated by the fact that complex roto-translation matrices are involved, as a head often moves sharply and brusquely, yielding high frequencies in the signals. Even integrating data with other sensors such as an accelerometer could be not so easy. The most natural local frame of reference would be the one integral with the device (and thus with the user).
- *Perspective*. The majority of hand detection and gesture recognition methods in literature address the problem from the perspective shown in Figure 3-1, i.e framing the hands' palm, from the front. From a first person perspective, the back

(or, even worse, the side) of the hand is much more often seen, making gesture recognition harder.

- *Real time requirements.* The video processing algorithm must meet real time requirement, while dealing with the limited computational resources and power supply carried by a wearable device.

3.2 A unified hierarchical framework for hand-related methods

As highlighted above, FPV videos offer important benefits and challenges to computer vision . As the main benefit, it is the first time that a wearable device is recording exactly what the user have in front of him. However, being it mobile and wearable, implies highly variable environments with different illumination [139] and without any kind of static reference system [87]. In FPV, unlike in static cameras video processing, both the background and the foreground are in constant motion. An intuitive implication of the FPV camera location is the general belief that the user hands are being constantly recorded and thus the large number of studies based on their gestures and trajectories. The hand presence is particularly important in the field, because, for first time, hand gestures (conscious or unconscious) can be considered the more intuitive way of interaction with the device.

Indeed, hands have played an important role in a large group of methods, for example in activity recognition [70], user-machine interaction [200], or even to infer the gaze of the user [134]. In a recent work, the authors in [19] point out the effects of wrongly assume full time presence of the hands in front of the user. Hand-based methods are traditionally divided in two large groups, namely model-based and data-based methods [155]. The former aims to find the best configuration of a computational hand model to match the image in the video, while the latter are lead by video features such as color histograms, shape, texture, orientation, among others. Figure 3-2 reviews some of the most relevant hand-based papers in FPV.

The classic taxonomy of hand-based methods is too broad and several authors have suggest further extensions according to the features used, the task addressed, and the required sensors. In [22, 19] the authors propose a hierarchical division of the processing steps that can be independently solved e.g. hand-detection, hand-segmentation, hand-tracking, etc. In practice, nowadays, it is common to find a well trained pixel-by-pixel

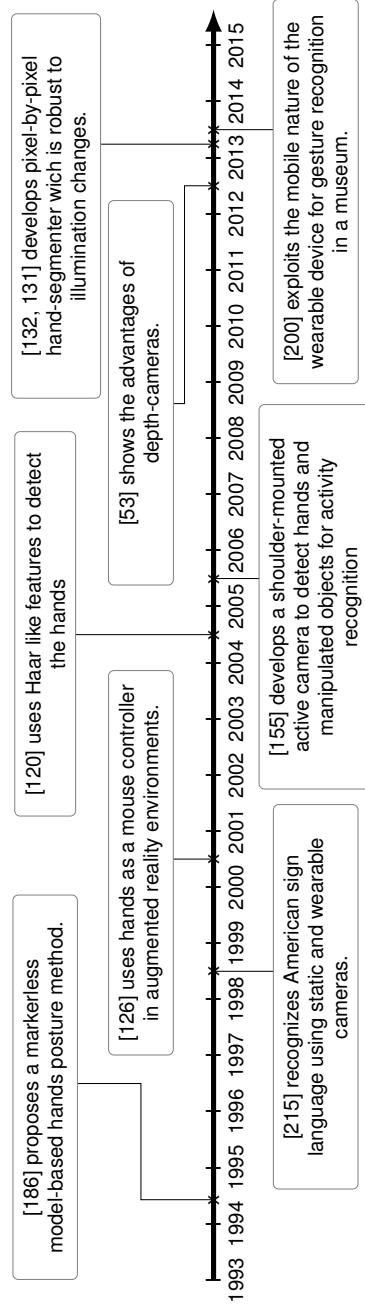


Figure 3-2: Some of the most relevant papers on hand-based methods in FPV

hand-segmenter taking control of the whole system. The segmenter is thus responsible for understanding whether hands are present (after exhaustive negative classification of every pixel), define the shape of the hands (task for which it is trained), and suggest the areas to be tracked keeping record across the frames of the segmented shapes. This approach achieves good results, particularly in finding hand-like pixels; however it rises several issues: i) it exhaustively uses computational resources even when the hands are not present, which is the common case in daily real videos; ii) it could disseminate noise in the system, produced by false-positives; iii) it usually does not exploit temporal correlation between frames. Figure 3-3 shows the possible results obtained by a pixel-by-pixel hand-segmenter. Assuming full time presence of hands may lead to important issues: *i*) possible wrong hand measurements, particularly in no-hand frames, would be propagated to other levels of the system and create wrong conclusions or unwanted feedback from the device and *ii*) unnecessary searching for local features in the image, meaning an inefficient use of computational resources and reduction of the battery life. Note that a pixel-by-pixel hand segmentation of a frame with a resolution of 1280×720 pixels involves 921.600 classification tasks. For practical purposes, some authors reduce the resolution of the images without compromising the quality of the results, however the calculation is still (O^2).

At this point, an intuitive question arises: why to go into detailed pixel-by-pixel classification without knowing first if it is worth it? In order to answer this question, and following the same reasoning of [166] on video analysis, two different tasks should be differentiated, namely *hand-detection* and *hand-segmentation*. The former term has been extensively used (and possibly leading to a misunderstanding) for tasks in which the localization of the hands in the scene was required. In this work, this term refers to a step in which a global answer is given to whether hands are present in the scene or not. The latter aims at delineating the hands in a frame at a pixel level. Both problems are closely related, being possible to use *hand-detection* as a pre-filtering stage for *hand-segmentation* under the framework of sequential classification, in which the output of the first classifier is used to decide whether the second one will be used [252]. Furthermore, different features could be applied in each level, being preferable the use of global features for *hand-detection* purposes [173].

Under this lines of thought, this chapter attempts to highlight the importance of a proper fusion of the above-mentioned tasks in a unified hierarchical structure. In this structure, each level is developed to perform a specific task and provide its results to the rest of the levels. To optimize resources, the structure must switch between its components when it is required e.g. the hand-detector must work only when the hands are not present, while

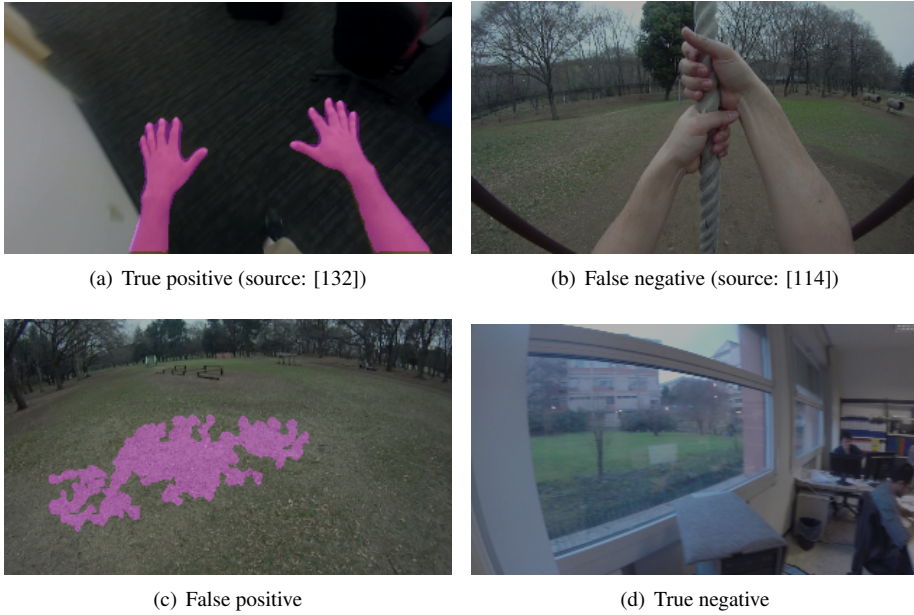


Figure 3-3: Examples of hand-segmentation.

the hand-segmenter (along with a dynamic tracker) is active in the opposite case, but must give back the control of the system to the hand-detector when the hands leave the scene. Finally, the system design must give priority to shared features to optimize extra resources and make the real-time dream closer. The latter is not straightforward and explains why the current methods are usually evaluated in a post-processing framework, restricting the eventual applicability of the field.

The remaining of this section explores some of the ideas behind a unified framework for hand-based methods in FPV, and highlights some of the current challenges of the field for real-life applications.

3.2.1 Levels structure

As already mentioned, one of the main challenges in FPV video analysis is to understand user's hand movements in uncontrolled activities. A proper interpretation of hands (e.g. trajectories, gestures, interactions) opens the door to advanced task such as activity

recognition and user machine interaction, but more importantly it could be the cornerstone to move the wearable devices from experimental state to useful technology. Given the camera location and the user proximity, a system that is able to capture hand gestures could allow smart glasses to do things that other devices like smart-phones cannot. Incidentally, this technology could help to alleviate everyday difficulties of people with visual [149], speaking [215], or motor issues [57]

Current methods have reported remarkable results for tasks like detecting hands presence in front of the user [19], segmenting the silhouette [92, 131, 132, 160], recognizing basic gestures [200, 232], inferring hand posture [191, 192], and identifying whether a hand belongs to the user or to a third person [128]. In general, these subtasks could be considered partially solved, but for an ideal smart wearable camera they are supplied as independent pieces of software resting over different sets of assumptions. Two examples of this are: i) the already mentioned case of the pixel-by-pixel classifier, which despite of being developed to solve the hand-segmentation problem is used to detect, segment and track the hands on its own, at a high computational cost, ii) the hand-detector that, once it is sure about the hands presence, keeps working in parallel to detect if the hands leave, instead of using the detailed results of the hand-segmenter.

To design a unifying system for hand-based methods in FPV it is important to identify some of the more important components, standardize its inputs/outputs and define their role in the overall system. Our approach stands over the task division proposed in [19, 22] and is summarized in the hierarchical structure proposed in Figure 3-4. The figure shows the most important steps in hand-based methods. Some of them could be non necessary for some applications (e.g. not every application needs to identify the left and right hand) or extra levels could be included (e.g. pose recognition). In the bottom part of the diagram are the raw frames, while in the upper part lie the higher inference methods that search for patterns in the hand movements and trajectories. The diagram shows a feature extractor that can be re-used by all the levels: a system that is able to use the same features in multiple levels can save valuable computational resources and processing time.

The diagram makes it evident the importance of a system that is able to optimally decide which is the minimum number of methods running in parallel for each time instance. This switching behaviour is crucial in the bottom levels (hand-detection and hand-segmentation), as well as in the identification and tracking levels to model each hand separately. In the first case, an optimized version of the sequential classifier proposed in [19] is used. The optimized version of this method switches from the hand-detection to the hand-segmentation level whenever the decision moves from “no hands”

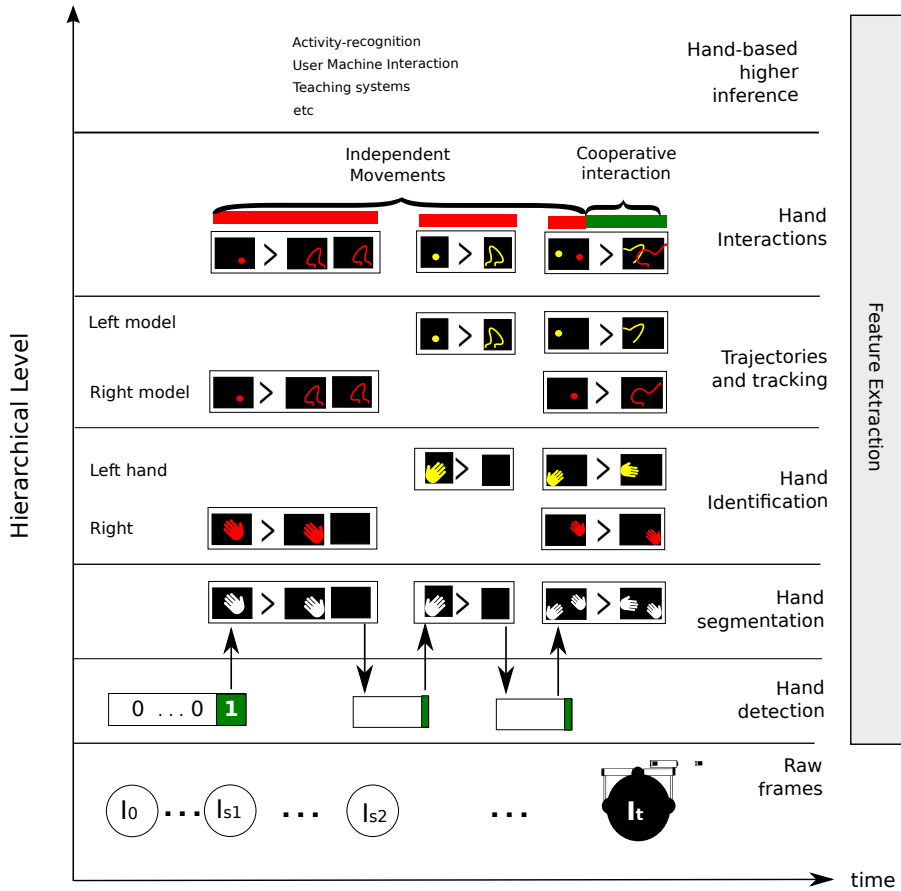


Figure 3-4: Hierarchical levels for hand-based FPV methods.

to “hands”; and from the hand-segmentation to the hand-detection level if there are no more positive pixels in the frame. In the second case, the switching models literature [49, 177] suggests useful strategies to decide which hand models need is to be used at that time. The hand-id (left-right) of the segmented hands emerges as a good switching variable. The hand-id can be estimated using the angles and the positions of the segmented shapes.

The diagram shows a bottom-up design starting from the features and arriving to simpli-

fied trajectories to be used by different applications. However, it is worth to mention that a top-down analysis focused on the application field can remove some of the assumptions in different levels and lead to considerable improvements to the overall performance. As example, the authors in [128] take advantage of psychological experiments among kids and adults to relax the illumination assumption of their proposed method.

In the following, we briefly describe each hierarchical level and discuss some approaches to face the problems that can be encountered. Some results presented in the following chapters are also anticipated.

Hand-detection: This level answers the yes-or-no question of the hands' presence in the frame. The problem is addressed in [19] as a frame by frame classification problem. In their experiments the authors report that the best result is achieved with the combination of Histogram of Oriented Gradients (HOG) features with a Support Vector Machine (SVM). One of the main problems of this frame-by-frame approach is its sensibility to small changes between frames, which makes unstable in time the decisions taken by the classifier. In recent experiments this issue is alleviated using a Dynamic Bayesian Network (DBN) that filters a real valued representation of the SVM classifier. Table 3.1 shows the performance of both approaches (HOG-SVM and the DBN) for each of the 5 testing uncontrolled locations of the UNIGE-egocentric dataset [23]. In the framework of the unified system, the hand-detector must be optimized to detect as fast as possible the frames on which the hands enter the scene. The problem will be addressed in details in chapter 4.

Table 3.1: Comparison of the performance of the HOG-SVM and the proposed DBN.

	True positives		True negatives	
	HOG-SVM	DBN	HOG-SVM	DBN
Office	0.893	0.965	0.929	0.952
Street	0.756	0.834	0.867	0.898
Bench	0.765	0.882	0.965	0.979
Kitchen	0.627	0.606	0.777	0.848
Coffee bar	0.817	0.874	0.653	0.660
Total	0.764	0.820	0.837	0.864

Hand-segmentation: It is probably the more explored problem in FPV. The main task is to delineate the silhouette of the hands at a pixel level. The more promising results are reported in [132, 131, 160] achieving F-scores around 83% under different illumination levels. The main challenge in the pixel-by-pixel approach is the computational complexity of the task, involving the decision for each pixel in each frame. For instance, the camera of the Google glasses has a resolution of 720p and records 30 frames per second, implying 928.800 pixel classifications per frame and a total of 27'864.000 per second of video. A promising strategy to reduce this number is to simplify the frames as SLIC superpixels [2] and classify the simplified image as done in [200]. Within this approach, in [161] an optimized initialization of the SLIC algorithm is proposed. It allows to segment 13 frames per second, while the original SLIC is able to process only 1. Figure 3-5 shows an example of the optimized SLIC algorithm. Chapter 5 is dedicated to methods to be applied at segmentation level.

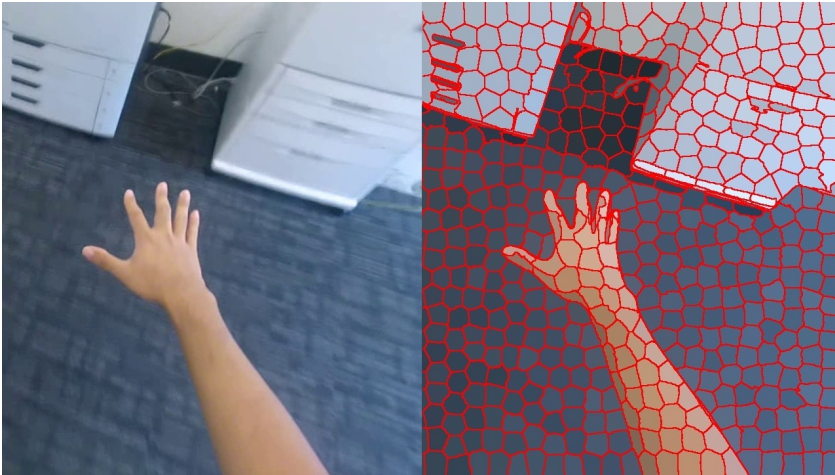


Figure 3-5: Optimized superpixels of a frame with hands [161]

Hand-identification: It is an intuitive but challenging task. The objective is to identify the left and the right hand. The hand-identification problem is extended in [128], proposing a Bayesian method to identify, using the relative positions, the hands of the user as well as the hands of a third person in the video. At this point it is worth to mention the robustness of the proposed hand-detector to the presence of third person hands. However, in the segmentation level, extra effort must be done to segment only the user hands. Assuming a reliable hand-segmentation it is possible to build a simple identifica-

tion system based on the angle and the side of the frame from which the hand appears, as detailed in chapter 6. We found that in realistic scenarios this approach properly differentiate the left and the right hand in almost all the frames at low computational cost. Two difficult scenarios of this approach are: i) The hands are close enough to create a single shape; ii) the appearance of hands is divided by an external object as a bracelet or a watch, creating several hand-like shapes. Figure 3-6 shows a preliminary example of our identification algorithm based on manually segmented shapes.

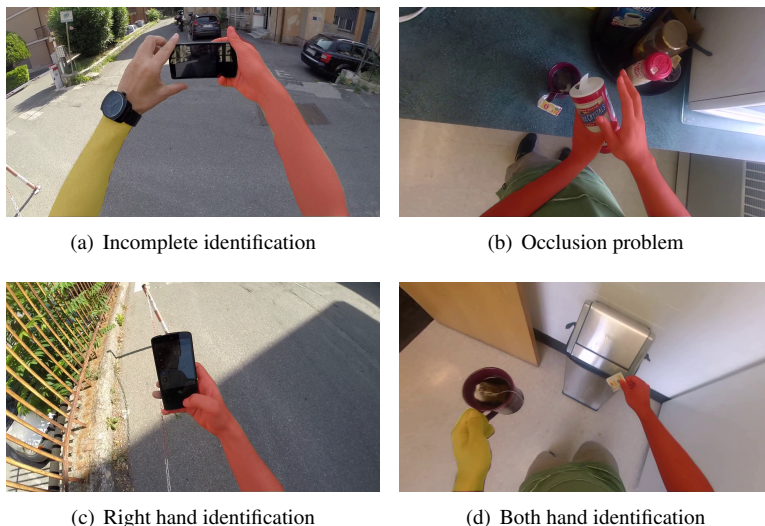


Figure 3-6: Hand-Identification.

Tracking and trajectories: For a wearable camera it is important to record, track and denoise hands trajectories. An intuitive and straightforward approach is to keep history of the hands centroid as done in [160]. However, the use of dynamic filters could help to increase the accuracy of the trajectories, reduce the sampling rate (the lower the sampling rate the closer to real time performance), and manage the switching process between the hand-detection and the hand-segmentation level. Regarding the initial conditions (e.g. initial coordinates of the tracked hand) the best choice is to use dynamic filters like the h-filter, which only requires the empirical distribution of the initial coordinates [25]. The initial distribution can be found using empirical data as shown in [180] (Figure 3-7(a)). Regarding the dynamic model, our preliminary experiments suggest that a simple linear model can achieve promising results for high frequency sampling. However additional

tests are required before a conclusion can be made. Figure 3-7(b) shows frame by frame tracking of a hand center.

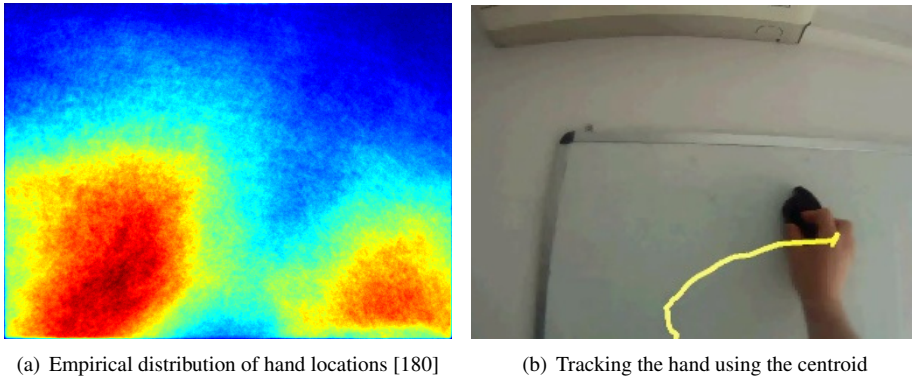


Figure 3-7: Hand-Tracking.

Hands Interactions: Once hands are located and their trajectories are inferred, a possible next step is to understand the interactions between them. For instance, if each hand is performing an independent task (e.g. moving an object, making a gesture) or if both hands are cooperating to accomplish particular objective (e.g. making tea, spreading butter, driving, etc.). At this level important features can be found in the center of mass, the location of the handled objects, the distance between the hands and the relationship between the left and right trajectory. One of the most important works about hand-interaction is [70], where the spatial relation of the hands as well as the handled objects is used to infer some cooking task like (e.g. Pout, stir, spread, etc.).

Hand-based higher inference: in the upper level of the structure are the methods on which the results are built using the information of the hands, some examples are activity-recognition [70] and user-machine interaction [70]. At this point we highlight the relevance of a deep discussion about the real applicability of hand-based methods in FPV and which are the benefits of using single wearable RGB cameras over other systems, like stereoscopic cameras, the Kinect or the Leap-Motion.

On the one side, the miniaturization of RGB cameras makes them the most promising candidate to be worn. On the other side, in exchange of extra battery consumption and

an increase in the size of the device, the use of other sensors can bring important improvements to hand-based methods. For example, the depth component can reduce the complexity of the hand-segmentation level and extra information like the pose of the hands can be straightforwardly inferred. Figure 3-8 shows an example of a RGB and a stereoscopic wearable device. Regarding external devices, like the Kinect or the Leap-Motion, they can, under certain conditions, acquire a wider perspective of the body and, as a result, provide a better understanding of hand movements. As a counterpart, the wearability of these external devices is highly restricted. In summary, external devices can represent a default choice for applications based on static locations without battery restrictions. However, if the application field includes a user moving around with restricted battery availability then a wearable RGB cameras is the most promising option.

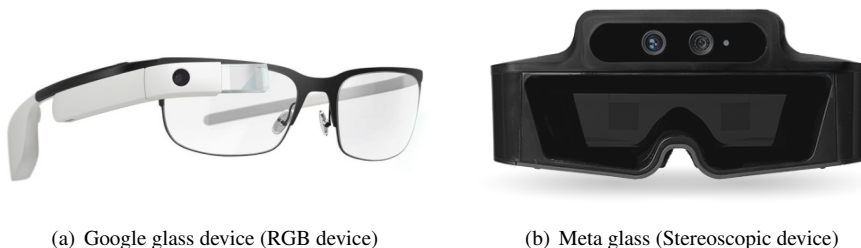


Figure 3-8: RGB and RGB-D wearable devices.

Discussion We have justified the importance of a systemic hierarchical approach to develop hand-related methods. The proposed hierarchical switching structure for hand-related methods in FPV can reduce the computational requirements and under further analysis of the sampling rates could help to reach real time performances. The latter would expand the application areas of FPV video analysis, which by now has been mainly focused on offline processing applications. Each level of the proposed structure addresses a well defined and scoped tasks, allowing us to strategically design each of our methods for a unique purpose e.g. hand-detection, hand-segmentation, hand-tracking, etc. This will be the scope of the remaining chapters of this thesis.

We also point out the convenience of an application-based analysis of the field in order to better understand its real scope and the advantages of a particular sensor choice. We highlight the mobility and the battery cost as the main advantage of RGB cameras. However, if battery restrictions are removed, stereoscopic cameras could lead to more reliable results. From our point of view this discussion must lead the coming developments to

be focused on tasks only achievable by wearable cameras and not by other devices like smart-phones, smart watches or static systems (e.g. Kinect, Leap-Motion).

We consider this as a good moment to analyze the lessons learned by the Glass Project and the current approaches of other companies like Microsoft with the Holo-Lens device. A brief analysis of the media reports about the glass project ending reveals two valuable lessons: i) People would be willing to use these device only if they are able to do things that existing technologies cannot do ii) There are big opportunities for the task oriented approaches, such as medical and industrial applications, on which privacy issues are minimum and the scenarios faced by the user can be partially restricted. On the other hand, the available information of the Holo-Lens project, sketches a device with an exhaustive use of hand-gestures as way of interaction. From this perspective hand-based methods would clearly play an important role in the future of these devices.

In the following section we propose a possible extension of the presented hierarchical framework following the paradigms of Cognitive Dynamic Systems.

3.3 Cognitive framework

As mentioned above, diagram 3-4 makes it evident the importance of a system that is able to optimally decide what methods to run in parallel for each time instance. This switching behaviour is crucial and in fact also implies a feedback-feedforward mechanism across the levels together with some degree of self-awareness for the whole system: i.e. the global algorithm must be able to measure and monitor its performance at each level, in order to devise an optimal behavior. The framework of Cognitive Dynamic Systems provide such functionalities, and will be presented in the following, as an extension of the hierarchy proposed above.

Cognitive science and cognitive neuroscience aim at understanding and clarifying human cognition [141] and, in the last decades, the Signal Processing community has experienced fruitful fertilization from such disciplines, in order to replicate the human characteristic of adaptation. In particular, of primary interest is the human capability of dealing with new situations. This feature can be very valuable especially in non stationary stochastic environments [93] and can be seen as the result of the actuation of the so called cognitive cycle (Figure 3-9). Every step (Sensing, Analysis, Decision and Action) is linked to a learning phase [65]. These concepts have been lately applied to the *computer vision* research field, aiming to design more robust, resilient and adaptable

computer vision systems, by mimicking human capabilities [108] [225], suggesting also how vision should be *active* [89].

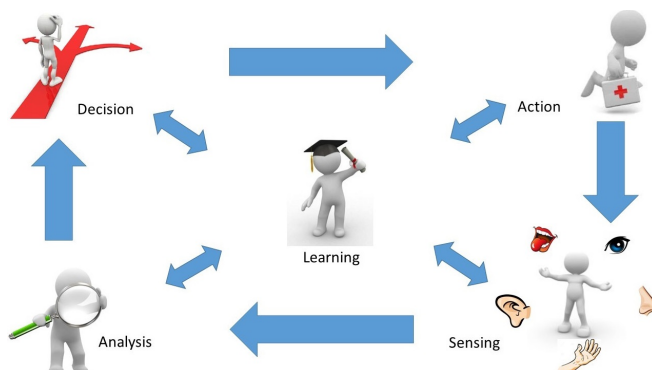


Figure 3-9: Cognitive cycle of a human being.

To this end, Haykin *et al.* have recently proposed a functional decomposition framework [94] to realize a bio-inspired artificial cognitive cycle-based system, based on three main blocks: a *Cognitive Perception unit*, a *Probabilistic Reasoning unit* and a *Cognitive Control unit*. The interactions among these three components and between system and environment are suggested to allow an artificial system to mimic some brain (prefrontal cortex) functions using a probabilistic approach. Abstractness of this architecture makes it scalable to a wide range of applications, however efforts have to be done to translate these general guidelines in working *Cognitive Dynamic Systems* (CDSs): in fact, the main known attempt to implement a real application [69] still grounds on a computational experiment. We propose it here as a container

3.3.1 Functional model

The problem of understanding our hands from a first person perspective is here approached exploiting the framework of Cognitive Dynamic Systems developed in the last few years by Haykin and colleagues [95]. In [94] they introduced a functional representation of a cognitive dynamic system by mimicking the brain functionalities within an artificial information processing framework. This defines an architecture usually structured in two main parts: the *perceptron* with the purpose of perceiving the surrounding environment and generating an internal representation of it, and the *actuator*, which

transfers decision into an action to be performed on the environment from which observation was generated.

More in details, three are the main blocks proposed in [94]: the Cognitive Perceptor (CP), the Cognitive Controller (CC) and the Probabilistic Reasoning Machine (PRM) (Figure 3-10). These blocks form a hierarchical closed-loop feedback system (namely *perception-action cycle*), where both an environmental and an internal reward plays a critical role in how the world is internally represented by means of perception. Each numbered layer in the figure represents a different level of inference that an entity can realize.

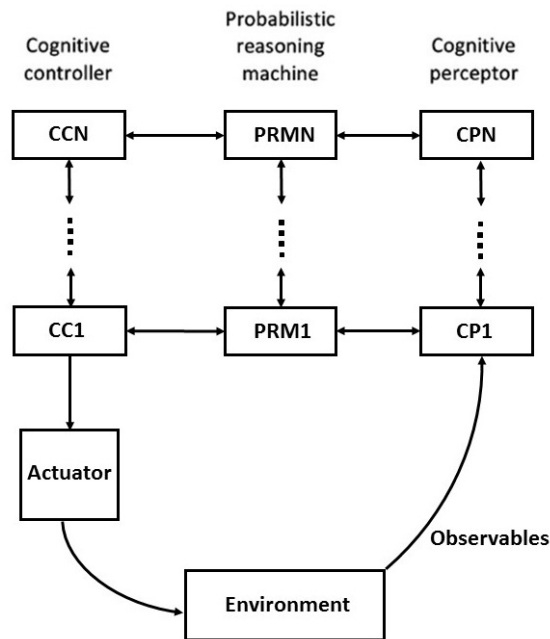


Figure 3-10: Haykin's hierarchy for Cognitive Dynamic Systems. Cognitive Perceptor (CP) unit; Cognitive Controller (CC) unit; Probabilistic Reasoning Machine (PRM) [94].

At the beginning of a perception-action cycle, the Cognitive Perceptor unit processes the measurements coming from the environment (frames from the wearable camera) to generate a representation of the external world. A side computation of a Perceptor unit at a given level, beyond making available to higher inference levels a more symbolic environment description, is to compute a reward estimated by means of the local instan-

taneous perception error. The information coming from the feedback is passed through the system and in particular, reaches the Cognitive Control unit, which chooses the best action in order to maximize the next reward.

The feedback provided to the Cognitive Control unit is compared with previous errors in order to choose the best action to be performed. Such choice is driven by the expected maximization of the next reward that will be observed. In Haykin's model a third component is represented by a feedback modulating unit, the so called Probabilistic Reasoning Module (PRM), aiming at providing a dynamic statistical coupling between perception and action. The PRM keeps track of Perceptor error as well as of other errors related to possible uncertainties in control actions in order to be capable to perform changes of strategies in order to stabilize the overall system.

Each level is structured according to the same architecture and the information passing between layers influences local perception and action strategies and relates to its lower level as at an *environment* that generates observations and whose parameters can be controlled through appropriate learned action knowledge. This resembles the hierarchical structure proposed in the previous section. The cycle continues indefinitely, in order to maintain a dynamic stability. This homoeostatic behaviour is the key aspect that allows a cognitive system to *adaptively* interact with dynamic environments.

In this work, the CDS architecture is applied, with some extensions, to hands-related methods. In general, such methods are not able to directly modify the environment with an action. However, they are given the possibility to modify their internal parameters. This is why we extend the concept of *Environment* with the meaning of both *internal* and *external* environment and refer to the system as to a *Proactive Passive CDS* (PPCDS) architecture: strategies implemented by the control unit do not directly translate into physical actions that can be perceived by hands. On the contrary, they do translate into *cognitive actions* [69] which change the internal capability of the system to adaptively modify its behaviour. In fact, the perceptor may also be viewed as the internal environment for the controller. Incidentally, this is a key property in the CDS paradigm and the foundation of the so-called *two-state model*: one state vector pertaining to the state of the actual environment, and one to the state of the perceptor (also referred to as the entropic state). Indeed, there is no environmental reward involved in the controller, in the sense of external environment, but only internal reward.

More in detail, the self-aware capability of the system itself to pro-actively change its behaviour, as a consequence of freely-varying hands behaviors, can be represented and characterized as separate objects inside the CDS: the perception modules can be asso-

ciated to the representation of hands at different abstraction levels, while the control modules can be associated with the self-aware representation of the system itself.

After a measure is gathered, a probabilistic representation of the external world in the Cognitive Perceptor unit is updated. This representation (Figure 3-11) is afflicted by an intrinsic error (i.e. *perception error*) due to the imperfect state information problem. The indication provided by the perception error, called Perception Entropic State (PES) and denoted with H_p^k , provides a performance measure of the CP.

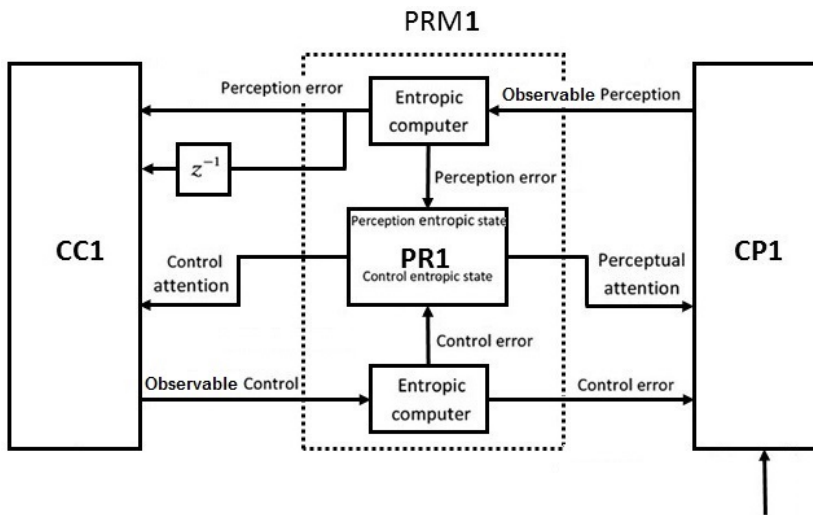


Figure 3-11: Perception-action cycle: details.

By *observing* the current target state, the perception error and its *incremental deviation* [69], the Cognitive Controller unit can identify decision regions in some action space. Actions are associated to regions in a continuous learning process, with the objective of maximizing a parametrized objective function defined over the actual system performance. Parameters consist in different rewards related to the set of actions that CC can take.

Since action selection is afflicted as well by a certain degree of uncertainty, the PRM must receive also an evaluation of the control error state, that is represented by the Control Entropic State (CES) denoted with H_c^k . The PRM block is then delegated to maintain a coherent representation of the global uncertainty level of the CC and CP ensemble. This block has to receive measures from the perception and the control part and glob-

ally evaluate the current system performance. Therefore, the two Entropic Computer blocks act as sensors observing CC and CP estimates as posterior density functions and producing measurement regarding the two-unit dynamic uncertainty levels. As entropic variations gets higher, the Probabilistic Reasoning unit has the role to bring back the system in a homoeostatic stability regimen by using the so called *Perceptual* and *Control Attention*. It is worth noting that a step forward in the direction of a more symmetric architecture of a CDS is here introduced respect the Haykin model.

The general architecture of a CDS outlined above is applied in this thesis to hand-related methods in a First person Vision System as discussed in the following section.

3.3.2 Discussion

The hierarchical nature of Haykin's perception-action cycle (Figure 3-10) is quite evident. As already mentioned, each level is structured according to the same architecture and the information passing between layers influences local perception (and action) strategies and relates to both its lower level (seen as an environment that generates observations) and upper level.

More in detail, the information processed at each Perception module can be associated to a representation of hands at different abstraction levels, enriched with new semantics as inference is refined. It is thus straightforward to associate each level of Figure 3-4 to a Perception unit of Figure 3-10, where the output of a level is exploited at the one above.

However, while this correspondence is manifest for what concerns the Perception side of the functional diagram, it is not as clear yet for what concerns the Control side. As earlier discussed in this section, the key issue introduced by Cognitive Control is the so called two-state model. While the Perceptrons take care of monitoring the state of the external environment (hands), the Controllers examine the state of the Perceptrons (also referred to as the entropic state). Variations of entropy of the CP units triggers an internal reward mechanism which reflects into cognitive action, which are intended to maintain stability of the algorithms in the sensing phase of the cycle. Roughly speaking the system possesses a certain degree of *self-awareness* and is given the possibility to act, modifying its internal parameters. Actions can be selected in a state-less reinforcement learning fashion, as explained in [69] and are anyway dependent by the specific sensing algorithm.

In this thesis the latter mechanism is not implemented, but only hinted while discussing specific methods. For instance, the Dynamic Bayesian Network designed in chapter 4

is in some sense self aware of how good classification of frames is evolving. However, the Control mechanism is not implemented, and the parameters of the algorithm are optimized offline. Another example is the optimization framework presented in section 5.2.2, where the residual error of the Superpixel algorithm is employed as a measure of performance.

Chapter 4

Hand Detection

Classifying frames, or parts of them, is a common way of carrying out detection tasks in computer vision. However, frame by frame classification suffers from sudden significant variations in image texture, colour and luminosity, resulting in noise in the extracted features and consequently in the decisions taken. Support Vector Machines have been widely validated as powerful tools for frame by frame detection of non-separable datasets, but are extremely sensitive to these variations between adjacent frames, creating as consequence sudden flickering in the classification results. This chapter proposes a Dynamic Bayesian Network to smooth the classification results of Support Vector Machines (SVM) in detection tasks¹.

Classification-for-detection is a widely studied area in computer science. Its main objective is to decide whether a particular object O is present in the environment. The variety of objects to detect is broad and multiple applications were investigated, such as pedestrian detection [229, 52, 63, 230, 83], hand detection [19], face detection [113], intrusion detection [165], among others. In computer vision, a common approach to detect O in an image (or in a video sequence) is to exploit a classifier under a supervised framework, using a balanced training dataset with O and non- O sample images. In particular, samples (especially non- O) should be sufficiently heterogeneous in order to allow a good discrimination of the two classes.

The problem of *detection* is often related to the *localization* of an object in a frame

¹The results presented in this chapter, together with the dataset, have been published in [21] and [23]

(equivalently, the problem can be formulated as the detection of an object in a localized sub-part of the frame). This task is frequently faced in an iterative way, classifying images framed by a sliding window of different sizes moving across the image. All these approaches are derived from the seminal work by Viola and Jones [228], in turn inspired by [80]. Despite being computationally expensive, these strategies are widely accepted as a powerful strategy for object detection and localization.

Instead of classifying raw images directly, it is usually preferable to classify extracted features. Multiple alternatives have been previously evaluated depending of the detection goal. An extensive literature is available: some of the more popular image features are color histograms [107, 106] to detect parts of the human body, global features as GIST [166] to detect general properties of the scene, rotation and scale invariant features as SIFT [137] to detect and identify multiple objects at different scales and positions, and shape features as Histogram Oriented Gradients (HOG) [52] to exploit particular characteristic in the shape of objects. Recent approaches use mixtures of features at different levels under the deep learning framework [112]. Regarding the classifiers, multiple alternatives are available. However, a general consensus has been achieved about the powerful combination between HOG and Support Vector Machines (SVM), particularly for non separable datasets [52, 19].

These approaches are developed and trained without using temporal information, therefore their application in video sequences is usually carried out as a naive frame by frame classification [202], which is extremely sensitive to small frame-to-frame features' variations. To alleviate this problem some researchers smooth the features to reduce their spatial and temporal variations [102]. Temporal stability of the detections can be considered as a common goal for many video processing applications, thus a dynamic smoothing is typically more important than the spatial approach. For instance, this is done for depth videos [179] and for RGB first-person videos [14] at pixel level.

Existing literature points out several promising applications of this video perspective. Among them, hand-based methods stand as the most explored ones, aiming to exploit the conscious or unconscious hands movements for performing higher inference about the user [20] as in activity recognition [72, 180] and user-machine interaction [200]. A common practice in FPV is to assume that hands are always recorded by the camera and, as a consequence, they can be located and tracked to infer more complex information. As it can be concluded after a quick scan of uncontrolled datasets like Disney [71] or UTE [87], this assumption is not entirely true. In fact, the predominance of one or the other type of frames (with/without hands) in a video sequence is not a consequence of the advantageous camera location but also of the activity performed e.g. hands are more

frequent when the user is cooking than when he is walking in the street.

Despite the practical advantages of assuming full time hands presence, this fact introduces important issues when the proposed methods are applied on uncontrolled videos, for example wasted computational resources or noisy signals in the hand-segmentation stage, that could be propagated to other levels of the system. The authors in [19] propose a characterization of the two distinct problems, namely *hand-detection* and *hand-segmentation*, and combine them in a sequential structure to improve the overall system performance. Following the definition of [19], the *hand-detection* level answers the yes-or-no question of the hands' presence in the frame using global features and classifiers, while the *hand-segmentation* level locates and outlines the hands' region in a positive frame using low level features like color under an exhaustive pixel by pixel classification framework [134, 200, 160].

Regarding data availability, there are several FPV datasets available for research purposes. In general the technical characteristics of these datasets are similar and the videos are carefully recorded to guarantee the basic requirements identified by Schiele in 1999 [197]: i) Scale and texture variations, ii) Frame resolution, iii) Motion blur and iv) Hand occlusions. Undoubtedly, these requirements are important, but, under the light of the recent technological trends, some extra characteristics must be taken into account. An example is the necessity of balanced datasets in terms of hands presence as described by [19] and [21], to face the *hand-detection* problem under a classification framework. A balanced dataset is a realistic assumption for wearable devices and could lead to important improvements in the battery life, as well to the performance of higher-level methods like hand-based activity-recognition[70] and user-machine interaction [200]. It is worth to mention that, as shown in section 4.1, existing datasets does not guarantee this condition, which makes them inappropriate to face the classification problem of the *hand-detection* level.

This chapter focuses indeed on *hand-detection*, and its contributions are three-folded: i) It presents the UNIGE-HANDS dataset for *hand-detection*, which guarantees a balanced number of frames with and without hands in 5 realistic locations, as well as changes in illumination, camera motion and hands occlusions.² ii) Multiple *hand-detectors* (feature-classifier) are evaluated over the dataset, following [19], without considering the temporal dimension of the data. iii) The best *hand-detector* (HOG-SVM) is extended using a Dynamic Bayesian Network (DBN), which is tuned to smooth the decision process. The presented method improves the performance of [19], taking advantage of the temporal dimension of the video, and of [21], tuning the parameters through an heuristic

²[Dataset:] <http://www.isip40.it/resources/UNIGEhands>

tic optimization. The computational complexity of the proposed approach is taken into account by filtering the classification certainty of the SVM directly, instead of a generic *multidimensional* array of features. Namely, we perform the filtering step at a higher hierarchical level in the estimation process as depicted in Figure 4-1.

The remainder of this chapter is organized as follows: Section 4.1 summarizes the evolution of *hand-detection and segmentation* methods and shows why the existent datasets are not suitable to solve the *hand-detection* problem. Section 4.2, presents the UNIGE-HANDS dataset and evaluates multiple frame by frame *hand-detectors* (combinations of image features and classifiers). Later, section 4.3 extends the state-of-the-art method using a DBN and briefly describes each of its components. Section 4.4 tunes the DBN using a classic Genetic Algorithm (GA) and the Nelder-Mead simplex (NM) algorithm in a cooperative fashion. Subsequently, the performance of the DBN is evaluated, and under the light of the results, the challenges offered by the UNIGE-HANDS dataset are presented. Finally, in section 4.5 conclusions are drawn and some lines for future research are proposed.

4.1 State of the art

In the recent years, thanks to the growing availability of FPV recording devices, the number of methods to process related videos, as well as datasets, has increased quickly. To the best of our knowledge a total of 16 datasets have been published between 2005 and 2014, each of them especially designed to face a particular objective, i.e. Object recognition and tracking, activity recognition, computer machine interaction, video summarization, physical scene reconstruction, and interaction detection. Table 4.1 summarizes the existent datasets and their basic characteristics. The table also highlights the evolution of the camera location, moving from shoulder, to head-mounted. This trend can be explained by the interest of technology companies to develop smart glasses and action cameras.

Existing datasets can be divided in two main groups: datasets where hands are almost always present, and datasets where hands barely appear. The first group has been used for object recognition (Mayol05, Intel), activity recognition (Kitchen, GTEA11, GTEA12) and user-machine interaction (Virtual-Museum). These datasets are usually recorded in fixed locations, like a kitchen or the office, while the user performs different tasks. Regarding the *hand-detection* problem, these datasets are not suitable because it is not possible to extract a set of negative samples in the same location and light conditions as

Table 4.1: Current datasets and sensors availability [22].

		# Objects				C. Location			
		Year	Objective	Activities	Objects	Num. of People	Shoulder	Chest	Head
MayoI05	[155]	2005	O1	5	1		✓		
Intel	[180]	2009	O1	42	2		✓		
Kitchen.	[210]	2009	O2	3	18				✓
GTEA11	[70]	2011	O2	7	4				✓
VINST	[3]	2011	O2		1			✓	
UEC Dataset	[115]	2011	O2	29	1				✓
ADL	[182]	2012	O2	18	20			✓	
UTE	[87]	2012	O4	4				✓	
Disney	[71]	2012	O6		8				✓
GTEA gaze	[72]	2012	O2	7	10				✓
EDSH	[132]	2013	O1	-	-	-			✓
JPL	[193]	2013	O6	7	1				✓
Virtual Museum	[200]	2013	O3	5	1				✓
BEOID	[54]	2014	O2	6	5				✓
EGO-GROUP	[7]	2014	O6		19				✓
EGO-HPE	[6]	2014	O1		4				✓

* **Objectives:** [O1] Object Recognition and Tracking. [O2] Activity Recognition. [O3] User-Machine Interaction. [O4] Video Summarization. [O5] Physical Scene Reconstruction. [O6] Interaction Detection.

the positive ones to train binary classifiers. The second group of datasets are frequently used for activity recognition (VINST, UEC, ADL), video segmentation (UTE, BEOID), Interaction Detection (Disney, JPL, Bristol, EGO-GROUP, EGO-HPE). In general these datasets are large and contain sequences of the user moving through several realistic locations. The number of frames with hands is low compared with the length of the videos, and the locations with frames with hands are sparse, making impossible to extract a large

enough balanced training set with similar locations. It is worth to highlight the importance of having frames with and without hands in the same location. This would lead the classifiers to learn patterns related with the hands presence and not from the changes in the location.

In general, all of these datasets guarantee the basic requirements identified in [197]: i) Scale and texture variations, ii) Frame resolution, iii) Motion blur and iv) Hand occlusions. However, they are mainly formed by clips framing users' hands, which is not realistic and makes it infeasible to evaluate *hand-detection* methods. Recently, the authors in [220] perform a comparative study about the characteristic of FVP and Third Person Vision (TPV) datasets. The authors found that using blur, illumination changes and optical flow as input features is possible to differentiate with 80.9% of accuracy between FPV and TPV datasets.





















According to [155], known for being the first public dataset in FPV for object recognition, *hand-detection/segmentation* methods can be grouped in two: model-driven and data-driven. The former uses a computerized model of the hands to recreate the image of the videos [217], while the latter exploit image features to infer about hand location, shape and position [134, 200, 160].

Regarding *hand-detection*, a data-driven sequential classifier is proposed in [19], which in a first stage detects hands, and in a second stage finds the hands silhouette at a pixel level *only for positive frames*. In their experiments, the authors report the performance of multiple classifiers and image features, to finally conclude that the best-performing combination is HOG plus SVM achieving 90% of true-positives and 93% of true-negatives. The authors in [246] follow a color-based approach in the same line of [134] which, as is shown in [19], could introduce noise in the results under large illumination changes. To conclude the overview, [128] proposes a probabilistic approach to detect if the hands in the video belongs to the user or to another person.

4.2 UNIGE-HANDS dataset

The UNIGE-HANDS dataset for *hand detection* is a set of FPV videos, carefully recorded to guarantee a good balance between frames with hands and without hands, and offers challenging characteristics such as changes in illumination, camera motion and hand occlusions. The UNIGE-HANDS dataset, videos and ground truth, is distributed for public use. The dataset contains videos recorded in 5 uncontrolled locations (1. Office, 2. Coffee Bar, 3. Kitchen, 4. Bench, 5. Street). Each location in the dataset is in turn

Table 4.2: Examples of the dataset frames.

		Office	Street	Bench	Kitchen	Coffe Bar
Training	Hands					
	No Hands					
Testing	Hands					
	No Hands					

divided in training and testing videos. Table 4.2 shows some examples of the frames in each location.

To record the dataset we used a *GoPro hero3+* head mounted camera with a resolution of 1280×720 *pixels* and 50 *fps*. The whole dataset, including training and testing videos, contains one-hour and thirty eight minutes of video. In total, the training videos have 37.21 and 37.63 minutes of positives and negative sequences, respectively. The training videos for each location are formed by 2 positives and 2 negatives videos approximately 3.34 minute-long each (10020 frames). Regarding the testing videos, they comprise 12.6 minutes of positive and 12.7 minutes of negative segments. The testing video of each location lasts approximately 4 minutes (12000 frames), changing from positive to negative in intervals of about one minute.

Following the procedure described in [19], multiple combinations of classifiers and video features are evaluated over the new dataset. The classifiers are: Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF). The video features are: Histogram of Oriented Gradients (HOG), the global scene descriptor GIST, three color spaces (RGB, HSV, LAB) and its concatenation (RHL). The SVM uses a linear kernel with a regularization parameter $C = 1$. To compute the features, each frame is compressed to 200×112 *px*. The HOG extractor uses a block size of 16 *px*, a cell size of 8 *px*, and 9 directional bins, while color features are estimated over a grid of 25×14 cells (which are indeed 8×8 *px* cells).

Table 4.3: Performance of the proposed *hand-detectors*.

		True Positives			True Negatives		
		SVM	DT	RF	SVM	DT	RF
10-fold	HOG	0.89	0.77	0.81	0.90	0.76	0.88
	GIST	0.78	0.75	0.72	0.79	0.74	0.88
	RGB	0.77	0.72	0.73	0.77	0.73	0.86
	HSV	0.72	0.76	0.78	0.72	0.78	0.88
	LAB	0.75	0.85	0.89	0.75	0.85	0.90
	<i>RHL</i> ¹	0.78	0.85	0.86	0.77	0.85	0.91
Training	HOG	0.93	0.80	0.83	0.91	0.80	0.91
	GIST	0.83	0.81	0.80	0.82	0.80	0.91
	RGB	0.82	0.76	0.78	0.82	0.78	0.90
	HSV	0.77	0.80	0.83	0.78	0.82	0.92
	LAB	0.80	0.88	0.92	0.79	0.88	0.93
	<i>RHL</i> ¹	0.81	0.87	0.88	0.81	0.87	0.93
Testing	HOG	0.76	0.72	0.70	0.84	0.75	0.83
	GIST	0.51	0.51	0.43	0.67	0.58	0.70
	RGB	0.57	0.60	0.57	0.72	0.64	0.68
	HSV	0.60	0.65	0.65	0.66	0.67	0.75
	LAB	0.56	0.75	0.74	0.69	0.73	0.77
	<i>RHL</i> ¹	0.57	0.74	0.71	0.68	0.71	0.78

¹ *RHL* is the concatenation of RGB, HSV and LAB.

Table 4.3 reports the performance of each feature-classifier combination under three different evaluation strategies: i) *Cross-validation*: 10-fold validation performed using the training frames as described in [19]. This procedure requires to train each classifier 10 times using 90% of the sampled frames for training and 10% for testing. The reported performances are computed using as training data 2203 frames with hands and 2233 without hands. These frames are gathered by sampling the training videos once every second. ii) *Frame by frame in the training videos*: The classifier is trained using the sampled frames, and tested in the remaining frames of the training videos. This approach only requires to train the classifiers once, which is particularly useful for the tuning

procedure explained in section 4.3. iii) *Frame by frame in the testing videos*: The classifier is trained in the sampled frames but tested in the testing videos. This approach is the more realistic to test the classifier because, despite being recorded in the same locations, the testing videos are completely independent of the training stage.

The first finding in the table is that the performance reported in the 10-fold is slightly lower than the reported by the authors in the original paper. This reduction is explained by the challenges intentionally introduced in the dataset, namely the illumination changes and the number of locations. The 10-fold performance validates the conclusion of [19], where HOG-SVM stands as the best performing combination, although here the LAB-RF achieve a similar performance. In general the first (10-fold) and second group (Training) of performances are similar, which validates the use of the second strategy to tune the DBN in a computationally efficient way. To evaluate the performances in a dynamic perspective (video sequences), each frame of the testing videos is classified using the already trained *hand-detectors*. In general, these performances are lower than the first and second group, showing the importance of the testing videos. The optimistic performance reported by the cross-validation method is extensively explained in the literature and is known as the bias in the cross validation procedure [17].

It is worth to note that HOG-SVM is the best performing combination in all the evaluation strategies, particularly in the third one (*testing videos*), where it achieves 76% of true-positives and 84% of true-negatives. Noteworthy is also the performance of LAB-RF, which despite of being lower than HOG-SVM in the testing case, could offer important cues for to improve computational efficiency of the hand-detector. In addition to the outstanding classification rate, the HOG-SVM combination shows an extra advantage, given by its theoretical formulation, which naturally provides could provide a real valued confidence measurement of hands presence. The latter is particularly important in the dynamic approach as explained in the next section. The remainder of this chapter is focused on the HOG-SVM detector and the dynamic strategy to improve its results.

4.3 Hand-detection DBN

In this section, a SVM-based detector is extended with dynamic information using the DBN proposed in Figure 4-1. The figure sketches a multi-level Bayesian filter for state estimation where the bottom level contains the raw images and the upper level the filtered decision. In general, the measurement (z_k) is a real valued representation of the SVM classifier applied to set of features F_k extracted from the k^{th}

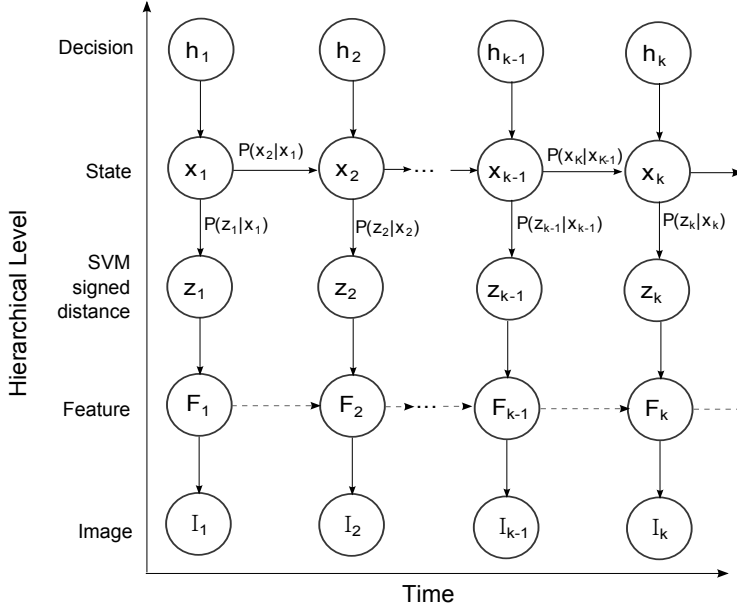


Figure 4-1: Dynamic Bayesian Network for smoothing the decision process.

frame I_k . The state $x_k \in R^2$ is the filtered SVM confidence enriched with its speed: $x_k = [f(F_k), \dot{f}(F_k)]$. Finally, h_k is the binary decision based on the filtered value of the state: $h_k = \text{sign}(x_k[0] + t_h)$. The latter allows t_h to take values different from 0, in order to capture the effects of the dynamic filter to the decision threshold of the SVM. The dotted line of Figure 4-1 is drawn to illustrate the possible filtering at features level, as discussed at the beginning of the chapter. However, in our case only the state of the system is filtered. The remaining part of this section briefly introduces the SVM notation, the dynamic filtering, and the heuristic tuning of the DBN parameters. See [21] for extra details about the mathematical formulation of the SVM and the dynamic filter.

i) Support Vector Machine: Let's assume a dataset composed by N pairs of training data: $(F_1, y_1), (F_2, y_2), \dots, (F_N, y_N)$, with $F_i \in R^p$ and $y_i \in \{-1, 1\}$. Equation (4.1) defines a classification hyperplane and equation (4.2) its induced classification rule, where β is a unit vector. Assuming that the classes are not separable then the values of β and β_0 are the solution of the optimization problem given by (4.3), where $\xi = (\xi_1, \xi_2, \dots, \xi_N)$ are referred to as the slack variables, and K is constant.

$$\{F : f(F) = F^T \beta + \beta_0 = 0\} \quad (4.1)$$

$$G(F) = \text{sign}(f(F)) = \text{sign}(F^T \beta + \beta_0) \quad (4.2)$$

$$\min_{\beta, \beta_0} \|\beta\| \text{ subject to: } y_i(F_i^T \beta + \beta_0) \geq 1 - \xi_i, \forall i, \quad (4.3)$$

$$\xi_i \geq 0, \sum \xi_i \leq K$$

For the *hand-detection* problem we use the signed distance to the classification hyperplane, $f(F_k)$, as the measurement ($f(F_k)$ is denoted as z_k in the DBN diagram, using the common notation for measurements in Bayesian filtering), where F_k is a global feature extracted from the k -th frame. It is important to note that the signed distance to the decision boundary $f(F)$ gives both a description of the result $G(F)$ of the classification (i.e. $\text{sign}(f(F))$) as well as its level of certainty. In addition, augmenting the state with the speed ($\dot{f}(F)$) would allow us to control sudden variations of such confidence. In some sense the DBN is thus self-aware of how good the classification is evolving, and can introduce some feedback mechanism to compensate for poor classification. This can be seen as an implementation of the reward mechanisms described in chapter 3, where framework of Cognitive Control was introduced.

ii) Kalman Filter: Once the certainty level from the SVM is extracted, we address the problem of transferring and stabilizing that measurement from time to time. This strategy aims to reduce the number of wrong decisions caused by little variations in the features between frames. For this purpose we use a discrete linear Kalman filter. In general notation, the process and measurement model is given by (4.4), where $x_k \in \mathbb{R}^n$ is the state and $z_k \in \mathbb{R}^m$ is the measurement. The matrix $A_{n \times n}$ relates the state at previous step, x_{k-1} , with the state at current step, x_k . The matrix $H_{m \times n}$ relates the state with the measurement. Finally, w and v are the process and measurement noise respectively, which are assumed Gaussian with zero mean and covariances $Q_{n \times n}$ and $R_{m \times m}$ respectively. In our case $n = 2$ and $m = 1$, x_k is then a two dimensional vector, whose first component contains the decision certainty and the second its changing speed. At this point the binary decision, h_k , is calculated using $\text{sign}(x_0 + t_h)$, which as already mentioned, is equivalent to allow changes in the original SVM decision threshold.

$$x_k = Ax_{k-1} + w_k, \quad z_k = Hx_k + v_k \quad (4.4)$$

Based on these equations, the prediction stage is given by (4.5), which, using the current values of \hat{x}_{k-1} and P_{k-1} approximates their next values \hat{x}_k^- and P_k^- . P_k is the error covariance at time k and \hat{x} is an estimator of x .

$$\hat{x}_k^- = A\hat{x}_{k-1}, \text{ and } P_k^- = AP_{k-1}A^T + Q \quad (4.5)$$

Once a new measurement is available the values of x_k and P_k are updated using (4.6), where K is known as the Kalman gain.

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \\ P_k &= (I - K_k H) P_k^- \end{aligned} \quad (4.6)$$

At this point it is possible to use \hat{x}_k and P_k for a new predicting stage. In the DBN, \hat{x}_k is a two dimensional vector, and is used to decide the value of h_k by taking $\hat{h}_k = \text{sign}(\hat{x}_k[0])$ (as already mentioned, this is equivalent to have a decision threshold equal to 0).

Ultimately, extracted features which are really close to the decision boundary can jump from one side to the other in consecutive frames, being their (signed) measured distance z_k slightly positive or slightly negative. Filtering such a distance together with its variation significantly reduces binary classification hopping as shown in the next section.

iii) Tuning the DBN: Within the general framework presented above, there are two sets of parameters to be estimated. The first set are the parameters defining the classification hyperplane of the SVM, namely β and β_0 . These parameters are estimated using the training dataset and the SVM implementation of sklearn library [178] for python . The second set are the Kalman filter parameters and the decision threshold, namely Q , R and t_h . The tuning of the parameters of a dynamic filter is a widely explored field, and different approaches are usually followed according to the requirements of the system, restrictions in the measurements, and the ground truth availability.

Following the work of [1] the main idea behind the tuning procedure is to decompose the joint distribution of the system $p(z_{0:T}, x_{0:T}, h_{0:T})$, using the Bayesian notation, and, given the data availability and characteristics of the marginal distributions, find the op-

timal values of the parameters. In our case the more appropriated approach, taking advantage of the ground truth, and given the non-differentiability the binary decision boundary, is to minimize the residual prediction error in an heuristic way. With this in mind we look to minimize the squared error of the DBN decisions, defining the optimization problem as (4.7).

In our case the more appropriated approach, taking advantage of the ground truth, and given the non-differentiability of $h_t = \text{sign}(x_t)$, is to minimize the residual prediction error in an heuristic way. In our formulation we look to minize the squared error of the DBN decisions, so the optimization problem could be stated as (4.7). A common approach to solve this problem is to use a method like Nelder-Mead simplex (NM) algorithm to find a solution close to an initial solution. NM is a numerical method widely used to solve optimization problems when there is not knowledge about the derivatives of the objective function. It has been proven to be a good approach finding local optimals close to an initial point. Under the absence of intuition about the initial point, the authors in [45] suggest to use an combination of a basic Genetic Algorithm (GA), to find some initial points, and latter improve them using NM.

$$< Q, R, t_h > = \arg \min_{Q, R, t_h} \sum_{k=0}^T (h_k - \hat{h}_k)^2 \quad (4.7)$$

This optimization problem is usually faced using a method like the Nelder-Mead simplex (NM) algorithm to find a optimal solution close to an initial solution. Under the absence of intuition about the initial point, the authors in [45] suggest to use a combination of a basic Genetic Algorithm (GA), to find some initial points, and later improve them using NM. In our case we design a classical GA where each genome is an instance of the parameters to be optimized, and each generation contains 100 genomes. The algorithm starts with an initial population of 100 random genomes to select the best 4, named parents. The subsequent generation is then composed by two parts. The first 64 genomes are crossovers: combinations of the parents, and the remaining 36 genomes are mutations: random modifications of the parents. In the mutation stage, the parents are selected randomly, and each element is modified with a probability of 0.5. Once the algorithm achieved an acceptable decaying rate of the objective function, the 4 best genomes among all the generations are used as initial points in NM. The best of the NM results is selected as the optimal combination.

4.4 Results

The results presented in this section are two-fold. First, we introduce two different optimization cases for the proposed filter. Second, we show how the DBN approach considerably improves the performance of the naive HOG-SVM detector (detailed results are presented for the best optimization problem only, but they enhancement is significant even in the worst case).

The Kalman filter is formulated as a kinematic model of the “position” (distance to the separation hyperplane) enriched with the speed, and a sampling rate Δ_t . Equation (4.8) shows the process and measurement model, where $w_k \sim \mathcal{N}(0, Q)$ and $v_k \sim \mathcal{N}(0, r)$. There is not exact knowledge of the differential equation regulating the dynamic process, thus it is not possible to precisely state the law that moves the decision back and forth the decision boundary. Actually, it is not known if such differential equation exists or can be solved in closed form. For this reason, we borrow from physics a constant force model, which we think is a good starting point. This is equivalent to suppose there is some constant (oscillating) force that keeps the features away from the decision hyper-surface or make them cross it, with a constant acceleration a .

$$\begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + w_k, \text{ and } z_k = [1, 0] \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + v_k \quad (4.8)$$

More in detail, the first equation in (4.8) models an exact constant acceleration, where a is the *effect* of a control input which generates exactly the time-dependent noise term. On the other hand, employing a state augmented with the second derivative as well, would allow small variations of a , accounted for in the noise term w_k . In our optimization framework, this is equivalent to parametrize each of the elements of Q . In this case the genomes are given by instances of $[Q_{1,1}, Q_{1,2}, Q_{2,1}, Q_{2,2}, r, t_h]$, and the elements of each crossover are selected randomly from one of the current parents. In the second optimization case, we suppose instead that the acceleration is constant, and the matrix Q is factorized isolating the sampling rate as in (4.9). In this case the genomes are of the form $[q, r, t_h]$ and the crossovers are all the possible combinations of the current parents. To keep control of the search space we bound the elements of Q as well as q and r to move between 0 and 1000. The decision criteria t_h is bounded between -0.5 and 0.5 . The number of iterations is set to 20. To evaluate the objective function for each combination we merge the testing videos and calculate the overall accuracy under

the second strategy of Table 4.3. We point out that the second strategy is used because of computational advantages and to keep the training and tuning process independent of the testing videos.

$$Q = q * \begin{bmatrix} \frac{\Delta_t^4}{4} & \frac{\Delta_t^3}{2} \\ \frac{\Delta_t^3}{2} & \Delta_t^2 \end{bmatrix} \quad (4.9)$$

From the tuning process of the two cases presented above we found that the best accuracy is achieved for the genomes

$$[+1.15e^{-9}, +1.39e^{-7}, +8.72e^{-8}, +2.07e^{-5}, +60.78, -7.63e^{-2}]$$

$$[+0.039, +32.54, -0.151]$$

for the general and factorized case respectively. The final number of frames misclassified by each case are 3505 and 3391 over a total of 220610. As a comparison, the total of misclassified frames using naive HOG-SVM is 18211. It is remarkable the fact that both optimization scenarios reach a similar value in the objective function, validating the use of the constant acceleration model to reduce the flickering in the decision. The remaining of this section present more in detail the results achieved by the factorized case over the testing videos. Figure 4-2 shows, in red line, the measurement z_k and, in blue line, the filtered state x_k . The horizontal axis is the decision threshold. Taking the value of 4, 5, 6 (-4, -5, -6) the figure shows the ground truth, the decision of the HOG-SVM method and DBN, respectively. These decisions takes positive values if there are hands and negative if not. The noisy movements of z_k confirm the dependence of the measurement to little changes between frames. As it is intended, the Kalman filter reduces the noise and preserve the trend of z_k .

It can be noted from the pointwise decisions of HOG-SVM (Dec. HOG-SVM) that it is difficult to obtain continuous segments of the video with or without hands. This effect is the consequence of the measurement noise changing frequently the sign of z_k . Once the noise is reduced using the DBN, the decisions stabilizes and continuous segments appear. It is particularly remarkable the performance of the DBN in the *Office* and the *Bench* sequences. However, because of the poor performance of the HOG-SVM, the DBN misclassifies long segments in the *Kitchen* and the *Coffee bar* sequences. The poor performance of the HOG-SVM in these sequences can be explained by the 3D perspective created by the table, which creates lines in the same positions and directions of those created by the hands.

Table 4.4 summarizes the performance for each location of the dataset. In total the DBN improves the number of true-positives by 5.6 percentage points, moving from 76.4%

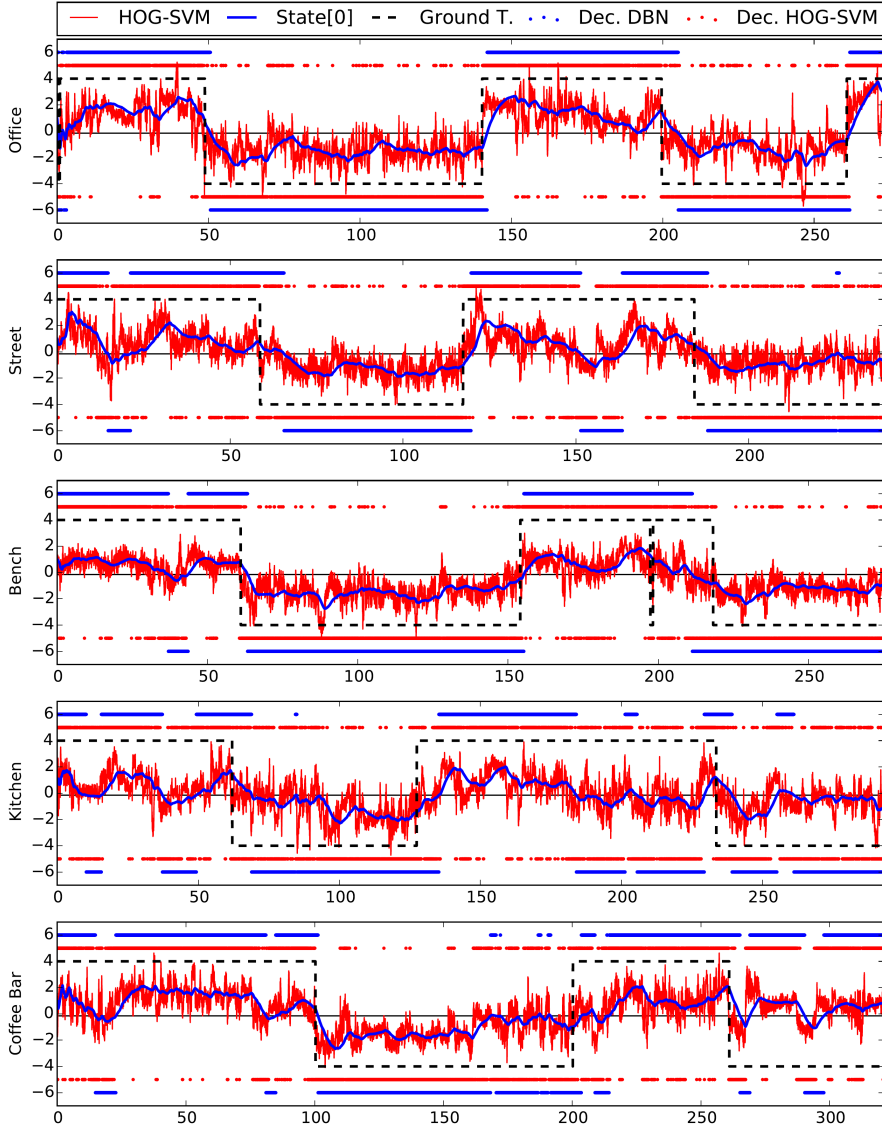


Figure 4-2: Performance of the DBN in each of the locations in the UNIGE-HANDS dataset.

Table 4.4: Comparison of the performance of the HOG-SVM and the proposed DBN.

	True positives		True negatives	
	HOG-SVM	DBN	HOG-SVM	DBN
Office	0.893	0.965	0.929	0.952
Street	0.756	0.834	0.867	0.898
Bench	0.765	0.882	0.965	0.979
Kitchen	0.627	0.606	0.777	0.848
Coffee bar	0.817	0.874	0.653	0.660
Total	0.764	0.820	0.837	0.864

to 82.0%. The number of true-negatives is improved by 2.7 percentage points, changing from 83.7% to 86.4%. The only performance which suffer a reduction is the true-positives of the *Kitchen*. This reduction is explained by a long segment (Figure 4-2 between second 150 and 250) in which the measurements are switching between positive and negative values with no trend. An extra analysis of the corresponding video validates the hypothesis of the 3D perspective created by the used table, and points out an interesting research idea regarding the fusion of color and shape features to deal with this kind of scenarios. A similar case is found in the last segment of the *Coffee Bar* location, which despite showing an improvement of 0.7 percentage points in the true-negatives, is one of the worst performing. In all the other scenarios the improvement is remarkable. Particularly, the true-positives of the *Bench* location is the one with the largest improvement (11.7 percentage points). The improvement in the true-positives of the *Office* (7.2 percentage points) and the true-negatives of the *Kitchen* (7.1 percentage points) are also noteworthy. Based on these improvements we validate the *Kitchen* and *Coffee Bar* locations as the more challenging in the UNIGE-HANDS dataset.

4.5 Conclusions and future research

This chapter presents the UNIGE-HANDS dataset for *hand-detection* and extends a state-of-the-art method proposed in [19] incorporating a dynamic perspective. The dataset is recorded in 5 different locations and guarantees realistic conditions like, changes in the illumination, occlusions and fast camera movements. Additionally, the dataset is

divided in training and testing videos to guarantee fair comparisons of coming methods.

To validate the consistence of the dataset with previous studies we evaluate the state-of-the-art method using cross validation, as suggested in [19, 21], and using the testing videos of the dataset. Three conclusions arises from the results: i) The dataset is challenging enough, and the testing videos are a good approach to avoid the bias in the cross validation results, ii) Little variations between frames highly affects the performance of the existing frame-by-frame *hand-detectors*, iii) The performances reported validates the results of previous studies on which SVM-HOG is the best combination for *hand-detection*.

The HOG-SVM frame by frame approach is extended using a Dynamic Bayesian Network where the dynamic part is carried by a Kalman filter with a constant acceleration model. The parameters of the KF, as well as the decision threshold, are tuned using a genetic algorithms and the Nelder-Mead simplex algorithm. The DBN is evaluated in each of the dataset locations and its performance is presented as the baseline to be used with the UNIGE-HANDS dataset. We highlight the model selection as an interesting research line that could lead to further improvements in the performance of the classifier.

Chapter 5

Hand Segmentation

After detection has taken place, answering the yes-or-no question using global features and classifiers, the hand-segmentation level locates and outlines the hands' region in a positive frame using low level features like colour under an exhaustive local classification framework [134, 200, 160].

This chapter includes some stand-alone pieces of work, each based on a published paper. These are cited at the beginning of each dedicated (sub)section. More in detail, section 5.1 analyses the extent to which color can be discriminative to segment hand at a pixel level. Section 5.2 introduces Superpixels as a powerful strategy to divide an image into meaningful contiguous regions. A novel fast superpixel method is presented in 5.2.1; although it is not applied to egocentric vision, it represents an effective general way of accomplishing the segmentation step. Eventually, an attempt to optimize this class of algorithms is discussed in subsection 5.2.2; results are presented for hand segmentation in first person videos.

5.1 Pixel-wise colour-based segmentation

Skin colour is definitely the most distinctive and significant feature to be exploited to segment hands, being also one of the most common used in literature. After concentrating on RGB colour for a while [106], researchers realized that other colour spaces such

as L^*a^*b , HSV [174] and YCbCr [242] proved to be more suitable for colour-based segmentation, not only for hands [163]. Various models have been proposed for capturing the information carried by colour, the most common being GMM [236].

However, we realized that colour alone does not bring enough information to reliably segment hands, or better, to reliably segment hands *only*. For this reason we exploit optic flow information in order to filter out false segmented blobs, by, roughly speaking, subtracting the global motion of the camera where possible. The way skin-like coloured targets from the background are removed will be explained in what follows¹.

Colour

Although a GMM better captures complex variations of skin colour due to suntan, gender, age etc. [236] we have argued that for a single user a single Gaussian is enough to satisfactorily grasp the relevant colour information. The space which better shows clustering of skin pixels turns out to be, from our experience, the CbCr subplane of YCbCr.

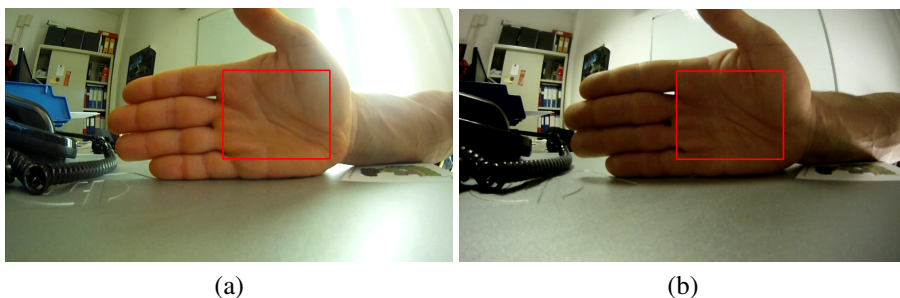


Figure 5-1: Experiment: hand colour characterization.

The experiment that was carried out is relatively simple and it is depicted in Figures 5-1, 5-2 and 5-3. The device used is a GoPro Hero, outputting a 848x480 video at 50 fps (bitrate is approximately 8000-9000 kbps). Many video sequences were shot, framing a slightly moving hand and gradually changing luminosity in the environment as shown in Figure 5-1. It can be noticed how illumination conditions were stresses to a good extent. Statistics were calculated over the manually drawn red box (the box is drawn in the first frame; it is then checked that it only encompass skin pixels through the video

¹The results presented in this section have been published in [160].

sequence). The typical resulting colour histogram for a generic frame is shown in Figure 5-2. As one can see very peaked distribution appear for the Cb and Cr channels, while a changing luminosity results in a larger Y histogram. Mean and standard deviation were calculated for each channel for each frame. Means are plotted in Figure 5-3 for one of the five 1500-frame sequences. Standard deviations are always around 3 (2,95 on average) for the Cb channel and around 4 (4.3 on average) for the Cr channel. It can be seen the extent to which illumination was altered.

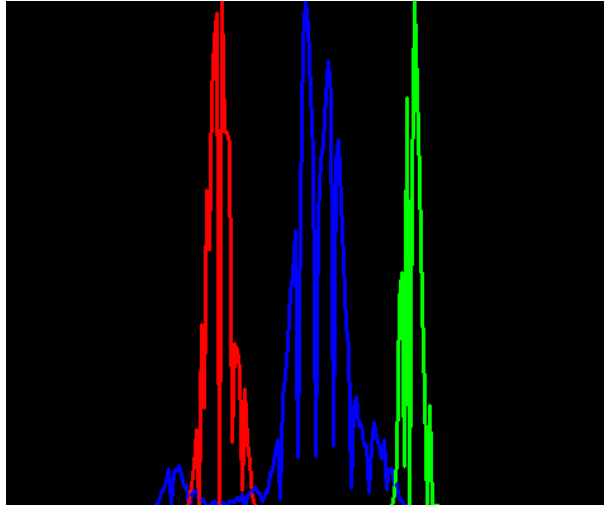


Figure 5-2: Histogram of hand pixels' colour (relative to a single frame, ROI is shown in Figure 5-1). Blue line is Y channel, green is Cr and red is Cb.

Figure 5-4(a) shows how the vectors (μ_{Cb}, μ_{Cr}) cluster in the (Cb, Cr) plane. The covariance matrix of the 2d distribution clearly have eigenvectors which are not parallel to the axis, thus we opted for a bi-dimensional Gaussian to describe the colour model, instead of two separate Gaussians, one for each channel.

Figure 5-4(b) shows instead how the scaled feature $(Cb/Y, Cr/Y)$ cluster along what seems to be a straight line (three different hands, three different coefficient). This model does not introduce much improvement, thus we set aside this observation for future works which may include classifying hands sides based on colour.

The model was test on several sequences shot while performing different activities, like drawing, writing on a whiteboard, typing. Figure 5-5 (a) shows a sample frame. Figure 5-5 (b) shows colour-based segmentation using a single Gaussian. Given a model (μ, Σ) ,

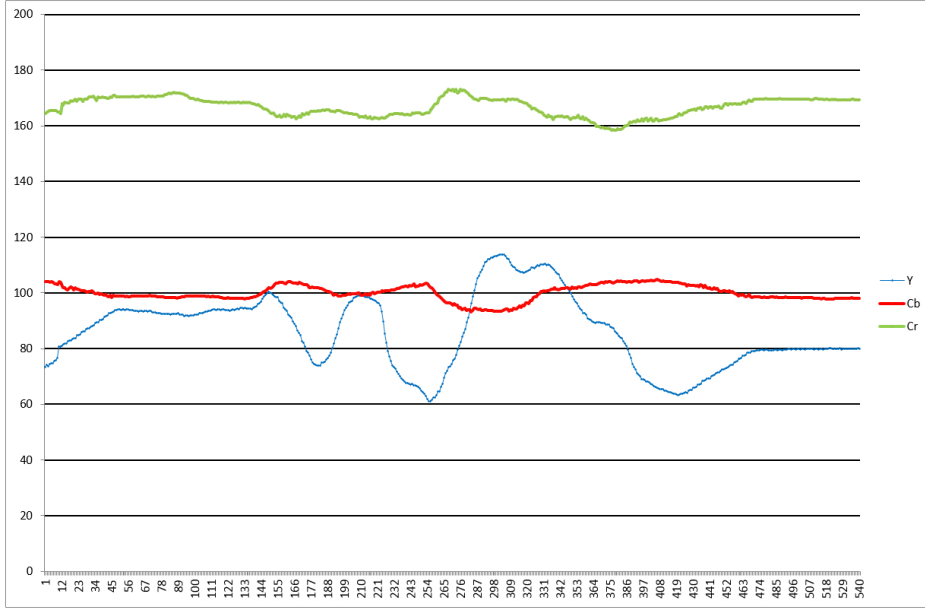


Figure 5-3: Average hand pixels' colour frame by frame, changing illumination in the scene

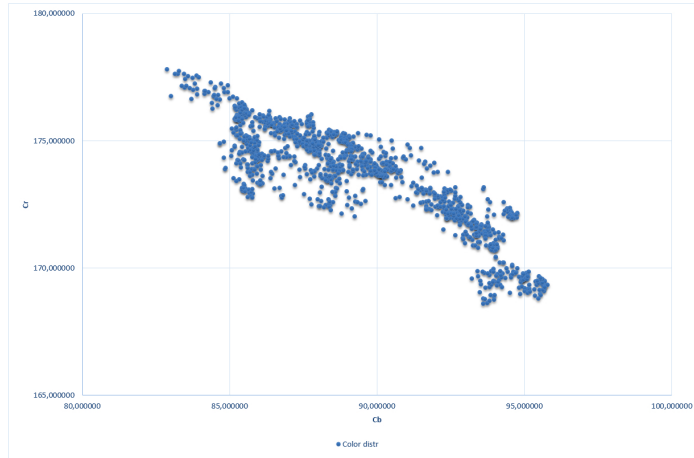
segmentation is obtain by setting the condition

$$(x - \mu)^T \Sigma^{-1} (x - \mu) \leq 2, \quad x \in (Cb \otimes Cr) \subset \mathbb{R}^2. \quad (5.1)$$

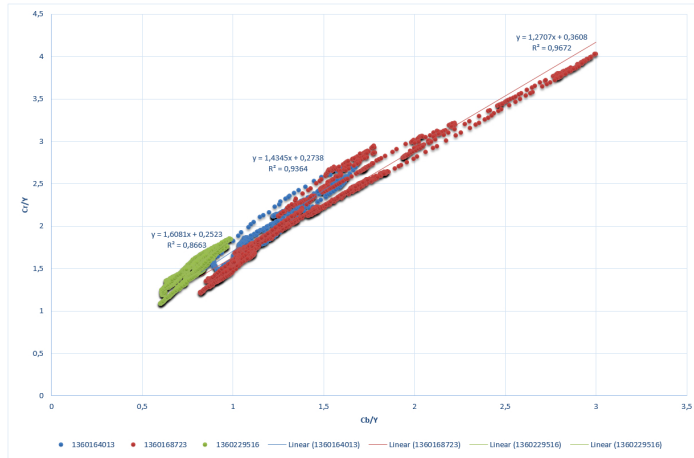
It can be clearly seen that objects with skin-like colour (as the mouse pad for instance) are segmented as well as the two hands in the proposed sequence. The way uninteresting targets from the surrounding environment are filtered out is explain below.

Optic flow refinement

We propose that hands doing things hardly move jointly with the head, rather they show different displacements from one frame to another. For this reason we employ optic flow to estimate the average movement of the camera based on the flow vectors calculated through the method proposed in [219] and exploiting the features proposed in [204] and refined in [104]. This way, things moving disjointly from the head can be identified



(a)



(b)

Figure 5-4: Colour based segmentation. (a) Clustering of Cb and Cr features (b) Clustering of Cb/Y and Cr/Y features.

for they show different optic flow vectors associated to their interest points. Results are shown in figure 5-6. The head is almost still, thus the majority of the flow vectors has very little module and a direction which is opposite to the one towards which the head is (slightly) moving. On the other hand the moving hand show marked vectors, which

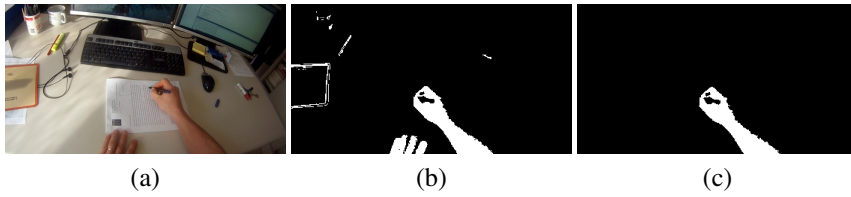


Figure 5-5: Colour based segmentation. (a) Sample frame (b) Colour-based segmentation (c) Optic flow-improved segmentation.

vectorially sum up their and head movements.

Blobs which show no interest points, or which flow is similar to the global one are eliminated as shown in Figure 5-5(c). Unfortunately this leads in most frames to the removal of the blob generated by the left hand, which lies still on the table. This suggest that the proposed method works for hands which are acting relevantly only.

The two algorithms do not required massive computational resources, however the 50 fps of the GoPro camera are not supported. On average, frame processing time is around 50 ms.

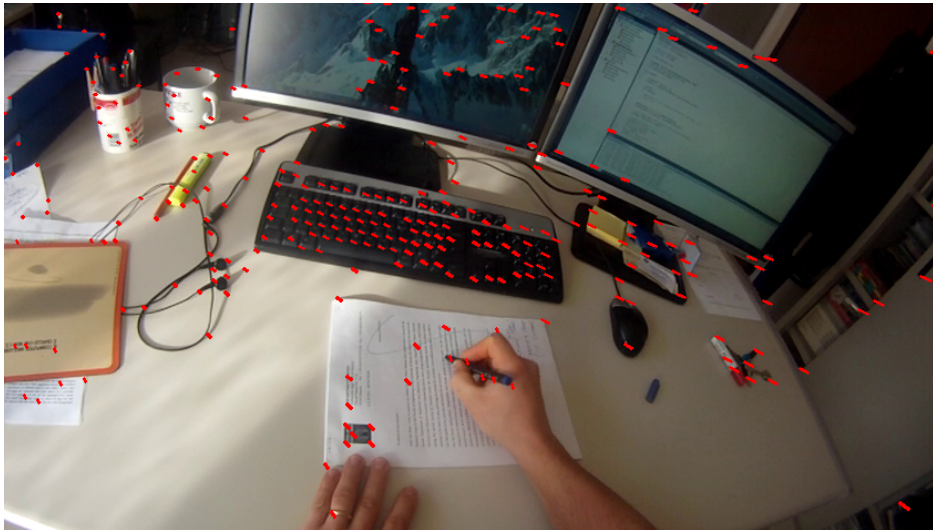


Figure 5-6: Optic flow

Discussion

An approach for hand segmentation in first person videos was proposed and tested on sequences recorded with a GoPro wearable camera. Such an approach strongly relies on the most natural feature usable for skin detection, namely colour. Fusion with camera movement information, extracted from optic flow vectors allow for filtering undesired detected blobs which are integral with the environment. Unfortunately this also leads to the removal of inactive hand targets, which could be seen either as a negative or as a positive feature.

5.2 Superpixel-based segmentation

There has passed nearly ten years since the concept of over-segmentation evolved to the one of Superpixels (Fig. 5-7). The idea has matured and by now has been widely explored within the computer vision community. Superpixel algorithms are meant to group *similar* pixels into meaningful regions, or clusters, in order to create a higher-level structure in an image. They capture similarities mainly by jointly considering colour and spatial proximity and thus try to provide a semantic clustering of an image.



Figure 5-7: Trend for the interest in Superpixels along the last 10 years (Web searches).

Although the majority of image processing algorithms operate at pixel level, processing higher-level representations (see Figure 5-8) can turn out to be more efficient. For example, one can reduce the hundreds of thousands of pixels to hundreds or thousands of superpixels while still maintaining very accurate boundaries of objects or other key

features in an image, such as colour statistics. In fact, it is often even more convenient to get rid of noisy (and often redundant) pixel-level information.

For these reasons, Superpixels methods have been exploited for many purposes, ranging from segmentation to feature computation and are becoming a really popular preprocessing tool in many computer vision applications. Just to mention some: foreground-background segmentation [168], object localization [82], tracking (and extended tracking) [159] [233] [249]. By providing a mid level representation of an image, Superpixel methods are also used in visual saliency detection [237], in order to extract relevant information from images. Eventually, Superpixels can be extended to Supervoxels, which are widely employed in biomedical applications [140].

The majority of superpixel methods are segmentation-oriented and in fact the line between over-segmentation and superpixel-segmentation is not neat at all. [44] proposes that superpixel segmentation is a particular oversegmentation which preserves a sufficient amount of the salient features of its underlying pixel-level representation. This is why superpixels can be exploited for different purposes other than segmentation; however, again, such a sufficient amount is extremely arbitrary, yet depending on the application. We suggest that superpixel algorithms are simply a class of methods for extracting arbitrarily higher level features from images.

There are several approaches to generating Superpixels. Each method can be considered to perform better only depending on the kind of problem it is applied to. For instance, graph-based methods such as [205] seem to provide better adherence to boundaries. On the other hand, one may want to construct a graph out of a Superpixel grid in which case enforcing Superpixels connections is an issue [162]. Also, some methods give more regular cluster's contours [130], while some older approaches construct irregular shapes with inhomogeneous sizes [226]. State-of-the-art algorithms can be categorized in 2 main groups, as either *graph-based* or *gradient ascent* methods. *Graph-based* approaches consider each pixel as a node in a network (or graph), connected to his neighbours through edges. Weights of edges are related to pixel's similarity in a given colour space and Superpixels are generated by gradually cutting such overconnected graphs, by minimizing some cost function defined over the whole image [203], or by finding minimum spanning trees [75]. Gradient-descent-based algorithms start instead from a given rough clustering of the image. Clusters are then refined iteratively until some global convergence criterion is met [224], or after a fixed number of iterations [2].

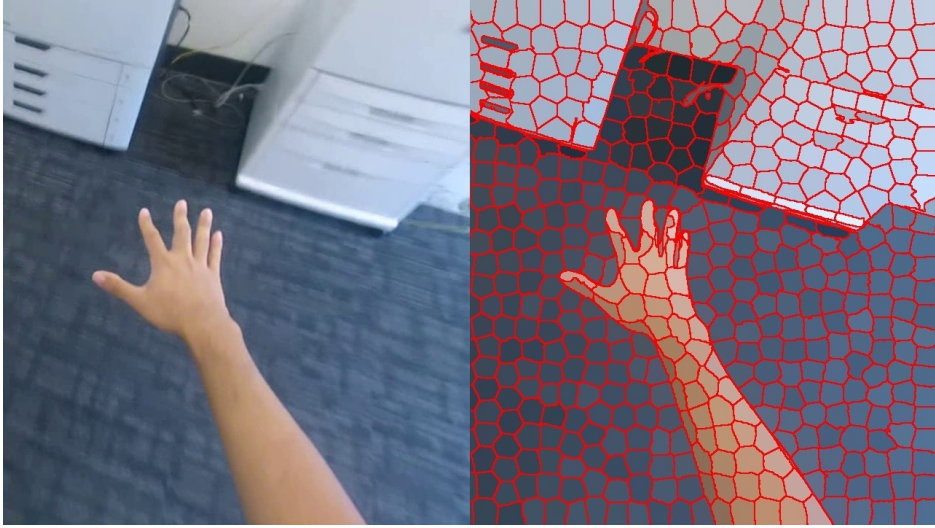


Figure 5-8: Superpixel methods provide a higher level representation of an image. Such a representation is particularly suitable for segmentation purposes and in fact often used to this end.

5.2.1 A Generative Superpixel Method

Superpixel methods have become popular in recent years as they provide an efficient preprocessing tool for a manifold of computer vision applications. In this section, we present a method based on a self-adapting and self-growing network, which is bred starting from two random initialization seeds in the image. Such a network, which is a modification of the Instantaneous Topological Map (ITM), is inspired to a Growing Neural Gas (GNG) and like many other self adapting tools employs a Hebbian learning framework. Key point in competitive learning is the definition of a suitable distance function, which we analyse in depth here. Distance is indeed the notion which allows to link unsupervised competitive learning with segmentation, where cluster formation reduces to node creation and adaptation within the exploration of a suitable multidimensional input space.²

As already mentioned, state-of-the-art algorithms can be categorized in 2 main groups as either *graph-based* or *gradient ascent* methods. The method here presented is indeed based on the construction of a graph, and would thus fall in the first category. However,

²The results presented in this subsection have been published in [162].

there is a substantial difference in how the graph is constructed. As already pointed out, graph-based approaches are usually initialized with an overcomplete and overconnected graph, which is progressively “disassembled”, or cut, according to a similarity criterion. These approaches are usually quite computationally expensive as the graphs must be walked through several times. Moreover, they often start from a regular lattice, well aware of the fact that the final desired output is something not regular at all. This issue can also be spotted in gradient-ascent-based methods, where moving far from the initial regular clustering of the image may cost several iteration steps.

We therefore propose a *Generative Superpixel (GSP)* approach, which progressively, and in one iteration only, “grows” a graph, starting from 2 initialization seeds, according to inputs coming from the image. The network is generated by the inputs themselves, following the Neural Gas approach [81], exploring the input space with no constraints. More in detail, we modified the Instantaneous Topological Mapping Model for Correlated Stimuli (ITM) proposed in [105]. This network turns out to be more agile than the standard GNG as it does not require the maintenance of expensive averages accumulated over time and the tuning of life parameters.

Inputs for the map are randomly extracted from the image and are constructed as a four dimensional vector obtained by fusing spatial location and chrominance values of the pixel. A weighted distance is presented which measures similarities in such a space, in order to implement competitive unsupervised learning of the space structure. The network is fed by randomly selecting input pixels, in order to avoid distortions given by a raster scanning of the image. This also allows to obtain good results by employing only a fraction of the total number of pixels.

We present a practical investigation on the proposed algorithm, together with a comparison of the proposed method with the *Simple Linear Iterative Clustering (SLIC)* method [2], which, to best of our knowledge, represents the current cutting-edge superpixel algorithm, although based on a previous work [253]. A particular stress is given to the parameter tuning part, which allows for an almost fair qualitative and quantitative comparison. Eventually, future research directions are also proposed.

Algorithm

We propose a new method for extracting superpixels, which is based on a wider concept of space which we denote as *image space*. Namely, this space is a $2 + 3$ vector space which spans over the (x, y) position of the pixels and 3 colour channels. To be more

precise we explore a subspace of the *image space* for the practical implementation of the algorithm, which in turn comes from a particular choice of the colour channels, i.e. the (Cb, Cr) plane in the YCbCr colour space. However, many other colour (and spatial) representations can be explored which may give similar results.

This choice is motivated by the fact that segmentation is often compromised by changes in illumination over curved surfaces, the best example being objects' borders, which often show a dark pattern which is difficult to separate. YCbCr representation of colour has been often exploited for segmentation thanks to its ability of separating luminance from chrominance information [216] [160].

Four dimensional vectors are thus constructed as (x, y, Cb, Cr) for each pixel. Such vectors constitute the inputs to stimulate the proposed growing neural network, which is presented in the following.

A neural network needs of course to be supplied with a distance in the input space. One of the most common choices is of course the Euclidean distance, while sometimes the Mahalanobis metric is employed in statistical analysis. The most suitable distance is here a weighted distance, whose general expression for an N -dimensional vector space is given by

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^N a_i (x_i - y_i)^2}. \quad (5.2)$$

Here a_i is a vector of weights which has the dual purpose of normalizing colour and spatial dimension and to control spatial compactness of the superpixels. It will be shown how one parameter only is needed in practice to specify the N -dimensional vector \mathbf{a}

Within this framework, generating superpixels reduces to finding a graph which fits the considered input image subspace. The construction of the graph is addressed in details in the following. Superpixels are then nothing but the 2D projection of a 4D cluster of the resulting trained network. Superpixels can be visualized in space as all the pixels which has the minimum distance from a given graph node (with respect to all other nodes).

As it will become clearer in the following, the number of resulting superpixels cannot be controlled directly, due to the self-growing (*generative*) and self-organizing nature of the process. This can be seen either as a drawback of the method or as an additional freedom of the network of fitting the input space. Anyway, superpixels' dimension can be limited by means the parameter r_{max} of the ITM. This actually produces an implicit limitation on the number of clusters, being the input space bounded in all its four dimensions.

For what concerns the training of the ITM, we soon realized that segmentation results are affected by how raster scanning of the image is performed. We tried to move around this by exploiting space filling curves [185] to determine the ordering of the inputs. However, the construction of such curves is quite expensive in terms of computational resources. We eventually opted for a random exploration of the space, based on random number generation (sampling from a uniform distribution over the image rectangle) as explained in details in the next paragraphs. This is the reason why the ITM algorithm had to be modified accordingly.

The ITM for sparse stimuli Back in 1999 an Instantaneous Topological Mapping Model for Correlated Stimuli was presented in [105]. The idea was to overcome difficulties arising when considering sequences of highly correlated stimuli, such as trajectories. The resulting method turns out to be computationally lighter and faster in adaptation with respect to the standard Growing Neural Gas algorithm [81], basically as it does not require the maintenance of expensive averages accumulated over time and life parameters. In fact, two parameters only are needed, namely a shift parameter ε and a resolution r_{max} . According to the authors ε could even be safely set to zero in the original algorithm. This cannot be done here, as nodes must be given the possibility of adapting to surrounding stimuli. Otherwise, one could come to the absurd situation that a new node is triggered on an edge, without the possibility of shifting away. This would end up in a superpixel centred on an edge.

We propose a slight modification of the ITM algorithm (Algorithm 5.1) for dealing with sparse stimuli, as the original algorithm, as it is, prevents creation of new nodes inside big spheres in the input space. This is because the input space was originally supposed to be explored along continuous trajectories. Moreover, isolated nodes are here allowed, as disconnected components in the graph are encouraged, as they should represent really different areas and may have a semantic meaning, such as background-foreground or may represent separate objects. Sparse inputs may cause huge Thales spheres to form and prevent node formation in the original Node adaptation step: in fact, new node insertion is regulated by the scale factor $r >_{max}$ only here. The network is initialized with 2 random seeds in the input space, i.e. 2 nodes with random weights, connected by an edge.

In terms of computational complexity, the Matching step scales with the number of neurons, which is increasing at training stage, but is bounded and can be implicitly controlled by the parameter r_{max} . Edge adaptation scales with the average number of neighbours, which is related to the dimensionality of the input data. This constitutes

Algorithm 5.1: Modified ITM for sparse stimuli.

Data: input vector \mathbf{x} ; given distance $d(\cdot, \cdot)$; set of N nodes with weights \mathbf{w}_i

Parameters: shift ε ; resolution r_{max} ;

Result: Network adapted to the new stimulus \mathbf{x}

1. Matching: find nearest neighbour n and second nearest s ;

Initialize $d_n = MAX_VAL$ and $d_s = MAX_VAL - 1$

for $i = 1 : N$ **do**

$d = d(\mathbf{x}, \mathbf{w}_i)$;

if $d < d_n$ **then**

$d_s = d_n$;

$d_n = d$;

$s = n$;

$n = i$;

else if $d < d_s$ **then**

$d_s = d$;

$s = i$;

2. Weight adaptation:

$\mathbf{w}_n = \mathbf{w}_n + \varepsilon(\mathbf{x} - \mathbf{w}_n)$;

3. Edge adaptation:

if $n \leftrightarrow s$ **then**

$n \leftrightarrow s$;

$N(n)$: set of connected neighbours of n

for $\forall j \in N(n)$ **do**

$S(\mathbf{w}_n, \mathbf{w}_j)$: Thales sphere through \mathbf{w}_n and \mathbf{w}_j ;

if $w_s \in S(w_n, w_i)$ **then**

$n \leftrightarrow j$;

4. Node adaptation:

if $d(\mathbf{x}, \mathbf{w}_i) > r_{max}$ **then**

 add new node m with $w_m = x$;

$n \leftrightarrow m$;

if $d = d(\mathbf{w}_n, \mathbf{w}_s) < \frac{1}{2}r_{max}$ **then**

 remove node s ;

an additional reason why we restricted the input image space to a four dimensional subspace. All other steps are independent of the number of neurons involved allowing the algorithm to execute fast even for large networks.

As already pointed out, two parameters only are needed by the ITM algorithm, namely r_{max} and ε . The threshold r_{max} can be interpreted as a mapping resolution. The method is substantially different from providing a learning rate as in the GNG, as nodes are

created at a maximum speed of one per stimulus if inputs are too far apart. As nodes are allowed to adapt by moving by a small amount, a criterion is provided to remove nodes that are too close to each other. The threshold used is derived from r_{max} .

ε can instead be seen as a smoothing parameter, which regulates weight adaptation. It has a small value and, in principle, could have a different value for each of the four coordinates, allowing different shifts in colour and space. This issue will be discussed more in detail when addressing the problem of parameter tuning.

Space and Distance The ITM algorithm (Algorithm 5.1) is very general and its implementation does not depend on the distance used. However, superpixels correspond to clusters in the *image-space*, which is a four dimensional space. This presents a problem in defining a distance measure, which is not trivial. For our purposes, the standard Euclidean distance is clearly not suitable: simply defining $d(\cdot, \cdot)$ to be the 4D Euclidean distance in the (x, y, Cb, Cr) causes non-consistent clustering behaviours for different image sizes. A pixel's colour is represented in the (Cb, Cr) colour subspace, whose range of possible values is known. On the other hand, the pixel's position may take a range of values that varies according to the size of the image. For large images, spatial distances outweigh colour proximity, giving more relative importance to spatial proximity. This is why a distance like the one proposed in equation 5.2 is employed here.

Actually, equation 5.2 can be extremely simplified in our case, by reducing the number of parameters needed to one only:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{d_s^2 + a d_c^2}. \quad (5.3)$$

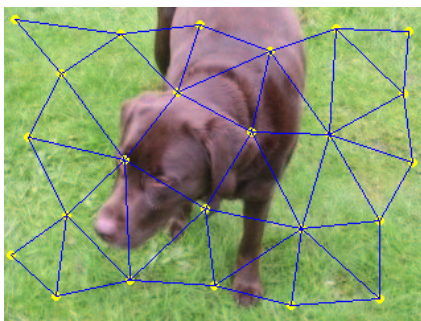
where

$$\begin{aligned} d_s &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \\ d_c &= \sqrt{(Cb_1 - Cb_2)^2 + (Cr_1 - Cr_2)^2}. \end{aligned} \quad (5.4)$$

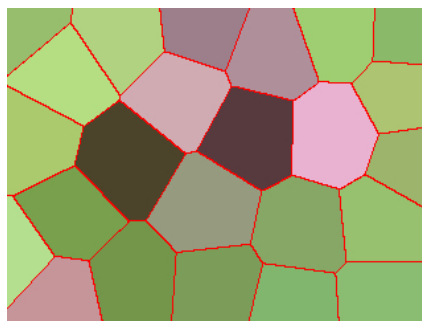
Practically, there is no need of having different weights for the two colour channels, and the same holds for the two spatial coordinates: the weight would be thus immediately reduced to two only. The spatial weight can be further eliminated by means of an overall multiplicative constant (which of course modifies the colour weight) that can be safely set to 1, yielding equation 5.3.

As an intuitive explanation of how the weight a works, Figure 5-9 shows results of

ITM clustering for $a = 0$ (i.e. $a_{Cb} = a_{Cr} = 0$): only space is considered and cells reflect the homogeneous nature of a 2D even distribution of pixels. Figure 5-10 shows results obtained by naively employing Euclidean distance ($a_x = a_y = a_{Cb} = a_{Cr} = 1$): spatial proximity of pixels still outweighs colour similarity, even though the silhouette of the dog starts to take its shape. The ITM network is more irregular as it is the projection onto two dimension of a map “living” in four. On the opposite hand, Figure 5-11 show the weights configuration $a_x = a_y = 0$: pixels are clustered only based on colour proximity: as a matter of fact only 3 clusters are formed, whose visual effect is very close to that of a thresholding algorithm, which is not what we desired. The network actually “lives” in the (Cb, Cr) space.

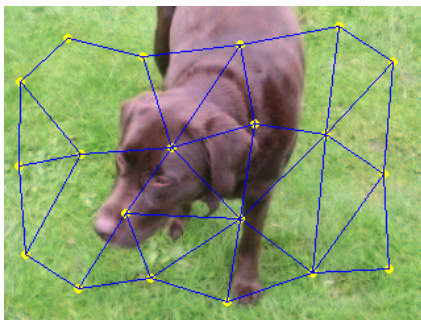


(a) 2D ITM network

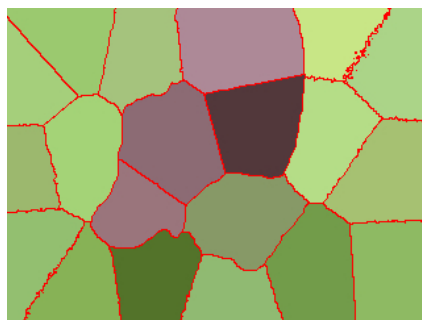


(b) ITM 2D Voronoi cells

Figure 5-9: ITM segmentation with $r_{max} = 60$, $\varepsilon = 0.005$ and spatial distance only.



(a) 2D projection of the 4D ITM network



(b) 2D projection of 4D clusters

Figure 5-10: ITM segmentation with $r_{max} = 60$, $\varepsilon = 0.005$ and Euclidean distance.

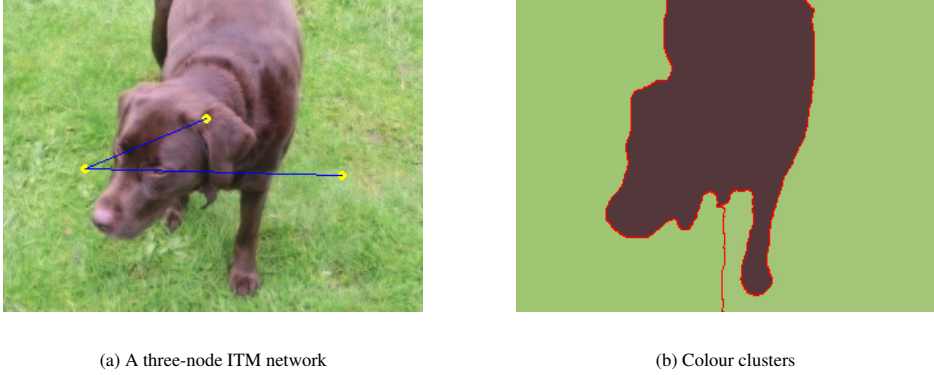


Figure 5-11: ITM segmentation with $r_{max} = 60$, $\varepsilon = 0.005$ and 2D colour distance.

To conclude this section, we point out that introducing the weight a in equation 5.3 is equivalent to employ the scaled colour features

$$Cb' = \sqrt{a} Cb, \quad Cr' = \sqrt{a} Cr, \quad (5.5)$$

or, equivalently, scaled spatial features

$$x' = x/\sqrt{a}, \quad y' = y/\sqrt{a}. \quad (5.6)$$

Tuning the weight a is then equivalent to choosing a suitable feature vector to be given as input to the ITM.

Input randomization and thinning As already mentioned, ordering of the input vector sequence for the ITM represents an issue, as results are strongly affected by correlations brought in by raster scanning of the image. Covering the image with a space filling curve has proven to be effective in many applications facing this issue, however, their construction is quite expensive in terms of computational load.

Opting for a random exploration of the space turns out to be a suitable choice. Random number generation is fast and can guarantee inputs which are evenly distributed over the image. We sample from a uniform distribution over the image rectangle to select pixels, which have thus the same probability of being extracted. Keeping track of all pixels extracted, to be sure that a pixel is not extracted twice, would imply a huge set of conditional statements which would slow down significantly the algorithm. However,

the probability of multiple extractions is quantifiable, being expressed by a binomial distribution mass function.

$$f(k; n, p) = \binom{n}{k} p^k (1 - p)^{n-k}. \quad (5.7)$$

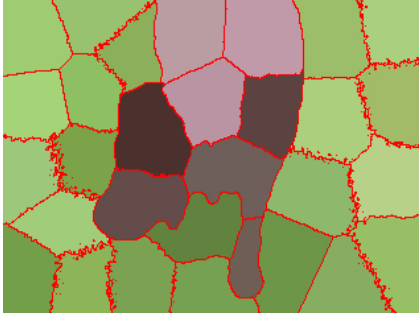
The probability of extracting the pixel twice ($k = 2$) is even lower if we extract only a small fraction f of the total amount of available inputs. Here $p = 1/N$, where N is the total number of pixels in the image (as we extract from a uniform distribution) and $n = f \cdot N$ with $f < 1$. As a matter of fact, the result of picking the same value more than once can be simply seen as a very small amount of noise added while training the network.

Thanks to this observation, we realized that using only one tenth of the available pixels still provides a well trained network, while consistently speeding up the algorithm. The trade-off between the noise injected and the boost in speed is positive. By the way, using a small percentage of pixels for Neural Network training has also proven to be effective in other contexts, such as shape fitting for object detection [216].

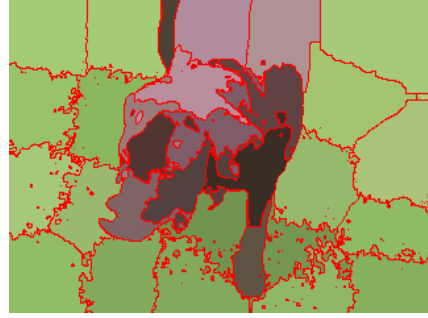
Postprocessing As other superpixel algorithms (e.g. [75] and [2]), our method does not enforce spatial connectivity explicitly. After the network is trained and *all pixels* in the image (also the ones that were not used for training the network) are assigned to its nearest ITM node, some spatially isolated pixels may appear (figure 5-12), which do not belong to their surrounding connected component. This issue arises as the representation of the image in the *image space* is not a surjective map nor has a simply connected range in the (x, y, Cb, Cr) codomain. The 4D clusters are thus not guaranteed to be connected and so are their 2D projections, although in principle the Voronoi cells generated by the ITM are connected (in a simply connected space). This happens because colour is not a continuous function of space over the image, namely the image representation is not guaranteed to be dense in the *image space*. Intuitively, there is spatial contiguity between pixels but colour contiguity is not guaranteed at all.

Experimental results

Some partial results have been already presented in the previous section. In the following we provide a discussion on parameter tuning and an analysis of the performance of the proposed algorithm.



(a) GSP: $r_{max} = 50$, $\varepsilon = 0.1$, $a = 2$



(b) SLIC: $N_c = 30$, superpixels = 40

Figure 5-12: Segmentation without postprocessing step: isolated pixels are a common issue, especially along superpixel's borders.

Parameter tuning Parameter tuning is a matter of fact for any algorithm. This allows parameters to be set optimally, as no algorithm just work as it is. Also, for a fair comparison of methods, it is often necessary to specify which choice of parameter has been done.

The 3 parameters appearing in the proposed algorithm are summarized in table 5.1. We show in this section that some constraints can be worked out, which reduce the number of parameters to 2 only. Moreover, a specific combination of the two remaining parameters can be given the interpretation of the number of superpixels N_{sp} , thus giving (implicit) control over such a quantity, which is often considered influential in a superpixel algorithm and allows for a comparison with [2], which has N_{sp} as an explicit parameter.

The size S of the 4D space where input are extracted for the ITM training is given by the product of the maximum range of the four coordinates, namely $S = w \cdot h \cdot 256^2$, where w and h are the image's width and height respectively. However, the introduction of the parameter a can be seen as a way of "stretching" colour features by a factor \sqrt{a} .

Thus, if we fix a , S becomes $S \rightarrow a \cdot S = a \cdot w \cdot h \cdot 256^2$. By approximating a superpixel as a 4D hypercube of volume $V_{sp} = (2r_{max})^4$, the volume S should be able to host approximately

$$N_{sp} = \frac{S}{V_{sp}} = \frac{a \cdot w \cdot h \cdot 256^2}{16 \cdot r_{max}^4} \quad (5.8)$$

superpixels. However, typical image's colour histograms do not span over the whole

Table 5.1: Parameters appearing in the Generative Superpixel method

Parameter	Meaning
a	- Weight in the distance - Colour features' scaling
ε	- Adaptation capability of the nodes
r_{max}	- Resolution of the network

8 bit range of 256 values: the value N_{sp} thus represents an upper bound on the actual number of superpixels. We will refer to it as to N_{max} instead of N_{sp} . For example, the combination of parameters considered in figure 5-12 should give approximately a limit of ~ 100 superpixels (the original image is 320×240). Indeed, only 29 superpixels (one third) can be spotted in the image, as two dominant colours only are present and a large input space region is not hit by any stimulus.

The meaning of ε becomes clear from the second step of algorithm 5.1. A big ε makes the adaptation of a neuron very unstable, as its weight will be drastically modified by the last input hitting it. On the other hand a too small ε makes the node slowly adapting to inputs, preventing us from reaching our main goal of growing an adapting network over the *image space*. Another way of writing the adaptation equation is

$$\Delta w_n = \varepsilon(x - w_n). \quad (5.9)$$

Roughly, as the *image space* is able to host a network of N_{max} nodes and as we take only a fixed fraction $f = 0.10$ of the available pixels for training, we can imagine that each neuron will, *on average*, be hit at most by a number of inputs equals to

$$n_{hits} = \frac{f \cdot w \cdot h}{N_{max}} = C \cdot \frac{f \cdot r_{max}^4}{a}, \quad C = 2^{-12} \quad (5.10)$$

Noticeably, this number is independent of the dimension of the image, depending only on the parameters of the network.

If we suppose that the stimuli x hitting the node n are uniformly distributed inside the 4-sphere $S_4(w_n, r_{max})$ (centred in w_n and with radius r_{max}), there exists an average shift $|x - w_n|$, which will be the radius of the 4-sphere with a volume which is half of

that of $S_4(w_n, r_{max})$:

$$< |x - w_n| > \approx \sqrt[4]{\frac{1}{2}} r_{max}. \quad (5.11)$$

We would like the weight of neuron n to resemble the average of all the inputs hitting it at the end of the adaptation step. This can be obtained by “weighting” the average contribution (eq. 5.9) with a factor $1/n_{hits}$. That is, we can suppose

$$\epsilon \approx \frac{a}{C \cdot f \cdot r_{max}^4} \quad (5.12)$$

This way, we get a constraint which allows us to get rid of one of the three parameters. As a practical example, the configuration of parameters in figure 5-13 is consistent with eq. 5.12.

Performances Main strength of our method lies in its speed. This is obtained thanks to the fact that it does not requires multiple iterations as k -means based algorithms such as [2]. In addition, superpixel’s centres are built by training a network by sampling only a fraction of the total available training data.

Execution times for the examples given in this work are given in table 5.2. We employ an Intel Xeon 2.66 GHz processor with 4 GB RAM for our tests and the c++ code provided by the authors of SLIC.

Results are comparable as the number of generated superpixels is the same. The weight parameter a of GSP is related to the two parameters of SLIC by the following equivalence

$$a \rightarrow \frac{w \cdot h}{N_c^2 \cdot N_{sp}}, \quad (5.13)$$

where N_{sp} is the desired number of superpixels and N_c is a free parameter. Creating such correspondence allows for a fair comparison.

As it can be noticed from Figure 5-13, result of the two algorithms are comparable for a medium number of superpixels. However, SLIC tends to fail for a small N_{sp} (Figure 5-14), while GSP gives surprisingly good results. On the other hand, SLIC looks more accurate when the number of superpixels increases to 100, as depicted in Figure 5-15. Here SLIC’s superpixels are less regular, adapting better to the shape. Moreover, many borders in the green region are inexplicably irregular in GSP.

Table 5.2: Execution time (milliseconds)

Test	GSP	SLIC
Figure 5-12. (320 x 240)	454	2052
Figure 5-13. (320 x 240)	545	2174
Figure 5-14. (320 x 240)	241	1227
Figure 5-15. (320 x 240)	900	2612

Conclusion and future research lines

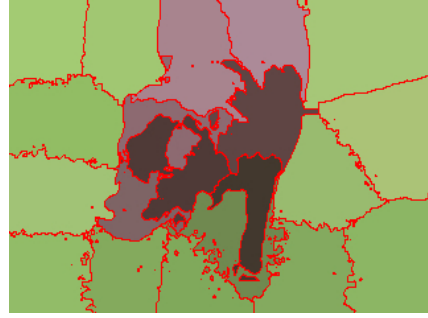
In this work, we proposed a Generative Superpixel method. The algorithm lay on the training of a self-growing and self-adapting neural network: such a network is a modification of the existing ITM map, which has been extended to cope with sparse stimuli. The modified ITM explores the input *image space* and creates clusters in a multidimensional space, based on a suitable definition of distance. Pixels are then assigned to clusters based on proximity to neurons, relying on the very same distance used to train the network. The projection onto the spatial coordinates of the obtained clusters is the desired superpixel segmentation.

The algorithm is compared to the *Simple Linear Iterative Clustering* (SLIC) method, showing comparable results, while significantly reducing segmentation time. GSP does not provide direct control over the number of superpixels, however an upper bound is given by a combination of its two parameters. Control over superpixel compactness is provided by the parameter r_{max} . Extension to supervoxels or to other colour representations of the image is straightforward: formulas can be easily generalized for three colour channels and for supervoxels by simply adding extra dimensions. In addition, as shown in figure 5-11, GSP can be used directly as an object segmentation algorithm by neglecting spatial coordinates.

GSP offers a wide range of possibilities for future investigations, which were not included in this work, starting from a quantitative analysis of its *adherence to boundaries* and *segmentation accuracy*, given a ground-truth. The algorithm may also need some



(a) GSP: $r_{max} = 50$, $a = 2$ ($\varepsilon = 0.013$)



(b) SLIC: $N_{sp} = 30$, $N_c = 35$, 10 iterations.

Figure 5-13: Parameter comparison between GSP and SLIC superpixel methods. We fixed $r_{max} = 50$ and $a = 2$ for the Generative Superpixel method. This result in $\varepsilon = 0.013$. The algorithm then generates 30 superpixels, which are set as a parameter in SLIC. Setting $N_c = 35$ in slic is then equivalent to setting $a = 2$ in our method.

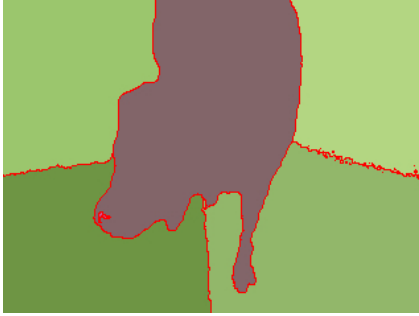
improvement for enhancing segmentation performances when considering a high number of superpixels.

Different and more exotic distance functions can be found, which give better segmentation results. In particular, the idea of considering an image as a Riemann manifold, where curvature is pointwise induced by colour, is interesting and deserves future deepening

Eventually, an analysis of the stability of the algorithm along frames in a video sequence could open the possibility of using the nodes of the neural network as point to be tracked in an extended-tracking framework.

5.2.2 Video optimization of Superpixel algorithms

In this section, we propose a strategy for optimizing a superpixel algorithm for video signals, in order to get closer to real time performances which are on the one hand needed for egocentric vision applications and on the other must be bearable by wearable technologies. Instead of applying the algorithm frame by frame, we propose a technique inspired to Bayesian filtering and to video coding which allows to re-initialize superpixels using the information from the previous frame. This results in faster convergence and demonstrates how performances improve with respect to the standard application of the



(a) GSP: $r_{max} = 150$, $a = 10$ ($\epsilon = 0.0008$)



(b) SLIC: $N_{sp} = 6$, $N_c = 14$, 10 iterations.

Figure 5-14: SLIC breaks for small N_{sp} (setting $a = 10$ in our method, is equivalent to having $N_c = 14$).

*algorithm from scratch at each frame.*³

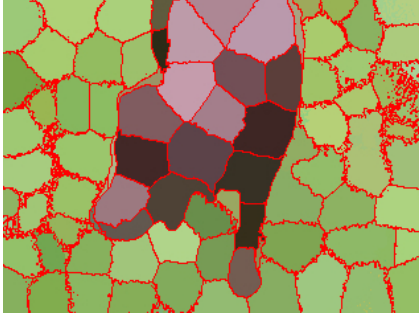
Overview

Despite the prolific literature on the topic, the application of superpixels to video analysis is still at his early stages and have been employed in first-person-vision only in [200] for hand segmentation and tracking purposes. More in details, this work exploits the by now consolidated *Simple Linear Iterative Clustering* (SLIC) algorithm proposed in [2], by simply applying the method from scratch frame by frame. This turns out to be quite slow, still not allowing real-time performances, required by egocentric-vision applications. [187] actually provides a real time implementation of SLIC, but it needs a graphic card to exploit GPU and the NVIDIA CUDA framework. Such implementation is hardly portable to a wearable device.

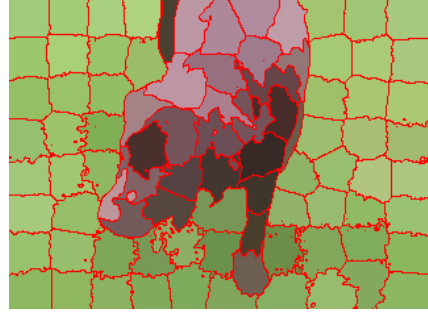
Other attempts to provide fast superpixel algorithms are [206] and [67]. However, while the first focuses again on single images and not on videos, the second sacrifices accuracy for the sake of performance.

To the best of our knowledge, the only effective attempt to bridge the gap between Superpixels and video is addressed by [44], where a characterization of Temporal Superpixels (TSPs) is provided. Here a complex generative model is proposed in an attempt of treating the video signal not as a simple sequence of images, but with a special stress on time.

³The results presented in this subsection have been published in [161].



(a) GSP: $r_{max} = 30$, $a = 2$ ($\epsilon = 0.1$)



(b) SLIC: $N_{sp} = 100$, $N_c = 20$, 10 iterations.

Figure 5-15: For a high number of superpixels, the SLIC method provides a better superpixel representation, although execution time is high above GSP's. In particular, many borders in the green region are extremely irregular.

The method differs from the Supervoxels approach [239], which works well for actual volumetric data (e.g. medical imaging), in that time is not simply treated as an extra dimension. It is probably the first probabilistic model to represent superpixels. However, the method, although very effective, is far from providing real-time performances, requiring tens of seconds for performing Bayesian inference over a single frame.

In this subsection we suggest a novel way of performing a smart re-initialization of superpixels in consecutive frames, in order to optimize frame's elaboration time. The primary goal is here to get closer to real-time video elaboration. The proposed approaches take some inspiration from Bayesian filtering and from video coding. In particular, results are shown for SLIC [2], but they can be easily extended to other methods such as [162] and with slight modifications to graph-based approaches as well.

Proposed method

In the next paragraph, we first briefly review SLIC in order to better explain the proposed optimization methods.

SLIC SLIC considers a 5-dimensional feature vector for each pixel, composed of its (x, y) position, and its three *Lab* colour channel values. The algorithm is initialized with a fixed number of cluster's centres, equally spaced and arranged in a regular grid of step

S (refer to Figure 5-16). k -means clustering is then performed by searching for similar pixel in overlapping $2S \times 2S$ regions (here is the key to speeding up with respect to standard k -means). Once each pixel has been associated to the nearest cluster, centres' positions are adjusted to be the mean feature vector of all the pixels belonging to the cluster. A residual error E between the new and the previous cluster centre locations is then computed. These steps are repeated iteratively until convergence. However, the authors claim that 10 iterations suffice for most images.

In order to measure the amount to which SLIC's performances can be improved, we follow the original algorithm, by fixing a threshold $0 \leq E_{max} \ll 1$. We then measure execution time (or, equivalently the number of iterations, which is directly proportional as shown in Figure 5-17) needed for convergence.

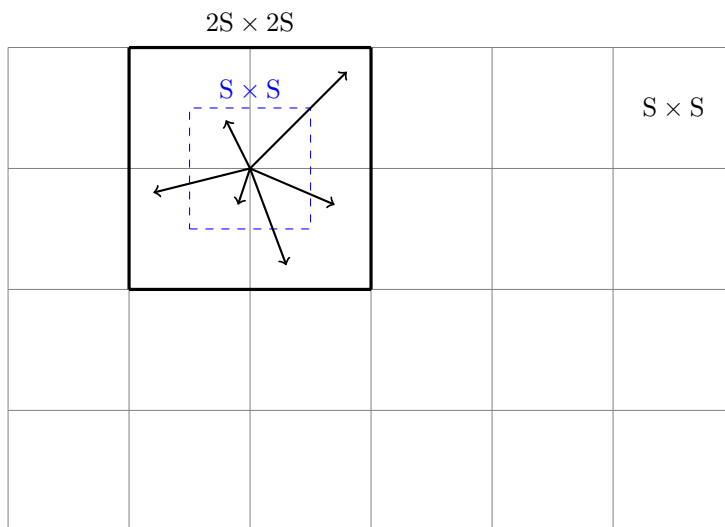


Figure 5-16: Unlike standard k -means algorithms, SLIC searches a limited $2S \times 2S$ region only. The expected superpixels' size is $S \times S$, as the initialization grid cells. S is derived from the image dimensions and the number of desired superpixels

Bayesian approach As we deal with videos, the most natural question we could ask ourselves is: can we exploit the information carried by a frame to process the next incoming? And, if yes, how?

A very simple answer is given by the *Bayesian filtering* framework, also known as *re-*

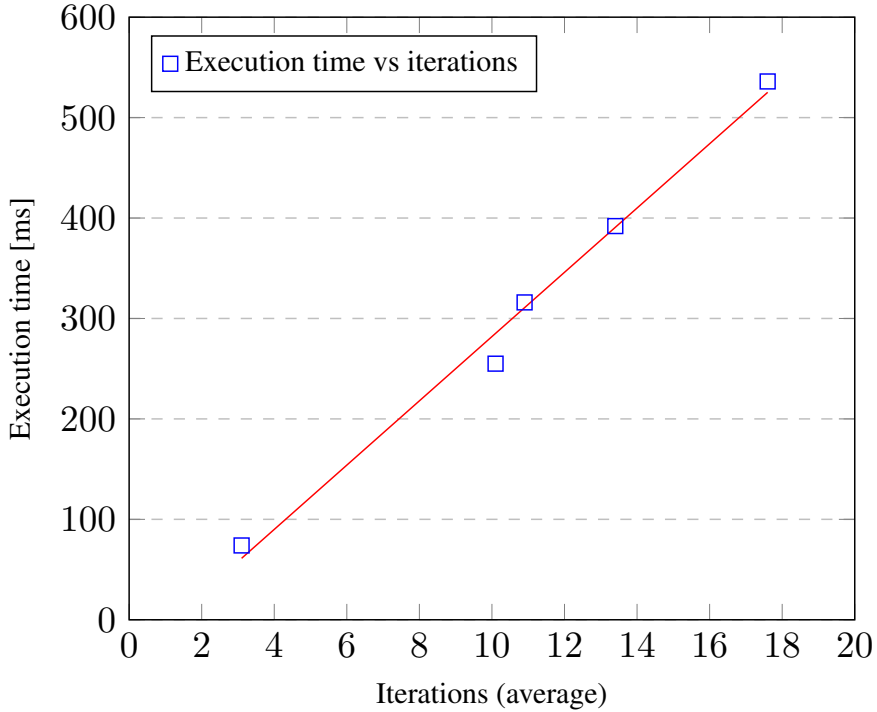


Figure 5-17: Execution time [ms] against average number of iterations required for convergence: the relation is approximately linear. Data are provided in table 5.3.

cursive Bayesian estimation. This well known probabilistic approach aims at estimating an unknown probability density function recursively over time using incoming measurements and a mathematical process model. Its simplest analytical implementation, the Kalman filter (KF) [109], is so widely employed that needs no introduction. Such filter provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modelled system is unknown. In particular, it consists of a *prediction* step, followed by an *update* step where the actual measure is incorporated in the estimation. Many extensions have been developed for non-linear equations and non-Gaussian noise (Extended KF, Unscented KF, Cubature Filter [10], Particle Filter [189]).

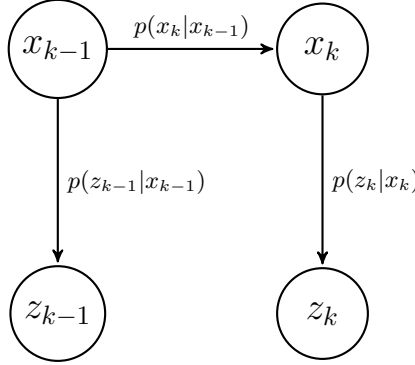


Figure 5-18: Graphical model (Dynamic Bayesian Network) for a Bayes filter.

The estimation of new cluster centres can be modelled as a Markov process (Figure 5-18), where the hidden state at discrete time k is the set of all cluster centres $\mathbf{x}_k = [\mathbf{l}_k, \mathbf{a}_k, \mathbf{b}_k, \mathbf{x}_k, \mathbf{y}_k]$ and the measurement z_k is the whole k -th frame. Applying SLIC independently on frames of a video is equivalent to skip the *prediction* step in a KF, relying on measurements only. We here instead try to model the process entirely, as it is common understanding that even a trivial prediction can remove a lot of noise from the process.

Consider the general time-discrete process, identifying the dynamic model of the system and the measurement model.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \quad (5.14)$$

$$z_k = Cx_k + Du_k + v_k, \quad (5.15)$$

$$\text{where } p(w_k) = \mathcal{N}(0, Q), \quad p(v_k) = \mathcal{N}(0, V). \quad (5.16)$$

In a KF, A, B, C, D are matrices (both models are linear, though usually B and D are zero) and the random variables w_{k-1} and v_k are the process and measurement noise respectively (Gaussian, zero-mean).

As for the measurement model C (modelling $p(z_k | x_k)$), we here assume it is the super-pixel algorithm itself which links the state $\mathbf{x}_k = [\mathbf{l}_k, \mathbf{a}_k, \mathbf{b}_k, \mathbf{x}_k, \mathbf{y}_k]$ and the measurement z_k (the whole frame). In this perspective SLIC can be associated to C^{-1} . It is not a linear measurement model, although SLIC does preserve linearity to some extent, as cluster centres x_k are averages over a certain fraction of pixels values.

On the other hand, for what concerns the dynamics ($p(k_k|x_{k-1})$), we enforce a linear model. Common practice when no knowledge on the actual dynamics of system is given (e.g a complex moving camera scenario, as in our case), is to simply allow a sufficient amount of noise over a constant state, i.e. $A = \mathbb{I}$. This means that we suppose that the cluster centre in the following frame will be somewhere in the surroundings (namely, in his previous position plus some shift extracted from $p(w_k)$). We note that this is not far from formulating SLIC as a Gaussian mixture as proposed in [44].

What we propose in this section is not a rigorous Bayesian state estimation (although we generously draw inspiration from the Kalman filter framework). We rather suggest that, instead of re-initializing the starting regular grid (Figure 5-16) in SLIC at each frame, a more effective choice could be to try a guess. Common practice in KFs tells us that if we do not have a clue on the precise dynamics of the problem, a still (statistically) good option is to suppose the state to be the same as in the previous time instant (plus some noise).

We are able to measure how better this new re-initialization method perform by evaluating the number of iterations needed for convergence at each frame. As shown in the next section, less iterations are needed, since, statistically, clusters centres are closer to the actual ones and SLIC needs less iterations to converge.

To conclude this section, we would like to stress again what we propose is not a Kalman filter, although it draws inspiration from it, in a not completely rigorous fashion, in order to cope with the practical issue of speeding a superpixel algorithm up to real time performances, which are needed for first-person vision applications.

Video coding approach A minor issue arises however when adopting the approach proposed in the previous section. As shown in Figures 5-19 and 5-20, small rectangular patterns appear in many superpixels after a while. The reason why these artifacts arise is to be ascribed to the fact that, as shown in Figure 5-16, SLIC does not implement an exact k -means clustering, but searches in a $2S \times 2S$ window. Therefore, due to divergent flows in the video scene, sometimes cluster centres are pull too far apart, leaving a gap between adjacent windows, which is not searched. Such a gap has a fairly regular shape, being delimited by the borders of two or more square boxes. Gaps' pixels are not elaborated in the current frame, and maintain the labels they were assigned in the previous time instant. This issue mainly arises around object moving in contrast with the general dynamic of the scene (e.g. hands performing gestures).

To overcome this drifting issue we adopt a strategy similar to the one exploited for

similar reasons in video codecs such as h264 [234]. Here I-frames (Intra frames or key frames) are employed to decode from scratch without reference to any other frame. Similarly, we insert I-frames, where SLIC is applied from scratch, in order to prevent the aforementioned drifting. In a Bayesian perspective, this is equivalent to regularly cut the Markov chain and force the model with a new prior, namely SLIC’s standard initialization grid.

We have found that sending I-frames at a ratio of 1/30 is usually enough. Increasing the dimension of the window used by SLIC would be a solution, but it should be done in an adaptive way, which would be quite expensive. In addition, the method should also deal with superpixels’ births and deaths once windows’ sizes grow too big. The issue of superpixels merging-splitting and birth-death mechanism is addressed in [44] and proves to be extremely time consuming.

Results

We run experiments on the egocentric video dataset provided by Kitani and colleagues [131]. The approximately six minute long video `EDSH1.avi` counts 11290 frames and records the perspective of a single user along different indoor and outdoor scenes, with really heterogeneous illumination conditions. Frames are 1280x720 pixels. SLIC parameters are set as follows $m = 30$, $N = 1000$. Residual error is $E = 0.25$ and it is of course fixed for comparison purposes.

Table 5.3: Performances

METHOD	Execution time (ms)	Number of iterations	fps
1. SLIC	537	17,6	1,9
2. NAIVE	74	3,1	13,5
3. NAIVE + INTRA	316	10,9	3,2
4. NOISE	256	10,1	3,9
5. NOISE + INTRA	393	13,4	2,5

Table 5.3 presents quantitative data averaged over the 11290 frames. Performances are referred to the tests we run on an Intel Xeon 2.66 GHz processor with 4 GB RAM. Our

code is publicly available at <https://github.com/ClaudiuGeorgiu/VideoSLIC>. Applying SLIC from scratch at each frame results in an execution time of more than half a second, which means not even 2 fps. In fact over 17 iterations are on average needed for having the k -means algorithm converge up to the residual error E .

By naively initializing SLIC's centre states with the values outputted by the previous frame (with no noise), produces the astonishing boosts in performances of almost 13 frames per second. This is to be ascribed to the fact that consecutive frames are extremely similar and, neglecting border effects and occlusions, superpixels require little effort to be adjusted. However, results are corrupted by some drifting effects as shown in Figure 5-19. The rectangular patterns or sharp angles appear after a while. This drifting phenomenon is due to the fact that SLIC does not implement an exact k -means clustering, but searches in $2S \times 2S$ windows, which can leave gaps in case of divergent flows. The absence of noise in dynamic model makes the phenomenon even more evident.

The issue of rectangular patterns becomes less predominant when injecting Gaussian noise in the dynamic model, as drifting is somehow mitigated by noise. Noise injection has of course a computational cost as random Gaussian-distributed numbers must be generated at each time step. In the test, noise was extracted from the distribution $\mathcal{N}(\mu = 0, \sigma = S/5)$. The higher the noise, the lower the drifting effect although of course more iterations are needed for convergence.

Inserting intra frames removes drifting effects both in the noisy and non-noisy case, at the price of increasing on average the number of iterations needed for convergence. Here again, the more frequently intra frames are sent, the slower the convergence. Results in Table 5.3 are obtained by sending I-frames at a frequency of $1/30$. However, the rate at which intra frames are to be inserted can depend on several factors: Superpixels' dimension, frame rate and Dynamics of the scene, to mention some. The algorithm is not self-adaptive on this parameter, which must be fixed heuristically, finding a correct trade-off between performance and other parameters.

Final considerations

In this subsection, we have presented a general framework for exploiting superpixels algorithms in videos. The approach does not simply consist in Bayesian filtering of cluster centres 5D positions, since the measurement model $p(z|x)$ is embedded in the superpixel algorithm itself. The practical drifting issue arising specifically in SLIC, due to the non exact nature of the k -means algorithm is addressed by periodically giving

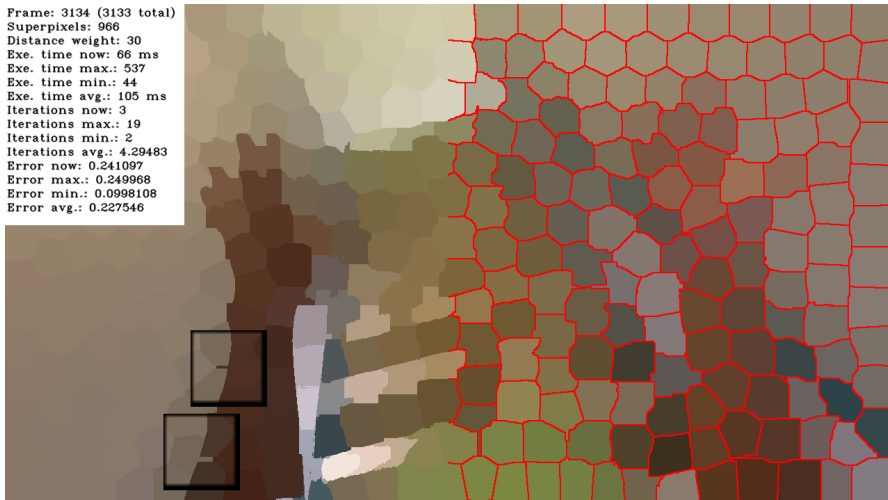


Figure 5-19: Rectangular patterns appear after a while. This drifting phenomenon is due to the fact that SLIC does not implement an exact k -means clustering, but searches in a $2S \times 2S$ window, which can leave gaps in case of divergent flows. The absence of noise in dynamic model makes the phenomenon even more evident



Figure 5-20: The issue of rectangular patterns is less predominant when injecting Gaussian noise in the dynamic model.

new priors $p(x_0)$ to the Bayesian filter. The inspiration comes from video coding where I-frames are periodically sent precisely to avoid drifting effects.

Substantial improvement in performance shows the bounty of the proposed approach, which was tested on Kitani’s egocentric activities dataset. This represents a good starting point in optimizing computer vision techniques for egocentric video analysis. Such optimization must so far be done on the software side, as dedicated hardware (as GPUs) is yet to be available on wearable devices.

Future research directions include the applications of the presented framework to other superpixels algorithms. However, some work can still be done on SLIC (which has already proved to be an extremely powerful and versatile algorithm in many applications), for instance by measuring the drifting and making the method self-aware and self-adaptive for what concerns I-frames and noise insertion.

Chapter 6

Left/Right Identification

Identification is an intuitive but challenging task, needed for many purposes ranging from activity to gesture recognition. The objective is to identify the left and the right hand in a manifold of complex scenarios which include the case hands are close enough to create a single shape that has to be split (occlusion disambiguation). This chapter investigates the issue, although the explorations is still in its early stage and both the presented discussion and the results are only preliminary.

6.1 Introduction and related work

The importance of hands in First Person Vision is well validated across the literature and have been discussed already in this thesis.

In the following, an essential bibliography is provided, comprising works which mention the problem of hand-identification. None of them, however, addresses the matter explicitly, reducing it to a minor post-processing issue. Recent methods and strategies to process these First Person Videos are summarized in [22]. In [20] a hierarchical structure to develop hand-based methods for wearable cameras was proposed and an extension of this work was presented in chapter 3. [153] shows that discriminating the active and passive objects makes it possible to improve the recognition rate. In their approach the active object is the one being manipulated by the user. [74] proposes a visual background-foreground segmentation based on graphcut. Subsequently [132, 131]

propose a pixel level hand-segmentation method based on color. Recently the authors in [251] propose a shape-aware classifier, and [19, 23] shows the importance of separating the steps of hand-detection and segmentation, since hands are not always framed in the scene. The pioneer work of [134] shows the strong link between the objects, hands and gaze. In the same line the authors in [36] shows how using the relative position of the hands is possible to infer the gaze of the user.

The hand-identification problem is extended in [128], proposing a Bayesian method to identify, using the relative positions, the hands of the user as well as the hands of a third person in the video. It is worth to mention the robustness of the proposed hand-detector to the presence of third person hands. However, in the segmentation level, extra effort must be done to segment only the user hands.

Although the problem of identification is somehow described in the provided papers, it is always addressed as a segmentation post-processing step and is usually based on naive heuristic rules. The following investigation tries to fill this gap in the literature with a more structured and rational approach, analyzing all the possible cases and solving issues related to hand-to-hand occlusion.

Hands identification is performed straight after the segmentation step, as discussed in chapter 3. However, the rigid structure proposed in Figure 3-4 show here one of its limits. Not only a post processing step will be needed after segmentation in order to remove noise, as detailed further on in this chapter; but also segmentation disambiguation is sometimes needed in case of hand-to-hand occlusion. Under this perspective, segmentation and identification blocks do have definite hierarchical order, but are exchanging valuable information, as modeled in Figure 3-10, establishing a sort of feedback loop between two different levels.

We found that in realistic scenarios the proposed approach properly differentiate the left and the right hand in almost all the frames at low computational cost. Two challenging situations are: i) The hands are close enough to create a single shape; ii) The appearance of hands is divided by an external object as a bracelet or a watch, creating several hand-like shapes.

6.2 Hands-Identity

At this stage we assume that detection has already taken place, and processed frames are only positive ones, as depicted in Figure 6-1. The specific segmentation algorithm is

not relevant¹: we only suppose that the corresponding block outputs a certain number of blobs, possibly including some false positive, which are removed in the post-processing step with the aid of some heuristics (e.g. very small blobs are removed).

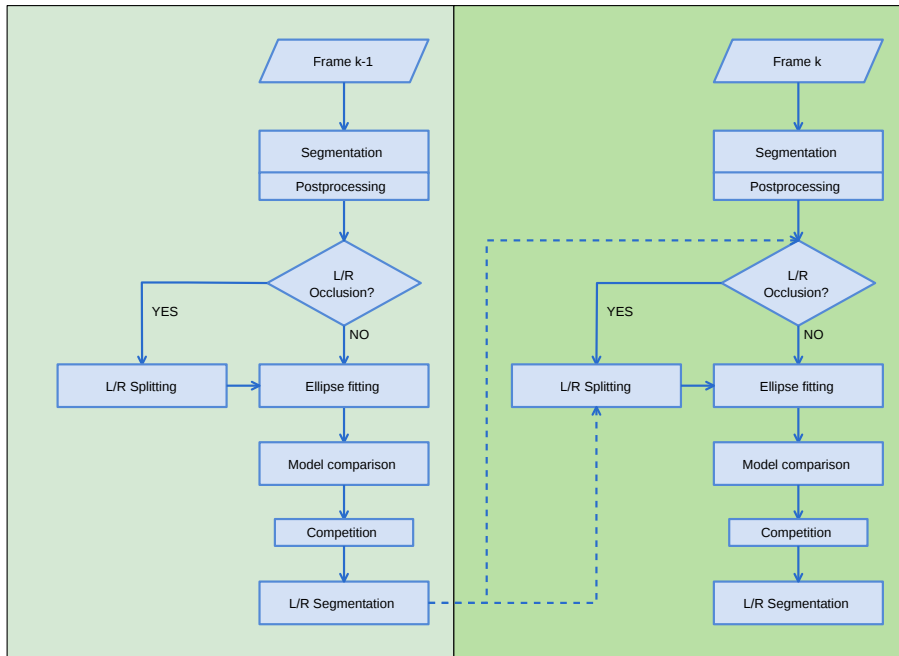


Figure 6-1: Block diagram of the proposed approach.

6.2.1 Building the L/R model

For what concerns the very first investigation, we employ manually-generated masks from the GTEA dataset [72], with manual left/right labeling (Figure 6-2), using the code provided by [132]. This simulates a perfect knowledge of L/R hand locations and shapes in each frame (occlusions are manually solved.) and allow the construction of two different ellipses (Figure 6-3). These are constructed with the OpenCV implementation of the ellipse fitting method described in [77].

¹In the results presented in the following, we developed a method derived from [131], which we will not discuss.

The two relevant parameters extracted from the ellipses are: i) the distance x of the centroid to the image border. ii) the angle θ of the major axis with respect to the image bottom horizontal border.

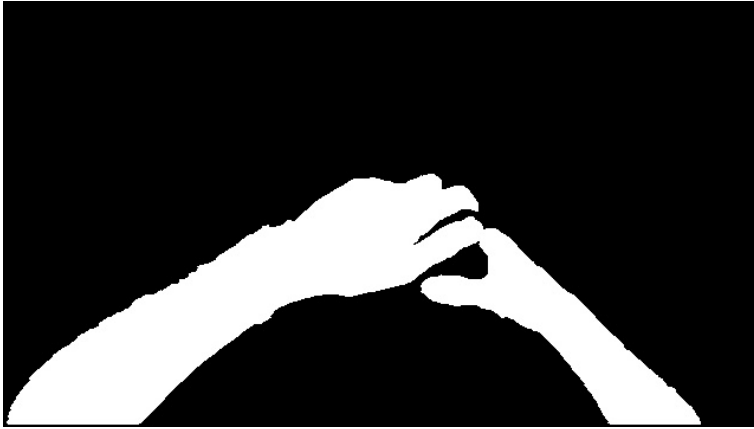


Figure 6-2: Manually segmented hands

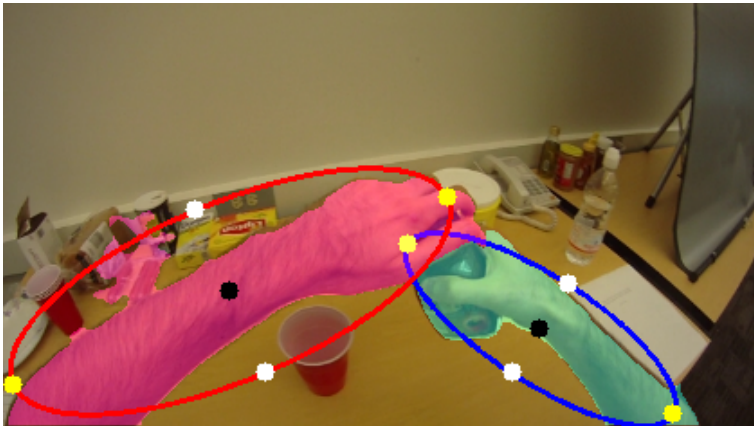
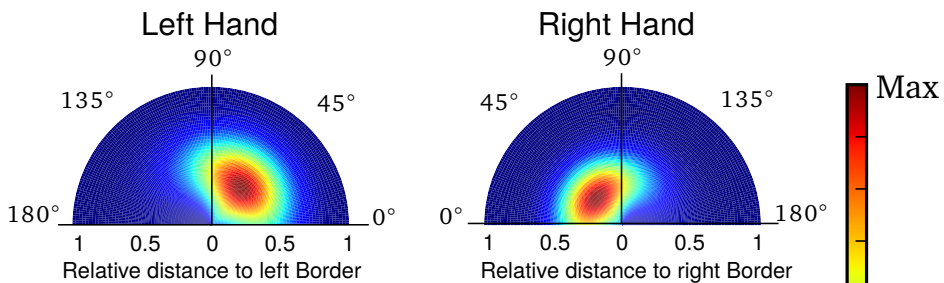


Figure 6-3: Fitting segmentation blobs with ellipses

The observed empirical distribution of the ellipses obtained from the manually generated masks is shown in figure 6-4 (top). Interestingly there is a small amount of asymmetry between the left and right distributions, meaning that one of the two hands is used for a wider variety of movements than the other. Although the problem is interesting, since it

could allow to personalize the models for right-handed and left-handed users on board of a (personal) wearable device, we have decided to neglect this fact in the current investigation.

Empirical Distribution



Proposed Model

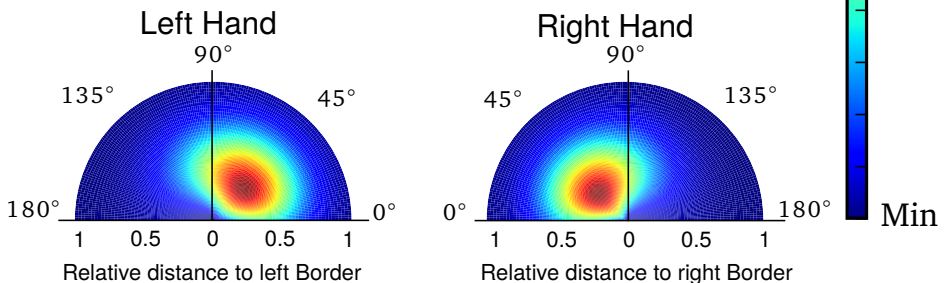


Figure 6-4: Empirical (**Top**) and theoretical (**Bottom**) hand distribution function given the distance to relative distance to the sides of the image. For the left(right) the relative distance to the left(right) side is used.

Based on these observation, we generate a mathematical model, trying to make it as similar to the observed distribution as possible. This is shown in the bottom part of figure 6-4. For both the angular and spatial dependencies, the shape is inspired by Maxwell distributions.

For what concerns the x parameter (position of the ellipses), the functions are:

$$p_l(x) = \sqrt{\frac{2}{\pi}} \frac{(x+dx)^2}{a^3} e^{-\frac{(x+dx)^2}{2a^2}}, \quad (6.1)$$

$$p_r(x) = \sqrt{\frac{2}{\pi}} \frac{(1-x+dx)^2}{a^3} e^{-\frac{(1-x+dx)^2}{2a^2}}. \quad (6.2)$$

Here x is the distance from the border of the image of ellipse's centroid (normalized with respect to the frame width); dx translates the Maxwell distribution to the side; a determines the width of the distribution (empirically set to $a = 0.22$).

The angular part is also described as a Maxwell distribution function, as follows:

$$p_l(\theta) = \sqrt{\frac{2}{\pi}} \frac{(\theta+d\theta)^2}{a^3} e^{-\frac{(\theta+d\theta)^2}{2a^2}} \quad (6.3)$$

$$p_r(\theta) = \sqrt{\frac{2}{\pi}} \frac{(\pi-\theta+d\theta)^2}{a^3} e^{-\frac{(\pi-\theta+d\theta)^2}{2a^2}} \quad (6.4)$$

where theta is the angle of the major axis of the ellipse calculated

Eventually the models plotted in Figure 6-4 (bottom) are:

$$p_l(x, \theta) = p_l(x)p_l(\theta) \quad (6.5)$$

$$p_r(x, \theta) = p_r(x)p_r(\theta) \quad (6.6)$$

If the functions 6.5 and 6.6 are normalized they can be interpreted as probability distributions.

6.2.2 Hands occlusions

As already mentioned, one of the main challenges of identification derives from hands touching or self-occluding. In this case, any segmentation algorithm would output a single big shape (actually, in some cases hands need not to touch: a single shape is created by the segmenter if hands are close enough).



Figure 6-5: Hand-to-hand occlusion: a single blob is created and thus a single ellipse is generated.

This situation is depicted in Figure 6-5: here a single ellipse would be outputted with characteristics which would be not compatible with the models discussed above. How do we detect it? As it can be seen from the work-flow diagram 6-1, the module detecting occlusions relies on L/R segmentation performed in the previous frame. Pseudocode is provided in algorithm 6.1.

Algorithm 6.1: How occlusions are detected.

Data: CurrentBigBlob, Previous L/R segmentation

Result: Occlusion: Y/N

```

if  $previousLeft \neq \emptyset$  AND  $previousRight \neq \emptyset$  then
    totalPrevious =  $previousLeft \cup previousRight$  ;
    intersection =  $CurrentBigBlob \cap totalPrevious$ ;
    if  $0.8 * Area(totalPrevious) \leq Area(intersection) \leq 1.2 * Area(totalPrevious)$  then
        | Occlusion  $\leftarrow$  YES;
    else
        | Occlusion  $\leftarrow$  NO;
else
    | Occlusion  $\leftarrow$  NO;

```

6.2.3 Segmentation disambiguation

Once occlusion have been determined, a module is dedicated to disambiguate the segmentation into left and right hands, namely to *split* the unified blob. At this stage we rely on the ability of a superpixel algorithm in adhering to objects' contours. Again, valuable information is received from the previous L/R segmentation step, which can provide hints about the L/R boundary. Superpixels are assigned a L/R id based on the superposition with the previous frame. The pseudocode for this processing block is outlined in algorithm 6.2.

Algorithm 6.2: How blobs are split in case of hand-to-hand occlusion.

Data: CurrentBigBlob, Previous L/R segmentation, Previous Superpixel segmentation
Result: Current L/R segmentation
 Get Superpixel clustering σ_i^k , $i = \{1 \dots N\}$ in the current frame k ;
for $i = 1 : N$ **do**
 if $\text{centroid}(\sigma_i^k) \in \text{CurrentBigBlob}$ **then**
 if $\text{centroid}(\sigma_i^k) \in \text{previousLeft} \cap$ **then**
 $\text{Id}(\sigma_i^k) = L$;
 else if $\text{centroid}(\sigma_i^k) \in \text{previousRight}$ **then**
 $\text{Id}(\sigma_i^k) = R$;
 else
 $\text{Id}(\sigma_i^k) = \text{Id}(\sigma_{\text{closest}}^{k-1})$; /* id of the closest sp in the
 previous frame */

6.3 Results

6.3.1 Perfect segmentation

As a first result, we want to show that, using the orientation and position of the segmented blobs, it is possible to accurately decide if the hand-like shapes in the frame are left or right hands. With this in mind we initially evaluate the decision functions presented in section 6.2.1 assuming the availability of a perfect segmentation and no occlusion ambiguity, i.e. using again manually labeled masks.

The two sections of Table 6.1 show results based on two different evaluation schemes. The confusion matrix on the left side refers to the id assignment based only on the best

Table 6.1: Left and right hand identification at contour level

	No-competition		With competition	
	Left	Right	Left	Right
Left	0.994	0.006	0.997	0.003
Right	0.012	0.988	0.000	1.000

fitting model, with no constraints. However, it is quite obvious that if two hands are present, they cannot be assigned the same id. The two blobs thus compete for both the left and the right model. Using this scheme, the classification rates are increased and even reach 100 % from the right hand. This appears at the bottom of the block diagram of Figure 6-1.

6.3.2 Disambiguating occlusions

To evaluate the occlusion detection block of Figure 6-1 we manually select the masks showing occlusion evidence. Those masks do not correspond to the video interval with hand occlusions, but are a good approximation to understand if the decisions rules presented in the last section are properly identifying difficult cases. In total the 5 testing videos contains 51 masks with evident hand occlusion. In total our decisions rules were able to identify 98% of them.

Actually, since in this case we employ an actual segmentation algorithm, we have a lot of false positive occlusions, that is hands are not touching, but are close enough to make the segmenter output a single shape. This is not a problem, since we simply split them as if they were occluded, losing only a little efficiency. Since the segmenter needs a training phase [131], all the results presented from now on will refer to only four videos of the GTEA dataset [72]: the fifth video, namely the "Coffee" sequence, is always the one used for training.

Table 6.2 presents the segmentation results only in the frames where occlusion is detected (the task is addressed as a three-class classification problem: left hand, right hand, background, thus results are given the shape of a confusion matrix describing pixels' assignments to classes).

Table 6.3 compares instead the case splitting is performed with the case it is not. The two

Table 6.2: Evaluation of hand segmentation when split is required

	Background	Left	Right
Background	0.984	0.007	0.009
Left	0.058	0.934	0.009
Right	0.080	0.006	0.914

confusion matrices are of course identical in the Background part, since the segmenter is the same in the two experiments. Segmentation accuracy gains almost ten percentage points when L/R disambiguation is performed. Eventually, table 6.4 provides the detailed results for each testing video.

Table 6.3: Confusion matrices with and without occlusion disambiguation.

	Without split			With split		
	Background	Left	Right	Background	Left	Right
Background	0.992	0.004	0.004	0.992	0.004	0.004
Left	0.073	0.821	0.106	0.073	0.923	0.004
Right	0.096	0.066	0.838	0.096	0.001	0.903

6.4 Conclusions and future research

Unlike most of the results presented in the other chapters of this thesis, the ones provided above have not been published yet, still representing the starting point for a more exhaustive discussion. However we have proved the benefit of hands occlusion disambiguation within the L/R identification problem. Results were presented in the shape of segmentation confusion matrices, although related algorithm was not reviewed in detail. As mentioned, the segmentation algorithm is not important itself, although a superpixel-based method would required less additional computation in the Occlusion Detection and L/R Splitting blocks.

Future developments of this chapter include F1 scores for evaluating segmentation and evaluation of the performance jointly with the detection block.

Table 6.4: Detailed segmentation results for each video. The "Coffe" sequence is used for training. Confusion matrix for each testing video and the overall result are provided at a pixel level. The results include occlusion detection, segmentation disambiguation (split) and id-competition.

CofHoney			Hotdog			Tea			Pealette			Total			
	BG	Left	Right	BG	Left	Right	BG	Left	Right	BG	Left	Right	BG	Left	Right
BG	0.990	0.003	0.007	0.989	0.005	0.006	0.996	0.002	0.002	0.991	0.006	0.003	0.992	0.004	0.004
Left	0.064	0.932	0.004	0.040	0.958	0.002	0.056	0.943	0.001	0.120	0.871	0.009	0.073	0.923	0.004
Right	0.092	0.002	0.906	0.136	0.001	0.864	0.082	0.000	0.918	0.112	0.002	0.886	0.096	0.001	0.903

Chapter 7

Hand Pose

*Hand pose recognition plays a fundamental role in tasks such as gesture and activity recognition, which in turn represent the base for developing human-machine interfaces or augmented reality applications. In this chapter we propose a graph-based representation of hands seen from the point of view of the user, obtained through the shape-fitting capability of a particular neural network. Spectral analysis of the graph Laplacian allows to arrange eigenvalues in vectors of features, which prove to be discriminative in classifying the hand poses considered.*¹

7.1 Pose and gesture recognition

Hand-related methods have quickly gained importance in FPV [21] since the spreading of wearable technology. Hands represent in fact one of the main interaction media with the surrounding environment [70] and thus also one of the most natural way to communicate with the device. Besides, most of people's activity do involve hands, which often perform gestures in the field of view of the users and of the body worn cameras [23]. Recognizing user's activity is believed to be essential in providing an augmented reality experience through the display of smart-glasses.

¹The results presented in this chapter have been submitted for publication to ICIP 2016 (23rd IEEE International Conference on Image Processing).

The problem of automatically recognizing hand poses has only recently been investigated in FPV by Kitani et al. [38]. The same authors propose a method to understand the functionality of human hands by analysing grasps poses using state of the art computer vision techniques [101]. In addition, their research exploits object recognition and hands-objects interaction analysis to deeper infer on users' activities [103]. Although the problem has already been addressed from third-person viewpoint, to the best of our knowledge the latter is the first attempt to analyse poses from the first-person perspective, even though only limited to grasp poses.

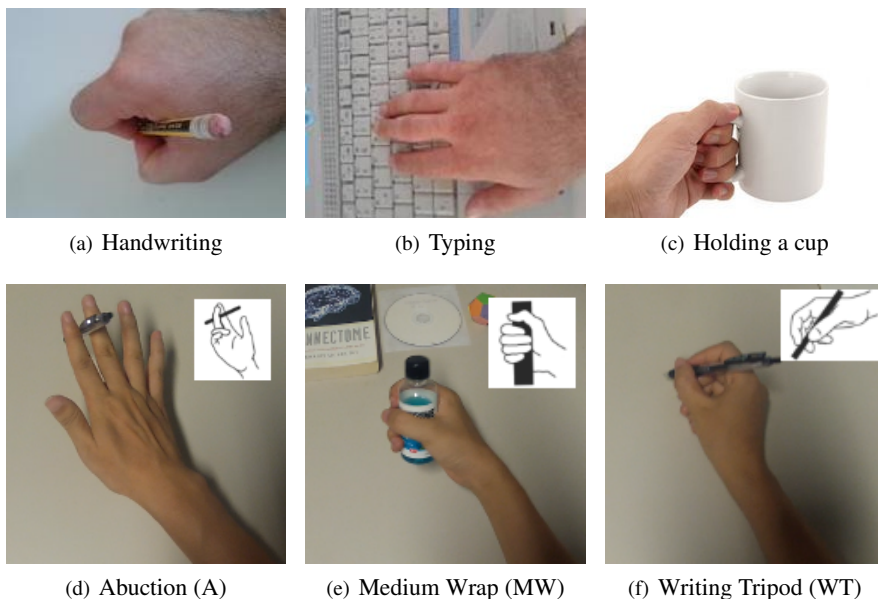


Figure 7-1: (a,b,c) Typical poses corresponding to the three different activities considered in our preliminary results (E1). (d,e,f) Three grasps of the UTC dataset [101] (masks are provided), corresponding to three different taxonomic categories [34] used in E2.

In this work, we propose a hand pose recognition method which exploits the property of graph-signals to encode shape information of objects. Representing objects with a graph is an idea which has become established with the growing interest in graph signal processing in the last few years. Such a representation is for instance used in visual tracking, where tracked features are encoded in a graph, whose tracking is accomplished by exploiting graph matching techniques [39] from one frame to the following. It is

indeed thanks to the growing interest in graph signal processing that techniques such as graph spectral analysis have been recently re-discovered. Just to mention, [143] uses graph spectral analysis for identifying properties of dynamically allocated data structures, while [8] even propose an extension the Nyquist-Shannon theory of sampling to signals defined on arbitrary graphs.

It is common understanding that gesture recognition methods should be based on a first segmentation step in order to separate the hand region from the background [131]. In the field of FPV both pixel-by-pixel [160] [132] and superpixel-based methods [200] [161] have been exploited to this end. In this chapter we employ a pixel-by-pixel approach, since single pixel will be used as inputs of a Neural Network in order to construct a graph representation of hands, as it will be detailed in section 7.2.2.

The remaining of this chapter is organized as follows: section 7.2 presents the proposed approach for hand pose recognition: a subsection is dedicated to each step of the proposed procedure; the experimental set-up and our findings are presented in section 7.3, while conclusions eventually are drawn in section 7.4.

7.2 Pose recognition framework

The proposed method follows the flow depicted in Figure 7-2. After an image is acquired, a colour segmentation step outputs a binary image of the segmented hand. Such image is given as input to an artificial neural network, which outputs a graph whose topology mirrors the shape of the hands. The graph Laplacian is then extracted and diagonalized, in order to compute its eigenvalues. A vector of ordered eigenvalues is then produced and given as input to a SVM classifiers. In the following we will go into the details of each processing step.

7.2.1 Colour segmentation

Input: Colour frame

Output: Binary image with hand segmentation

A manifold of segmentation methods have been proposed in computer vision. The aim here is to separate the foreground image represented by the hand region from the background. Being skin complexion extremely discriminative, we rely here on colour-based segmentation. Besides, it has been argued in [160] that being wearable devices personal

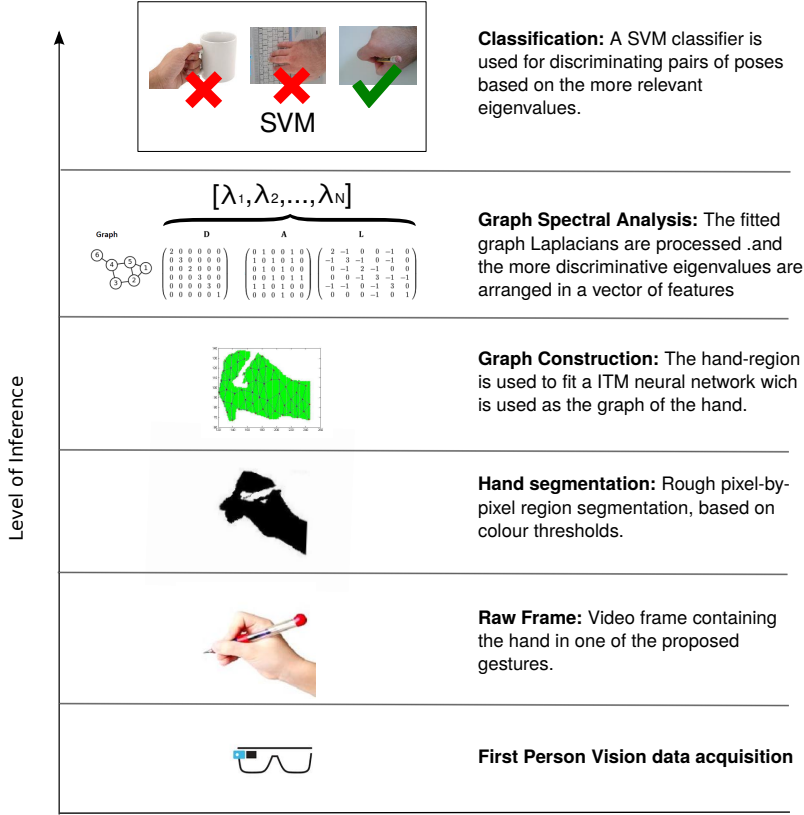


Figure 7-2: Work-flow diagram of the proposed method.

(precisely as smart-phones are), they can be trained specifically on the skin shade of the user.

Here we exploit a segmentation rule based on thresholds as proposed in [216], but we remark that the choice of the segmentation method is not the key aspect. For example, many approaches put great stress on smooth contours, which are not needed here. The aim is simply to extract as many pixels as possible from the hand region, but even a poor segmenter can allow in the next step to have a good hand representation, namely to construct a graph which is representative of the hand pose. Any method with low false positives rates could work as good. The reader can refer to chapter 5 for a more detailed

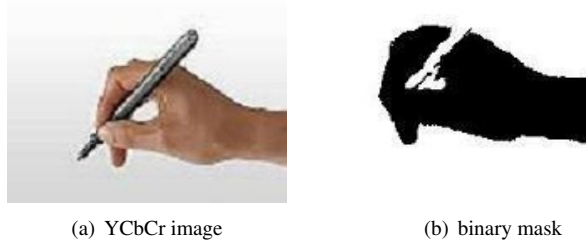


Figure 7-3: Segmentation step

discussion on segmentation methods and on chapter 2 for an extensive state-of-the art review.

The skin segmentation rules are devised in the $YCbCr$ colour space as suggested by both [216] and [160] and can be summarized as follows:

$$\begin{cases} Cb(i, j) \in R_{Cb} \cap (Cr(i, j) \in R_{Cr}) \Rightarrow (i, j) \in hand, \\ D_1 \cup D_2 \cup D_3 \cup D_4 < T \Rightarrow (i, j) \in hand \end{cases} \quad (7.1)$$

where T is a threshold and D_1, D_2 are the euclidean distances between $Cb(i, j)$ and the upper and lower bounds of the range R_{Cb} and D_3, D_4 are the equivalent for the Cr channel.

A typical output of this block is the binary image shown in Figure 7-3(b)

7.2.2 Graph construction

Input: Binary mask image

Output: Graph representing the hand shape

A graph is a structure composed by nodes connected by edges. We here consider non-directed graphs and take for granted that the reader has some basic knowledge about them.

As already mentioned, graphs has recently been employed for representing visual objects to be tracked [39]. The main idea is there that objects can be represented as an

ensemble of sub-parts (nodes), one related to the other (edges). We here propose that a geometrical shape can be encoded in a graph and such a representation has some discriminative power for what concerns the classification of hand pose. This block takes care of the constructions of such graph, exploiting the fitting capabilities of a particular neural network, namely a modified version of the Instantaneous Topological Map (ITM)² [105], depicted in Algorithm 7.1. Note how the edge adaptation step is modified with respect to Algorithm 5.1 to cope with concave shapes.

The network is fed with the pixels from the hand mask, which fire neurons making them adapt to the stimuli. Links are modified accordingly. The main difference from the standard ITM lies in the fact that the map is able to fit concave shapes, since links which do cross white regions are removed. Such links may appear when the parameter r_{max} is larger than a concavity, as it can be noticed from Figure 7-4(c). However, a very low r_{max} results in a very large number of nodes as in the case of 7-4(a). This is not desirable since it will produce a huge Laplacian matrix to be processed, as discussed in the next subsection. The value is here fixed to $r_{max} = 10$ in order to have a number of nodes in the range 50 – 100. The number of node can of course change among the different images considered, but it is of the same order of magnitude. For what concern the tuning of the other parameter ε , this is tuned heuristically and its value is fixed to $\varepsilon = 0.1$. For a detailed discussion on the tuning of the two parameters the reader can refer to [162].

In terms of computational complexity, the Matching step scales with the number of neurons, which can be implicitly controlled by the parameter r_{max} . Edge adaptation scales with the average number of neighbours, which is related to the dimensionality of the input data (two in this case). All other steps are independent of the number of neurons involved allowing the algorithm to execute reasonably fast even for large networks.

7.2.3 Graph spectral analysis

Input: Graph

Output: Vector of features

It is very common to deal with a graph in one of its matrix representation. The one employed here is the normalized version of the Laplacian, given by

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \quad (7.2)$$

²Please note that the modification is different from the one introduced in algorithm 5.1 of chapter 5.

Algorithm 7.1: Modified ITM to avoid outside edges. The network is initialized with 2 seeds in the input space, connected by an edge.

Data: input vector $\mathbf{x} = (i, j)$; set of N nodes with weights $\mathbf{w}_i = (x_i, y_i)$

Parameters: shift ε ; resolution r_{max} ;

Result: Network adapted to the new pixel \mathbf{x}

1. Matching: find nearest neighbour n and second nearest s ;

Initialize $d_n = MAX_VAL$ and $d_s = MAX_VAL - 1$

for $i = 1 : N$ **do**

$d = d(\mathbf{x}, \mathbf{w}_i)$;

if $d < d_n$ **then**

$d_s = d_n$;

$d_n = d$;

$s = n$;

$n = i$;

else if $d < d_s$ **then**

$d_s = d$;

$s = i$;

2. Weight adaptation:

$\mathbf{w}_n = \mathbf{w}_n + \varepsilon(\mathbf{x} - \mathbf{w}_n)$;

3. Edge adaptation:

if $n \leftrightarrow s$ **then**

 └ If not outsider link $n \leftrightarrow s$;

$N(n)$: set of connected neighbours of n

for $\forall j \in N(n)$ **do**

$S(\mathbf{w}_n, \mathbf{w}_j)$: Thales sphere through \mathbf{w}_n and \mathbf{w}_j ;

if $w_s \in S(w_n, w_i)$ **then**

 └ $n \leftrightarrow j$;

4. Node adaptation:

if $d(\mathbf{x}, \mathbf{w}_i) > r_{max}$ **then**

 add new node m with $w_m = x$;

$n \leftrightarrow m$;

if $d = d(\mathbf{w}_n, \mathbf{w}_s) < \frac{1}{2}r_{max}$ **then**

 └ remove node s ;

where D is the degree matrix (number of connection of each node on the diagonal) and A is the adjacency matrix (1 where a link exist, 0 elsewhere). Theory shows that the eigenvalues λ_i of this matrix (which are real, being it symmetric) have interesting property and if *ordered* from the smallest can provide information about the structure of the graph. They lie in the range $[0 : 2]$ and $\lambda_0 = 0$ with a multiplicity equal to the number of connected components of the graph; λ_1 carries some information about

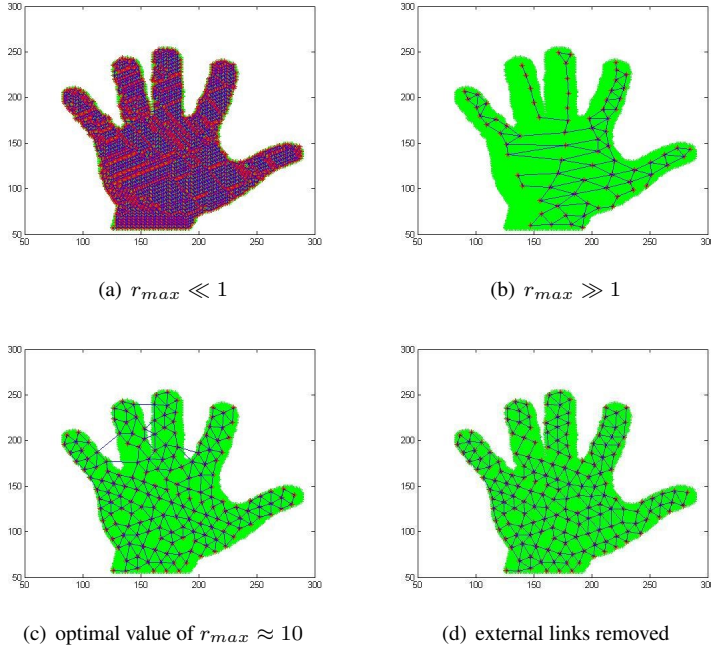


Figure 7-4: Graph construction through modified ITM algorithm (typing pose)

the general average connectivity of graph. Other eigenvalues cannot be given a precise meaning, but for sure they encode valuable information about the graph structure. This is why, starting from λ_1 , we consider a vector of Laplacian eigenvalues as features to discriminate among hands poses. In Section 7.3 we will verify that just the smallest 5 eigenvalues can be enough for accomplishing this task in two cases.

Diagonalization of matrices is a $O(n^3)$ -complex problem, this is the reason why the number of nodes must be contained within a reasonable amount through the tuning of the r_{max} parameter.

7.2.4 Classification

Input: Vector of features

Output: Pose id

As a first experiment, we compare pairs of poses, by training different binary SVMs. This is a first step to verify whether the proposed feature representation is discriminative enough for the problem of hand pose recognition. A more applicable method will need a structured multi-class approach.

7.3 Major findings

7.3.1 Preliminary investigation (E1)

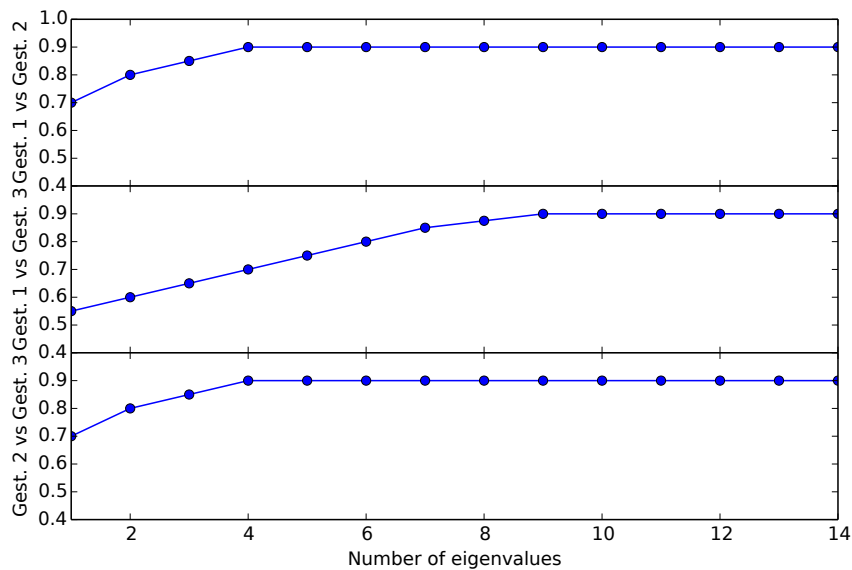
Three different gestures with three typical poses are considered for a preliminary investigation. Samples of the employed images are shown in Figure 7-1. For simplicity we will refer to them as to pose 1, 2 and 3 respectively. A set of 40 heterogeneous pictures was collected for each pose. After extracting graph eigenvalues from each image, as illustrated in the previous section, poses were compared pair by pair, in a k -fold validation framework (more precisely we adopted the leave-one-out scheme).

Table 7.1: Pairwise confusion of the classification accuracy with 10 eigenvalues (E1).

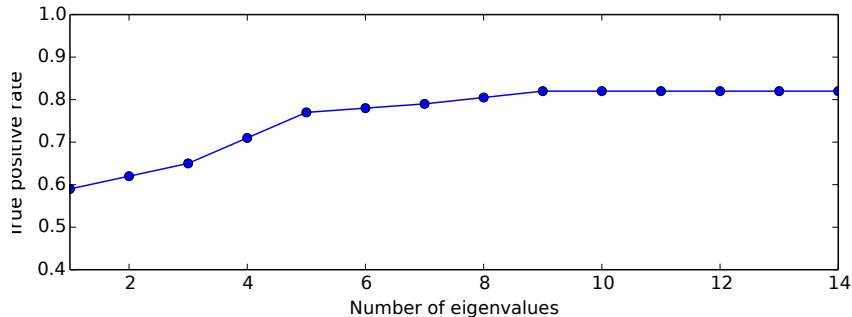
(a) Pose 1 vs Pose 2				(b) Pose 1 vs Pose 3			
		SVM				SVM	
		Pose 1	Pose 2			Pose 1	Pose 3
Value	Pose 1	0.85	0.15	Value	Pose 1	0.80	0.20
	Pose 2	0.05	0.95		Pose 3	0.00	1.00
(c) Pose 2 vs Pose 3							
		SVM					
		Pose 2	Pose 3				
Value	Pose 2	0.95	0.05				
	Pose 3	0.15	0.85				

Classification accuracies for the three experiments are reported in Figure 7-5(a), for different number of eigenvalues considered. Using only the first non-zero eigenvalue yields of course very poor results, since the number of connections within the graphs are

comparable. By adding more and more information, in the form of new eigenvalues in the vector, accuracy increases till it stabilizes to the best performance.



(a) E1



(b) E2

Figure 7-5: The effect of the number of eigenvalues on the accuracy of the classifiers in the two experiments.

It can be noticed how in the case of Gesture 1 vs 3 more eigenvalues are needed for the classifier to reach the maximum performance. This is easily explained by the fact that the two poses are more similar, being the hand closed in a grasp in both cases.

Tables 7.1(a)(b)(c) show the confusion matrices for the three experiments. The most unbalanced case is the one of Gesture 1 vs 3.

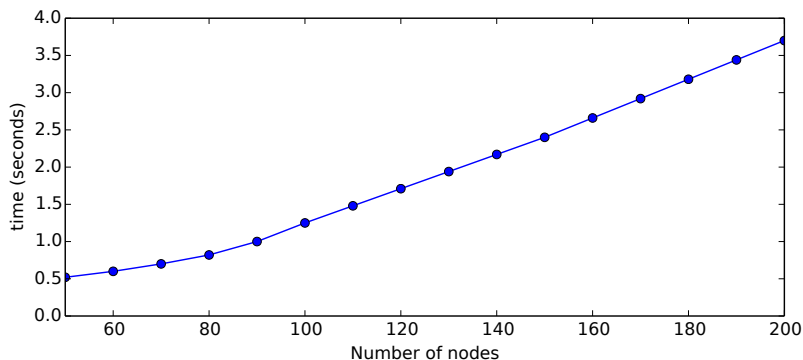


Figure 7-6: Execution time

For what concerns the complexity of the method, the most time consuming step is graph construction, as already discussed in subsection 7.2.2: it strongly depends on the number of nodes, to be controlled by r_{max} . Such a number also influences the complexity of the subsequent matrix diagonalization. Figure 7-6 show the execution time of the algorithm for different number of nodes. By limiting the average quantity of nodes generated by the graph construction step one could potentially further reduce complexity. Please note the time refers to non-optimized MATLAB code.

7.3.2 UTC dataset (E2)

Given the encouraging results of the preliminary investigation, we proceeded by validating our findings on a more structured public dataset [101]. For each of the 3 categories proposed in [34] we chose one kind of grasp (Fig. 7-1). Since the UTC dataset provides masks, we here skip the segmentation step and jump to the graph formation directly. A one-versus-all SVM is trained, again by taking an increasing number of eigenvalues. The accuracy is plotted in Fig. 7-5(b), while four confusion matrices are shown in Table 7.2.

Table 7.2: Confusion matrices of the classification accuracy (E2).

(a) 1 eigenvalue					(b) 5 eigenvalues				
		SVM					SVM		
		A	MW	WT			A	MW	WT
Value	A	0.00	0.39	0.61	Value	A	0.54	0.34	0.12
	MW	0.00	0.95	0.05		MW	0.03	0.97	0.00
	WT	0.00	0.19	0.81		WT	0.12	0.10	0.78
(c) 9 eigenvalues					(d) 13 eigenvalues				
		SVM					SVM		
		A	MW	WT			A	MW	WT
Value	A	0.69	0.20	0.11	Value	A	0.69	0.20	0.11
	MW	0.03	0.96	0.00		MW	0.03	0.96	0.00
	WT	0.13	0.10	0.77		WT	0.13	0.10	0.77

7.4 Conclusion

In this chapter we have shown how a graph representation of hand shapes can be exploited in a pose recognition problem. Vectors of graph Laplacian eigenvalues proved to be robust features in discriminating pairs of poses. In the case of similar poses, more information is needed for reaching an optimal classification accuracy.

Results are encouraging although far from real time, which can be approached by optimizing our Matlab code and by migrating to better performing programming language.

Still, a more applicable method will need a structured multi-class approach [100] with a more extended dataset (more gestures). Another interesting future research line may include an analysis of the robustness of the graph eigenvalue representation against segmentation noise.

Chapter 8

Conclusion and Future Work

This thesis have investigated hand-related methods in First Person Vision, as a way for providing new functionalities to wearable devices. Inspired by a detailed state-of-the-art investigation, a unified hierarchical structure was proposed, that optimally organizes processing levels to reduce the computational cost of the system. Such structure was also extended borrowing concept from the theory of Cognitive Dynamic Systems. Most of the levels sketched in the global framework were also deeply investigated. For the proposed algorithms specific conclusions have been drawn in the dedicated chapters.

8.1 Summary of contribution and major findings

Main contribution of this thesis are summarized as follows

- A detailed and comprehensive review of First Person Vision and its evolution was presented, together with a categorization of methods and a discussion of challenges and opportunity within the field.
- A global framework for hand-related egocentric-vision methods was introduced. A hierarchical structure was proposed for the design of a First Person Vision system and some hints were given on how to provide it with cognitive functionalities.
- Four levels of the proposed framework were fully investigated at algorithmic level, namely:

- *Hand detection* level: a classifier for detecting hands' presence in frames was developed. Temporal smoothing of the underlying decision process allows to increase performances of the algorithm.
- *Hand segmentation* level: a naive investigation on color as a discriminative feature for segmentation was carried on. A superpixel algorithm was developed (GSP), with the purpose of segmenting hands in each frame; a general optimization scheme was eventually design for exploiting temporal correlation of frames in videos.
- *Hand identification* level: an identification algorithm based on a position-angle model was presented, which is able to distinguish left from right hand with high precision.
- *Hand pose recognition* level: a graph representation of hands was investigated and proved to be effective in providing discriminative features in a multiclass pose classification problem.

8.2 Future developments

The work presented in this thesis has no claim of being comprehensive and in fact opens the way to several interesting future research topics. Most of them, pertaining to the specific algorithms, have been already highlighted in the dedicated chapters.

An interesting general line of research concerns the implementation of the Control side of the Cognitive framework proposed, which was only superficially treated in this thesis. As already mentioned, such an implementation is strictly algorithmic dependent, since both measuring the performances of the Perceptor units and selecting appropriate cognitive action are very specific issues.

In addition, many of the levels of inference here proposed (tracking, interaction analysis and higher levels) have not been investigated at all. Any possible future investigation at higher level should be however more application-oriented than the one carried on in this work, which, sticking to the lower levels of the hierarchy, could disregard the final objective.

Bibliography

- [1] Pieter Abbeel and Adam Coates. Discriminative Training of Kalman Filters. In *Robotics: Science and Systems*, pages 1–8, Cambridge, MA, USA, 2005.
- [2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.
- [3] O Aghazadeh, J Sullivan, and S Carlsson. Novelty Detection from an Ego-centric Perspective. In *Computer Vision and Pattern Recognition*, pages 3297–3304, Pittsburgh, PA, June 2011. Ieee.
- [4] K Aizawa. Digitizing Personal Experiences: Capture and Retrieval of Life Log. In *International Multimedia Modelling Conference*, pages 10–15. Ieee, 2005.
- [5] K Aizawa, K Ishijima, and M Shiina. Summarizing Wearable Video. In *International Conference on Image Processing*, volume 2, pages 398–401. Ieee, 2001.
- [6] Stefano Alletto, G Serra, Simone Calderara, and Rita Cucchiara. Head Pose Estimation in First-Person Camera Views. In *International Conference on Pattern Recognition*, page 4188. IEEE Computer Society, 2014.
- [7] Stefano Alletto, G Serra, Simone Calderara, Francesco Solera, and Rita Cucchiara. From Ego to Nos-Vision: Detecting Social Relationships in First-Person Views. In *Computer Vision and Pattern Recognition*, pages 594–599. Ieee, June 2014.
- [8] A. Anis, A. Gadde, and A. Ortega. Towards a sampling theorem for signals on arbitrary graphs. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3864–3868, May 2014.

- [9] H Aoki, B Schiele, and A Pentland. Realtime Personal Positioning System for a Wearable Computer. In *Wearable Computers*, pages 37–43, San Francisco, CA, USA, 1999. IEEE Comput. Soc.
- [10] I. Arasaratnam and Simon Haykin. Cubature kalman filters. *Automatic Control, IEEE Transactions on*, 54(6):1254–1269, 2009.
- [11] Ido Arev, Hyun Soo Park, Yaser Sheikh, Jessica Hodgins, and Ariel Shamir. Automatic Editing of Footage from Multiple Social Cameras. *ACM Transactions on Graphics*, 33(4):1–11, July 2014.
- [12] R Bane and T Hollerer. Interactive Tools for Virtual X-Ray Vision in Mobile Augmented Reality. In *International Symposium on Mixed and Augmented Reality*, pages 231–239. Ieee, 2004.
- [13] R Bane and M Turk. Multimodal Interaction with a Wearable Augmented Reality System. *Computer Graphics and Applications*, 26(3):62–71, 2006.
- [14] Lorenzo Baraldi, Francesco Paci, and G Serra. Gesture Recognition in Ego-Centric Videos using Dense Trajectories and Hand Segmentation. In *Computer Vision and Pattern Recognition*, pages 688–693, Columbus, Ohio, 2014. IEEE Computer Society.
- [15] Joseph W. Barker and James W. Davis. Temporally-Dependent Dirichlet Process Mixtures for Egocentric Video Segmentation. In *Computer Vision and Pattern Recognition*, pages 571–578. Ieee, June 2014.
- [16] G Ben-Artzi, M Werman, and S Peleg. Event Matching from Significantly Different Views using Motion Barcodes. *arXiv preprint*, December 2015.
- [17] Yoshua Bengio and Yves Grandvalet. No Unbiased Estimator of the Variance of k-fold Cross-Validation. *The Journal of Machine Learning Research*, 5:1089–1105, 2004.
- [18] T.L. Berg and D.A. Forsyth. Animals on the Web. In *Computer Vision and Pattern Recognition*, volume 2, pages 1463–1470. IEEE, 2006.
- [19] A. Betancourt, M.M. Lopez, C.S. Regazzoni, and M. Rauterberg. A sequential classifier for hand detection in the framework of egocentric vision. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 600–605, June 2014.

- [20] A Betancourt, P Morerio, L Marcenaro, E Barakova, M Rauterberg, and C.S. Regazzoni. Towards a Unified Framework for Hand-based Methods in First Person Vision. In *IEEE International Conference on Multimedia and Expo (Workshops)*, Turin, 2015. IEEE.
- [21] A. Betancourt, P. Morerio, L. Marcenaro, M. Rauterberg, and C. Regazzoni. Filtering svm frame-by-frame binary classification in a detection framework. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2552–2556, Sept 2015.
- [22] A Betancourt, P Morerio, C.S. Regazzoni, and M Rauterberg. The Evolution of First Person Vision Methods: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2015.
- [23] Alejandro Betancourt, Pietro Morerio, EmiliaI. Barakova, Lucio Marcenaro, Matthias Rauterberg, and CarloS. Regazzoni. A dynamic approach and a new dataset for hand-detection in first person vision. In George Azzopardi and Nicolai Petkov, editors, *Computer Analysis of Images and Patterns*, volume 9256 of *Lecture Notes in Computer Science*, pages 274–287. Springer International Publishing, 2015.
- [24] V Bettadapura, I Essa, and C Pantofaru. Egocentric Field-of-View Localization Using First-Person Point-of-View Devices. In *Winter Conference on Applications of Computer Vision*, volume Jan, pages 626–633, Waikoloa, HI, 2015.
- [25] VP Bhuvana. Distributed object tracking based on square root cubature H-infinity information filter. In *Information Fusion*, pages 1 – 6, Salamanca, 2014. IEEE Signal Processing.
- [26] S. Bilal, R. Akmeliawati, M.J.E. Salami, A.A. Shafie, and E.M. Bouhabba. A hybrid method using haar-like and skin-color algorithm for hand posture detection, recognition and tracking. In *Mechatronics and Automation (ICMA), 2010 International Conference on*, pages 934–939, Aug.
- [27] Igor Bisio, Alessandro Delfino, Fabio Lavagetto, and Mario Marchese. Opportunistic detection methods for emotion-aware smartphone applications. *Creating Personal, Social, and Urban Awareness Through Pervasive Computing*, page 53, 2013.
- [28] I. Bloch and H. Maitre. Data fusion in 2d and 3d image processing: an overview. In *Computer Graphics and Image Processing, 1997. Proceedings., X Brazilian Symposium on*, pages 127–134, Oct.

- [29] M Blum, A Pentland, and G Tröster. InSense : Life Logging. *MultiMedia*, 13(4):40–48, 2006.
- [30] M Bolanos, Maite Garolera, and Petia Radeva. Video Segmentation of Life-Logging Videos. In *Articulated Motion and Deformable Objects*, pages 1–9, Palma de Mallorca, Spain, 2014. Springer Verlag.
- [31] A Borji, D Sihite, and L Itti. Probabilistic Learning of Task-Specific Visual Attention. In *Computer Vision and Pattern Recognition*, pages 470–477, Providence, RI, June 2012. Ieee.
- [32] Hugo Boujut, J Benois-Pineau, and Remi Megret. Fusion of Multiple Visual Cues for Visual Saliency Extraction from Wearable Camera Settings with Strong Motion. *Internantional Conference on Computer Vision*, pages 436–445, 2012.
- [33] G Bradski. Real Time Face and Object Tracking as a Component of a Perceptual User Interface. In *Applications of Computer Vision*, pages 14–19. IEEE, 1998.
- [34] Ian M. Bullock, Thomas Feix, and Aaron M. Dollar. Finding small, versatile sets of human grasps to span common objects. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1068–1075, 2013.
- [35] Vincent Buso, Jenny Benois-Pineau, and Jean-Philippe Domenger. Geometrical Cues in Visual Saliency Models for Active Object Recognition in Egocentric Videos. In *International Workshop on Perception Inspired Video Processing*, pages 9–14, New York, New York, USA, 2014. ACM Press.
- [36] Vincent Buso, Iván González-Díaz, and Jenny Benois-Pineau. Goal-oriented top-down probabilistic visual attention model for recognition of manipulated objects in egocentric videos. *Signal Processing: Image Communication*, 39(June), 2015.
- [37] D Byrne, A Doherty, and C Snoek. Everyday Concept Detection in Visual Lifelogs: Validation, Relationships and Trends. *Multimedia Tools and Applications*, 49(1):119–144, 2010.
- [38] Minjie Cai, Kris M Kitani, and Yoichi Sato. A scalable approach for understanding the visual structures of hand grasps. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1360–1366. IEEE, 2015.
- [39] Zhaowei Cai, Longyin Wen, Zhen Lei, N. Vasconcelos, and S.Z. Li. Robust deformable and occluded object tracking with dynamic graph. *Image Processing, IEEE Transactions on*, 23(12):5497–5509, Dec 2014.

- [40] F Camastra and A Vinciarelli. *Machine Learning for Audio, Image and Video Analysis: Theory and Applications*. Springer, 2007.
- [41] R Castle, D Gawley, G Klein, and D Murray. Towards Simultaneous Recognition, Localization and Mapping for Hand-held and Wearable Cameras. In *Conference on Robotics and Automation*, pages 4102–4107. Ieee, April 2007.
- [42] R Castle, D Gawley, G Klein, and D Murray. Video-rate Recognition and Localization for Wearable Cameras. In *British Machine Vision Conference*, pages 112.1–112.10, Warwick, 2007. British Machine Vision Association.
- [43] Robert Castle, Georg Klein, and David W. Murray. Video-rate Localization in Multiple Maps for Wearable Augmented Reality. In *International Symposium on Wearable Computers*, pages 15–22, Pittsburgh, PA, 2008. IEEE.
- [44] J. Chang, Donglai Wei, and J.W. Fisher. A video representation using temporal superpixels. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2051–2058, June 2013.
- [45] Rachid Chelouah and Patrick Siarry. Genetic and Nelder–Mead algorithms hybridized for a more accurate global optimization of continuous multim minima functions. *European Journal of Operational Research*, 148(2):335–348, July 2003.
- [46] S. Chiappino, P. Morerio, L. Marcenaro, E. Fuiano, G. Repetto, and C. Regazzoni. A multi-sensor cognitive approach for active security monitoring of abnormal overcrowding situations in critical infrastructure. *15th International Conference on Information Fusion*, July 2012.
- [47] Simone Chiappino, Lucio Marcenaro, Pietro Morerio, and Carlo Regazzoni. Event Based Switched Dynamic Bayesian Networks for Autonomous Cognitive Crowd Monitoring. In *Augmented Vision and Reality*, Augmented Vision and Reality, pages 1–30. Springer Berlin Heidelberg, 2013.
- [48] Simone Chiappino, Pietro Morerio, Lucio Marcenaro, and Carlo S. Regazzoni. A bio-inspired Knowledge Representation Method for Anomaly Detection in Cognitive Video Surveillance Systems. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 242–249, July 2013.
- [49] Simone Chiappino, Pietro Morerio, Lucio Marcenaro, and Carlo S. Regazzoni. Bio-inspired relevant interaction modelling in cognitive crowd management. *Journal of Ambient Intelligence and Humanized Computing*, Feb(1):1–22, February 2014.

- [50] B Clarkson, K Mase, and A Pentland. Recognizing User Context Via Wearable Wensors. In *Digest of Papers. Fourth International Symposium on Wearable Computers*, pages 69–75, Atlanta, GA, USA, 2000. IEEE Comput. Soc.
- [51] B Clarkson and A Pentland. Unsupervised Clustering of Ambulatory Audio and Video. In *Acoustics, Speech, and Signal Processing*, pages 1520–6149, Phoenix, AZ, 1999. IEEE.
- [52] N Dalal and B Triggs. Histograms of Oriented Gradients for Human Detection. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. Ieee, 2005.
- [53] Dima Damen, Andrew Gee, Walterio Mayol-Cuevas, and Andrew Calway. Ego-centric Real-time Workspace Monitoring using an RGB-D camera. In *RSJ International Conference on Intelligent Robots and Systems*, pages 1029–1036. IEEE, October 2012.
- [54] Dima Damen and Osian Haines. Multi-User Egocentric Online System for Unsupervised Assistance on Object Usage. In *European Conference on Computer Vision*, 2014.
- [55] A.P. Dani, Zhen Kan, N.R. Fischer, and W.E. Dixon. Structure and motion estimation of a moving object using a moving camera. In *American Control Conference (ACC), 2010*, pages 6962–6967, 30 2010-July 2.
- [56] A Davison, I Reid, N Molton, and O Stasse. MonoSLAM: Real-time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–67, June 2007.
- [57] C de Boer, J van der Steen, R J Schol, and J J M Pel. Repeatability of the timing of eye-hand coordinated movements across different cognitive tasks. *Journal of neuroscience methods*, 218(1):131–8, August 2013.
- [58] A. Del Bue, D. Comaniciu, V. Ramesh, and C. Regazzoni. Smart cameras with real-time video object generation. In *IEEE International Conference on Image Processing*, volume 3, pages III/429–III/432, 2002.
- [59] R DeVaul, A Pentland, and V Corey. The Memory Glasses: Subliminal Vs. Overt Memory Support with Imperfect Information. In *IEEE International Symposium on Wearable Computers*, pages 146–153. Ieee, 2003.

- [60] M Devyver, A Tsukada, and T Kanade. A Wearable Device for First Person Vision. In *International Symposium on Quality of Life Technology*, volume Jul, pages 1–6, 2011.
- [61] A Doherty and N Caprani. Passively Recognising Human Activities Through Lifelogging. *Computers in Human Behavior*, 27(5):1948–1958, 2011.
- [62] A Doherty, S Hodges, and A King. Wearable Cameras in Health. *American Journal of Preventive Medicine*, 44(3):320–323, 2013.
- [63] P Dollár and C Wojek. Pedestrian Detection: a Benchmark. In *Computer Vision and Pattern Recognition*, pages 304–311, Miami, FL, 2009. IEEE.
- [64] A Dore, M Soto, and C Regazzoni. Bayesian Tracking for Video Analytics. *Signal Processing Magazine*, 5(27):46–55, 2010.
- [65] Alessio Dore, Matteo Pinasco, Lorenzo Ciardelli, and Carlo S. Regazzoni. A bio-inspired system model for interactive surveillance applications. *JAISE*, 3(2):147–163, 2011.
- [66] V Dovgalecs, R Megret, H Wannous, and Y Berthoumieu. Semi-supervised Learning for Location Recognition from Wearable Video. In *International Workshop on Content Based Multimedia Indexing*, pages 1–6. Ieee, June 2010.
- [67] F. Drucker and J. MacCormick. Fast superpixels for video analysis. In *Motion and Video Computing, 2009. WMVC '09. Workshop on*, pages 1–8, Dec 2009.
- [68] J Farrington and V Oni. Visual Augmented Memory. In *International Symposium on wearable computers*, pages 167–168, Atlanta GA, 2000.
- [69] M. Fatemi and S. Haykin. Cognitive control: Theory and application. *Access, IEEE*, 2:698–710, 2014.
- [70] A Fathi, A Farhadi, and J Rehg. Understanding Egocentric Activities. In *International Conference on Computer Vision*, pages 407–414. IEEE, November 2011.
- [71] A Fathi, J Hodgins, and J Rehg. Social Interactions: A First-Person Perspective. In *Computer Vision and Pattern Recognition*, pages 1226–1233, Providence, RI, June 2012. IEEE.
- [72] A Fathi, Y Li, and J Rehg. Learning to Recognize Daily Actions Using Gaze. In *European Conference on Computer Vision*, pages 314–327, Florence, Italy, 2012. Georgia Institute of Technology.

- [73] A Fathi and J.M. Rehg. Modeling Actions through State Changes. In *Computer Vision and Pattern Recognition*, pages 2579–2586. Ieee, June 2013.
- [74] A Fathi, X Ren, and J Rehg. Learning to Recognize Objects in Egocentric Activities. In *Computer Vision and Pattern Recognition*, pages 3281–3288, Providence, RI, June 2011. IEEE.
- [75] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, September 2004.
- [76] S Feng, R Caire, B Cortazar, M Turan, A Wong, and A Ozcan. Immunochromatographic Diagnostic Test Analysis Using Google Glass. *ACS nano*, 1(1), February 2014.
- [77] Andrew Fitzgibbon and Robert B. Fisher. A buyer’s guide to conic fitting. In *British Machine Vision Conference*, pages 513–522, 1995.
- [78] G. L. Foresti, C. S. Regazzoni, and P. K. Varshney. *Multisensor Surveillance Systems: The Fusion Perspective*. Kluwer Academic, Boston, 2003.
- [79] G.L. Foresti and C.S. Regazzoni. Multisensor data fusion for autonomous vehicle navigation in risky environments. *IEEE Transactions on Vehicular Technology*, 51(5):1165–1185, 2002.
- [80] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.
- [81] Bernd Fritzke. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, 1995.
- [82] Brian Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 670–677, Sept 2009.
- [83] Wen Gao, Xiaogang Chen, Qixiang Ye, and Jianbin Jiao. Pedestrian detection via part-based topology model. *19th IEEE International Conference on Image Processing (ICIP)*, pages 445–448, September 2012.
- [84] José García-Rodríguez and Juan Manuel García-Chamizo. Surveillance and human-computer interaction applications of self-growing models. *Applied Soft Computing*, 11(7):4413 – 4431, 2011. Soft Computing for Information System Security.

- [85] J Gemmell, R Lueder, and G Bell. The Mylifebits Lifetime Store. In *Transactions on Multimedia Computing, Communications and Applications*, pages 0–5, New York, New York, USA, 2002. ACM Press.
- [86] J Gemmell, L Williams, and K Wood. Passive Capture and Ensuing Issues for a Personal Lifetime Store. In *Workshop on Continuous Archival and Retrieval of Personal Experiences*, pages 48–55, New York, NY, 2004.
- [87] J Ghosh and K Grauman. Discovering Important People and Objects for Egocentric Video Summarization. In *Computer Vision and Pattern Recognition*, pages 1346–1353. IEEE, June 2012.
- [88] Iván González Díaz, Vincent Buso, Jenny Benois-Pineau, Guillaume Bourmaud, and Rémi Megret. Modeling Instrumental Activities of Daily Living in Egocentric Vision as Sequences of Active Objects and Context for Alzheimer Disease Research. In *International workshop on Multimedia indexing and information retrieval for healthcare*, pages 11–14, New York, New York, USA, 2013. ACM Press.
- [89] Gösta Granlund. Does vision inevitably have to be active? In *Proceedings of the 11th Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland, June 7–11 1999. SCIA. Also as Technical Report LiTH-ISY-R-2247.
- [90] R Grasset, T Langlotz, and D Kalkofen. Image-driven View Management for Augmented Reality Browsers. In *ISMAR*, pages 177–186. Ieee, November 2012.
- [91] D Guan, W Yuan, A Jehad-Sarkar, T Ma, and Y Lee. Review of Sensor-based Activity Recognition Systems. *IETE Technical Review*, 28(5):418, 2011.
- [92] Seungyeop Han, Rajalakshmi Nandakumar, Matthai Philipose, Arvind Krishnamurthy, and David Wetherall. GlimpseData: Towards Continuous Vision-based Personal Analytics. In *Workshop on physical analytics*, volume 40, pages 31–36, New York, New York, USA, 2014. ACM Press.
- [93] S. Haykin, M. Fatemi, P. Setoodeh, and Yanbo Xue. Cognitive control. *Proceedings of the IEEE*, 100(12):3156–3169, Dec.
- [94] S. Haykin and J.M. Fuster. On cognitive dynamic systems: Cognitive neuroscience and engineering learning from each other. *Proceedings of the IEEE*, 102(4):608–628, April 2014.

- [95] Simon Haykin. Cognitive dynamic systems. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV-1369-IV-1372, April 2007.
- [96] M Hebert and T Kanade. Discovering Object Instances from Scenes of Daily Living. In *International Conference on Computer Vision*, pages 762-769. Ieee, November 2011.
- [97] S Hodges, L Williams, E Berry, S Izadi, J Srinivasan, A Butler, G Smyth, N Kapur, and K Wood. Sensecam: a Retrospective Memory Aid. In *International Conference of Ubiquitous Computing*, pages 177-193. Springer Verlag, 2006.
- [98] T Hori and K Aizawa. Context-based Video Retrieval System for the Life-log Applications. In *International Workshop on Multimedia Information Retrieval*, page 31, New York, New York, USA, 2003. ACM Press.
- [99] Y Hoshen and S Peleg. Egocentric Video Biometrics. *arXiv preprint*, November 2015.
- [100] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415-425, Mar 2002.
- [101] De-An Huang, Wei-Chiu Ma, Minghuang Ma, and Kris M. Kitani. How do we use our hands? discovering a diverse set of common grasps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [102] H. Huang, J.J. Legarsky, S. Gudimetla, and C.H. Davis. Post-classification smoothing of digital classification map of st. louis, missouri. In *Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International*, volume 5, pages 3039-3041 vol.5, Sept 2004.
- [103] Tatsuya Ishihara, Kris Kitani, Wei-Chiu Ma, Hironobu Takagi, and Chieko Asakawa. Recognizing hand-object interactions in wearable camera videos. In *International Conference on Image Processing*, 2015.
- [104] Jinwei Jiang and A. Yilmaz. Good features to track: A view geometric approach. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 72-79, Nov.
- [105] J. Jockusch and H. Ritter. An instantaneous topological map for correlated stimuli. In *Proceedings of the international joint conference on neural Networks*, volume 1, pages 529-534, Washington, USA, 1999.

- [106] M Jones and J Rehg. Statistical Color Models with Application to Skin Detection 2 Histogram Color Models. In *Computer Vision and Pattern Recognition*, volume Jun, pages 1–23, Fort Collins, CO, 1999. IEEE Computer Society.
- [107] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A Survey of Skin-color Modeling and Detection Methods. *Pattern Recognition*, 40(3):1106–1122, March 2007.
- [108] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422, July 2012.
- [109] R.E. Kalman. A new approach to linear filtering and prediction problems. *T-ASME*, 1960:35–45, March 1960.
- [110] T Kanade and M Hebert. First-person Vision. *Proceedings of the IEEE*, 100(8):2442–2453, August 2012.
- [111] S Karaman, J Benois-Pineau, R Megret, V Dovgalecs, J Dartigues, and Y Gaestel. Human Daily Activities Indexing in Videos from Wearable Cameras for Monitoring of Patients with Dementia Diseases. In *International Conference on Pattern Recognition*, pages 4113–4116. Ieee, August 2010.
- [112] Thomas P. Karnowski, I Arel, and D. Rose. Deep spatiotemporal feature learning with application to image classification. In *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*, pages 883–888, Dec 2010.
- [113] ZW Kim and J Malik. Fast Vehicle Detection with Probabilistic Feature Grouping and Its Application to Vehicle Tracking. In *Computer Vision*, pages 524 – 531, Nice, France, 2003. IEEE.
- [114] K Kitani. Ego-Action Analysis for First-Person Sports Videos. *Pervasive Computing*, 11(2):92–95, 2012.
- [115] K Kitani and T Okabe. Fast Unsupervised Ego-action Learning for First-person Sports Videos. In *Computer Vision and Pattern Recognition*, pages 3241–3248, Providence, RI, June 2011. IEEE.
- [116] A. Knittel. Learning Feature Hierarchies under Reinforcement. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8, June 2012.
- [117] Y Kojima and Y Yasumuro. Hand Manipulation of Virtual Objects in Wearable Augmented Reality. In *Virtual Systems and Multimedia*, pages 463 – 469, 2001.

- [118] M Kölsch, R Bane, T Höllerer, and M Turk. Touching the Visualized Invisible: Wearable Ar with a Multimodal Interface. *IEEE Computer Graphics and Applications*, Jun(1):62–71, 2006.
- [119] M Kolsch and M Turk. Fast 2d Hand Tracking with Flocks of Features and Multi-cue Integration. In *Computer Vision and Pattern Recognition Workshop*, pages 158–158. IEEE Comput. Soc, 2004.
- [120] M Kölsch and M Turk. Robust Hand Detection. In *FGR*, 2004.
- [121] M Kolsch, M Turk, and T Hollerer. Vision-based Interfaces for Mobility. In *Mobile and Ubiquitous Systems: Networking and Services*, pages 86 – 94. Ieee, 2004.
- [122] Win Kong, A. Hussain, M.H.M. Saad, and N.M. Tahir. Hand detection from silhouette for video surveillance application. In *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on*, pages 514–518, March.
- [123] N Krahnstoever, J Rittscher, P Tu, K Chean, and T Tomlinson. Activity Recognition using Visual Tracking and RFID. In *IEEE Workshops on Applications of Computer Vision (WACV/MOTION’05)*, volume 1, pages 494–500, Breckenridge, CO, January 2005. IEEE.
- [124] T Kurata and T Okuma. The Hand Mouse: Gmm Hand-color Classification and Mean Shift Tracking. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 119 – 124, Vancouver, Canada, 2001. IEEE.
- [125] T Kurata, T Okuma, and M Kourogi. Vizwear: Toward Human-centered Interaction Through Wearable Vision and Visualization. *Lecture Notes in Computer Science*, 2195(1):40–47, 2001.
- [126] T Kurata, T Okuma, M Kourogi, and K Sakaue. The Hand-mouse: A Human Interface Suitable for Augmented Reality Environments Enabled by Visual Wearables. In *Symposium on Mixed Reality*, pages 188–189, Yokohama, 2000.
- [127] M Land and M Hayhoe. In What Ways Do Eye Movements Contribute to Everyday Activities? *Vision research*, 41(25-26):3559–65, January 2001.
- [128] S Lee, S Bambach, D Crandall, J Franchak, and C Yu. This Hand Is My Hand: A Probabilistic Approach to Hand Disambiguation in Egocentric Video. In *Computer Vision and Pattern Recognition*, pages 1–8, Columbus, Ohio, 2014. IEEE Computer Society.

- [129] Yong Jae Lee and Kristen Grauman. Predicting Important Objects for Egocentric Video Summarization. *International Journal of Computer Vision*, Jan(1):1–19, January 2015.
- [130] A. Levinstein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2290–2297, Dec 2009.
- [131] C Li and K Kitani. Model Recommendation with Virtual Probes for Egocentric Hand Detection. In *ICCV 2013*, Sydney, 2013. IEEE Computer Society.
- [132] C Li and K Kitani. Pixel-Level Hand Detection in Ego-centric Videos. In *Computer Vision and Pattern Recognition*, pages 3570–3577. Ieee, June 2013.
- [133] Hui Li, Lei Yang, Xiaoyu Wu, and Jun Zhai. Hands detection based on statistical learning. In *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on*, volume 2, pages 227–230, Oct.
- [134] Y Li, A Fathi, and J Rehg. Learning to Predict Gaze in Egocentric Video. In *International Conference on Computer Vision*, pages 1–8. Ieee, 2013.
- [135] KY Liu, SC Hsu, and CL Huang. First-person-vision-based Driver Assistance System. In *Audio, Language and Image Processing*, pages 4–9, 2014.
- [136] Yang Liu, Youngkyoon Jang, Woontack Woo, and Tae-Kyun Kim. Video-Based Object Recognition Using Novel Set-of-Sets Representations. In *Computer Vision and Pattern Recognition*, pages 533–540. Ieee, June 2014.
- [137] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.
- [138] D. Lu and Q. Weng. A Survey of Image Classification Methods and Techniques for Improving Classification Performance. *International Journal of Remote Sensing*, 28(5):823–870, March 2007.
- [139] Z Lu and K Grauman. Story-Driven Summarization for Egocentric Video. In *Computer Vision and Pattern Recognition*, pages 2714–2721, Portland, OR, USA, June 2013. IEEE.
- [140] A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua. Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. *Medical Imaging, IEEE Transactions on*, 31(2):474–486, Feb 2012.

- [141] Tamas Madl, Bernard J. Baars, and Stan Franklin. The timing of the cognitive cycle. *PLoS ONE*, 6(4):e14803, 04 2011.
- [142] K Makita, M Kanbara, and N Yokoya. View Management of Annotations for Wearable Augmented Reality. In *Multimedia and Expo*, pages 982–985. Ieee, June 2009.
- [143] M.Z. Malik and S. Khurshid. Dynamic shape analysis using spectral graph properties. In *Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on*, pages 211–220, April 2012.
- [144] S Mann. Wearable Computing: a First Step Toward Personal Imaging. *Computer*, 30(2):25–32, 1997.
- [145] S Mann. “WearCam” (Wearable Camera): Personal Imaging Systems for Long-term Use in Wearable Tetherless Computer-mediated Reality and Personal Photo/videographic Memory Prosthesis. In *Wearable Computers*, pages 124–131, Pittsburgh, 1998. IEEE Computer Society.
- [146] S Mann. Continuous Lifelong Capture of Personal Experience with Eyetap. In *Continuous Archival and Retrieval of Personal Experiences*, pages 1–21, New York, New York, USA, 2004. ACM Press.
- [147] S. Mann, M.A. Ali, R. Lo, and Han Wu. Freeglass for developers, "haccessibility", and digital eye glass + lifelogging research in a (sur/sous)veillance society. In *Information Society (i-Society), 2013 International Conference on*, pages 48–53, June 2013.
- [148] S Mann, J Nolan, and B Wellman. Sousveillance: Inventing and Using Wearable Computing Devices for Data Collection in Surveillance Environments. *Surveillance & Society*, 1(3):331–355, 2003.
- [149] Steve Mann, Jason Huang, Ryan Janzen, Raymond Lo, Valmiki Rampersad, Alexander Chen, and Taqveer Doha. Blind navigation with a wearable range camera and vibrotactile helmet. In *International Conference on Multimedia*, page 1325, New York, New York, USA, 2011. ACM Press.
- [150] L. Marcenaro, L. Marchesotti, and C. S. Regazzoni. Self-organizing shape description for tracking and classifying multiple interacting objects. *Image Vision Comput.*, 24(11):1179–1191, 2006.

- [151] L. Marcenaro, F. Oberti, and C. S. Regazzoni. Change detection methods for automatic scene analysis by using mobile surveillance cameras. In *ICIP*, pages 1244–1247, 2000.
- [152] Francis Martinez, Andrea Carbone, and Edwige Pissaloux. Combining First-person and Third-person Gaze for Attention Recognition. *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–6, April 2013.
- [153] K Matsuo, K Yamada, S Ueno, and S Naito. An Attention-Based Activity Recognition for Egocentric Video. In *Computer Vision and Pattern Recognition*, pages 565–570. Ieee, June 2014.
- [154] W Mayol, A Davison, B Tordoff, and D Murray. Applying Active Vision and SLAM to Wearables. In Paolo Dario and Raja Chatila, editors, *Springer Tracts in Advanced Robotics*, volume 15 of *Springer Tracts in Advanced Robotics*, pages 325–334. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [155] W Mayol and D Murray. Wearable Hand Activity Recognition for Event Summarization. In *International Symposium on Wearable Computers*, pages 1–8. IEEE, 2005.
- [156] W Mayol, B Tordoff, and D Murray. Wearable Visual Robots. In *International Symposium on Wearable Computers*, pages 95–102, Atlanta, 2000. IEEE Computer Society.
- [157] Xingbao Meng, Jing Lin, and Yingchun Ding. An extended hog model: Schog for human hand detection. In *Systems and Informatics (ICSAI), 2012 International Conference on*, pages 2593–2596, May.
- [158] Wu Min, Xiao Li, Cheston Tan, Bappaditya Mandal, Liyuan Li, and Joo Hwee Lim. Efficient Retrieval from Large-Scale Egocentric Visual Data Using a Sparse Graph Representation. In *Computer Vision and Pattern Recognition Workshops*, pages 541–548. Ieee, June 2014.
- [159] A.P. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.
- [160] P Morerio, L Marcenaro, and C Regazzoni. Hand Detection in First Person Vision. In *Information Fusion*, pages 1502 – 1507, Istanbul, 2013. University of Genoa.

- [161] Pietro Morerio, Gabriel Claudiu Georgiu, Lucio Marcenaro, and Carlo Regazzoni. Optimizing superpixel clustering for real-time egocentric-vision applications. *IEEE Signal Processing Letters*, 2014.
- [162] Pietro Morerio, Lucio Marcenaro, and Carlo S Regazzoni. A generative superpixel method. In *17th IEEE International Conference on Information Fusion (FUSION 2014)*, 2014.
- [163] Pietro Morerio, Lucio Marcenaro, Carlo S. Regazzoni, and Gianluca Gera. Early fire and smoke detection based on colour features and motion analysis. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1041–1044, 30 2012-Oct. 3.
- [164] M Morshidi and T Tjahjadi. Gravity Optimised Particle Filter for Hand Tracking. *Pattern Recognition*, 47(1):194–207, 2014.
- [165] Srinivas Mukkamala, Guadalupe Janoski, and Andrew Sung. Intrusion detection using neural networks and support vector machines. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 2, pages 1702–1707. IEEE, 2002.
- [166] K Murphy, A Torralba, D Eaton, and W Freeman. Object Detection and Localization Using Local and Global Features. *Toward Category-Level Object Recognition*, 4170:382–400, 2006.
- [167] S Narayan, M Kankanhalli, and K Ramakrishnan. Action and Interaction Recognition in First-Person Videos. In *Computer Vision and Pattern Recognition*, pages 526–532. Ieee, June 2014.
- [168] F. Navarro, M. Escudero-Viñolo, and J. Bescós. Sp-sift: enhancing sift discrimination via super-pixel-based foreground-background segregation. *Electronics Letters*, 50(4):272–274, February 2014.
- [169] Haung Wei Ng, Y Sawahata, and K Aizawa. Summarization of Wearable Videos Using Support Vector Machine. In *IEEE International Conference on Multimedia and Expo*, volume Aug, pages 325–328. IEEE, 2002.
- [170] D Nguyen, G Marcu, G Hayes, K Truong, J Scott, M Langheinrich, and C Roduner. Encountering Sensecam: Personal Recording Technologies in Everyday Life. In *International Conference on Ubiquitous Computing*, 2009.

- [171] K Ogaki, K Kitani, Y Sugano, and Y Sato. Coupling Eye-motion and Ego-motion Features for First-person Activity Recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–7. Ieee, June 2012.
- [172] M Okamoto and K Yanai. Summarization of Egocentric Moving Videos for Generating Walking Route Guidance. In *Image and Video Technology*, pages 431–442, 2014.
- [173] A Oliva and A Torralba. Modeling the Shape of the Scene: a Holistic Representation of the Spatial Envelope. *International journal of computer vision*, 42(3):145–175, 2001.
- [174] Y Pai, L Lee, and S Ruan. Honeycomb Model Based Skin Color Detector for Face Detection. In *Mechatronics and Machine Vision in Practice*, pages 2–4, Auckland, 2010. IEEE Computer Society.
- [175] HS Park, Eakta Jain, and Yaser Sheikh. 3d Social Saliency from Head-mounted Cameras. *Advances in Neural Information Processing Systems*, pages 431–439, 2012.
- [176] D Patterson, D Fox, H Kautz, and M Philipose. Fine-Grained Activity Recognition by Aggregating Abstract Object Usage. In *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 44–51. IEEE, 2005.
- [177] Vladimir Pavlovic, JM Rehg, and J MacCormick. Learning switching linear models of human motion. *NIPS*, 2000.
- [178] F. Pedregosa, G. Varoquaux, A Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, V. and Weiss, R. and Dubourg, J. Vanderplas, A Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Research, Journal of Machine Learning*, 12:2825–2830, 2011.
- [179] Zongju Peng, Gangyi Jiang, Mei Yu, Shihua Pi, and Fen Chen. Temporal pixel classification and smoothing for higher depth video compression performance. *Consumer Electronics, IEEE Transactions on*, 57(4):1815–1822, November 2011.
- [180] M Philipose. Egocentric Recognition of Handled Objects: Benchmark and Analysis. In *Computer Vision and Pattern Recognition*, pages 1–8, Miami, FL, June 2009. IEEE.

- [181] M Philipose and K Fishkin. Inferring Activities from Interactions with Objects. *Pervasive Computing*, 3(4):50–57, 2004.
- [182] H Pirsiavash and D Ramanan. Detecting Activities of Daily Living in First-Person Camera Views. In *Computer Vision and Pattern Recognition*, pages 2847–2854. IEEE, June 2012.
- [183] Y Poleg, C Arora, and S Peleg. Head Motion Signatures from Egocentric Videos. In *Asian Conference on Computer Vision*, volume Nov, pages 1–15, Singapore, 2014. Springer.
- [184] Y Poleg, C Arora, and S Peleg. Temporal Segmentation of Egocentric Videos. In *Computer Vision and Pattern Recognition*, pages 2537–2544. Ieee, June 2014.
- [185] C.S. Regazzoni and A. Teschioni. A new approach to vector median filtering based on space filling curves. *Image Processing, IEEE Transactions on*, 6(7):1025–1037, Jul 1997.
- [186] J Rehg and T Kanade. DigitEyes: Vision-Based Hand Tracking for Human-Computer Interaction. In *Workshop on Motion of Non-Rigid and Articulated Bodies*, pages 16–22. IEEE Comput. Soc, 1994.
- [187] C Ren and I Reid. gSLIC: A Real-time Implementation of SLIC Superpixel Segmentation. Technical report, University of Oxford, Department of Engineering Science, Oxford, UK, 2011.
- [188] X Ren and C Gu. Figure-ground Segmentation Improves Handled Object Recognition in Egocentric Video. In *Conference on Computer Vision and Pattern Recognition*, pages 3137–3144. IEEE, June 2010.
- [189] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Radar Library. Artech House, 2004.
- [190] Giuseppe Riva, Francesco Vatalaro, Fabrizio Davide, and M. Alcañiz, editors. *Ambient Intelligence – The Evolution of Technology, Communication and Cognition Towards the Future of Human-Computer Interaction*, volume 6. IEEE, January 2005.
- [191] Gregory Rogez and JS Supancic III. 3D Hand Pose Detection in Egocentric RGB-D Images. In *ECCV Workshop on Consumer Depth Camera for Computer Vision*, volume Sep, pages 1–14, Zurich, Switzerland, November 2014. Springer.

- [192] Gregory Rogez, James S. Supancic, and Deva Ramanan. Egocentric Pose Recognition in Four Lines of Code. In *Computer Vision and Pattern Recognition*, volume Jun, pages 1–9, November 2015.
- [193] M Ryoo and L Matthies. First-Person Activity Recognition: What Are They Doing to Me? In *Conference on Computer Vision and Pattern Recognition*, pages 2730–2737, Portland, OR, US, 2013. IEEE Comput. Soc.
- [194] Y Sawahata and K Aizawa. Wearable Imaging System for Summarizing Personal Experiences. *Multimedia and Expo*, Jul(1):1–45, 2003.
- [195] A Scheck. Seeing the (Google) Glass as Half Full. *EMN*, pages 20–21, 2014.
- [196] B Schiele, T Starner, and B Rhodes. Situation Aware Computing with Wearable Computers. In *Augmented Reality and Wearable Computers*, pages 1–20, 1999.
- [197] Bernt Schiele, Nuria Oliver, Tony Jebara, and Alex Pentland. An Interactive Computer Vision System DyPERS: Dynamic Personal Enhanced Reality System. In *Computer Vision Systems*, volume 1542 of *Lecture Notes in Computer Science*, pages 51–65. Springer Berlin Heidelberg, Berlin, Heidelberg, September 1999.
- [198] M Schlattman and R Klein. Simultaneous 4 Gestures 6 Dof Real-time Two-hand Tracking Without Any Markers. In *Symposium on Virtual Reality Software and Technology*, pages 39–42, New York, NY, USA, 2007. ACM Press.
- [199] M Schlattmann, F Kahlesz, R Sarlette, and R Klein. Markerless 4 Gestures 6 Dof Real-time Visual Tracking of the Human Hand with Automatic Initialization. *Computer Graphics Forum*, 26(3):467–476, September 2007.
- [200] G Serra, M Camurri, and L Baraldi. Hand Segmentation for Gesture Recognition in Ego-vision. In *Workshop on Interactive Multimedia on Mobile & Portable Devices*, pages 31–36, New York, NY, USA, 2013. ACM Press.
- [201] C Shan, T Tan, and Y Wei. Real-time Hand Tracking Using a Mean Shift Embedded Particle Filter. *Pattern Recognition*, 40(7):1958–1970, July 2007.
- [202] A Shashua, Y. Gdalyahu, and G. Hayun. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 1–6, June 2004.
- [203] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 731–737, Jun 1997.

- [204] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, Jun.
- [205] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.
- [206] P. Siva and A Wong. Grid seams: A fast superpixel algorithm for real-time applications. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 127–134, May 2014.
- [207] IEEE Computer Society. News briefs. *Computer*, 45(7):21–23, 2012.
- [208] M Spain and P Perona. Measuring and Predicting Object Importance. *International Journal of Computer Vision*, 91(1):59–76, August 2010.
- [209] M. Spirito, C. S. Regazzoni, and L. Marcenaro. Automatic detection of dangerous events for underground surveillance. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2005)*, pages 195–200, Como, Italy, September 2005. IEEE Computer Society.
- [210] E Spriggs, F De La Torre, and M Hebert. Temporal Segmentation and Activity Classification from First-person Sensing. In *Computer Vision and Pattern Recognition Workshops*, pages 17–24. IEEE, June 2009.
- [211] V Spruyt, A Ledda, and W Philips. Real-time, Long-term Hand Tracking with Unsupervised Initialization. In *International Conference on Image Processing*, Melbourne, Australia, 2013. IEEE Comput. Soc.
- [212] T Starner. *Wearable Computing and Contextual Awareness*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [213] T Starner. Project Glass: An Extension of the Self. *Pervasive Computing*, 12(2):125, 2013.
- [214] T Starner, B Schiele, and A Pentland. Visual Contextual Awareness in Wearable Computing. In *International Symposium on Wearable Computers*, pages 50–57. IEEE Computer Society, 1998.
- [215] T Starner, J Weaver, and A Pentland. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

- [216] E. Stergiopoulou and N. Papamarkos. Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence*, 22(8):1141 – 1158, 2009.
- [217] L Sun, U Klank, and M Beetz. Eyewatchme—3d Hand and Object Tracking for Inside out Activity Analysis. In *Computer Vision and Pattern Recognition*, pages 9–16, 2009.
- [218] S Sundaram and W Cuevas. High Level Activity Recognition Using Low Resolution Wearable Vision. *Computer Vision and Pattern Recognition Workshops*, pages 25–32, June 2009.
- [219] S.N. Tamgade and V.R. Bora. Motion vector estimation of video image by pyramidal implementation of lucas kanade optical flow. In *Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on*, pages 914–917, Dec.
- [220] Cheston Tan, Hanlin Goh, and Vijay Chandrasekhar. Understanding the Nature of First-Person Videos: Characterization and Classification using Low-Level Features. In *Computer Vision and Pattern Recognition*, pages 535–542, 2014.
- [221] D Tancharoen, T Yamasaki, and K Aizawa. Practical experience recording and indexing of Life Log video. In *Workshop on Continuous Archival and Retrieval of Personal Experiences*, page 61, New York, New York, USA, 2005. ACM Press.
- [222] Robert Templeman, Mohammed Korayem, D.J. Crandall, and Kadapia Apu. PlaceAvoider: Steering first-person cameras away from sensitive spaces. In *Network and Distributed System Security Symposium*, pages 23–26, February 2014.
- [223] R Tenmoku, M Kanbara, and N Yokoya. Annotating User-viewed Objects for Wearable Ar Systems. In *International Symposium on Mixed and Augmented Reality*, pages 192–193. Ieee, 2005.
- [224] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *ECCV (4)*, volume 5305 of *Lecture Notes in Computer Science*, pages 705–718. Springer, 2008.
- [225] David Vernon. Cognitive vision: The case for embodied perception. *Image and Vision Computing*, 26(1):127 – 140, 2008. Cognitive Vision-Special Issue.

- [226] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(6):583–598, Jun 1991.
- [227] P. Viola and M. Jones. Robust real-time face detection. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 747–747, 2001.
- [228] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2001*, december 2001.
- [229] Paul Viola, Michael J. Jones, Daniel Snow, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. In *In ICCV*, pages 734–741, 2003.
- [230] S Walk, N Majer, K Schindler, and B Schiele. New Features and Insights for Pedestrian Detection. In *Conference on Computer Vision and Pattern Recognition*, pages 1030–1037. Ieee, June 2010.
- [231] H Wang and X Bao. Insight: Recognizing Humans Without Face Recognition. In *Workshop on Mobile Computing Systems and Applications*, pages 2–7, New York, NY, USA, 2013.
- [232] Jingtao Wang and Chunxuan Yu. Finger-fist Detection in First-person View Based on Monocular Vision Using Haar-like Features. In *Chinese Control Conference*, pages 4920–4923. Ieee, July 2014.
- [233] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, pages 1323–1330, Washington, DC, USA, 2011. IEEE Computer Society.
- [234] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A Luthra. Overview of the h.264/avc video coding standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):560–576, July 2003.
- [235] J Wu and A Osuntogun. A Scalable Approach to Activity Recognition Based on Object Use. In *Internantional Conference on Computer Vision*, pages 1–8, Rio de Janeiro, 2007. IEEE.
- [236] Shipeng Xie and Jing Pan. Hand detection using robust color correction and gaussian mixture model. In *Image and Graphics (ICIG), 2011 Sixth International Conference on*, pages 553–557, Aug.

- [237] Yulin Xie, Huchuan Lu, and Ming-Hsuan Yang. Bayesian saliency via low and mid level cues. *Image Processing, IEEE Transactions on*, 22(5):1689–1698, May 2013.
- [238] Bo Xiong and Kristen Grauman. Detecting Snap Points in Egocentric Video with a Web Photo Prior. In *Internantional Conference on Computer Vision*, 2014.
- [239] Chenliang Xu and J.J. Corso. Evaluation of super-voxel methods for early video processing. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1202–1209, June 2012.
- [240] K Yamada, Y Sugano, and T Okabe. Can Saliency Map Models Predict Human Egocentric Visual Attention? In *Internantional Conference on Computer Vision*, pages 1–10, 2011.
- [241] K Yamada, Y Sugano, T Okabe, Y Sato, A Sugimoto, and K Hiraki. Attention Prediction in Egocentric Video Using Motion and Visual Saliency. In *Pacific Rim Conference on Advances in Image and Video Technology*, pages 277–288, 2012.
- [242] G Yang, H Li, L Zhang, and Y Cao. Research on a Skin Color Detection Algorithm Based on Self-adaptive Skin Color Model. In *International Conference on Communications and Intelligence Information Security*, pages 266–270. Ieee, October 2010.
- [243] A Yarbus. *Eye Movements and Vision*. Plenum Press, New York, New York, USA, 1967.
- [244] W Yi and D Ballard. Recognizing Behavior in Hand-eye Coordination Patterns. *International Journal of Humanoid Robotics*, 6(3):337–359, 2009.
- [245] C Yu and Dana Ballard. Learning To Recognize Human Action Sequences. In *Development and Learning*, pages 28–33, 2002.
- [246] J Zariffa and MR Popovic. Hand Contour Detection in Wearable Camera Video Using an Adaptive Histogram Region of Interest. *Journal of NeuroEngineering and Rehabilitation*, 10(114):1–10, 2013.
- [247] Kai Zhan, Steven Faux, and Fabio Ramos. Multi-scale Conditional Random Fields for first-person activity recognition. In *International Conference on Pervasive Computing and Communications*, pages 51–59. Ieee, March 2014.

- [248] Jingzhe Zhang, Lishuo Zhuang, Yang Wang, Yameng Zhou, Yan Meng, and Gang Hua. An Egocentric Vision Based Assistive co-robot. *Conference on Rehabilitation Robotics*, 2013(Jun):1–7, June 2013.
- [249] Pifu Zhang, E.E. Milios, and J. Gu. Graph-based automatic consistent image mosaicking. In *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*, pages 558–563, Aug 2004.
- [250] K Zheng, Y Lin, Y Zhou, D Salvi, X Fan, D Guo, Zibo Meng, and Song Wang. Video-based Action Detection using Multiple Wearable Cameras. In *Workshop on ChaLearn Looking at People*, 2014.
- [251] Xiaolong Zhu, Xuhui Jia, and Kwan-yeek K Wong. Pixel-Level Hand Detection with Shape-aware Structured Forests. In *Asian Conference on Computer Vision*, pages 1–15, Singapore, 2014.
- [252] Y Zhu and S Schwartz. Efficient Face Detection with Multiscale Sequential Classification. In *Image Processing*, pages 121–124. IEEE, 2002.
- [253] C. Lawrence Zitnick and Sing Bing Kang. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75(1):49–65, 2007.