UNIVERSITÀ DEGLI STUDI DI CATANIA
Dipartimento di Matematica e Informatica
Dottorato di Ricerca in Matematica e Informatica XXIX Ciclo

Antonino Furnari

Context Awareness in First Person Vision

Tesi di Dottorato di Ricerca

Tutor: Prof. Sebastiano Battiato

Anno Accademico 2015 - 2016

"I consider it a challenge before the whole human race, and I ain't gonna loose."

F. Bulsara

Abstract

The First Person Vision (FPV) paradigm allows to seamlessly acquire images of the world from the user's perspective. Compared to standard Third Person Vision, FPV is advantageous for building intelligent wearable systems able to assist the user and augment his abilities. Given their intrinsic mobility and the ability to acquire user-related information, FPV systems have to deal with a continuously evolving environment. Moving from the observation that data acquired from a first person perspective is highly personal, we investigate contextual awareness for First Person Vision systems. We first focus on the task of recognizing personal locations of interest from egocentric videos. We consider personal locations at the instance level and address the problem of rejecting locations not of interest for the user. To challenge the problem, we introduce three datasets of 10 personal locations which we make publicly available, and perform a benchmark of different wearable devices and state-of-the-art representations. Moreover, we propose and evaluate methods to reject negative locations and perform personal location-based temporal segmentation of egocentric videos. As a second aspect, we investigate the anticipation of object interaction. We propose and define the task of next-active-object prediction as recognizing which objects are going to be interacted with, before the actual interaction begins. Even if recognizing next-active-objects is in general not trivial in unconstrained settings, we show that the First Person Vision paradigm provides useful cues to address the challenge. We propose a next-active-object prediction method based on the analysis of egocentric object trajectories and assess its superior performances with respect to other cues such as object appearance, distance from the center of the frame, presence of hands and visual saliency. In appendix, we also report some investigations on extraction features directly from wide angle images.

Acknowledgements

First and foremost, I would like to thank my advisors Prof. Sebastiano Battiato and Dr. Giovanni Maria Farinella. Without their guide and enthusiasm, I would have not been able to accomplish any of the achievements of these three years. They always pushed me towards excellence in a sincere and selfless way, giving me many opportunities to learn and grow professionally. I would like to give special thanks to all faculty and staff involved in IPLAB, including Prof. Giovanni Gallo and Prof. Filippo Stanco, for giving birth to and continuously supporting the IPLAB research group.

I would like to express my sincere gratitude to Prof. Kristen Grauman for giving me the opportunity to visit her lab at the University of Texas at Austin. Under her supervision, I spent four memorable and productive months in which I learned many things about my research and myself. I would like to thank all members of Prof. Grauman's group, Suyog, Dinesh, Aron, Bo, Yu-Chuan, Ruohan, Wei-Lin and Danna for the fruitful discussions and reading group meetings, and for making me feel part of the group.

I also wish to thank my two external reviewers, Dr. Dima Damen and Prof. Carlo Regazzoni, for helping me improve this thesis with their useful feedback.

This three-years journey would have not been equally engaging without my colleagues (past and present) at IPLAB: Marco, Valerio, Davide, Matteo, Oliver, Filippo, Dario, Nino, Daniele, Emiliano, Alessandro, Vito, Giusy, Tiziana, Ram. With them I shared many professional and life-experiences including doctoral summer schools and reading groups. I would like to thank them all for letting me feel part of a big family since the first day.

Thanks to my parents and my sister Paola for everything they gave me in all these years, even when I was too proud to ask. Their deep and sincere love helped me through the hard times and defined the good ones. Way before I started my PhD, they taught me to always pursue the truth and to be honest with the others.

Thanks to Nicoletta. She always knew what was best for me before I did, and she always pushed me towards improvement, realization and happiness. She is more than I ever asked for, and I couldn't ask for better.

Contents

Chapter 1

Introduction

1.1 Motivation

Visual perception is the primary means by which humans sense and understand the world they live in. Thus, it is not surprising how the research community has invested considerable efforts in trying to understand and replicate the amazing abilities of our visual system. Since vision is a form of intelligence itself, we should expect that building the intelligent systems which are likely to characterize our future will require the development of advanced forms of artificial visual intelligence. While the sensing technologies needed to obtain a suitable visual representation of the world are already available and ready to use, much work is still to be done to enable the design of truly intelligent systems capable of making a real difference in our lives.

1.1.1 Prevalence of the Third Person Vision Paradigm and its Limits

In the past decades, Computer Vision has had a tremendous impact in many scenarios which made its application feasible, albeit constrained under given circumstances. Some success examples include (but are not limited to) face detection [1], visual object tracking [2, 3, 4, 5], 2D image stitching [6, 7], 3D reconstruction [8] and content-based image retrieval [9]. Most of these results assumed the "Third Person Vision" (TPV) paradigm, according to which the scene is acquired by a static camera which remains neutral to the observed events. Even in the case of automotive and autonomous vehicles, which are obviously moving, the camera is relatively

stable and the layout of the acquired scene is generally constrained. One of the motivations behind the TPV paradigm is to remove some nuances such as fast camera movements and blur from the acquired images. This makes perfect sense since our visual system is able to remove the very same nuances in a transparent way, making our cognition system somewhat coherent with the TPV paradigm. While simplifying the visual perception paradigm and enabling powerful applications, however, the TPV design runs the risk of limiting the visual intelligence of the developed systems, making them mere observers of the world, while human beings are not.

Indeed, while static cameras can sense the world from a limited number of perspectives, humans are able to look around and select their favorite view of the scene; while static cameras are bound to a limited number of physical locations, humans can explore the world and acquire an incredible variety of visual stimuli; while static cameras can make few assumptions on the observed actors, humans sense the world from their unique perspective and can make strong assumptions on the observed scene. In practice, while the TPV design is clearly appropriate in many cases in which a specific task needs to be accomplished (e.g., in the surveillance domain), it might not be adequate when the system is designed in support to an active agent interacting with the environment. This is, for instance, the case of wearable intelligent systems designed to assist the user and augment his abilities [10, 11]. Other examples include autonomous robots which, apart from being able to observe and understand the environment, are supposed to move and interact with it. In the aforementioned scenarios, in fact, a "First Person Vision" (FPV) paradigm is more convenient [10].

1.1.2 Advantages of the First Person Vision Paradigm

While the TPV paradigm assumes that images are acquired by a fixed camera placed in some convenient location with respect to the considered task (e.g., attached to the ceiling for indoor surveillance), according to the FPV paradigm, images are seamlessly acquired by means of a wearable camera which is carried by the user at all times. Figure 1.1 shows an example of fixed camera used in the TPV scenario and an example of wearable camera used in the FPV scenario. One of the main differences between a wearable camera and a standard fixed one is the intrinsic mobility of the former. On one hand, this introduces new challenges due to the lack

(a) fixed camera (TPV) (b) wearable camera (FPV)

Figure 1.1: An example of (a) fixed camera used in a standard Third Person Vision scenario and (b) wearable camera used in a First Person Vision scenario.[1]

of stability which can entail artifacts such as motion blur. On the other hand, the content acquired from FPV cameras always "tells something" about the user, the location in which he is operating, the activities he is involved in and, ultimately his true intent and goals [10]. An example of the advantages of First Person Vision systems is shown in Figure 1.2, which compares images of the same human activity acquired according to the two discussed paradigms. The clear advantage of FPV systems lies in the ability to be carried by the user and observe the world from his perspective. Moreover, the continuous nature of the acquired information enables easy acquisition of huge quantities of data, which can be useful for both off-line and on-line learning.

1.1.3 Context Awareness in First Person Vision

As the user moves and interacts with the scene, many factors related to the surrounding environment are deemed to change. These include the location in which the user operates, the performed activities, objects and people present on the scene, the time in which activities take place and the goals the user is trying to achieve. The ensemble of all such factors is broadly referred to as "context" in the literature [12, 13]. As pointed out by Dey and Abowd [14], context is important to improve human-machine interaction. Indeed, humans use context as an implicit

[1]Image (a) by M.O. Stevens, licensed under Creative Commons and acquired from this URL. Image (b) by A. Zugaldia, licensed under Creative Commons and acquired from this URL.

(a) Third Person Vision (TPV) image (b)First Person Vision (FPV) image

Figure 1.2: Examples of (a) Third Person Vision (TPV) and (b) First Person Vision (FPV) images. The two images are synchronized frames acquired by different cameras recording the same human activity. The TPV perspective is neutral to the observed events, while the FPV one allows to capture information about what the user is doing.[2]

means of additional information to effectively communicate with each other and react appropriately. Given their intrinsic mobility, FPV systems have to deal with a continuously changing environment [11]. Therefore, they need to be able to sense and correctly understand context in order to adapt their behavior to the different situations in which the user may be involved and, ultimately, to improve their intelligence.

As discussed in Section 1.1, due to their intrinsic mobility, First Person Vision systems have to deal with a continuously changing context. Even if it is difficult to formalize the concept of context, different authors have attempted to formulate suitable working definitions [12, 13, 14]. Among such efforts, Dey and Abowd [14] debated that context-aware systems look at the "who", "where", "when", "what" of entities and use this information to determine "why" the situation is occurring. The authors hence introduce four context categories which are deemed to be more important than others: "location", "identity", "activity" and "time". We complement the list by adding the "intent" category, which we find of crucial importance for the development of intelligent systems. The complete list of fundamental context categories is as follows:

[2]Frames from the Carnegie Mellon University Multimodal Aativity (CMU-MMAC) dataset [15].

- **Location**: provides information about "where" the current situation is happening. This kind of contextual information is useful to design systems able to adapt their behavior on the basis of the sensed location (see Figure 1.3(a));
- **Identity**: provides information about "who" is present on the scene. This kind of information allows to build "socially intelligent" systems which can adapt their behavior and address their communication on the basis of the social context [16] (see Figure 1.3(b));
- **Activity**: provides information about "what" is happening. Such information is essential to gain knowledge of what the user is doing, for instance to monitor his behavior or assist him (see Figure 1.3(c));
- **Time**: provides information about "when" the current situation is taking place. Time can be trivially exploited to correct contextual prediction (e.g., it is unlikely to be at the office at 4 A.M.) and issue time-triggered reminders or alerts;
- **Intent**: provides information about "why" the user is performing the current activity and relates to "what" he wants to achieve in the long run. Being able to understand the future intentions of the user or anticipate object interactions is important for human-machine interaction and to provide tailored assistance (see Figure 1.3(d)).

1.2 Aims and Approach

The aim of this thesis is to investigate context awareness in First Person Vision. Specifically, we concentrate on two of the five aforementioned context categories, which are *location* and *intent*. We move from the assumption that information acquired by First Person Vision systems is very personal for the user. Hence, we study how such information can be leveraged to model the personal environment in which the user operates and to predict his short term goals by anticipating future object interactions.

(a) location

(b) identity

(c) activity

(d) intent

Figure 1.3: Some visual examples of four context categories. (a) First Person Vision systems acquire visual content related to different locations. (b) First Person Vision systems should be aware of the social context (i.e., people present in the scene). (c) First Person Vision systems should be able to recognize the action performed by the user. (d) First Person Vision systems should be able to understand user's intent and predict what the user is going to do next. Image (b) is part of the dataset proposed in [17]. Image (c) is part of the dataset proposed in [18]. Image (d) is part of the dataset proposed in [19].

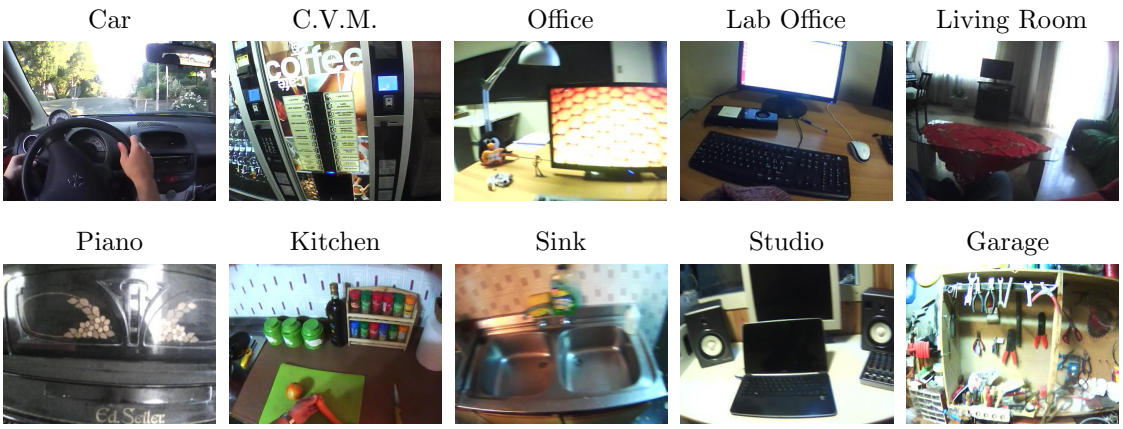


Figure 1.4: Some sample frames from the proposed dataset. C.V.M. stands for coffee vending machine.

We consider locations at the instance level (e.g., my office) rather than at the category level (e.g., an office) and investigate methods to recognize personal locations specified by the user from first person videos. We assume a supervised scenario in which the user indicates the locations he wants to monitor by providing minimal training data. To account for the huge variability in terms of visual content that FPV systems can acquire, we design methods to perform the rejection of negative locations (i.e., locations not of interest for the user). We benchmark different image representation techniques and provide methods to perform temporal segmentation of personal locations from videos. To support our analysis, we collected three datasets of first person videos acquired by a user while performing his daily activities in different locations: car, coffee vending machine, office, lab office, living room, piano, kitchen, sink, studio and garage. Figure 1.4 shows some visual examples of the considered locations. We made the datasets publicly available to foster future research on the topic.

We further explore the contextual insights given by the First Person Vision paradigm addressing the task of anticipating object interactions by predicting "next-active-objects", i.e., objects which are going to be manipulated by the user in a short time. In particular, we analyze the role of egocentric object trajectories in the proposed task of next-active-object prediction and compare them to other cues which might be available on the scene. Figure 1.5 shows some examples of next-active-objects, as compared to passive ones.

Figure 1.5: A sequence illustrating next-active-objects (in red) and passive ones (in cyan) along with their egocentric trajectories.

We also report complementary investigations on methods to perform direct feature extraction from images acquired using wide-angular sensors. Wide-angular sensors provide a representation of the scene which allows to increase the Field Of View and include more information about the environment than regular sensors. Therefore, they are usually employed in the design of wearable cameras.[3] Since these topics are not directly related to the main aim of this thesis, they are reported in appendix.

1.3 Contributions

The main contributions of this thesis are the following:

- The definition of the task of recognizing personal locations from first person videos;
- The introduction of three labeled datasets of first person videos acquired by a user in 10 different locations of interest;
- A benchmark of different state-of-the-art methods for scene and object classification on the proposed task of personal location recognition;
- The formulation and investigation of methods to perform the rejection of negative locations to extend multi-class classification to work in real scenarios;
- A system for the temporal segmentation of first person videos to highlight personal locations of interest;

[3]Some examples include GoPro (http://gopro.com), Authographer (http://www.autographer.com/) and Narrative Clip 2 (http://getnarrative.com/).

- The formulation of new the task of predicting next-active-objects from first person videos and the investigation of the role of object trajectories in the proposed task;

Other contributions include:

- A study of the applicability of affine covariant region detectors directly on wind angle images;
- The derivation of a novel family of generalized Sobel filters for the direct estimation of the gradient of wide angle images;
- The definition of Distortion Adaptive Descriptors, a new paradigm for the computation of gradient-based descriptors directly on wide angle images.

The contribution of this thesis have been published in international journals and conferences:
International Journals:

- A. Furnari, G. M. Farinella, and S. Battiato. "Recognizing Personal Locations From Egocentric Videos". In: *IEEE Transactions on Human-Machine Systems* 47.1 (2017), pp. 6–18. DOI: 10.1109/THMS.2016.2612002
- A. Furnari, G. M. Farinella, R. Bruna, and S. Battiato. "Affine Covariant Features for Fisheye Distortion Local Modeling". In: *IEEE Transactions on Image Processing* 26.2 (2017), pp. 696–710. DOI: 10.1109/TIP.2016.2627816
- A. Furnari, G. M. Farinella, R. Bruna, and S. Battiato. "Distortion Adaptive Sobel Filters for the Gradient Estimation of Wide Angle Images". In: *under review in Journal of Visual Communication and Image Representation* (2017)

International Conferences:

- A. Furnari, G. M. Farinella, and S. Battiato. "Temporal segmentation of egocentric videos to highlight personal locations of interest". In: *International Workshop on Egocentric Perception, Interaction and Computing (EPIC) in conjunction with ECCV*. 2016, pp. 474–489

- A. Furnari, G. M. Farinella, and S. Battiato. "Recognizing Personal Contexts from Egocentric Images". In: *Workshop on Assistive Computer Vision and Robotics (ACVR) in conjunction with ICCV*. 2015

- A. Furnari, G. M. Farinella, A. R. Bruna, and S. Battiato. "Distortion Adaptive Descriptors: Extending Gradient-Based Descriptors to Wide Angle Images". In: *Image Analysis and Processing (ICIAP)*. vol. 9280. Lecture Notes in Computer Science. Springer, 2015, pp. 205–215

- A. Furnari, G. M. Farinella, A. R. Bruna, and S. Battiato. "Generalized Sobel filters for gradient estimation of distorted images". In: *IEEE International Conference on Image Processing*. 2015, pp. 3250–3254

- A. Furnari, G. M. Farinella, G. Puglisi, A. R. Bruna, and S. Battiato. "Affine region detectors on the fisheye domain". In: *2014 IEEE International Conference on Image Processing (ICIP)*. 2014, pp. 5681–5685

Appendix B reports a list of other works not directly related to this thesis published during my Ph.D.

1.4 Overview of First Person Vision

First Person Vision has been investigated since the 90s. Of particular importance is the work by Steve Mann, who designed and developed many wearable computers equipped with visual processing capabilities [28, 29]. The main applications proposed by Mann in the mid 90s were targeted towards improving visual perception, augmenting memory and assisting the visually impaired [28, 30]. In the early years of FPV, other researchers investigated topics related to context awareness [11, 31], interaction [32] and augmented reality [33].

The appearance on the market of commercial wearable cameras featuring small dimensions and long battery life[4] has subsequently renewed the interest of the research community [34]. Moreover, the potential of modern computer vision technologies has promoted the development of important topics such as activity recognition [18, 19, 35, 36], video indexing and summarization [37, 38, 39, 40, 41] and visual attention modeling [42, 43, 44, 45].

1.4.1 Terminology

Over the last 20 years, First Person Vision has gone under different names. The first works on the topic referred to First Person Vision with the term "wearable vision" to underline the different design of such systems [11, 31, 28, 37, 46]. Unlike traditional TPV cameras, wearable systems are worn by the user and hence are able to acquire images from his viewpoint. Some researchers investigated the benefits of applying active vision to wearable systems and used the term "wearable active vision" [46, 47]. More recently, the term "egocentric vision" has been consistently used to put the emphasis on the personal nature (i.e., "related to me") of the acquired data [18, 38, 40, 45, 48, 49]. Similarly, other authors have adopted the term "First Person Vision" to highlight the different acquisition paradigm and the non-neutrality of the observations with respect to the standard TPV paradigm [10, 19, 34, 50, 51]. While we find the term "First Person Vision" more specific for the computer vision community, the other terms will be also adopted in this thesis. Specifically, we will adopt the term "wearable" when it will be appropriate to highlight the possibility to wear such systems and the term "egocentric" to highlight the personal nature of the acquired data.

1.4.2 Context Awareness

Context awareness in First Person Vision has been investigated since the early days. Starner et al. [11] introduced an assistant for playing the "Patrol" game. The proposed wearable system was able to track the location of the user and understand the current task without using off-body infrastructure. Aoki et al. [52] designed a

[4] GoPro (https://gopro.com/), Narrative Clip (http://getnarrative.com/) and Google Glass (https://www.google.com/glass/start/) are some examples.

dynamic programming algorithm to recognize previously visited places on the basis of image sequences acquired by the user while approaching to the considered locations. Schiele et al. [33] proposed DyPERS, the "Dynamic Personal Enhanced Reality System". The system allowed to record audio-visual clips and associate them with specific visual objects. Recorded "media memories" were retrieved and played back to the user when the specific visual object was detected on the scene. Schiele et al. [53] present a wearable system which can be used as a museum's guide. The system was able to recognize objects in the user's field of view and display multimedia information that the user previously identified as being relevant to the object. Torralba et al. [54] propose a context-based wearable vision system which is able to identify familiar locations, categorize new environments and provide contextual priors for object recognition. Templeman et al [51] design a system to detect images of sensible spaces automatically acquired by always-on wearable cameras. Detected sensitive spaces (like bathrooms and bedrooms) can be blacklisted in order to preserve privacy. Sundaram and Mayol-Cuevas [55] classify actions from an egocentric field of view. The system also recognizes the user's location using a SLAM system and refine action classification using pre-learned action-location priors. Sundaram and Mayol-Cuevas [56] investigate a method that recognizes human activity observed from a moving camera and references such information to a previously mapped environment. Rhinehart and Kitani [57] learn a predict action maps of large environments. Action maps encode the ability of the user to perform activities in specific locations.

1.4.3 Activity and Action Recognition

Activity and action recognition are among the most investigated problems in First Person Vision. Activity and action recognition are different objectives: action recognition concerns the detection and correct classification of atomic actions and interactions with objects [50, 58], while activities are defined as complex sequences and interactions with objects to achieve a given goal. Examples of actions are, for instance, "beating eggs" or "opening the box", while examples of activities are "preparing a meal" or "doing the laundry" [18, 19]. Even if they are different objectives, the tasks of action and activity recognition have been often investigated together [19, 36, 50]. Spriggs et al. [50] used Intertial Measurement Units and a wearable camera

to segment human motion into actions and perform activity recognition. Kitani et al. [59] investigated methods to segment egocentric videos of sports into action categories. The approach assumed an unsupervised scenario where labeled training videos are not available and the number of action categories is not known in advance. Fathi et al. [18] presented a method to analyze daily activities such as meal preparation. The method performed inference about activities, actions, hands, and objects. Doherty et al. [60] investigated methods to recognize human activities from visual life-logs. Fathi et al. [43] designed a method to predict gaze and action labels jointly. Pirsiavash and Ramanan [19] investigated the recognition of daily activities from egocentric videos using an object-centric representation. Ryoo and Matthies [61, 62] considered a robot-centric scenario and proposed to recognize or predict actions performed by other subjects during their interaction with the robot. Li et al. [35] benchmarked different egocentric cues in the context of activity recognition. Yan et al. [48] designed a multi-task clustering algorithm to learn egocentric activities from multiple subjects in an unsupervised way. Castro et al. [63] presented a method to analyze egocentric images to recognize the egocentric activities of an individual. Singh et al. [64] proposed a Convolutional Neural Network for end to end learning and classification of egocentric actions. The method incorporated egocentric cues such as hand pose, head motion and saliency map. Ma et al [36] designed an activity recognition method which integrated hand segmentation, detection of objects of interest and action detection. Zhou et al [65] introduced cascaded interactional targeting deep neural networks to infer both hand and active object regions.

1.4.4 Indexing and Summarization

Long egocentric videos and egocentric photo streams are difficult to browse. To improve accessibility to such content, researchers have investigated method to index, summarize and extract egocentric visual data. Among the first researchers on this topic, Aizawa et al. [37] proposed an approach to automatic structuring and summarization of egocentric video. The approach used a sensor of brain waves and video features to automatically extract events of interest for the subject. Doherty et al. [66] investigated automatic segmentation of egocentric photo streams into events.

The proposed approach exploited the concept of novelty and a face-to-face conversation detector to help determine the importance of events in a lifelog. Jojic et al. [67] designed an unsupervised algorithm to create a visual summary from egocentric photo streams by discovering recurrent scenes, familiar faces and common actions. Aghazadeh et al. [68] demonstrated a system for the automatic extraction of novelty in egocentric images based on image sequence alignment. Lee et al. [69] introduced an approach to summarize egocentric videos focusing on the most important objects and people with which the user interacts. Lu and Grauman [38] designed a summarization approach that discovers the story of an egocentric videos by selecting a short chain of video subshots depicting the essential events. Poleg et al. [40, 41] investigated methods to segment egocentric videos according to long term activities such as standing, walking and biking, by analyzing user's motion. Bolaños et al. [70] surveyed methods to summarize egocentric photo streams arising from visual lifelogs. Xu et al. [71] formulated a gaze-enabled egocentric summarization method. Bo et al. [72] proposed a storyline representation of egocentric videos with an application story-based search using AND-OR graphs. Battadapura et al. [73] presented an approach for identifying highlights from large amount of egocentric vacation videos.

1.4.5 Attention Modeling

Other researches investigated the problem of modeling the user's visual attention from wearable devices. Yamada et al. [42] proposed a method for predicting egocentric attention combining bottom-up visual saliency and egomotion. Fathi et al. [43] introduced a probabilistic generative model for simultaneously recognizing daily actions and predicting gaze locations in egocentric video. Li et al. [44] presented a model for gaze prediction in egocentric video by exploiting different egocentric cues such as the wearer's head motion and hand location. Leelasawassuk et al. [74] investigated methods for the estimation of the user's visual attention from a head-worn Inertial Measurement Unit (IMU). Other authors argued that the availability of a gaze tracker can be beneficial for first person vision systems [10, 75, 76].

1.5 Thesis Outline

The thesis is divided into 4 chapters, plus two appendices. Each chapter treats a specific aspect of the investigated topics.

Chapter 2 investigates personal location recognition from first person videos.

Chapter 3 defines the task of next-active-object recognition and investigates the role of object trajectories.

Chapter 4 concludes the thesis and gives insights for future directions.

Appendix A reports complementary investigations on feature extraction from wide angle images.

Appendix B reports a list of works published during my Ph.D not directly inherent to this thesis.

Chapter 2

Recognizing Locations of Interest from Egocentric Videos

Contextual awareness is a desirable property in wearable computing [11, 12]. Context-aware systems can leverage the knowledge of the user's context to provide a more natural behavior and a richer human-machine interaction. Although different factors contribute to define the context in which the user operates, two important aspects seem to emerge from past research [12, 13]:

1. context is a dynamic construct and hence it is usually infeasible to enumerate a set of canonical contextual states independently from the user or the application;

2. even if context cannot be simply reduced to location, the latter still plays an important role in the definition and understanding of the user's context.

In particular, we argue that being able to recognize the locations in which the user performs his daily activities at the instance level (i.e., recognizing a particular environment such as "my office"), rather than at the category-level, (e.g., "an office"), can provide important information on the user, and help understanding his behavior and current objectives. Specifically, we define a personal location as:

> *a fixed, distinguishable, spatial environment in which the user can perform one or more activities which may or may not be specific to the considered location.*

An example of personal location may be the personal office desk in which the user can perform a number of activities, such as surfing the Web or writing e-mails.

Differently from the concept of scene category (as intended in [77]) personal locations are bound to the specific user and hence carry information related to his behavior and objectives. This relates to different applications in the domains of life-logging and personal information retrieval [78, 40], as well as to the domain of assistive technologies [79, 80].

For instance, a personal location aware system would allow to organize and access the acquired visual information on the basis of the detected personal locations and provide statistics on the behavior of the user (e.g., for stress monitoring), answering questions such as "how much time did I spend in my office last week?", "how many coffees did I have today?", "how many hours per-week do I usually spend driving?". The system could also be programmed to trigger specific behaviors or alerts according to the sensed location. This could include turning off unessential notifications when the user enters his office, assisting elder users in the interaction with a particular environment (e.g., reminding how to operate the TV or the microwave) or notifying the user that it's time to have a break after a long working session.

In this Chapter, we study how personal locations can be recognized from egocentric videos. Specifically, in Section 2.1 we discuss the related works. In Section 2.2 we discuss the specific challenges related to the recognition of personal locations from egocentric videos. In Section 2.5 we present a benchmark of the most popular visual representation techniques for scene and object recognition on the considered task. The benchmark is performed with respect to different wearable devices characterized by heterogeneous Fields of View (FOV) and wearing modalities. The analysis is extended in Section 2.6 with the introduction of a novel method for the rejection of negative locations (i.e., locations not of interest for the user) and with the augmentation of the preliminary benchmark dataset to a larger number of locations. In Section 2.7 we propose a method that further exploits temporal coherence to improve negative rejection as well as location recognition.

2.1 Related Work

Mobile and wearable cameras have been widely used in a variety of tasks, such as place and action recognition [11, 52], health and food intake monitoring [81, 82, 83], human-activity recognition and understanding [32, 18, 43, 75, 63, 48], video

indexing and summarization [40, 38, 84], as well as assistive-related technologies [79, 80]. The problem of recognizing personal locations from egocentric images, in particular, has already been investigated for different purposes and different methods have been proposed in the literature. The first investigations relevant to the considered problem date back to the late 90s. Starner et al. [11] proposed a context-aware system for assisting the users while playing the "patrol" game. The system proposed in [11] comprises a component able to recognize the room in which the player is operating combining RGB features and a Hidden Markov Model (HMM). Aoki et al. [52] proposed an image matching technique for the recognition of previously visited places. In this case, locations are not represented by a single frame, but rather by an image sequence of the approaching trajectory. Place recognition is implemented by computing the distance between a newly recorded trajectory and a dictionary of trajectories to known places. Torralba et al. [54] proposed a wearable system able to recognize familiar locations as well as categorize new environments. A low-dimension global representation based on a wavelet image decomposition is proposed in order to include textural properties of the image as well as their spatial layout. Familiar location recognition and new environment categorization are obtained separately training two distinct HMM models. More recently, in the wake of the popularity that always-on wearable cameras have recently gained, Templeman et al. [51] have proposed a system for "blacklisting" sensitive spaces (like bathrooms and bedrooms) to protect the privacy of the user when passively acquiring images of the environment. The system combines contextual information like GPS location and time with an image classifier based on local and global features and a HMM to take advantage of the temporal constraint on human motion. Images and short-video-based localization strategies have been already investigated in [85], where short videos are used to compute 3D-to-3D correspondences. The authors of [86] propose to model and recognize activity-related locations of interest to facilitate navigation in a visual lifelog. While the discussed approaches generally concentrate on video, some researchers have also investigated the use of low temporal-resolution devices. Such devices generally allow to acquire a few images per minute, but are characterized by a larger autonomy both in terms of memory and battery-life, which makes them particularly suited to acquire large amounts of visual data. In [63], daily activities are recognized from static images within a low temporal-resolution lifelog.

In [87], a method for semantic indexing and segmentation of photo streams is proposed. The reader is referred to the work by Bolaños et al. [70] for a review of the advances in egocentric data analysis.

As highlighted in [54], location recognition and place categorization are two related tasks and hence they are likely to share similar features in real-world applications. In this regard, much work has been devoted to designing suitable image representation for place categorization. Torralba and Oliva described a procedure for organizing real world scenes along semantic axes in [88], while in [89] they proposed a computational model for classifying real world scenes. Efficient computational methods for scene categorization have been proposed for mobile and embedded devices by Farinella et al. [90, 91]. More recently, Zhou et al. [92] have successfully applied Convolutional Neural Networks (CNNs) to the problem of scene classification.

Please note that, while past literature primarily focused on classification, we pay special attention to the problem of rejecting negative locations (i.e., locations not of interest for the user) which is an essential component for building real, robust and effective systems.

2.2 Challenges

Recognizing personal locations from egocentric videos poses some challenges due to the user-specific nature of the acquired visual information. In a real system the user should be able to specify a set of personal locations which he wishes to monitor. Since the locations are not known in advance by the system and they must be recognized at the instance-level, the user needs to provide training data for each location in order to "instruct" the system about what is meaningful for him. The data collection procedure should be simple enough to be performed by the inexperienced user. Moreover, relying on the acquired set of user-specified data, at run time the system should be able to: 1) detect the considered locations and 2) reject negative frames, i.e., frames not depicting any of the locations interesting for the user. Negative frames, in particular, naturally arise from two factors: 1) the user is likely to spend time in locations which he does not want to monitor (e.g., his colleague's office) and 2) as the user moves from a location to another, samples not related to a specific location may be acquired (e.g., the corridor). Considering

Challenge	**Constraints**	**Desired Feature**
negative samples	no training negatives	negative rejection abilities
user-gathered data	few training samples	learning from few samples
similar personal locations	scene recognition methods not suitable	instance-level recognition

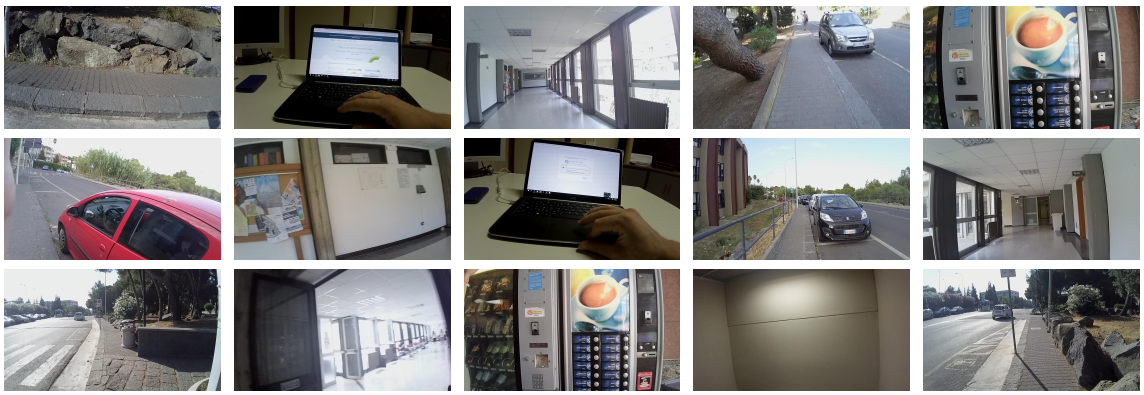
Table 2.1: Main challenges of a personal location recognition system.

possible real scenarios as above, in addition to the general issues which may be associated with egocentric data (e.g., camera blur and non-intentionality of the framing), recognizing personal locations involves some unique challenges:

- real-world systems must be able to correctly detect and manage negative samples, i.e., images depicting scenes not belonging to any of the desired locations of interest;
- given that an always-on wearable camera is likely to acquire a great variability of different scenes, gathering representative negative samples for modeling purposes is not always possible. In a real scenario, a system able to reject negatives given only user-specific positive samples for learning purposes is hence desirable;
- since personal locations are user-specific, few labeled samples are generally available as hence it is not feasible to ask the user to collect and annotate huge amounts of data for learning purposes;
- large intra-class variability usually characterizes the appearance of the different views related to a given location of interest;
- personal locations belonging to the same high level category (e.g., two different offices) tend to be characterized by similar scene shape and objects, making the discrimination challenging.

Table 2.1 summarizes the most important challenges of a personal location recognition system. Figure 2.1(a) shows some sample images acquired in different personal locations using a wearable camera. Figure 2.1(b) also reports some negative samples. Note that, since we define personal locations at the instance level, negative samples can be very similar to positive ones. For instance, a different coffee vending machine

(a) positive samples



(b) negative samples

Figure 2.1: Some egocentric images of possible personal locations of interest for the user. (a) Positive samples: each column reports two different shots of the same location acquired using a wearable camera. The following abbreviation holds: coffee v. machine - coffee vending machine. (b) Some negative samples.

(a) smart glasses (b) ear-mounted (c) chest-mounted

Figure 2.2: Three devices involving different wearing modalities: (a) smart glasses, (b) ear-mounted wearable camera, (c) chest-mounted wearable camera

or a different office should be classified as negatives. The figure illustrates the main variabilities described above. To take into account the discussed challenges, we will consider the following scenario: The user defines a number of locations of interest by providing minimal training data in the form of short videos (e.g., a 10 seconds video per location). The user is just asked to wear his camera and briefly look around while he is in the considered location. The user only provides positive samples and is not asked to acquire negative samples for training purposes.

2.3 Wearable Devices

The market proposes different wearable cameras, each with its distinctive features. We considered three main factors to characterize such devices: resolution, wearing modality and Field Of View (FOV). The resolution influences the amount of details that a given device is able to capture. While the first generation of wearable devices was characterized by very small resolutions (in the order of 0.1 mega-pixels), recent devices tend to adhere to the HD and 4K standards. The wearing modality influences the way in which the visual information is actually acquired. In particular, we identify three classes of devices characterized by different wearing modalities: smart glasses, ear mounted cameras and chest mounted cameras. Smart glasses are designed to substitute the user's glasses. Ear mounted cameras are worn similarly to bluetooth earphones and are a little more obtrusive than smart glasses. Both smart glasses and ear mounted devices have the advantage to capture the environment from the user's point of view. Chest mounted cameras are the least obtrusive since they are clipped to the user's clothes rather than mounted on his head (and easily ignored by both the wearer and the people he interacts with). However, the FOV captured by chest mounted cameras does not usually achieve much overlap

	Resolution		Wearing Modality			Field Of View	
	Medium	Large	Glasses	Ear	Chest	Narrow	Wide
RJ		✓	✓			✓	
LX2P	✓			✓		✓	
LX2W	✓			✓			✓
LX3		✓			✓		✓

Table 2.2: A summary of the main features of the considered devices. The following abbreviations hold: RJ - Recon Jet smart glasses, LX2P - Looxcie LX2 without wide-angular converter, LX2W - Looxcie LX2 with wideangular converter, LX3 - Looxcie LX3.

with the user's FOV. The Field Of View affects the quantity of visual information which is acquired by the device. A larger FOV allows to acquire more information in a similar way to the human visual system at the cost of the introduction of radial distortion. Figure 2.2 depicts three devices involving the aforementioned wearing modalities.

In order to assess the influence of the aforementioned device-specific factors for the problem of personal location recognition, we consider four different devices: the smart glasses Recon Jet (RJ)[1], two ear-mounted Looxcie LX2[2], and a wide-angular chest-mounted Looxcie LX3[3]. The Recon Jet and Looxcie LX3 devices produce images at the HD resolution (1280×720 pixels), while the Looxcie LX2 devices have a smaller resolution of 640×480 pixels. The Recon Jet and the Looxcie LX2 devices are characterized by narrow FOVs ($70°$ and $65,5°$ respectively), while the FOV of the Looxcie LX3 is considerably larger ($100°$). One of the two ear-mounted Looxcie LX2 is equipped with a wide-angular converter in order to achieve a large FOV (approximatively $100°$). The wide-angular LX2 camera will be indicated with the acronym LX2W, while the regular (perspective) LX2 camera will be indicated as LX2P. Table 2.2 summarizes the main features of the cameras used to acquire the data.

[1] http://www.reconinstruments.com/products/jet/
[2] http://www.looxcie.com
[3] http://www.looxcie.com

Dataset	**#L**	**Device**	**Overlap**
5-LOCATIONS	5	RJ, LX2P, LX2W, LX3	—
8-LOCATIONS	8	RJ, LX2P, LX2W, LX3	car, cvm, office, tv, h.office are taken from 5-LOCATIONS
10-LOCATIONS	10	LX2W	car, cvm, office, tv, h.office, lab office, garage are taken from 8-LOCATIONS

Table 2.3: Summary of the content of the three considered datasets. The table reports the number of locations in each dataset (#L), the devices used to acquire the data and the overlap with other datasets.

2.4 Datasets

For our analysis, we have collected three distinct datasets of egocentric videos acquired in different personal locations. The datasets have been collected in an incremental fashion (i.e., each dataset extends the previous one) by the same single user using the hardware described in the previous section. Therefore, the datasets share some footage and similar acquisition settings. The datasets contain videos of 5, 8 and 10 locations respectively, and hence they will be referred using three unique names: 5-LOCATIONS, 8-LOCATIONS and 10-LOCATIONS. The following sections discuss the details of each of the considered datasets. To help understand the differences and similarities between the datasets, a summary is reported in Table 2.3.

2.4.1 5-LOCATIONS

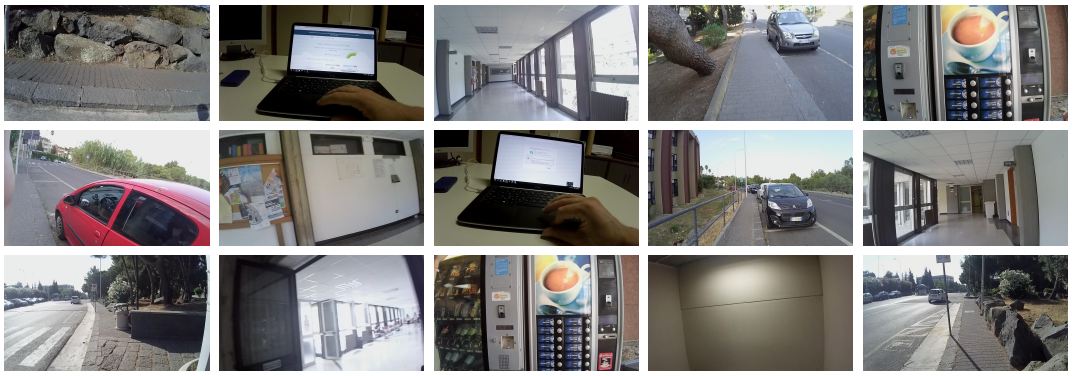
This dataset has been acquired by a single user in five different personal locations using the four devices discussed in Section 2.3. The considered five personal locations arise from the daily activities of the user and are relevant to assistive applications such as quality of life assessment and daily routine monitoring: car, coffee vending machine, office, TV and home office. Figure 2.3(a) shows some samples from the dataset with respect to the considered wearable devices. Since each of the considered location involves one or more static activities, we assume that the user is free to turn his head and move his body when interacting with the environment, but he does not change his position in the room. In line with the considerations discussed in Section 2.2, our training set is composed of very short videos ($\approx$ 10 seconds) of the locations of interest for a person (just one video per location) to be monitored.

During the acquisition of the training videos, the user is asked to turn his head (or chest, in the case of chest-mounted devices) in order to capture a few different views of the environment. Please note that, in the training stage, the user is assumed to be static and only one training video from a single position is acquired for each class. The test set consists in medium length (8 to 10 minutes) videos of normal activity in the given personal locations with the different devices. Three to five testing videos have been acquired for each location. We also acquired several short videos containing likely negative samples, such as indoor and outdoor scenes, other desks and other vending machines. Please note that the negative samples contained in this dataset are mainly related to the first of the two sources of negative samples discussed in Section 2.2, i.e., locations not of interest form the user. Few negatives related to transitions between different locations (e.g., corridors) are also included. Figure 2.3(b) shows some negative samples. Most of the negative-videos are used solely for testing purposes, while a small part of them is used to extract a fixed number (200 in our experiments) of frames which will be used as "optimization negative samples" in order to optimize the performances of the compared methods. At training time, all the frames contained in the "10-seconds" video shots are used, while at test time, only about 1000 frames per-class uniformly sampled from the testing videos are used. In order to perform fair comparisons across the different devices, we built four independent, yet compliant, device-specific datasets. Each dataset comprises data acquired by a single device and is provided with its own training and test sets. The device-specific datasets are available for download at the URL: http://iplab.dmi.unict.it/PersonalLocations/.

2.4.2 8-LOCATIONS

This dataset extends 5-LOCATIONS and contains about 7 more hours of new video. The material has been acquired by the same subject using the four considered devices. Specifically, three more locations have been included: Kitchen Top, Sink and Garage. This dataset also reuses the same negative samples included in the 5-LOCATIONS dataset. The full set of 8 personal locations arises from possible daily activities of a user: Car, Coffee Vending Machine (C. V. M.), Office, Living Room (L. R.), Home Office (H. Office), Kitchen Top (K. Top), Sink, Garage. The dataset includes similar looking locations (e.g., Office vs Home Office) and locations

(a) positive samples



(b) negative samples

Figure 2.3: (a) Some sample images of the five personal locations of the 5-LOCATIONS dataset. Images from the same locations are grouped by columns, while images acquired using the same device are grouped by rows. The following abbreviation holds: coffee v. machine - coffee vending machine. (b) Some negative samples used for testing purposes.

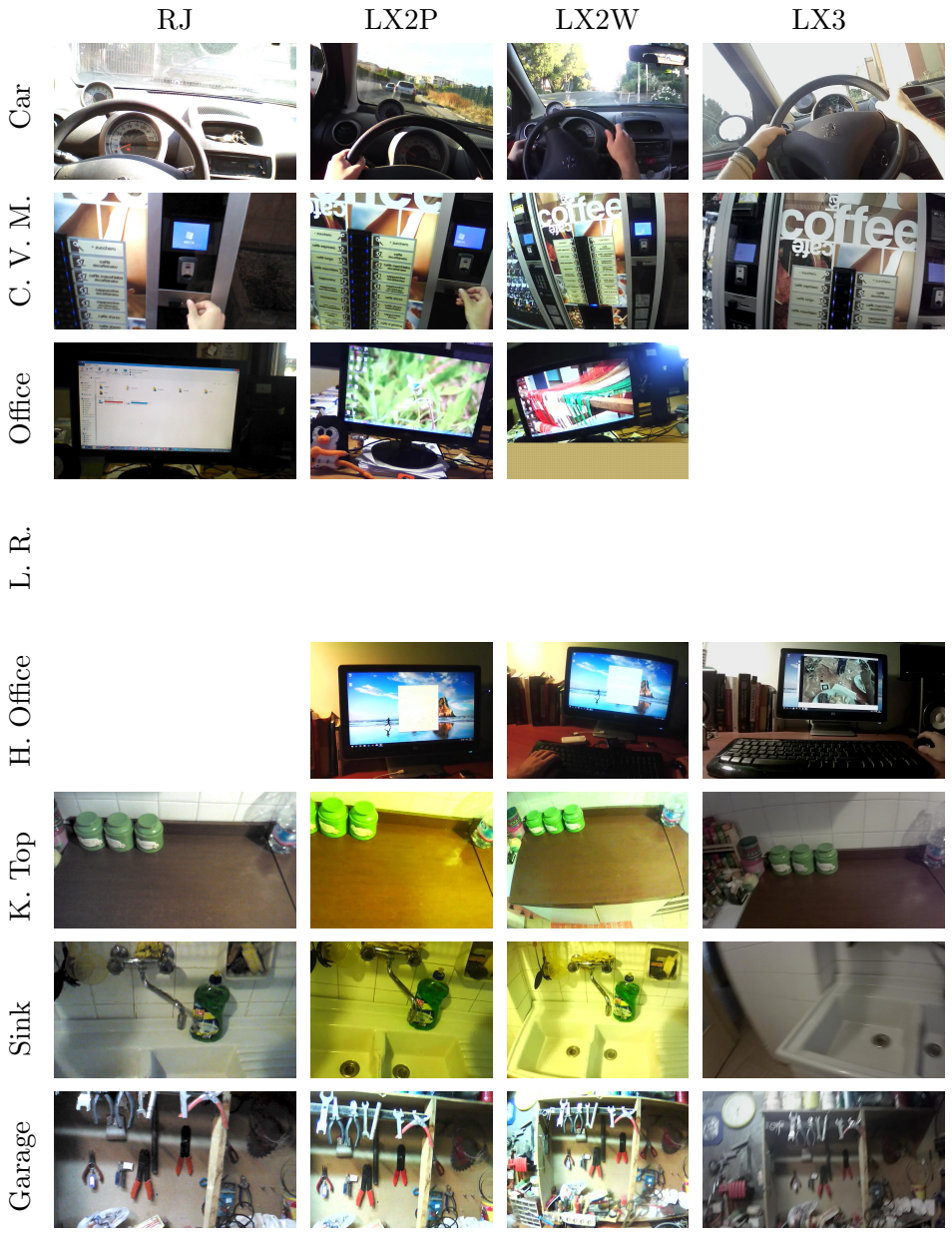
characterized by large intra-class variability (e.g., Garage). Figure 2.4 shows some sample frames belonging to the dataset.

This overall dataset amounts to more than 20 hours of video and more than one million frames in total. In order to facilitate the analysis of such a huge quantity of collected data, we extract each frame in the training videos and temporally sub-sample the testing videos. To reduce the amount of frames to be processed, for each location in the test sets, we extract 200 subsequences of 15 contiguous frames. This sub-sampling still allows to consider temporal coherence. The starting frames of the subsequences are uniformly sampled from the 5 videos available for each class. The same sub-sampling strategy is applied to the test negatives. We also extract 300 frames form the optimization negative videos. This amounts to a total of 133770 extracted frames to be used for experimental purposes. The dataset is available at the following URL: http://iplab.dmi.unict.it/PersonalLocations/.

2.4.3 10-LOCATIONS

This dataset has been acquired using only the LX2W camera (Looxcie LX2 + wideangular converter). It partially extends 8-LOCATIONS, and introduces some new footage related to two new location: Piano and Studio. Please note that footage related to the Sink, Kitchen and Living Room locations is not the same contained in 8-LOCATIONS and 5-LOCATIONS. The overall dataset contains video related to 10 different personal locations, plus various negative ones. The negative samples included in the 5-LOCATIONS and 10-LOCATIONS datasets are also included in this dataset. The full list of location is related to a possible daily routine: Car, Coffee Vending Machine (C.V.M.), Office, Lab Office (L.O.), Living Room (L.R.), Piano, Kitchen Top (K.T.), Sink, Studio, Garage. Similarly to the previously discussed datasets, the 10-LOCATIONS dataset exhibits a high degree of intra-class variability (e.g., Car and Garage classes) and small inter-class variability in some cases (e.g., Office, Lab Office and Studio classes).

Coherently with what discuss earlier in this section, we assume that the user is required to provide only minimal data to define his personal locations of interest. Therefore, the training set consists in 10 short videos (one per each location) with an average length of 10 seconds per video. Differently from previous datasets, the test set consists in 10 video sequences covering the considered personal locations of



(a) positive samples

(b) negative samples

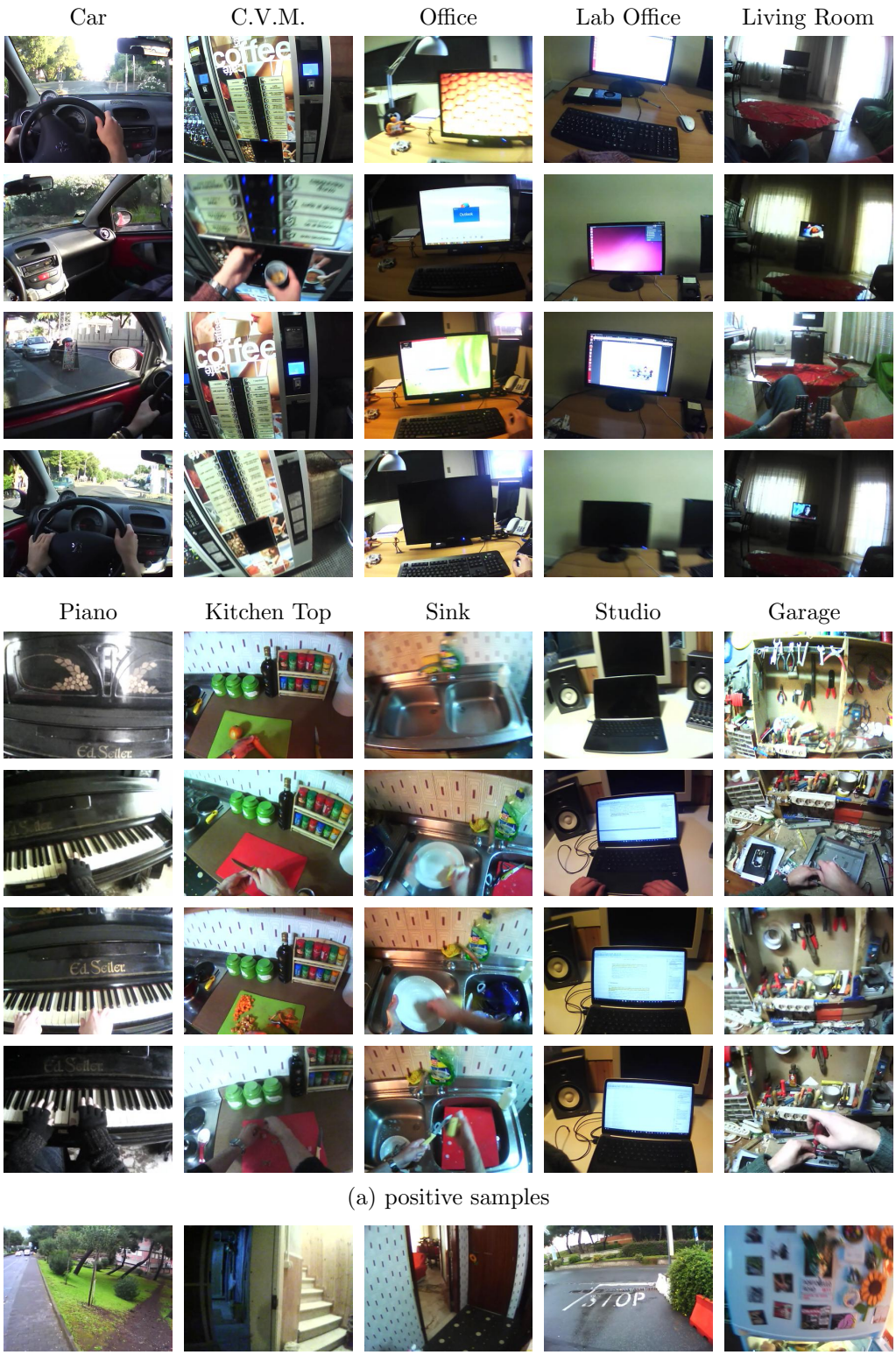
Figure 2.4: (a) Some sample images from the 8-LOCATIONS dataset. Images related to the same locations are on the same row, while images acquired using a specific device are on the same column. (b) Some negative samples used for testing purposes.

Sequence	**Context transitions**	**Length**
1	Car → N → Office → N → Lab Office	00:11:27
2	Office → N → Lab Office	00:05:55
3	Lab Office → N → Office → N → C.V.M.	00:07:24
4	TV → N → Piano → N → Sink	00:11:40
5	Kitchen → N → Sink → N → Piano	00:10:41
6	Kitchen → N → Sink → N → TV	00:11:18
7	Piano → N → Sink → N → TV	00:04:57
8	Studio → N → Car → N → Garage	00:06:51
9	Car → N → Garage → N → Studio	00:05:17
10	Car → N → Studio → N → Garage	00:06:05
	Total length	01:21:35

Table 2.4: A summary of the location transitions contained in the test sequences of the 10-LOCATIONS dataset. "N" represents a negative segment (to be rejected by the final system).

interest, negative locations and transitions among locations. Each frame in the test sequences has been manually labeled as either one of the 10 locations of interest or as a negative. Table 2.4 summarizes the content of the test sequences with the related transitions. Figure 2.5 shows some samples from the acquired dataset. We also report the total time spent by the user in each of the considered locations in Table 2.5. As can be noted, some locations (e.g., C.V.M.) tend to be less visited than others (e.g., Sink). It should be noted that such information is not available at training time and hence it cannot directly be used to improve recognition performances (for instance, by weighting classes differently on the basis of their natural occurrence in real scenarios).

The dataset is also provided with an independent validation set which can be used for optimize the hyper-parameters. The validation set contains 10 medium length (approximately 5 to 10 minutes) videos of activities performed in the considered locations (one video per location). Validation videos have been temporally subsampled in order to extract exactly 200 frames per-location, while all frames are considered for training and test videos. We have also acquired 10 medium length videos containing negative samples from which we uniformly extract 300 frames for training and 200 frames for validation. Negative samples are provided in order to allow comparisons with methods which explicitly learn from negatives. Please note that the proposed method does not need to learn from negatives and hence it discards them at training time. Please note that both of the sources of negative samples



(a) positive samples

(b) negative samples

Figure 2.5: Some sample frames from the 10-LOCATIONS dataset.

Location	**Time (seconds)**
Car	293
C.V.M.	35
Garage	269
Kitchen Top	392
Lab Office	478
Office	228
Piano	459
Sink	700
Studio	409
Living Room	459
Negatives	680

Table 2.5: Total time spent by the user in each location (including negatives) in the whole dataset.

discussed in Section 2.2 (i.e., locations not of interest and transitions between locations). The overall dataset contains 2142 positive, plus 300 negative frames for training, 2000 positive, plus 200 negative frames for validation and 132234 mixed (both positive and negative) frames for testing purposes. The dataset is publicly available at the web page http://iplab.dmi.unict.it/PersonalLocations/.

2.5 Benchmark of Representations and Wearable Devices

We begin to study the problem of recognizing personal locations of interest from egocentric images performing a benchmark of different state-of-the-art methods for scene and object classification. In order to assess the influence of device-specific factors, such as the wearing modality and the Field Of View (FOV), we consider the 5-LOCATIONS dataset, which contains egocentric videos of 5 different personal locations acquired using 4 wearable cameras. To make the analysis worth in real-world scenarios where personal locations of interest need to be discriminated from negative samples, we consider a simple classification pipeline which includes a mechanism for the rejection of negative samples.

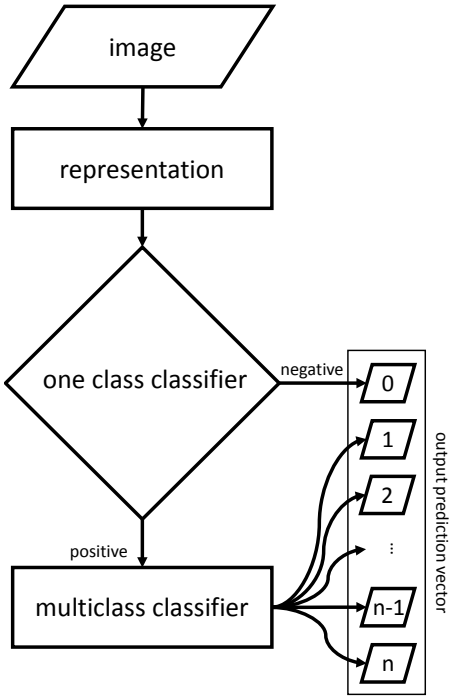


Figure 2.6: The considered classification pipeline combining a one-class with a multi-class classifier.

2.5.1 Classification Pipeline

As already discussed, a personal location recognition system should be able to discriminate among different personal locations specified by the user and reject negative frames (i.e., frames not related to any of the considered locations). Therefore, we consider a baseline classification pipeline made up of two main components: a linear one-class classifier to reject negative samples and a linear multiclass classifier to discriminate among different personal locations. Figure 2.6 depicts the considered pipeline. The classification into the $n+1$ different classes (the "negative" class, plus n location-related classes) is obtained using a cascade of a one-class SVM (OCSVM) and a regular multi-class SVM (MCSVM). The OCSVM detects the negative samples and assigns them to the negative class. All the other samples are fed to the MCSVM for location discrimination. Since the input to this pipeline is a single image, no temporal coherence is leveraged to perform the predictions. This aspect will be investigated in Section 2.6 and Section 2.7. Please note that the proposed classification pipeline is to be considered a baseline. At this stage, our main focus is

on performing a benchmark of representations/devices, and not on the recognition system itself.

2.5.2 Representations

We assume that the input image I can be mapped to a feature vector $\mathbf{x} \in \Re^d$ through a representation function. Specifically, we consider three different classes of representation functions: holistic, shallow and deep. All of these representations have been used in the literature for different tasks related to scene understanding [89, 92] and object detection [93, 94]. In the following subsections we discuss the details of the considered representations and the related parameters.

Holistic Representations

Holistic feature representations have been widely used in tasks related to scene understanding [89, 91]. Their aim is to provide an image description encoding distinctive scene-related features like global edge orientations (the so called "spatial envelope" [89]), while discarding instance-specific variabilities (e.g., the location of specific objects). As a popular representative of this class, we consider the GIST descriptor proposed in [89] and use the standard implementation and parameters provided by the authors[4]. According to the standard implementation, all input images are resized to the normalized resolution of 128×128 pixels prior to computing the descriptor. In this configuration, the output GIST descriptors have dimensionality $d = 512$.

Shallow Representations

With deep feature representations and Convolutional Neural Networks (CNNs) becoming mainstream in the Computer Vision literature, classic representation schemes based on the encoding of local features (e.g., Bag of Visual Word models) have been recently referred to as shallow feature representations [94]. The term "shallow" is used to highlight that features are not extracted hierarchically as in deep learning models. On the contrary, there is a single feature extraction layer where local feature are extracted (e.g., SIFT descriptors) and a description one where some encoding

[4] http://people.csail.mit.edu/torralba/code/spatialenvelope/

strategy is used to summarize the visual content of the image. Among the different Bag of Visual Word models, we consider Improved Fisher Vectors (IFV) [95] to encode densely-sampled SIFT features. This schema generally outperforms other encoding paradigms and can be considered the state-of-the-art in shallow representations for object classification [93, 94].

The IFV features are extracted following the procedures described in [93, 94]. As a first step, SIFT descriptors are densely extracted from each training and test image. As it is suggested in [93], we use the *vl_phow* function of the VLFeat library [96] to densely sample SIFT features at multiple scales. To make computation tractable on a large number of frames, each input image is resized to a normalized height of 300 pixels keeping the original aspect ratio. This produces images of resolutions 400×300 pixels and 533×300 pixels in our dataset. Afterwards, SIFT descriptors are component-wise square-rooted and their dimensionality is reduced to 80 components using Principal Component Analysis (PCA) [97]. Apart from the standard SIFT descriptors, we also consider the spatially-enhanced local descriptors discussed in [94]. Such descriptors are obtained concatenating the coordinates of the location from which the SIFT descriptor is extracted to the PCA-reduced SIFT features, obtaining a 82-dimensional vector as detailed in [94]. Gaussian Mixture Model (GMM) with $K = 256$ centroids are trained on the PCA-decorrelated SIFT descriptors extracted from all images in the training set (negatives are excluded) in order to build a visual codebook. We also consider large codebooks ($K = 512$) in our experiments. The IFV feature vectors are obtained concatenating the average first and second order differences between the local descriptors and the centers of the learned GMM [93]. The dimensionality d of IFV descriptors depends on the number of clusters K of the GMM codebook and the number of dimensions D of the local feature descriptors (i.e., SIFT) according to the formula: $d = 2KD$. Using the aforementioned parameters, the number of dimensions of our IFV representations ranges from a minimum of 40960 to a maximum of 83968 components. The VLFeat library [96] has been used to perform all the operations involved in the computation of the IFV representations.

Deep Representations

Convolutional Neural Networks (CNNs) have demonstrated state-of-the-art performances in a series of tasks including object and scene classification [92, 94, 98]. They allow to learn multi-layer representations of the input images which are optimal for a selected task (e.g., object classification). CNNs have also demonstrated excellent transfer properties, allowing to "reuse" a representation learned for a given task in a slightly different one. This is generally done extracting the representation contained in the penultimate layer of the network and reusing it in a classifier (e.g., SVM) or finetuning the pre-trained network with new data and labels. We consider three publicly available networks which have demonstrated state-of-the-art performances in the tasks of object and scene classification, namely AlexNet [98], VGG [94] and Places205 [92]. AlexNet and VGG have different architectures but they have been trained on the same data (the ImageNet dataset). Places205 has the same architecture as AlexNet, but it has been trained to discriminate locations on a dataset containing 205 different scene categories. The different network architectures allow us to assess the influence of both network architectures and original training data in our transfer learning settings. To build our deep representations, we extract for each network model the values contained in the penultimate layer when the input image (rescaled to the dimensions of the first layer) is propagated into the network. This consists in a compact 4096-dimensional vector which corresponds to the representation contained in the hidden layer of the final Multilayer Perceptron included in the network.

2.5.3 Experimental Settings

Experiments are performed on the 5-LOCATIONS dataset. The aim of the experiments is to study the performances of the state-of-the-art representations and acquisition devices discussed in the previous Section on the considered task. Following [93, 94], input feature vectors are transformed using the Hellinger's kernel prior to feed them to the linear SVM classifier. Since the Hellinger's kernel is additive homogeneous, its application can be efficiently implemented as detailed in [93]. Differently from [93, 94], we do not apply the L2 normalization to the feature vectors, but instead we independently scale each component of the vectors in the range $[-1, 1]$

subtracting the minimum and dividing by the difference between the maximum and minimum values. Minima and maxima for each component are computed from the training set and reported on the test set. This overall preprocessing procedure outperforms or gives similar results to the combination of other kernels (i.e., gaussian, sigmoidal) and normalization schemes (i.e., L1, L2) in preliminary experiments.

To implement the OCSVM component, we consider the method proposed in [99]. Its optimization procedure depends on a single parameter ν which is a lower bound on the fraction of outliers in the training set. In our settings, the training set consists in all the positive samples from the different locations and hence it does not contain outliers by design. Nevertheless, since the performances of the OCSVM are sensitive to the value of parameter ν, we use the small subset of negative samples available along with the training set, to choose the value of ν which maximizes the accuracy on the training-plus-negatives samples. It should be noted that the negative samples are only used to optimize the value of the ν parameter and they are not used to train the OCSVM. The multiclass component has been implemented with a multiclass SVM classifier. Its optimization procedure depends only on the cost parameter C. At training time, we choose the value of C which maximizes the accuracy on the training set using cross-validation similarly to what has been done in other works [93, 94].

The outlined training and testing pipeline is applied to different combinations of devices and representations/parameters in order to assess the influence of using different devices to acquire the data and different state-of-the-art representations. It should be noted that all the parameters involved in the classification pipeline are computed independently in each experiment in order to yield fair comparisons. We use LibSVM library in all our experiments [100].

2.5.4 Experimental Results

In order to assess the performances of each component of the considered baseline classification pipeline, we report the overall accuracy of the system, as well as the performance measures for the one-class and multi-class components working independently. The overall accuracy of the system (ACC) is computed simply counting the fraction of the input images correctly classified by the cascade pipeline into one

of the possible six classes (five locations, plus the "negative" class). The performances of the OCSVM component, are assessed reporting the True Positive Rate (TPR) and the True Negative Rate (TNR). Since the accuracy of the one-class classifier can be biased by the large number of positive samples (about 5000), versus the small number of negatives (about 1000), we report the average between TPR and TNR, which we refer to as True Average Rate (TAR):

$$TAR = \frac{TPR + TNR}{2}. \tag{2.1}$$

The performances of the MCSVM are assessed bypassing the OCSVM component and running the MCSVM only on the positive samples of the test set. We report the Multi-Class Accuracy (MCA), i.e., the fraction of samples correctly discriminated into the 5 locations, and the per-class True Positive Rates. Table 2.6 reports the results of all the experiments. Each row of the table corresponds to a different experiment and is denoted by a unique identifier in brackets (e.g., $[a_1]$). The GMM used for the IFV representations have been trained on all the descriptors extracted from the training set (excluding the negatives) using the settings specified in the table. The table is organized as follows: the first column reports the unique identifier of the experiment and the used representation; the second column reports the device used to acquire the pair of training and test sets; the third column reports the options of the representation, if any; the fourth column reports the dimensionality of the feature vectors; the fifth column reports the overall accuracy of the cascade (one-class and multi-class classifier) classifier on the six classes; the sixth column reports the One-Class Average Ratio (OAR) of the OCSVM classifier; the seventh and eighth columns report the TPR and TNR values for the OCSVM; the ninth column reports the accuracy of the MCSVM classifier (MCA) working independently from OCSVM on the five locations classes. The remaining columns report the true positive rates for the five different personal locations classes. To improve the readability of the table, the per-column maximum performance indicators among the experiments related to a given device are reported as boxed numbers, while the global per-column maxima are reported as underlined numbers.

In the reported results the performance indicators of the MCSVM are in average better than the ones of the OCSVM. This difference is partly due to the fact that one-class classification is usually "harder" than multi-class classification due to the

Method	Dev.	Options	Dim.	**ACC**	**TAR**	TPR	TNR	**MCA**	car	c.v.m.	office	TV	h. off.
[a_1] GIST	RJ	—	512	**38,96**	**50,52**	91,54	9,50	**49,85**	43,76	90,84	14,20	76,26	46,78
[b_1] IFV	RJ	KS 256	40960	**42,17**	**46,70**	91,20	2,20	**51,25**	62,28	53,82	34,69	98,69	38,37
[c_1] IFV	RJ	KS 512	81920	**42,16**	**46,61**	90,82	2,40	**51,21**	62,21	53,85	34,55	98,90	38,58
[d_1] IFV	RJ	KS SE 256	41984	**43,24**	**45,42**	85,14	5,70	**53,73**	69,08	50,22	34,65	99,11	46,62
[e_1] IFV	RJ	KS SE 512	83968	**36,06**	**45,68**	89,66	1,70	**44,03**	77,80	46,41	29,65	97,00	21,88
[f_1] IFV	RJ	DS 256	40960	**43,77**	**52,35**	93,50	11,20	**52,63**	65,58	49,50	27,98	91,51	86,92
[g_1] IFV	RJ	DS 512	81920	**47,46**	**48,82**	88,74	8,90	**60,33**	84,34	55,51	37,79	78,09	52,10
[h_1] IFV	RJ	DS SE 256	41984	**47,91**	**49,37**	91,74	7,00	**59,83**	78,92	70,49	40,73	66,96	88,15
[i_1] IFV	RJ	DS SE 512	83968	**49,51**	**45,77**	81,34	10,20	**67,51**	83,80	65,75	41,78	78,73	67,77
[j_1] CNN	RJ	AlexNet	4096	**49,26**	**48,17**	67,03	29,30	**79,50**	93,07	97,10	57,25	94,00	62,10
[k_1] CNN	RJ	Places205	4096	**55,19**	**53,02**	80,14	25,90	**78,02**	97,29	98,43	69,69	96,14	50,86
[l_1] CNN	RJ	VGG	4096	**54,54**	**53,78**	63,35	44,20	**85,26**	94,54	89,83	77,10	90,54	73,27
[a_2] GIST	LX2P	—	512	**48,62**	**61,53**	96,56	26,50	**54,15**	74,15	99,81	30,41	82,68	32,02
[b_2] IFV	LX2P	KS 256	40960	**51,19**	**55,68**	79,26	32,10	**70,93**	60,17	98,40	56,65	98,97	55,16
[c_2] IFV	LX2P	KS 512	81920	**63,83**	**54,97**	95,64	14,30	**76,90**	59,84	97,23	68,39	96,92	72,17
[d_2] IFV	LX2P	KS SE 256	41984	**50,66**	**56,43**	79,16	33,70	**69,75**	58,80	98,29	54,87	98,96	53,10
[e_2] IFV	LX2P	KS SE 512	83968	**59,08**	**50,54**	97,48	3,60	**71,99**	58,29	98,03	60,93	98,44	62,11
[f_2] IFV	LX2P	DS 256	40960	**46,62**	**52,10**	88,10	16,10	**61,73**	71,33	75,65	26,08	62,62	56,10
[g_2] IFV	LX2P	DS 512	81920	**50,59**	**52,20**	90,00	14,40	**65,15**	77,70	68,41	31,21	72,75	66,59
[h_2] IFV	LX2P	DS SE 256	41984	**41,79**	**47,22**	80,64	13,80	**57,61**	74,62	76,88	32,42	71,65	39,86
[i_2] IFV	LX2P	DS SE 512	83968	**56,24**	**55,85**	94,00	17,70	**68,29**	77,34	84,29	37,44	88,29	52,78
[j_2] CNN	LX2P	AlexNet	4096	**48,16**	**51,31**	66,31	36,30	**76,10**	80,54	78,98	50,45	100,0	70,66
[k_2] CNN	LX2P	Places205	4096	**54,84**	**57,30**	60,89	53,70	**87,14**	99,19	92,20	63,38	99,88	96,45
[l_2] CNN	LX2P	VGG	4096	**50,74**	**57,40**	56,39	58,40	**86,02**	98,60	81,04	74,11	99,75	80,21
[a_3] GIST	LX2W	—	512	**61,27**	**60,02**	93,66	26,37	**73,91**	87,51	100,0	80,05	83,84	48,29
[b_3] IFV	LX2W	KS 256	40960	**55,47**	**61,89**	89,92	33,87	**67,27**	55,46	99,30	38,77	98,78	61,73
[c_3] IFV	LX2W	KS 512	81920	**54,82**	**63,41**	88,46	38,36	**66,93**	57,55	99,30	40,58	99,26	57,14
[d_3] IFV	LX2W	KS SE 256	41984	**49,73**	**50,08**	88,38	11,79	**66,53**	63,29	99,69	42,45	99,28	47,94
[e_3] IFV	LX2W	KS SE 512	83968	**55,08**	**54,90**	91,52	18,28	**67,95**	53,43	99,80	46,75	100,0	55,86
[f_3] IFV	LX2W	DS 256	40960	**59,62**	**52,77**	94,36	11,19	**72,81**	87,40	95,28	66,94	97,33	48,22
[g_3] IFV	LX2W	DS 512	81920	**60,50**	**52,77**	95,86	9,69	**73,15**	75,52	90,04	73,72	99,81	53,60
[h_3] IFV	LX2W	DS SE 256	41984	**57,88**	**49,01**	87,84	10,19	**74,33**	82,26	93,71	74,33	99,60	51,99
[i_3] IFV	LX2W	DS SE 512	83968	**62,65**	**54,59**	96,40	12,79	**75,74**	69,61	97,51	79,32	98,85	58,93
[j_3] CNN	LX2W	AlexNet	4096	**71,23**	**70,00**	81,46	58,54	**91,34**	99,70	96,23	90,36	99,03	76,50
[k_3] CNN	LX2W	Places205	4096	**61,63**	**63,77**	66,49	61,04	**94,02**	99,90	99,90	93,90	99,65	80,17
[l_3] CNN	LX2W	VGG	4096	**66,02**	**71,91**	69,29	74,53	**94,42**	100,0	99,60	93,79	99,64	81,91
[a_4] GIST	LX3	—	512	**42,08**	**65,23**	77,86	52,59	**53,07**	65,16	95,24	31,91	58,36	26,55
[b_4] IFV	LX3	KS 256	40960	**40,51**	**49,88**	82,50	17,27	**62,07**	67,21	90,19	46,31	99,15	20,47
[c_4] IFV	LX3	KS 512	81920	**40,21**	**47,23**	83,38	11,08	**62,13**	67,33	90,19	46,37	99,15	20,74
[d_4] IFV	LX3	KS SE 256	41984	**41,48**	**47,61**	85,64	9,58	**61,49**	66,07	89,35	47,04	98,87	16,61
[e_4] IFV	LX3	KS SE 512	83968	**40,49**	**51,34**	81,92	20,76	**61,35**	66,19	89,23	45,58	99,15	19,17
[f_4] IFV	LX3	DS 256	40960	**59,07**	**61,20**	93,46	28,94	**68,81**	78,72	83,11	47,00	92,49	73,08
[g_4] IFV	LX3	DS 512	81920	**63,31**	**50,69**	89,50	11,88	**81,92**	90,82	92,61	59,97	99,81	86,23
[h_4] IFV	LX3	DS SE 256	41984	**67,54**	**58,78**	92,32	25,25	**82,70**	88,61	84,00	66,67	99,29	89,39
[i_4] IFV	LX3	DS SE 512	83968	**66,02**	**57,83**	91,80	23,85	**81,08**	91,42	90,55	58,93	99,84	78,23
[j_4] CNN	LX3	AlexNet	4096	**54,49**	**67,42**	75,16	59,68	**76,32**	99,80	99,90	50,85	97,88	20,23
[k_4] CNN	LX3	Places205	4096	**52,87**	**72,01**	55,19	88,82	**86,28**	95,97	98,21	62,36	97,47	99,12
[l_4] CNN	LX3	VGG	4096	**59,12**	**69,68**	74,59	64,77	**80,74**	99,60	100,0	51,31	99,01	77,63

Table 2.6: The results related to experiments performed on the 5-LOCATION dataset. Per-column maximum performance indicators among the experiments related to a given device are reported as boxed numbers, while the global per-column maxima are reported as underlined numbers.

limited availability of representative counterexamples. Furthermore, it can be noted that many of the considered representations yield inconsistent one-class classifiers characterized by large TPR values and very low TNR values. This effect is in general mitigated when deep features are used, which suggests that better performances could be achieved with suitable representations. Moreover, the performances of the one-class classifier have a large influence on the performances of the overall system, even in the presence of excellent MCA values as in the case of $[j_3]$, $[k_3]$ and $[l_3]$. For example, while the $[l_3]$ method reaches an MCA accuracy equal to $94,42\%$ when only discrimination between the five different locations is considered, it scores a OAR accuracy as low as $71,91\%$ on the one-class classification problem, which results in the overall system accuracy (ACC) of $66,02\%$.

The results related to the MCSVM are more consistent. In particular, the deep features systematically outperform any other representation methods, which suggests that the considered task can take advantage of transfer learning techniques, given the availability of a small amount of labeled data (i.e., we can use models already trained for similar tasks to build the representations). Interestingly, the simple GIST descriptor, gives remarkable performances when used on wide angle images acquired by the LX2W device (i.e., experiment $[a_3]$), where an MCA value of $73,91\%$ is achieved. The different experiments with the IFV-based representations highlight that the keypoint-based extraction scheme (KS) has an advantage over the dense-based (DS) extraction scheme only when the narrow FOV LX2P device is used, while dense-based extraction significantly outperforms the keypoints-based extraction scheme when the field of view is larger, i.e., for the LX2W and LX3 devices. Moreover, when a dense-based extraction scheme is employed, spatially-enhanced descriptors (SE) outperform their non-spatially-enhanced counterparts. The use of larger GMM codebooks (i.e., $K = 512$ clusters) often (but not always, as in the cases of $[e_1]$ vs $[d_1]$ and $[i_4]$ vs $[h_4]$) allows to obtain better performances. However, this come at the cost of dealing with very large representation vectors (in the order of $80K$ vs $40K$ dimensions).

As a general remark, devices characterized by larger FOVs tend to have a significant advantage over the narrow-FOV devices. This is highlighted in Figure 2.7 which reports the minima, maxima and average ACC values (accuracy of the overall system) for all the experiments related to a given device. These statistics clearly

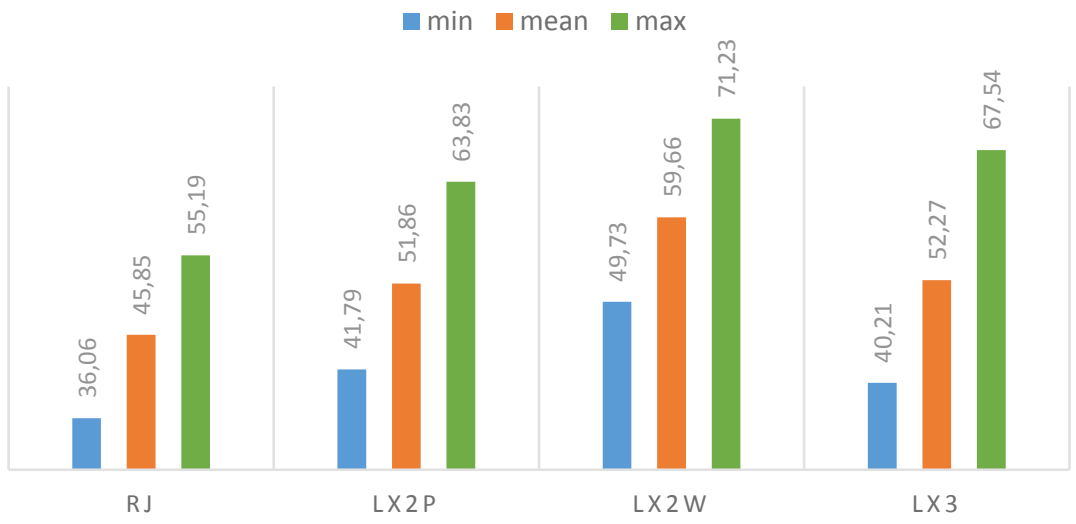
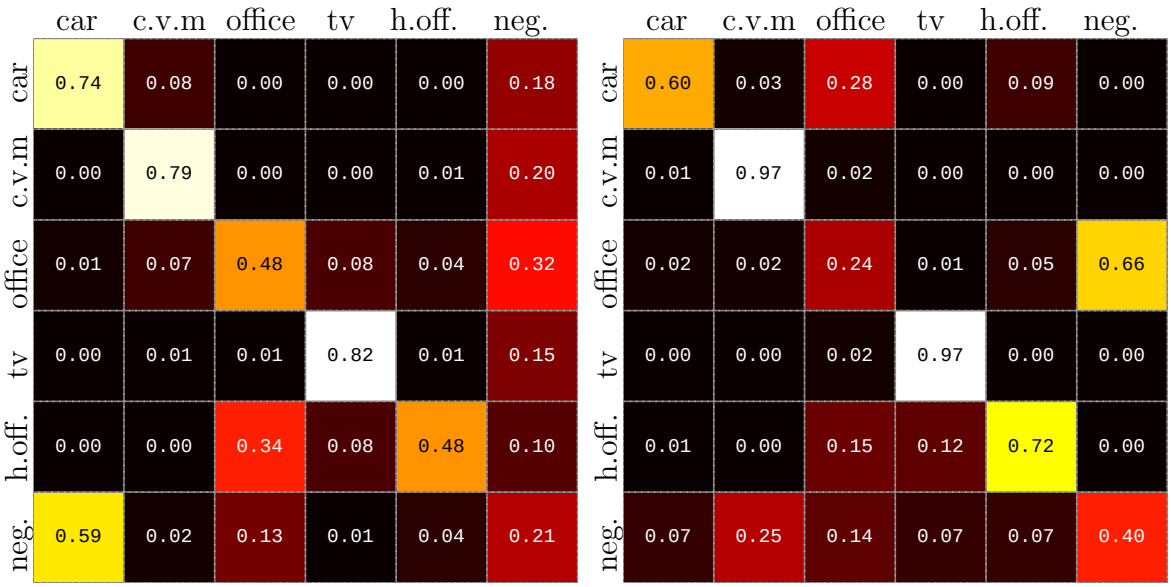


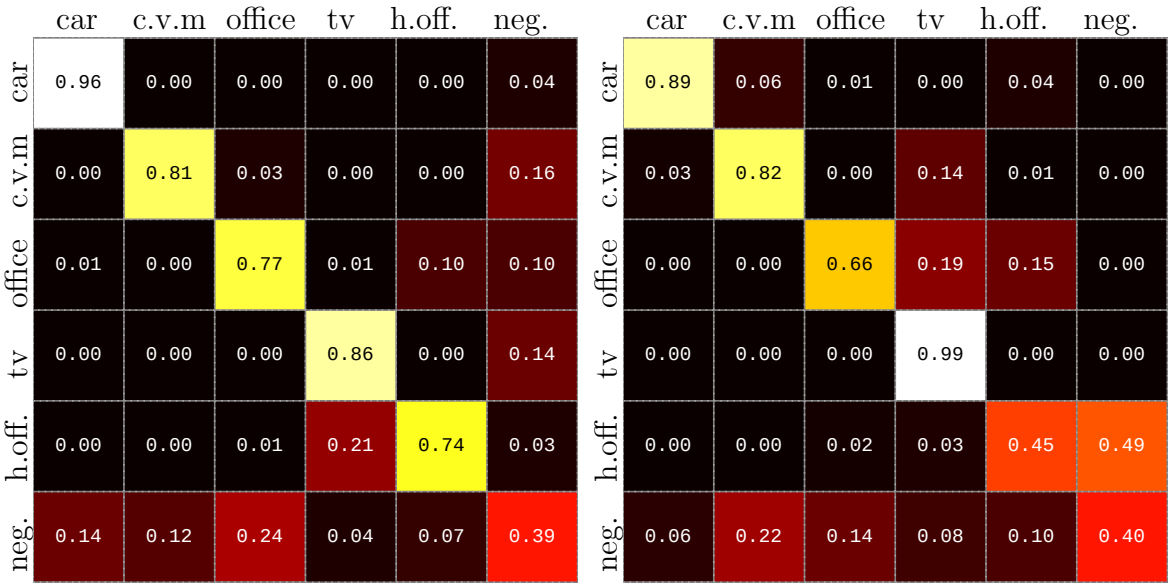
Figure 2.7: Minimum, average and maximum accuracies per device related to experiments performed on the 5-LOCATIONS dataset. As can be noted, all the statistics are higher for the LX2W-related experiments. This suggests that the task of recognizing personal locations is easier for images acquired using such device.

indicate that the LX2W camera is the most appropriate (among the ones we tested) for modelling the personal locations of the user. The success of such camera is probably due to the combination of the large FOV and the wearing modality, which allows to gather the data from a point of view very alike to the one of the user. Indeed, the LX3 camera, which has a similar FOV, but is worn differently, achieve the top-2 average and maximum results.

We conclude our analysis reporting the confusion matrices (Figure 2.8) and some success/failure examples (Figure 2.9 and Figure 2.10) for the best performing methods with respect to the four considered devices. These are: $[k_1]$ CNN Places205 for the RJ device, $[c_2]$ IFV KS 512 for the LX2P device, $[j_3]$ CNN AlexNet for the LX2W device and $[h_4]$ IFV DS SE 256 for the LX3 device. The confusion matrices reported in Figure 2.8 show that the most part of the error is introduced by the negatives, while there is usually less confusion among the 5 locations, especially in the case of $[j_3]$. This confirms our earlier considerations on the influence on the whole system of the low performances of the one-class component used for the rejection of locations not of interest for the user. It should be noted that a rejection mechanism (implemented in our case by the one-class component - see Section 2.5.1) is crucial



(a) RJ - CNN Places205 [k_1] (b) LX2P - IFV KS 512 [c_2]



(c) LX2W - CNN AlexNet [j_3] (d) LX3 - IFV DS SE 256 [h_4]

Figure 2.8: Confusion matrices of the four the best performing methods on the considered devices. Columns represent the ground truth classes, while rows represent the predicted labels. The original confusion matrices have been row-normalized (i.e., each value has been divided by the sum of all the values in the same row) so that each element on the diagonal represents the per-class True Positive Rate. Each matrix is related to the row of Table 2.6 specified by the identifier in brackets. Please note that all methods have been tested on a balanced test set. The following abbreviations are used: c.v.m - coffee vending machine, h.off - home office, neg. - negatives.

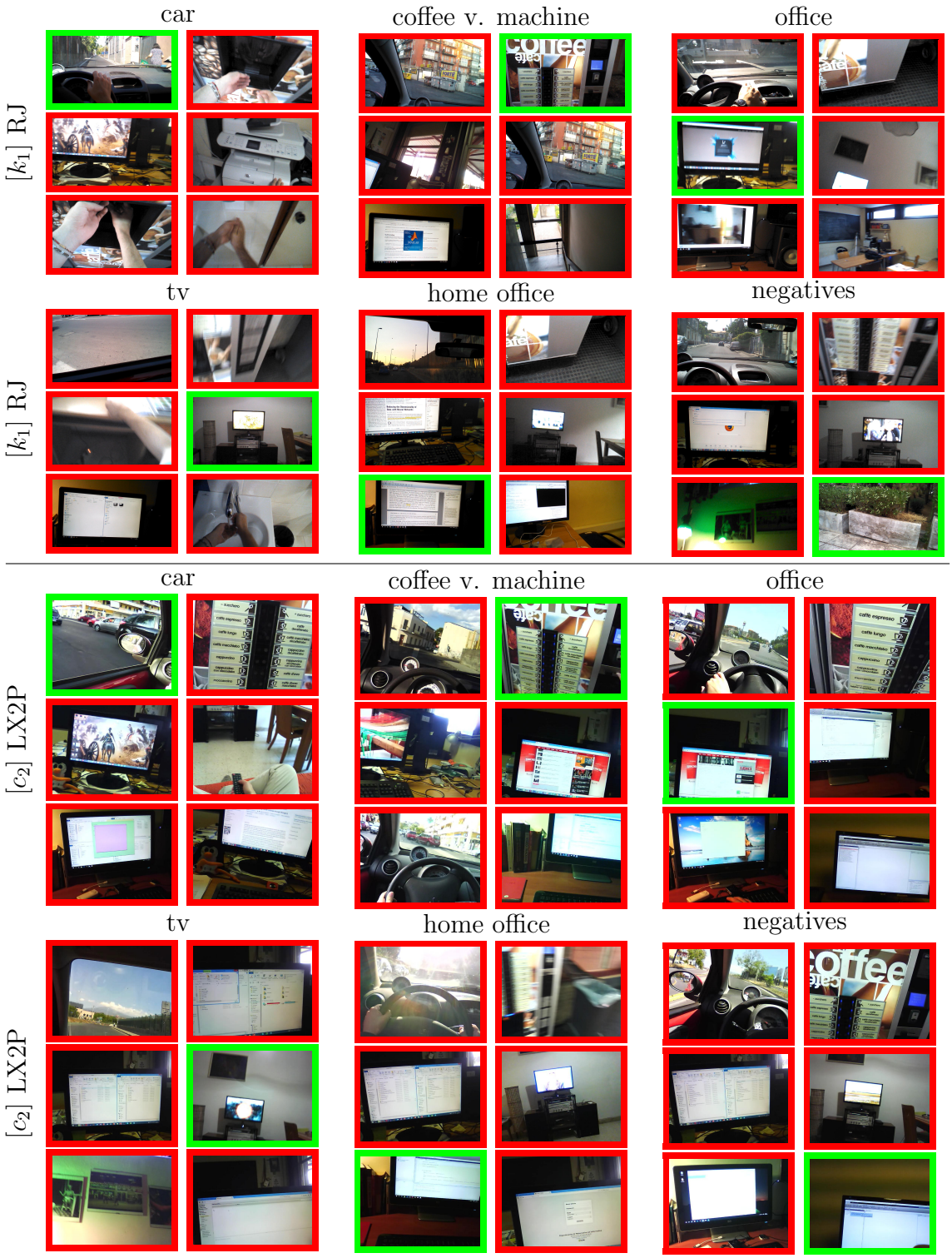


Figure 2.9: Some success (green) and failure (red) examples according to the best performing methods on the RJ and LX2P. Samples belonging to the same class are grouped by columns, while samples related to the same experiment are grouped by rows.

Figure 2.10: Some success (green) and failure (red) examples according to the best performing methods on the LX2W and LX3 devices. Samples belonging to the same class are grouped by columns, while samples related to the same experiment are grouped by rows.

for building effective systems, not only able to discriminate among a small set of known locations, but also able to reject outliers and that building such component can usually rely only on a small number of positive samples with few or no representative negative examples. Moreover, there is usually some degree of confusion between the office, home office and TV classes. This is not surprising, since all these classes are characterized by the presence of similar objects (e.g., a screen) and by similar user-location interaction paradigms. Such considerations suggest that discrimination among similar locations should be considered as a fine-grade problem and that the considered task could probably benefit from coarse-to-fine classification paradigms. All the considerations above are more evident looking at the samples reported in Figure 2.9 and Figure 2.10.

2.5.5 Discussion

The aim of this benchmark was to assess the performances of many state-of-the-art representations and acquisition devices on the task of recognizing personal locations of interest for the user. All experiments have been conducted on a dataset of 5 personal locations using 4 different devices. This dataset is available online for research purposes. The results revealed that, while the discrimination among a limited number of personal locations is an easier task, detecting the negative samples, which is a required step in real applications, is a hard one. The best results have been achieved considering deep representations and a wide angular, ear mounted wearable camera (LX2W). This highlights that the considered task can effectively take advantage of the transfer learning properties of CNNs and that wide FOV, head mounted cameras are the most appropriate to model the user's personal locations. Moreover, despite the good performances of the multiclass component, there is still some degree of confusion among personal locations belonging to the same, or similar categories (e.g., office, home office, tv). This suggests that better performances could be achieved fine-tuning the CNN-based representation to the required instance-level granularity.

2.6 Entropy-Based Negative Rejection and 8 Personal Locations

To overcome the main limitations discussed in the prevision Section, we have extend our analysis in the following ways:

1. the proposed dataset (5-LOCATIONS) has been augmented to 8 personal locations by introducing about 7 hours of new video (8-LOCATIONS dataset);

2. an entropy-based negative rejection method exploiting temporal coherence of neighboring predictions is proposed. The proposed method is compared to the baseline pipeline discussed in the previous benchmark;

3. fine-tuned CNNs have been considered in the analysis and are compared to models based on off-the-shelf CNN features.

We consider a classification pipeline similar to the baseline classification pipeline discussed in Section 2.5.1 and depicted in Figure 2.6. The pipeline is made up of two main components: 1) a multi-class location classifier, and 2) a mechanism for rejecting negative samples. The multi-class component is implemented using a number of standard supervised learning techniques (e.g., an SVM classifier or a fine-tuned CNN). In order to tackle negative rejection, we propose an entropy-based negative rejection mechanism which leverages the temporal coherence of class predictions within a small temporal window. The input to our system is a small sequence of neighboring frames. For each frame, the multi-class classifier estimates a posterior probability distribution on the considered personal locations. Posterior probabilities are hence smoothed to perform multi-class classification on the input sequence. The input sequence is either classified as a given location or rejected depending on how much the different predictions agree. The proposed pipeline is depicted in Figure 2.11 and detailed in the following.

We assume that very close frames in an egocentric video (e.g., less than 0.5 seconds apart) share the same class. This assumption is of course imprecise whenever there is a transition from a given location to another. This phenomenon however mostly affects the accuracy related to the localization of the exact transition frame between two different locations and it does not impact much (in average) the overall

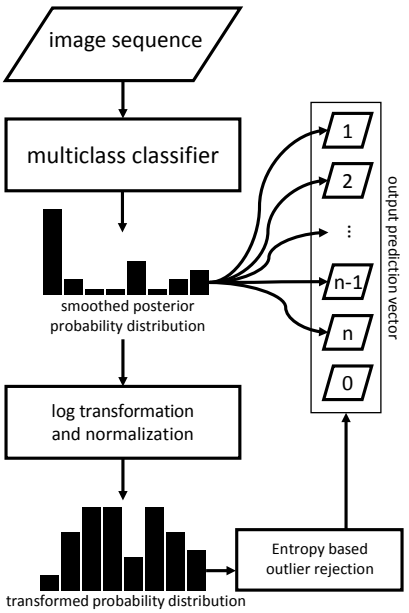


Figure 2.11: The proposed classification pipeline combining a multi-class classifier and an entropy-based negative rejection method.

recognition performances. According to this assumption, n subsequent observations $x_1, \ldots, x_n$ share the same class c. As it is depicted in Figure 2.12, this implies the conditional independence between the observations given class c:

$$x_i \perp\!\!\!\perp x_j | c, \ \forall i, j \in \{1, 2, \ldots, n\}. \tag{2.2}$$

Given the property reported in Equation (2.2), the posterior probability $p(c_k|x_1, \ldots, x_n)$ for the generic class c_k, can be expressed as:

$$\begin{aligned} p(c_k|x_1, \cdots, x_n) &= \frac{p(x_1, \cdots, x_n|c_k) \cdot p(c_k)}{p(x_1, \cdots, x_n)} = \\ &= \prod_{1 \leq i \leq n} p(x_i|c_k) \frac{p(c_k)}{p(x_1, \ldots, x_n)} = \\ &= \prod_{1 \leq i \leq n} \frac{p(c_k|x_i)p(x_i)}{p(c_k)} \frac{p(c_k)}{p(x_1, \cdots, x_n)} = \\ &= \prod_{1 \leq i \leq n} p(c_k|x_i) \frac{p(c_k)^{1-n} \prod_{1 \leq i \leq n} p(x_i)}{p(x_1, \cdots, x_n)}. \end{aligned} \tag{2.3}$$

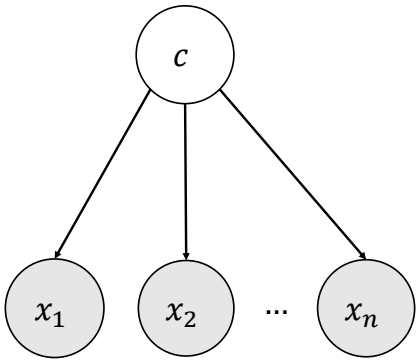


Figure 2.12: A graphical model depicting the conditional independence of a small number of subsequent frames $x_1, \ldots, x_n$, given their class c.

If we assume that all the considered locations of interest have equal probabilities $p(c_k) = \frac{1}{K}, \forall k \in \{1, \cdots, K\}$ (with K being the total number of classes), then Equation (2.3) simplifies to:

$$p(c_k|x_1, \cdots, x_n) = \frac{\prod_i p(c_k|x_i)}{\sum_k \prod_i p(c_k|x_i)} \tag{2.4}$$

where $p(c|x_i)$ denotes the posterior probability distribution on class c estimated by the multi-class classifier, given observation x_i.

Equation (2.4) is used to smooth the predictions of the multi-class classifier on multiple, contiguous frames of the input sequence for which we assume conditional independence as reported in Eq (2.2). The predicted class for the input sequence is determined as the one which maximizes the probability reported in Equation (2.4). When the samples are positive and hence they belong to a given class, we expect Equation (2.4) to produce a resulting posterior distribution which strongly agrees on the identity of the considered samples. On the contrary, when the sequence contains negative samples, we expect the resulting posterior distribution to exhibit a high degree of uncertainty. We propose to measure the uncertainty of the distribution reported in Equation (2.4) (i.e., entropy) to quantify the "outlierness" of the considered samples. Given a posterior distribution p, we measure the uncertainty as the entropy:

$$e(p; x_1, \cdots, x_n) = -\sum_k p(c_k|x_1, ..., x_k) log\big(p(c_k|x_1, ..., x_k)\big). \tag{2.5}$$

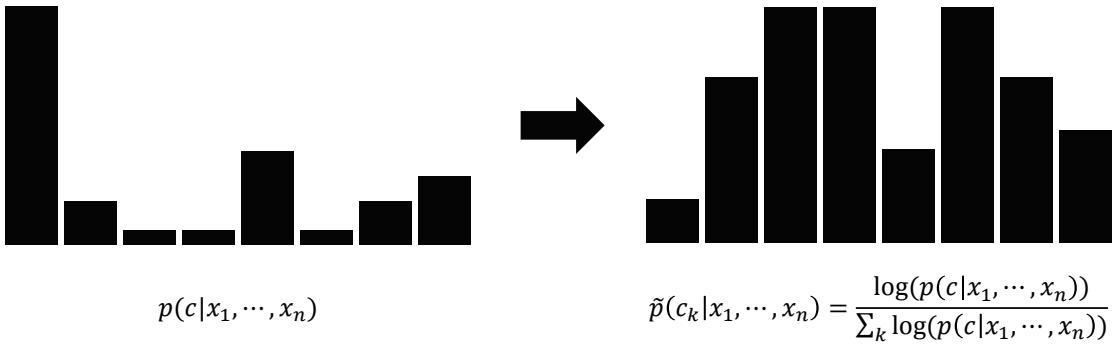


Figure 2.13: A visual example of the transformation operated by Equation (2.6).

The entropy reported in Equation (2.5) can be used to discriminate negative sequences (i.e., locations not of interest for the user) from positive ones using a threshold t_e. Sequences are classified as negative if $e(p; x_1, \cdots, x_n) > t_e$, whereas they are classified as positive if $e(p; x_1, \cdots, x_n) \leq t_e$. The optimal threshold t_e can be selected as the one which of best separates the training set from a small number of negatives used for optimization purposes.

In practice, instead of measuring the uncertainty directly from the distribution reported in Equation (2.4), we log-transform the original distribution p as follows:

$$\tilde{p}(c_k|x_1, \cdots, x_k) = \frac{log(p(c_k|x_1, \cdots, x_k))}{\sum_k log(p(c_k|x_1, \cdots, x_k))}. \tag{2.6}$$

The proposed transformation has the effect of "inverting" the degree of uncertainty carried by the distribution. Therefore, negative samples will be characterized by a high $e(p; x_1, \cdots, x_n)$ value and a low $e(\tilde{p}; x_1, \cdots, x_n)$ value. Figure 2.13 depicts a visual example of such transformation. In Section 2.6.2, we experimentally show that working with the log-transformed distribution shown in Equation (2.6), allows to compute the separation threshold t_e from the training/optimization-negatives set in a more robust way.

Please note that the maximum length n of the input sequence in our system should be carefully selected. Indeed, too small values would cause the rejection mechanism to fail for lack of data, while excessively large values would break the assumption reported in Equation (2.2) and would greatly affect the localization of the transition frame between two different locations.

2.6.1 Representations

Similarly to what done in our preliminary analysis, we consider three categories of image representations: holistic, shallow and deep representations. In particular, we consider the same representations for the holistic and shallow categories, i.e., GIST and IFV computed on dense SIFT descriptors. Such representations are extracted with the same modalities and parameters as the ones discussed in Section 2.5.2.

We update the considered representations including more recent CNN architectures and exploitation modalities. In particular, we consider two popular CNN architectures and two different transfer learning approaches. The considered architectures are AlexNet [98] and VGG16 [101]. Such models have been pre-trained by their authors on the ImageNet dataset [102] to discriminate among 1000 object categories. We also consider two models proposed by Zhou et al. [92], who train the same CNN architectures (AlexNet and VGG16) on the Places205 dataset, which contains images from 205 different place categories. Considering four different models allow us to assess the influence of both the network architectures (AlexNet and VGG16) and the original training data (ImageNet and Places205) in our transfer learning experiments. The considered transfer learning approaches are the following: extracting the feature representation contained in the penultimate layer of the network and reusing it in a classifier (e.g., SVM), and fine-tuning the pre-trained network with new data and labels

Reuse of pre-trained CNNs

We obtain the deep feature representations extracting the values contained in the penultimate layer of the network when the input image, appropriately rescaled to the dimensions of the data layer, is propagated into the network. Such feature representation is the one contained in the hidden layer of the multilayer perceptron in the terminal part of the network. For all the considered CNN models, these representations are compact 4096-dimensional vectors.

Fine-tuning of pre-trained CNNs

The pre-trained network is fine-tuned using the data contained in the training set. Fine-tuning is performed substituting the last layer of the network (the one carrying

the final probabilities) with a new layer containing 8 units (one per each personal location to be recognized) which is initialized with random weights. The training set is divided into two parts: 85% for training and 15% for validation. Optimization of the network is resumed starting from the pre-trained weights. We set a larger learning rate for the randomly initialized layer, and a smaller learning rate for pre-learned layers. The training procedure is stopped when a high validation accuracy is reached or when it is not able to grow any more and the model with maximum validation accuracy is selected. In this case the networks are not used to explicitly extract the representation but directly to predict posterior probabilities.

2.6.2 Experimental Settings

Experiments are performed on the 8-LOCATIONS dataset. The experiments aim at assessing the performances of the classification pipeline including the proposed negative rejection method. The proposed classification method will be compared with respect to the baseline classification discussed in the previous benchmark (Section 2.5.1). Jointly, we extend our benchmark to new CNN architectures and transfer learning methods, as well as to the larger dataset on 8 personal locations.

We adopt experimental settings conform to the ones adopted in the previous analysis (Section 2.5.3). Specifically, all experiments are performed considering different combinations of device and representations. The considered classification pipelines and all related parameters are independently trained and tested on the training/testing sets related to the different devices. In the following, we discuss the experiments designed to assess the performances of the considered representations with respect to 1) the overall location recognition system, 2) the negative rejection mechanism alone, and 3) the multi-class classifier alone.

Overall Personal Location Recognition System

The performances of the overall system are assessed considering the proposed classification pipeline and the baseline considered in the previous benchmark. When the proposed method including the entropy-based negative rejection mechanism is considered (Figure 2.11), the short sequences of 15 subsequent frames included in the dataset are used as inputs. Posterior probabilities estimated by the multi-class component for each of the 15 input frames are smoothed using Eq (2.4). The

smoothed posterior probability is used to reject the input sequence or classify it among the different locations. When the baseline classification pipeline proposed in [24] is considered (Figure 2.11), the first image of each sequence is used as input. Input frames are whether rejected by the one-class classifier or discriminated into the positive classes by the multi-class classifier.

Rejection of Negative Samples

Rejection of negative samples is known as a hard problem and it can be tackled in different ways. Since all our experiments are performed on unbalanced datasets (the number of positive samples is larger than the number of negative ones – see Section A.3), we don't use the accuracy to assess the performances of the methods under analysis. When the number of negative samples is low with respect to the positives one, a method with a high True Positive Rate (TPR) and a low True Negative Rate (TNR) still retains a high accuracy. Therefore, the performances of the proposed methods are assessed using the True Average Rate (average between the TPR and the TNR) defined in Equation (2.1).

The optimization procedure of the one-class SVM classifier involved in the benchmark classification pipeline discussed in Section 2.5.1 depends on a single parameter ν which is a lower bound on the fraction of outliers in the training set. We train the one-class component considering all the positive samples (the entire training set) and use the optimization negatives to choose the value of ν which maximizes the TAR value on the set of training samples plus optimization negatives. It should be noted that the classifier is learned solely from positive data, while the small amount of negatives is only used to optimize the value of the ν hyperparameter.

Entropy-Based Rejection Option

We apply the proposed entropy-based rejection method to discriminate negative from positive samples. For the experiments, we consider the short sequences of 15 subsequent frames contained in the proposed dataset. It should be noted that, given the standard rate of 30 fps, the length of each sequence is $0.5s$ long and hence the conditional independence assumption reported in Equation (2.2) of Section 2.6 is satisfied. For each experiment, we choose t_e as the threshold which best separates the training set from the optimization negative samples included in the dataset. All

thresholds are computed independently for each experiment (i.e., for each device-representation combination). Since the training set does not comprise 15-frames sequences, no temporal smoothing is performed on the training predictions and entropy is measured on the posterior probabilities predicted for each training sample.

In Section 2.6 we proposed to log-transform the smoothed posterior distribution (Equation (2.6)) in order to compute the entropy-based score (Equation (2.5)) used for negative rejection. To show that the considered log-transformation helps finding threshold t_e more reliably, in Figure 2.14 we report the Threshold-TAR curves for some representative experiments. The curves plot thresholds t_e against the True Average Rate (TAR) scores obtained using such thresholds. The depicted curves are used to effectively find the best discrimination threshold t_e (i.e., the x-value corresponding to the curve peak). The figure reports the curves computed on the training sets plus optimization negatives, as well as the ones computed on the test sets. As can be noted, the curves computed using the log-transformation are almost totally overlapped, while there is far less overlap between the curves computed avoiding the log-transformation. To assess the robustness of the estimated thresholds, we also report the True Average Rate (TAR) results for all performed experiments in Figure 2.15. The figure compares results obtained using the proposed method (i.e., thresholds t_e are computed from the training/optimization-negatives set) to those obtained with the optimal threshold computed directly on the test set using the ground truth labels. The average absolute difference between obtained and optimal results amounts to 0.06.

2.6.3 Multiclass Discrimination

To assess the performances of the considered representations with respect to the task of discriminating among the 8 personal locations, we train linear SVM classifiers on the training sets and test them on the corresponding test sets. Similarly to [93, 94], the input feature vectors are transformed using the Hellinger's kernel prior to using them in the linear SVM classifier. Differently from [93, 94], we do not apply L2 normalization to the feature vectors, but instead we independently scale each component of the vectors subtracting the minimum and dividing by the difference between the maximum and minimum values. Minima and maxima for each component are computed from the training set and reported on the test set.

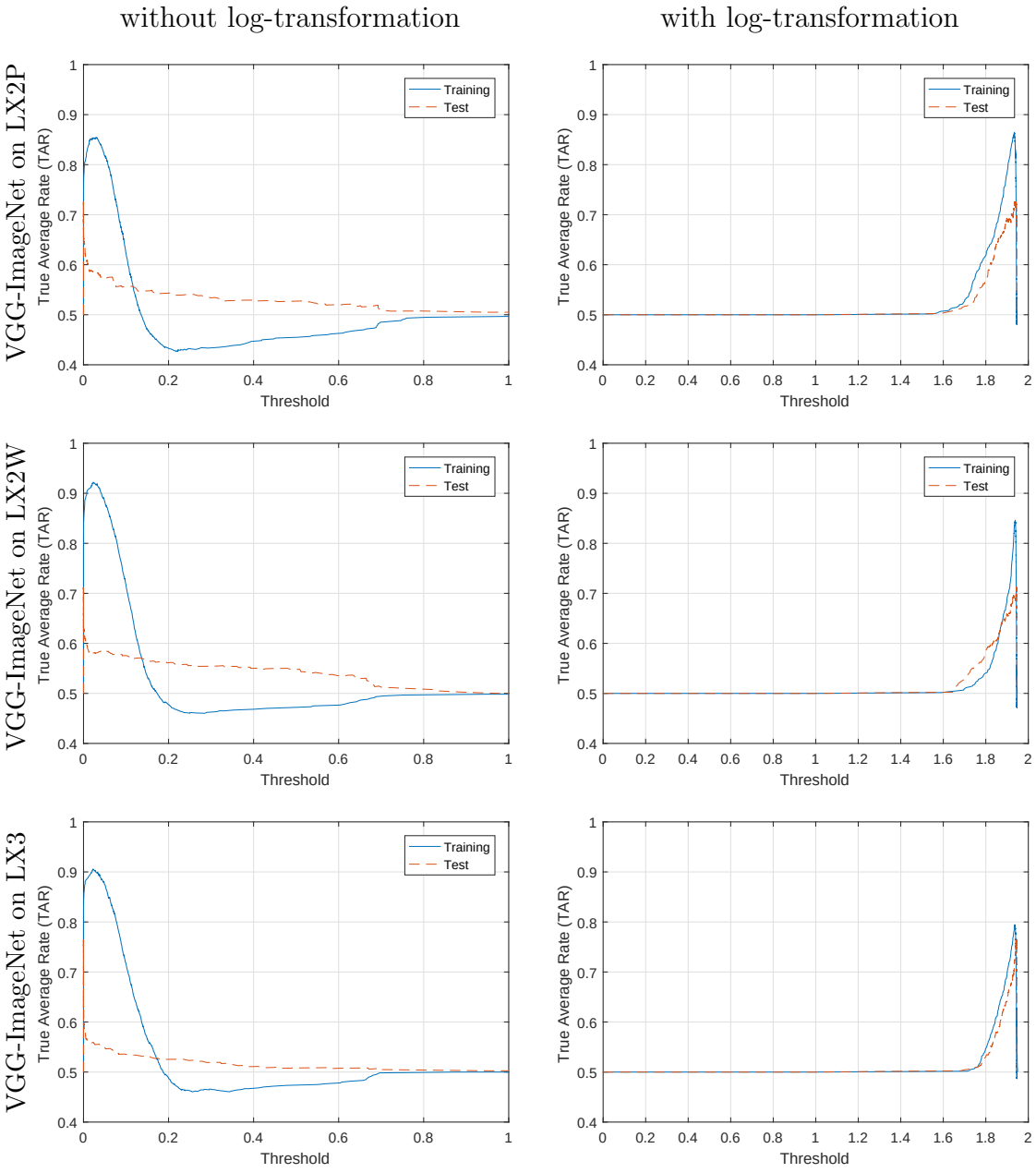


Figure 2.14: Threshold-TAR (True Average Rate) curves obtained without (left) and with (right) log-transformation. All plots are obtained from posterior probabilities estimated by an SVM model trained extracting VGG-ImageNet features from data acquired using three different devices: the LX2P camera (perspective Looxcie LX2 - first row), the LX2W camera (wideangular Looxcie LX2 - second row), and the LX3 device (chest mounted Looxcie LX3 - third row).

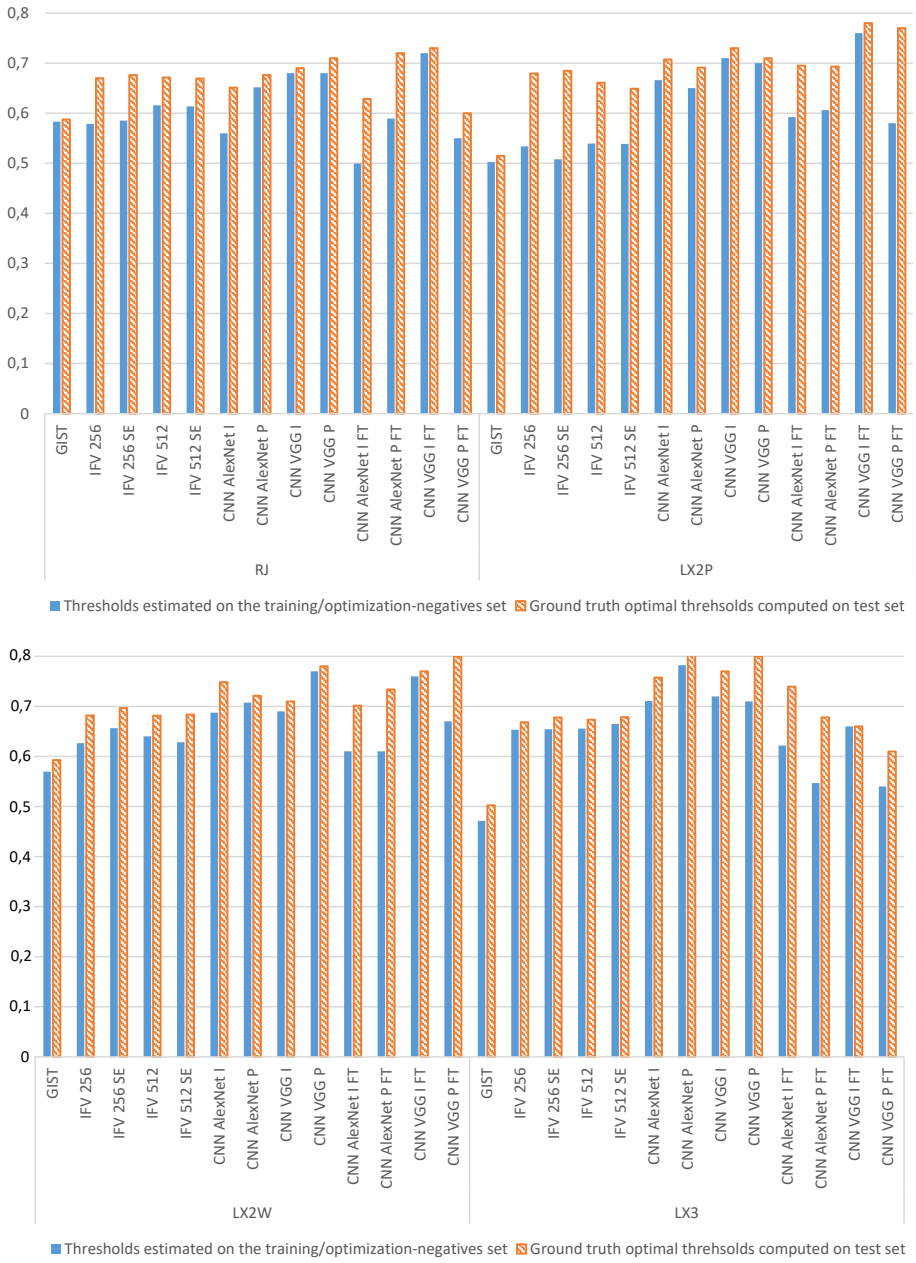


Figure 2.15: True Average Rate (TAR) scores obtained on the test sets considering different combinations of devices and representations. The figure reports results obtained using thresholds computed on the training/optimization-negatives sets. Results obtained using the ground truth optimal thresholds computed on the test set are also reported for reference. As can be noted, estimated thresholds often reach close-to-optimal results. The average absolute difference between obtained and optimal results amounts to 0.06.

The optimization procedure of the linear SVM classifier depends only on the cost parameter C, which is chosen in order to maximize the accuracy on the training set using cross-validation techniques [93, 94]. It should be noted that, in the case of fine-tuning, Convolutional Neural Networks are jointly used for feature extraction and classification. Therefore, in such cases, we do not rely on a SVM classifier for multi-class classification. When fine-tuned models are employed within the baseline pipeline, they are used both to extract features (on top of which the SVM One-Class classifier can be learned) and to directly perform multi-class classification. We would like to emphasize that in our experiments the multi-class classifier is learned using only positive samples.

2.6.4 Experimental Results

In this Section, we report the performances of the overall system implemented according to the two considered pipelines, as well as detailed performances of the discrimination and negative rejection components individually.

Overall System

Table 2.7 reports the accuracies of the overall system according to the proposed method and the baseline classification pipeline. Each row of the table corresponds to a different experiment and is denoted by a unique identifier in brackets (e.g., $[a_1]$). The first column (**Method**) reports the unique identifier and the representation method used in the experiment. The second column (**Dev.**) reports the device used to acquire the data. The third column (**Options**) reports the options related to the considered representation method. Specifically, in the case of representations based on the Improved Fisher Vectors (IFV), the values 256 or 512 represent the number of centroids used to train the GMMs, while "SE" indicates that the SIFT descriptors have been Spatially Enhanced. In the CNN-related experiments, "I" denotes that the considered model has been pre-trained on the ImageNet dataset, "P" denotes that the considered model has been pre-trained on the Places205 dataset, "FT" indicates that the network has been fine-tuned, while, when no "FT" tag is reported, the pre-trained network is only used to extract the representation vectors. The fourth column (**Dim.**) reports the dimensionality of the feature vectors. The fifth and sixth columns report the accuracies of the model according to the two compared

methods. To improve readability, for each method, the maximum accuracies among the experiments related to a given device are reported in **bold numbers**, while the global maximum accuracy is reported in **boxed bold numbers**.

The proposed entropy-based negative rejection method generally allows to obtain better results with respect to the baseline method when deep representations are used. Comparable or worse performances are generally obtained when using other representations. The holistic GIST representation is usually unable to model the personal locations with the appropriate level of detail (compare the methods $[a_1]$, $[a_2]$, $[a_3]$ and $[a_4]$ to others). Improved Fisher Vectors (IFV) generally work better than GIST, but provide inconsistent results in some cases (e.g., $[b_1]$ to $[e_1]$ and $[b_2]$ to $[e_2]$). Using larger codebooks allows to obtain better results in some cases (e.g., when smart glasses Recon Jet (RJ) and narrow-angle ear-mounted LX2P camera are used) at the cost of a significantly larger representation ($80k$ vs $40k$ dimensions). The Spatially Enhancement option (SE) does not in general result in significant improvements. The best performances are given by deep representations. Fine-tuning the model often, but not always (e.g., compare $[h_1]$ to $[l_1]$, $[f_3]$ to $[j_3]$ and $[h_4]$ to $[l_4]$) results in a significant performance improvement.

One important fact emerging from the analysis of the results in Table 2.7, consists in the superior performances obtained on the data acquired using the LX2W device. This observation is supported by Figure 2.16, which reports the minimum, maximum and average accuracies of the overall system for all the experiments related to a given device when the proposed method is considered. All three indicators are higher in the case of the LX2W camera, which suggest that, among the ones being tested, such device is the most appropriate for modelling the user's personal location. Such result is probably due to the combination of the large FOV which allows to capture a larger quantity of information and the wearing modality, which enables the acquisition of the data from the user's point of view.

In Figure 2.17 and Figure 2.18-2.19, we report confusion matrices and some success/failure examples (true/false positive) for the best performing methods on each device. All confusion matrices point out how the most part of the error is due to the need to handle negative samples. In fact, most false positive errors are due to the misclassification of negative samples as shown in Figure 2.18-2.19. Moreover, there is usually confusion between pairs of similar looking locations, e.g., Office - Home

				Accuracy	
Method	**Dev.**	**Options**	**Dim.**	**Proposed**	**Baseline**
[a_1] GIST	RJ	—	512	22,44	25,67
[b_1] IFV	RJ	256	40960	25,11	56,39
[c_1] IFV	RJ	256 SE	41984	26,28	58,56
[d_1] IFV	RJ	512	81920	31,67	55,78
[e_1] IFV	RJ	512 SE	83968	31,33	56,61
[f_1] CNN	RJ	AlexNet I	4096	58,11	58,94
[g_1] CNN	RJ	AlexNet P	4096	67,00	62,33
[h_1] CNN	RJ	VGG16 I	4096	71,61	43,83
[i_1] CNN	RJ	VGG16 P	4096	61,17	60,00
[j_1] CNN	RJ	AlexNet I FT	4096	65,94	60,00
[k_1] CNN	RJ	AlexNet P FT	4096	**76,83**	76,72
[l_1] CNN	RJ	VGG16 I FT	4096	64,11	**76,89**
[m_1] CNN	RJ	VGG16 P FT	4096	75,06	70,78
[a_2] GIST	LX2P	—	512	29,44	22,61
[b_2] IFV	LX2P	256	40960	17,50	51,39
[c_2] IFV	LX2P	256 SE	41984	12,56	55,11
[d_2] IFV	LX2P	512	81920	18,50	48,17
[e_2] IFV	LX2P	512 SE	83968	18,00	48,33
[f_2] CNN	LX2P	AlexNet I	4096	70,06	61,28
[g_2] CNN	LX2P	AlexNet P	4096	64,11	49,89
[h_2] CNN	LX2P	VGG16 I	4096	67,28	52,44
[i_2] CNN	LX2P	VGG16 P	4096	63,33	44,83
[j_2] CNN	LX2P	AlexNet I FT	4096	74,83	63,72
[k_2] CNN	LX2P	AlexNet P FT	4096	69,94	72,00
[l_2] CNN	LX2P	VGG16 I FT	4096	68,28	**75,89**
[m_2] CNN	LX2P	VGG16 P FT	4096	**80,06**	70,50
[a_3] GIST	LX2W	—	512	39,83	23,22
[b_3] IFV	LX2W	256	40960	37,50	59,17
[c_3] IFV	LX2W	256 SE	41984	42,83	58,44
[d_3] IFV	LX2W	512	81920	39,50	52,06
[e_3] IFV	LX2W	512 SE	83968	37,06	51,50
[f_3] CNN	LX2W	AlexNet I	4096	75,22	65,61
[g_3] CNN	LX2W	AlexNet P	4096	73,89	55,06
[h_3] CNN	LX2W	VGG16 I	4096	70,89	54,06
[i_3] CNN	LX2W	VGG16 P	4096	81,67	50,06
[j_3] CNN	LX2W	AlexNet I FT	4096	73,89	65,44
[k_3] CNN	LX2W	AlexNet P FT	4096	76,22	73,78
[l_3] CNN	LX2W	VGG16 I FT	4096	76,78	73,78
[m_3] CNN	LX2W	VGG16 P FT	4096	**87,28**	**80,11**
[a_4] GIST	LX3	—	512	29,50	29,22
[b_4] IFV	LX3	256	40960	39,94	29,11
[c_4] IFV	LX3	256 SE	41984	40,44	37,00
[d_4] IFV	LX3	512	81920	39,50	27,56
[e_4] IFV	LX3	512 SE	83968	39,89	27,28
[f_4] CNN	LX3	AlexNet I	4096	65,39	51,39
[g_4] CNN	LX3	AlexNet P	4096	**76,50**	55,72
[h_4] CNN	LX3	VGG16 I	4096	73,22	34,17
[i_4] CNN	LX3	VGG16 P	4096	76,11	51,94
[j_4] CNN	LX3	AlexNet I FT	4096	73,06	**66,94**
[k_4] CNN	LX3	AlexNet P FT	4096	67,61	56,28
[l_4] CNN	LX3	VGG16 I FT	4096	61,94	60,65
[m_4] CNN	LX3	VGG16 P FT	4096	71,39	44,00

Table 2.7: Performances of the overall system. Results are related to experiments performed on the 8-LOCATIONS dataset.

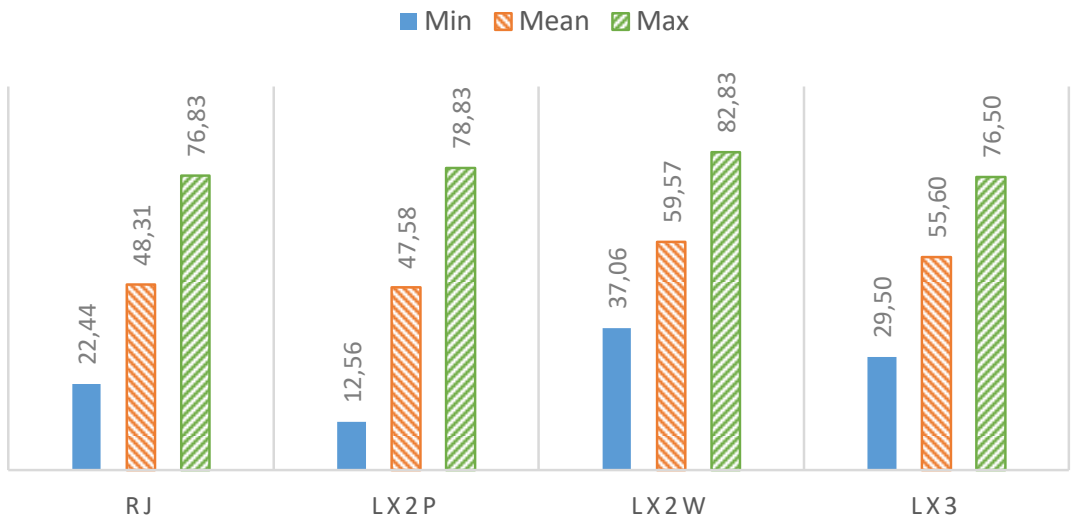
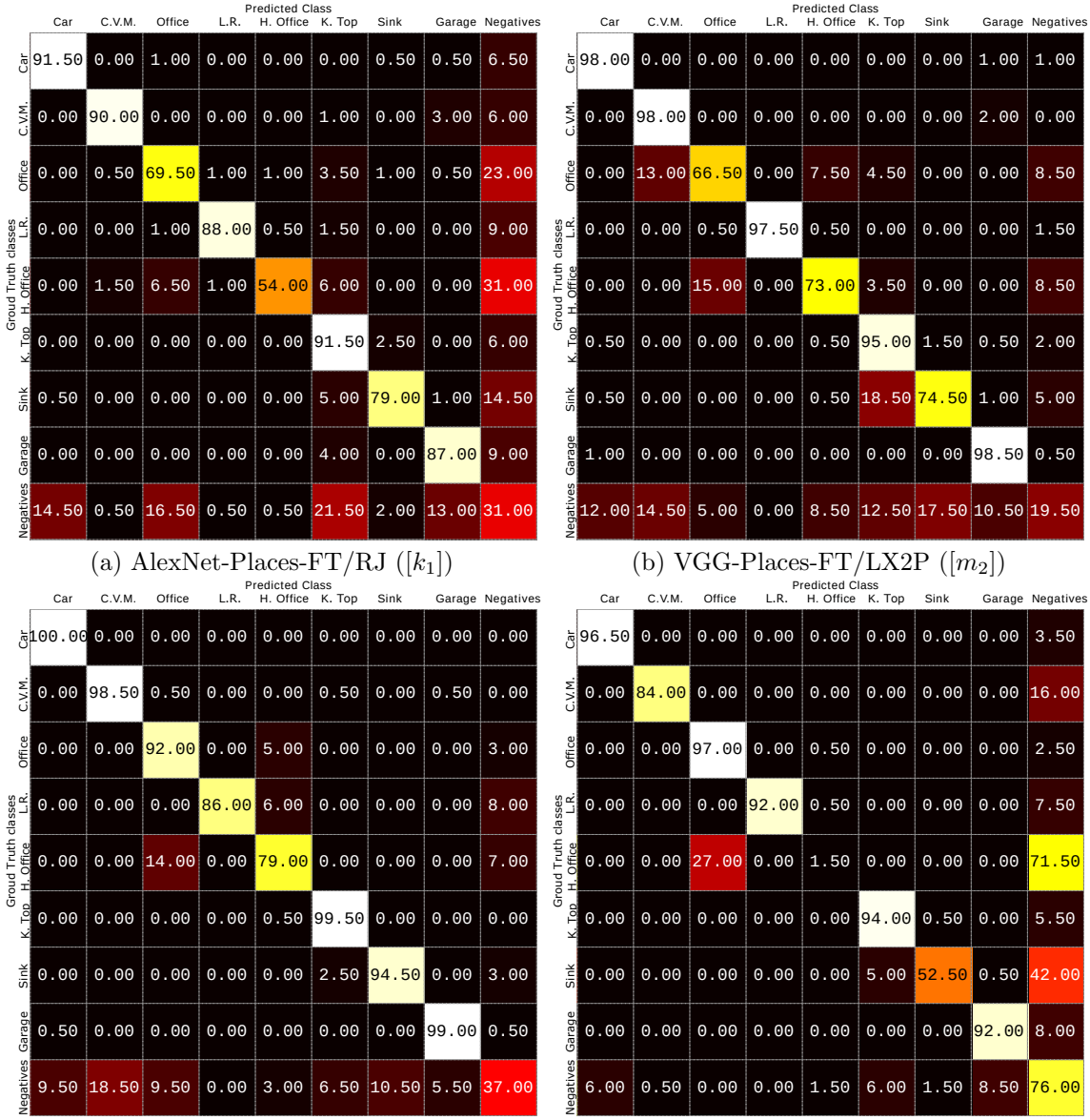


Figure 2.16: Minimum, average and maximum accuracies of the overall system with the different representations per device. Statistics are related to experiments performed on the 8-LOCATIONS dataset. All the statistics are higher for the LX2W-related experiments. This suggests that the task of recognizing personal locations is easier on images acquired using a head mounted, wide-FOV device.

Office, Sink - Kitchen Top, Living Room - Home Office (see Figure 2.18-2.19 for some examples). The confusion matrices shown in Figure 2.17(b) and Figure 2.17(c) use similar models (a fine-tuned VGG16 network pre-trained on the ImageNet dataset) trained on data acquired using similar devices, differing mainly in their Field Of View (FOV): a narrow-angle Looxcie LX2 (LX2P) and a wide-angle Looxcie LX2 (LX2W). This allows to make direct considerations on the influence of the Field Of View (FOV) in the task of detecting locations of interest. In particular, the use of a wide-angle camera (Figure 2.17(b)) allows to acquire a larger portion of the Field Of View, which is useful to reduce the confusion between similar locations (e.g., Sink vs Kitchen Top).

Rejection of Negative Samples

Table 2.8 reports the results related to the two rejection methods considered in our analysis: the proposed Entropy Based method (EB) and the One-Class SVM method proposed in [24] (OCSVM). The table is organized similarly to Table 2.7, except for the performance indicators used in this case. Columns 4 to 6 are related



(a) AlexNet-Places-FT/RJ ($[k_1]$) (b) VGG-Places-FT/LX2P ($[m_2]$)

(c) VGG-Places-FT/LX2W ($[m_3]$) (d) AlexNet-Places/LX3 ($[g_4]$)

Figure 2.17: Confusion matrices of the best performing methods on data acquired by each of the considered devices. Rows represent ground truth classes, while columns represent the predicted labels. Each element of the confusion matrix is normalized by the sum of the elements in the corresponding row. Hence, values along the principal diagonal are class-related true positive rates. Confusion matrices are related to the following methods: (a) AlexNet pre-trained on Places205 and fine-tuned on data acquired using the Recon Jet (RJ) smart glasses, (b) VGG16 pre-trained on Places205 and fine-tuned on data acquired using the ear-mounted perspective Looxcie LX2 camera (LX2P), (c) VGG16 pre-trained on Places205 and fine-tuned on data acquired using the ear-mounted wideangular Looxcie LX2 camera (LX2W), (d) SVM trained on features exacted by AlexNet pre-trained on the Places205 with data acquired using the chest mounted Looxcie LX3 camera. Best seen in color.

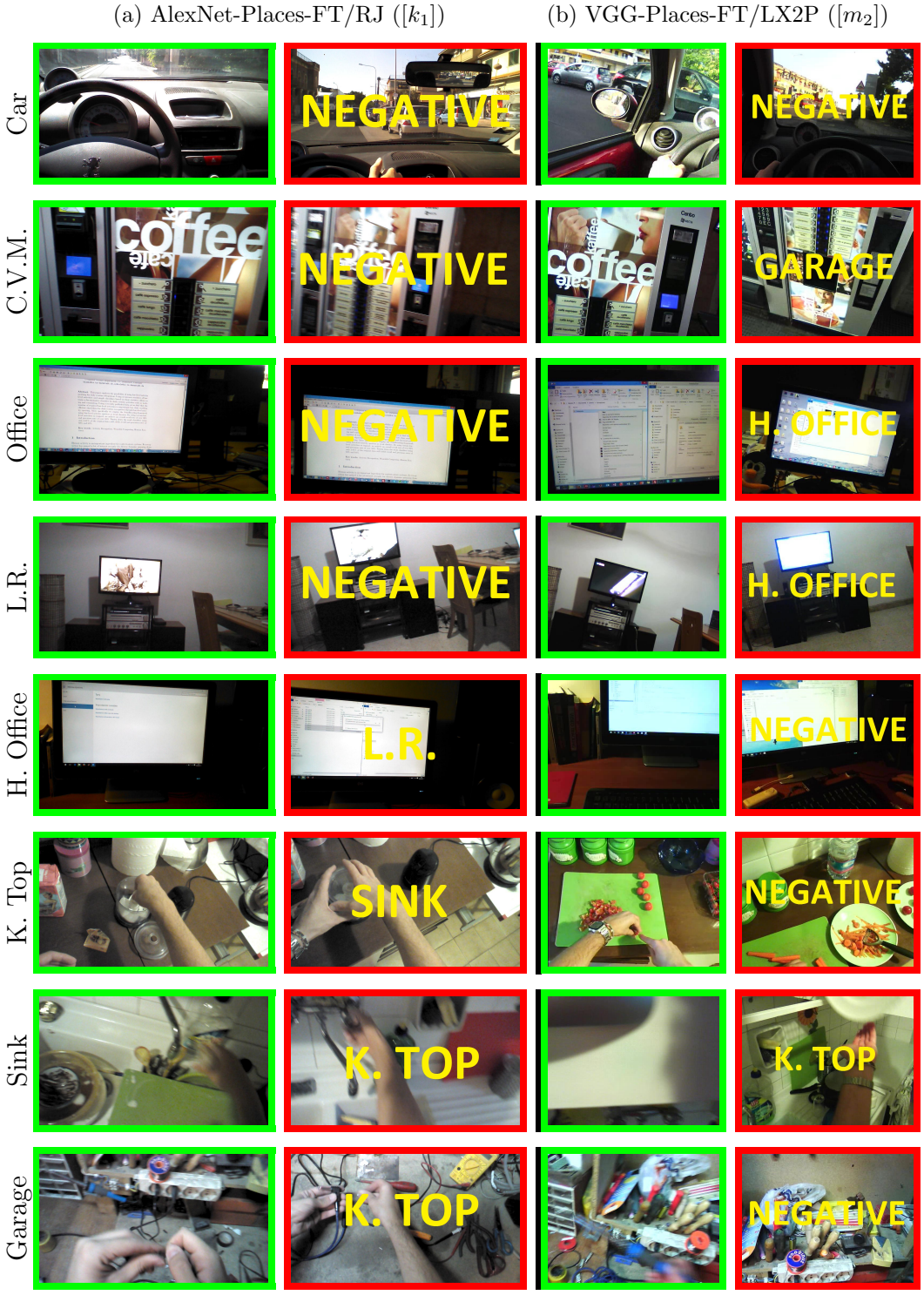


Figure 2.18: True positive (green) and false positive (red) samples related to the best performing methods on the RJ and LX2P devices. Rows represent the ground truth labels, while the predicted label is shown in yellow, in case of a failure. The shown samples are related to the the same methods considered in Figure 2.17. Best seen in color.

Figure 2.19: True positive (green) and false positive (red) samples related to the best performing methods on the LX2W and LX3 devices. Rows represent the ground truth labels, while the predicted label is shown in yellow, in case of a failure. The shown samples are related to the the same methods considered in Figure 2.17. Best seen in color.

to the proposed Entropy-Based method (EB), while columns 7 to 9 are related to the baseline One-Class SVM component (OCSVM). Columns 4 and 7 report the True Average Rate (TAR). Columns $\{5, 8\}$ and $\{7, 9\}$ report respectively the True Positive Rate (TPR) and True Negative Rate (TNR) scores related to the considered methods. The proposed entropy-based method systematically outperforms the one-class SVM baseline, with some exceptions, e.g., the GIST-related methods $[a_2]$, $[a_3]$, $[a_4]$, plus method $[m_2]$. Consistently with the observations made earlier, the best performing methods are in all cases related to the deep representations.

Multiclass Discrimination

Table 2.9 reports the results related to the multi-class discrimination component. It should be noted that, in these experiments, negative rejection is not considered and methods are evaluated ignoring negative samples. The structure of Table 2.9 follows the one of Table 2.7, with the following differences: column 5 reports the accuracy of the multi-class discrimination component when negative samples are removed from the test sets, columns 6 to 13 report the True Positive Rates related to each of the considered classes. It should be noted that the reported results are related to the proposed method and hence they have been obtained using the smoothed posterior probabilities. As noted for Table 2.7, the holistic GIST representations are unable to model the personal locations with the appropriate level of detail. Even if the accuracy values related to the GIST representations are always low, in some cases they are still able to model some classes like for instance Coffee Vending Machine (e.g., $[a_2]$,$[a_3]$ and $[a_4]$), Living Room (e.g., $[a_3]$) and Sink (e.g., $[a_4]$) which are characterized by distinctive spatial layouts. Interestingly, the shallow representations, albeit consistently outperformed by CNN, give remarkable performances in some cases (e.g., $[b_1]$ and $[c_1]$). Using larger codebooks (i.e., 512 centroids in the GMM) does not improve the performances of the IFV-related methods. In fact, in addition to providing a larger representation ($80k$ vs $40k$ dimensions), large codebooks systematically involve worse performances. The Spatially Enhancement option (SE) allows to achieve better performances in some cases (e.g., $[c_1]$ vs $[b_1]$), while it leads to worse performances other cases (e.g., $[c_4]$ vs $[b_4]$). The best performances (bold numbers) are given again by the deep representations. However, in contrast to what one could expect, fine-tuned models do not always outperform the correspondent

			EB			OCSVM		
Method	**Dev.**	**Options**	**TAR**	**TPR**	**TNR**	**TAR**	**TPR**	**TNR**
[a_1] GIST	RJ	—	58,31	37,63	79,00	53,72	55,44	52,00
[b_1] IFV	RJ	256	57,88	15,75	100,00	53,00	70,00	36,00
[c_1] IFV	RJ	256 SE	58,53	17,06	100,00	54,00	72,00	36,00
[d_1] IFV	RJ	512	61,56	23,13	100,00	53,97	71,94	36,00
[e_1] IFV	RJ	512 SE	61,38	22,75	100,00	54,38	71,75	37,00
[f_1] CNN	RJ	AlexNet I	56,00	65,50	46,50	55,06	86,63	23,50
[g_1] CNN	RJ	AlexNet P	65,19	70,88	59,50	53,28	80,56	26,00
[h_1] CNN	RJ	VGG16 I	67,59	73,69	61,50	48,06	46,63	49,50
[i_1] CNN	RJ	VGG16 P	68,44	66,88	70,00	**57,09**	84,19	30,00
[j_1] CNN	RJ	AlexNet I FT	49,97	99,94	00,00	52,53	88,56	16,50
[k_1] CNN	RJ	AlexNet P FT	58,94	86,88	31,00	56,97	96,94	17,00
[l_1] CNN	RJ	VGG16 I FT	**72,31**	62,13	82,50	48,59	96,19	1,00
[m_1] CNN	RJ	VGG16 P FT	54,75	92,00	17,50	49,78	93,56	6,00
[a_2] GIST	LX2P	—	50,25	63,00	37,50	59,16	34,81	83,50
[b_2] IFV	LX2P	256	53,38	07,25	99,50	43,03	71,56	14,50
[c_2] IFV	LX2P	256 SE	50,81	01,63	100,00	43,25	76,00	10,50
[d_2] IFV	LX2P	512	53,94	08,38	99,50	41,94	75,38	08,50
[e_2] IFV	LX2P	512 SE	53,88	07,75	100,00	41,41	74,81	08,00
[f_2] CNN	LX2P	AlexNet I	66,59	74,19	59,00	52,03	75,06	29,00
[g_2] CNN	LX2P	AlexNet P	65,03	71,56	58,50	52,88	57,25	48,50
[h_2] CNN	LX2P	VGG16 I	71,44	67,88	75,00	54,69	59,38	50,00
[i_2] CNN	LX2P	VGG16 P	69,59	70,19	69,00	56,28	56,56	56,00
[j_2] CNN	LX2P	AlexNet I FT	59,22	90,44	28,00	53,09	75,69	30,50
[k_2] CNN	LX2P	AlexNet P FT	60,63	87,75	33,50	54,00	96,50	11,50
[l_2] CNN	LX2P	VGG16 I FT	**76,34**	68,69	84.00	52,50	96,00	9,00
[m_2] CNN	LX2P	VGG16 P FT	58,06	96,63	19,50	**71,94**	80,38	63,50
[a_3] GIST	LX2W	—	56,97	66,44	47,50	64,25	50,50	78,00
[b_3] IFV	LX2W	256	62,66	30,31	95,00	51,47	79,44	23,50
[c_3] IFV	LX2W	256 SE	65,66	36,31	95,00	51,63	79,25	24,00
[d_3] IFV	LX2W	512	64,00	32,50	95,50	47,22	70,94	23,50
[e_3] IFV	LX2W	512 SE	62,84	29,69	96,00	47,25	72,50	22,00
[f_3] CNN	LX2W	AlexNet I	68,75	80,00	57,50	**67,19**	74,38	60,00
[g_3] CNN	LX2W	AlexNet P	70,75	77,00	64,50	57,28	60,56	54,00
[h_3] CNN	LX2W	VGG16 I	68,84	73,69	64,00	63,06	59,13	67,00
[i_3] CNN	LX2W	VGG16 P	**76,97**	84,44	69,50	59,41	50,31	68,50
[j_3] CNN	LX2W	AlexNet I FT	61,03	90,56	31,50	57,22	85,94	28,50
[k_3] CNN	LX2W	AlexNet P FT	61,03	90,56	31,50	62,56	88,63	36,50
[l_3] CNN	LX2W	VGG16 I FT	76,19	77,38	75,00	51,06	86,13	16,00
[m_3] CNN	LX2W	VGG16 P FT	67,16	97,31	37,00	59,03	91,06	27,00
[a_4] GIST	LX3	—	47,13	50,75	43,50	67,16	44,81	89,50
[b_4] IFV	LX3	256	65,34	32,69	98,00	31,94	40,88	23,00
[c_4] IFV	LX3	256 SE	65,44	33,38	97,50	34,59	53,19	16,00
[d_4] IFV	LX3	512	65,56	32,63	98,50	30,75	41,50	20,00
[e_4] IFV	LX3	512 SE	66,50	34,00	99,00	30,44	41,38	19,50
[f_4] CNN	LX3	AlexNet I	71,06	81,63	60,50	54,66	72,81	36,50
[g_4] CNN	LX3	AlexNet P	**78,22**	80,44	76,00	**70,06**	57,13	83,00
[h_4] CNN	LX3	VGG16 I	72,34	85,19	59,50	57,63	37,25	78.00
[i_4] CNN	LX3	VGG16 P	71,06	82,13	60,00	64,50	53,00	76,00
[j_4] CNN	LX3	AlexNet I FT	62,19	92,88	31,50	51,38	90,75	12,00
[k_4] CNN	LX3	AlexNet P FT	54,69	92,88	16,50	52,53	72,06	33,00
[l_4] CNN	LX3	VGG16 I FT	66,22	73,44	59,00	56,59	82,69	30,50
[m_4] CNN	LX3	VGG16 P FT	53,50	93,00	14,00	53,69	53,38	54,00

Table 2.8: Results related to the negative rejection methods. Results are related to experiments performed on the 8-LOCATIONS dataset.

pre-trained networks when they are just used for feature extraction. This is the case of methods $[j_1]$ vs $[f_1]$, $[l_2]$ vs $[h_2]$, $[m_3]$ vs $[i_3]$ and $[m_4]$ vs $[i_4]$. Nevertheless, fine-tuned models significantly outperform their pre-trained counterparts in other cases, e.g., $[k_1]$ vs $[g_1]$, $[m_2]$ vs $[i_2]$, $[l_3]$ vs $[h_3]$ and $[j_4]$ vs $[f_4]$. One of the possible reasons of the difficulty to further improve the internal representation of the networks with respect to the given problem is the availability of very few training data. The considered networks have been fine-tuned on a very small training sets containing about 2000 samples.

2.6.5 Discussion

The experimental results presented in the previous sections highlight the robustness of the proposed negative rejection method with respect to the baseline classification pipeline based on a one-class SVM classifier. Results also show how the considered problem is a challenging one. As discussed earlier, the performances of all the considered methods are dominated by the limits of the negative rejection module, while the multi-class discrimination remains an "easier" sub-task. This suggests that more efforts should be devoted to the design of efficient and robust negative rejection methods. The systematic emergence of deep representations as the best performing methods, not only indicates the higher representational power of such methods, but also suggests that the considered problem can take great advantage of transfer learning techniques. All the CNN-based representations have been obtained using models pre-trained on a large number of images, which compensates for the scarce quantity of training data assumed in this study. As already pointed out in our previous analysis, the LX2W device is the one collecting the highest performance indicators. This suggests that head-mounted wide-angular cameras are probably the best option when modeling the user's location. This is not surprising since such a configuration allows to better replicate the user's point of view and provides a FOV similar to the one characterizing the human visual system.

2.7 Temporal Coherence

In the previous Sections, we have benchmarked the main image representation techniques and acquisition devices on the problem of recognizing personal locations.

Method	**Dev.**	**Options**	**Acc**	**Car**	**C.V.M.**	**Office**	**L.R.**	**H. Office**	**K. Top**	**Sink**	**Garage**
[a_1] GIST	RJ	—	37,56	62,86	98,59	48,65	0,00	32,79	48,72	25,00	29,06
[b_1] IFV	RJ	256	80,13	96,43	85,37	88,38	100,00	97,09	59,86	95,00	59,70
[c_1] IFV	RJ	256 SE	82,94	95,50	88,83	89,23	100,00	97,86	68,42	95,83	59,70
[d_1] IFV	RJ	512	75,94	90,28	97,16	95,24	100,00	96,36	44,78	95,00	58,48
[e_1] IFV	RJ	512 SE	77,44	88,69	97,75	95,81	100,00	96,92	48,10	94,26	59,17
[f_1] CNN	RJ	AlexNet I	76,19	86,30	97,92	57,10	100,00	43,10	71,56	91,11	83,76
[g_1] CNN	RJ	AlexNet P	85,13	90,41	98,92	84,17	100,00	57,45	82,71	95,16	95,00
[h_1] CNN	RJ	VGG16 I	**93,50**	100,00	99,49	97,37	100,00	81,48	84,00	97,24	94,03
[i_1] CNN	RJ	VGG16 P	76,25	97,11	88,73	45,48	97,25	35,23	74,34	86,78	83,33
[j_1] CNN	RJ	AlexNet I FT	74,19	89,53	97,86	72,73	97,03	60,43	84,47	94,59	47,71
[k_1] CNN	RJ	AlexNet P FT	88,81	99,47	96,41	81,54	93,53	93,63	70,11	93,62	90,64
[l_1] CNN	RJ	VGG16 I FT	90,00	74,19	76,92	100,00	98,45	97,50	87,00	95,24	96,08
[m_1] CNN	RJ	VGG16 P FT	85,06	69,77	59,63	98,96	89,67	79,28	96,94	95,38	99,43
[a_2] GIST	LX2P	—	42,69	42,25	99,29	17,18	65,28	44,81	37,41	83,33	24,22
[b_2] IFV	LX2P	256	73,63	60,75	100,00	32,56	100,00	69,23	70,22	100,00	85,41
[c_2] IFV	LX2P	256 SE	74,44	64,47	100,00	36,28	100,00	60,14	72,97	100,00	86,15
[d_2] IFV	LX2P	512	67,81	63,79	100,00	29,70	100,00	71,17	50,52	100,00	75,09
[e_2] IFV	LX2P	512 SE	68,44	67,73	100,00	37,21	100,00	81,73	46,30	100,00	76,25
[f_2] CNN	LX2P	AlexNet I	87,69	91,55	84,75	73,91	100,00	77,88	81,20	94,25	97,52
[g_2] CNN	LX2P	AlexNet P	83,19	97,50	99,01	51,17	100,00	92,42	73,41	95,97	93,63
[h_2] CNN	LX2P	VGG16 I	**90,00**	100,00	96,14	72,17	100,00	83,03	77,65	99,34	98,98
[i_2] CNN	LX2P	VGG16 P	76,50	99,40	86,28	51,55	100,00	54,55	58,21	96,33	86.57
[j_2] CNN	LX2P	AlexNet I FT	84,75	98,92	73,80	67,72	99,00	68,72	80,52	95,21	97,92
[k_2] CNN	LX2P	AlexNet P FT	81,38	94,97	90,23	52,65	100,00	97,44	89,14	98,31	67,92
[l_2] CNN	LX2P	VGG16 I FT	88,06	96,80	77,27	100,00	92,23	98,02	62,50	93,88	100,00
[m_2] CNN	LX2P	VGG16 P FT	88,94	86,86	77,78	100,00	96,84	95,17	87,11	74,71	97,5
[a_3] GIST	LX2W	—	51,75	57,97	94,74	48,36	93,48	30,32	33,95	92,50	31,48
[b_3] IFV	LX2W	256	73,94	52,36	100,00	100,00	100,00	83,33	53,50	100,00	81,97
[c_3] IFV	LX2W	256 SE	74,06	53,76	100,00	97,50	100,00	83,78	53,04	100,00	80,97
[d_3] IFV	LX2W	512	69,69	46,51	100,00	100,00	100,00	87,70	53,11	100,00	72,99
[e_3] IFV	LX2W	512 SE	68,94	46,08	100,00	100,00	100,00	94,92	51,49	100,00	71,94
[f_3] CNN	LX2W	AlexNet I	90,31	97,99	100,00	85,71	100,00	84,24	70,18	100,00	98,99
[g_3] CNN	LX2W	AlexNet P	90,75	99,49	100,00	73,78	100,00	91,67	76,63	100,00	98,01
[h_3] CNN	LX2W	VGG16 I	90,06	99,50	100,00	99,44	100,00	80,00	65,15	100,00	100,00
[i_3] CNN	LX2W	VGG16 P	**95,44**	99,49	99,00	85,97	100,00	96,05	89,45	100,00	95,67
[j_3] CNN	LX2W	AlexNet I FT	83,19	100,00	97,04	55,87	98,30	74,83	71,48	100,00	90,09
[k_3] CNN	LX2W	AlexNet P FT	86,94	100,00	100,00	64,19	100,00	100,00	72,10	96,90	92,13
[l_3] CNN	LX2W	VGG16 I FT	94,81	99,01	100,00	100,00	100,00	96,62	97,55	73,86	100,00
[m_3] CNN	LX2W	VGG16 P FT	94,88	83,76	83,48	100,00	100,00	99,50	99,49	95,22	99,50
[a_4] GIST	LX3	—	46,31	42,41	84,07	35,56	45,60	33,33	56,05	83,52	23,26
[b_4] IFV	LX3	256	69,31	100,00	100,00	85,39	100,00	100,00	31,34	96,85	83,97
[c_4] IFV	LX3	256 SE	68,31	100,00	100,00	81,52	100,00	100,00	31,24	97,69	80,24
[d_4] IFV	LX3	512	62,88	100,00	100,00	100,00	100,00	100,00	26,75	97,76	81,22
[e_4] IFV	LX3	512 SE	63,00	100,00	100,00	100,00	100,00	100,00	26,76	98,47	80,57
[f_4] CNN	LX3	AlexNet I	76,38	99,49	100,00	52,53	100,00	6,67	60,91	90,91	94,71
[g_4] CNN	LX3	AlexNet P	85,44	100,00	100,00	54,87	97,56	96,97	79,28	98,17	95,05
[h_4] CNN	LX3	VGG16 I	84,38	100,00	100,00	51,71	100,00	80,00	78,26	98,27	97,83
[i_4] CNN	LX3	VGG16 P	**87,63**	100,00	99,48	62,26	91,28	96,77	88,21	92,93	92,02
[j_4] CNN	LX3	AlexNet I FT	81,88	100,00	100,00	49,75	98,96	0,00	90,29	90,82	80,25
[k_4] CNN	LX3	AlexNet P FT	77,38	100,00	100,00	49,01	100,00	75,00	63,38	96,46	86,92
[l_4] CNN	LX3	VGG16 I FT	81,56	18,92	50,89	100,00	97,16	92,09	100,00	79,68	100,00
[m_4] CNN	LX3	VGG16 P FT	81,81	60,00	49,62	99,44	97,55	93,75	100,00	76,89	97,04

Table 2.9: Results related to the multi-class component. Results are related to experiments performed on the 8-LOCATIONS dataset.

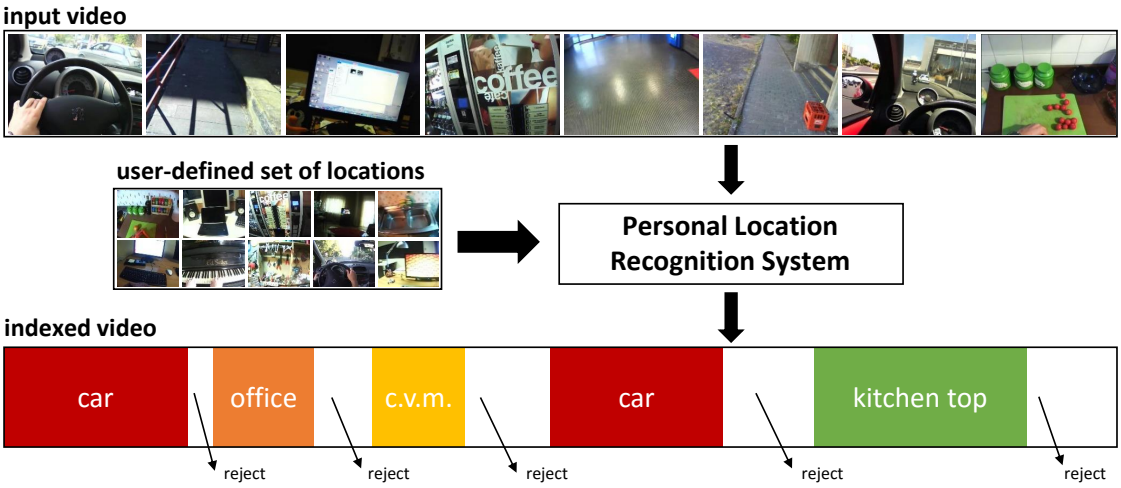


Figure 2.20: Overall schema of the proposed method.

Since the rejection of negative locations is one of the main challenges for the considered task, we have investigated a negative rejection method based on the entropy of neighboring predictions. When an egocentric video is to be analyzed, temporal coherence can be further exploited. Depending on the considered goal, long egocentric videos tend to contain much uninformative content like, for instance, transiting through a corridor, walking, or driving to the office. Therefore, as pointed out in [40], automated tools are needed to enable faster access to the information stored in such videos and index their visual content. Towards this direction, researches have investigated methods to produce short informative video summaries from long egocentric videos [38, 37, 71], recognize the actions performed by the wearer [18, 59, 103, 35, 50], and segment the videos according to detected ego-motion patterns [40, 41]. While current literature focuses on providing general-purpose methods which are usually optimized using data acquired by many users, we argue that, given the subjective nature of egocentric videos, more attention should be devoted to user-specific methods.

In this Section, we propose a system for personal location recognition which furthers exploits temporal coherence. Figure 2.20 shows a schema of the investigated method. Similarly to what assumed in the previous analysis, we consider a scenario where the user defines a number of locations of interest by providing minimal training data in the form of short videos (i.e., a 10 seconds video per location). Given the input egocentric video and the user-defined set of locations, the task is to establish

for each frame in the video if it is related to either one of the considered personal locations or none of them (i.e., it is a negative sample). As hypothesized before, we assume that the system is set up by the end user himself, hence training must be simple and achievable with few training data. Moreover, given the large variability exhibited by egocentric videos, it is unfeasible to ask the user to acquire a significant quantity of negative samples. Therefore, we assume that only positive samples of different locations are provided by the user and propose a method to detect negative samples automatically, without training on them.

Building on the results of analysis presented in Section 2.5 and Section 2.6, we have acquired a dataset of 10 personal locations using the head-mounted wideangular camera LX2W (which was the best performing in our benchmarks). We employ a fine-tuned Convolutional Neural Network (CNN) to discriminate among different locations and a Hidden Markov Model (HMM) to enforce temporal coherence among neighbouring predictions. To handle negative locations, we introduce a non-parametric method for the rejection of negative frames. Being non-parametric, the proposed method does not need any negative samples at training time. Considering possible real-time application of the proposed system, we analyze the computational performances of the proposed method and also suggest a simplified system which is efficient enough to run in real-time.

2.7.1 Proposed Method

Given an egocentric video as an ordered collection of image frames $\mathcal{V} = \{I_1, \ldots, I_n\}$, our system must be able to:

1. correctly classify each frame I_i as one of the considered locations;
2. reject negative frames;
3. enforce temporal coherence among neighboring predictions.

The system eventually returns the labeling $\mathcal{S} = \{C_1, \ldots, C_n\}$, where $C_i \in \{0, \ldots, M-1\}$ is the class label associated to frame I_i ($C_i = 0$ representing the negative class label) and M is the total number of classes including negatives ($M = 11$ in our case - 10 locations, plus the negative class). Rejection of negative samples is usually tackled increasing the number of classes by one and explicitly learning to recognize

negative samples. However, this procedure requires a number of training negative samples which may not be easily acquirable in our scenario. Indeed, given the large variability of visual content acquired by wearable devices, it would be infeasible to ask the user to acquire a sufficient number of representative negative samples. Therefore, we propose to treat negative rejection separately from classification and introduce a non-parametric rejection mechanism which does not need negative samples at training time.

We first consider a multi-class component which is trained solely on positive samples to discriminate among the considered positive $M-1$ classes. Since the multi-class model ignores the presence of negative frames, it only allows to estimate the posterior probability:

$$p(C_i|I_i, C_i \neq 0). \tag{2.7}$$

The probability reported in Equation (2.7) is the posterior probability over the 10 positive classes estimated by the classifier (the CNN model in our case), assuming that the input sample is not a negative. It should be noted that it sums to 1 over the positive classes, i.e.,

$$\sum_{j=1}^{10} p(C_i = j|I_i, C_i \neq 0) = 1, \tag{2.8}$$

while it does not say anything about the possibility of having a negative sample. Since we wish to correctly discriminate among the positive classes (the 10 locations of interest), as well as rejecting the negative samples, we want to model the following probability distribution:

$$p(C_i|I_i). \tag{2.9}$$

To this end, we propose to quantify the probability $p(C_i = 0|I_i)$ to be a negative sample of a given frame I_i, as the uncertainty of the discriminative model in predicting the class labels related to the previous k frames (in our experiments we use $k = 30$, which is equivalent to one second at 30 fps). Specifically, considering that both the visual content and class label are deemed to change slowly in egocentric videos, we assume that the past k frames $\mathcal{I}_i^k = \{I_i, I_{i-1}, \ldots, I_{\max(i-k+1,1)}\}$ are related to the same class. The notation "$\max(i-k+1, 1)$" is used to prevent including frames with negative indexes in the first k frames. Such assumption is of course

imprecise when $\mathcal{I}_i^k$ contains the boundary between two personal locations. However, such cases are rather rare and if k spans one second or less, the assumption only affects the boundary localization accuracy and does not have a huge impact on the overall accuracy. Since the discriminative model has been tuned only on positive samples, we expect it to exhibit low uncertainty when the frames in $\mathcal{I}_i^k$ are related to a positive class, while we expect a large amount of uncertainty in the case of negative samples. As suggested in [104], we measure the uncertainty of the model computing the variation ratio of the distribution of the labels $\mathcal{Y}_i^k = \{y_i, \ldots, y_{max(i-k+1,1)}\}$ predicted within $\mathcal{I}_i^k$ by maximizing the posterior probability reported in Equation (2.7): $y_i = \arg\max_j p(C_i = j|I_i, C_i \neq 0), j = 1, \ldots, M-1$. We finally assign the probability of I_i being a negative as the following expression:

$$p(C_i = 0|I_i) = 1 - \frac{\sum_j \mathbb{1}(y_j = \tilde{\mathcal{Y}}_i^k)}{|\mathcal{Y}_i^k|} \tag{2.10}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function and $\tilde{\mathcal{Y}}_i^k$ represents the mode of $\mathcal{Y}_i^k$. It should be noted that the definition reported in Equation (2.10) is totally arbitrary and encodes the belief that the model should agree on similar inputs if they are positive samples. In practice, given a number of predictions computed within a small temporal window, we quantify the probability of having a negative sample as the fraction of labels disagreeing with the mode.

Given that $C_i = 0$ and $C_i \neq 0$ are disjoint events and marginalizing, Equation (2.9) can be written in the following form:

$$\begin{aligned} p(C_i|I_i) &= p(C_i, C_i = 0|I_i) + p(C_i, C_i \neq 0|I_i) = \\ &= p(C_i|I_i, C_i = 0)p(C_i = 0|I_i) + p(C_i|I_i, C_i \neq 0)p(C_i \neq 0|I_i). \end{aligned} \tag{2.11}$$

Considering that $p(C_i = 0|I_i, C_i = 0) = 1$, $p(C_i = 0|I_i, C_i \neq 0) = 0$, and $p(C_i \neq 0|I_i, C_i = 0) = 0$, the expression in Equation (2.11) can be written as follows:

$$p(C_i|I_i) = \begin{cases} p(C_i = 0|I_i) & \text{if } C_i = 0 \\ p(C_i \neq 0|I_i) \cdot p(C_i|I_i, C_i \neq 0) & \text{otherwise} \end{cases}. \tag{2.12}$$

Equation (2.12) allows to combine the probabilities in Equation (2.7) and (2.10).

The final class prediction for frame I_i (including the rejection of negative samples) can be obtained maximizing Eq (2.12) as follows:

$$C_i^* = \arg\max_j p(C_i = j|I_i) \tag{2.13}$$

Given the nature of egocentric videos, it is likely that subsequent frames are related to the same location of interest, while a change of location is a rare event. Such prior can be taken into account in the computation of the final labeling, using a Hidden Markov Model. We consider the probability $p(\mathcal{S}|\mathcal{V})$, which, according to the Bayes' rule, can be expressed as follows:

$$p(\mathcal{S}|\mathcal{V}) \propto p(\mathcal{V}|\mathcal{S})p(\mathcal{S}). \tag{2.14}$$

Assuming conditional independence of the frames with respect to each other given their classes ($I_i \perp\!\!\!\perp I_j|C_i,\ \forall i,j \in \{1,2,\ldots,n\}, i \neq j$), and applying the Markovian assumption on the conditional probability distribution of the class labels ($p(C_i|C_{i-1}\ldots C_1) = p(C_i|C_{i-1})$), Equation (2.14) can be written as:

$$p(\mathcal{S}|\mathcal{V}) \propto p(C_1)\prod_{i=2}^{n} p(C_i|C_{i-1})\prod_{i=1}^{n} p(I_i|C_i). \tag{2.15}$$

Probability $p(C_1)$ is assumed to be constant over the different classes and can be ignored when maximizing Equation (2.14). Probability $p(I_i|C_i)$ can be inverted using the Bayes law:

$$p(I_i|C_i) \propto p(C_i|I_i)p(I_i). \tag{2.16}$$

Since I_i is observed, term $p(I_i)$ can be ignored, while $p(C_i|I_i)$ is estimated using Equation (2.12). Equation (2.14) can be hence written as:

$$p(\mathcal{S}|\mathcal{V}) \propto \prod_{i=2}^{n} p(C_i|C_{i-1})\prod_{i=1}^{n} p(C_i|I_i). \tag{2.17}$$

Term $p(C_i|C_{i-1})$ is the HMM state transition probability. Transition probabilities in Hidden Markov Models can generally be learned from the data as done in [54], or defined ad hoc to express a prior belief as done in [51]. Since we assume that few training data should be provided by the user and no labeled sequences are available

at training time, we define an ad-hoc transition probability as suggested by [51]:

$$p(C_i|C_{i-1}) = \begin{cases} \varepsilon, & \text{if } C_i \neq C_{i-1} \\ 1-(M-1)\varepsilon, & \text{otherwise} \end{cases} \tag{2.18}$$

where ε is a small constant (we use the machine accuracy in double precision 2.22×10^{-16} in our experiments). The probability in Equation (2.18) enforces coherence between subsequent states and penalizes random state changes. The final labeling of the input egocentric video is obtained choosing the one which maximizes the probability in Equation (2.14) using the Viterbi algorithm [97]:

$$\mathcal{S}^* = \arg\max_{\mathcal{S}} p(\mathcal{S}|\mathcal{V}). \tag{2.19}$$

2.7.2 Experimental Settings and Results

Experiments are performed on the 10-LOCATIONS dataset. All compared methods are trained on the whole training set and evaluated on the test sequences. The validation set is used to tune hyper-parameters and select the best performing iteration in the case of CNNs. In the following sections, we study the performances of the proposed method, paying particular attention to the optimization. Specifically, we evaluate different architectural tweaks which help reducing over-fitting when fine-tuning Convolutional Neural Networks on our small realistic dataset ($\approx$ 200 samples per class) and reduce computational requirements. Moreover, we discuss the influence of the different components included in our method (i.e., multiclass classifier, rejection mechanism, and HMM). After studying the performances of the proposed method, we compare it with respect to some baselines, including the benchmark classification pipeline proposed in Section 2.5.1.

Proposed Method: Optimization and Performances Evaluation

The multi-class classifier employed in the proposed method could be implemented using any algorithm able to output posterior probabilities in the form of Equation (2.7). We consider Convolutional Neural Networks given their compactness and the superior performances shown on many tasks including personal location recognition [24]. We fine-tune the VGG-S network proposed in [94] on our training

		Accuracy			Comp. Performances	
Id	**Settings**	**Discrim.**	**+Rejection**	**+HMM**	**Dimensions**	**Time**
[*a*]		76.90	69.60	73.83	378 MB	13.23 ms
[*b*]	L	83.30	76.06	83.22	378 MB	13.13 ms
[*c*]	L ND	**94.53**	**85.00**	**88.63**	378 MB	13.10 ms
[*d*]	L 128	83.07	77.49	82.84	34 MB	10.32 ms
[*e*]	L ND 128	77.09	71.99	73.59	34 MB	10.28 ms
[*f*]	L LR	**92.31**	**81.00**	**85.37**	26 MB	10.23 ms

Table 2.10: Optimization of the multi-class classifier. All results are related to experiments performed on the 10-LOCATIONS dataset. Architectural settings: L the convolutional layers are locked, ND dropout is disabled, 128 fully connected layers are reduced to 128 units and reinitialized, LR fully connected layers are replaced by a single logistic regression layer. Reported times are average per-image processing times. Maxima per column are reported in **underlined bold digits**, while second maxima are reported in **bold digits**.

set. Since the VGG network has been trained on the ImageNet dataset, we expect the learned features to be related to objects and hence relevant to the task of location recognition, as highlighted in [92].

Optimization of the Multi-Class Classifier

Fine-tuning a large CNN using a small training set ($\approx$ 200 samples per class) is not trivial and some architectural details can be tuned in order to optimize performances. Specifically, we assess the impact of the following architectural settings:

1. locking the convolutional layers (i.e., setting their relative learning rate to zero);
2. disabling dropout in the fully connected layers;
3. reducing the number of units in the fully connected layers from 4096 to 128;
4. removing the fully connected layers and attaching a logistic regression (softmax) layer directly to the last convolutional layer.

In the following, we discuss different combinations of the aforementioned architectural settings in order to assess the influence of each considered setting. Results for these experiments are reported in Table 2.10 and Table 2.11.

Table 2.10 is organized as follows. Each row of the table is related to a different experiment. The first column (Id) reports unique identifiers for the considered methods. The second column (Settings) summarizes the architectural settings related to the specific method. The third column (Discrimination) reports the accuracy of the multi-class model alone (i.e., class labels are directly computed using Equation (2.7)). Note that such accuracy values are computed removing all negative samples from the test set. The fourth column (+Rejection) reports the accuracy of the models after applying the proposed rejection method (i.e., labels are obtained using Equation (2.13)). The fifth column (+HMM) reports the accuracy of the complete method including the Hidden Markov Model (i.e., final labels are obtained using Equation (2.19)). Column 6 (Dimensions) reports the size of the models in megabytes. Finally column 7 (Time) reports the average time needed to predict the class label of a single frame[5]. Table 2.11 reports per-class true positive rates for the considered configurations.

The reported results highlight the importance of tuning the considered architectural settings to improve both computational performances and accuracy. In particular, locking the convolutional layers allows to significantly improve the performances of the fine-tuned model (compare [*b*] to [*a*] in Table 2.10)[6]. Significant performance improvements are observable when the CNN is evaluated alone (Discrimination column) as well as when the model is integrated in the proposed system (columns +Rejection and +HMM). This result highlights how the unlocked network suffers from over-fitting, due to the high number of parameters to optimize with relatively few training data. It should be noted that, in our experiments, only convolutional layers are locked, while fully connected ones are still optimized. Locking convolutional layers, hence, allows to use part of the network as a bank of object-related feature extractors (the pre-trained convolutional layers), while optimizing the way such features are combined in the fully connected layers.

Disabling dropout has a positive impact when convolutional layers are locked and fully connected layers are fine-tuned ([*c*] vs [*b*]). This indicates that dropout is causing the model to underfit due to the scarcity of training data. Interestingly, when fully connected layers are reduced to 128 units and hence reinitialized with

[5]Times have been estimated running the CNN models on a NVIDIA GeForce GTX 480 GPU using the Caffe framework [105]. They include the rejection of negative frames but do not take into account the application of the Hidden Markov Model.

[6]SVM models are tested on a Intel(R) Core(TM) i7-3930K CPU @ 3.20GHz with LIBSVM [100].

		Per-Class True Positive Rate (TPR)										
Id	**Settings**	**Car**	**C.V.M.**	**Gar.**	**K.T.**	**L.Off.**	**Off.**	**Piano**	**Sink**	**Stud.**	**L.R.**	**Neg.**
[*a*]		91.28	**98.73**	**98.71**	**100.0**	**95.87**	**94.81**	**98.52**	**100.0**	99.40	**99.20**	36.91
[*b*]	L	90.71	**98.53**	**98.41**	99.60	93.83	**93.57**	98.48	99.00	98.50	98.91	47.77
[*c*]	L ND	75.57	92.42	87.60	97.95	84.08	71.67	93.32	96.69	94.09	89.73	**82.34**
[*d*]	L 128	**99.09**	94.36	74.90	89.46	93.51	84.66	98.16	98.90	**99.72**	99.09	51.22
[*e*]	L ND 128	**99.57**	95.43	98.31	**100.0**	**98.54**	90.25	**98.68**	**99.51**	**99.82**	**99.14**	36.51
[*f*]	L LR	94.53	78.93	85.88	78.39	89.91	60.28	93.57	96.91	97.46	98.20	**61.66**

Table 2.11: Per-class true positive rates for the considered configurations. Results are related to experiments performed on the 10-LOCATIONS dataset. See Table 2.10 for a legend.

Gaussian noise, disabling dropout seems to favor overfitting as one would generally expect (compare [*e*] to [*d*]). This behavior is probably due to the inclination of randomly reinitialized layers to easily co-adapt [106]. Reducing the dimensionality of the fully connected layers to 128 units helps reducing the dimensions of the network and improving its speed, but results in a substantial loss in accuracy due to the needed reinitialization of the weights (compare [*d*] to [*c*]).

In order to devise a more compact model, we finally consider replacing the fully connected layers with a logistic regressor (i.e., a layer with 10 units followed by softmax). In this case, the locked convolutional layers of the VGG-S network are used as feature extractors, while predictions are performed combining them using a simple logistic regressor classifier. This configuration allows to greatly reduce memory and time requirements at the cost of a modest loss in terms of accuracy (compare [*f*] to [*c*], [*d*], [*e*]).

Among all compared method, the most accurate is [*c*], followed by the computationally efficient [*f*]. Both methods outperform the others by a good margin. Moreover, it is worth noting that [*f*] is more than 90% smaller and 20% faster than [*c*] while only about 3% less accurate. Such result is particularly interesting in real-time scenarios involving low-resources and embedded devices (e.g., in smart glasses or in a drone). Finally, as can be noted from Table 2.11, only the two best configurations (methods [*c*] and [*f*]) succeed in correctly rejecting negative samples, while other methods yield lower true positive rates.

Performances of the proposed method

As discussed above, columns 3 to 5 in Table 2.10 report performances related to the main components involved in the proposed method, i.e., multi-class classifier, rejection mechanism and Hidden Markov Model. As can be noted, high accuracies can be achieved when discriminating among a finite number of possible locations (column Discrimination). The need for a rejection mechanism in real-world scenarios makes the problem much harder, decreasing classification accuracy by 10% in average (compare Discrimination with +Rejection columns). These results suggest that more efforts should be devoted to effective rejection mechanisms in order to make current classification systems useful in real world applications. Indeed, any real system devoted to distinguish among a number of classes must be able to deal with the negative ones. Enforcing temporal coherence using a Hidden Markov Model generally helps reducing the gap between simple discrimination and discrimination + rejection (consider for instance methods [*c*] and [*f*]). The effects of the rejection and HMM modules are qualitatively illustrated in Figure 2.21. As can be noted, simple class discrimination (top row) yields noisy predictions when ground truth frames are negative. The rejection mechanism (second row) successfully detects negative samples. The use of a HMM (third row) finally helps reducing sudden changes in the predicted labels.

Comparison with the State of the Art

To assess the effectiveness of the proposed method, we compare it with respect to two baselines and an existing method for personal location recognition [24]. The first baseline tackles the location recognition problem through feature matching. The system is initialized extracting SIFT feature points from each test image and storing them for later use. Given the current frame, SIFT features are extracted and matched with all images in the training set. To reduce the influence of outlier feature points, for each considered image pair, we perform a geometric verification using the MSAC algorithm [107] based on an affine model. Classification is hence performed considering the training set image presenting the highest number of inliers and selecting the class to which it belongs. In this case, the most straightforward way to perform rejection probably consists in setting a threshold on the number of inliers: if an image is a positive, it is expected to yield a good match with some

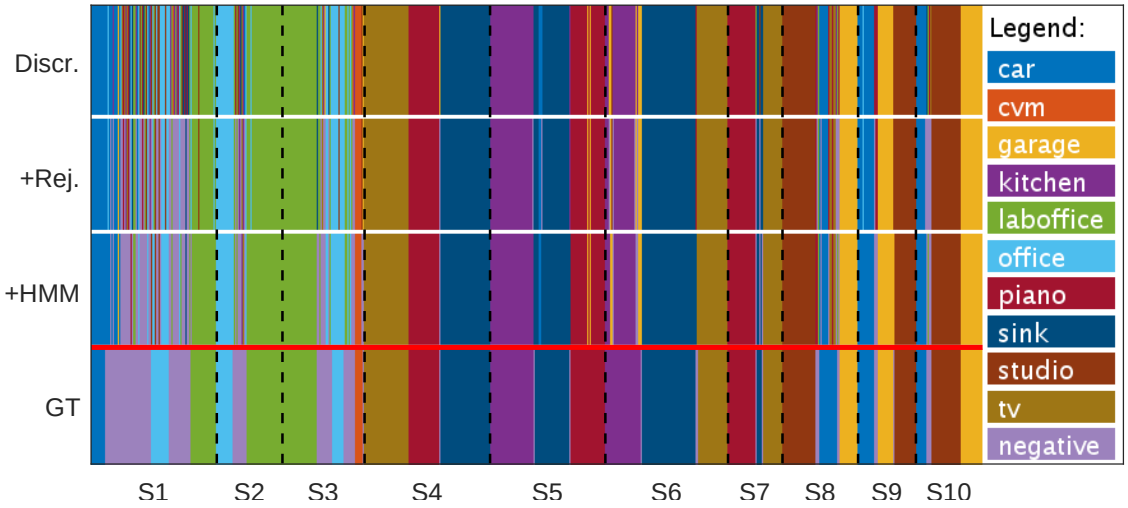


Figure 2.21: Graphical representation of the labels produced by the proposed method (method [*c*] in Table 2.10). Each row reports the concatenation of labels produced for all test sequences. Boundaries between sequences are highlighted with black dashed lines and "S1" ... "S10" labels. The visualization is intended to qualitatively assess the influence of the rejection and HMM components on the performances of the overall system. Specifically, the first three rows report labels obtained using the multi-class classifier, the proposed rejection mechanism and the HMM, similarly to what discussed for Table 2.10. The last row reports the ground truth. Best seen in color.

example in the dataset, otherwise only weak matches should be obtained. Since it is not clear how such a threshold should be arbitrarily set, we learn it from data. To do so, we first normalize the number of inliers by the number of features extracted from the current frame. We then select the threshold which best separates the validation set from the training negatives. To speed up computation, input images are rescaled in order to have a standard height of 256 pixels (the same size to which images are resized when fed to CNN models), keeping the original aspect ratio.

The second considered baseline consists in a CNN trained to discriminate directly between locations of interest and negatives. In contrast with the proposed method, the baseline explicitly learns from negative samples. Hence, in our settings, the model is trained on 11 classes comprising 10 locations of interest, plus the negative class. This baseline is implemented adopting the same architecture as the one of method [*c*], which is the best performing configuration in our experiments. It should be noted that training negatives are independent from validation and test negatives. We also compare our method with respect to the baseline introduced in Section 2.5.1,

		Accuracy			Comp. Performances	
Id	**Settings**	**Discrim.**	**+Rejection**	**+HMM**	**Dimensions**	**Time**
[c]	L ND	**94.53**	**85.00**	**88.63**	378 MB	13.10 ms
[f]	L LR	**92.31**	**81.00**	**85.37**	26 MB	10.23 ms
[g]	SIFT	34.64	33.16	–	71 MB	5170.1 ms
[h]	L ND NE	73.84	76.42	79.69	378 MB	12.82 ms
[i]	SVM	87.76	74.14	79.64	423 MB	97.83 ms

Table 2.12: Comparisons with the state of the art. Results are related to experiments performed on the 10-LOCATIONS dataset. Methods [c] and [f] are reported from Table 2.10 for convenience. Architectural settings: L the convolutional layers are locked, ND dropout is disabled, LR fully connected layers are replaced by a single logistic regression layer, SIFT the SIFT feature matching baseline, NE the model is trained on both positive and negative samples, SVM classification based on one-class and multiclass SVM classifiers.

		Per-Class True Positive Rate (TPR)										
Id	**Settings**	**Car**	**C.V.M.**	**Gar.**	**K.T.**	**L.Off.**	**Off.**	**Piano**	**Sink**	**Stud.**	**L.R.**	**Neg.**
[c]	L ND	75.57	92.42	**87.60**	**97.95**	84.08	**71.67**	**93.32**	**96.69**	94.09	89.73	**82.34**
[f]	L LR	**94.53**	78.93	85.88	78.39	**89.91**	60.28	**93.57**	**96.91**	**97.46**	**98.20**	61.66
[g]	SIFT	4.90	5.55	0.02	71.45	15.37	16.62	84.98	22.21	12.80	79.77	24.22
[h]	L ND NE	**78.16**	**95.23**	71.48	97.53	73.54	50.03	71.95	93.43	**95.70**	73.49	**95.72**
[i]	SVM [24]	74.97	**98.16**	**97.63**	**98.45**	**88.60**	**92.27**	79.13	69.25	59.16	**99.13**	06.58

Table 2.13: Per-class true positive rates of the compared methods. Results are related to experiments performed on the 10-LOCATIONS dataset. See Table 2.12 for a legend.

which performs negative rejection and location recognition using a cascade of One-Class and multiclass SVM classifiers trained on features extracted employing the VGG network [94].

Table 2.12 and Table 2.13 compare the performances of the considered methods. As can be noted, the proposed methods [c] and [f] retain the highest accuracies in Table 2.12. Requiring about 5 seconds to process each frame, the SIFT matching method ([g] in Table 2.12) is the slowest among the compared ones. Moreover, SIFT matching achieves poor results on the considered task, which indicates that it is not able to generalize to new views of the same scene and to cope with the many variabilities typical of egocentric videos. It should be noted that, since the SIFT baseline does not output any probability values, the HMM cannot be applied in this case.

The baseline [h] retains a high TPR on negative samples (see Neg. column in Table 2.13). However TPRs related to other classes and the accuracy of the overall

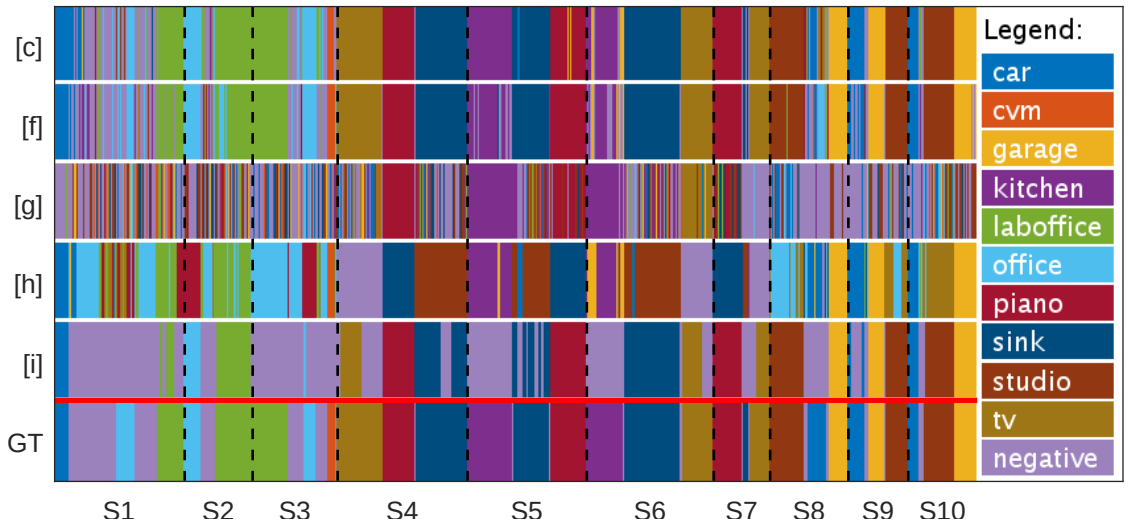


Figure 2.22: Graphical representation of the results produced by the considered methods (see Table 2.12). Detailed visualizations for each sequence are available in the supplementary material.

system are lower when compared to the proposed approaches. This indicates how learning from negative samples is not trivial in the proposed problem. The method introduced in [24] is outperformed by the proposed methods (compare [*i*] to [*c*]-[*f*]) and gives inconsistent results in the rejection of negative frames (see column Neg. in Table 2.13). Moreover, the proposed approaches are significantly faster and have smaller size. Figure 2.22 reports the results of all compared methods for qualitative assessment. Figures 2.33 - 2.42, report detailed diagrams for all methods compared in Table 2.10. As can be noted, the proposed methods ([*c*] and [*f*] in Table2.10) in average outperform the competitors and reach remarkable performances in some cases (e.g., Figures 2.36, 2.37, 2.39, 2.41).

2.7.3 Discussion

The work discussed in this Section complements the analysis presented in Section 2.5 and Section 2.6 proposing a method to further exploit temporal coherence. Coherently with the premises made throughout this Chapter, the system can be trained with few positive samples provided by the user. The proposed system addresses some of the challenges identified in the benchmarks. This is done by providing a robust, non-parametric negative rejection component, tuning the employed CNN

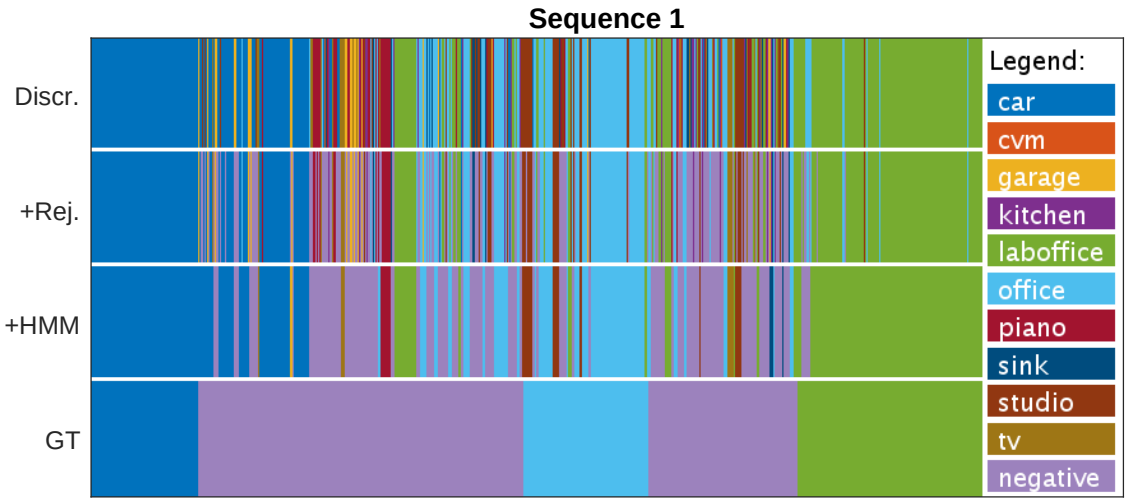


Figure 2.23: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence1.

models to reduce overfitting due to scarce training data, and enforcing temporal coherence among neighboring predictions using a Hidden Markov Model.

While evaluations show that the proposed method compares positively against baselines and state of the art methods, they also highlight the two main challenges of the considered task: the scarcity of training data and the challenging problem of negative location rejection. Among the possible ways to deal with such challenges, we identify at least two possible paths which may be pursued in future works. The former consists in leveraging data acquired by multiple users in order to exploit the commonalities of the training samples (i.e., multiple users might select similar locations). The latter consists in considering unsupervised or reinforcement learning techniques to leverage the huge quantity of data acquired by first person vision system in order to improve personal location recognition models. Moreover, future works will concentrate on complementing the analysis in order to assess the generality of the found results. In particular, the analysis will be extended to data acquired from multiple users to evaluate the generality of the methods with respect to different users and locations.

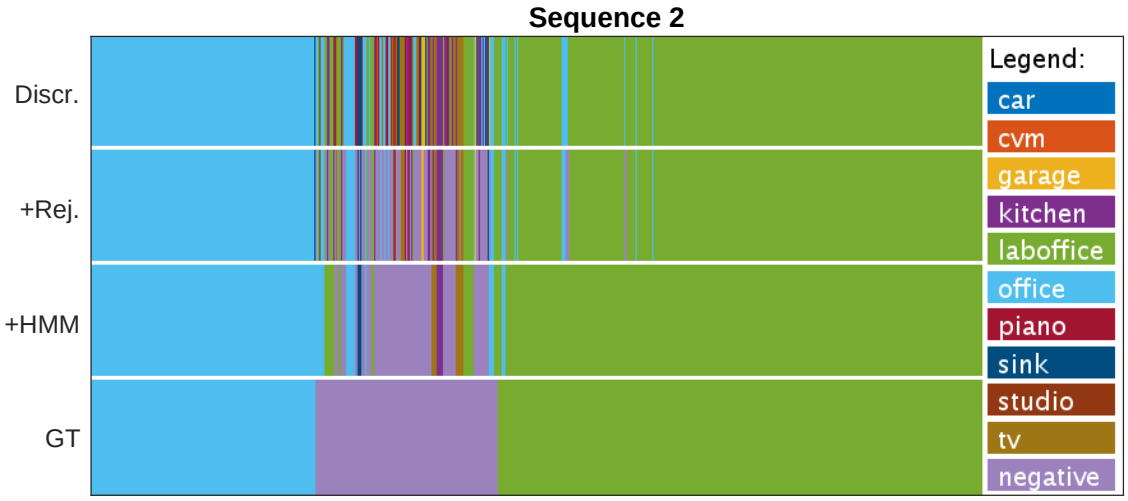


Figure 2.24: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence 2.

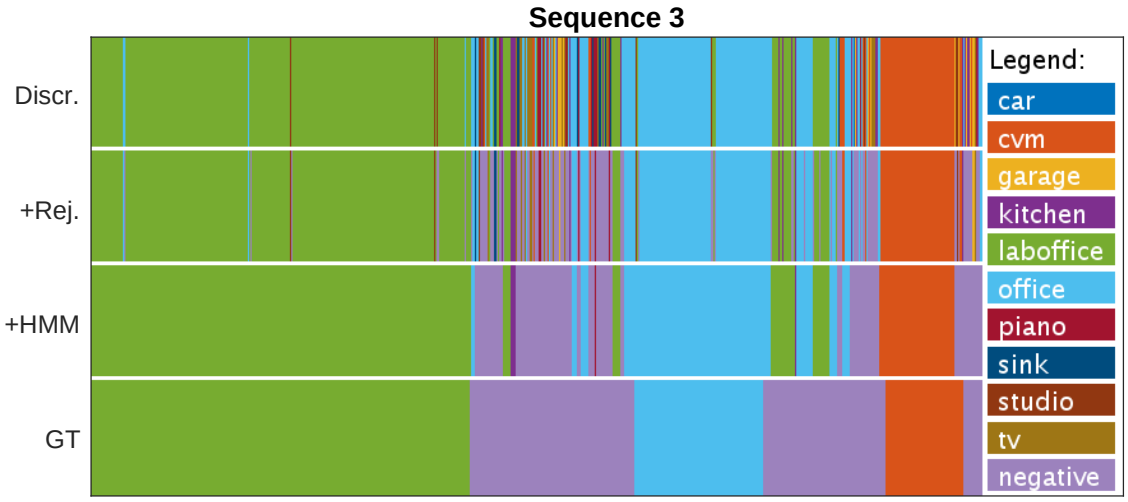


Figure 2.25: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence 3.

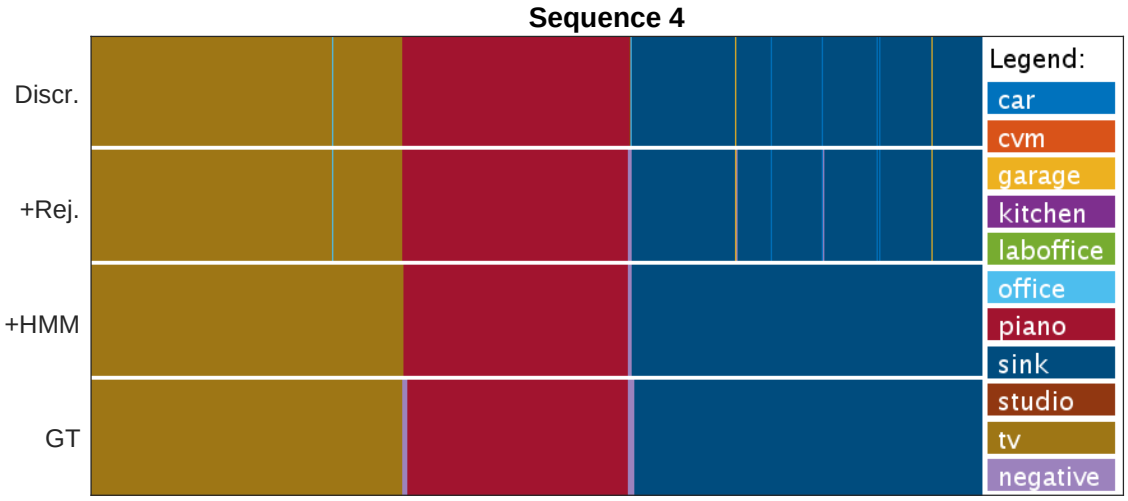


Figure 2.26: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence 4.

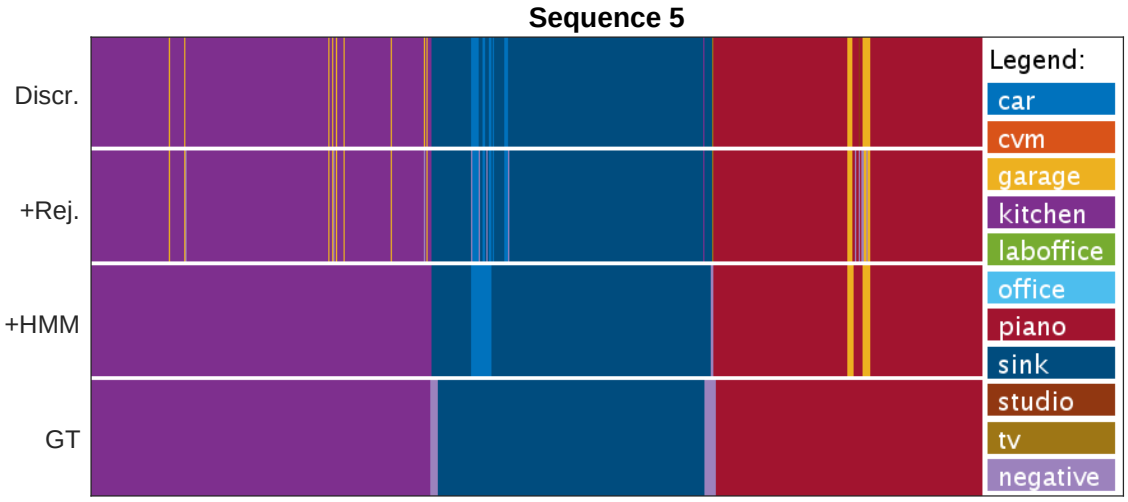


Figure 2.27: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence 5.

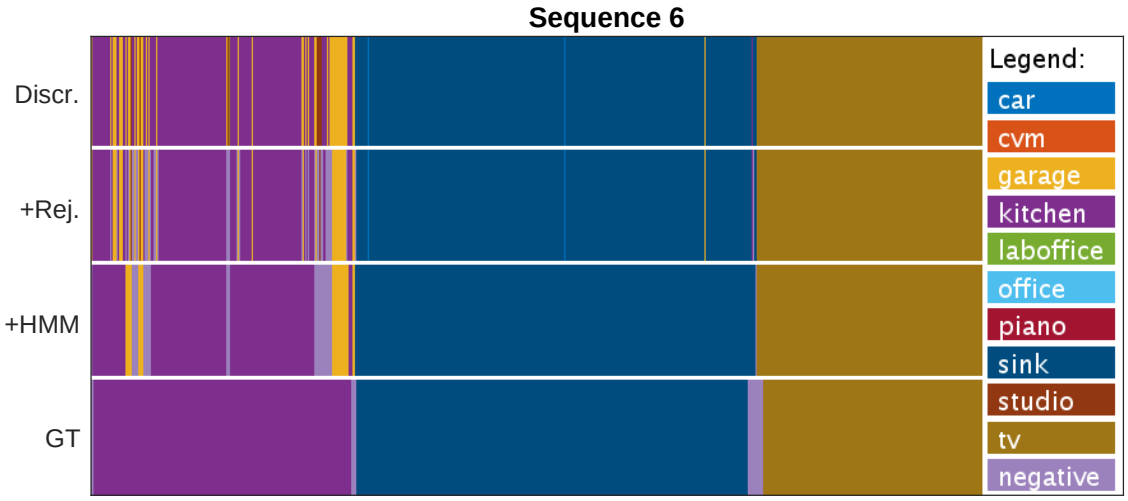


Figure 2.28: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence 6.

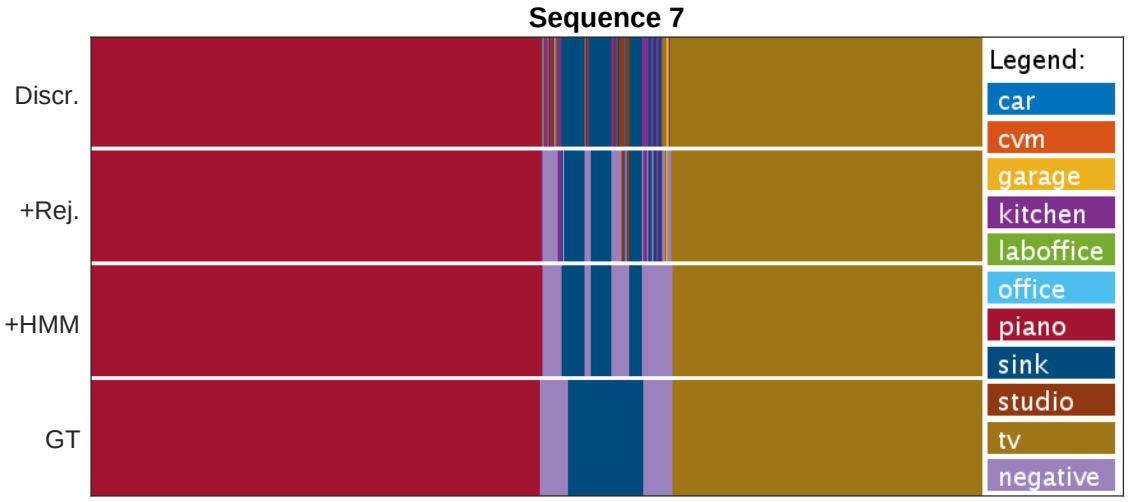


Figure 2.29: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence 7.

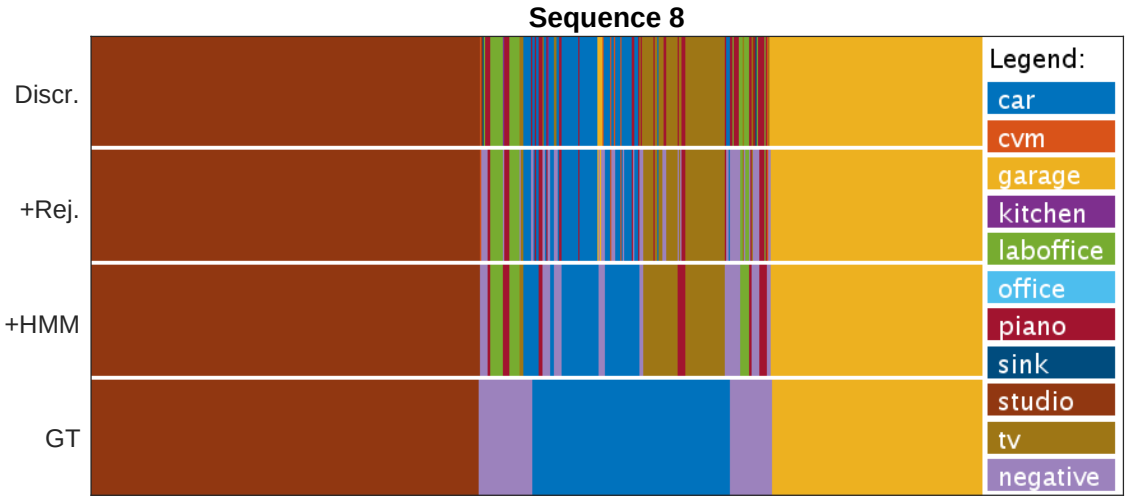


Figure 2.30: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence 8.

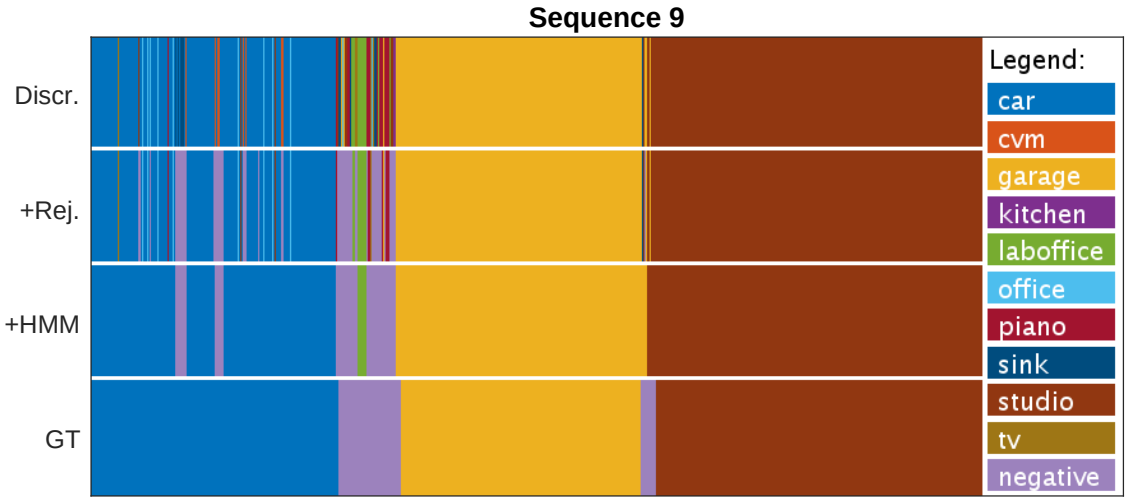


Figure 2.31: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence 9.

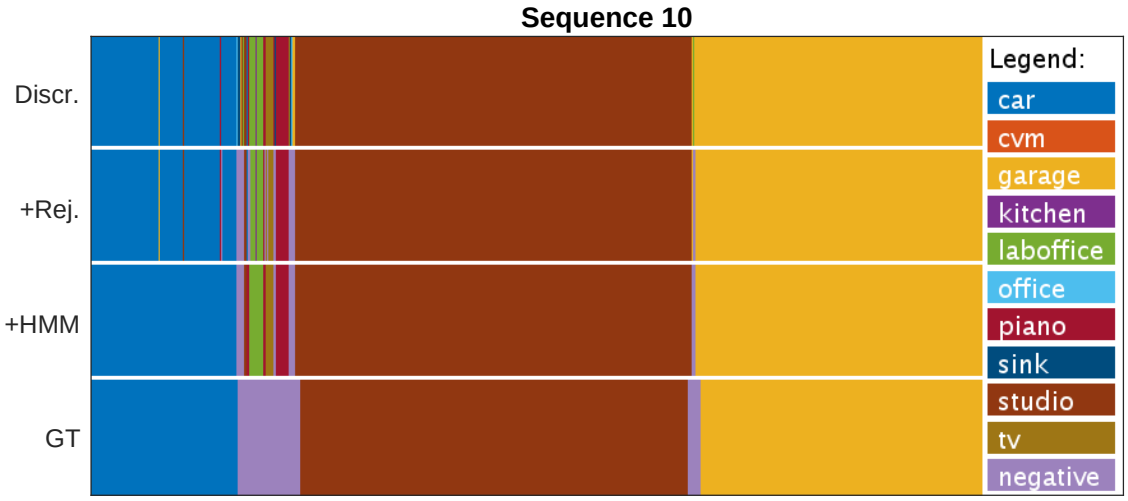


Figure 2.32: Results obtained with the proposed method [*c*] in Table 2.10 related to Sequence 10.

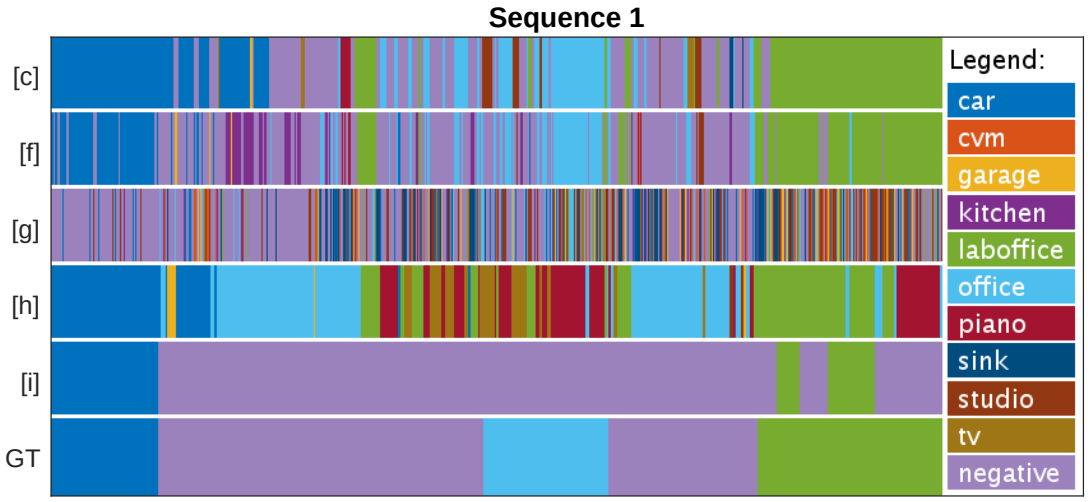


Figure 2.33: Comparative results of the methods reported in Table 2.10 related to Sequence 1.

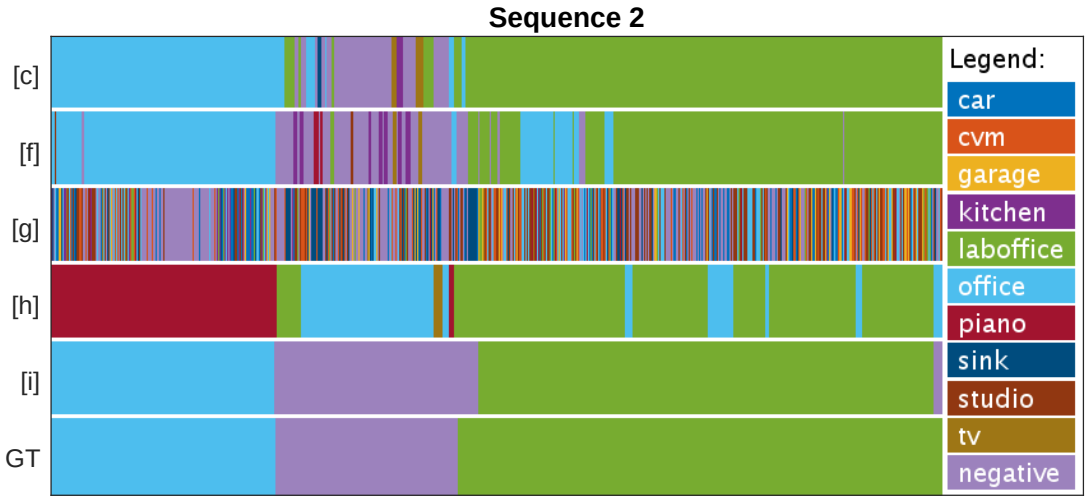


Figure 2.34: Comparative results of the methods reported in Table 2.10 related to Sequence 2.

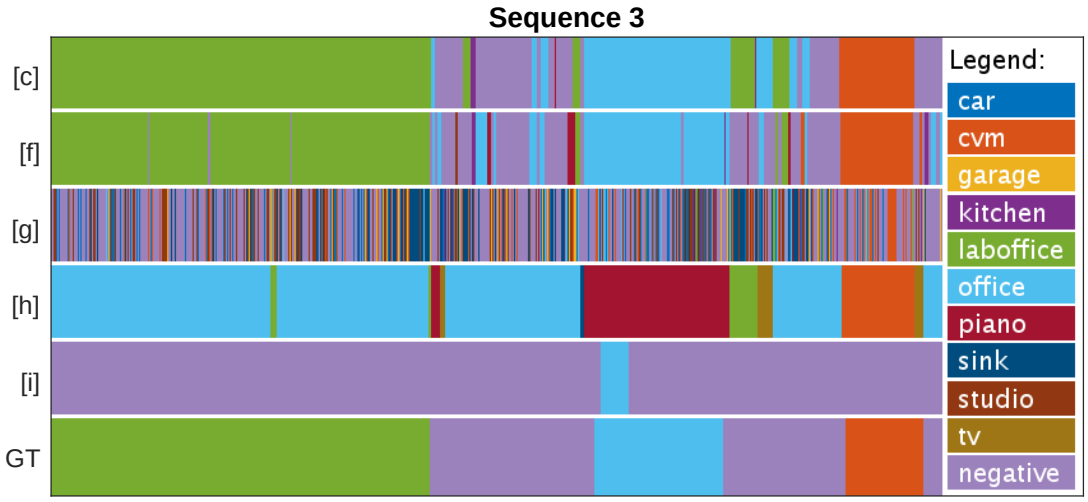


Figure 2.35: Comparative results of the methods reported in Table 2.10 related to Sequence 3.

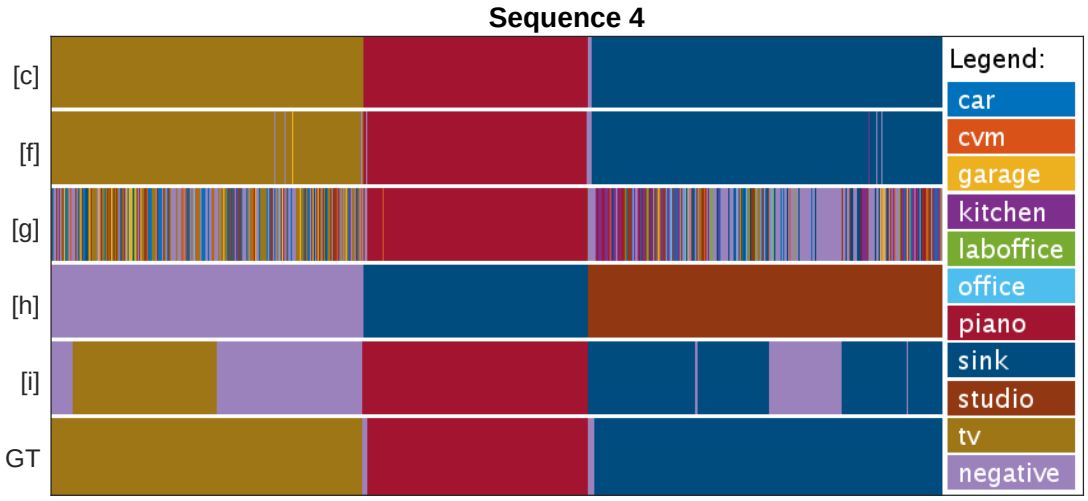


Figure 2.36: Comparative results of the methods reported in Table 2.10 related to Sequence 4.

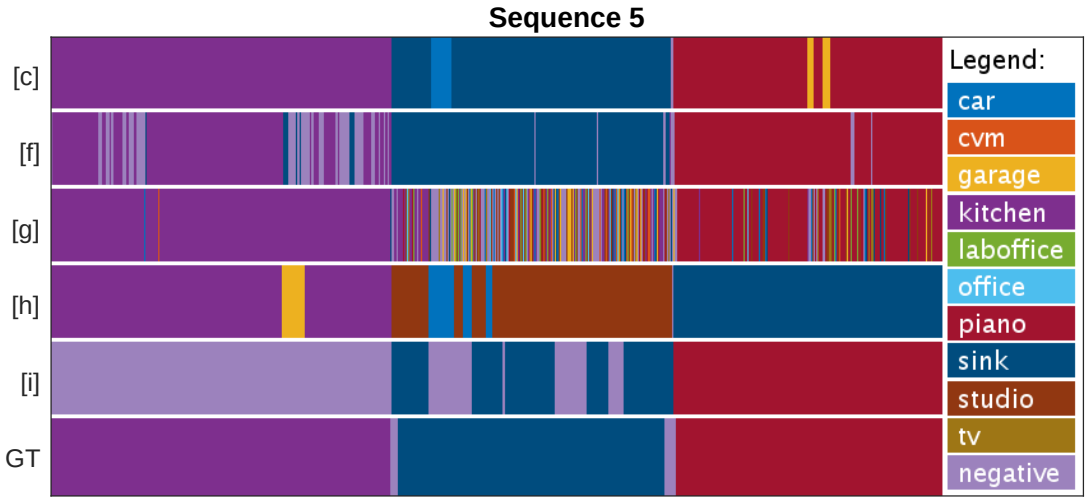


Figure 2.37: Comparative results of the methods reported in Table 2.10 related to Sequence 5.

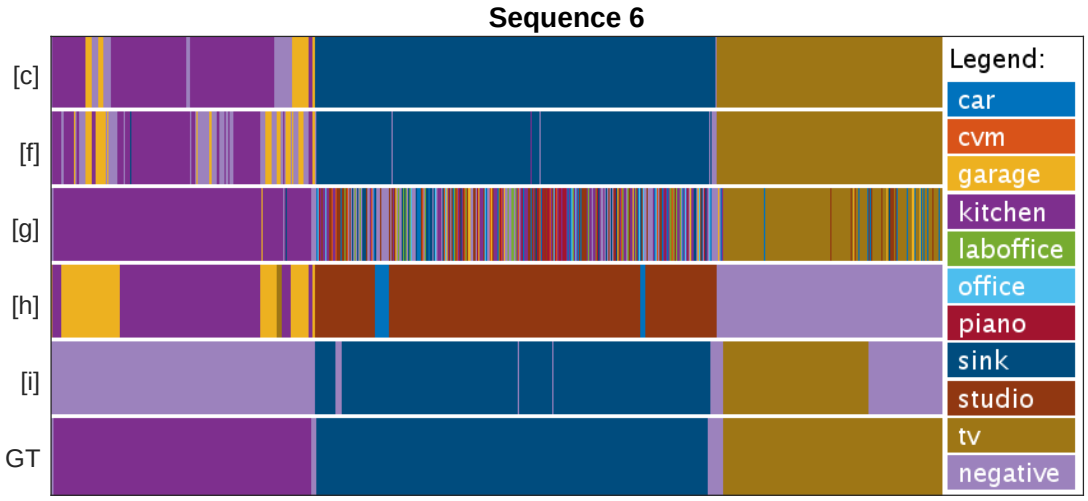


Figure 2.38: Comparative results of the methods reported in Table 2.10 related to Sequence 6.

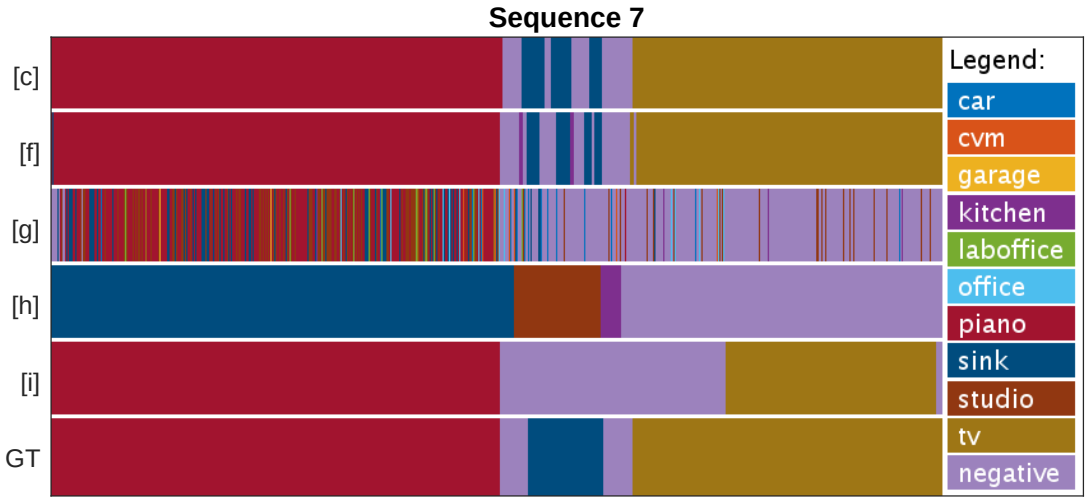


Figure 2.39: Comparative results of the methods reported in Table 2.10 related to Sequence 7.

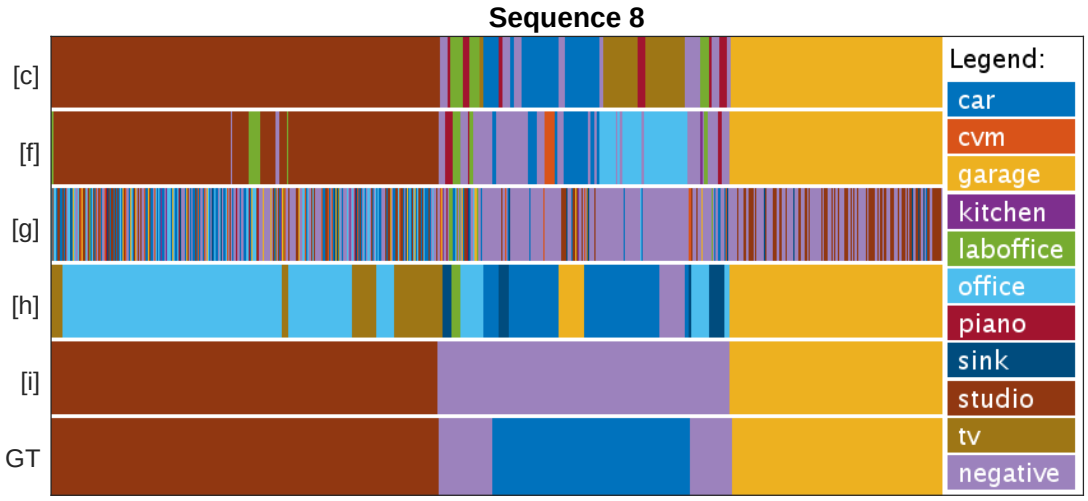


Figure 2.40: Comparative results of the methods reported in Table 2.10 related to Sequence 8.

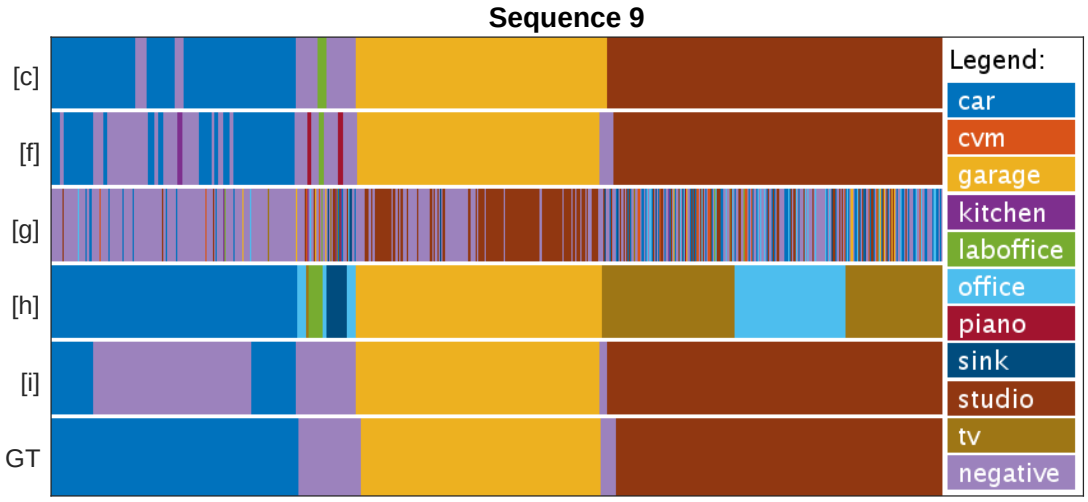


Figure 2.41: Comparative results of the methods reported in Table 2.10 related to Sequence 9.

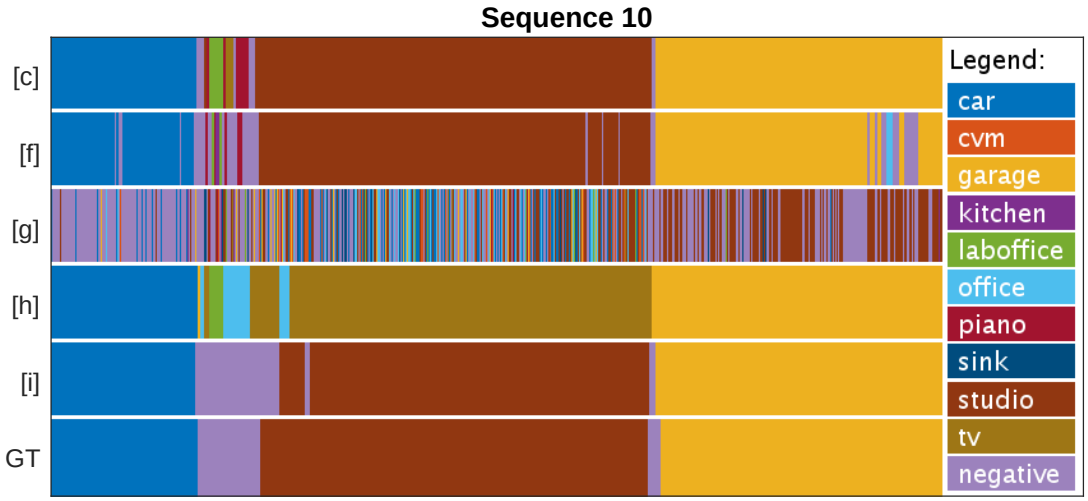


Figure 2.42: Comparative results of the methods reported in Table 2.10 related to Sequence 10.

Chapter 3

Next-Active-Object Prediction from Egocentric Videos[1]

One of the main advantages of First Person Vision systems is their ability to acquire information which is inherently meaningful for the user. Therefore, as pointed out by Kanade and Herbert [10], one of the main goals of a First Person Vision System is understanding the user's environments, behavior and intent. In Chapter 2, we discussed the importance of context and location awareness in First Person Vision systems and investigated methods to tackle the main challenges which originate from real scenarios. We also discussed that personal location awareness can be directly leveraged to infer behavioral information which can guide the construction of intelligent systems able to assist the user. While personal locations can help define context, the ability to understand and possibly anticipate the user's short and long term goals is still a key component for First Person Vision systems [10]. As claimed in previous works [108, 109, 110], the ability to anticipate the future is an essential property that humans exploit on a daily basis in order to communicate and interact with each other. For instance, predicting object interactions before they actually occur can be useful to provide guidance on object usage [76] or issue notifications [111]. Anticipated object interactions can tell us something more about the user's long term goals, as well as the intended activities. Indeed, as observed in [19, 36, 65], it is advantageous to decompose long term egocentric activities in terms of "atomic actions" and interactions with objects to improve the final activity recognition task. Taking advantage of the First Person Vision paradigm, in this

[1]The work presented in this chapter has been partially done while I was a visiting Scholar at the University of Texas at Austin, under the supervision of Professor Kristen Grauman.

chapter, we introduce the novel task of predicting which objects the user is going to interact with from egocentric videos. Following recent literature which claims the importance of "active objects" for activity understanding [19], we refer to our task as "next-active-object prediction".

The rest of this chapter is organized as follows. In Section 3.1 we define the next-active-object prediction task. Section 3.2 reviews the literature related to our investigation. In Section 3.3 we propose a next-active-prediction method based on the analysis of egocentric object trajectories. In Section 3.4 we present the experimental settings, whereas in Section 3.5 we discuss the results. Section 3.6 concludes the chapter.

3.1 Next-Active-Object Prediction

We consider a scenario in which the user is wearing a First Person Vision system (e.g., smart glasses) while performing his daily activities. As the user performs the intended activities, he will move through the environment and interact with specific objects. For instance, the activity of making tea will involve interactions with objects such as the kettle, tea bag and mug. We assume that the First Person Vision system is equipped with an object detector trained on a number of task-relevant object classes. Given a number of observed frames, our aim is to predict which objects are going to become active in order to distinguish them from the ones which will likely remain passive. Figure 3.1 shows a sketch of the considered problem. As the user moves through the environment, we aim at predicting the next interacted object (e.g., the fridge), while all other passive objects (in gray) should be discarded. Please note that next-active-object prediction needs to be performed before the interaction actually begins.

Predicting next-active-objects in unconstrained settings is hard since humans interact with objects on the basis of their final goal and the responses gathered from the environment. Nevertheless, we argue that the FPV paradigm can provide important cues related to the dynamic of the motion of the user with respect to the objects present in the scene. Our main intuition is that, when a user is performing a specific task, the way he moves and interacts with the environment is influenced by his goals and intended interactions with objects. According to this assumption,

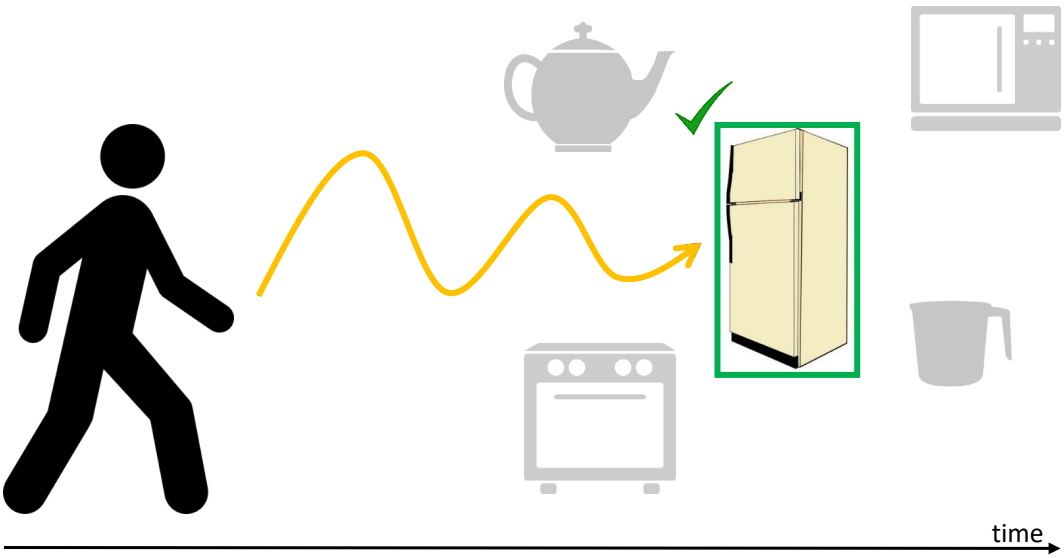


Figure 3.1: A sketch of the next-active-object prediction problem.

in an egocentric scenario, the relative motion of an object in the frame will vary depending on whether the user is willing to interact with that object or not. For instance, the user is expected to move towards an object before interacting with it. Figure 3.2 shows three sequences illustrating next-active-objects (in red) and passive ones (in cyan) along with their egocentric object trajectories. Our insight is that the shape of trajectories, as well as the positions in which they occur in the frame can help to predict the next-active-objects discriminating them from passive ones.

In this chapter, we investigate the relevance of egocentric object trajectories in the task of next-active-object prediction. Provided that an object detector/tracker is available, we propose to analyze object trajectories observed in a small temporal window to predict next-active-objects before the object-interaction is actually started. We investigate what properties of object motion are most discriminative and the temporal support with respect to which such motion should be analyzed. The proposed method compares favorably with respect to different baselines exploiting other cues which might be available in the scene, such as the distance of the objects from the center [19], the presence of hands [18, 43, 35, 36], changes in the object appearance [19] and the predictability of the user's visual attention [76].

Figure 3.2: Three sequences illustrating next-active-objects (in red) and passive ones (in cyan) along with their egocentric trajectories.

3.2 Related Work

Our work is related to different topics concerning activity recognition from egocentric videos, future prediction and active objects.

3.2.1 Activity Recognition in First Person Vision

Activity recognition from egocentric videos is an active area of research. Through the years, many approaches have been proposed to leverage specific egocentric cues. Spriggs et al. [50] proposed to use Inertial Measurement Units (IMU) and a wearable camera to perform activity classification and to segment the video into specific actions. Kitani et al. [59] addressed the problem of discovering egocentric action categories from first person sports videos in an unsupervised scenario. Fathi et al. [18] proposed to analyze egocentric activities to jointly infer activities, hands and objects. Fathi et al. [43] concentrated on activities requiring eye-hand coordination and proposed to predict graze sequences and action labels jointly. Pirsiavash and Ramanan [19] investigated an object-centric representation for recognizing daily

activities from first person camera views. McCandless and Grauman [112] proposed to learn the spatio-temporal partitions which were most discriminative for a set of egocentric activities. Ryoo and Matthies [61] considered videos acquired from a robot-centric perspective and proposed to recognize egocentric activities performed by other subjects while interacting with the robot. Li et al. [35] proposed a benchmark of different egocentric cues for action recognition. Ryoo et al. [103] proposed a feature pooling method to recognize egocentric activities. The authors of [36, 65] proposed to integrate different egocentric cues to recognize activities using deep learning. The aforementioned works assume that the activities can be fully observed before performing the recognition process and do not concentrate on future prediction from the observed data.

3.2.2 Future Prediction in Third Person Vision

Previous works have investigated the problem of early action recognition and future action prediction from a standard third person perspective. Even if such works do not consider egocentric scenarios, the main motivation behind them is related to ours: building systems which are able to recognize ongoing events from partial observations and react in a timely way. The considered application scenarios range from video surveillance to human-robot interaction. Ryoo [113] proposed a method to recognize ongoing activities from streaming videos. Huang et al. [114] introduced a system which copes with the ambiguity of partial observations by sequentially discarding classes until only one class is identified as the detected one. Hoai and De la Torre [115] exploited Structured Output SVM to recognize partial events and enable early recognition. Kong and Fu [116] designed compositional kernels to hierarchically capture the relationship between partial observations. Ma et al. [117] investigated a method to improve training of temporal deep models to learn activity progression for activity detection and early recognition tasks.

Beyond early action recognition, other methods have concentrated on the prediction of future actions before they actually occur. In particular, Kitani [118] modeled the effect of the physical environment on the choice of human actions in the scenario of trajectory-based activity analysis from visual input. Koppula et al. [108], studied how to enable robots to anticipate human-object interactions from visual input in order to provide adequate assistance to the user. Lan et al. [109] exploited a

hierarchical representation of human movements to infer future actions from a still image or a short video clip. Vondrick et al. [119] proposed to predict future image representations in order to forecast human actions from video.

3.2.3 Future Prediction in First Person Vision

Future prediction has been investigated also in the first person vision domain. The main application scenario related to such works concerns user assistance and aiding human-machine interaction. Zhou et al. [110] concentrated on the task of inferring temporal ordering from egocentric videos. Singh et al. [49] and Soo Park et al. [120] presented methods to predict future human trajectories from egocentric images. Soran et al. [111] proposed a system which analyzes complex activities and notifies the user when he forgets to perform an important action. Su and Grauman [121] proposed to predict the next object detector to run on streaming videos to perform activity recognition. Ryoo et al. [62] proposed a method for early detection of actions performed by humans on a robot from a first person, robot-centric perspective. Vondrick et al. [119] proposed to forecast the presence of objects in egocentric videos from anticipated visual representations. Our investigation is related to this line of works but, rather than considering prediction at the activity level, we focus on the granularity of user-object interaction and exploit the information provided by object motion dynamics in egocentric videos.

3.2.4 Active Objects

Our interest in next-active-object prediction has also been fostered by the importance of active objects in tasks such as egocentric activity recognition. In particular, Pirsiavash and Ramanan [19] proposed to distinguish active objects from passive ones. Active objects are objects being manipulated by the user and provide important information about the action being performed (e.g., using the kettle to boil water). Passive objects are non-manipulated objects and provide context information (e.g., a room with a fridge and a stove is probably a kitchen). The primary assumption made by Pirsiavash and Ramanan [19] is that active and passive objects can be discriminated by their appearance (e.g., an active fridge is probably open and looks different from a passive one) and the position in which they appear in the

frame (i.e., active objects tend to appear near the center). Active objects have also been been considered in recent research on egocentric activity recognition. Fathi et al. [18] suggested to pay special attention to objects manipulated by hands for egocentric activity recognition. Li et al. [35] used Improved Dense Trajectories to extract features from the objects the user is interacting with. Ma et al. [36] designed a deep learning framework which integrates different egocentric cues including optical flow, hand segmentation and objects of interest for egocentric activity recognition. Zhou et al. [65] presented a cascade neural network to collaboratively infer the hand segmentation maps and manipulated foreground objects.

The general idea that some objects are more important than others has been investigated also in other scenarios related to First Person Vision. Lee and Grauman [39] designed methods to summarize egocentric video by predicting important objects the user interacts with during the day. Bertasius et al. [122] designed a method for detecting action-objects, i.e., objects associated with seeing and touching actions. Damen et al. [76] proposed an unsupervised approach to detect task-relevant objects and provide gaze-triggered video guidance when the user intends to interact with the object.

3.3 Method

We propose to predict next-active-objects from egocentric videos by analyzing egocentric object trajectories. We assume that an object detector trained on a set of N object categories is available. A tracker is used to associate detections related to the same object instance in order to generate object tracks. At each time step, the system analyses the trajectories observed in the last h frames in order to recognize the next-active-objects before interaction actually takes place.

3.3.1 Object Tracks

For training purposes, we first assume that a set of egocentric videos is provided along with ground truth object annotations related to N different object classes. We also assume that each annotated object is labeled as active if the user is interacting with it or passive otherwise. Annotations related to the same object instance are grouped into tracks. We consider an object track as a sequence of bounding boxes

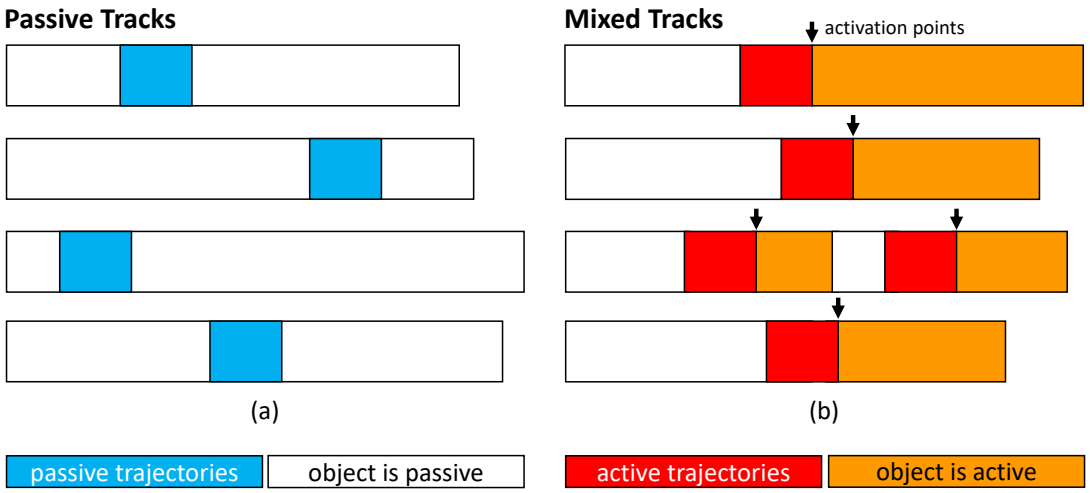


Figure 3.3: Passive (a) and mixed (b) tracks. Activation points are indicated by black arrows. The figure also illustrates the process of extracting passive and active trajectories discussed in Section 3.3.2.

annotated (or detected) across multiple subsequent frames of a video. All bounding boxes are related to the same object instance. Each bounding box is labeled as "active" if the user interacts with it or "passive" otherwise. We denote an object track as a tuple $\mathcal{T}_i = (C_i, \mathcal{B}_i, \mathcal{A}_i, \mathcal{F}_i)$, where $C_i \in \{1, \ldots, N\}$ is the object class label, $\mathcal{B}_i = \{b_1, b_2, \ldots, b_n\}, b_j \in \Re^4$ is the sequence of annotated bounding boxes, $\mathcal{A}_i = \{a_1, a_2, \ldots, a_n\}, a_j \in \{0, 1\}$ is the sequence of active/passive flags related to the bounding boxes in $\mathcal{B}_i$, and $\mathcal{F}_i = \{f_1, f_2, \ldots, f_n\}, f_j \in N$ are the IDs of the frames to which the bounding boxes $\mathcal{B}_i$ are related. Each bounding box $b_j \in \Re^4$ is represented by the four coordinates of the top-left and bottom-right corners. To generalize over different image sizes and aspect ratios, all coordinates are divided by frame width and height in order to be normalized between 0 and 1 and then centered around the normalized center point $(0.5, 0.5)$. Bounding boxes $b \in \Re^4$ are represented by the four coordinates of the top-left and bottom-right corners. To generalize over different image sizes and aspect ratios, all coordinates are divided by the frame dimensions in order to be normalized in the interval $[0, 1]$. Coordinates are then centered around the normalized center point $(0.5, 0.5)$. This let all coordinates range in the interval $[-0.5, 0.5]$. We divide object tracks into two categories: passive and mixed. Figure 3.3 illustrates the two considered types of object tracks. Tracks

$\mathcal{T}_i$ composed only by passive bounding boxes (i.e., $a_j = 0 \; \forall a_j \in \mathcal{A}_i$) are denoted as passive tracks. Tracks containing $\mathcal{T}_i$ both passive and active bounding boxes (i.e., $\exists \; a_h, a_k \in \mathcal{A}_i | a_h \neq a_k$) are denoted as mixed tracks. In this case, we refer to the points in which an object changes its status from passive to active as "activation points" (see Figure 3.3). Since we are interested in predicting next-active-objects, i.e., objects which are going to change their status from passive to active, we discard all tracks containing only active bounding boxes.

3.3.2 Active vs Passive Trajectory Classifier

We hypothesize that next-active-objects can be discriminated from passive ones by analyzing the egocentric object trajectories leading to the activation point. Therefore, we propose to train an active vs passive trajectory classifier in order to recognize next-active-objects by discriminating them from objects which will keep their passive status. We define a trajectory as a sequence of bounding boxes $T_i = \{b_1, b_2, \ldots, b_h\}$ related to video frames $F_i = \{f_1, f_2, \ldots, f_h\}$. We consider two classes of trajectories: active and passive. Active trajectories are those leading to a change of status from passive to active. Passive trajectories are related to passive objects that will maintain their passive status and hence they do not lead to any status change. While in principle we would like to predict next-active-objects arbitrarily in advance, we claim that the most discriminative part of active trajectories is the one immediately preceding the status change. Therefore, in order to train an active vs passive trajectory classifier, we consider fixed length trajectories of h-frames. Parameter h should be chosen carefully in order to include enough information for the discrimination while avoiding the noise due to long trajectories including data far away from the activation point. To compose a suitable training set, we extract passive and active trajectories from the aforementioned object tracks. Passive trajectories are randomly sampled from all passive tracks (we extract one trajectory per track). Active trajectories are sampled from mixed tracks by considering the last h frames preceding the activation point. Please note that, if possible, multiple trajectories are extracted from the same mixed tracks. Figure 3.3 illustrates the extraction of active (red) and passive (cyan) trajectories from object tracks.

We propose to describe trajectories including 1) the absolute positions in which bounding boxes appear in the frame, 2) differential information about positions, 3)

scale and differential information about scale. The main motivations behind point 1) is the observation that absolute position can help discriminate active from passive objects [19]. Point 2) is derived from the trajectory shape descriptor used within Dense Trajectories [123]. Point 3) is inspired by [124], where the derivative of the bounding box area is used to estimate Time to Contact. Each trajectory T_i is hence described as follows:

$$\begin{aligned}\mathcal{D}(T_i) = (&xc_1, yc_1, \ldots, xc_h, yc_h, s_1, \ldots, s_h,\\ &\Delta xc_2, \Delta yc_2, \ldots, \Delta xc_h, \Delta yc_h, \Delta s_2, \ldots, \Delta s_h)\end{aligned} \tag{3.1}$$

where xc_j and yc_j are the coordinates of the centers of the bounding box b_j, s_j is its area, $\Delta xc_j = (xc_j - xc_{j-1})$, $\Delta yc_j = (yc_j - yc_{j-1})$ and $\Delta s_j = (s_j - s_{j-1})$ encode differential information about position and scale. If the length of T_i is h, the dimension of the descriptor is $|\mathcal{D}(T_i)| = 6h - 3$.

3.3.3 Sliding Window Prediction

In order to predict which objects are going to become active and which are not over time, we use a sliding window approach. At each time step, the system analyzes the last h frames of the trajectories of each tracked object and classifies them as either active or passive. If an object has been tracked for less than h frames, it is discarded. For each analyzed object, the system draws a bounding box and assigns to it a confidence score equal to the probability given by the classifier. This way, likely next-active-objects will get a high score, while passive ones will retain a lower one. Figure 3.4 illustrates the proposed sliding window approach.

3.4 Experimental Settings

3.4.1 Dataset

We consider the ADL dataset for our experiments [19]. The ADL dataset contains several egocentric videos acquired using a chest-worn camera by 20 different subjects performing daily activities. Each video has been acquired at 30 fps. Each video is provided with annotations for 18 performed activities and 45 different object

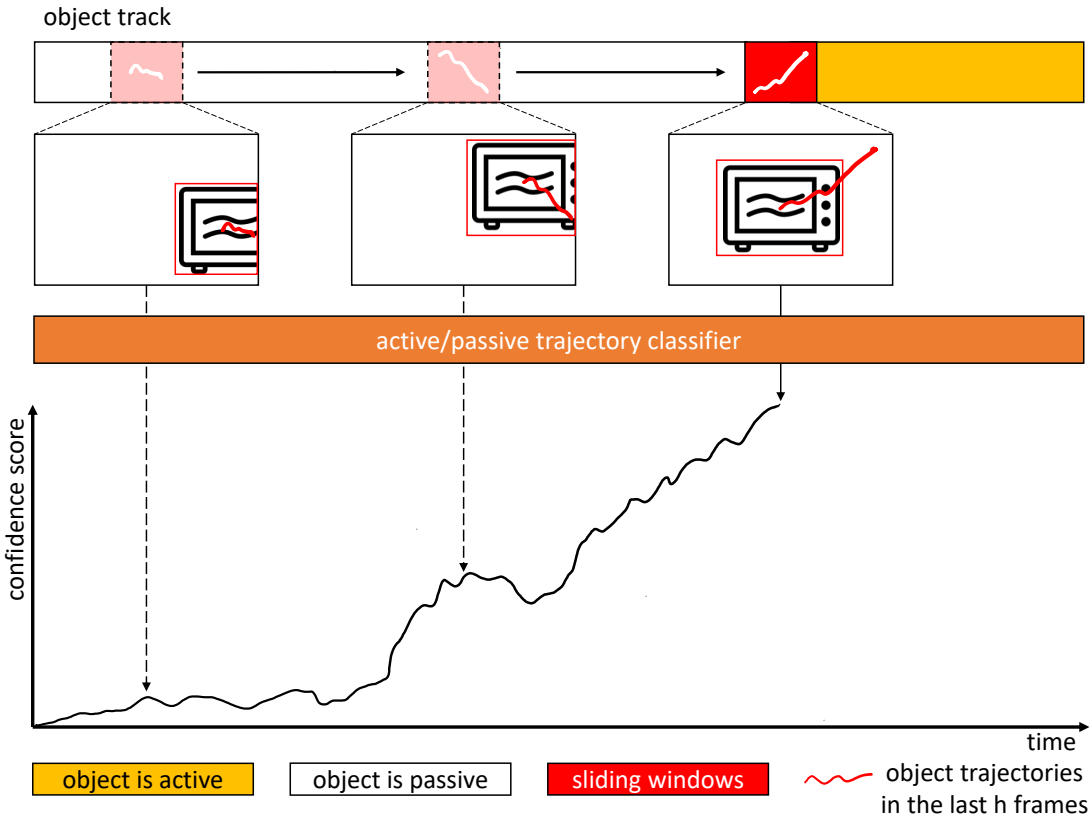


Figure 3.4: Sliding window processing of object tracks. At each time step, the trained binary classifier is ran over the trajectories observed in the last h frames and a confidence score is computed. We expect next-active-objects to be easier to detect when closer to the activation point.

classes (in the form of bounding boxes). Each object annotation is labeled as active if the user is interacting with it or as passive otherwise. Annotations related to the same object instance are grouped in object tracks. Figure 3.5 shows some annotated frames from the ADL dataset. We carry out our evaluations on the ADL dataset since it is the only publicly available dataset featuring untrimmed egocentric videos of object interactions "in the wild", including annotations for both active and passive objects. The main shortcomings of using the ADL dataset for the evaluations is that it contains data acquired using a chest-worn camera. While performing experiments also on a dataset acquired using a head-mounted camera would be useful to generalize our findings, it should be noted that acquiring and labeling such data is not trivial mainly because of the need for frame-wise annotations and the

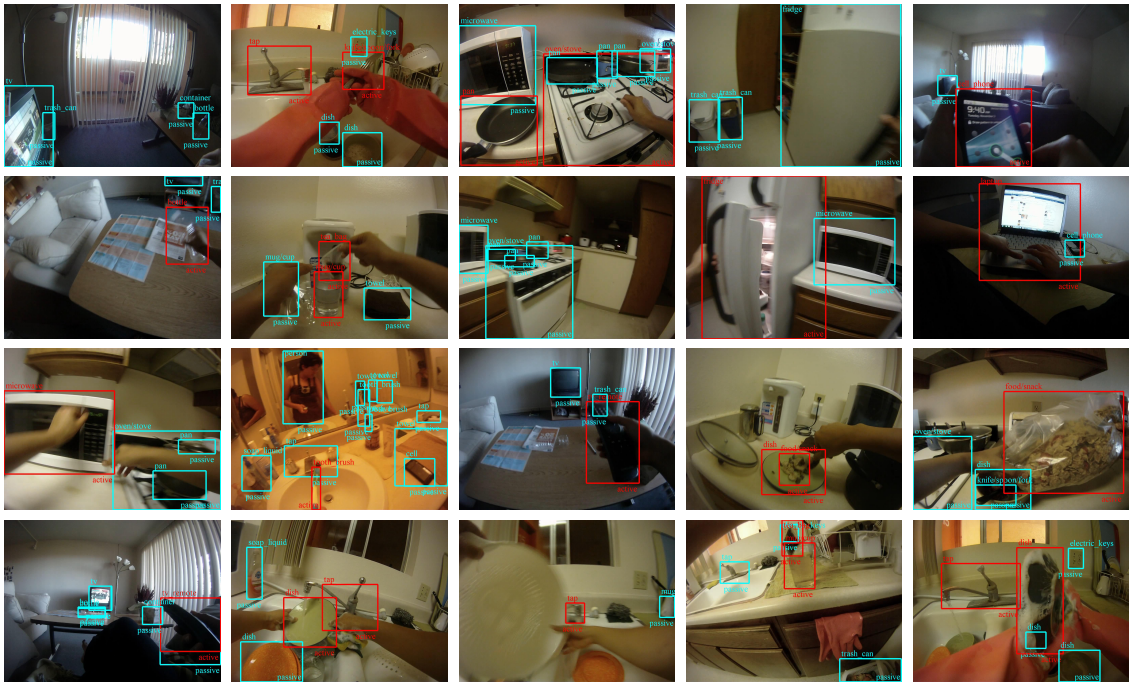


Figure 3.5: Some sample frames from the ADL dataset. Active object annotations are indicated in red. Passive object annotations are indicated in cyan.

difficulty in interpolating bounding boxes among neighboring frames due to head motion.

Unluckily, in the ADL dataset, objects are annotated every 30 frames, which makes reasoning about object trajectories difficult. To overcome this limitation, we temporally augment the original annotations by tracking objects in those frames which are not annotated. The tracker is always initialized using original ground truth object annotations and tracking is carried on until the next annotation is reached. Active/passive flags are interpolated accordingly. Note that tracking has to be performed only for a short time (1 second). To this aim, in our experiments, we use the short term tracker CMT proposed by Nebehay and Plugfelder in [125]. In order to account for objects which temporally disappear from the scene, object tracks are split into two parts every time the tracker is not able to track the object.

3.4.2 Object Detection and Tracking

To perform object detection, we consider the state of the art Faster R-CNN method [126] based on the VGG-16 network [127]. We follow the authors of [19], who consider

26 object classes including 21 passive objects and 5 active ones. Since in our work we propose to detect next-active-objects on the basis of their trajectories and not of their appearance, we don't train our object detector to distinguish between active and passive objects. Therefore we consider a corresponding dataset of 23 object classes. In this dataset, corresponding active and passive classes (e.g., active fridge and passive fridge) are merged into a single class (e.g., fridge). Considering that many samples are required in order to fine-tune the Faster-RCNN model, we remove 4 classes which are represented by less than 1000 images in the training set (the average number of annotated instances per object class in the dataset is around 4000). The final dataset contains 19 object classes: "book", "bottle", "cell phone", "detergent", "dish", "door", "fridge", "kettle", "laptop", "microwave", "mug/cup", "oven/stove", "pan", "pitcher", "soap/liquid", "tap", "tooth paste", "tv", "tv remote". As in [19], we train the object detector on images extracted from the first 6 videos, while the remaining 14 videos are used to train/test the proposed next-active-object prediction method. Note that, in order to train the object detector, we consider only the object annotations originally contained in the dataset, while tracked bounding boxes are discarded at this stage. The Faster R-CNN model is trained using the "end2end" procedure proposed in [126]. The trained detector achieves a mAP of 27.72 on the test set of 14 videos, which compares favorably with respect to the 15.15 mAP scored by the deformable part models employed in [19]. Please note that, as pointed out in [19], even performing object detection on the ADL dataset is hard due to the presence of small objects and non-iconic views. To obtain object tracks from detection, we use the lightweight SORT tracker proposed in [128]. The SORT tracker assumes that good object detections are available for each frame and performs object tracking by associating predicted bounding boxes into object tracks. Instead of using appearance-based features, the SORT tracker relies on the predicted bounding boxes and a simple motion model. The SORT tracker is highly real-time (260 Hz) and adds minimum overhead to the object detector component.

3.4.3 Trajectory Classification

We train Random Decision Forests to discriminate between passive and active object trajectories. In the considered dataset, the number of negative trajectories is usually far larger than the number of active ones. To mitigate such imbalance, at

training time, the number of passive trajectories is randomly subsampled to match the number of active ones, in order to obtain a balanced training set. Testing is always performed on the original unbalanced data. We assess the performances of the trained classifiers with respect to different factors, including the temporal support with respect to which trajectories are analyzed, the employed trajectory descriptor, the generalization to unseen object classes and the robustness of the classifier with respect to the distance from the activation point. All results are reported in terms of Precision-Recall curves and related Average Precision (AP) values. Please note that our trajectory classifier is trained independently from the object class, that is, a single classifier is learning from trajectories related to all object classes. The main reason of this choice, is that not enough data is contained in the considered dataset to train class-specific classifiers.

3.5 Results

We perform all our experiments in a leave-one-person-out fashion on the set of 14 videos which have not been used to train object detectors (as done in [19]). At each leave-one-out iteration, trajectory classifiers are learned on videos acquired by 13 subjects and tested on the remaining data. This makes sure that training and testing data are always acquired by different subjects. All reported results are averaged across the 14 leave-one-out iterations.

3.5.1 Performances of the Trajectory Classifier

The proposed trajectory classifier based on the descriptor introduced in Eq. (3.1) achieves best results setting $h = 30$ (which corresponds to 1 second in the considered dataset). Specifically, in the leave one out evaluation, our method scores an AP of 0.28, while the chance level is 0.09. In the following, we perform comparisons to motivate the design of the proposed system based on fixed-length trajectories and the selection of parameter h. All comparisons are based on the descriptor introduced in Eq. (3.1).

In Section 3.3.2, we assumed that the last part of an active trajectory is the most discriminative for our task. Therefore, we proposed a sliding window approach which analyzes fixed-length trajectories within a temporal window of size h. To support

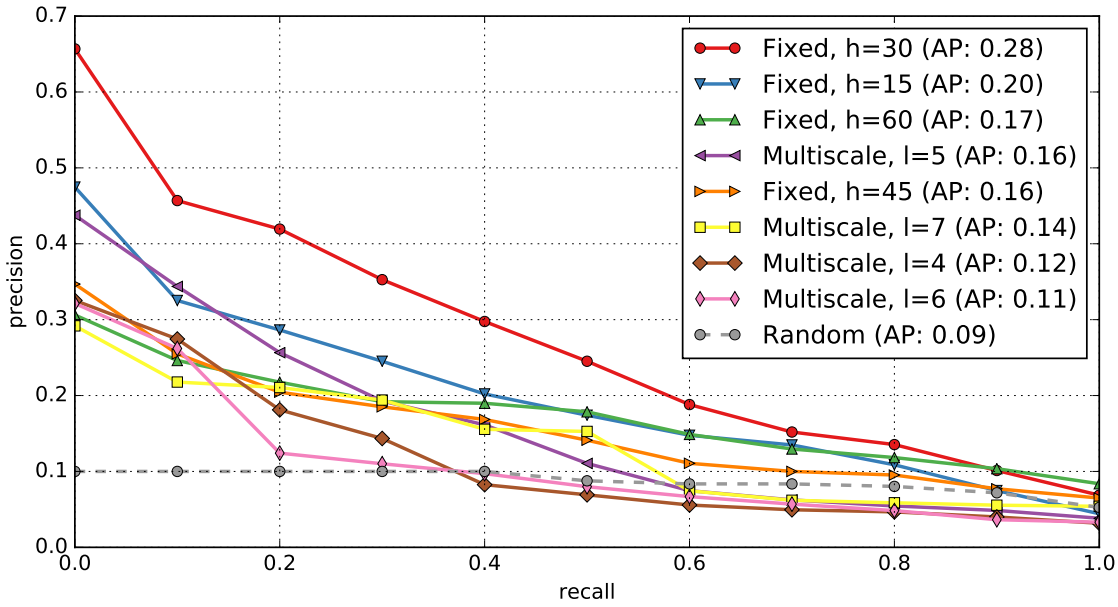


Figure 3.6: Precision-recall curves related to different trajectory description schemes. Average precision values are reported in parenthesis the legend. Elements in the legend are sorted by average precision in descending order.

that analyzing trajectories within a fixed-length temporal window is optimal, we compared the proposed method to a different schema which, at each time step, analyzes the whole trajectory observed up to that point. In this second schema, in order to obtain a fixed-length descriptor, trajectories are represented with a multiscale approach. Using a temporal pyramid with l levels, each trajectory is divided into $2^l - 1$ segments. Bounding boxes within the same segment are averaged and the results concatenated. This leads to fixed-length trajectories which are hence represented using the descriptor introduced in Eq. (3.1). Note that the maximum number of splits operated by the temporal pyramid is equal to $2^{(l-1)}$, therefore, trajectories shorter than this number are discarded in our experiments.

Figure 3.6 reports precision-recall curves of the classifiers learned on trajectories exacted according to the two considered schemes. The proposed fixed-length trajectory approach has been evaluated considering different lengths $h = \{15, 30, 45, 60\}$. Similarly, the multiscale approach has been evaluated considering different number of levels $l = \{4, 5, 6, 7\}$. Please note that the minimum trajectory lengths associated to the considered numbers of levels are respectively $\{8, 16, 32, 64\}$. The random

baseline is obtained performing classification with a binary random decision. As can be observed in Figure 3.6, classifiers based on fixed-length trajectories tend to outperform methods based on multiscale trajectories. This suggests that the last part of active trajectories is the most discriminative and that motion information too far away from the activation point introduces noise in the observations. Among the methods based on fixed-length trajectories, the best performing scheme is the one analyzing trajectories of length $h = 30$. This value will be used in all the following experiments.

3.5.2 Trajectory Descriptors

As discussed in Section 3.3.2, the proposed trajectory descriptor introduced in Eq. (3.1) includes information about absolute positions and scales, as well as differential information about position and scale. We analyze the impact of each of these kinds of information comparing the proposed descriptors against the following baselines:

- **Motion Magnitude**: we consider discriminating active trajectories from passive ones on the basis of the amount of motion characterizing the trajectory T_i under analysis. The amount of motion is measured as the sum of the magnitudes of the displacement vectors: $M(T_i) = \sum_{j=2}^{h} \sqrt{\Delta xc_j^2 + \Delta yc_j^2}$. Classification is hence performed by thresholding on M. The optimal threshold is selected at training time as the best discriminating active from passive trajectories in the training set;

- **Relative Trajectories**: are the descriptors proposed by Wang et al. in their work on Dense Trajectories [123]: $\mathcal{D}(T_i) = \frac{(\Delta xc_2, \Delta yc_2, \ldots, \Delta xc_h, \Delta yc_h)}{\sum_{j=2}^{h} \sqrt{\Delta xc_j^2 + \Delta yc_j^2}}$. These descriptors encode only the "shape" of the trajectory and do not include any information about absolute positions;

- **Absolute Trajectories**: described as the concatenation of the centers of all bounding boxes: $\mathcal{D}(T_i) = (xc_1, yc_1, \ldots, xc_h, yc_h)$. Such descriptors include positional information but do not encode scale and differential information;

- **Absolute Trajectories + Differential Positions**: described as the concatenation of positions and differential information about position: $\mathcal{D}(T_i) =$

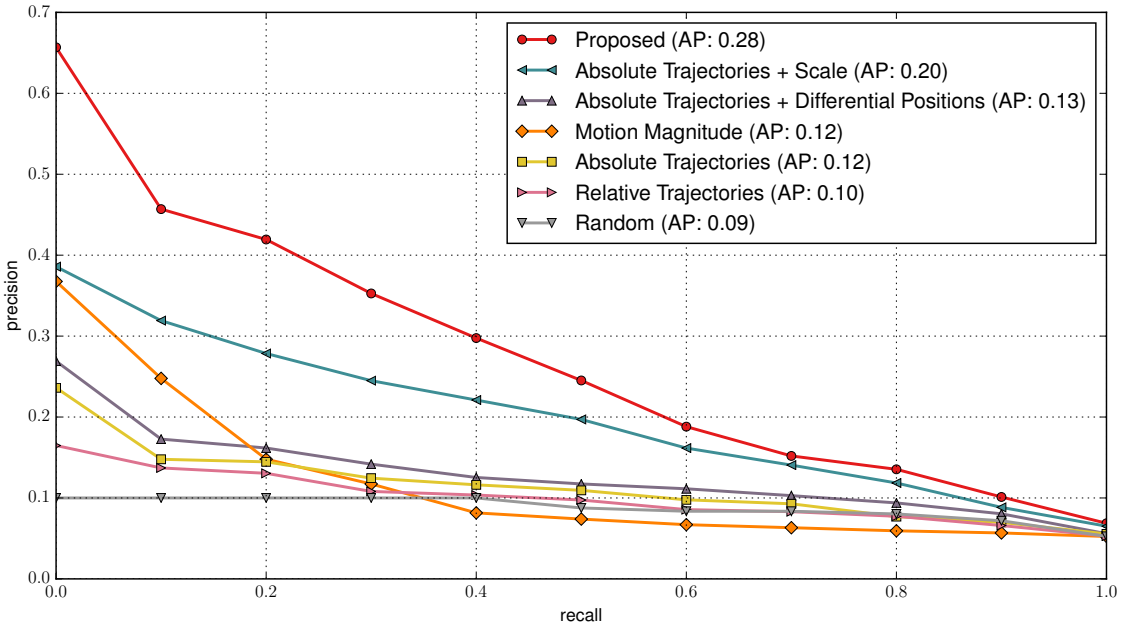


Figure 3.7: Precision-recall curves related to the proposed method and compared baselines. Average precision values are reported in parenthesis the legend. Elements in the legend are sorted by average precision in descending order.

$(xc_1, yc_1, \ldots, xc_h, yc_h, \Delta xc_2, \Delta yc_2, \ldots, \Delta xc_h, \Delta yc_h)$. These descriptors encode location and trajectory shape but do not include scale information;

- **Absolute Trajectories + Scale**: described as the concatenation of positions and bounding box scales: $\mathcal{D}(T_i) = (xc_1, yc_1, \ldots, xc_h, yc_h, s_1, \ldots, s_2)$. These descriptors encode location and scale but do not include differential information.

Figure 3.7 shows precision-recall curves for the proposed method and the compared baselines. As can be observed, relative trajectories (AP: 0.10) are less discriminative than absolute trajectories (AP: 0.12) for the next-active-object prediction task. This confirms the observation according to which position can help discriminate active and passive objects [19]. Combining absolute and differential positional information improves performances marginally (AP: 0.13). Adding scale (AP: 0.20) and above all, combining with differential information as we propose (AP: 0.28), allows to obtain the best results. Interestingly the motion magnitude baseline performs better than some competitors (AP: 0.12).

	AP	
Object	**w/o**	**with**
oven/stove	0.60	0.82
tap	0.47	0.59
door	0.19	0.18
tv remote	0.58	0.73
bottle	0.50	1.00
pan	0.56	0.83
tv	1.00	1.00

	AP	
Object	**w/o**	**with**
fridge	0.47	0.73
book	1.00	1.00
microwave	0.67	0.40
kettle	0.33	0.75
mug/cup	0.52	0.62
dish	0.51	0.32
laptop	0.63	0.65

Table 3.1: Average precision results related to the leave-one-object-out experiment.

3.5.3 Generalization to Unseen Object Classes

We have trained a single active vs passive classifier including data from all considered object classes. While training object-specific trajectory classifiers might be advantageous, the limited number of samples related to a single object class could pose a challenge. Moreover, a real system needs to be able to handle situations in which previously unseen objects may become active. We find that next-active-object trajectory classification can generalize to previously unseen object classes. To assess this property, we performed a leave-one-object-out experiment. For each object class, we trained trajectory classifiers on data related to all other object classes. Classifiers have been hence tested on data including only the object class which was removed from the training set.

Table 3.1 reports the results for the considered object classes. Classes missing from Table 3.1 are those which were not represented by any sufficiently long trajectory (at least h frames) in the dataset. Classifiers learned from training sets not containing the target object class ("w/o" column) are compared to classifiers learned from training sets containing also instances from the target object class ("with" column). Similar performances are achieved for many object classes (e.g., door, tv, book, mug/cup, laptop), whereas for others the learning from the instances of the same object class is more beneficial. On average, removing the object class from the training set implies a reasonable performance loss of 0.11 AP.

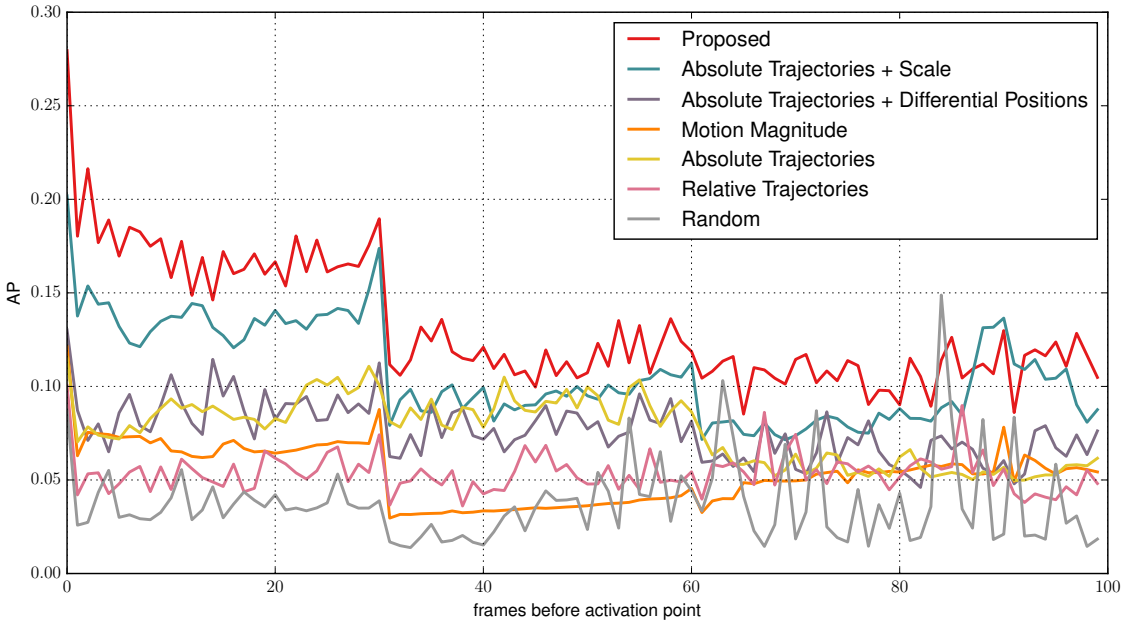


Figure 3.8: Performances of trajectory classifiers as a function of the distance from the activation point. Average precision values are reported in parenthesis the legend. Elements in the legend are sorted by average precision in descending order.

3.5.4 Robustness to Distance from Activation Point

In the proposed system, next-active-objects are predicted using a sliding window which analyzes the last h frames of each observed object trajectory. However, trajectory classifiers have been trained on the h frames immediately preceding the activation point. We perform experiments to assess how many frames before the activation point we can predict next-active-objects from egocentric video. Figure 3.8 shows the Average Precision of the considered trajectory classifiers as a function of the distance from the activation point at which the sliding window is placed to perform predictions. All trajectory classifiers work best when they analyze trajectories which are close in time to the activation point, while performances decay when they analyze trajectories which are observed far from it. All methods keep reasonable performances when the distance from the activation point is less than one second, while performances decay afterwards. The proposed trajectory descriptor outperforms the others and is generally above chance level.

3.5.5 Comparative Experiments

In order to compare different methods in a common evaluation scheme, we frame next-active-object prediction as an object detection task. We assume that, at each time step, each method produces a series of bounding boxes around predicted next-active-objects and assigns a confidence score to them. We define our ground truth starting from the object annotations of the ADL dataset augmented by tracking as described in Section 3.4.1. Since we wish to predict next-active-objects as soon as possible, all annotations which are on the passive segments of a mixed track (see Figure 3.3) are considered as valid detections. All other annotations, namely, the ones which are on passive tracks and the ones which are in the active part of mixed tracks are not considered valid detections. The performances of the investigated methods are measured computing precision-recall curves and Average Precision (AP) values as defined in [129]. A prediction is considered correct if there is a significant overlap (area of intersection over union (IOU) ≥ 0.5) with an annotation of the same object class. We compare the proposed method with respect to a series of baselines:

- **Motion Magnitude**: the same baseline discussed in Section 3.5.2 based on thresholding over motion magnitude;
- **Relative Trajectories**: the same baseline discussed in Section 3.5.2 based on the trajectory descriptors introduced by Wang et al. [123];
- **Center Bias**: this baseline considers the assumption made by Pirsiavash and Ramanan [19], according to which active objects tend to appear near the center of the frame. The baseline analyzes the object detections produced by the Faster-RCNN detector and takes into account the confidence score assigned to each predicted bounding box s_o. For each detected object, we compute a score s_c which is inversely proportional to its distance from the center of the frame. The final confidence score is obtained as $s = s_c \cdot s_o$;
- **Hand Bias**: the presence of hands is a cue often considered for detecting active objects [18, 43, 35, 36]. To leverage this cue, we detect hands from the input videos by using the models proposed in [130]. Similarly to the center bias baseline, for each object detection we compute two scores s_{lh} and s_{rh} which are inversely proportional to the distances of the object from the left

and right hand respectively. If one of the two hands is missing, a score equal to zero is assigned. The final confidence score is obtained by $s = s \cdot (s_{lh} + s_{rh})$;

- **Active/Passive Objects**: a method inspired by the work of [19]. Predictions are obtained using a Faster R-CNN object detector trained to detect active and passive objects separately. The detector is hence trained on 38 classes (19 active objects and the corresponding 19 passive ones);

- **Saliency-Based Models**: this set of baselines follow Damen et al. [76], who propose to detect task relevant objects using a gaze tracker, exploiting the anticipatory nature of eye gaze fixation [131]. Since we do not assume the availability of a gaze tracker, we implement such baselines using saliency prediction models. The baseline works as follows. Saliency maps are first extracted from each frame. Starting from the Faster-RCNN detections, each predicted bounding box is assigned a score equal to the mean saliency value within the bounding box. Given the different levels at which saliency is defined [132], we consider the model proposed by Vig et al. [133] for eye fixation prediction, the model proposed by Seo et al. [134] for dynamic saliency from videos, and the model proposed by Zhang et al. [135] for salient object segmentation;

- **Random**: starting from the Faster-RCNN detection, each bounding box is assigned a random score in the interval $[0, 1]$.

Figure 3.9 reports the precision-recall curves scored by our method and all baselines. To reduce computational burden, the methods indicated by the "*" symbol have been evaluated on a subset of the data obtained taking one frame every 30 frames. The proposed method is the best performing one (AP: 0.0680), followed by the motion magnitude (AP: 0.0478) and relative trajectory baselines (AP: 0.0437). It is worth noting that, the best performing methods are all based on egocentric object motion. The method based on center bias outperforms the appearance-based baseline derived from [19] (0.0412 vs 0.0298 AP values). Our main insight about this behavior is that object appearance is likely to change while the object is being manipulated rather than before. The baseline based on hand bias does not achieve good performances (AP: 0.0200). This is probably due to different factors. First, detecting hands in unconstrained egocentric videos is not trivial [130]. Second,

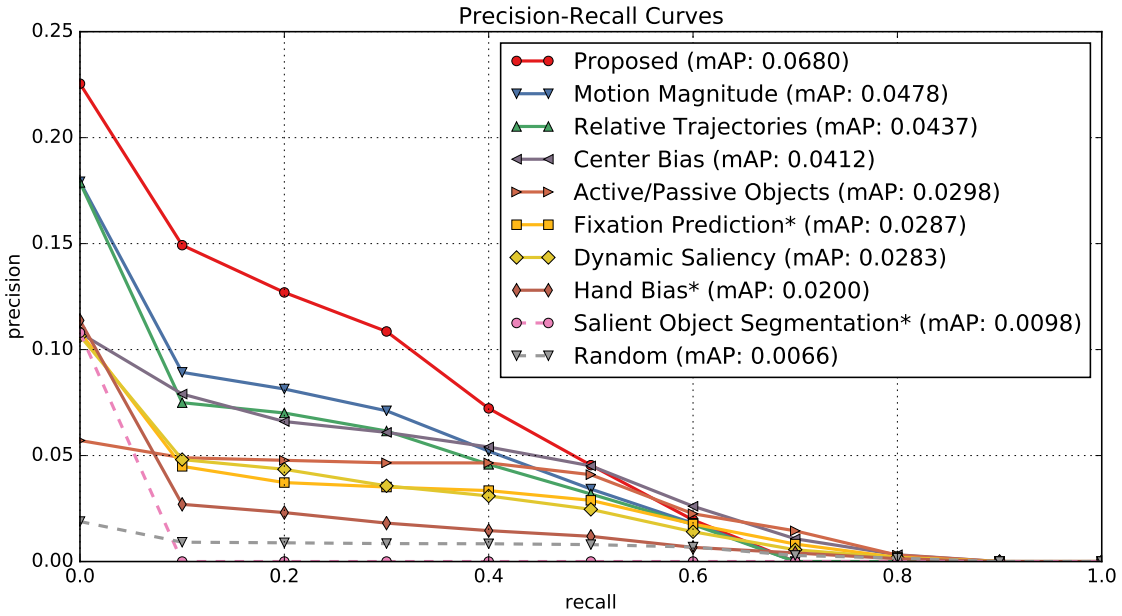


Figure 3.9: Precision-recall curves of the compared methods. Average precision values are reported in parenthesis the legend. Elements in the legend are sorted by average precision in descending order. Methods indicated by "*" have been evaluated on a subset of the data obtained taking one frame every 30 frames.

hands are not always visible until the object manipulation actually begins. Saliency-based baselines perform worse than others. It should be noted that such methods have been designed to predict current and not future visual attention mechanisms and that such methods have not been specifically designed for the egocentric scenario. Moreover, while we perform our evaluations on the ADL dataset, which have been acquired using a chest-worn camera, the state-of-the-art has been designed for head-mounted cameras. In particular, attention-based methods, might be unable to leverage head-motion cues as expected.

Figure 3.10 and Figure 3.11 report some visual examples of success/failure sequences related to the proposed method. Positive model predictions are indicated in green, while negative predictions are indicated in cyan. The examples also report the observed egocentric object trajectories, the predicted class and the confidence score. Ground truth next-active-objects are reported in red. In the examples of correct predictions (Figure 3.10), the model correctly assigns a hight score (positive prediction) to next-active-objects and a low score (negative prediction) to passive ones. In the failure examples (Figure 3.11), the model predicts the wrong object or

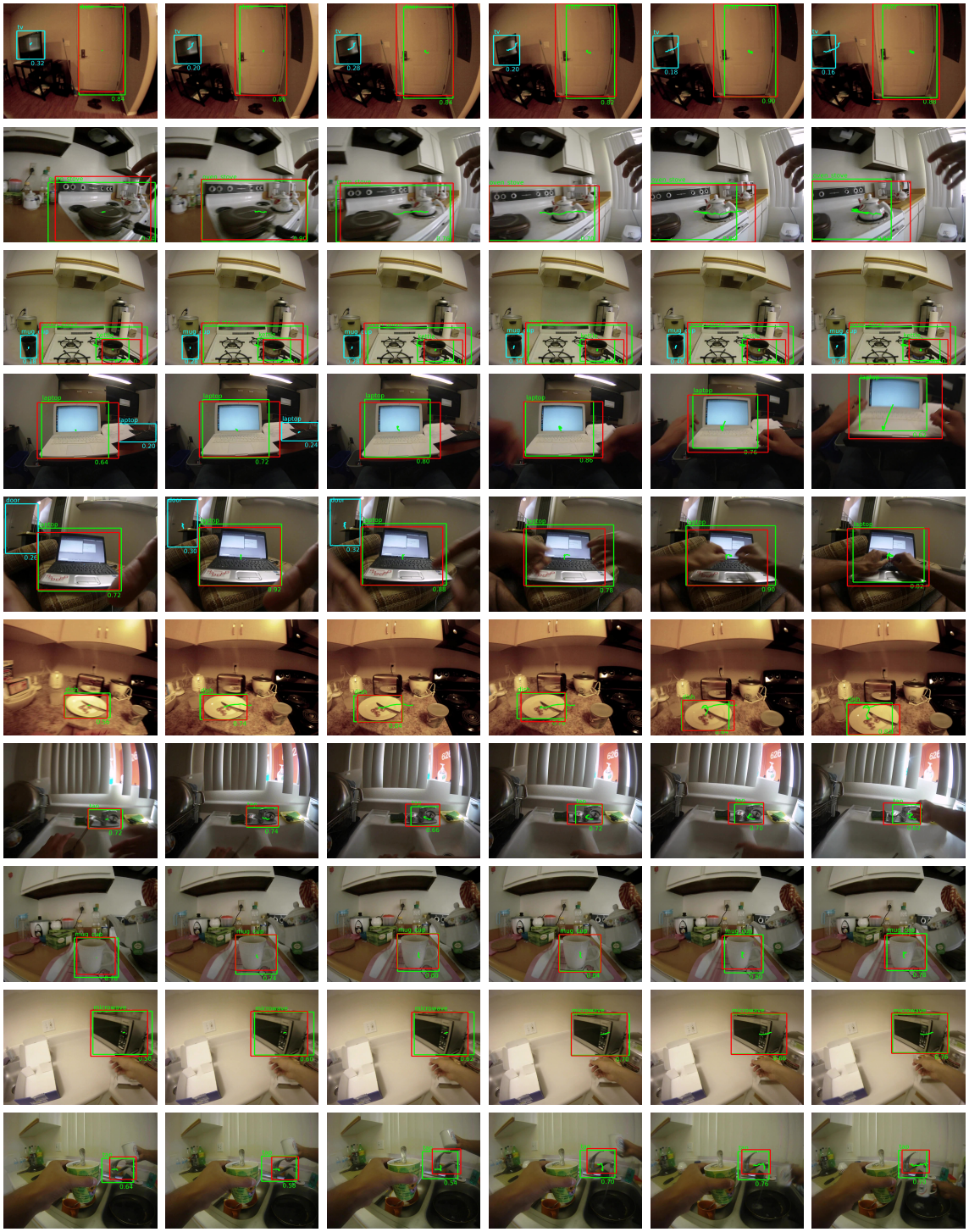


Figure 3.10: Some success examples of the proposed method. Red bounding boxes represent ground truth next-active-objects. Positive predictions are indicated in green, while negative predictions are indicated in cyan. For each prediction, the object class and confidence scores are reported.

Figure 3.11: Some failure examples of the proposed method. Red bounding boxes represent ground truth next-active-objects. Positive predictions are indicated in green, while negative predictions are indicated in cyan. For each prediction, the object class and confidence scores are reported.

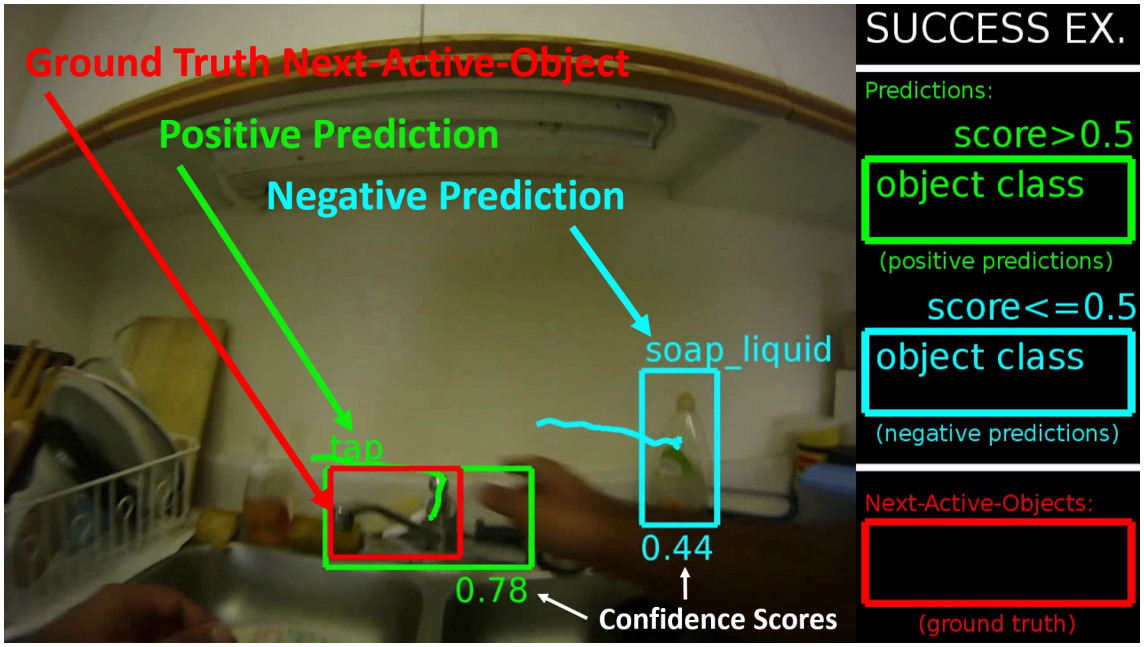


Figure 3.12: A frame from one of the two demo videos included in the supplementary material. The videos show several sequences along with he predicted next-active-objects. Ground truth next-active-objects are indicated in red. Positive model predictions are indicated in green, while negative predictions are reported in cyan. For each detected next-active-object we report the predicted class, the observed trajectory and the computed confidence score.

fails to detect all next-active-objects.

We also provide demo videos of correct predictions and failure examples which can be downloaded at the URL http://iplab.dmi.unict.it/N-A-O/. Figure 3.12 reports a sample frame from one of the videos.

3.6 Discussion

In this chapter, we have introduced and investigated the problem of next-active-object prediction from egocentric videos. While the task is not trivial in unconstrained settings, we have shown that egocentric object trajectories provide a useful cue to address the challenge. Experiments have highlighted that 1) the last part of active egocentric object trajectories is the most suitable to predict next-active-objects, 2) active trajectory classifiers can generalize to unseen object classes up

to a given extent, 3) egocentric cues based on object motion outperform baselines based on static observations on the considered dataset. In future works, we will investigate how the task of next-active-object prediction can be exploited for early action prediction and how such integration can be beneficial for both tasks.

Chapter 4

Conclusion

The main contributions of this thesis are related to context awareness in First Person Vision. Our investigation has been driven by the observation that, differently from traditional Third Person Vision, data acquired by First Person Vision systems is very related to the user and hence it can be used to provide assistance in a "personal way" and predict the intent of the user [10]. Within the broad scope of context awareness in First Person Vision, we have investigated two of the five context categories discussed in Section 1.1.3, namely *location* and *intent*.

Chapter 2 investigated personal location recognition from egocentric videos. Differently from previous works, we considered personal locations at the instance level (e.g., my office), rather than at the category level (e.g., an office). We considered a real scenario in which the user is willing to monitor a selected set of personal locations of interest and proposed a suitable definition of the task. Our definition involves that 1) the user provides minimum training data for the locations he wants to monitor, 2) the system has to deal with the rejection of negative locations. To investigate the problem, we proposed three datasets of egocentric videos acquired in 10 personal locations and performed a benchmark of different wearable cameras and representation techniques.

Our investigation pointed out the following:

- Recognizing personal locations of interest from egocentric videos involves some specific challenges. The two main challenges are: 1) supervised learning can rely only on few training data provided by the user, 2) the system has to correctly reject negative locations learning only from positive samples;

- Head mounted, wide angle cameras have a significant advantage over other camera designs on the personal location recognition task;

- Representations based on deep learning outperform other representation methods due to their transfer learning abilities. However, fine-tuning Convolutional Neural Networks is not trivial with small datasets and many architectural settings can be tuned to improve performances;

- The assumption of temporal coherence between neighboring predictions arising from egocentric data can be used to 1) formulate affective negative rejection methods, 2) improve location recognition in neighboring frames;

- Due to the large variability in terms of visual content that wearable cameras are likely to acquire, learning a policy for the rejection of negative locations directly from negative samples is not trivial. Specifically, we show that designing a robust rejection option is advantageous over explicitly learning the "negative location class" from negative samples.

Chapter 3 proposed the problem of predicting next-active-objects from egocentric videos. While predicting the future is in general hard, we argued that the First Person Vision paradigm can provide important cues to address the challenge. Specifically, we investigated the predictive power of egocentric object trajectories as a means for encoding information about the dynamic of the scene and proposed a system to perform next-active-object prediction. Our investigation pointed out that:

- Predicting next-active-objects from egocentric videos is not trivial. However, egocentric object trajectories provide a useful cue to address the challenge;

- Describing the shape of egocentric object trajectories without including information on absolute positions is not enough for next-active-object prediction. Better results are obtained including 1) absolute positions, 2) differential information about position and scale;

- The last part of an active egocentric trajectory is the most discriminative for the task of next-active-object prediction. Including trajectory information too far away from the activation point seems to add only noise to the observations;

- Active vs passive trajectory classifiers can be trained independently from object classes;

- Cues based on appearance of objects, their distance from the center of the frame, presence of hands and saliency models are not effective for the considered task. Our main insight into these results is that such factors are not relevant until the object interaction actually takes place.

4.1 Future Directions

First Person Vision systems are characterized by their intrinsic mobility and their ability to acquire visual information which is very personal for the user. Therefore, we argue that context awareness constitutes a big challenge and opportunity for such systems. In this thesis, we have investigated some aspects related to context awareness in FPV systems. Our investigation has been guided by the assumption that context can be more than mere location sensing and it can encode many other aspects related to the user such as, for instance, his intent.

Many challenges still need to be faced both in location sensing and intent understanding. Location sensing methods need to be improved in terms of accuracy and usability. This can be achieved designing better negative rejection methods and algorithms able to learn from few samples and with little supervision. With modern data-hungry methods such as those related to Deep Learning, a promising direction would be to leverage data from multiple users to improve both location detection and negative rejection.

The ability to predict the intent of the user is likely to be an important feature for modern wearable systems. While anticipating interactions with objects is an important feature, advanced system should be able to take into account different aspects, including location sensing, anticipation of object interactions, and forecasting of future goals.

In conclusion, our main insight is that context is complex and its understanding can allow for the construction of more sophisticated human-interaction mechanisms. In this sense, modeling context can serve as an occasion to improve the intelligence of First Person Vision systems. To build better models of context, future investigations should take into account, not only the study, but also the integration of different aspects such as location, user behaviors, object and scene accordances, attention and future anticipation.

Appendix A

Wide-Angle Sensors and Feature Extraction[1]

First Person Vision systems are deemed to be able to "see what the actor sees" in order to sense the world from his perspective [10]. To conform the human visual system, they should be able to acquire a large enough quantity of visual information related to the surrounding environment. This is usually done employing wide-angular cameras which can acquire a large part of the scene at the cost of introducing radial distortion. Many wearable cameras such as GoPro[2], Authographer[3] and Narrative Clip 2[4] employ wide-angle cameras to achieve this result. In [24, 20] we show that wide-angular cameras have a clear advantage over standard narrow-angle ones when modeling the visual context of the user in a First Person Vision application. Depending on the extent of radial distortion characterizing the acquired images, it is usually necessary to explicitly account for the geometric distortion introduced by wide-angle sensors during the feature extraction process [136]. While the standard way to deal with such distortion is to explicitly compensate for it [137], direct approaches not requiring any specific coordinate remapping and interpolation process are preferable in many cases [136].

In this chapter, we investigate how feature extraction can be performed directly on wide-angular images. We review wide-angle cameras in Appendix A.1 and fisheye camera models (which are a specific class of wide-angle sensors) in Appendix A.2. We

[1]All the work presented in this Chapter has been performed in collaboration with ST-Microelectronics Catania within the project PANORAMA, co-funded by grants from Belgium, Italy, France, the Netherlands, the United Kingdom, and the ENIAC Joint Undertaking.

[2]http://gopro.com

[3]http://www.autographer.com/

[4]http://getnarrative.com/

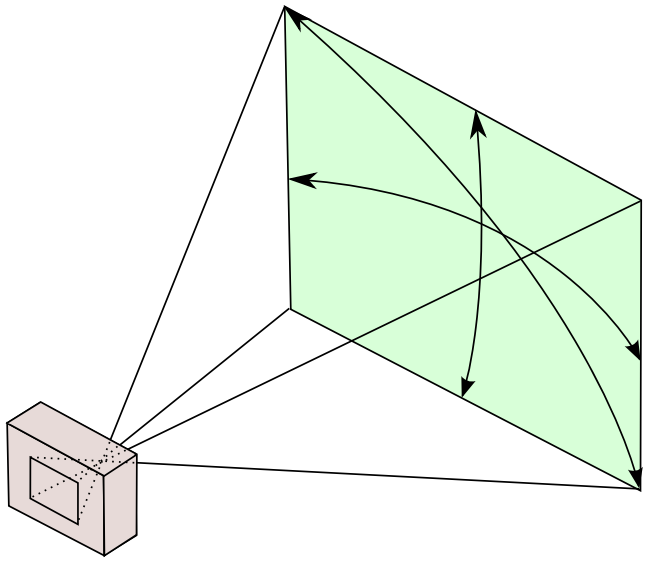


Figure A.1: Field Of View (FOV) of an image acquisition system. FOVs can be measured horizontally, vertically and diagonally.

present the experimental datasets used in the rest of this chapter in Appendix A.3. In Appendix A.4 we analyze how affine covariant region detectors can be applied efficiently directly on wide-angular images when the source camera is unknown and hence it cannot be calibrated. In Appendix A.5 we introduce a family of distortion adaptive Sobel filters for the direct estimation of the gradient of distorted images. In Appendix A.6 we present the Distortion Adaptive Descriptors which allow to compute gradient-based descriptors, such as SIFT [7] and HOG [138], directly on fisheye images. Finally Appendix A.7 summarizes the findings of this chapter.

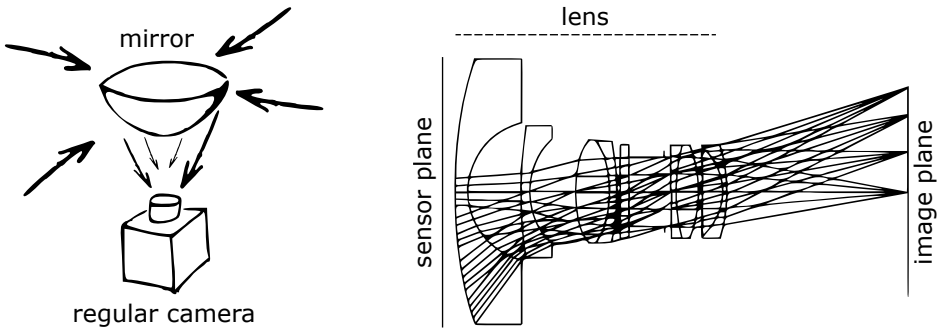
A.1 Wide Angle Sensors

Each image acquisition system can be characterized by its Field Of View (FOV), which is defined as the solid angle through which the system is sensible to the incoming light. FOVs can be measured horizontally, vertically or diagonally. Figure A.1 shows an illustration of how the Field Of View can be measured. The normal human binocular FOV is about 180° horizontally and 120° vertically [139]. Most regular cameras are designed to follow the perspective projection, which characterizes the ideal model of the pinhole camera. Such model has the convenient property to map

Figure A.2: From left to right: examples of images characterized by different FOVs and increasing rates of radial distortion. As can be noted, larger FOVs allow to acquire a larger portion of the scene at the cost of larger degrees of radial distortion. The examples have been obtained artificially adding different degrees of radial distortion to a high resolution source rectilinear image. More details on this process will be discussed in Appendix A.2.3.

lines that appear to be straight in the real world, to straight lines in the final image, thus producing a representation of the scene which is coherent with our visual perception. Due to the adherence to the perspective model, most regular cameras available on the market cannot achieve large FOVs (the FOV of most perspective cameras cannot exceed 140°) [139]. This shortcoming has motivated the design of a different class of sensors usually referred to as wide-angular or omni-directional visual systems [140]. Such systems are available in different designs and allow to obtain wider FOVs up to 180° and 360°. However, as pointed out in [141], this flexibility comes at the cost of the introduction of noticeable radial distortion, as it is depicted in Figure A.2. Since it is not possible to project an hemisphere on a finite plane using the perspective projection, different projection functions are usually considered when designing such systems. Wide-angle cameras can be built following two main designs: catadioptric [140, 142, 143] and dioptric [137, 141]. Figure A.3 illustrates the two camera designs for wide-angular cameras. Catadioptric systems employ a concave mirror to project a large FOV representation of the scene to a regular camera following the perspective projection. In this case the introduced radial distortion is determined by the specific geometry of the mirror. Dioptric systems simply substitute the regular lens of perspective cameras with lenses following a different design, generally referred to as "fisheye lenses". In this case, radial distortion is determined by the different projection function that the lens is designed to follow.

Characterizing the radial distortion introduced by wide-angle cameras, in order to be able to map points on the scene to points on the image, can be useful to remove radial distortion [144, 145, 146], extract features [136, 147, 148], and perform higher level tasks such as human and object detection [149, 150]. For this reason, different camera models and calibration techniques have been proposed to establish a



(a) catadioptric systems (b) dioptric systems

Figure A.3: Two main designs for wide-angular cameras: (a) catadioptric system combining a mirror and with regular camera and (b) dioptric systems using a fisheye lens.

mapping between the distorted wide angle images and their ideal purely perspective counterparts. Some calibration techniques require a special pattern to be present in the scene [151, 152] while others just require a few images of the scene and no other information [144, 145]. In the following sections, we discuss camera designs and the main mathematical models which can be used to describe the related image formation processes.

A.1.1 Catadioptric Systems

Catadioptric systems combine a concave mirror and a regular camera to achieve large FOVs. According to this design, the mirror diverts light rays coming from the scene to a regular camera, which acquires them. Figure A.3(a) illustrates the operation of such a design, while Figure A.4(a) shows a sample image obtained using a catadioptric system. The shape of the mirror allows to obtain the desired FOV deviating light rays non-linearly as a function of the angle formed with the optical axis [140, 142]. Even if different mirror shapes can be employed when designing catadioptric sensors, a few classes of mirrors are commonly used: parabolic, hyperbolic and elliptic mirrors. Camera models for catadioptric sensors generally have to take into account the geometry of the mirror. Nevertheless, under given circumstances (i.e., for central systems), unified models can be defined [143]. The sphere camera model [140, 143], in particular, allows to describe standard perspective cameras as well as catadioptric systems making use of hyperbolic, parabolic and elliptic mirrors.

(a) catadioptric systems (b) dioptric systems

Figure A.4: Sample images obtained using: (a) a catadioptric and (b) a dioptric system.

According to such model, central catadioptric systems (i.e., those characterized by a single effective viewpoint [143, 153]) are isomorphic to a projective mapping from the unit sphere to the plane. Figure A.5 illustrates the sphere camera model. The image formation process is hence modeled by the following projection:

- A 3D point $\mathbf{Q}$ of the scene is projected to two antipodal points on the unit sphere $\mathbf{s}_+$ and $\mathbf{s}_-$;
- Given a projection center situated on a sphere diameter at a distance ξ from the center of the sphere, the antipodal points are projected to an image plane placed at the focal distance f from the projection center. The two projected points are denoted by $\mathbf{q}_+$ and $\mathbf{q}_-$.

The model is characterized by the parameter ξ, which defines the geometry of the mirror. While catadioptric systems are mainstream in the robotic literature, their cumbersome design and the presence of a non-illuminated spot in the center of the image, make them unsuitable for FPV applications.

A.1.2 Dioptric Systems

Dioptric systems are obtained replacing the standard lens of a perspective camera with a fisheye lens. While standard lenses are designed to adhere to the perspective projection, fisheye lenses divert rays non-linearly with respect to the angle formed

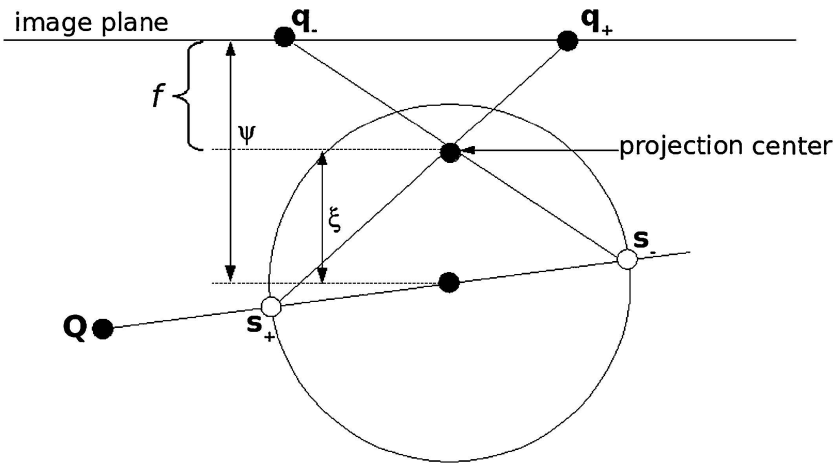


Figure A.5: Sphere camera model.

with the optical axis. The result of such projection are images characterized by a wider Field Of View and noticeable radial distortion. Figure A.3(b) depicts the image formation process of a dioptric system, while Figure A.4(b) shows a sample image obtained using a dioptric system. The Field Of View characterizing a given fisheye camera depends on the design of the lens, its focal length and the sensor size. Two configurations are particularly relevant: full frame and full circle [146]. Full frame images are characterized by a diagonal FOV equal to 180°. Such configuration is convenient since it allows to get the largest FOV which still allows to cover the full sensor. This means that the whole sensor is illuminated and the image does not contain dark non-illuminated areas. Full circle images are characterized by a vertical FOV equal to 180°. Such configuration does not allow full coverage of the sensor (the image is formed on a circular region in the center of the sensor), but allows to obtain the projection of the full hemispheric field on the final image. Figure A.6 shows some synthetic examples of the two configurations. Fisheye cameras are usually more compact than catadioptric systems and therefore they are more suited to be used in many application scenarios including First Person Vision systems. Moreover, images acquired using these cameras do not exhibit a dead spot in the center as happens for catadioptric systems (see Figure A.4). Therefore, in the rest of this chapter, we will focus mainly on dioptric systems. This type of systems are usually referred in literature as fisheye cameras.

(a) rectilinear (b) full frame (c) full circle

Figure A.6: Examples of perspective (a), full frame (b) and full circle (c) images. The two fisheye images are obtained by artificially adding different amounts of radial distortion to the rectilinear image (a).

A.2 Fisheye Camera Models

As discussed earlier, fisheye cameras allow to achieve a large Field Of View (FOV) by performing a non-uniform spatial sampling of the incoming light, which introduces radial distortion. The result is a non-Euclidean representation of the environment such that straight lines in the scene are not mapped to straight lines in the image [139, 141]. Since many applications assume that the input images are the result of the perspective projection of the scene to a finite plane, geometrical considerations are often needed when processing wide angle images [136, 137, 139]. A number of methods have been proposed in the literature to establish a mapping between the distorted wide angle images and their ideal purely perspective counterparts [144, 145, 151, 153]. When such a mapping is known and invertible, the most straightforward way to deal with wide angle images consists in explicitly compensating for radial distortion through a rectification process [137]. In the following Sections we review the main theoretical and practical fisheye camera models which will be used for our experimental analysis.

A.2.1 Theoretical Projection Functions

Cameras are designed to adhere to a specific projection function which allows to map points $\mathbf{P}$ on the 3D scene to their 2D counterparts $\mathbf{p}$ on the sensor. To describe the possible projection functions in a unified framework, we will consider a scheme similar to sphere camera model discussed in Appendix A.1.1. According to this scheme, 3D points of the scene are first projected to the unit sphere centered at the

optical center of the lens, then to the image plane situated at at a distance equal to the focal length f of the lens. Points of the scene $\mathbf{P} \equiv (X, Y, Z)$ can be projected to the unit sphere and so expressed in spherical coordinates $\mathbf{P} \equiv (1, \theta, \varphi)$ through the following Equations[5]:

$$\theta = \arccos\left(\frac{Z}{\sqrt{X^2 + Y^2 + Z^2}}\right) \tag{A.1}$$

$$\varphi = \arctan\left(\frac{Y}{X}\right). \tag{A.2}$$

Point $\mathbf{P}$ is hence mapped to the 2D point on the image plane $\mathbf{p} \equiv (x, y)$ expressed in polar coordinates as follows:

$$\mathbf{p} \equiv (\rho = \pi(\theta), \varphi) \tag{A.3}$$

where π is referred to as the projection function. According to this model, θ is the angle formed between the incoming light ray and the principal axis of the lens. Figure A.7 illustrates the considered camera model.

For regular cameras (i.e., the so called perspective cameras), the projection function π in Equation (A.3) has the form of a perspective projection [141]:

$$\rho = f_P \tan\theta \tag{A.4}$$

where ρ is the radial coordinate of the 2D projected point and f_P is the focal length of the perspective lens. Fisheye lenses, instead, are designed to approximatively obey one of the following projection functions [141, 151]:

$$\hat{\rho} = 2f_F \tan(\theta/2) \quad \text{(stereographic projection)} \tag{A.5}$$

$$\hat{\rho} = f_F \theta \quad \text{(equidistance projection)} \tag{A.6}$$

$$\hat{\rho} = 2f_F \sin(\theta/2) \quad \text{(equisolid angle projection)} \tag{A.7}$$

$$\hat{\rho} = f_F \sin\theta \quad \text{(orthogonal projection)} \tag{A.8}$$

[5]It should be noted that the arctan function denoted in Equation (A.2) should be defined taking into account the correct (x, y) quadrant in which the point lies. Such function is best implemented using the standard function "atan2" available in the standard libraries of the main programming languages.

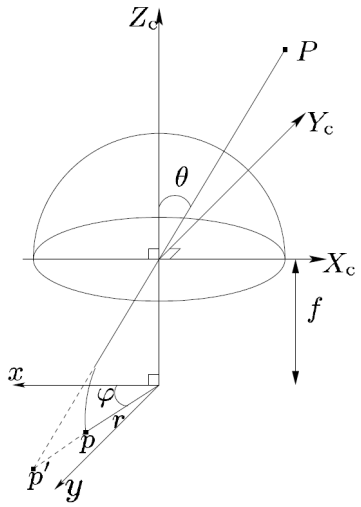


Figure A.7: Camera model considered to describe the image formation process of fisheye cameras. The point from the scene P is first mapped to the unit sphere, then projected to the image plane. In the illustration, p is the projection of the scene point P according to some fisheye projection (one among Equations (A.5) - (A.8)), while it would have been p' according to a perspective projection (Equation (A.4)).

where $\hat{\rho}$ is the radial coordinate of the distorted 2D point projected on the sensor and f_F is the focal length of the fisheye lens.

Given a fisheye camera, a mapping between the rectilinear (i.e., perspective) space and the distorted (i.e., fisheye) one, can be established observing how incoming light rays forming an angle with the principal axis equal to θ are projected by the different projection functions reported in Equations (A.4) - (A.8). The mapping function hence can be derived by considering one of the fisheye projection functions reported in Equations (A.5) - (A.8) (the one which best describes the considered fisheye camera) and the perspective projection reported in Equation (A.4), solving

in terms of θ and equating [137]. Considering all Equations (A.5) - (A.8), we obtain:

$$\hat{\rho} = 2f_F \tan\left(\frac{\arctan\left(\frac{\rho}{f_P}\right)}{2}\right) \quad \text{(stereographic projection)} \tag{A.9}$$

$$\hat{\rho} = f_F \arctan\left(\frac{\rho}{f_P}\right) \quad \text{(equidistance projection)} \tag{A.10}$$

$$\hat{\rho} = 2f_F \sin\left(\frac{\arctan(\frac{\rho}{f_P})}{2}\right) \quad \text{(equisolid angle projection)} \tag{A.11}$$

$$\hat{\rho} = f_F \sin\left(\arctan\left(\frac{\rho}{f_P}\right)\right) \quad \text{(orthogonal projection)} \tag{A.12}$$

which allow to derive the distortion function Ψ mapping a point belonging to the undistorted space $\mathbf{u} \equiv (\rho, \varphi)$ to its counterpart in the distorted space $\mathbf{x} \equiv (\hat{\rho}, \varphi)$ which will be denoted as follows:

$$\mathbf{x} = \Psi(\mathbf{u}) \tag{A.13}$$

The distortion function characterizes the lens and can be used to simulate the image formation process of a fisheye camera, given an image acquired with a standard camera. The effect is the artificial introduction of radial distortion. With similar considerations, we can derive the following inverse mappings [137]:

$$\rho = f_P \tan\left(2\arctan\left(\frac{\hat{\rho}}{2f}\right)\right) \quad \text{(stereographic projection)} \tag{A.14}$$

$$\rho = f_P \arctan\left(\frac{\hat{\rho}}{f_F}\right) \quad \text{(equidistance projection)} \tag{A.15}$$

$$\rho = f_P \tan\left(2\arcsin\left(\frac{\hat{\rho}}{2f_F}\right)\right) \quad \text{(equisolid angle projection)} \tag{A.16}$$

$$\rho = f_P \tan\left(\arcsin\left(\frac{\hat{\rho}}{f_F}\right)\right) \quad \text{(orthogonal projection)} \tag{A.17}$$

and the related distortion function will be denoted as follows:

$$\mathbf{u} = \Psi^{-1}(\mathbf{x}) \tag{A.18}$$

The inverse distortion function Ψ^{-1} can be used to remove the radial distortion from an image acquired using a fisheye camera in order to make it look like a standard image acquired using a perspective camera. This process is usually referred to as

"rectification" because it has the effect to correct the appearance of curved lines which should correspond to straight contours in the scene.

It should be noted that, since the design of real fisheye lenses can be quite complex [154], a deviation from the ideal models reported in Equations (A.5) - (A.8) is usually expected. Therefore, more generic camera models and calibration techniques, as the one proposed in [145] and reviewed in Appendix A.2.2, are usually preferable when modeling fisheye cameras.

A.2.2 Division Model

The Division Model [145] establishes a relationship between the image point $\mathbf{x}$ in the distorted space and its undistorted counterpart $\mathbf{u}$ in the rectilinear one as follows:

$$\mathbf{u} = \Psi^{-1}(\mathbf{x}) = \frac{\mathbf{x}}{1 + \xi||\mathbf{x}||^2} \tag{A.19}$$

where the distortion parameter $\xi < 0$ regulates the amount of radial distortion in the image, and point coordinates are referred to the principal point. It should be noted that the Division Model is characterized by a single parameter ξ, which makes its employment and calibration very convenient. The relationship reported in Equation (A.19) can be inverted in order to derive the distortion function Ψ which maps an undistorted point $\mathbf{u}$ in the rectilinear space to the distorted point $\mathbf{x}$ in the image:

$$\mathbf{x} = \Psi(\mathbf{u}) = \frac{2\mathbf{u}}{1 + \sqrt{1 - 4 \cdot \xi||\mathbf{u}||^2}}. \tag{A.20}$$

According to Equations (A.19) and (A.20), a point of radial coordinate r in the undistorted space is related to a point of radial coordinate $\hat{r}$ in the distorted image by the following expressions:

$$r = g^{-1}(\hat{r}) = \frac{\hat{r}}{1 + \xi\hat{r}^2} \tag{A.21}$$

$$\hat{r} = g(r) = \frac{2r}{1 + \sqrt{1 - 4 \cdot \xi r^2}} \tag{A.22}$$

where function g maps undistorted rays r to distorted rays $\hat{r}$. As shown in [146], despite its simplicity, the Division Model can effectively model real fisheye lenses.

Extending the Division Model

Unfortunately, the interpretation of the values assumed by ξ is not intuitive and the effects of setting a specific value for ξ clearly depend on the size of the input image. This makes the characterization of a given amount of distortion difficult, since setting the same value of ξ for two images of different sizes will result in perceptually different amounts of distortion. To overcome these limitations, we propose to characterize the amount of distortion present in an image with the distortion rate d, which we define as follows:

$$d = 1 - \frac{\hat{r}_M}{r_M} \tag{A.23}$$

where r_M represents the distance from the center of the distortion to the corner of the distorted output image and $\hat{r}_M$ represents its distorted counterpart. It should be noted that such a definition is perceptually coherent and independent from the image size. Considering that between r_M and $\hat{r}_M$ holds the relationship in Equation (A.22), the parameter ξ can be straightforwardly computed from a given distortion rate d using the following formula:

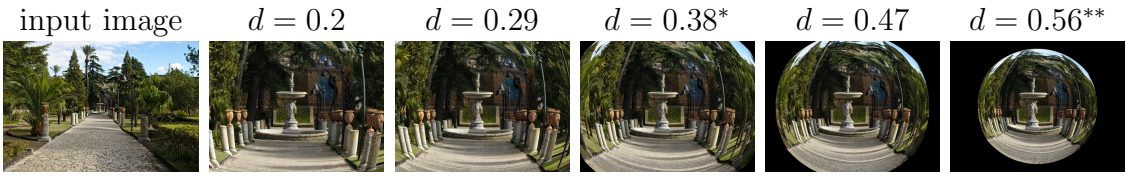
$$\xi = -\frac{d}{[r_M(1-d)]^2}. \tag{A.24}$$

Even if no direct relationship between the Field Of View of a given image and parameter ξ is provided by the Division Model, the exact values of ξ can be derived for the full frame and full circle configurations discussed above. In both cases we want the distortion function in Equation (A.20) to project points at infinity to points on the image having a specific radius $\overline{r}$. In the case of full frame images, we set $\overline{r} = r_M$ to obtain a diagonal FOV equal to 180°. In the case of full circle images, we set $\overline{r} = h/2$ where h is equal to the image height in order to obtain a vertical FOV equal to 180°. Let us consider the limit of expression in Equation (A.22) as r approaches $+\infty$:

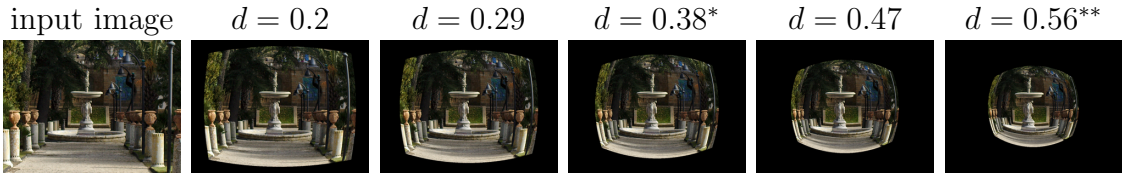
$$\lim_{r \to +\infty} \frac{2r}{1 + \sqrt{1 - 4 \cdot \xi r^2}} = \frac{2}{\sqrt{-4\xi}}. \tag{A.25}$$

Equating such expression to $\overline{r}$, we get:

$$\xi = -\frac{1}{\overline{r}^2}. \tag{A.26}$$



(a) simulation of fisheye images with variable FOV



(b) simulation of fisheye images with static FOV

Figure A.8: Some examples of synthetic fisheye images obtained adding different amounts of radial distortion to input rectilinear images by using the Division Model. (a) The input image is a high resolution image (5204×3472 pixels). (b) The input image is a low resolution image (1024×768 pixels). All the output distorted images have resolution equal to 1024×768 pixels. The * and ** symbols denote the full frame and full circle distortion rates respectively.

Equation (A.26) can be used to compute the distortion parameter ξ allowing the projection of a point at infinity in the rectilinear space to a point with radial coordinate $\overline{r}$ in the fisheye space. Combining Equations (A.23) and (A.26) and considering the values which $\overline{r}$ assumes in the case of full frame and full circle images, it is possible to obtain the following expressions:

$$d_{full-frame} \approx 0.38 \tag{A.27}$$

$$d_{full-circle} = \frac{2\alpha^2 - \sqrt{4\alpha^2 + 5} + 3}{2\alpha^2 + 2} \tag{A.28}$$

where $\alpha = \frac{w}{h}$ is the image aspect ratio and w is the image width. For a square image (i.e., $\alpha = 1$), the distortion rate inherent to a full circle image would be exactly 0.5. For a standard aspect ratio of $\alpha = \frac{4}{3}$, the distortion rate inherent to full circle configurations is $d_{full-circle} \approx 0.56$. Figure A.8(a) shows some synthetic images obtained adding different amounts of radial distortion to a source rectilinear image.

A.2.3 Image Rectification and Camera Simulation

Radial distortion affects the way objects appear in wide-angle images and therefore it can deceive the feature extraction process usually employed by computer vision algorithms [148]. In order to avoid the influence of radial distortion, a rectification process is usually applied to wide-angle images as a preprocessing step. If the inverse distortion function Ψ^{-1} is known, rectification can be performed operating a coordinate remapping and interpolating where needed. Let $\hat{I}$ be the source distorted image, its rectified counterpart I will be denoted by:

$$I(\Psi^{-1}(\mathbf{x})) = \hat{I}(\mathbf{x}). \tag{A.29}$$

Similarly, it is possible to artificially simulate the radial distortion of a lens for which the distortion function Ψ is known. In this case, the distorted image I will be denoted by:

$$\hat{I}(\Psi(\mathbf{u})) = I(\mathbf{u}). \tag{A.30}$$

Simulating radial distortion using Equation (A.30) can be convenient for experimental purposes as discussed in [136]. In these settings indeed, it is possible to model radial distortion independently from the more complex image formation process. This allows to control the exact amount of radial distortion present in the image and establish a precise mapping with the undistorted space, which can be used to create a reference ground truth. When the FOV is large, the distortion function Ψ is generally designed to project an infinite rectilinear image I to a finite fisheye image $\hat{I}$. In practice the resolution of the input image I should be sufficiently larger than the one of the output image $\hat{I}$ to achieve consistent results. Mapping high resolution input images to low resolution ones allows to cover a larger part of the artificially distorted image, which is a preferable result. Figure A.8 shows some examples of synthetic fisheye images obtained using input rectilinear images of variable sizes. Note that the full frame image shown in Figure A.8(a) still exhibits black corners. This is due to the fact that the input image is finite while an infinite image would be required in principle. For the same reason the full circle image shown in Figure A.8(a) is not perfectly circular and slightly smaller than what a real full circle image should look like. Despite such considerations, the synthetic images are worth to be considered since they are characterized by the amounts of radial distortion

inherent to the full frame and full circle configurations and still cover most of the related Field Of View. Finally, it should be noted that distorting rectilinear images using Equation (A.30) or rectifying wide-angle images using Equation (A.29) is not totally accurate since the the image formation process cannot be modeled in its totality due to the lack of depth information of the acquired scene.

A.3 Experimental Datasets

As denoted at the beginning of the chapter, we are investigating how feature extraction can be performed directly on wide angle images. In this Section, we review the three datasets used in the experimental analysis. Two of the three considered datasets comprise rectilinear images to which radial distortion has been artificially added following the methodologies discussed in Appendix A.2.3. Working in these settings is convenient since it allows to control the exact amount of distortion present in the images used for the experiments. The third dataset comprises real images acquired using three different fisheye cameras.

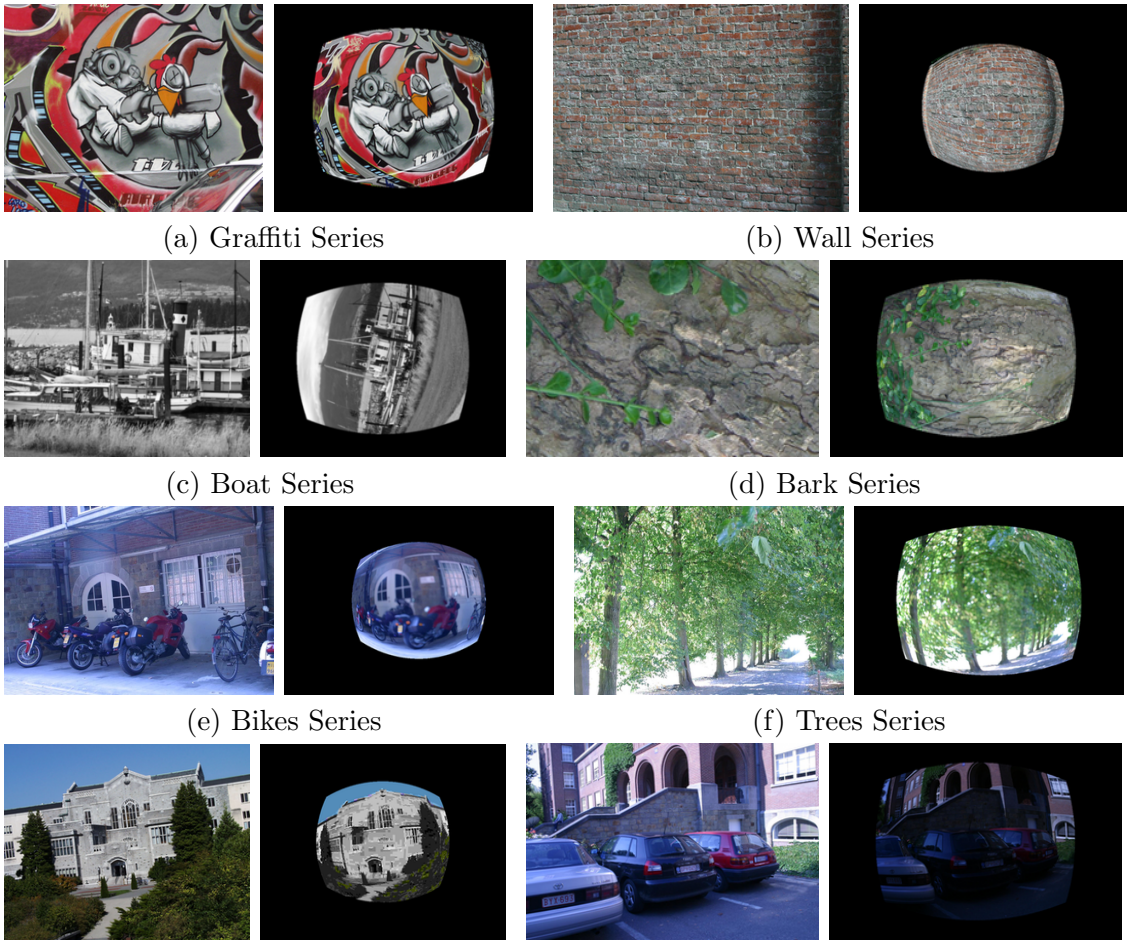
A.3.1 OXFORD-48

To analyze the performances of feature extraction methods on wide-angle images, we consider the popular dataset proposed in [155]. It provides 8 image series, each characterized by different variabilities: change of viewpoint angle, scale changes, image blur, JPEG compression, light changes. The dataset comprises both structured and textured scenes. Each series consists of a reference image, containing the least amount of the specified variability (i.e., the zero-variability) and 5 test images characterized by increasing amounts of the specified variability. The dataset contains 48 images in total. To assess the influence of the combination of radial distortion with the aforementioned variabilities, we artificially add radial distortion to each test image in the dataset. It should be noted that no distortion is added to the reference images. Depending on the experiment, we consider different camera models to artificially add radial distortion to the images. In particular, we consider the division model to generate series characterized by the amounts of distortion inherent to the full frame and full circle configurations, as well as camera models based on the theoretical projection functions discussed in Appendix A.2.1. For the division

model, the exact distortion rates are computed using Equations (A.27) and (A.28) in order to account for the different aspect ratios characterizing the input images. Since the resolution of the images in the OXFORD-48 dataset is not high (640×480 pixels), we have to keep the resolution of the output distorted image equal to the one of the input image. Unfortunately, images generated in this way present black borders for the reasons discussed in Appendix A.2.3 which have to be considered to properly conduct the experiments. We refer to this dataset as OXFORD-48. Figure A.9 shows some samples from the considered series.

A.3.2 DASF-HIRES-100 and DASF-HIRES-50

In order to perform experiments with respect to varying rates of distortion, we collected a dataset of 100 high resolution images belonging to the following scene categories: indoor, outdoor, natural, handmade, urban, car, pedestrian, street. The considered categories are relevant to the main application domains where the image gradients are usually employed [7, 138], and consistent with the scene categorization proposed by Torralba & Oliva [77]. Each image is provided with one or more tags related to the above specified scene categories. All images have been acquired using a Canon 650D camera mounting a Canon EF-24mm lens and have resolution equal to 5204×3472 pixels. We have first introduced this dataset in [26] and extended it with scene-based tags in [22]. The dataset will be referred to as DASF-HIRES-100. Figure A.10 shows some examples of the input images used for the evaluations, whereas Table A.1 reports some statistics about the scene-related tags present in the dataset. High resolution images are mapped to low resolution distorted counterparts using the Division Model in order to simulate increasing degrees of radial distortion according to the modalities discussed in Appendix A.2.3. This way, it is possible to create image series similar to the ones of the OXFORD-48 dataset, containing a reference image (the high resolution undistorted image) and a number of test images characterized by increasing degrees of radial distortion. Please note that, in these settings, the mapping between the reference image and each test image is known by design. In all experiments, the test distorted images have resolution 1024×768 pixels, while the exact distortion rates (and related ξ parameters) used to produce the distorted images depend on the experiment (and will be discussed in the appropriate sections). Figure A.11 shows some sample series of 6 images consisting of



(a) Graffiti Series (b) Wall Series

(c) Boat Series (d) Bark Series

(e) Bikes Series (f) Trees Series

(g) UBC Series (h) Leuven Series

Figure A.9: Some examples from OXFORD-48 dataset. The leftmost image in each pair is always the reference image, while the rightmost image is one of the test images in the series characterized by a given amount of distortion. The distortion is coupled with the variabilities considered in [155]: (a) Change of viewpoint angle for a structured scene (full frame distortion). (b) Change of viewpoint angle for a textured scene (full circle distortion). (c) Scale changes for a structured scene (full frame distortion). (d) Scale changes for a textured scene (full frame distortion). (e) Image blur for a structured scene (full circle distortion). (f) Image blur for a textured scene (full frame distortion). (g) JPEG compression (full circle distortion). (e) Light change (full frame distortion).

the reference full resolution rectilinear image, plus 5 test images affected by different rates of radial distortion. In our experiments [21], we have considered also a smaller dataset obtained by randomly sampling 50 images from DASF-HIRES-100. We will refer to this sub-dataset as DASF-HIRES-50. Both datasets are publicly available to

Figure A.10: Some randomly chosen images from the dataset, for each considered scene-based tag.

Scene	**Indoor**	**Outdoor**	**Natural**	**Handmade**	**Urban**	**Car**	**Pedestrian**	**Street**
Count	13	87	44	93	51	49	19	50

Table A.1: Numbers of images containing a specific scene-tag.

the research community and can be downloaded from the following URLs: http://iplab.dmi.unict.it/DASF/ and http://iplab.dmi.unict.it/FisheyeAffine/.

A.3.3 RDSIFT-39

In order to perform tests with real fisheye images, we consider the benchmark dataset introduced in [136]. It comprises three image series acquired using fisheye cameras characterized by different amounts of radial distortion. Calibration images and division model parameters for each camera are included in the dataset. Figure A.12 shows some sample images from the considered dataset. For each image series, we report the distortion rates computed according to our model: 0.13, 0.19 and 0.54.

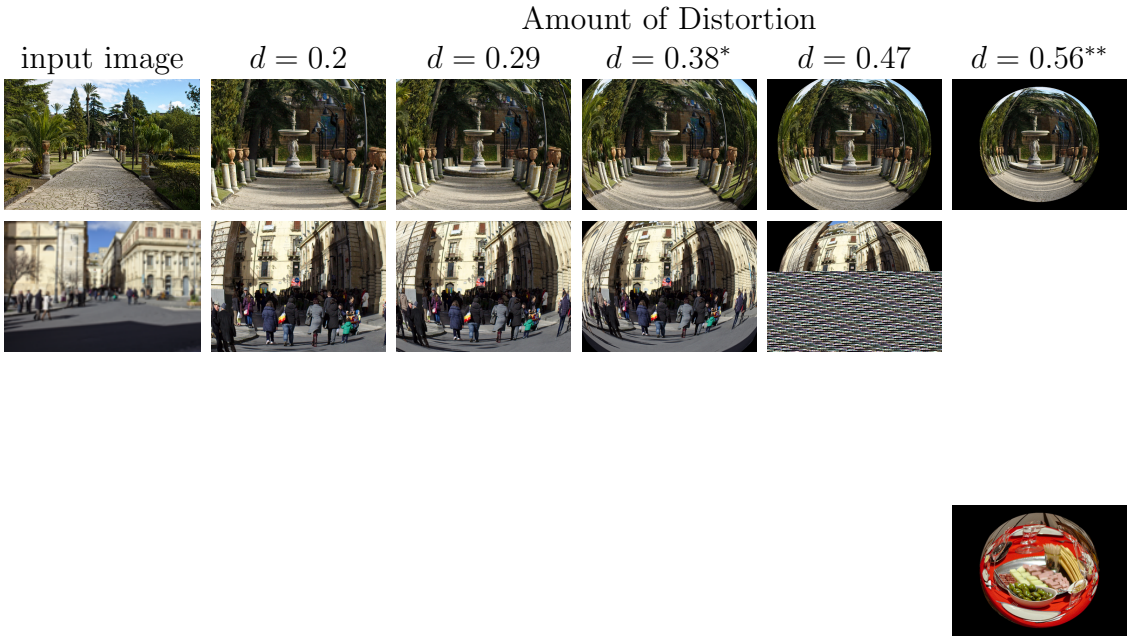


Figure A.11: Four image series from DASF-HIRES-100. The * and ** symbols denote the full frame and full circle distortion rates respectively.

(a) Series 1 (S1) - $d = 0.13$

(a) Series 2 (S2) - $d = 0.19$

(a) Series 3 (S3) - $d = 0.54$

Figure A.12: Some sample images from the three image series in RDSIFT-39.

Each series consists of 13 images related by different transformations including viewpoint change, rotation and scale. The dataset contains 39 images in total. Images within a series represent a scene containing the same planar object acquired from different positions. All image pairs within a series are provided with an homography relating their undistorted counterparts. Differently from the OXFORD-48 dataset, the amount of variability present in each image (e.g., viewpoint angle, or scaling factor) is not quantified with respect to a given reference image. Hence, instead of considering only reference-test image pairs, all possible 78 image pairs within a series are considered in the experiments. We refer to this dataset as RDSIFT-39.

A.4 Affine Covariant Region Detectors on Fish-eye Images

As discussed earlier, the most straightforward approach to deal with wide-angle images consists in explicitly removing radial distortion through a rectification process. Such process however has some major limitations:

1. it can be computationally expensive (especially in mobile and embedded settings) due to the need of interpolation to account for the spatially non-uniform sampling performed by wide angle cameras;

2. interpolation introduces artifacts in the image which can affect the feature extraction process;

3. it requires the camera to be calibrated (and hence known in advance) in order to establish a mapping between the distorted points and their positions in the rectilinear image plane.

As many authors claim [136, 149, 150, 156], it would be advantageous to be able to perform feature extraction directly on wide-angle images, without performing any rectification. Some works assume that the camera is known in advance and hence it can be calibrated. The authors of [156, 157] studied how to compute the scale space of omnidirectional images, in [136, 158, 159] the Scale Invariant Feature Transform (SIFT) pipeline [7] is modified in order to be used directly on wide angle images. In [147] scale invariant features are derived from wide angle images mapping them

to a sphere. A direct approach to detect people using omnidirectional cameras is proposed in [149, 150]. In [160] an algorithm to extract straight edges from distorted images is presented, whereas in [136, 26, 160] methods to estimate geometrically correct gradients of distorted images are investigated.

A second category of algorithms works directly on the distorted images. In this case, the camera does not need to be calibrated and radial distortion is treated as an additional variability affecting the images. For instance, in [161] the Perspectively Invariant Normal features (PIN) are computed using a depth map in order to be independent from the acquisition point of view and from the employed camera. PIN can be successfully used to match regions between rectilinear and wide angle images as pointed out by the authors of [161]. In [162] an approach to match features between uncalibrated omnidirectional images (not rectified) and perspective images is presented. People detection and tracking are performed directly on fisheye images using a probabilistic appearance model in [163]. The authors of [164] perform feature matching on omnidirectional images through descriptor learning.

In this Section, we concentrate on this second category of approaches which do not require calibration in order to perform feature extraction. Such approaches can be particularly advantageous in applications in which the input images are acquired by different cameras which are not generally known in advance and hence difficult to calibrate. Examples of such applications include image retrieval (e.g., images on the web), object detection on generic cameras, and registration (e.g., camera networks in surveillance applications). In particular, we study how the detection, description and retrieval of local features can be reliably performed on uncalibrated wide angle images acquired by an unknown device. Considering the amount of work already done by the research community in the field of affine covariant detectors and descriptors in the undistorted domain [155, 165, 166], we investigate whether the state-of-the-art affine detectors are suitable to be used directly on wide angle images. We support our analysis by theoretically showing that, even if the radial distortion introduced by fisheye cameras is not an affine transformation, it can be locally approximated as a linear function with a small error. We consider three state-of-the-art affine region detectors [155], namely the Maximally Stable Extremal Regions (MSER) [167], the Harris affine region detector and the Hessian affine region

detector [168, 169]. We assess experimentally how the aforementioned detectors behave under the influence of increasing radial distortion and the variabilities included in the OXFORD-48 dataset [155], i.e., change of viewpoint angle, scale changes, blur, JPEG compression and light changes. The analysis presented in this Section has been first published in [27], then extended in [21].

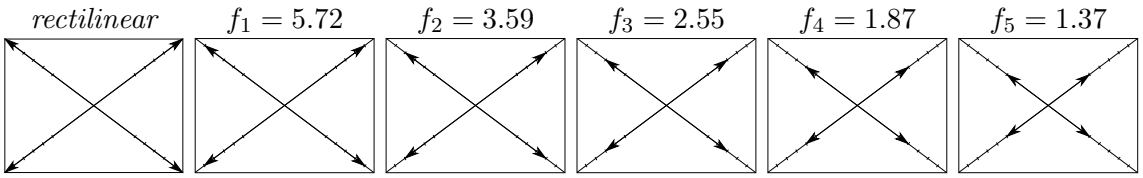
A.4.1 Theoretical Camera Models

We consider a class of camera models directly derived from the theoretical projection functions of fisheye cameras discussed in Appendix A.2.1. For our baseline camera model, we consider a $1/2.5''$ sensor ($5.76\ mm \times 4.29\ mm$)[6] and a fisheye lens following the one of the projection functions reported in Equations (A.5) - (A.8), with its principal point corresponding to the center of the sensor. The fisheye distortion is simulated by mapping the pixel coordinates of a rectilinear image to a fisheye image of the same resolution as described in Appendix A.2.3. We let $f = f_F = f_P$, to avoid the scale change effects due to projecting images with different focal lengths. The conversion between the millimeters world coordinates and the pixel image coordinates is performed with respect to the chosen sensor and the resolutions of images considered for the experiments. To analyze the behavior of the models with respect to different amount of distortion, different focal lengths are considered. We set the smallest focal length to the one giving a full-circle image (i.e., a vertical FOV equals to 180°) when the equidistance projection function is considered (Equation (A.6)). The exact focal length is computed considering the following formula:

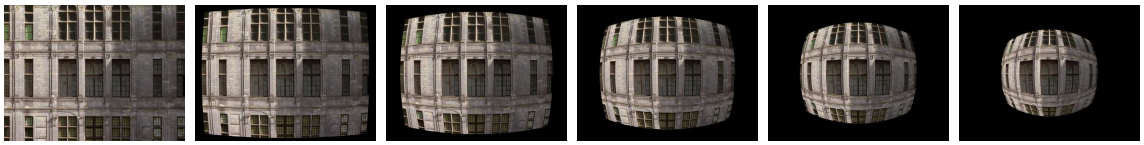
$$FOV = \frac{l}{f} \tag{A.31}$$

where FOV is the Field Of View of the camera in radiants, f is the focal length, and l is the length of the dimension of the sensor with respect to which the FOV is computed (i.e., vertical, horizontal or diagonal). To impose a full-circle geometry, we consider $FOV = \pi$ and $l = 4.29\ mm$ (which is the height in millimeters of a $1/2.5''$ sensor). The focal length f can be obtained using the inverse relationship

[6] We choose the dimension of a $1/2.5''$ sensor, because it is widely used in the manufacturing of commercial 180° fisheye cameras.



(a) relationship between focal lengths and projected sensor diagonal



(b) synthetic fisheye images

Figure A.13: (a) Relationship between focal lengths and projected sensor diagonal. (b) Synthetic images obtained artificially adding radial distortion to a rectilinear image considering the described camera model and the equidistance projection.

derived from Equation (A.31):

$$f = \frac{l}{FOV} = \frac{4.29\ mm}{\pi} \approx 1.37\ mm. \tag{A.32}$$

The remaining four focal lengths are set in order to obtain perceptively uniform degrees of distortion by imposing that the projection of the diagonal vary at a uniform step. Figure A.13(a) shows the relationship between focal lengths and projected sensor diagonals. Figure A.13(a) reports some synthetic images obtained artificially adding radial distortion to a rectilinear image considering the described camera model and an equidistance projection function. The focal lengths considered for the analysis are: $f_1 = 5.72\ mm$, $f_2 = 3.59\ mm$, $f_3 = 2.55\ mm$, $f_4 = 1.87\ mm$, $f_5 = 1.37\ mm$.

A.4.2 Local Linearity of Fisheye Distortion Functions

In order to provide theoretical evidence to support the applicability of affine covariant region detectors on fisheye images, in this Section we show that, even if the radial distortion introduced by fisheye cameras is not an affine transformation, it can be modeled as a linear function in small local neighborhoods. This is done both for the ideal projection functions discussed in Appendix A.2.1 and for the more practical Division Model discussed in Appendix A.2.2.

Theoretical Distortion Functions

While the distortion mappings reported in Equations (A.10) - (A.12) are not affine transformations, it can be assessed that locally (i.e., in regions with small radii) they can be approximated as linear functions. Without loss of generality, we consider the equidistance projection reported in Equation (A.6), and the related distortion mapping reported in Equation (A.20). We then consider the first order Taylor polynomial approximation of Equation (A.20), where we set $f = f_F = f_P$ assuming two equivalent perspective and fisheye cameras:

$$\hat{\rho}(\rho, \rho_0) \approx \frac{\rho - \rho_0}{1 + (\frac{\rho_0}{f})^2} + f \cdot \arctan \frac{\rho_0}{f}. \tag{A.33}$$

The approximation reported in Equation (A.33) gives small errors in sufficiently small neighborhoods of ρ_0. The mapping reported in Equation (A.20) can hence be approximated by a number of local linear approximations which takes the form of Equation (A.33). In particular, we select a number of ρ_0^i points at a step of 2ε ($\varepsilon > 0$) and define the approximation as follows:

$$\hat{\rho} \approx \begin{cases} \hat{\rho}(\rho, \rho_0^1) & \text{if} \quad \rho \in (\rho_0^1 - \varepsilon, \rho_0^1 + \varepsilon) \\ \hat{\rho}(\rho, \rho_0^2) & \text{if} \quad \rho \in (\rho_0^2 - \varepsilon, \rho_0^2 + \varepsilon) \\ \dots & \\ \hat{\rho}(\rho, \rho_0^n) & \text{if} \quad \rho \in (\rho_0^n - \varepsilon, \rho_0^n + \varepsilon) \end{cases} \tag{A.34}$$

Figure A.14 shows the mean reprojection error for the radial coordinates when the mappings in Equations (A.9) - (A.12) are approximated using Equation (A.34) for different step values ε and for the two extremal focal lengths ($f_1 = 1.37$, $f_5 = 5.72$). It should be noted that, for regions with radii below 26 pixels, the reprojection error is under 0.1 pixels and for regions with radii up to 70 pixels, the reprojection error is below 0.67 pixel. Please note that both errors are negligible for most applications, where only sub-pixel precision is required [145].

Division Model Distortion Function

The considerations made in the previous section are based on the theoretical projection functions discussed in Appendix A.4.2. For sake of generality, we extend our

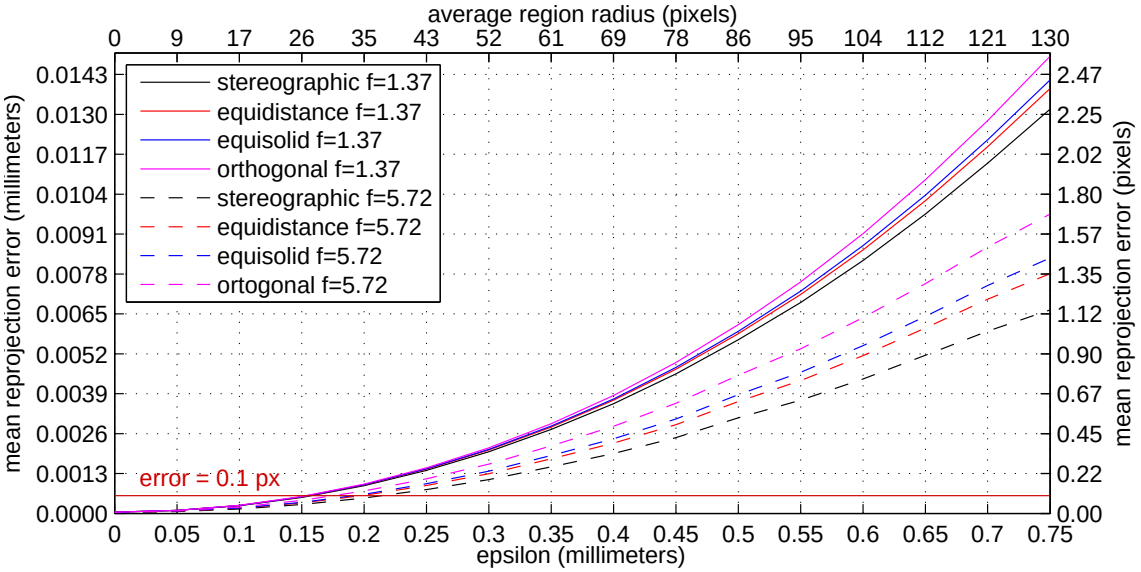


Figure A.14: Reprojection errors for the linear approximations of mappings reported in Equations (A.9) - (A.12). The results are reported in millimeters (bottom and left axes) and pixels (top and right axes). The pixel values are obtained considering a medium resolution of 1000×700 pixels.

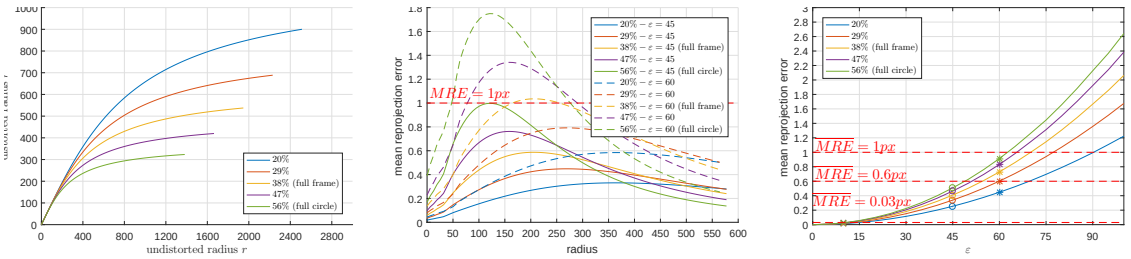
analysis to the Division Model, which has proved to be able to model real fisheye cameras [146]. Specifically, in this section we show that the radial distortion function of the division model (Equation (A.22)) can be linearly approximated locally and that if the neighborhood is sufficiently small, the approximation error is negligible.

Let us consider the first order Taylor polynomial approximation of the mapping function reported in Equation (A.22), centered at an arbitrary point $\hat{r}_0$ and restricted to the local neighborhood of radius ε centered at $\hat{r}_0$ denoted by $\mathcal{N}(r_0, \varepsilon) = (\hat{r}_0 - \varepsilon, \hat{r}_0 + \varepsilon)$:

$$g(\hat{r})_{|\mathcal{N}(r_0,\varepsilon)} \approx \tilde{g}(\hat{r}, \hat{r}_0) = g(r_0) + (r - r_0)g'(r0). \tag{A.35}$$

We expect the error given by such an approximation to be proportional to the extent of the chosen radius ε. To measure such error, we define the Mean Reprojection Error of expression (A.35) in a given point r_0 and for a chosen radius ε as follows:

$$MRE(r_0, \varepsilon) = \frac{\int_{r \in \mathcal{N}(r_0,\varepsilon)} |g(r) - \tilde{g}(r, r_0)| dr}{\int_{r \in \mathcal{N}(r_0,\varepsilon)} dr}. \tag{A.36}$$



(a) projections (b) MRE curves (c) $\overline{MRE}$ curves

Figure A.15: (a) The plot of the division model projection function (Equation (A.22)) for different distortion rates. (b) The Mean Reprojection Error curves for fixed values of ε. (c) The average Mean Reprojection Error for varying neighborhood radii ε.

Moreover, for a fixed value of ε, we define the average MRE value as follows:

$$\overline{MRE}(\varepsilon) = \frac{\int_{r=0}^{r_{max}} MRE(r,\varepsilon)dr}{\int_{r=0}^{r_{max}} dr}. \tag{A.37}$$

In Equation (A.37), r_{max} is introduced in order to avoid to carry the integration up to infinity, where the curves related to Equation (A.22) tend to become rectilinear as shown in Figure A.15(a) and the MRE value would be close to zero. In particular, we set r_{max} to the half diagonal of the distorted images of resolution 1024×768 pixels which will be considered in the experiments, i.e., $r_{max} = \frac{1}{2}\sqrt{(1024^2 + 768^2)} = 640$ pixels. Figure A.15(b) shows the MRE curves for two selected values of ε (i.e., 45 and 60 pixels) and different amounts of distortion, while Figure A.15(c) shows the average MRE for varying values of ε and different amounts of distortion. In particular Figure A.15(c) shows that the fisheye distortion of local regions having radii smaller than $\varepsilon = 60$ pixels can be approximated as a linear function with average subpixel precision (see points marked with the symbol "*" in Figure A.15(c)). The average error drops to about 0.6 pixels for radii smaller than 45 pixels (see points marked with the symbol "o" in Figure A.15(c)) and to about 0.03 pixels for radii smaller than 10 pixels (see points marked with the symbol "+" in Figure A.15(c)). Figure A.15(b) shows how the MRE values vary in the different parts of the image. Specifically, the error is small in the central and peripheral areas of the image and higher in between. It is worth noting that for regions with radii smaller than 45 pixels, the MRE is always under 1 pixel for all distortion rates.

Our analysis points out that, up to a given extent, circular regions can be mapped

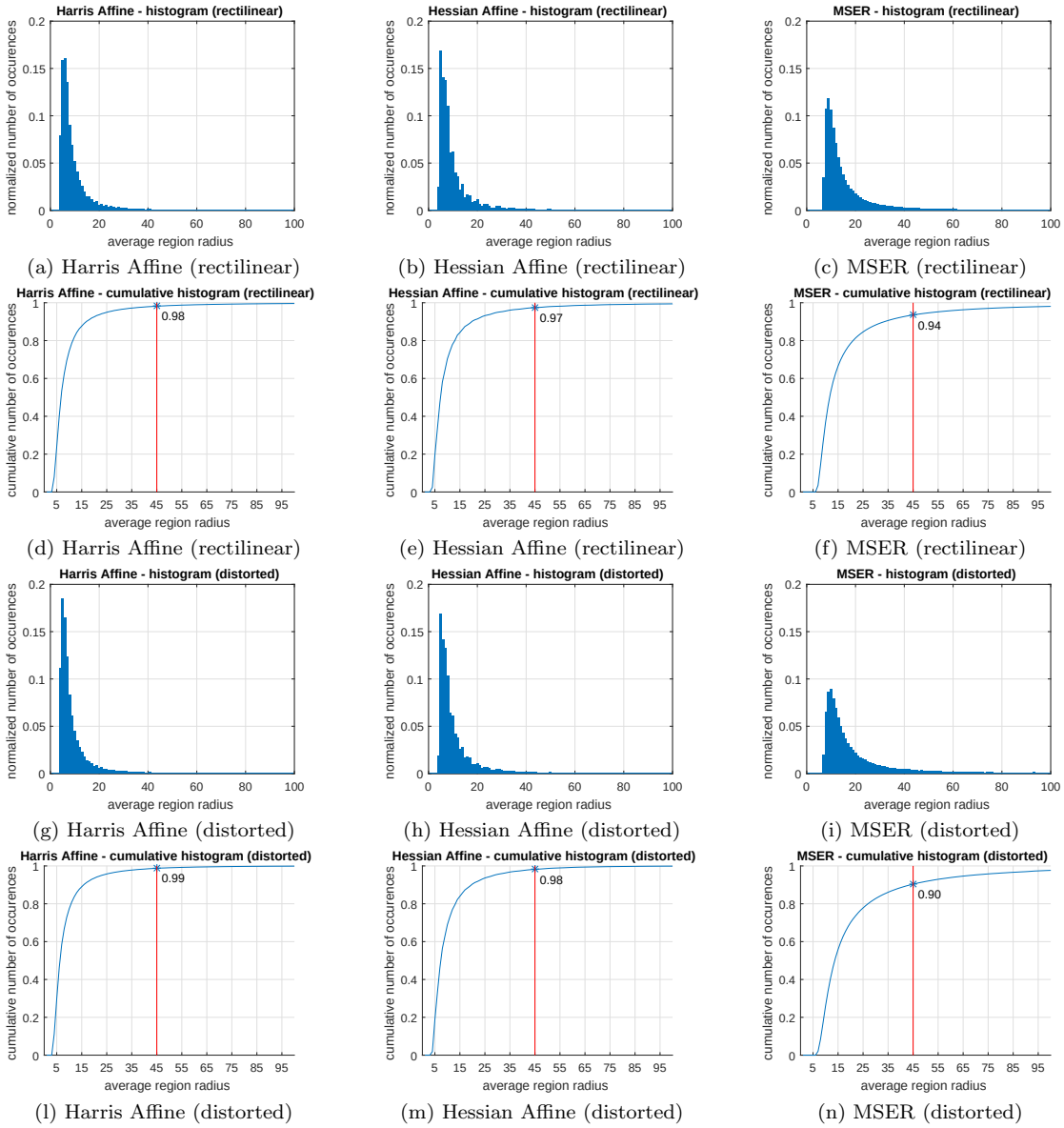
from a reference non-distorted space to its distorted counterpart in the fisheye image using an appropriate linear function with a small projection error. If the error is low enough, an affine covariant region detector should be able to correctly extract both the reference and distorted regions modeling the latter as an affine transformation of the former. Moreover, in the description stage, the distorted region will be mapped to its undistorted counterpart with a small error using the inverse of the affine transformation estimated by the region detector. Hence we expect small linear approximation errors to be beneficial for both the feature detection and description steps.

Analysis of Region Size for DASF-HIHRES-50 and Discussion

To assess the applicability of affine covariant region detectors on distorted images, we have performed an analysis of the distribution of sizes of regions extracted using the detectors under analysis. In particular, we extracted regions using the considered three detectors on all images present in DASF-HIRES-50. An average radius is computed for each elliptical region as the average between the lengths of semi-major and semi-minor axes. Interestingly, all the considered detectors tend to extract regions characterized by a strong locality. This is summarized in Figure A.16, which shows the normalized histograms of average radii for all rectilinear and distorted images in DASF-HIRES-50. In particular, normalized histograms reported in Figure A.16(a) to (c) and Figure A.16(g) to (i) show how the majority of regions have average radii around 10 pixels. Moreover, the cumulative histograms reported in Figure A.16(d) to (f) and Figure A.16(l) to (n), show how in any case more than 90% of the detected regions have an average radius smaller than 45 pixels. As it has been pointed out in the previous sections, the linear approximation errors for both the theoretical distortion functions and the division model are low for regions with average radii below 45 pixels and negligible for regions with average radii below 10 pixels. These results suggest that affine covariant features are able to model the radial distortion introduced by fisheye images as a local variability.

A.4.3 Experimental Protocol

We evaluate the performances of the three affine regions detectors which best performed in the benchmark by Mikolajczyk et al. [165], namely Harris Affine [169],



(a) Harris Affine (rectilinear) (b) Hessian Affine (rectilinear) (c) MSER (rectilinear)

(d) Harris Affine (rectilinear) (e) Hessian Affine (rectilinear) (f) MSER (rectilinear)

(g) Harris Affine (distorted) (h) Hessian Affine (distorted) (i) MSER (distorted)

(l) Harris Affine (distorted) (m) Hessian Affine (distorted) (n) MSER (distorted)

Figure A.16: (a) to (c) Normalized histograms of average radii of regions extracted by the three detectors on the rectilinear images of dataset DASF-HIRES-50. (d) to (f) Normalized cumulative histograms of average radii of regions extracted by the three detectors on the rectilinear images of dataset DASF-HIRES-50. (g) to (i) Normalized histograms of the average radii of the regions extracted by the three detectors on the distorted images of dataset DASF-HIRES-50. (l) to (n) Normalized cumulative histograms of the average radii of the regions extracted by the three detectors on the distorted images of dataset DASF-HIRES-50.

Hessian Affine [169] and MSER [167]. All the considered region detectors extract affine covariant regions in the form of ellipses as described in [165]. Following the protocol in [155], all experiments are performed on series of 6 images $S = \{I_0, I_1, \ldots, I_5\}$ affected by a specific variability. The first image in the series I_0 is affected by the least amount of the considered variability (the zero-variability) and is referred to as the reference image, while the remaining five images $\{I_i\}_{1 \leq i \leq 5}$ are affected by increasing amounts of the considered variability and are referred to as test images. Given an image series S, we assess the performances of the detectors on each of the 5 image pairs $\{(I_0, I_i)\}_{1 \leq i \leq 5}$ using the reference image to define the ground truth. We assume that for each image pair it is possible to establish a mapping ψ_{i0} between the points of the test image I_i and the ones of the reference image I_0. Specifically, for the images in the dataset DASF-HIRES-50, such mapping is given by the inverse of the distortion function f_{0i} used to generate the test image from the reference one:

$$\psi_{i0}^{A} = f_{0i}^{-1}. \tag{A.38}$$

The OXFORD-48 dataset provides homographies h_{0i} relating the reference image I_0 to the test images I_i. Hence, for the undistorted series contained in OXFORD-48, we define:

$$\psi_{i0}^{B1} = h_{0i}^{-1}. \tag{A.39}$$

In the case of the distorted series of OXFORD-48, instead, the projection from the distorted test image I_i to the undistorted reference image I_0 is carried through the following composition:

$$\psi_{i0}^{B2} = f_i^{-1} \circ h_{0i}^{-1} \tag{A.40}$$

where f_i is the distortion function used to generate the distorted test image I_i. As proposed by Mikolajczyk [155], we measure two important properties of the affine detectors under analysis:

1. the repeatability, i.e., the ability to extract regions which correspond to the same geometrical areas under the considered variabilities;

2. the matching ability, which is the ability to extract distinctive regions that, given a suitable descriptor, can be matched reliably under the considered variabilities.

Repeatability

Let be $\mathcal{D}$ the affine region detector under analysis and let be $\mathcal{F}_i = \mathcal{D}(I_i)$ the set of elliptical features extracted from the generic image I_i using detector $\mathcal{D}$. Since the projection of an ellipse using a distortion function in the form of Equation (A.20) is not an ellipse in general, we sample the elliptical features at an angular step of $\frac{\pi}{30}$ in order to obtain the set of polygonal regions $\mathcal{R}_i$. The repeatability of detector $\mathcal{D}$ is assessed counting how many test regions in $\mathcal{R}_i$ overlap significantly with the reference regions in $\mathcal{R}_0$. In order to measure the overlap, the test regions are first mapped to the reference space using the mapping function ψ_{i0}:

$$\mathcal{R}_{i0} = \{r' = \psi_{i0}(r), \forall r \in \mathcal{R}_i\} \tag{A.41}$$

where r is a polygon and $\psi_{i0}(r)$ is the point-wise projection of r through the mapping function ψ_{i0}. It should be noted that, even if the reference and test images I_0 and I_i are related by the mapping ψ_{i0}, in general they don't cover the same physical areas and hence not all the regions in the sets $\mathcal{R}_0$ and $\mathcal{R}_{i0}$ are guaranteed to lay in the part of the scene present in both images. Given the generic set of regions $\mathcal{R}$, we denote with the notation $\mathcal{R}^{(0,i)}$ the subset of regions of $\mathcal{R}$ entirely contained in the common part of the scene of images I_0 and I_i. For each pair of regions $(r_h, r_k) : r_h \in \mathcal{R}_0^{(0,i)}, r_k \in \mathcal{R}_{i0}^{(0,i)}$, we compute the overlap error as following:

$$err_{hk} = 1 - \frac{area(\alpha \cdot r_h \cap \alpha \cdot r_k)}{area(\alpha \cdot r_h \cup \alpha \cdot r_k)} \tag{A.42}$$

where α is a scaling factor such that $area(\alpha \cdot r_h) = \pi r^2$ and r is a normalized radius. Following the protocol of [155], we set $r = 30$ pixels. Unions, intersections and areas are computed numerically. In order to compute the set of most likely correspondences x_{hk} between regions $r_h \in \mathcal{R}_0^{(0,i)}$ and $r_k \in \mathcal{R}_i^{(0,i)}$, such that the overlap error in Equation (A.42) between r_h and r_k is under a given overlap threshold o_t, we solve the following assignment problem using the Hungarian algorithm [170]:

$$\begin{cases} \min(\sum_{hk} e_{hk} x_{hk}) & \\ \sum_k x_{hk} \leq 1 & \forall h : 1 \leq h \leq |\mathcal{R}_0^{(0,i)}| \\ \sum_h x_{hk} \leq 1 & \forall k : 1 \leq k \leq |\mathcal{R}_{i0}^{(0,i)}| \\ x_{hk} \ \in \{0,1\} & \forall h, \ \forall k : 1 \leq h \leq |\mathcal{R}_0^{(0,i)}| \\ & \wedge 1 \leq k \leq |\mathcal{R}_{i0}^{(0,i)}| \end{cases} \tag{A.43}$$

where:

$$e_{hk} = \begin{cases} err_{hk} & \text{if } err_{hk} \leq o_t \\ +\infty & \text{otherwise} \end{cases}. \tag{A.44}$$

Threshold o_t is set to $o_t = 0.4$ as discussed and justifiedì in [155]. The repeatability score is defined as the number of correspondences normalized by the minimum number of regions detected in the two images (excluding the regions not entirely contained in the common part):

$$repeatability\ score = \frac{\sum_{hk} x_{hk}}{\min(|\mathcal{R}(I_0)^{(i,0)}|, |\mathcal{R}(I_{i0})^{(i,0)}|)}. \tag{A.45}$$

The repeatability score measures the ability of the detector to extract features corresponding to the same geometrical regions under varying amounts of a given variability.

Matching Ability

In order to measure the matching ability of the detectors, we count how many test features in $\mathcal{F}_i$ are correctly matched to the reference features in $\mathcal{F}_0$ given a suitable descriptor. The ground truth matchings are given by the correspondences x_{hk} computed solving the assignment problem in Equation (A.43). Each elliptical feature is normalized to a circular region of dimensions 20×20 pixels and the Local Intensity Order Pattern (LIOP) descriptor is computed over that region [166]. We compute the nearest neighbour matchings between the reference and test descriptors and denote them by m_{hk}, where $m_{hk} = 1$ if f_h matches f_k in the nearest neighbour sense and $m_{hk} = 0$ otherwise. The matching ability is defined as the number of correct nearest neighbour matchings normalized by the minimum number of regions detected in the two reference and test images (excluding the regions not entirely

contained in the common part):

$$matching\ score = \frac{\sum_{hk}(m_{hk} \cdot x_{hk})}{\min(|\mathcal{R}(I_0)^{(i,0)}|, |\mathcal{R}(I_{i0})^{(i,0)}|)}. \tag{A.46}$$

The matching score measures the ability of the detector to extract distinctive features, i.e., regions which can be reliably described and matched under different variabilities. As pointed out in [155], the matching results should follow the repeatability scores if the regions extracted are distinctive. It should be noted that we use the LIOP descriptor to compute the matching ability instead of using the standard SIFT algorithm as proposed by Mikolajczyk et al. in [155]. Our choice is motivated by recent studies [166] in which the LIOP descriptor outperforms SIFT on the OXFORD-48 dataset and supplementary image pairs with complex illumination changes. Since we are benchmarking the ability of the detectors to extract highly distinctive features and we are not interested in assessing the performances of the descriptors themselves, we choose LIOP as the best performing algorithm up-to-date for our evaluations.

Precision-Recall Curves

To better assess the matching ability of the detectors with respect to increasing radial distortion, we also compute 1-precision vs recall curves following the scheme proposed in [165]. According to such scheme, two descriptors match if their euclidean distance is smaller than a given threshold t. Each test descriptor is compared with each reference descriptor and the number of false and correct matchings is counted in order to compute the precision and recall values corresponding to threshold t using the following formulas:

$$precision = \frac{\#correct\ matchings}{\#matchings} \tag{A.47}$$

$$recall = \frac{\#correct\ matchings}{\#correspondences}. \tag{A.48}$$

The curves are obtained varying the threshold t. An ideal 1-precision vs recall curve would have recall equal to 1 for any precision, while in practice the recall increases as the precision decreases. A steep curve denotes a detector able to produce distinctive regions with a reduced amount of non-distinctive regions. We also report

the threshold vs F-measure curves, where the F-measure is computed as follows [171]:

$$F_\beta = \frac{(1+\beta^2) precision \times recall}{\beta^2 \times precision + recall} \tag{A.49}$$

where $\beta^2 = 0.3$ to weigh precision more than recall. The threshold vs F-measure curves have a retrieval-based interpretation: a good curve would have a high peak for a small threshold, indicating that a high number of regions can be retrieved with little noise.

Note on the Normalization Scheme

The repeatability and matching scores reported in Equations (A.45) and (A.46) are defined normalizing the number of correspondences and matchings by the minimum number of regions detected in the test and reference images. Such normalization scheme, proposed in [155] and fully recognized by the Computer Vision community, is based on the observation that the chosen normalization value is the maximum number of correspondences or matchings which it is possible to achieve. This normalization scheme accounts for those situations in which, due to an extreme amount of the considered variability (e.g., increasing radial distortion, change of viewpoint angle), most of the regions extracted from the reference image are unlikely to be detected by any algorithm in the test image since they are represented by just a few pixels. Nevertheless, it should be noted that, according to such definitions, the scores referring to the same image series but different test images are not in general normalized by the same number. As it shall be clearer later on in our analysis, for this reason, the scores related to different test images of the same series are not directly comparable in a quantitatively fashion and the reported results should be considered indicative instead as pointed out in [155].

A.4.4 Experimental Results and Analysis

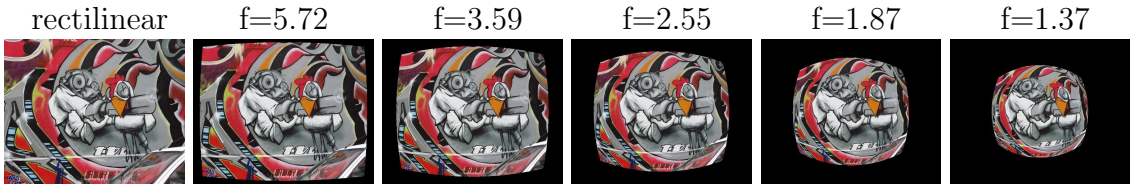
We have performed experiments using both the theoretical camera models based on the ideal projection functions and the division model. For the theoretical camera models, we have used only the OXFORD-48 dataset, while the analysis with the Division Model has been extended to all considered three datasets: OXFORD-48, DASF-HIRES-50 and RDSIFT-39.

Experiments related to Theoretical Distortion Functions

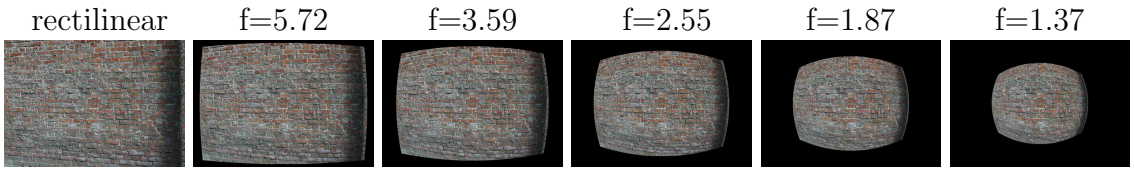
Experiments with the theoretical distortion functions have been performed considering the fisheye camera model and focal lengths discussed in Appendix A.2 and the OXFORD-48 dataset. We employ the equidistance projection reported in Equation (A.6), which is one of the most common for fisheye lenses [151]. We perform two experiments: 1) to evaluate the robustness of the detectors with respect to increasing degrees of fisheye distortion; 2) to evaluate the robustness of the detectors with respect to geometric and photometric variabilities (i.e., viewpoint changes, zoom and rotation, image blur, JPEG compression, light changes) combined with a medium fisheye distortion.

In order to evaluate the robustness of the detectors with respect to the fisheye distortion, two image series (one structured and one textured) characterized by increasing degrees of fisheye distortion are built from the reference images of the "Graffiti" and "Wall" series of OXFORD-48 dataset. Figure A.17(a) and Figure A.17(b) show the image series used to test the robustness of the detectors with respect to increasing degrees of fisheye distortion. Figure A.17(c) shows the results related to the robustness of the detectors with respect to the increasing fisheye distortion. It can be noted that the repeatability performances of the detectors decline slowly, with overall better performances on the structured scene. The MSER extractor performs slightly better on both the structured scene and the textured one. MSER has a more discriminative power (i.e., it is able to detect regions which are more distinctive) both on the structured scene and on the textured one, while the other detectors have similar matching performances. These results show that all the compared detectors are robust to the variability introduced by increasing degrees of fisheye distortion.

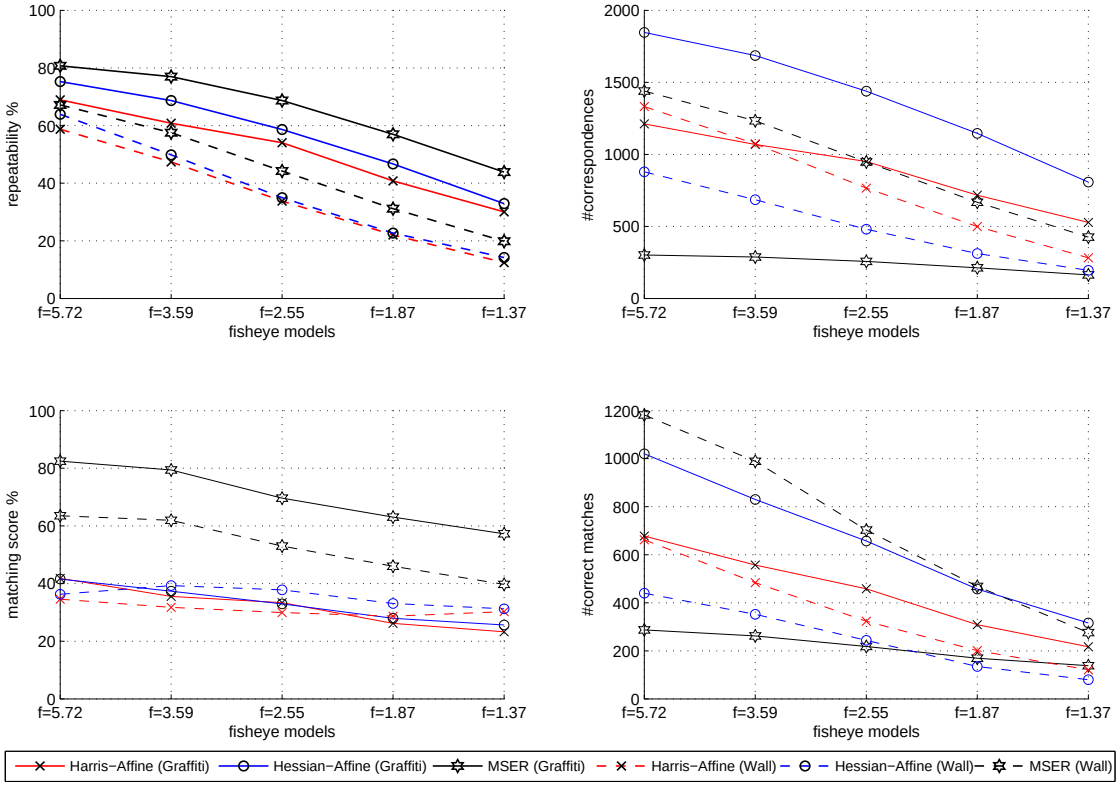
Figure A.18(a,b) - A.20(a,b) and Figure A.21(a) - A.22(a) show the image series used to test the robustness of the detectors with respect to the combination of fisheye distortion and a specific photometric or geometric variability. The image series are obtained distorting the images of the OXFORD-48 dataset with the theoretical camera model considering a medium focal length equal to $f = 2.55\ mm$. Figure A.18(c) - A.20(c) and Figure A.21(b) - A.22(b) show the related repeatability and matching results, as well as the number of correspondences and number of matches. The relative ordering of the detectors and the decay of the performances in



(a) "Graffiti" series used to test the performances of the descriptors against increasing degrees of fisheye distortion.

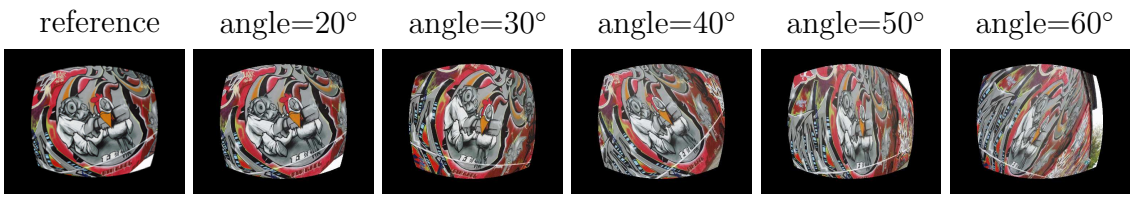


(b) "Wall" series used to test the performances of the descriptors against increasing degrees of fisheye distortion.

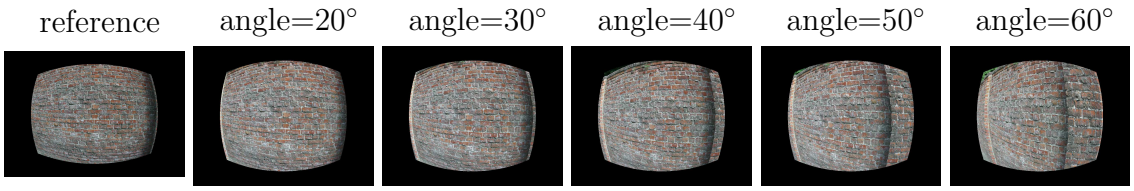


(c) Performances of the descriptors with respect to increasing degrees of fisheye distortion.

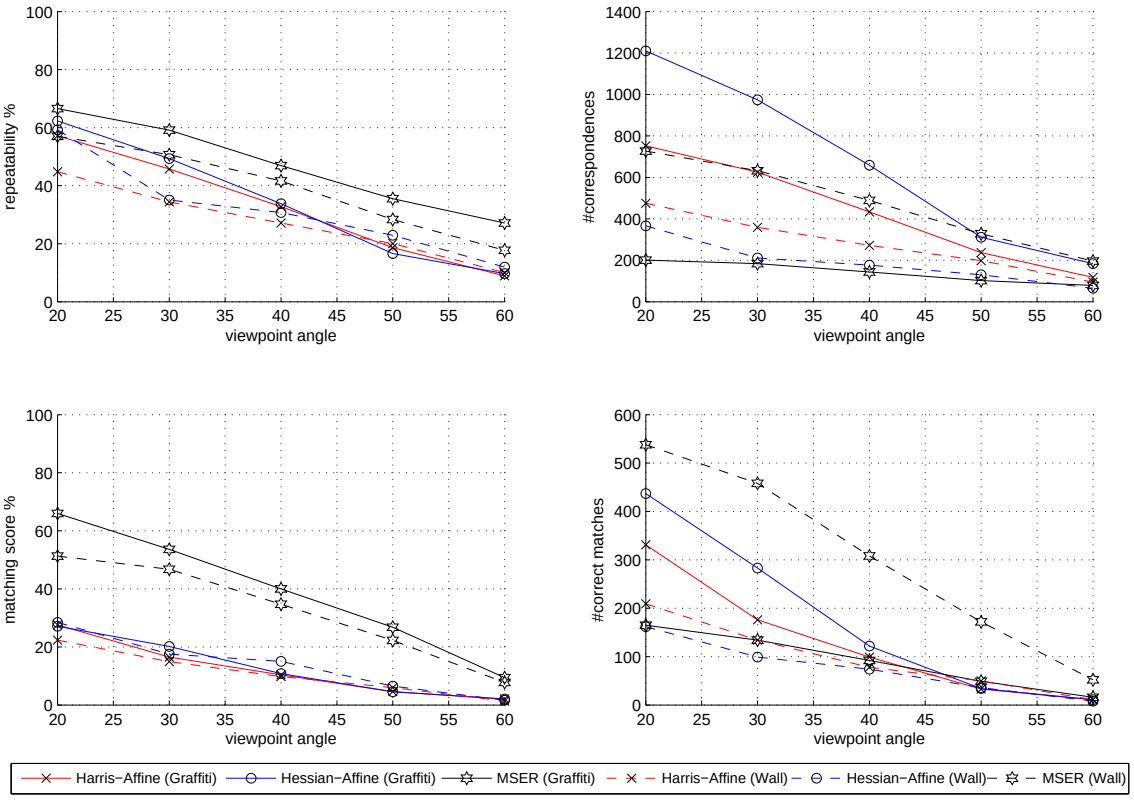
Figure A.17: Experiments performed using a theoretical fisheye camera model (Appendix A.4.1) making use of the equidistance projection function reported in Equation (A.6). Data (a) - (b) and results (c) related to the experiments to evaluate the performances of the detectors with respect to the increasing fisheye distortion.



(a) "Graffiti" series: fisheye distortion + viewpoint change on a structured scene.

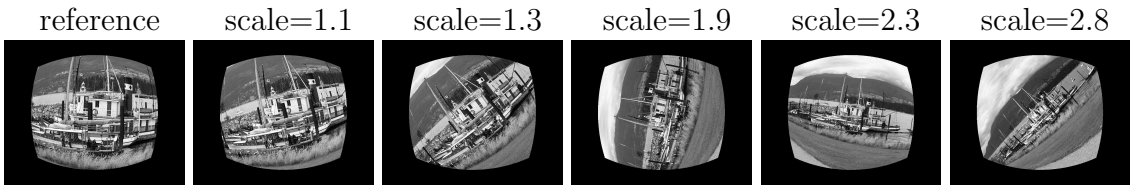


(b) "Wall" series: fisheye distortion + viewpoint change on a textured scene.

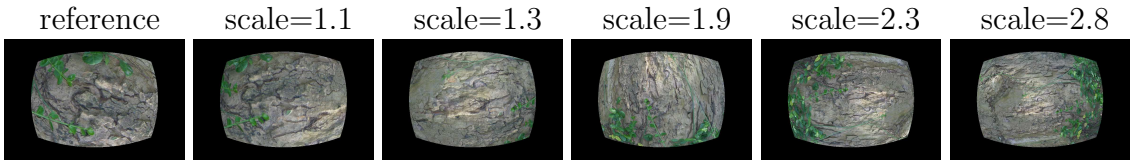


(c) results related to the performances of the descriptors with respect to fisheye distortion + viewpoint change.

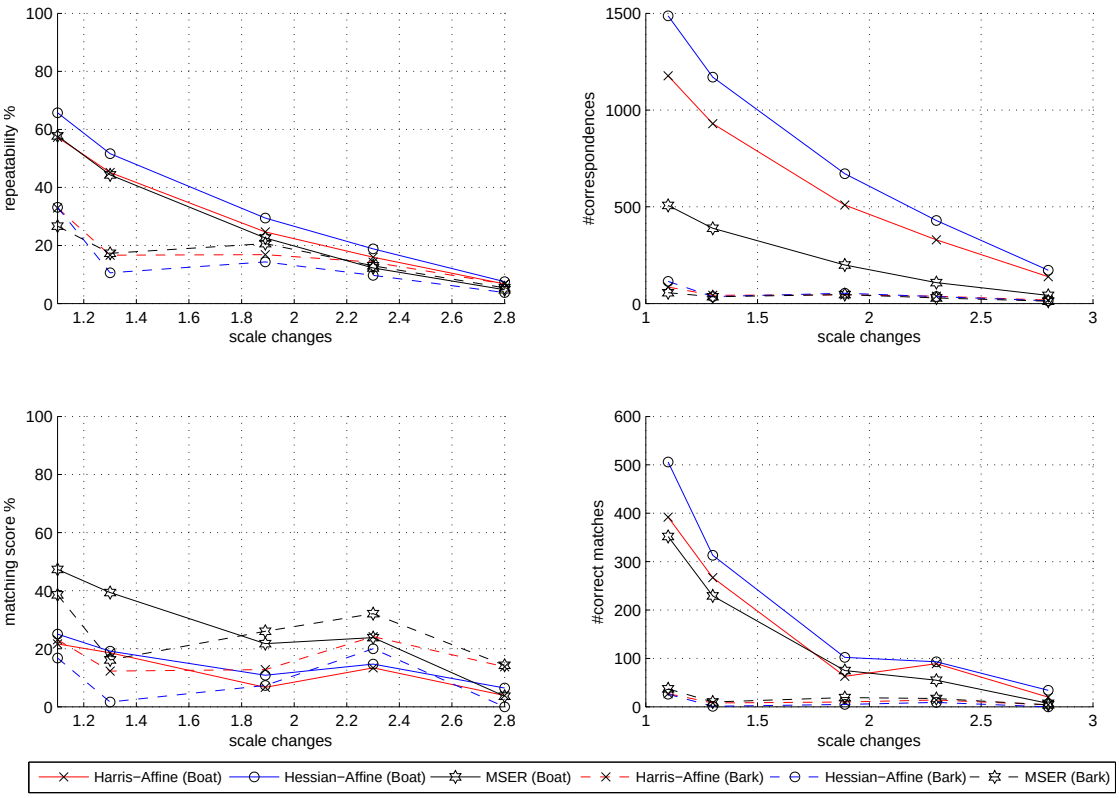
Figure A.18: Experiments performed using a theoretical fisheye camera model (Appendix A.4.1) making use of the equidistance projection reported in Equation (A.6). Data (a) - (b) and results (c) related to the experiments to evaluate the performances of the detectors with respect to the combination of a medium fisheye distortion ($f = 2.55\ mm$) and the change of viewpoint.



(a) "Boat" series: fisheye distortion + scale change on a structured scene.

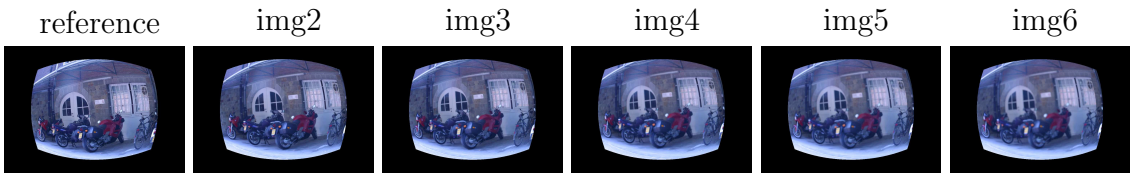


(b) "Bark" series: fisheye distortion + scale change on a textured scene.



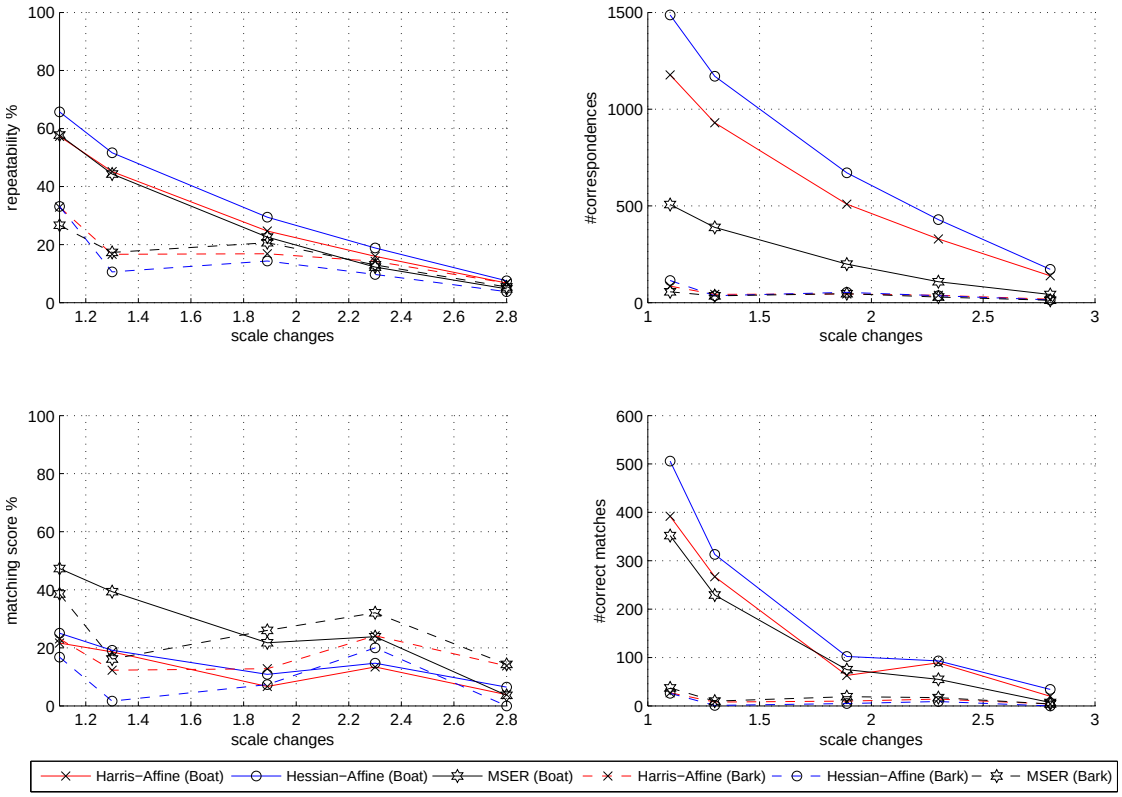
(c) results related to the performances of the descriptors with respect to the fisheye distortion + zoom and rotation.

Figure A.19: Experiments performed using a theoretical fisheye camera model (Appendix A.4.1) making use of the equidistance projection reported in Equation (A.6). Data (a) - (b) and results (c) related to the experiments to evaluate the performances of the detectors with respect to the combination of a medium fisheye distortion ($f = 2.55\ mm$) and zoom + rotation.



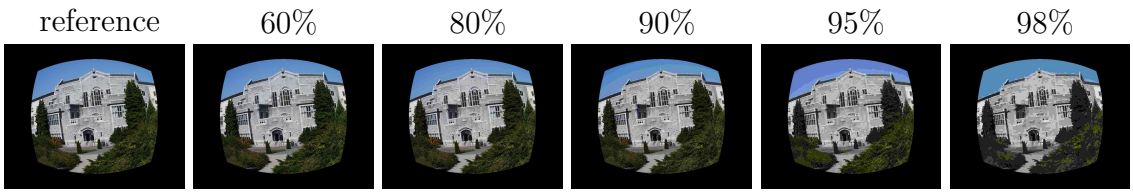
(a) "Bikes" series: fisheye distortion + increasing blur on a structured scene.

(b) "Trees" series: fisheye distortion + increasing blur on a textured scene.

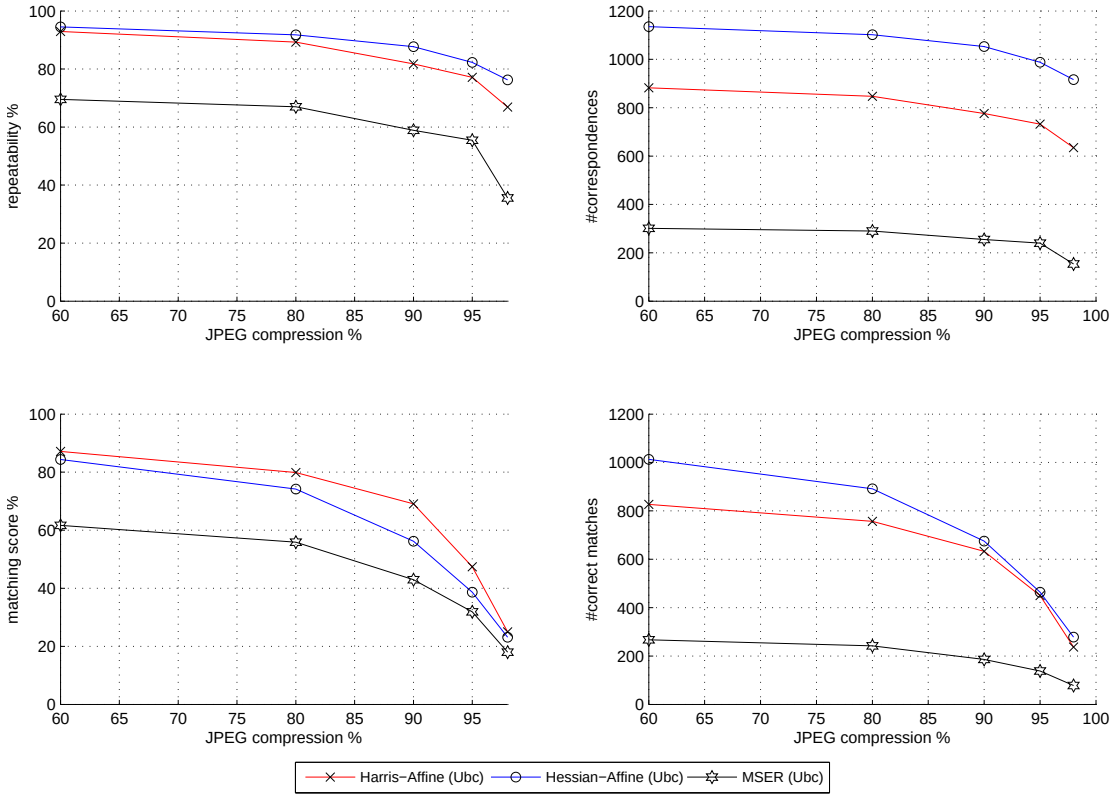


(c) results related to the performances of the descriptors with respect to the fisheye distortion + increasing blur.

Figure A.20: Experiments performed using a theoretical fisheye camera model (Appendix A.4.1) making use of the equidistance projection reported in Equation (A.6). Data (a) - (b) and results (c) related to the experiments to evaluate the performances of the detectors with respect to the combination of a medium fisheye distortion ($f = 2.55\ mm$) and increasing blur.

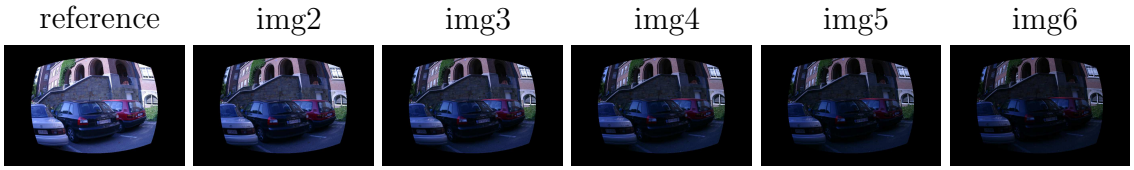


(a) "Ubc" series: fisheye distortion + increasing jpeg compression.

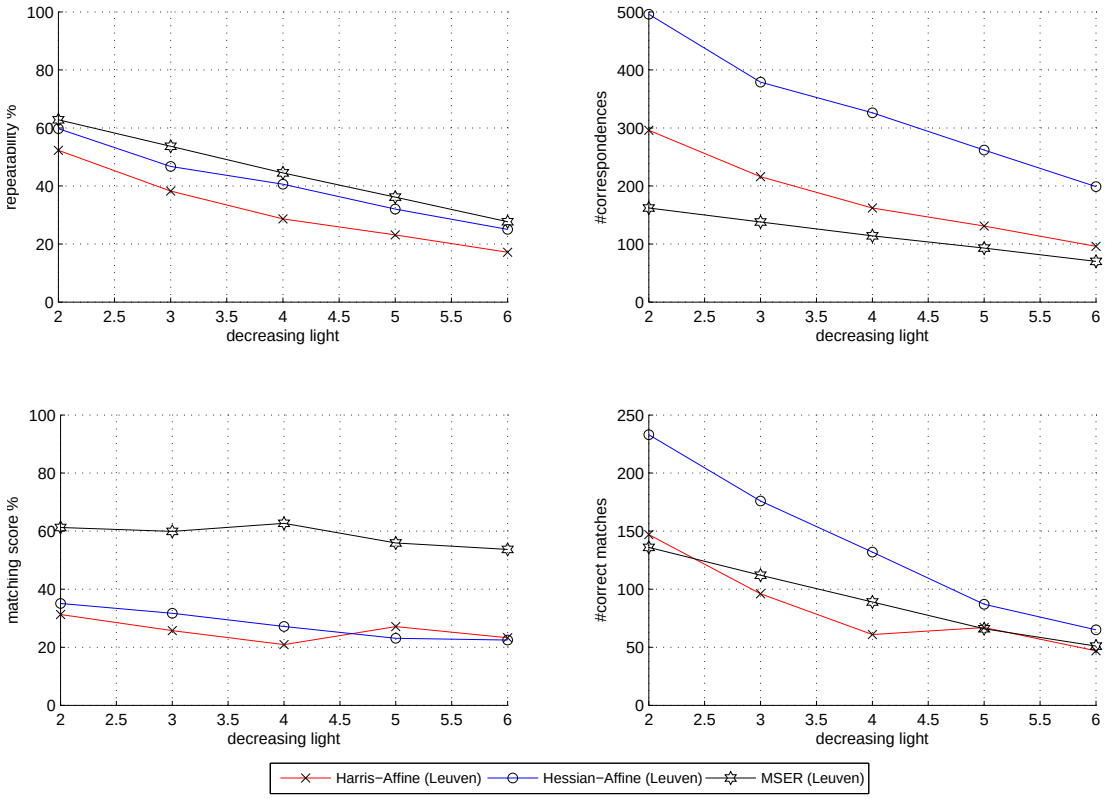


(b) results related to the performances of the descriptors with respect to the fisheye distortion + increasing jpeg compression.

Figure A.21: Experiments performed using a theoretical fisheye camera model (Appendix A.4.1) making use of the equidistance projection reported in Equation (A.6). Data (a) and results (b) related to the experiments to evaluate the performances of the detectors with respect to the combination of a medium fisheye distortion ($f = 2.55\ mm$) and increasing jpeg compression.



(a) "Leuven" series: fisheye distortion + decreasing light.

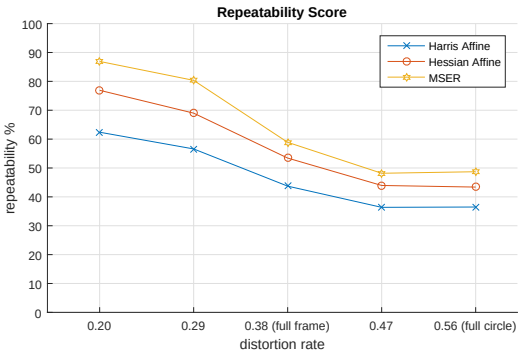


(b) results related to the performances of the descriptors with respect to the fisheye distortion + decreasing light.

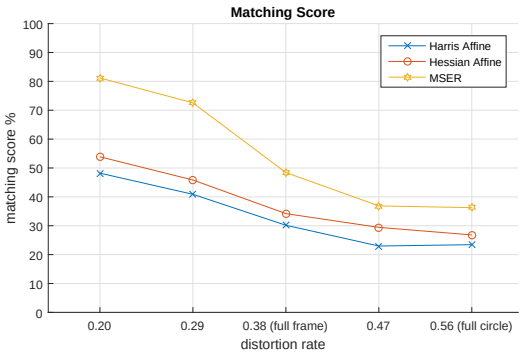
Figure A.22: Experiments performed using a theoretical fisheye camera model (Appendix A.4.1) making use of the equidistance projection reported in Equation (A.6). Data (a) and results (b) related to the experiments to evaluate the performances of the detectors with respect to the combination of a medium fisheye distortion ($f = 2.55\ mm$) and decreasing light.

reference 20% 29% 38%* 47% 56%**

(a) Sample Image Series from DASF-HIRES-50 (Increasing Radial Distortion)



(b) Repeatability Scores



(c) Matching Scores

Figure A.23: Results performed using the division model on the DASF-HIRES-50 dataset to assess the robustness of affine detectors with respect to increasing degrees of radial distortion. All numbers are obtained averaging the results for all the image series contained in DASF-HIRES-50. (a) Sample image series from DASF-HIRES-50. (b) Repeatability scores for different amounts of radial distortion. (c) Matching scores for different amounts of radial distortion.

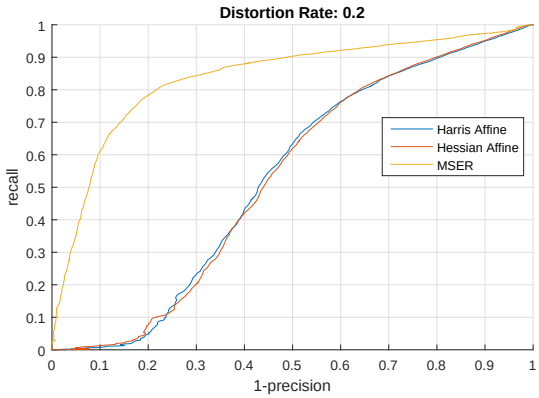
the fisheye domain are in agreement with the results of similar tests in the rectilinear one reported in [165]. This underlines that the selected detectors behave similarly in both the fisheye domain and the rectilinear one, and suggests that, according to theoretical projection functions, affine covariant region detectors can still be used for real applications directly in the fisheye domain.

Experiments related to the Division Model Distortion Function

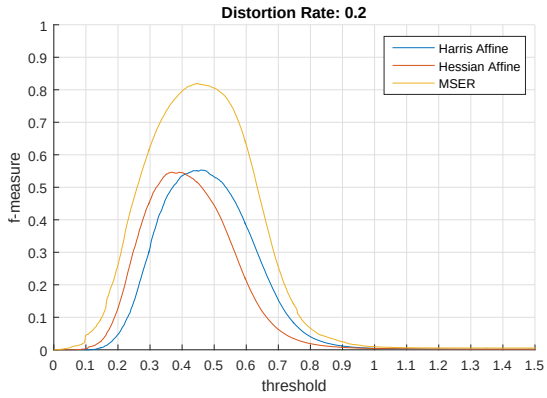
Similar experiments have been performed using the division model. Specifically, we have performed three sets of experiments using all three datasets discussed in Appendix A.3. The first set of experiments is aimed at assessing the robustness of the detectors to increasing amounts of radial distortion and is performed on the DASF-HIRES-50 dataset. To perform tests on different degrees of distortion, we consider the following distortion rates: 0.2, 0.29, 0.38 (full frame configuration), 0.47 and 0.56

(full circle configuration). The reference high resolution images of DASF-HIRES-50 are mapped to distorted images of resolution 1024×768 pixels employing the division model and using the methods discussed in Appendix A.2.3. The second set of experiments is aimed at assessing the performances of the considered detectors when the variabilities present in the OXFORD-48 dataset are combined with radial distortion. These experiments have been performed on OXFORD-48. The third set of experiments is performed on the RDSIFT-39 dataset and is intended to extend the analysis to images acquired using real fisheye lenses. Figure A.23 to A.32 report the results of the performed experiments. It should be noted that, due to the normalization scheme discussed in Appendix A.4.3, the curves related to the repeatability and matching scores are not guaranteed to be strictly monotonically decreasing with respect to increasing amounts of a considered variability as the reader could expect. As pointed out earlier and in [155], such results have an indicative rather than quantitative value and the reader is advised to focus more on the general trends of the presented curves rather than on local configurations. In the following, we present and discuss the results related to the three considered sets of experiments.

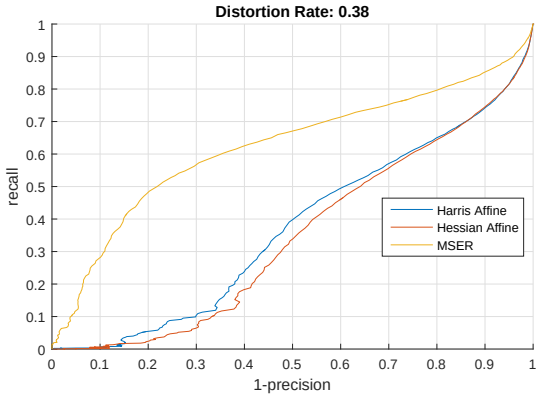
Figure A.23 and Figure A.24 report the results performed on the DASF-HIRES-50 dataset to assess the robustness of affine detectors with respect to increasing degrees of radial distortion. All the reported scores have been obtained by averaging the results for the different image series contained in the dataset. This allows us to draw general conclusions on the performances of the detectors under analysis. Figure A.23(a) reports a sample image series from the dataset. Figure A.23(b) and Figure A.23(c) show that all detectors retain good performances for increasing degrees of fisheye distortion. Interestingly, MSER clearly outperforms the other detectors on both the repeatability and matching tests. In particular, the superior performances of MSER in the matching test highlight that the regions extracted by MSER tend to be more distinctive than the ones extracted by the competitors under the influence of radial distortion. This observation is strengthen by the 1-precision vs recall and threshold vs F-measure curves shown in Figure A.24. Moreover, the decays of the curves shown in Figure A.23(b) and Figure A.23(c) are reminiscent of the results related to the robustness of the detectors with respect to affine variabilities such as the change of viewpoint angle (solid lines in Figure A.25). This observation



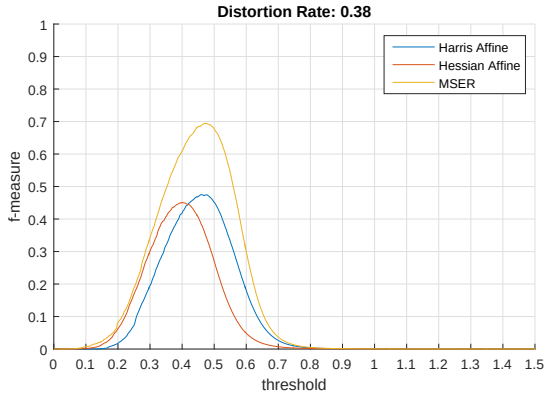
(a) Precision-Recall curve (little distortion)



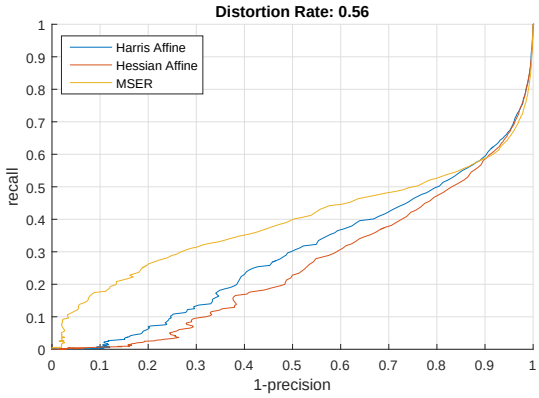
(b) F-Measure curve (little distortion)



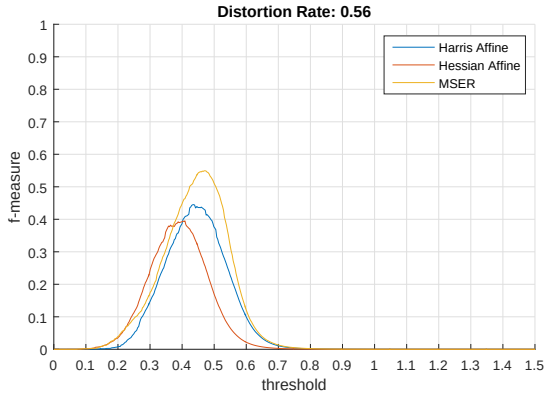
(c) Precision-Recall curve (full frame)



(d) F-Measure curve (full frame)



(e) Precision-Recall curve (full circle)



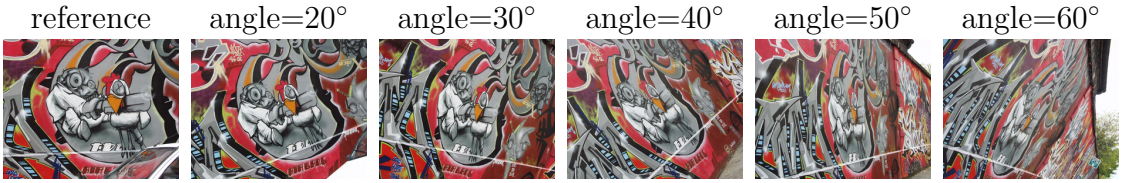
(f) F-Measure curve (full circle)

Figure A.24: Results performed using the division model on the DASF-HIRES-50 dataset to assess the robustness of affine detectors with respect to increasing degrees of radial distortion. All numbers are obtained averaging the results for all the image series contained in DASF-HIRES-50. (a), (c) and (e) 1-precision vs recall curves for different amounts of radial distortion. (b), (d) and (f) threshold vs F-measure curves for different amounts of radial distortion.

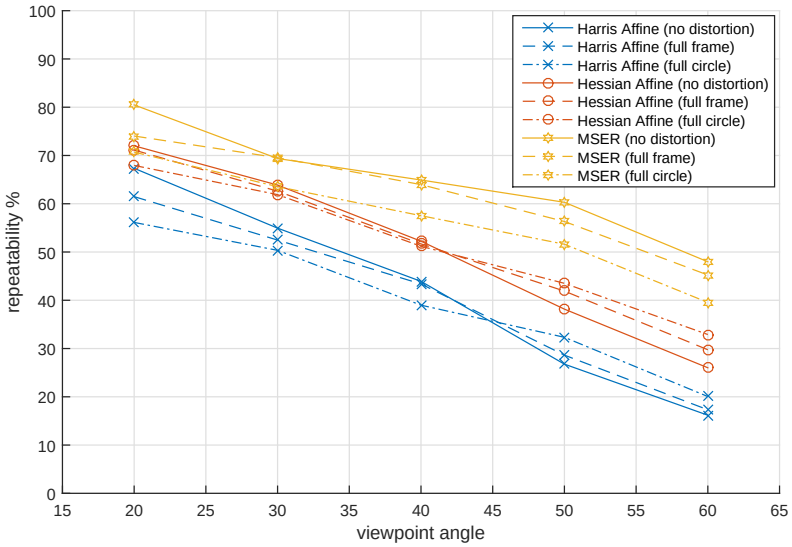
supports our premise that affine covariant region detectors can locally model the radial distortion introduced by a fisheye camera as an affine variability.

Figure A.25 to A.32 show the results of the experiments performed on the OXFORD-48 dataset to assess the performances of affine detectors with respect to the combination of fisheye distortion and a specific variability. Each figure reports the original image series on which experiments are performed, the repeatability and matching scores related to a specific variability (i.e., change of viewpoint angle, scale change, increasing blur, JPEG compression, decreasing light). Specifically, each figure reports the results related to the original image series when no radial distortion is introduced (solid lines), the results related to the series to which a full frame distortion has been added (dashed lines) and the results related to the series to which a full circle distortion has been added (dot-dashed lines). It should be noted that, since the reference image is never distorted in OXFORD-48, in each plot all the data series are related to the same reference image. The results are in line with the ones reported in the benchmark of [155] also when radial distortion is added. No detector performs systematically better than the competitors on all considered image series and the relative ordering of the curves tends to change for the structured and textured scenes even when the variability under analysis is the same. However, some general considerations can be made. The combination of radial distortion and the variabilities present in the OXFORD-48 dataset (dashed and dot-dashed lines) degrades the performances of the detectors. Nevertheless, the curves related to the distorted series are often characterized by decays and relative ordering similar to the ones of the original series not affected by distortion (solid lines). This is especially true for the structured scenes both for repeatability and matching scores. This observation is a further evidence of how the introduction of the fisheye distortion is in most of the cases handled by the detectors as an additional variability to cope with. As general remarks, moreover, the Hessian Affine detector achieves the best repeatability results in most of the configurations, while the MSER detector extracts highly distinctive regions in all the cases (i.e., the matching results follow the repeatability results).

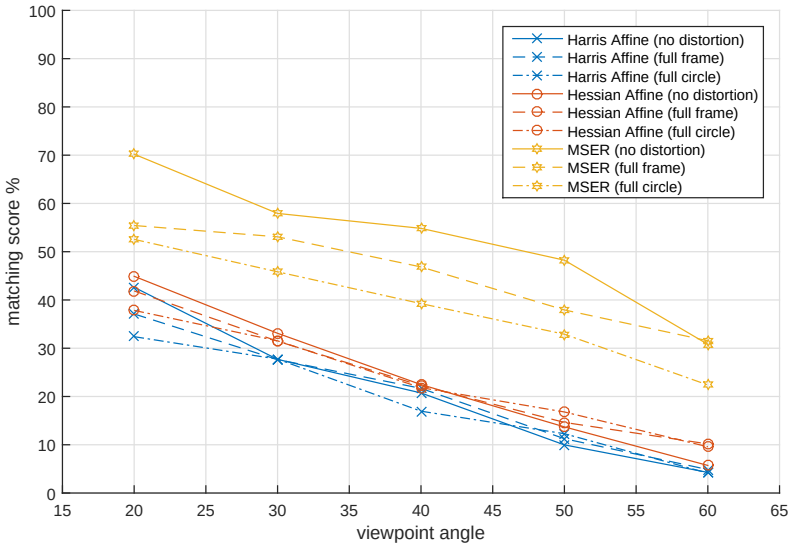
Table A.2 finally reports the results of the experiments performed on the RDSIFT-39 dataset comprising real fisheye images. For each image series and feature detector, we report the average repeatability and matching ability scores over the 78 image



(a) Graffiti Image Series (Structured - Viewpoint Change)

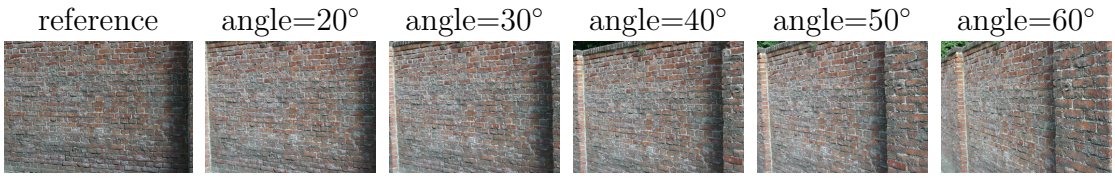


(b) Repeatability Scores - Graffiti (structured)

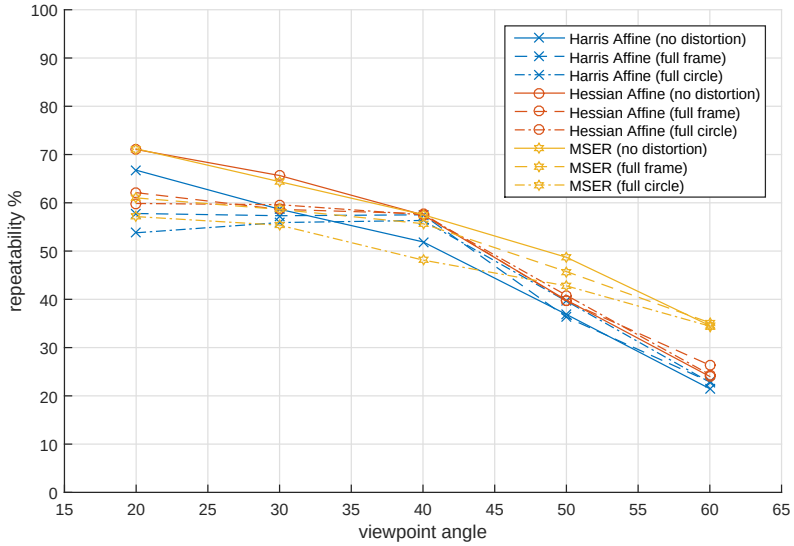


(c) Matching Scores - Graffiti (structured)

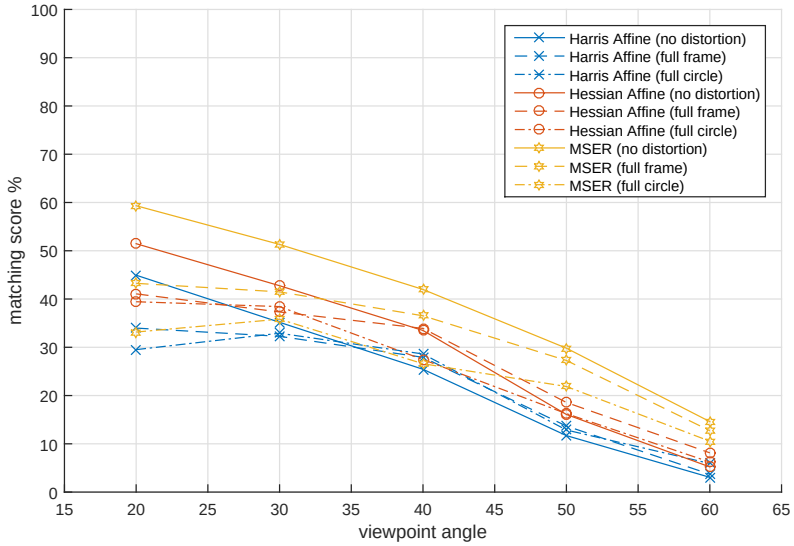
Figure A.25: Results of the experiments performed using the division model on the OXFORD-48 dataset to assess the robustness of affine detectors with respect to the combination of fisheye distortion and change of viewpoint angle for a structured scene. (a) Graffiti image series (structured scene). (b) Repeatability scores for the graffiti image series. (c) Matching scores for the graffiti image series.



(a) Wall Image Series (Textured - Viewpoint Change)

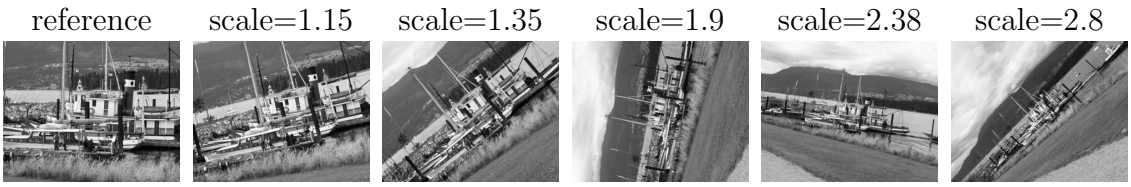


(b) Repeatability Scores - Wall (textured)

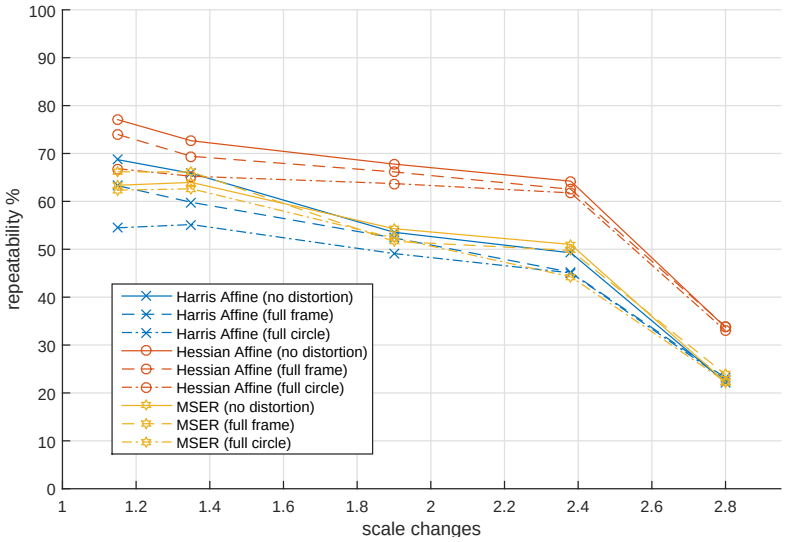


(c) Matching Scores - Wall (textured)

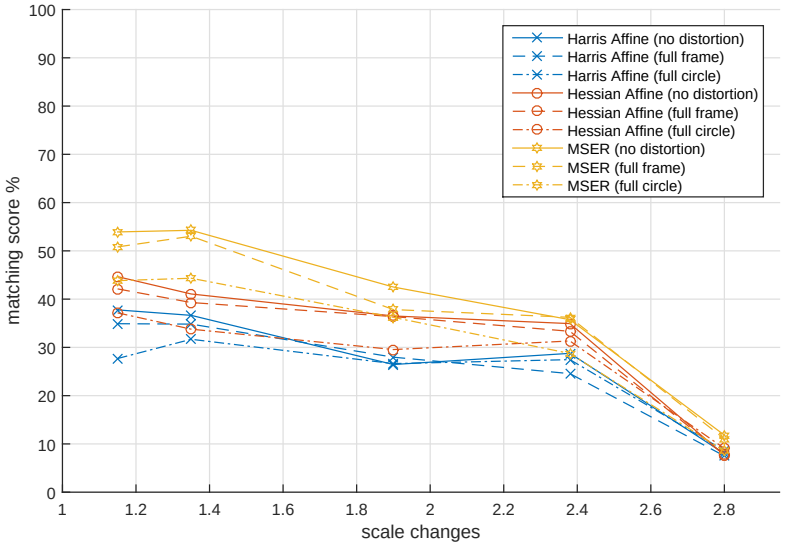
Figure A.26: Results of the experiments performed using the division model on the OXFORD-48 dataset to assess the robustness of affine detectors with respect to the combination of fisheye distortion and change of viewpoint angle for both a textured scene. (a) Wall image series (textured scene). (b) Repeatability scores for the wall image series. (c) Matching scores for the wall image series.



(a) Boat Image Series (Scale Changes - Structured)

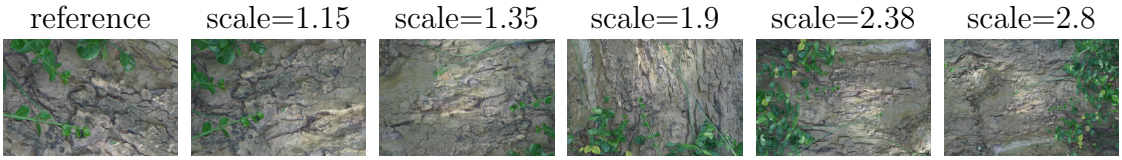


(b) Repeatability Scores - Boat (structured)

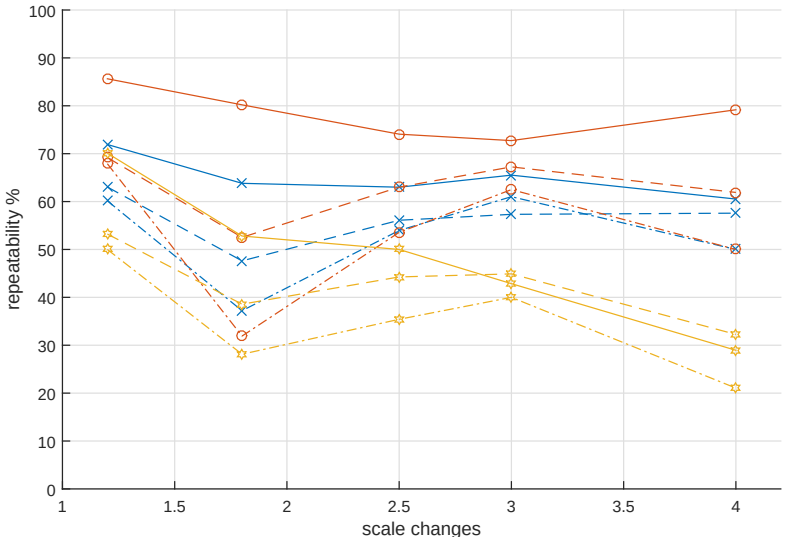


(c) Matching Scores - Boat (structured)

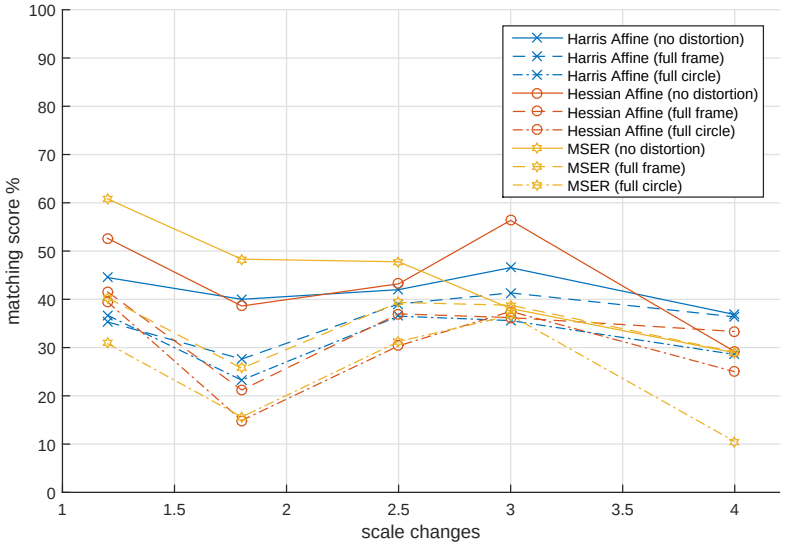
Figure A.27: Results of the experiments performed using the division model on the OXFORD-48 dataset to assess the robustness of affine detectors with respect to the combination of fisheye distortion and scale changes for a structured scene. (a) Boat image series (structured scene). (b) Repeatability scores for the boat image series. (c) Matching scores for the boat image series.



(a) Bark Image Series (Scale Changes - Textured)



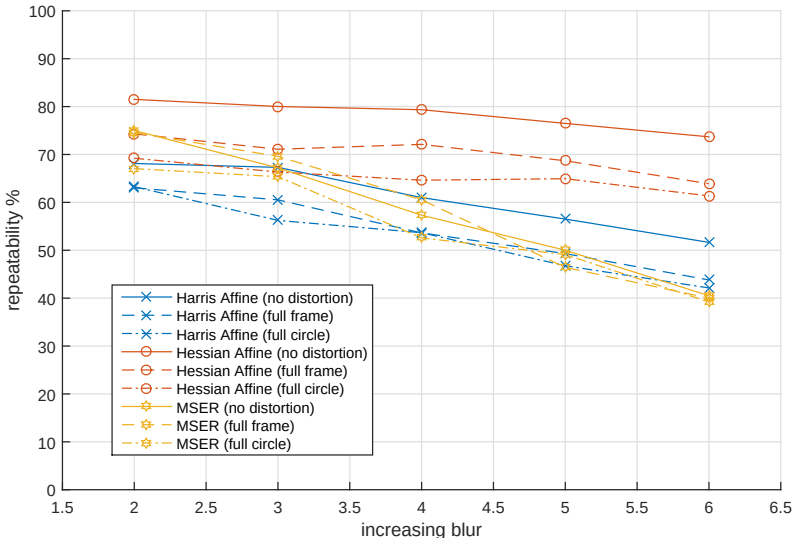
(b) Repeatability Scores - Bark (textured)



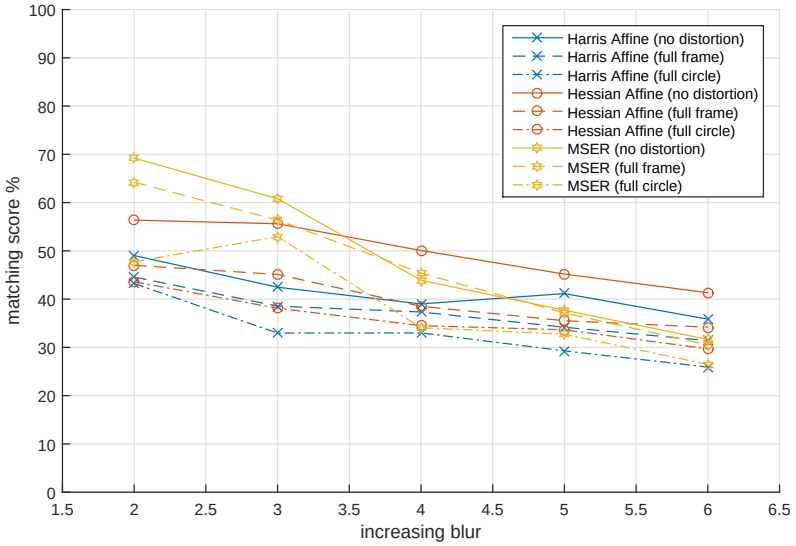
(c) Matching Scores - Bark (textured)

Figure A.28: Results of the experiments performed using the division model on the OXFORD-48 dataset to assess the robustness of affine detectors with respect to the combination of fisheye distortion and scale changes for a textured scene. (a) Bark image series (b) Repeatability scores for the bark image series. (c) Matching scores for the bark image series. The legend of (b) applies to (c) as well.

(a) Bikes Image Series (Increasing Blur - Structured)



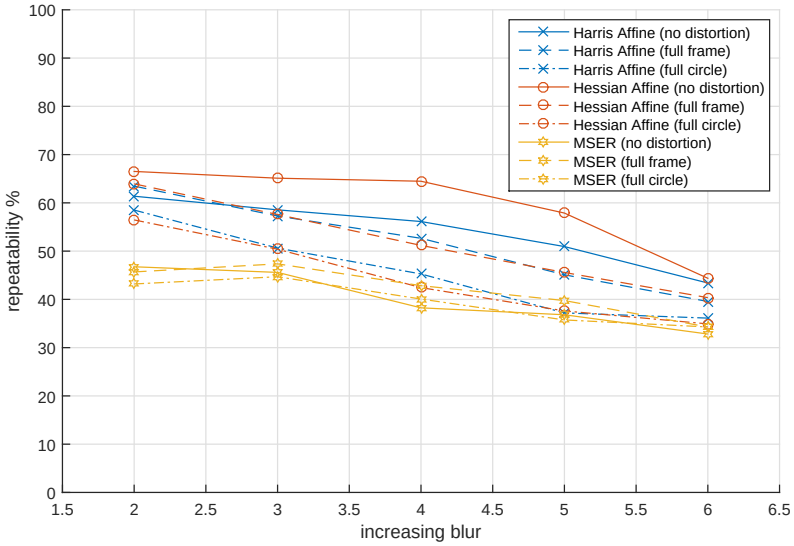
(b) Repeatability Scores - Bikes (structured)



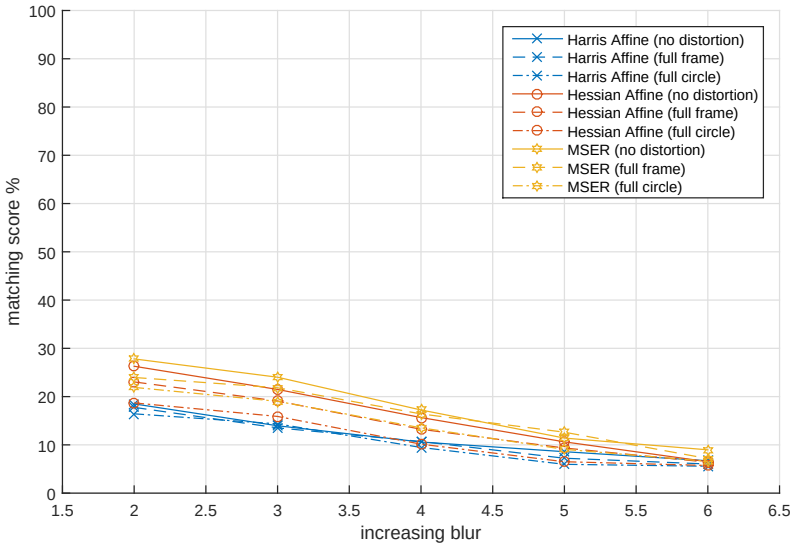
(c) Matching Scores - Bikes (structured)

Figure A.29: Results of the experiments performed using the division model on the OXFORD-48 dataset to assess the robustness of affine detectors with respect to the combination of fisheye distortion and increasing blur for a structured scene. (a) Bikes Image Series. (b) Repeatability scores for the bikes image series. (c) Matching scores for the bikes image series.

(a) Trees Image Series (Increasing Blur - Textured)

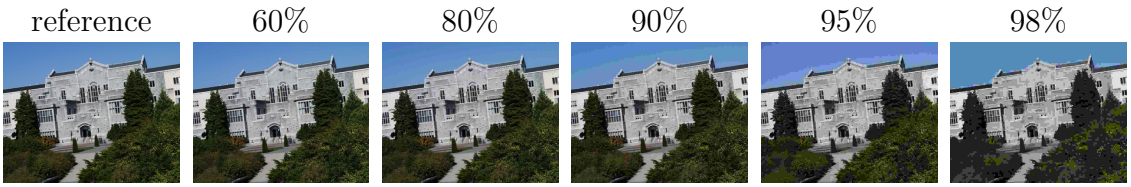


(b) Repeatability Scores - Trees (textured)

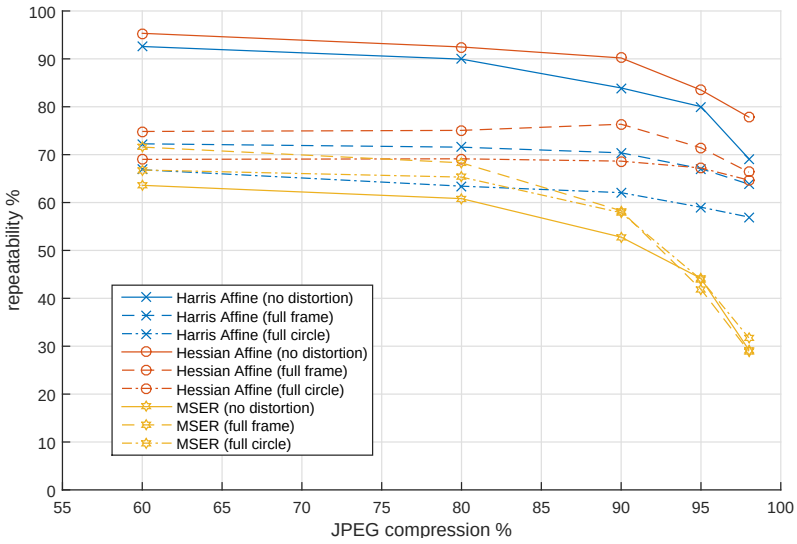


(c) Matching Scores - Trees (textured)

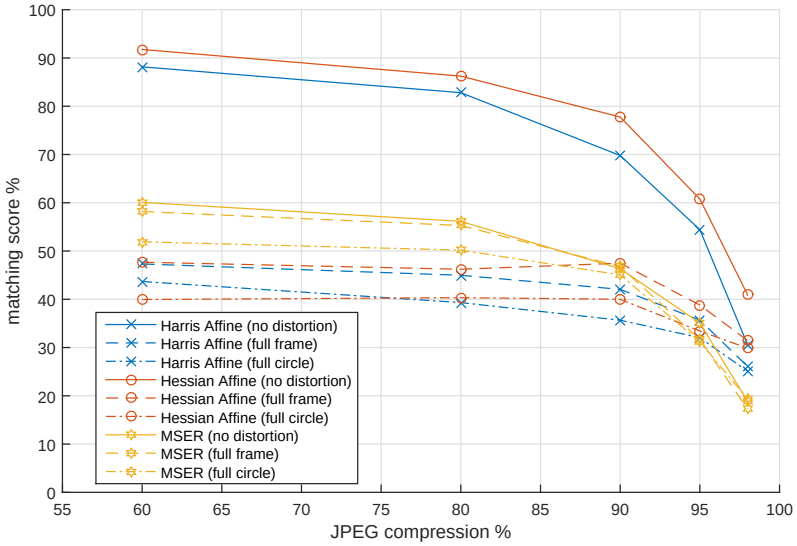
Figure A.30: Results of the experiments performed using the division model on the OXFORD-48 dataset to assess the robustness of affine detectors with respect to the combination of fisheye distortion and increasing blur for a textured scene. (a) Trees Image Series (textured scene). (b) Repeatability scores for the trees image series. (c) Matching scores for the trees image series.



(a) UBC image series (JPEG compression)



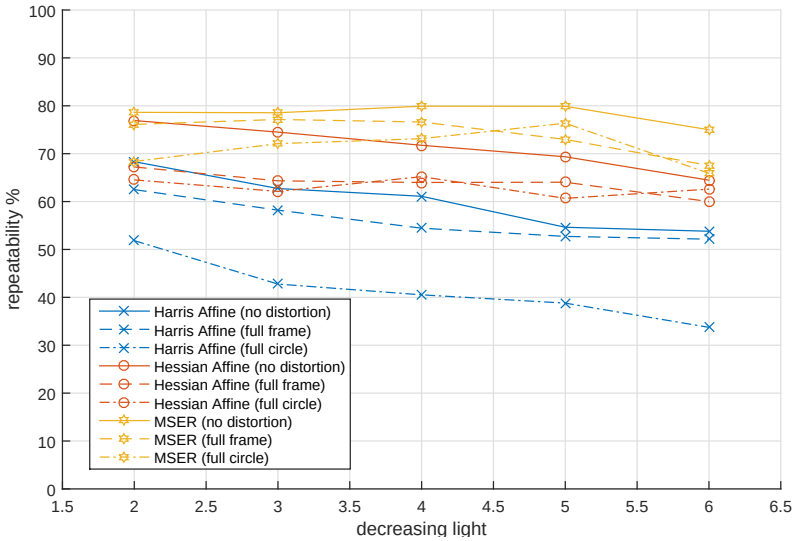
(b) Repeatability Scores - UBC



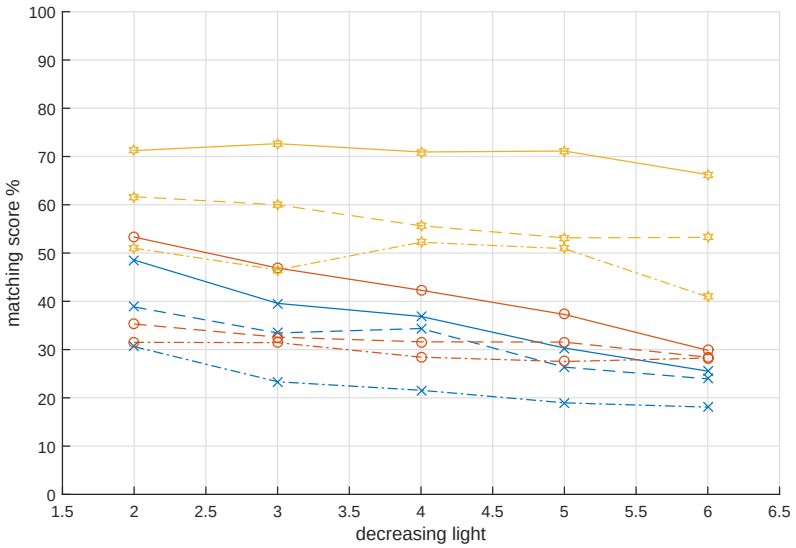
(c) Matching Scores - UBC

Figure A.31: Results performed using the division model on the OXFORD-48 dataset to assess the robustness of affine detectors with respect to the combination of fisheye distortion and increasing JPEG compression. (a) UBC Image Series. (b) Repeatability scores for the UBC image series. (c) Matching scores for the UBC image series.

(a) Leuven Image Series (Decreasing Light)



(b) Repeatability Scores - Leuven



(c) Matching Scores - Leuven

Figure A.32: Results performed using the division model on the OXFORD-48 dataset to assess the robustness of affine detectors with respect to the combination of fisheye distortion and decreasing light. (a) Leuven image series. (b) Repeatability scores for the leuven image series. (c) Matching scores for the leuven image series. The legend of (b) applies to (c) as well.

Series	**Repeatability %**			**Matching ability %**		
Affine Detector	**Harris**	**Hessian**	**MSER**	**Harris**	**Hessian**	**MSER**
S1 ($d = 0.13$)	61.94	69.34	**74.73**	36.09	39.14	**59.20**
S2 ($d = 0.19$)	60.14	71.00	**72.24**	32.32	37.33	**54.23**
S3 ($d = 0.54$)	23.54	27.97	**32.88**	12.80	13.65	**25.68**
S1 ($d = 0.13$), rect	68.00	75.47	**77.22**	40.28	41.73	**62.29**
S2 ($d = 0.19$), rect	63.10	73.88	**73.97**	33.55	38.74	**53.53**
S3 ($d = 0.54$), rect	43.41	52.91	**57.32**	26.87	24.99	**44.37**

Table A.2: Results related to the RDSIFT-39 dataset of images acquired using real fisheye lenses. The dataset contains three series (S1 to S3) acquired using three different cameras. For each series, experiments have been performed on all 78 available image pairs.

pairs. The last three rows report results obtained performing rectifying the image prior to extracting affine covariant features. The results reported in Table A.2 confirm the general findings discussed in the previous sections. In particular, repeatability and matching scores computed on real fisheye images are generally lower, but still consistent with the ones reported in Figure A.25-A.26 (viewpoint change + radial distortion) and Figure A.27-A.28 (scale and rotation transformations + radial distortion). As observed in the previous experiments, regions extracted by the MSER detector are highly distinctive (matching scores marked in bold in Table A.2 follow the trend of repeatability scores). In agreement with the experiments performed on DASF-HIRES-50, the MSER detector systematically outperforms the competitors both in terms of repeatability and matching ability. Moreover, when the distortion rate is low (i.e., S1 and S2 in Table A.2), affine covariant feature detectors perform reasonably well directly on fisheye images as compared to employing rectification. In the case of low distortion, in fact, using affine covariant region detectors directly implies an average performance drop under the 3% with respect to both repeatability and matching ability scores, which suggests that radial distortion is successfully modeled as an additional affine variability. When distortion is severe (i.e., S3 in Table A.2), performing rectification allows to improve both repeatability and matching ability by a good margin, leading to average gains of about 23% for repeatability and 15% for matching ability. It should be noted that, even in the case of severe distortion, results obtained on RDSIFT-39 are still coherent with those obtained on OXFORD-48, suggesting that direct employment of affine covariant region detectors on fisheye images is able to produce usable results. This can be particularly useful

when rectification is not a viable option, e.g., when the camera is not known (and hence cannot be calibrated) in advance.

A.4.5 Discussion

The proposed analysis was carried to investigate the direct applicability of affine covariant region detectors on fisheye images. Relying on both theoretical fisheye camera models and the Division Model for modeling radial distortion, we have provided both theoretical and experimental evidence that affine region detectors can successfully deal with radial distortion as a local affine transformation. Specifically, inspired by the work of Mikolajczyk et al. [155], we have designed a series of experiments aimed at assessing the performances of three popular region detectors, i.e., MSER, Hessian Affine, Harris Affine, with respect to increasing radial distortion. We have also tested the combination of the variabilities included in the OXFORD-48 dataset with two different degrees of radial distortion and performed testes on images acquired using three real fish-eye lenses. Interestingly, MSER outperformed the Hessian and Harris affine region detectors in both the repeatability and matching tests in the experiments related to the increasing radial distortion and on images acquired using real fisheye lenses. The evaluations carried on the OXFORD-48 dataset have shown that the detectors behave consistently when the scene variability is combined with radial distortion, providing further evidence that radial distortion is effectively modeled as an additional affine variability by the detectors. Tests on images acquired using real fisheye lenses show that affine region detectors are able to handle low levels of radial distortion making rectification avoidable. When distortion is severe, affine region detectors yield results consistent with the ones obtained in the presence of strong scale and rotation transformation with artificially distorted images. The proposed analysis can be exploited in all the application domains where the input images are acquired by unknown, non-calibrated cameras (both fisheye and rectilinear).

A.5 Direct Estimation of the Gradient of Distorted Images

Image gradients are fundamental features commonly used in a number of applications including image enhancement and edge extraction [172, 173], object, scene and key-point representation [7, 138, 174, 90], and gradient-domain-based image processing [175, 176, 177, 178]. As it is shown in Figure A.33, a conventional computation of the gradients directly on wide angle images would be deceived by the presence of radial distortion, while in practice many applications require a result similar to the ideal gradients depicted in Figure A.33(b) [136, 150]. While images could be rectified in order to compute correct gradients, this approach is not desirable for motivations similar to the ones discussed earlier in Appendix A.4. Hence, we focus on the direct (i.e., without rectification) geometrically-correct estimation of the gradients of distorted images.

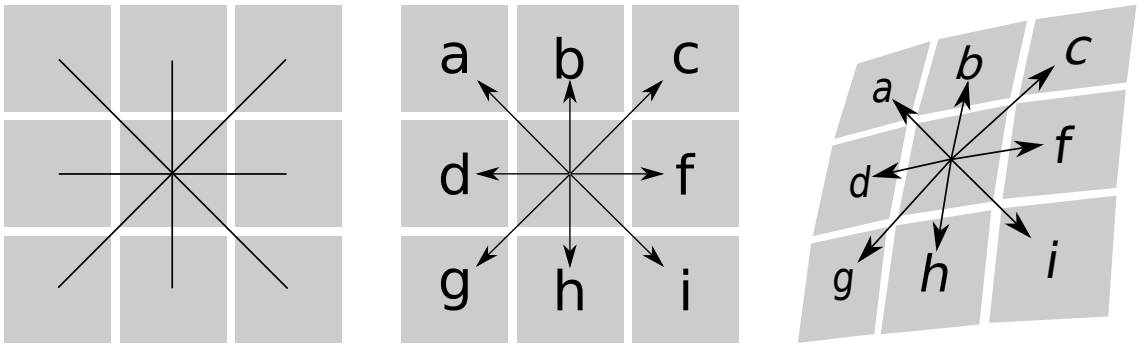
Some methods for estimating the gradients of wide angle images without performing the rectification have already been investigated by the researchers. In [136, 160] the gradient estimated in the distorted domain with standard Sobel filters is corrected using an adaptive Jacobian correction matrix derived from the differential chain rule. Authors of [150] propose to estimate the gradients of catadioptric images by using an operator defined according to the geometry of the catadioptric mirror.

We derive a family of adaptive kernels for the geometrically-correct estimation of the gradients of wide angle images. The proposed kernels aim to be invariant to radial distortion and hence they are designed to be beneficial for a number of gradient-based applications (such as key-point matching, object and people detection [7, 138]), when they are deployed to wide angle camera systems. The derivation of our method is obtained by generalizing the standard Sobel operator to the case of non-Euclidean surfaces in order to take into account the geometrical transformation affecting the image. The derived filters adapt their shape according to the location on which they are computed in order to unevenly weighting the contributes of the estimated directional derivatives to compensate for distortion. The only requirement to compute the proposed filters is that the distortion function is known and invertible. The distortion function can be obtained by calibration when the camera is known, as usually happens in surveillance, automotive and robotics. In the

(a) Rectilinear Image

(b) Wide Angle Image

Figure A.33: Gradient estimation of not-distorted and wide angle images. (a) A not-distorted image along with the gradient directions (solid red arrows) of some sample edges. (b) Wide angle counterpart of (a) along with the gradient directions (solid red arrows) of some sample edges deceived by the radial distortion. The ideal gradient directions are reported as dashed blue arrows.



(a) (b) (c)

Figure A.34: A diagram of Sobel's rationale. (a) The 4 main directions in a 3×3 neighborhood of a given point. (b) The 8 simple directional derivative estimates along with the appropriate unit vectors. (c) An example of distorted neighborhood along with its directional derivative estimates.

following, we present two formulations for the proposed filters. The first formulation has been proposed in [26], it will be referred to as "Generalized Sobel Filters (GSF)" and is discussed in Appendix A.5.1. The second formulation has been presented in [22] and extends the Generalized Sobel Filters with the introduction of a normalization factor. It is discussed in Appendix A.5.2 and will be referred to as "Distortion Adpative Sobel Filters (DASF)".

A.5.1 Generalized Sobel Filters (GSF)

The Sobel operator was originally proposed by Irwin Sobel in 1968 to estimate the gradient of a digital image [173, 179, 180]. Sobel proposed to estimate the gradient of the image at a given point performing the vector summation of the simple central gradient estimates along the 4 main directions in a 3×3 neighborhood (see Figure A.34(a)). According to Sobel, each of the 4 simple central gradient estimates can be expressed as a vector sum of a pair of orthogonal vectors. Each vector is a directional derivative estimate multiplied by a unit vector specifying the derivative's direction. The value of the directional derivative estimate for a pair of antipodal pixels in a 3×3 neighborhood is defined as:

$$\frac{\text{density difference}}{\text{distance to neighbour}}. \tag{A.50}$$

The direction associated to the derivative estimate is given by the unit vector to the appropriate neighbour. Figure A.34(b) shows a schema of the directional derivative estimates. The reader is referred to [180] for a review of Sobel's rationale. The gradient estimation defined by Sobel can be formulated as the average of the eight oriented derivative vectors as follows:

$$\nabla I(x,y) = \frac{1}{8} \sum_{(s,t)\in S} \left(\frac{I^{s,t}_{x,y} - I^{-s,-t}_{x,y}}{\delta^{s,t}_{x,y}} \cdot \frac{(s,t)}{\sqrt{s^2+t^2}} \right) \tag{A.51}$$

where I is the considered image, $S = \{(s,t) : -1 \leq s,t \leq 1\}$, $I^{s,t}_{x,y} = I(x+s, y+t)$, $\delta^{s,t}_{x,y} = \delta((x+s, y+t), (x-s, y-t))$, (s,t) denotes the vector of components s,t and magnitude $\sqrt{s^2+t^2}$ and δ is a given metric (e.g., Euclidean).

We start from Equation (A.51) to build the distortion adaptive Sobel filters. Considering the symmetry of Equation (A.51) with respect to the sign of s and t, it is convenient to define:

$$\{S_1, S_2\} \text{ partition of } S \setminus \{(0,0)\} \text{ s.t. } p \in S_1 \Leftrightarrow -p \in S_2. \tag{A.52}$$

Given the definition in Equation (A.52), Equation (A.51) can be rewritten in the following form:

$$\begin{aligned} \nabla I(x,y) = &\frac{1}{8} \sum_{(s,t)\in S_1} \left[\frac{I^{s,t}_{x,y} - I^{-s,-t}_{x,y}}{\delta^{s,t}_{x,y}} \cdot \frac{(s,t)}{\sqrt{s^2+t^2}} \right] + \\ &+ \frac{1}{8} \sum_{(s,t)\in S_2} \left[\frac{I^{s,t}_{x,y} - I^{-s,-t}_{x,y}}{\delta^{s,t}_{x,y}} \cdot \frac{(s,t)}{\sqrt{s^2+t^2}} \right]. \end{aligned} \tag{A.53}$$

Given the definition in Equation (A.52), it is possible to substitute the set S_2 with S_1 changing the signs of s and t in the second summation:

$$\begin{aligned} \nabla I(x,y) = &\frac{1}{8} \sum_{(s,t)\in S_1} \left[\frac{I^{s,t}_{x,y} - I^{-s,-t}_{x,y}}{\delta^{s,t}_{x,y}} \cdot \frac{(s,t)}{\sqrt{s^2+t^2}} \right] + \\ &+ \frac{1}{8} \sum_{(s,t)\in S_1} \left[\frac{I^{-s,-t}_{x,y} - I^{s,t}_{x,y}}{\delta^{-s,-t}_{x,y}} \cdot \frac{(-s,-t)}{\sqrt{s^2+t^2}} \right]. \end{aligned} \tag{A.54}$$

Grouping the terms related to I and considering that δ is a metric (and hence

$\delta((x_1, y_1), (x_2, y_2)) = \delta((x_2, y_2), (x_1, y_1)),\ \forall\ (x_1, y_1), (x_2, y_2) \in \Re^2)$, we obtain Equation (A.55):

$$\begin{aligned}\nabla I(x,y) =& \frac{1}{8}\sum_{(s,t)\in S_1}\left[\frac{I^{s,t}_{x,y} - I^{\text{-}s,\text{-}t}_{x,y}}{\delta^{s,t}_{x,y}}\left(\frac{(s,t)}{\sqrt{s^2+t^2}} - \frac{(-s,-t)}{\sqrt{s^2+t^2}}\right)\right] \\ =& \frac{1}{8}\sum_{(s,t)\in S_1}\left[\frac{I^{s,t}_{x,y}}{\delta^{s,t}_{x,y}}\left(\frac{(s,t)}{\sqrt{s^2+t^2}} - \frac{(-s,-t)}{\sqrt{s^2+t^2}}\right)\right] + \\ &+ \frac{1}{8}\sum_{(s,t)\in S_1}\left[\frac{I^{\text{-}s,\text{-}t}_{x,y}}{\delta^{s,t}_{x,y}}\left(\frac{(-s,-t)}{\sqrt{s^2+t^2}} - \frac{(s,t)}{\sqrt{s^2+t^2}}\right)\right].\end{aligned} \tag{A.55}$$

Leveraging the definition in Equation (A.52), we obtain:

$$\begin{aligned}\nabla I(x,y) =& \frac{1}{8}\sum_{(s,t)\in S_1}\left[\frac{I^{s,t}_{x,y}}{\delta^{s,t}_{x,y}}\left(\frac{(s,t)}{\sqrt{s^2+t^2}} - \frac{(-s,-t)}{\sqrt{s^2+t^2}}\right)\right] + \\ &+ \frac{1}{8}\sum_{(s,t)\in S_2}\left[\frac{I^{s,t}_{x,y}}{\delta^{\text{-}s,\text{-}t}_{x,y}}\left(\frac{(s,t)}{\sqrt{s^2+t^2}} - \frac{(-s,-t)}{\sqrt{s^2+t^2}}\right)\right].\end{aligned} \tag{A.56}$$

Considering again the symmetric property of δ and considering the definition in Equation (A.52), the Equation (A.56) can be finally written as:

$$\nabla I(x,y) = \frac{1}{8}\sum_{(s,t)\in S}\left[\frac{I^{s,t}_{x,y}}{\delta^{s,t}_{x,y}}\left(\frac{(s,t)}{\sqrt{s^2+t^2}} + \frac{(s,t)}{\sqrt{s^2+t^2}}\right)\right]. \tag{A.57}$$

Let be $h_1(x,y,s,t)$ and $h_2(x,y,s,t)$ defined as follows:

$$h_1(x,y,s,t) = \begin{cases} 0 & \text{if}(s,t) = (0,0) \\ \frac{1}{4}\cdot\frac{1}{\delta^{s,t}_{x,y}}\frac{s}{\sqrt{(s^2+t^2)}} & \text{otherwise} \end{cases} \tag{A.58}$$

$$h_2(x,y,s,t) = \begin{cases} 0 & \text{if}(s,t) = (0,0) \\ \frac{1}{4}\cdot\frac{1}{\delta^{s,t}_{x,y}}\frac{t}{\sqrt{(s^2+t^2)}} & \text{otherwise.} \end{cases} \tag{A.59}$$

Considering the equations above, the gradient estimation can be expressed as follows:

$$\nabla_x I(x,y) = \sum_{(s,t)\in S} I(x+s, y+t) \cdot h_1(x,y,s,t) \tag{A.60}$$

$$\nabla_y I(x,y) = \sum_{(s,t)\in S} I(x+s, y+t) \cdot h_2(x,y,s,t). \tag{A.61}$$

Note that if the image has a planar geometry, the neighborhood is not-distorted, similarly to what is shown in Figure A.34(b) and δ is naturally chosen as the Euclidean distance. In this case, the filters defined in Equation (A.58) and Equation (A.59) are independent from the location of point (x, y) on which they are applied and Equation (A.60) and Equation (A.61) are equivalent to standard convolutions with Sobel filters (up to a factor of 16 in the estimation of the gradient magnitudes as discussed below). If the neighborhood is not Euclidean (i.e., it is distorted) as in the case of wide angle images (see Figure A.34(c)), the δ function should be chosen according to the underlying geometrical model. In particular, if the distortion function $\Psi : \Re^2 \rightarrow \Re^2$ which maps the not-distorted point of coordinates (u, v) to the distorted point (x, y) is known and invertible, the easiest way to choose a geometrically correct distance metric is to compose the Euclidean distance d with the inverse distortion function Ψ^{-1} as follows:

$$\delta((x_1,y_1),(x_2,y_2)) = d(\Psi^{-1}(x_1,y_1), \Psi^{-1}(x_2,y_2)), \forall (x_1,y_1) \in \Re^2, (x_2,y_2) \in \Re^2. \tag{A.62}$$

Since Ψ^{-1} is an inverse function, it is also bijective and hence the function δ defined above is a metric. The exploitation of Equation (A.62) corresponds to the projection of the coordinates of the neighborhood points into the Euclidean space prior to computing the distances in the classic way. In this general case, the terms related to δ in Equation (A.58) and Equation (A.59) depend on the considered point (x, y) and hence the kernels are adaptive. Figure A.36 shows some graphical examples of the proposed distortion adaptive Sobel filters by considering the position in the image to which they are applied. Figure A.35 shows some sample GSF computed in the center, top left and top right corners. As it can be noted, the proposed formulation yields kernels which adapt their shape in order to compensate for the radial distortion intrinsic to the different locations of the image. In this case, the computation defined in Equation (A.60) and Equation (A.61) is not strictly a convolution since the signals defined in Equation (A.58) and Equation (A.59) also depend on variables s and t.

	upper left corner			center			upper right corner		
x	-0.3326	0	0.8951	-1.0000	0	1.0000	-0.8951	0	0.3326
	-0.7843	0	0.7843	-2.0000	0	2.0000	-0.7843	0	0.7843
	-0.8951	0	0.3326	-1.0000	0	1.0000	-0.3326	0	0.8951
y	-0.3326	-1.0279	-0.8951	-1.0000	-2.0000	-1.0000	-0.8951	-1.0279	-0.3326
	0	0	0	0	0	0	0	0	0
	0.8951	1.0279	0.3326	1.0000	2.0000	1.0000	0.3326	1.0279	0.8951

Figure A.35: Some examples of Generalized Sobel Filters for a fisheye camera. The filters are computed on the center of the image, top left and top right corners.

We refer to this computation as "adaptive convolution" of image I with the adaptive kernels h_1 and h_2 as it is intended in [136]. It should be noted that the Sobel filters overestimate the magnitude of the gradients by a scaling factor of 16 [180], thus a totally compatible formulation of the proposed filters can be achieved multiplying Equation (A.58) and Equation (A.59) by a factor of 16. As discussed above, the just derived filters will be referred to as Generalized Sobel Filters (GSF).

A.5.2 Distortion Adaptive Sobel Filters (DASF)

Given the spatially non-uniform sampling of the incoming light operated by wide angle sensors, wide angle images are intrinsically multi-scale, which implies that the corresponding distance between neighboring pixels in real world coordinates increases with the distance from the center of the image. Considering that Generalized Sobel Filters have been are derived using the corresponding distances between neighboring pixels by means of Equation (A.62), filters computed in the peripheral areas of the image will have smaller coefficients in absolute values. This observation is illustrated in Figure A.36 where filters computed near the borders of the hemispherical image tend to have a lower magnitude in average. As a result, the magnitudes

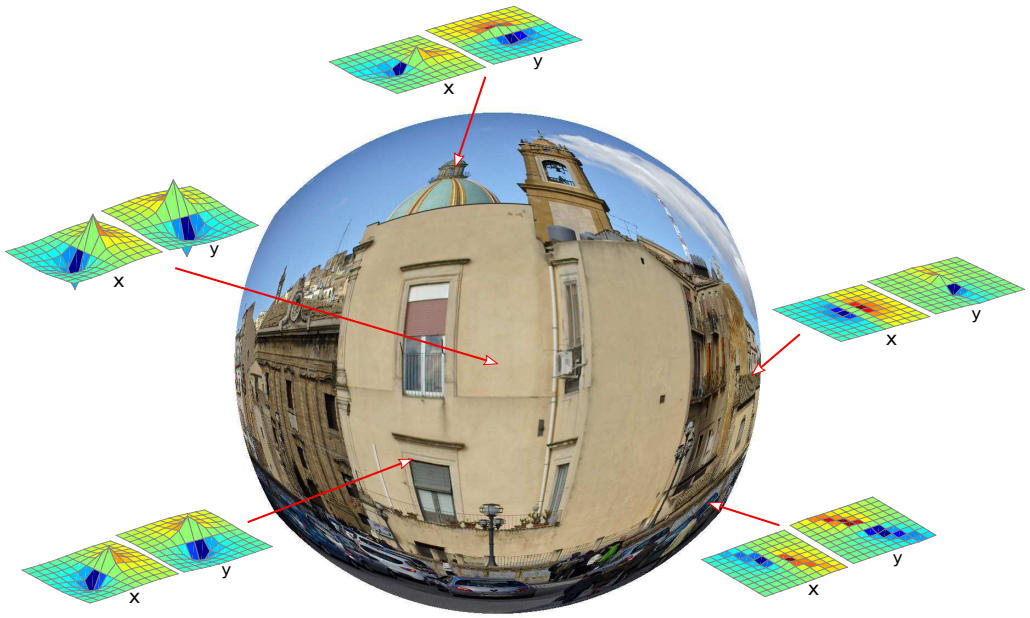
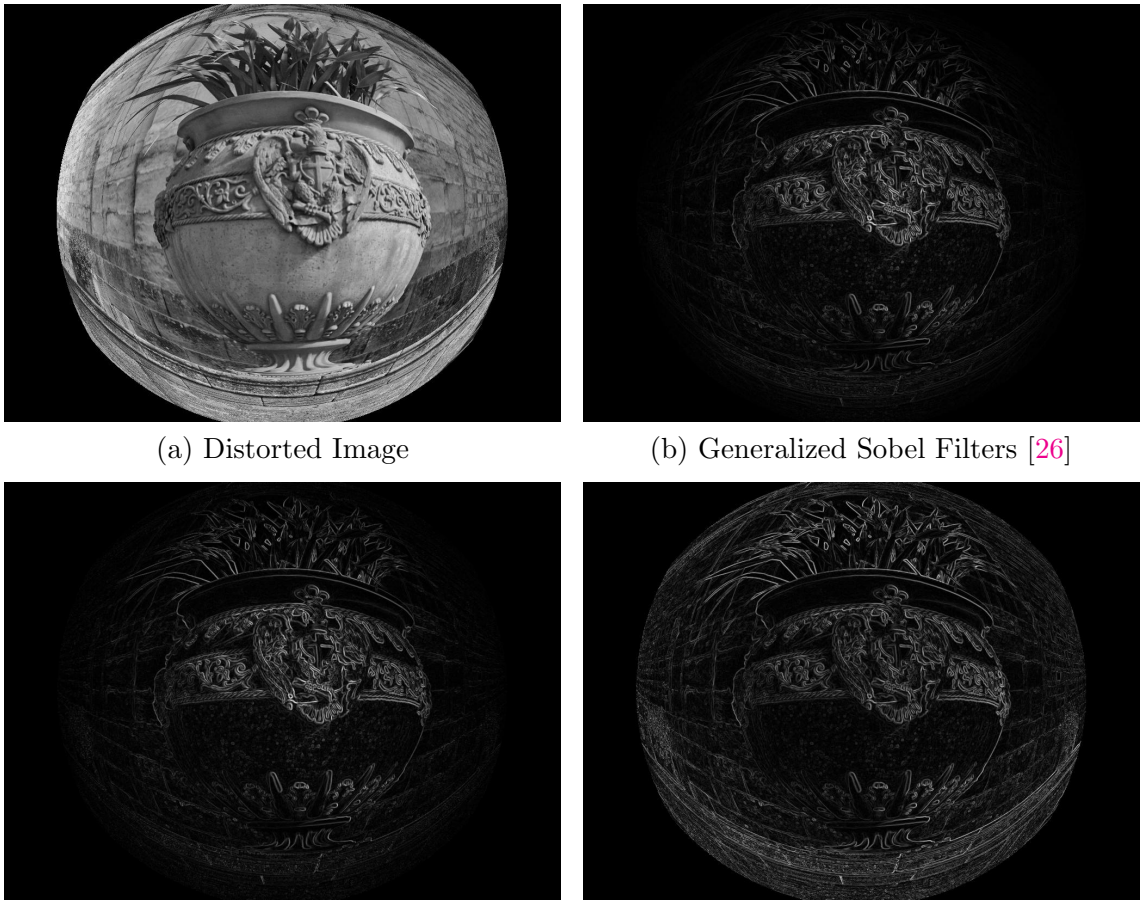


Figure A.36: Some graphical examples of the proposed kernels related to specific positions in the image on which they are computed. The shape of the kernels adapts to compensate the distortion characterizing a particular image location.

of the estimated gradients decay in the peripheral areas of the image as it is depicted in Figure A.37(a,b,c). It should be noted that such problem is not specific to our method, but is common to other direct gradient estimation techniques like for instance, the Gradient Correction Jacobian (GCJ) method proposed in [136, 160]. To overcome this limit and produce gradients with uniform magnitudes, we propose to locally normalize the derived GSF filters by the sum of the distances (computed according to the metric defined in Equation (A.62)) between the antipodal pairs in the 3×3 neighborhood:

$$\overline{h}_1(x,y,s,t) = \begin{cases} 0 & \text{if}(s,t) = (0,0) \\ \frac{1}{4} \cdot \frac{1}{\Delta_{x,y}} \cdot \frac{1}{\delta_{x,y}^{s,t}} \frac{s}{\sqrt{(s^2+t^2)}} & \text{otherwise} \end{cases} \tag{A.63}$$

$$\overline{h}_2(x,y,s,t) = \begin{cases} 0 & \text{if}(s,t) = (0,0) \\ \frac{1}{4} \cdot \frac{1}{\Delta_{x,y}} \cdot \frac{1}{\delta_{x,y}^{s,t}} \frac{t}{\sqrt{(s^2+t^2)}} & \text{otherwise} \end{cases} \tag{A.64}$$



(a) Distorted Image (b) Generalized Sobel Filters [26]

(c) JGC [136, 160] (d) Proposed DASF

Figure A.37: An example distorted image (a) along with the magnitudes of gradients estimated using (b) the Generalized Sobel Filters (GSF) defined in Equation (A.58)-(A.59), (c) the Jacobian Gradient Correction (JCG) method proposed in [136, 160], and (d) the proposed Distortion Adaptive Sobel Filters (DASF).

where the normalization factor is defined as follows:

$$\Delta_{x,y} = \sum_{(s,t)\in S} \frac{1}{\delta_{x,y}^{s,t}}. \tag{A.65}$$

Figure A.37(d) shows the magnitudes of the gradients of the image depicted in Figure A.37(a) as computed using the filters defined in Equation (A.63) and Equation (A.64). It should be noted that computing the gradients using the proposed locally normalized filters, allows to recover a huge quantity of details in the peripheral areas of the wide angle image. As discussed in the previous sections, this normalized

formulation of the proposed filters will be referred to as Distortion Adaptive Sobel Filters (DASF).

A.5.3 Experimental Evaluation of the Proposed Filters

We evaluate the performances of the proposed filters on the high resolution images of the DASF-HIRES-100 dataset. To assess the performances of the algorithms with respect to fisheye distortion, we artificially add different degrees of radial distortion to the reference rectilinear images employing the division model and following the methodologies described in Appendix A.2.3. This experimental approach allows to control the exact amount of distortion characterizing the target images. Moreover, the source (not-distorted) images are used to compute the reference gradients which serve as a ground truth for the evaluations. Specifically, the rectilinear input images of resolution 5204×3472 pixels are mapped to distorted images of resolution 1024×768 pixels considering 21 different degrees of radial distortion ranging from $d = 0.1$ to $d = 0.5$. This leads to the creation of 100 image series of 22 images comprising one reference high resolution non-distorted image and 21 low resolution distorted ones.

We perform two experiments to assess the performances of the proposed method. The former experiment aims at measuring the error committed by the compared methods in estimating the image gradients, independently from any specific application. The latter experiment aims at assessing the impact of the proposed method on real-world applications. In particular, considering the importance of local feature description and matching [136, 7, 25], we assess the impact of the considered method when the computed gradients are used to compute and match densely sampled SIFT features.

Evaluation of Gradient Estimation Error

The image gradients are usually exploited separating the magnitudes from the orientations. The orientations carry important information about the distribution of edges in the scene, while the magnitudes give insights on the importance of each orientation. For this reason, many algorithms rely on the weighted histograms of gradient orientations [7, 138]. We define an error measure by considering the average distance between the local populations (i.e., weighted histograms) of the gradient

orientations in the reference image I and in its distorted counterpart $\hat{I}$. Let $\mathcal{S}$ denote the Sobel operator, $\mathcal{G}$ the gradient estimator under analysis and let the distorted image $\hat{I}$ be divided into n non overlapping regular tiles of size $k \times k$ covering the entire surface: $\{\hat{T}_i\}_{1 \leq i \leq n}$. For each tile $\hat{T}_i$ in the distorted image $\hat{I}$, we consider the related not-distorted tile T_i in the reference image I, which contains the not-distorted counterparts of all the points in $\hat{T}_i$. The error related to the gradient estimator $\mathcal{G}$ given the image pair $(\hat{I}, I)$ is defined as:

$$\epsilon(\mathcal{G}, \hat{I}, I) = \frac{1}{n} \sum_{i=1}^{n} \rho(\mathcal{H}(\mathcal{G}\hat{T}_i), \mathcal{H}(\mathcal{S}T_i)) \tag{A.66}$$

where $\mathcal{H}(\mathcal{G}\hat{T}_i)$ and $\mathcal{H}(\mathcal{S}T_i)$ denote the weighted histograms of the estimated and reference gradient orientations (i.e. Sobel on the reference image) of tiles $\hat{T}_i$ and T_i, and ρ is the metric based on the Bhattacharyya coefficient as defined in [4]:

$$\rho(H_1, H_2) = \sqrt{1 - \sum_{u=1}^{m} \sqrt{H_1^u \cdot H_2^u}}. \tag{A.67}$$

In Equation (A.67), m is the number of bins of histograms H_1 and H_2 and H^u denotes the $u - th$ component of H. Figure A.38 illustrates the computation of the Bhattacharyya distance for two corresponding non-overlapping tiles. In our experiments, the following parameters have been used: 1) each image is divided into tiles of size 24×24 pixels and 2) the histograms have 18 bins evenly spacing the interval $[-180°, 180°]$.

Figure A.39 shows the mean error committed by the considered methods with respect to different amounts of radial distortion. Each curve is obtained averaging the error scores related to the 100 images in the dataset and computed using Equation (A.66). The legend of Figure A.39 reports in parenthesis the average value of each curve, which should reflect the average performances of the methods with respect to all the considered amounts of distortion. The rectification method allows to achieve good results for low distortion rates (where the lost information can still be "guessed" by the rectification process), whereas the error gets higher as the distortion rate increases. The proposed filters perform better than the competitors for distortion rates over the 35%. Table A.3 summarizes the average errors

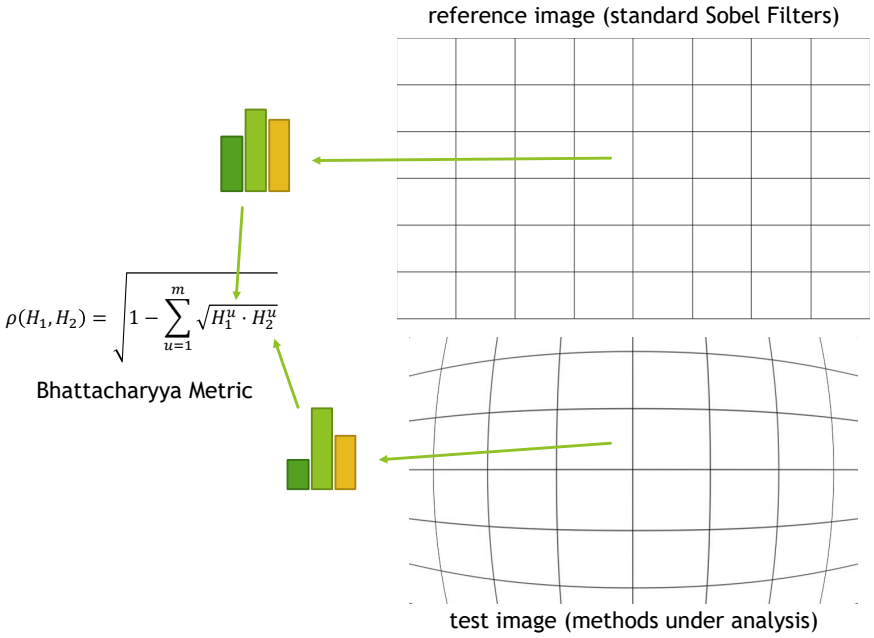


Figure A.38: Computation of the Bhattacharyya distance for two corresponding non-overlapping regular tiles.

related to the subset of images characterized by a specific scene-based tag. The reported results suggest that the proposed filters, the GSF and the GCJ methods offer significant improvements over the distorted gradients. In particular, the proposed method is the best performing when all scene categories are considered (top row of Table A.3) and it is always among the two best performing methods for each of the scene categories. As it appears clear from Figure A.39 and Table A.3, the DASF method closely matches the performances of the GSF method in this experiment. This is due to the fact that, considering regular tiles as small as 24×24 pixels, the effect of the local normalization introduced in Appendix A.5.2 is negligible with respect to the normalization operated by the computation of the weighted histogram of gradient orientations.

Evaluation of Impact on SIFT Matching Ability

The aim of this second experiment is to assess the performances of the considered methods on the task of local feature description and matching. To this aim, we consider the popular gradient-based SIFT descriptor [7], computed using the estimated

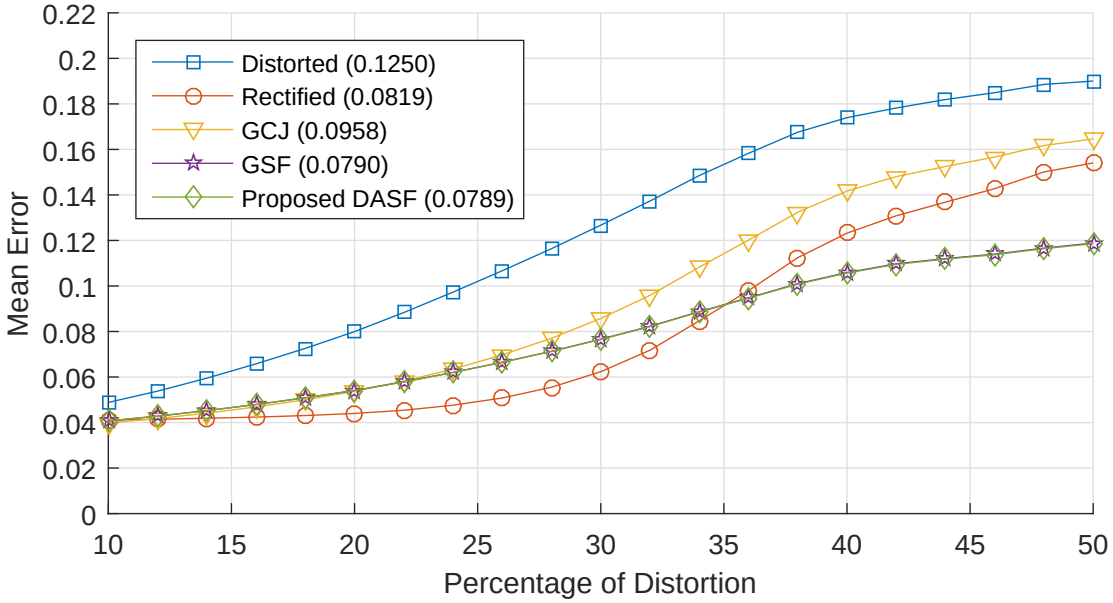
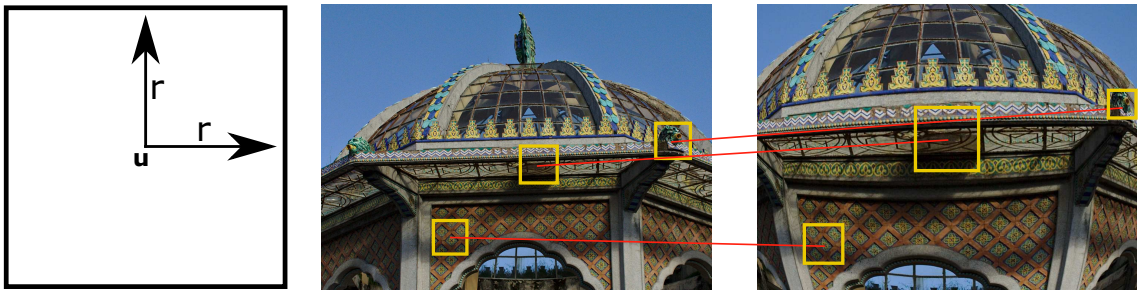


Figure A.39: Mean error curves for different gradient estimators on the GSF-HIRES-100 dataset for varying percentages of distortion. The average value of each curve is reported in parenthesis in the legend.

Scene	Distorted	Rectified	GCJ	GSF	Proposed
All	0.1250	0.0819	0.0958	0.0790	**0.0789**
Indoor	0.1172	0.0885	0.1049	**0.0826**	**0.0826**
Outdoor	0.1262	0.0809	0.0945	0.0785	**0.0784**
Natural	0.1129	0.0816	0.0980	**0.0736**	**0.0736**
Handmade	0.1280	0.0819	0.0952	0.0801	**0.0800**
Urban	0.1358	**0.0796**	0.0910	0.0816	0.0813
Car	0.1366	**0.0807**	0.0919	0.0827	0.0824
Pedestrian	0.1417	**0.0748**	0.0846	0.0807	0.0805
Street	0.1357	**0.0807**	0.0921	0.0823	0.0821

Table A.3: Average errors for different methods and scene types. In each row, the two smallest values are underlined, while the minimum is reported in **bold** letters.



(a) (b)

Figure A.40: (a) A schema of a support region and (b) an example of projecting support regions from the undistorted space to the distorted one.

gradients. In order to cope with radial distortion, we consider DAD-SIFT, a distortion adaptive variant of the popular gradient-based SIFT descriptor [7] which we proposed in [25]. As it will be discussed in Appendix A.6, where the computation of the DAD-SIFT descriptor is detailed, Distortion Adaptive Descriptors (DAD) provide a way to compute regular gradient-based descriptors directly on the distorted images accounting for radial distortion. In the following, we summarize the evaluation protocol considered for the experiments.

Given a reference-target image pair $(I, \hat{I})$, square support regions are considered at multiple scales on the reference image I. In particular, we sample support regions of radii: 32, 64, 128 and 256 pixels at a regular step equal to 50 pixels. We consider a support region as an entity $\mathcal{R}(\mathbf{u}, r)$ composed of two elements: a center $\mathbf{u}$ and a radius r (see Figure A.40(a)). Each support region $\mathcal{R}(\mathbf{u}, r)$ is mapped to a corresponding region $\hat{\mathcal{R}}(\mathbf{x}, \hat{r})$ in the distorted space using Equation (A.20) to map the not-distorted point $\mathbf{u}$ to its distorted counterpart $\mathbf{x}$. The radius of the distorted support region $\hat{r}$ is computed using Equation (A.22):

$$\hat{r} = g(r) = \frac{2r}{1 + \sqrt{1 - 4 \cdot \xi r^2}}. \tag{A.22}$$

Figure A.40(b) shows an example of such projection. All the projected regions not entirely contained in the distorted image $\hat{I}$ or with projected radii under 16 pixels are discarded along with their not-distorted counterparts.

Standard SIFT descriptors $\mathcal{D}$ are computed on the reference support regions

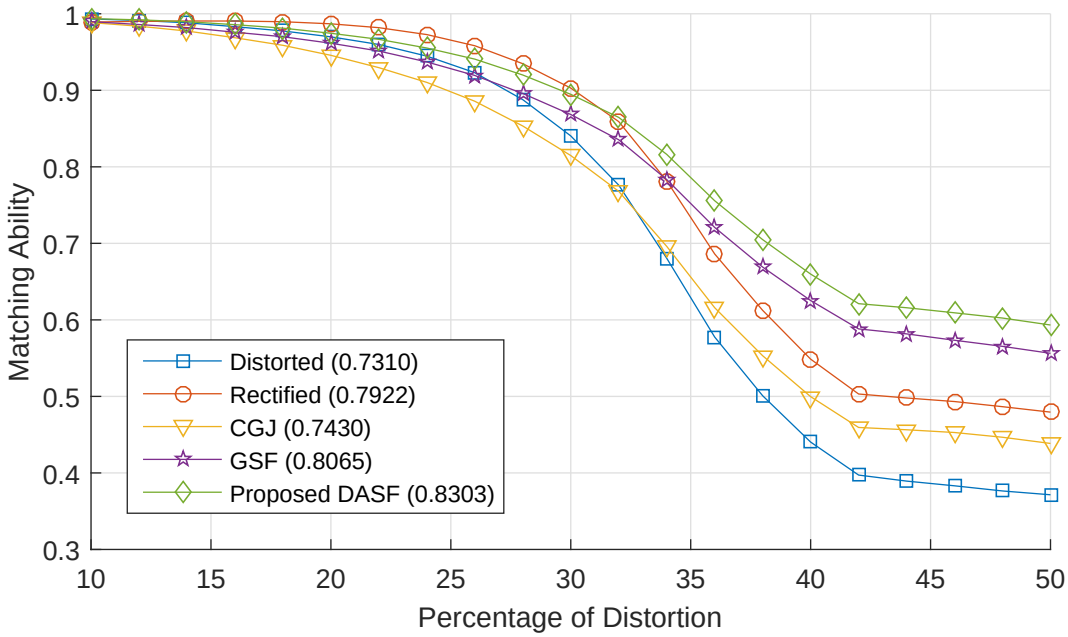


Figure A.41: Results related to Experiment 2. The matching ability for different gradient estimators on the considered dataset at varying of the percentage of distortion.

$\mathcal{R}$ using the reference gradient estimated with the Sobel operator. DAD-SIFT descriptors $\hat{\mathcal{D}}$ are computed from the projected support regions $\hat{\mathcal{R}}$ using the generic estimator $\mathcal{G}$ as detailed later in Appendix A.6. Matchings between the reference $\mathcal{D}$ and target $\hat{\mathcal{D}}$ descriptors are computed using the nearest neighbor criterion, i.e., descriptor $d \in \mathcal{D}$ is matched to its nearest neighbor $\hat{d} \in \hat{\mathcal{D}}$. Given the known correspondences between the reference and target descriptors, the matching ability score is measured as follows:

$$matching\ ability\ score = \frac{\#correct\ matches}{\#matches} \tag{A.68}$$

where parameter ξ allows to control the amount of distortion in the image.

Figure A.41 shows the matching ability achieved by the considered methods with respect to different amounts of radial distortion. Each curve is obtained averaging the matching ability scores related to the 100 images in the dataset. As in Figure A.39, the legend of Figure A.41 reports the average value of each curve which should reflect the average performances of the methods with respect to all the considered amounts of distortion. The proposed method retains the highest matching

Scene	Distorted	Rectified	GCJ	GSF	Proposed
All	0.7310	0.7922	0.7430	0.8065	**0.8303**
Indoor	0.6394	**0.7434**	0.6685	0.7091	0.7366
Outdoor	0.7447	0.7995	0.7541	0.8210	**0.8444**
Natural	0.6958	0.7594	0.6885	0.7600	**0.7911**
Handmade	0.7472	0.8027	0.7601	0.8236	**0.8452**
Urban	0.7776	0.8260	0.8010	0.8644	**0.8813**
Car	0.7779	0.8248	0.7990	0.8601	**0.8768**
Pedestrian	0.7890	0.8403	0.8252	0.8834	**0.8993**
Street	0.7780	0.8235	0.7974	0.8591	**0.8758**

Table A.4: Results related to Experiment 2. Average matching ability scores for different methods and scene types. In each row, the two largest values are underlined, while the maximum is reported in **bold** letters.

ability for amounts of distortion over the 30%, while it performs comparably to the rectification methods for distortion rates below the 30%. The evident improvement of DASF over GSF is due to the introduced normalization mechanism. Normalization is particularly effective in the case of large support region. When support regions are large indeed, the magnitude normalization mechanism of the SIFT descriptor cannot cope with non-uniform gradient magnitudes. This observation also explains the little difference between DASF and GSF in the previous experiment, where support regions where small (24×24 pixels). Similarly to Table A.3, Table A.4 reports the average matching abilities related to images characterized by specific image tags. The proposed method is always the best performing (highest score), except in the case of indoor images, where, probably due to the regularity of straight edges, the standard baseline employing rectification performs slightly better.

A.5.4 Discussion

Distortion Adaptive Sobel filters can be used to correctly estimate the gradient of distorted images. The proposed filters are independent from the adopted distortion model and only require the distortion function Ψ to be known and invertible. We have assessed the performances defining an evaluation protocol which measures the

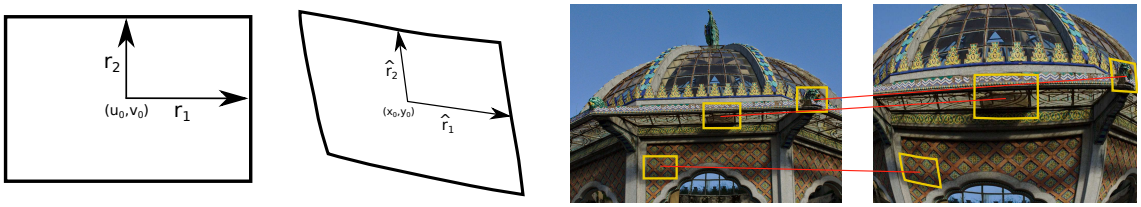
error between estimated and reference gradients and allows to compare the performances on the task of local feature matching. The experiments show that our method outperforms the competitors.

A.6 Distortion Adaptive Descriptors

The presented techniques allow to compute more geometrically-correct gradients directly from distorted images. However, radial distortion can severely affect the way local feature descriptors are computed due to the non-Euclidean representation of the scene as discussed in [148]. In this Section, we study how gradient-based descriptors such as SIFT [7] and Histogram of Oriented Gradients (HOG) [138] can be modified in order to be computed directly in the distorted domain. We propose the Distortion Adaptive Descriptors (DAD), a new paradigm for computing local descriptors directly on the distorted images. The proposed adaptive descriptors assume that the camera is calibrated and hence the distortion function Ψ is known and invertible. We combine the DAD paradigm with existing methods for the correct estimation of the gradient of distorted images in order to derive distortion adaptive variants of the SIFT [7] and Histogram of Oriented Gradients (HOG) [138] descriptors. The adaptation of such descriptors to the distorted domain virtually enables a number of applications in which they have proven to be successful, such as object and people detection [7, 138], video stabilization [181], object class recognition [182] and panorama stitching [183]. Experiments show that the DAD variants significantly outperform the regular SIFT and HOG descriptors when they are applied directly in the distorted domain. Moreover, we show that there is still space for improving direct gradient estimation techniques.

A.6.1 Formulation of Distortion Adaptive Descriptors

In this Section, we introduce the Distortion Adaptive Descriptors (DAD). Rather than a new set of descriptors, the DAD constitute a paradigm for correctly computing existing local descriptors directly on the distorted images. For sake of generality we consider a generic descriptor $D(\mathcal{N}, \mathcal{M}(I, \mathcal{N}))$ computed on a rectangular neighborhood $\mathcal{N}$ using some measurements $\mathcal{M} = \mathcal{M}(I, \mathcal{N})$ performed in the locations of the input image I specified by the neighborhood $\mathcal{N}$. The measurements can be of



(a) (b) (c)

Figure A.42: (a) A rectilinear neighborhood and (b) its distorted counterpart. (c) Examples of rectilinear neighborhoods along with their distorted counterparts.

any kind and are related to the feature extraction process required by the specific descriptor. In the SIFT descriptor, for instance, the measurements $\mathcal{M}$ are the image gradients estimated at the relevant locations. The rectangular neighborhood centered at point (u_0, v_0) with radii r_1 and r_2 is naturally defined as the set of points:

$$\mathcal{N}(u_0, v_0, r_1, r_2) = \{(u, v) : |u - u_0| \leq r_1 \wedge |v - v_0| \leq r_2\} \tag{A.69}$$

Figure A.42(a) shows an example of rectangular local neighborhood. When the descriptor has to be computed on a distorted image, the shape of the neighborhood $\mathcal{N}$ depends on its position in the image. Some examples of such assertion are illustrated in Figure A.42(c). The rectilinear neighborhood in Equation (A.69) is easily mapped to its distorted counterpart centered at point (x_0, y_0) with radii $\hat{r}_1$ and $\hat{r}_2$ using the inverse distortion function Ψ and the radial distortion function g:

$$\begin{aligned}\hat{\mathcal{N}}(x_0, y_0, \hat{r}_1, \hat{r}_2) = \{(x, y) \quad : \quad & |\Psi^{-1}(x) - \Psi^{-1}(x_0)| \leq g^{-1}_{x_0,y_0}(\hat{r}_1) \wedge \\ & |\Psi^{-1}(y) - \Psi^{-1}(y_0)| \leq g^{-1}_{x_0,y_0}(\hat{r}_2)\}\end{aligned} \tag{A.70}$$

where $(x_0, y_0) = \Psi(u_0, v_0)$ and $\hat{r}_1$ and $\hat{r}_2$ are obtained from r_1 and r_2 using Equation (A.22). Figure A.42(b) shows an example of distorted neighborhood. Let be $\hat{\mathcal{M}}(\hat{I}, \hat{\mathcal{N}})$ the geometrically correct measurement performed in the locations of the distorted image $\hat{I}$ specified by the distorted neighborhood $\hat{\mathcal{N}}$. The Distortion Adaptive Descriptor related to $\mathcal{D}$ is hence defined as:

$$\hat{\mathcal{D}} = D(\Psi^{-1}(\hat{\mathcal{N}}), \hat{\mathcal{M}}(\hat{I}, \hat{\mathcal{N}})) \tag{A.71}$$

where $\Psi^{-1}(\hat{\mathcal{N}}) = \{\Psi^{-1}(x, y) : (x, y) \in \hat{\mathcal{N}}\}$.

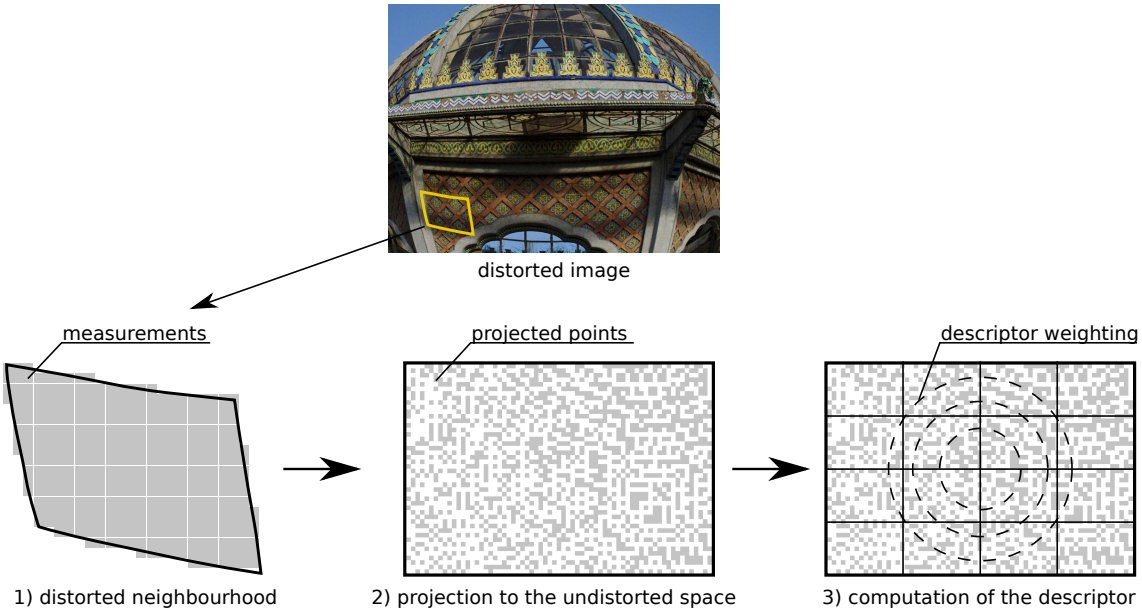


Figure A.43: A scheme of the computation of the Distortion Adaptive Descriptors. 1) The distorted neighborhood is extracted from the input image. 2) The measurements are projected to the rectilinear space. As it can be noted, this yields to samples of non uniform density. 3) The regular descriptor is computed accounting for the correct arrangement of the measurements in the rectilinear space.

The computation defined above is carried in three key steps: 1) given a point (x_0, y_0) in the distorted space and two radii $\hat{r}_1$, $\hat{r}_2$, the distorted neighborhood $\hat{\mathcal{N}}(x_0, y_0, \hat{r}_1, \hat{r}_2)$ is considered; 2) all the coordinates of the points in $\hat{\mathcal{N}}$ are projected back to the rectilinear space $(\Psi^{-1}(\hat{\mathcal{N}}))$; 3) the regular descriptor is computed using the geometrically correct measurements $\hat{\mathcal{M}}(\hat{I}, \hat{\mathcal{N}})$ and the projected coordinates $\Psi^{-1}(\hat{\mathcal{N}})$. It should be noted that step 2) is important since it allows the descriptor to weigh the measurements according to their position in the undistorted space. Specifically, the projection leads to samples of non-uniform density which are correctly dislocated in the undistorted circular neighborhood. The new locations for the considered measurements ensure a correct isotropic spatial weighting. Figure A.43 shows a scheme of the computation of the Distortion Adaptive Descriptors.

A.6.2 Experimental Evaluation of Distortion Adaptive Descriptors

We argue that a combination of gradient estimation techniques, such as the one introduced in Appendix A.5, and the DAD scheme proposed in Appendix A.6.1 can improve the matching ability of gradient based local descriptors on distorted images. What we want to evaluate is the invariance of the descriptors with respect to radial distortion, i. e., the ability to produce similar descriptors for two image regions representing the same physical area of the scene despite they are affected by different amounts of distortion. An ideal descriptor, for instance, would give identical results when computed on the matching neighborhoods shown in Figure A.42(c). In the following we discuss the experimental settings including the images used for the evaluations, the considered descriptors and the evaluation pipeline. Experiments are performed on the DASF-HIRES-100 dataset introduced in Appendix A.3.2.

We apply the DAD scheme to the SIFT and HOG descriptors using different gradient estimation techniques to obtain the measurements. To assess the improvement due to the DAD scheme independently from the employed gradient estimation technique, we also consider an ideal estimator by wrapping the ground truth gradients to the distorted locations. Moreover, we consider the standard SIFT and HOG descriptors computed directly in the distorted domain (without adaptation) combined with the different gradient estimation techniques. Hence we derive the 18 descriptors summarized in Table A.5. The SIFT-based descriptors are computed using the implementation provided by the VLFeat library [96], which produces standard 128-dimensional descriptors. For the HOG-based descriptors we consider the variant of HOG proposed in [184] as implemented by the VLFeat library [96]. Moreover, in our settings, the HOG-based descriptors are computed dividing the support region into 4×4 cells and the gradients are computed using 3×3 filters (in place of the non-smoothing $[-1\ 0\ 1]$ and $[-1\ 0\ 1]^T$ filters originally proposed by the authors [138]) in order to allow the gradient estimation techniques to compensate for the distortion exploiting neighborhood information. This configuration returns a 496-dimensional HOG descriptor for input support region of any size.

Acronym	**Description**
SIFT_{DIST} HOG_{DIST}	Regular SIFT/HOG descriptor computed on the distorted images using the distorted gradients as measurements.
SIFT_{RECT} HOG_{RECT}	Regular SIFT/HOG descriptor computed on the rectified images using the Sobel filters to estimate the gradients.
SIFT_{GCJ} HOG_{GCJ}	Regular SIFT/HOG descriptor computed on the distorted images using the GCJ gradients as measurements.
SIFT_{GSF} HOG_{GSF}	Regular SIFT/HOG descriptor computed on the distorted images using the GSF gradients as measurements.
SIFT_{IDEAL} HOG_{IDEAL}	Regular SIFT/HOG descriptor computed on the distorted images using the ground truth gradients as measurements.
DAD-SIFT_{DIST} DAD-HOG_{DIST}	SIFT/HOG descriptor computed with the DAD scheme on the distorted images using the distorted gradients as measurements.
DAD-SIFT_{GCJ} DAD-HOG_{GCJ}	SIFT/HOG descriptor computed with the DAD scheme on the distorted images using the GCJ gradients as measurements.
DAD-SIFT_{GSF} DAD-HOG_{GSF}	SIFT/HOG descriptor computed with the DAD scheme on the distorted images using the GSF gradients as measurements.
DAD-SIFT_{IDEAL} DAD-HOG_{IDEAL}	SIFT/HOG descriptor computed with the DAD scheme on distorted images using the ground truth gradients as measurements.

Table A.5: The descriptors considered in the experiments.

Experimental Results

Figure A.44 shows the 1-precision vs recall and the threshold vs F-Measure curves of the considered descriptors for different amounts of distortion. As it can be noted, all the DAD variants systematically outperform their non-adaptive counterparts independently from the employed gradient estimation technique. Moreover, using the GCJ [160, 136] and GSF techniques for the measurements allows to improve the performances of all the descriptors over the distorted gradients. Interestingly such techniques, combined with the DAD paradigm, allow to reach the performances obtained through the rectification process for low amounts of distortion (15%). In general, however, the rectification provides better results for higher distortion rates at the cost of the computational time required by the unwrap. The GCJ and GSF techniques have similar performances when used both in the HOG-based and SIFT-based descriptors. The descriptors based on the ground truth gradients always have the best performances, which confirms the power of the DAD paradigm. Moreover, the gap between the performances given by the ground truth gradients and the ones given by the considered gradient estimation techniques suggests that there is still space for improvement for such techniques.

Experimental Protocol

For our evaluations, we measure the matching ability of the considered descriptors when they are densely extracted from the test images. Dense descriptors are appropriate for our analysis since they allow us to draw conclusions which are independent from any interest point detector. Moreover dense descriptors have proven powerful in a variety of tasks [185, 186, 187]. Given the reference-distorted image pair $(I, \hat{I})$, we densely extract square support regions from the reference image at a regular step of 50 pixels. To account for multiscale features, different layers of overlapping support regions are extracted considering radii ranging from 32 to 256 pixels. In this context, a support region is an entity $\mathcal{S}(\mathbf{u}, r)$ made of two elements: a center $\mathbf{u}$ and a radius r. Each support region $\mathcal{S}$ is mapped to the corresponding support region $\hat{\mathcal{S}}$ in the distorted image using Equations (A.20) and (A.22): $\hat{\mathcal{S}}(\Psi(\mathbf{u}), g(r))$. All projected support regions which are not entirely contained in the distorted image $\hat{I}$ or which projected radius is under 16 pixels are discarded together with their undistorted counterparts. This settings lead to support regions of variable sizes

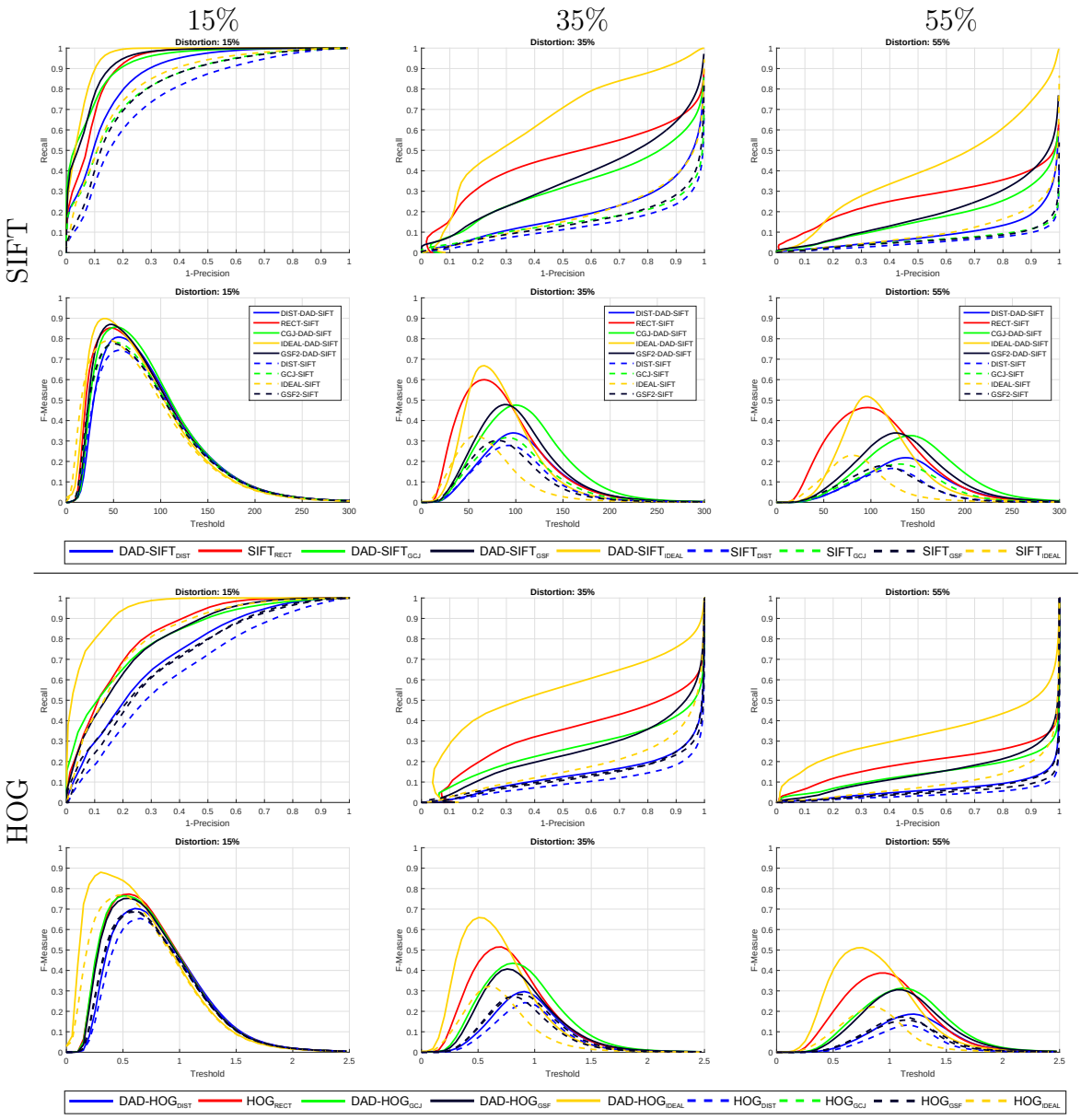


Figure A.44: The 1-precision vs recall curves (rows 1 and 3) and the threshold vs F-Measure curves (rows 2 and 4) for the SIFT-based and the HOG-based descriptors.

ranging from 32×32 pixels to 512×512 pixels which cover the entire FOV of the distorted images. The number of support regions per image ranges from 887 to 3881 depending on the distortion rate. We refer to the set of reference support regions as $S = \{\mathcal{S}_i\}$ and to the set of projected support regions as $\hat{S} = \{\hat{\mathcal{S}}_i\}$. The reference support regions S are used to compute the standard SIFT and HOG descriptors,

while the projected support regions $\hat{S}$ are used to compute the descriptors under evaluation. For instance, let be $\hat{\mathcal{D}}$ one of the SIFT-based descriptors in Table A.5, we define the set of reference descriptors as $D = SIFT(S)$ and the set of test descriptors $\hat{D} = \hat{\mathcal{D}}(\hat{S})$. Similar definitions hold for the HOG-based descriptors. To evaluate the matching ability of descriptor $\hat{\mathcal{D}}$, we follow an evaluation protocol similar to the one discussed in Appendix A.4.3. Specifically, we assume that two support regions $\mathcal{S}$ and $\hat{\mathcal{S}}$ match if the distance between their descriptors is below a threshold t. Each descriptor from the reference image is compared to each descriptor from the distorted one and the numbers of correct and false matches are counted. The threshold t is varied to obtain the curves. A matching between two descriptors is considered correct only if they have been computed on corresponding support regions. For each threshold t, the precision and recall values are computed using the formulas introduced in Appendix A.4.3 and reported in the following:

$$Precision = \frac{\#correct\ matches}{\#matches} \tag{A.47}$$

$$Recall = \frac{\#correct\ matches}{\#support\ regions}. \tag{A.48}$$

The 1-precision vs recall curves have a straightforward interpretation: a perfect descriptor would give a recall equal to 1 for any precision. In practice increasing the value of threshold t increases the recall and decreases the precision. The rate at which those values vary with respect to the threshold tells how an algorithm is able to produce distinctive descriptors, which are similar for corresponding regions. As reported in [165], this kind of evaluation is independent from the matching scheme one could adopt (e.g., nearest neighbor with or without rejection of ambiguous matches) and respect the distribution of the descriptors in the space. We also report the threshold vs F-Measure curves. The F-Measure values are computed as already discussed in Appendix A.4.3:

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{A.49}$$

where $\beta^2 = 0.3$ to weigh precision more than recall. The threshold vs F-Measures curves can be interpreted from a retrieval point of view: a good descriptor allows

to get a high number of positives with a small amount of noise. This situation is represented by a F-Measure curve with a high peak for a low threshold value.

A.6.3 Discussion

We have tackled the problem of improving the matching ability of gradient-based descriptors when they are directly computed on wide angle images. We have proposed the Distortion Adaptive Descriptors, a new paradigm for the correct computation of local descriptors in the distorted domain. Combining the DAD paradigm with existing techniques for estimating the gradient of distorted images, we show that it is possible to improve the matching ability of the SIFT and HOG descriptors. Even if the proposed descriptors can be computed directly on the wide angle images, the performances obtained through the rectification process are not matched yet. The results convey that improving the gradient estimation techniques would allow to significantly improve the performances of gradient-based local descriptors on wide angle images.

A.7 Findings

In this chapter, we have investigated direct feature extraction from distorted wide angle images. This investigation has been carried on in three stages. First, we studied the performances of affine covariant region detectors directly on distorted images. Second, we proposed methods for the direct estimation of the gradient of distorted images. Third, we formulated a paradigm for the direct computation of gradient-based local descriptors on distorted images. The main finding of this investigation are as follow:

- While the radial distortion introduced by wide angle sensors is not a linear transformation, locally it can be approximated as an affine mapping. Affine covariant region detectors can model such distortion reasonably up to a certain degree of radial distortion;
- Gradients of distorted images can be estimated by convolution filters derived generalizing Sobel filters to non-Euclidean manifolds;

- The computation of gradient-based local descriptors directly on distorted images can be improved combining the simple DAD paradigm with direct gradient estimation methods.

Appendix B

Other Publications

In the following, it is reported a list of works published during my Ph.D. but not directly related to this thesis.
International Journals:

- S. Battiato, G. M. Farinella, A. Furnari, G. Puglisi, A. Snijders, and J. Spiekstra. "An integrated system for vehicle tracking and classification". In: *Expert Systems with Applications* 42.21 (2015), pp. 7263–7275

International Conferences:

- F. Ragusa, V. Tomaselli, A. Furnari, S. Battiato, and G. M. Farinella. "Food vs Non-Food Classification". In: *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*. ACM. 2016, pp. 77–81

- A. Furnari, G. M. Farinella, and S. Battiato. "An Experimental Analysis of Saliency Detection with respect to Three Saliency Levels". In: *Workshop on Assistive Computer Vision and Robotics (ACVR) in conjunction with ECCV (2014)*. Vol. 8927. Lecture Notes in Computer Science. 2014, pp. 806–821

- S. Battiato, G. M. Farinella, A. Furnari, G. Puglisi, A. Snijders, and J. Spiekstra. "Vehicle tracking based on customized template matching". In: *International Conference on Computer Vision Theory and Applications*. Vol. 2. 2014, pp. 755–760

- B. Sebastiano, F. Giovanni Maria, F. Antonino, and P. Giovanni. "A Customized System for Vehicle Tracking and Classification". In: *European Conference on Mathematics for Industry (ECMI)*. 2014

- D. Scandura, S. Battiato, V. Bruno, F. Cannavo, G. M. Farinella, A. Furnari, M. Mattia, G. Pappalardo, G. Puglisi, and U. Weigmuller. "Image Processing Techniques to Estimate the Propagation of Ground Deformation at Mt. Etna (Italy) from ALOS PALSAR InSAR Data". In: *AGU Fall Meeting Abstracts*. Vol. 1. 2014

Bibliography

[1] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. 2001, pp. I–511.

[2] Z. Kalal, K. Mikolajczyk, and J. Matas. "Tracking-learning-detection". In: *IEEE transactions on pattern analysis and machine intelligence* 34.7 (2012), pp. 1409–1422.

[3] B. D. Lucas and T. Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision". In: *International Joint Conference on Artificial Intelligence*. Apr. 1981, pp. 674–679.

[4] D. Comaniciu, V. Ramesh, and P. Meer. "Kernel-based object tracking". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.5 (2003), pp. 564–577.

[5] G. R. Bradski. "Real time face and object tracking as a component of a perceptual user interface". In: *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*. 1998, pp. 214–219.

[6] M. Brown and D. G. Lowe. "Automatic panoramic image stitching using invariant features". In: *International journal of computer vision* 74.1 (2007), pp. 59–73.

[7] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[8] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[9] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. "Content-based Multimedia Information Retrieval: State of the Art and Challenges". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 2.1 (2006), pp. 1–19.

[10] T. Kanade and M. Hebert. "First-Person Vision". In: *Proceedings of the IEEE* 100.8 (2012), pp. 2442–2453.

[11] T. Starner, B. Schiele, and A. Pentland. "Visual contextual awareness in wearable computing". In: *International Symposium on Wearable Computing*. 1998, pp. 50–57.

[12] A. K. Dey, G. D. Abowd, and D. Salber. "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications". In: *Human-Compututer Interaction* 16.2 (2001), pp. 97–166.

[13] S. Greenberg. "Context As a Dynamic Construct". In: *Human-Compututer Interaction* 16.2 (2001).

[14] A. K. Dey and G. D. Abowd. "Towards a Better Understanding of Context and Context-Awareness". In: *Computing Systems* 40.3 (1999), pp. 304–307.

[15] F. De la Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran. "Guide to the carnegie mellon university multimodal activity (cmu-mmac) database". In: *Robotics Institute* (2008), p. 135.

[16] V. Alessandro, P. Maja, and B. Hervé. "Social signal processing: Survey of an emerging domain". In: *Image and Vision Computing* 27.12 (2009). Visual and multimodal analysis of human spontaneous behaviour: pp. 1743–1759.

[17] S. Alletto, G. Serra, S. Calderara, F. Solera, and R. Cucchiara. "From Ego to Nos-vision: Detecting Social Relationships in First-Person Views". In: *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014.

[18] A. Fathi, A. Farhadi, and J. M. Rehg. "Understanding egocentric activities". In: *The IEEE International Conference on Computer Vision*. 2011, pp. 407–414.

[19] H. Pirsiavash and D. Ramanan. "Detecting Activities of Daily Living in First-person Camera Views". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 2847–2854.

[20] A. Furnari, G. M. Farinella, and S. Battiato. "Recognizing Personal Locations From Egocentric Videos". In: *IEEE Transactions on Human-Machine Systems* 47.1 (2017), pp. 6–18. DOI: 10.1109/THMS.2016.2612002.

[21] A. Furnari, G. M. Farinella, R. Bruna, and S. Battiato. "Affine Covariant Features for Fisheye Distortion Local Modeling". In: *IEEE Transactions on Image Processing* 26.2 (2017), pp. 696–710. DOI: 10.1109/TIP.2016.2627816.

[22] A. Furnari, G. M. Farinella, R. Bruna, and S. Battiato. "Distortion Adaptive Sobel Filters for the Gradient Estimation of Wide Angle Images". In: *under review in Journal of Visual Communication and Image Representation* (2017).

[23] A. Furnari, G. M. Farinella, and S. Battiato. "Temporal segmentation of egocentric videos to highlight personal locations of interest". In: *International Workshop on Egocentric Perception, Interaction and Computing (EPIC) in conjunction with ECCV*. 2016, pp. 474–489.

[24] A. Furnari, G. M. Farinella, and S. Battiato. "Recognizing Personal Contexts from Egocentric Images". In: *Workshop on Assistive Computer Vision and Robotics (ACVR) in conjunction with ICCV*. 2015.

[25] A. Furnari, G. M. Farinella, A. R. Bruna, and S. Battiato. "Distortion Adaptive Descriptors: Extending Gradient-Based Descriptors to Wide Angle Images". In: *Image Analysis and Processing (ICIAP)*. Vol. 9280. Lecture Notes in Computer Science. Springer, 2015, pp. 205–215.

[26] A. Furnari, G. M. Farinella, A. R. Bruna, and S. Battiato. "Generalized Sobel filters for gradient estimation of distorted images". In: *IEEE International Conference on Image Processing*. 2015, pp. 3250–3254.

[27] A. Furnari, G. M. Farinella, G. Puglisi, A. R. Bruna, and S. Battiato. "Affine region detectors on the fisheye domain". In: *2014 IEEE International Conference on Image Processing (ICIP)*. 2014, pp. 5681–5685.

[28] S. Mann. "Wearable computing: A first step toward personal imaging". In: *Computer* 30.2 (1997), pp. 25–32.

[29] S. Mann, M. A. Ali, R. Lo, and H. Wu. "FreeGlass for developers, "haccessibility", and Digital Eye Glass+ Lifeglogging research in a (sur/sous) veillance society". In: *Information Society (i-Society), 2013 International Conference on*. 2013, pp. 48–53.

[30] S. Mann. "Humanistic Intelligence : ' WearComp ' as a new framework and application for intelligent signal processing". In: *Proceedings of IEEE* 86.11 (1998), pp. 2123–2151.

[31] B. Schiele, T. Starner, B. Rhodes, B. Clarkson, and A. Pentland. "Situation aware computing with wearable computers". In: *Augmented Reality and Wearable Computers* (1999), pp. 1–20.

[32] T. Starner, J. Weaver, and A. Pentland. "Real-time american sign language recognition using desk and wearable computer based video". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.12 (1998), pp. 1371–1375.

[33] B. Schiele, N. Oliver, T. Jebara, and A. Pentland. "An Interactive Computer Vision System DyPERS: Dynamic Personal Enhanced Reality System". In: *Computer Vision Systems* 1542 (1999), pp. 51–65.

[34] A. Betancourt, P. Morerio, C. S. Regazzoni, and M. Rauterberg. "The Evolution of First Person Vision Methods: A Survey". In: *IEEE Transactions on Circuits and Systems for Video Technology* 25.5 (2015), pp. 744–760.

[35] Y. Li, Z. Ye, and J. M. Rehg. "Delving into Egocentric Actions". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 287–295.

[36] M. Ma, H. Fan, and K. M. Kitani. "Going Deeper into First-Person Activity Recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1894–1903.

[37] K. Aizawa, K. Ishijima, and M. Shiina. "Summarizing wearable video". In: *International Conference on Image Processing*. Vol. 3. 2001, pp. 398–401.

[38] Z. Lu and K. Grauman. "Story-driven summarization for egocentric video". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2714–2721.

[39] Y. J. Lee and K. Grauman. "Predicting important objects for egocentric video summarization". In: *International Journal of Computer Vision* 114.1 (2015), pp. 38–55.

[40] Y. Poleg, C. Arora, and S. Peleg. "Temporal segmentation of egocentric videos". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2537–2544.

[41] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora. "Compact CNN for Indexing Egocentric Videos". In: *Winter Conference on Applications of Computer Vision*. 2016.

[42] K. Yamada, Y. Sugano, T. Okabe, Y. Sato, A. Sugimoto, and K. Hiraki. "Attention prediction in egocentric video using motion and visual saliency". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 7087 LNCS. PART1. 2011, pp. 277–288.

[43] A. Fathi, Y. Li, and J. M. Rehg. "Learning to Recognize Daily Actions Using Gaze". In: *European Conference on Computer Vision*. Vol. 7572. 2012, pp. 314–327.

[44] Y. Li, A. Fathi, and J. M. Rehg. "Learning to predict gaze in egocentric video". In: *The IEEE International Conference on Computer Vision*. 2013, pp. 3216–3223.

[45] Y.-C. Su and K. Grauman. "Detecting Engagement in Egocentric Video". In: *European Conference on Computer Vision*. 2016.

[46] W. W. Mayol, B. J. Tordoff, and D. W. Murray. "Wearable visual robots". In: *Personal and Ubiquitous Computing* 6.1 (2002), pp. 37–48.

[47] W. Mayol-Cuevas, B. Tordoff, T. DeCampos, A. Davison, and D. Murray. "Active Vision for Wearables". In: *IEE Eurowearable,* 2003.

[48] Y. Yan, E. Ricci, G. Liu, and N. Sebe. "Egocentric daily activity recognition via multitask clustering". In: *Transactions on Image Processing* 24.10 (2015), pp. 2984–2995.

[49] K. K. Singh, K. Fatahalian, and A. A. Efros. "KrishnaCam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks". In: *Winter Conference on Applications of Computer Vision*. 2016.

[50] E. H. Spriggs, F. De La Torre, and M. Hebert. "Temporal segmentation and activity classification from first-person sensing". In: *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2009, pp. 17–24.

[51] R. Templeman, M. Korayem, D. Crandall, and K. Apu. "PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces". In: *Annual Network and Distributed System Security Symposium*. 2014, pp. 23–26.

[52] H. Aoki, B. Schiele, and A. Pentland. "Recognizing personal location from video". In: *Workshop on Perceptual User Interfaces*. 1998, pp. 79–82.

[53] B. Schiele, T. Jebara, and N. Oliver. "Sensory-augmented computing: wearing the museum's guide". In: *IEEE Micro* 21.3 (2001), pp. 44–52.

[54] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. "Context-based vision system for place and object recognition". In: *The IEEE International Conference on Computer Vision*. 2003.

[55] S. Sundaram and W. W. Mayol-Cuevas. "Egocentric visual event classification with location-based priors". In: *International Symposium on Visual Computing*. Springer. 2010, pp. 596–605.

[56] S. Sundaram and W. W. Mayol-Cuevas. "What are we doing here? egocentric activity recognition on the move for contextual mapping". In: *International Conference on Robotics and Automation*. 2012, pp. 877–882.

[57] N. Rhinehart and K. M. Kitani. "Learning Action Maps of Large Environments via First-Person Vision". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. June 2016.

[58] M. Wray, D. Moltisanti, W. Mayol-Cuevas, and D. Damen. "SEMBED: Semantic Embedding of Egocentric Action Videos". In: *European Conference on Computer Vision*. 2016, pp. 532–545.

[59] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. "Fast unsupervised ego-action learning for first-person sports videos". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 3241–3248.

[60] A. R. Doherty, N. Caprani, C. Ó. Conaire, V. Kalnikaite, C. Gurrin, A. F. Smeaton, and N. E. O'Connor. "Passively recognising human activities through lifelogging". In: *Computers in Human Behavior* 27.5 (2011), pp. 1948–1958.

[61] M. S. Ryoo and L. Matthies. "First-person activity recognition: What are they doing to me?" In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2730–2737.

[62] M. S. Ryoo, T. J. Fuchs, L. Xia, J. K. Aggarwal, and L. Matthies. "Robot-Centric Activity Prediction from First-Person Videos: What Will They Do to Me?" In: *Annual ACM/IEEE International Conference on Human-Robot Interaction*. 2015, pp. 295–302.

[63] D. Castro, S. Hickson, V. Bettadapura, E. Thomaz, G. Abowd, H. Christensen, and I. Essa. "Predicting Daily Activities from Egocentric Images Using Deep Learning". In: *International Symposium on Wearable Computing* (2015).

[64] S. Singh, C. Arora, and C. V. Jawahar. "First Person Action Recognition Using Deep Learned Descriptors". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. June 2016.

[65] Y. Zhou, B. Ni, R. Hong, X. Yang, and Q. Tian. "Cascaded Interactional Targeting Network for Egocentric Video Analysis". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1904–1913.

[66] A. R. Doherty and A. F. Smeaton. "Combining face detection and novelty to identify important events in a visual lifelog". In: *The IEE International Conference on Computer and Information Technology Workshops*. 2008, pp. 348–353.

[67] N. Jojic, A. Perina, and V. Murino. "Structural epitome: a way to summarize one's visual experience". In: *Advances in neural information processing systems*. 2010, pp. 1027–1035.

[68] O. Aghazadeh, J. Sullivan, and S. Carlsson. "Novelty detection from an egocentric perspective". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 3297–3304.

[69] Y. J. Lee, J. Ghosh, and K. Grauman. "Discovering important people and objects for egocentric video summarization." In: *The IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 6. 2012, p. 7.

[70] M. Bolaños, M. Dimiccoli, and P. Radeva. "Towards storytelling from visual lifelogging: An overview". In: *Transactions on Human-Machine Systems* (2015).

[71] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh. "Gaze-enabled Egocentric Video Summarization via Constrained Submodular Maximization". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2235–2244.

[72] B. Xiong, G. Kim, and L. Sigal. "Storyline representation of egocentric videos with an applications to story-based search". In: *The IEEE International Conference on Computer Vision*. 2015, pp. 4525–4533.

[73] V. Bettadapura, D. Castro, and I. Essa. "Discovering picturesque highlights from egocentric vacation videos". In: *Winter Conference on Applications of Computer Vision*. IEEE. 2016, pp. 1–9.

[74] T. Leelasawassuk, D. Damen, and W. W. Mayol-Cuevas. "Estimating visual attention from a head mounted IMU". In: *Proceedings of the 2015 ACM International Symposium on Wearable Computers*. 2015, pp. 147–150.

[75] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. Mayol-Cuevas. "You-Do, I-Learn: Discovering Task Relevant Objects and their Modes of Interaction from Multi-User Egocentric Video". In: *British Machine Vision Conference*. 2014.

[76] D. Damen, T. Leelasawassuk, and W. Mayol-Cuevas. "You-Do, I-Learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance". In: *Computer Vision and Image Understanding* 149 (2015), pp. 98–112.

[77] A. Torralba and A. Oliva. "Statistics of natural image categories". In: *Network: computation in neural systems* 14.3 (2003), pp. 391–412.

[78] C. Gurrin, A. F. Smeaton, and A. R. Doherty. "Lifelogging: Personal big data". In: *Foundations and trends in information retrieval* 8.1 (2014), pp. 1–125.

[79] M. L. Lee and A. K. Dey. "Capture & Access Lifelogging Assistive Technology for People with Episodic Memory Impairment Non-technical Solutions". In: *Workshop on Intelligent Systems for Assisted Cognition*. 2007, pp. 1–9.

[80] P. Wu, H.-K. Peng, J. Zhu, and Y. Zhang. "Senscare: Semi-automatic activity summarization system for elderly care". In: *International Conference on Mobile Computing, Applications, and Services*. 2011, pp. 1–19.

[81] E. Thomaz, A. Parnami, I. Essa, and G. D. Abowd. "Feasibility of Identifying Eating Moments from First-person Images Leveraging Human Computation". In: *SenseCam and Pervasive Imaging Conference*. 2013, pp. 26–33.

[82] J. Hernandez, L. Yin, J. M. Rehg, and R. W. Picard. "BioGlass: Physiological parameter estimation using a head-mounted wearable device". In: *Wireless Mobile Communication and Healthcare*. 2014.

[83] D. Ravì, B. Lo, and G. Yang. "Real-Time Food Intake Classification and Energy Expenditure Estimation on a Mobile Device". In: *Body Sensor Network, MIT, Boston, MA, USA* (2015).

[84] A. Ortis, G. M. Farinella, V. D'Amico, L. Addesso, G. Torrisi, and S. Battiato. "RECfusion: Automatic Video Curation Driven by Visual Content Popularity". In: *ACM Multimedia*. 2015.

[85] G. Lu, Y. Yan, L. Ren, J. Song, N. Sebe, and C. Kambhamettu. "Localize Me Anywhere, Anytime: A Multi-Task Point-Retrieval Approach". In: *The IEEE International Conference on Computer Vision*. 2015.

[86] H. Wannous, V. Dovgalecs, R. Mégret, and M. Daoudi. "Place recognition via 3d modeling for personal activity lifelog using wearable camera". In: *International Conference on Multimedia Modeling*. 2012, pp. 244–254.

[87] M. Dimiccoli, M. Bolaños, E. Talavera, M. Aghaei, S. G. Nikolov, and P. Radeva. "SR-Clustering: Semantic Regularized Clustering for Egocentric Photo Streams Segmentation". In: *arXiv preprint arXiv:1512.07143* (2015).

[88] A. Torralba and A. Oliva. "Semantic organization of scenes using discriminant structural templates". In: *The IEEE International Conference on Computer Vision* 2 (1999), pp. 1253–1258.

[89] A. Oliva and A. Torralba. "Modeling the shape of the scene: A holistic representation of the spatial envelope". In: *International Journal of Computer Vision* 42.3 (2001), pp. 145–175.

[90] G. M. Farinella and S. Battiato. "Scene classification in compressed and constrained domain". In: *IET Computer Vision* 5.5 (2011), pp. 320–334.

[91] G. M. Farinella, D. Ravì, V. Tomaselli, M. Guarnera, and S. Battiato. "Representing Scenes for Real–Time Context Classification on Mobile Devices". In: *Pattern Recognition* 48.4 (2015), pp. 1086–1100.

[92] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. "Learning deep features for scene recognition using places database". In: *Advances in Neural Information Processing Systems*. 2014, pp. 487–495.

[93] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman. "The devil is in the details: an evaluation of recent feature encoding methods." In: *British Machine Vision Conference*. Vol. 2. 2011, p. 8.

[94] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. "Return of the Devil in the Details: Delving Deep into Convolutional Nets". In: *British Machine Vision Conference*. 2014.

[95] F. Perronnin, J. Sánchez, and T. Mensink. "Improving the fisher kernel for large-scale image classification". In: *The IEEE International Conference on Computer Vision*. 2010, pp. 143–156.

[96] A. Vedaldi and B. Fulkerson. "VLFeat: An open and portable library of computer vision algorithms". In: *ACM international conference on Multimedia*. 2010, pp. 1469–1472.

[97] C. M. Bishop. *Pattern recognition and Machine Learning*. Springer, 2006.

[98] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[99] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. "Estimating the support of a high-dimensional distribution". In: *Neural computation* 13.7 (2001), pp. 1443–1471.

[100] C. Chang and C. Lin. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011), pp. 271–2727.

[101] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014).

[102] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[103] M. S. Ryoo, B. Rothrock, and L. Matthies. "Pooled motion features for first-person videos". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2014.

[104] Y. Gal and Z. Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *arXiv:1506.02142* (2015).

[105] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. "Caffe: Convolutional Architecture for Fast Feature Embedding." In: *ACM Multimedia*. Vol. 2. 2014, p. 4.

[106] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[107] P. H. S. Torr and A. Zisserman. "MLESAC: A new robust estimator with application to estimating image geometry". In: *Computer Vision and Image Understanding* 78.1 (2000), pp. 138–156.

[108] H. S. Koppula and A. Saxena. "Anticipating human activities using object affordances for reactive robotic response". In: *IEEE Transcations on Pattern Analysis and Machine Intelligence* 38.1 (2013), pp. 14–29.

[109] T. Lan, T.-C. Chen, and S. Savarese. "A hierarchical representation for future action prediction". In: *European Conference on Computer Vision*. 2014, pp. 689–704.

[110] Y. Zhou and T. L. Berg. "Temporal Perception and Prediction in Ego-Centric Video". In: *The IEEE International Conference on Computer Vision*. 2015, pp. 4498–4506.

[111] B. Soran, A. Farhadi, and L. Shapiro. "Generating Notifications for Missing Actions: Don't forget to turn the lights off!" In: *The IEEE International Conference on Computer Vision*. 2015, pp. 4669–4677.

[112] T. McCandless and K. Grauman. "Object-Centric Spatio-Temporal Pyramids for Egocentric Activity Recognition". In: *British Machine Vision Conference (BMVA)*. 2013, pp. 301–3011.

[113] M. S. Ryoo. "Human activity prediction: Early recognition of ongoing activities from streaming videos". In: *The IEEE International Conference on Computer Vision*. 2011, pp. 1036–1043.

[114] D. Huang, S. Yao, Y. Wang, and F. De La Torre. "Sequential max-margin event detectors". In: *European conference on computer vision*. 2014, pp. 410–424.

[115] M. Hoai and F. De La Torre. "Max-margin early event detectors". In: *International Journal of Computer Vision* 107.2 (2014), pp. 191–202.

[116] Y. Kong and Y. Fu. "Max-Margin Action Prediction Machine". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.9 (2016), pp. 1844–1858.

[117] S. Ma, L. Sigal, and S. Sclaroff. "Learning Activity Progression in LSTMs for Activity Detection and Early Detection". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[118] K. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. "Activity Forecasting". In: *European Conference on Computer Vision*. 2012, pp. 201–214.

[119] C. Vondrick, H. Pirsiavash, and A. Torralba. "Anticipating Visual Representations with Unlabeled Video". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[120] H. Soo Park, J.-j. Hwang, Y. Niu, and J. Shi. "Egocentric Future Localization". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4697–4705.

[121] Y.-C. Su and K. Grauman. "Leaving Some Stones Unturned: Dynamic Feature Prioritization for Activity Detection in Streaming Video". In: *European Conference on Computer Vision*. 2016.

[122] G. Bertasius, H. S. Park, S. X. Yu, and J. Shi. "First Person Action-Object Detection with EgoNet". In: *ArXiv* 1 (2016).

[123] H. Wang, A. Kläser, C. Schmid, and C. L. Liu. "Dense trajectories and motion boundary descriptors for action recognition". In: *International Journal of Computer Vision* 103.1 (2013), pp. 60–79.

[124] R. Cipolla and A. Blake. "Surface orientation and time to contact from image divergence and deformation". In: *European Conference on Computer Vision*. 1992, pp. 187–202.

[125] G. Nebehay and R. Plugfelder. "Clustering of Static-Adaptive Correspondences for Deformable Object Tracking". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[126] S. Ren, K. He, and R. Girshick. "Faster R-CNN Towards Real-Time Object Detection With Region Proposal Networks". In: *Advances In Neural Information Processing Systems*. 2015, pp. 1–9.

[127] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recoginition". In: *International Conference on Learning Representations*. 2015, pp. 1–14.

[128] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. "Simple online and realtime tracking". In: *IEEE International Conference on Image Processing*. Sept. 2016, pp. 3464–3468.

[129] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2 (2010), pp. 303–338.

[130] B. Sven, L. Stefan, C. David, and Y. Chen. "Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions". In: *The IEEE International Conference on Computer Vision*. 2015.

[131] M. F. Land. "Eye movements and the control of actions in everyday life". In: *Progress in retinal and eye research* 25.3 (2006), pp. 296–324.

[132] A. Furnari, G. M. Farinella, and S. Battiato. "An Experimental Analysis of Saliency Detection with respect to Three Saliency Levels". In: *Workshop on Assistive Computer Vision and Robotics (ACVR) in conjunction with ECCV (2014)*. Vol. 8927. Lecture Notes in Computer Science. 2014, pp. 806–821.

[133] E. Vig, M. Dorr, and D. Cox. "Large-scale optimization of hierarchical features for saliency prediction in natural images". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2798–2805.

[134] H. J. Seo and P. Milanfar. "Static and space-time visual saliency detection by self-resemblance". In: *Journal of vision* 9.12 (2009), pp. 15–15.

[135] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Měch. "Minimum Barrier Salient Object Detection at 80 FPS". In: *The IEEE International Conference on Computer Vision*. 2015.

[136] M. Lourenço, J. P. Barreto, and F. Vasconcelos. "sRD-SIFT: Keypoint detection and matching in images with radial distortion". In: *IEEE Transactions on Robotics* 28.3 (2012), pp. 752–760.

[137] C. Hughes, P. Denny, E. Jones, and M. Glavin. "Accuracy of fish-eye lens models". In: *Applied Optics* 49.17 (2010), pp. 3338–47.

[138] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. 2005, pp. 886–893.

[139] M. M. Fleck. "Perspective projection: the wrong imaging model". In: *Department of Computer Science, University of Iowa* (1995).

[140] L. Puig and J. J. Guerrero. *Omnidirectional Vision Systems*. Springer, 2013.

[141] K. Miyamoto. "Fish eye lens". In: *Journal of the Optical Society of America (JOSA)* (1964), pp. 2–3.

[142] S. Baker and S. K. Nayar. "A theory of catadioptric image formation". In: *International Conference on Computer Vision*. 1998, pp. 35–42.

[143] C. Geyer and K. Daniilidis. "A unifying theory for central panoramic systems and practical implications". In: *European Conference on Computer Vision*. 2000, pp. 445–461.

[144] F. Devernay and O. Faugeras. "Straight lines have to be straight". In: *Machine vision and applications* 13.1 (2001), pp. 14–24.

[145] A. W. Fitzgibbon. "Simultaneous linear estimation of multiple view geometry and lens distortion". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. 2001.

[146] C. Hughes, E. Jones, M. Glavin, and P. Denny. "Validation of Polynomial-based Equidistance Fish-Eye Models". In: *Signals and Systems Conference*. 2009.

[147] P. Hansen, P. Corke, W. Boles, and K. Daniilidis. "Scale-invariant features on the sphere". In: *The IEEE International Conference on Computer Vision*. 2007.

[148] M. Lourenço, J. P. Barreto, and A. Malti. "Feature detection and matching in images with radial distortion". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2010.

[149] I. Cinaroglu and Y. Bastanlar. "A direct approach for human detection with catadioptric omnidirectional cameras". In: *Signal Processing and Communications Applications Conference*. 2014, pp. 2275–2279.

[150] I. Cinaroglu and Y. Bastanlar. "A direct approach for object detection with catadioptric omnidirectional cameras". In: *Signal, Image and Video Processing* 10.2 (2016), pp. 413–420.

[151] J. Kannala and S. Brandt. "A generic camera calibration method for fish-eye lenses". In: *International Conference of Pattern Recognition*. 2004.

[152] D. Scaramuzza, A. Martinelli, and R. Siegwart. "A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion". In: *International Conference on Computer Vision Systems*. 2006.

[153] D. Scaramuzza, A. Martinelli, and R. Siegwart. "A Toolbox for Easily Calibrating Omnidirectional Cameras". In: *International Conference on Intelligent Robots and Systems*. 2006, pp. 5695–5701.

[154] J. J. Kumler and M. L. Bauer. "Fish-eye lens designs and their relative performance". In: *International Symposium on Optical Science and Technology*. 2000, pp. 360–369.

[155] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. "A Comparison of Affine Region Detectors". In: *International Journal of Computer Vision* 65 (2005), pp. 43–72.

[156] I. Bogdanova, X. Bresson, J. Thiran, and P. Vandergheynst. "Scale space analysis and active contours for omnidirectional images." In: *IEEE transactions on image processing* 16.7 (2007), pp. 1888–901.

[157] L. Puig and J. J. Guerrero. "Scale space for central catadioptric systems: Towards a generic camera feature extractor". In: *The IEEE International Conference on Computer Vision*. 2011, pp. 1599–1606.

[158] Z. Arican and P. Frossard. "OmniSIFT: Scale invariant features in omnidirectional images". In: *International Conference on Image Processing*. 2010, pp. 3505–3508.

[159] J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire, and J. Thiran. "Scale Invariant Feature Transform on the Sphere: Theory and Applications". In: *International Journal of Computer Vision* 98.2 (2011), pp. 217–241.

[160] M. S. Islam and L. J. Kitchen. "Straight-edge extraction in distorted images using gradient correction". In: *Digital Image Computing: Techniques and Applications*. 2009.

[161] K. Koser and R. Koch. "Perspectively Invariant Normal Features". In: *The IEEE International Conference on Computer Vision*. 2007, pp. 1–8.

[162] L. Puig, J. J. Guerrero, and P. Sturm. "Hybrid matching of uncalibrated omnidirectional and perspective images". In: *Informatics in Control, Automation and Robotics*. 2002.

[163] M. Saito, K. Kitaguchi, G. Kimura, and M. Hashimoto. "People Detection and Tracking from Fish-eye Image Based on Probabilistic Appearance Model". In: *Society of Instrument and Control Engineers Annual Conference* (2011), pp. 435–440.

[164] J. Masci, D. Migliore, M. M. Bronstein, and J. Schmidhuber. "Descriptor Learning for Omnidirectional Image Matching". In: *Registration and Recognition in Images and Videos*. Vol. 532. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2014, pp. 49–62.

[165] K. Mikolajczyk and C. Schmid. "Performance evaluation of local descriptors." In: *IEEE transactions on pattern analysis and machine intelligence* 27.10 (2005), pp. 1615–30.

[166] Z. Wang, B. Fan, and F. Wu. "Local Intensity Order Pattern for Feature Description". In: *The IEEE International Conference on Computer Vision*. 2011, pp. 603–610.

[167] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions". In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2002.

[168] K. Mikolajczyk and C. Schmid. "An Affine Invariant Interest Point Detector". In: *Proceedings of the 7th European Conference on Computer Vision (ECCV)*. 2002.

[169] K. Mikolajczyk and C. Schmid. "Scale & affine invariant interest point detectors". In: *International Journal of Computer Vision* 60.1 (2004), pp. 63–86.

[170] H. W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[171] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. "Frequency-tuned salient region detection". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 1597–1604.

[172] J. Canny. "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8.6 (1986), pp. 679–698.

[173] P. E. Duda and O. Richard. "Hart, Pattern Classification and Scene Analysis". In: John Wiley and Sons, New York, 1973, pp. 271–272.

[174] J. Domke and Y. Aloimonos. "Deformation and Viewpoint Invariant Color Histograms." In: *The British Machine Vision Conference*. 2006, pp. 509–518.

[175] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless. "Gradientshop: A gradient-domain optimization framework for image and video filtering". In: *ACM Transactions on Graphics* 29.2 (2010).

[176] S. Battiato, G. M. Farinella, G. Puglisi, and D. Ravì. "Saliency Based Selection of Gradient Vector Flow Paths for Content Aware Image Resizing". In: *IEEE Transactions on Image Processing* 23.5 (2014), pp. 2081–2095.

[177] G. Messina, S. Battiato, M. Mancuso, and A. Buemi. "Improving image resolution by adaptive back–projection correction techniques". In: *IEEE Transactions on Consumer Electronics* 48.3 (2002), pp. 409–416.

[178] P. Pérez, M. Gangnet, and A. Blake. "Poisson image editing". In: *ACM Transactions on Graphics*. Vol. 22. 3. 2003, pp. 313–318.

[179] K. K. Pingle. "Visual perception by a computer". In: *Automatic interpretation and classification of images* (1969), pp. 277–284.

[180] P. E. Danielsson and O. Seger. "Generalized and separable Sobel operators". In: *Machine Vision for Three-dimensional Scenes* (1990), pp. 347–379.

[181] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato. "SIFT features tracking for video stabilization". In: *Image Analysis and Processing*. 2007, pp. 825–830.

[182] G. Dorkó and C. Schmid. "Selection of scale-invariant parts for object class recognition". In: *The IEEE International Conference on Computer Vision*. IEEE. 2003, pp. 634–639.

[183] M. Brown and D. G. Lowe. "Recognising panoramas." In: *The IEEE International Conference on Computer Vision*. Vol. 3. 2003, p. 1218.

[184] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. "Object detection with discriminatively trained part-based models". In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2010), pp. 1627–1645.

[185] C. Liu, J. Yuen, and A. Torralba. "Sift flow: Dense correspondence across scenes and its applications". In: *IEEE transactions on Pattern Analysis and Machine Intelligence* 33.5 (2011), pp. 978–994.

[186] G. M. Farinella, D. Allegra, and F. Stanco. "A benchmark dataset to study the representation of food images". In: *European Conference on Computer Vision Workshops*. 2014, pp. 584–599.

[187] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 2169–2178.

[188] S. Battiato, G. M. Farinella, A. Furnari, G. Puglisi, A. Snijders, and J. Spiekstra. "An integrated system for vehicle tracking and classification". In: *Expert Systems with Applications* 42.21 (2015), pp. 7263–7275.

[189] F. Ragusa, V. Tomaselli, A. Furnari, S. Battiato, and G. M. Farinella. "Food vs Non-Food Classification". In: *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*. ACM. 2016, pp. 77–81.

[190] S. Battiato, G. M. Farinella, A. Furnari, G. Puglisi, A. Snijders, and J. Spiekstra. "Vehicle tracking based on customized template matching". In: *International Conference on Computer Vision Theory and Applications*. Vol. 2. 2014, pp. 755–760.

[191] B. Sebastiano, F. Giovanni Maria, F. Antonino, and P. Giovanni. "A Customized System for Vehicle Tracking and Classification". In: *European Conference on Mathematics for Industry (ECMI)*. 2014.

[192] D. Scandura, S. Battiato, V. Bruno, F. Cannavo, G. M. Farinella, A. Furnari, M. Mattia, G. Pappalardo, G. Puglisi, and U. Weigmuller. "Image Processing Techniques to Estimate the Propagation of Ground Deformation at Mt. Etna (Italy) from ALOS PALSAR InSAR Data". In: *AGU Fall Meeting Abstracts*. Vol. 1. 2014.