ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN
INFORMATICA

Ciclo XXVIII

Settore Concorsuale:
09/H1 - Sistemi di elaborazione delle informazioni

Settore Scientifico Disciplinare:
ING-INF/05 - Sistemi di elaborazione delle informazioni

Computer Vision Techniques for Ambient Intelligence Applications

Presentata da: Simone Buoncompagni

Coordinatore Dottorato:
Prof. Paolo Ciaccia

Relatore:
Prof. Dario Maio

Esame finale anno: 2016

Abstract

Ambient Intelligence (AmI) is a muldisciplinary area which refers to environments that are sensitive and responsive to the presence of people and objects. The rapid progress of technology and simultaneous reduction of hardware costs characterizing the recent years have enlarged the number of possible AmI applications, thus raising at the same time new research challenges. In particular, one important requirement in AmI is providing a proactive support to people in their everyday working and free-time activities. To this aim, Computer Vision represents a core research track since only through suitable vision devices and techniques it is possible to detect elements of interest and understand the occurring events. The goal of this thesis is presenting and demonstrating efficacy of novel machine vision research contributes for different AmI scenarios: object keypoints analysis for Augmented Reality purpose, segmentation of natural images for plant species recognition and heterogeneous people identification in unconstrained environments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Ambient Intelligence and smart environments

The rapid progress of technology and the simultaneous reduction of hardware costs characterizing the last decades have made computers more and more present in everyday life. Initially dedicated to very specific applications, microprocessors are now embedded in various common equipments/objects and used, more or less consciously, by a wide range of population. Moreover, computer and portable devices with different capabilities and interfaces are increasingly becoming part of our home, cities and working places since sensors, actuators and processing units can nowadays be purchased at very affordable prices. The challenge is now to go a step forward making these technological facilities "smart", *i.e.* sensitive and reactive to the presence of people and objects.

The main idea is to embed technology in the environment, enriching it with the capability of detecting/recognizing objects and people in a scene and adapting itself to user specific needs and preferences [60]. This ability is usually referred to as Ambient Intelligence (AmI): in an intelligent ambient, technology can be networked and used with the coordination of highly effective and efficient software. In particular, software is not only responsible for events and relevant contexts understanding in a specific environment, but must also take sensible decisions in real time or *a posteriori* [116].

On hardware side, the physical infrastructure which supports a generic AmI system is referred to as *smart environment* [44]. The electronics devices placed inside the environment must be user friendly, not intrusive, ubiquitous, sensitive, adaptive responsive and, obviously, smart. In a smart environment, devices work in concert to support people in carrying out their usual activities, tasks and rituals in an easy, natural way using information and intelligence that is hidden in the network connecting such devices.

The popularity of AmI greatly increased in recent years not only due to much higher affordability of devices but also because of the increasing presence of connectivity. Such feature allowed in particular the concept of the *Internet of Things* (IoT) [87], which has tied in closely with the popularization of AmI. In IoT, physical objects or "things" are embedded with electronics, software, sensors and network connectivity thus enabling these objects to collect and exchange information. Furthermore, in the IoT definition, objects should be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, thus resulting in improved efficiency, accuracy and economic benefit.

With reference to AmI, we can point out many examples of intelligent (or smart) environments: a home where lighting, heating, security and entertainment systems are automatically managed depending on the presence or the absence of people inside or outside, a factory where equipment can interact with workers and maintenance technicians, a museum where technology can help the fruition of the exhibition giving real time information to a certain user, a smart school where students can avail themselves of assisted learning with modern instruments, a public park or game reserve where technology helps visitors in understanding the surrounding nature and all the kind of applications that are typically placed in smart city contexts [34, 138].

Without moving here into details of the above mentioned examples, the reader can easily notice that AmI is a multidisciplinary area which embraces a variety of pre-existing fields like Artificial Intelligence and Robots, Sensors Network, Human Computer Interaction, Multi-Agent Systems and Pervasive Computing and Communication. While such areas are offering to AmI designers various ideas to enrich the user experience with novel and useful services, it is undeniable that Computer Vision represents a valuable enabling science for AmI in order to reveal and understand human behavior in its different social settings thus providing rich information in an unobtrusive modality. Computer Vision has applications for ambient assisted living [3], human-computer interaction [80, 140], surveillance and identification [153], abnormal event detection [117] and for many more challenges

falling under the domain of AmI [137].
The main focus of this thesis is about how machine vision and novel techniques can support AmI in its peculiar application. To this end, we will illustrate in the next chapters some significant research contributes in different scenarios whereas in the following part of this introductory chapter we are going to deeply discuss main links between AmI and Computer Vision, in order to further emphasize the role of Computer Vision.

1.2 The role of Computer Vision in AmI

As explained in Section 1.1, a digital environment is able to support people in their life in a proactive and sensible way [11]. To pursue such goal, vision and sensor networks play a decisive role [116], providing techniques to acquire information from the environment, detect the presence of elements in a scene and understand the occurring events. As illustrated in Figure 1.1, vision networks can offer a variety of inferences from the event and human activities of interest [4]. The acquired information and data are transferred to high-level reasoning modules for knowledge accumulation in applications involving behavior monitoring or for reacting to specific situations. Another type of interaction between vision and higher-level application context may involve visualization of events or human actions which can take the form of video distribution or avatar-based visual communication [5]. In both kinds of interfaces, two-way communication between vision/data processing modules and high-level reasoning/visualization modules can enable effective information acquisition by guiding vision to the features of interest and validating the vision outputs based on prior knowledge, context and accumulated behavior models.
While many novel sensory modalities are offering to AmI applications designers new possibilities to realize these goals, Computer Vision remains the most important modality in providing rich information in an unobtrusive way [137]. Hereinafter we provide a short overview of the latest trends in algorithm and application development based on visual inputs and Computer Vision processing.

- Distributed vision algorithms: dealing with a large number of cameras have been a hot topic for Computer Vision research in recent years. Nevertheless, many multi-camera algorithms presuppose that digital content captured

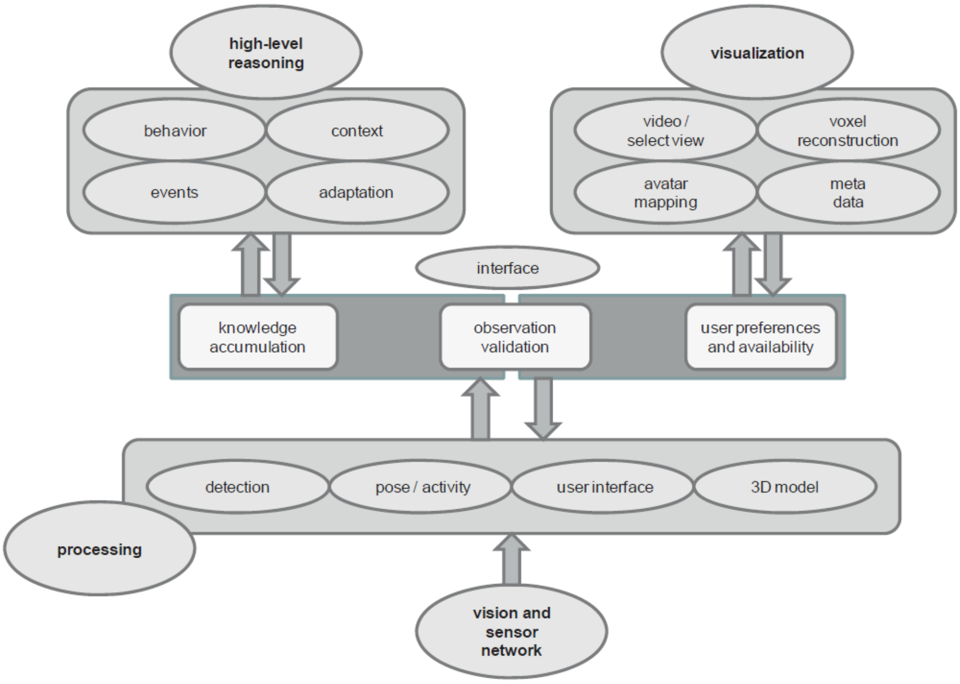


Figure 1.1: The role of vision and sensor networks in interaction with high-level reasoning and visualization modules (figure from [116]).

from all cameras is transmitted in a loss-less fashion to a central processor that finally processes the received data to solve the problem or perform the required task. This hypothesis is not realistic for emerging wireless camera networks, which may contain processing units with reduced capability, batteries with degraded duration and antennas with limited power. Therefore, methods for topology estimation, camera calibration and visual tracking are necessary to cope with this new type of computationally constrained devices and cameras [129].

- Vision - based people tracking and biometric recognition: human pose tracking is a key enabling technology for a lot of applications, such as the analysis of human activities for perceptive environments and innovative man-machine interfaces. Despite years of research, 3D human pose tracking remains a challenging problem. Pose estimation and tracking is currently possible in constrained environments with multiple cameras and under limited occlusions and scene clutter. Monocular 3D pose tracking in unconstrained environment is much more difficult, but recent results are promising. In these areas, important contributions come from biometric recognition research [116].

- Human behavior understanding: human behavior has been a focus of Computer Vision research for a long time, mostly on a personal signal level, where the face and body of a person are tracked and analyzed for a specific purpose. Automatic classification of human behavior involves understanding of bodily motion, gestures and signs, analysis of facial expressions and other affective signals. On a higher level, these signals are integrated with the contextual properties of an application domain, to constrain the otherwise immense variation in expressive human behavior. Research now moves towards analysis in more natural settings with uncontrolled conditions, adding more stringent real time constraints and most importantly, interaction dynamics. The relative position of people in interaction, their postures, gestures, non-verbal behavior and the way they respond to each other carry significant social clues which are essential to correctly infer contextual properties of the interaction. The recently flourishing field of social signal processing is very relevant for AmI researchers, as it attempts to systematically categorize these signals, developing tools for their automatic recognition [137].

- Object detection, recognition, tracking and segmentation: a smart environment must respond effectively not only according to the presence and behavior of people that are inside it, but also according to presence and movements of generic objects (especially in video surveillance applications). For example, one sub-area in the context of AmI concerns the support of moving objects, *i.e.* to monitor the course of events while an object crosses a smart environment and intervene if the environment should provide assistance [116]. For this purpose, a smart environment has to employ methods of knowledge representation and spatio-temporal reasoning. Moreover, as well as people recognition and tracking, Computer Vision has to deal with realistic and unconstrained conditions.

1.3 Computer Vision and AmI: main research contributes

As illustrated in Section 1.2, AmI includes many different concrete applications and Computer Vision performs a crucial role in each of them. By enriching an environment with vision devices and providing suitable algorithms it is possible to obtain awareness and responsiveness to the presence of humans and objects, in a user friendly, interactive and efficient way. On one hand, each application

requires specific solutions and dedicated vision techniques to deal with occurring environmental conditions and use cases. On the other hand, techniques must be scalable, efficient and reusable in order to adapt to evolving requirements.
In this thesis, we will provide different research contributes falling under different AmI scenarios: in Chapter 2 we will examine the Augmented Reality field providing a novel technique for keypoint selection and ranking [26, 27] in order to build compact but effective object models and offer support for real time object recognition and pose estimation, in Chapter 3 we will investigate segmentation of natural images and we will apply it to leaf segmentation thus providing a novel approach to extract accurately the leaf shape in loosely controlled conditions [148]. Finally, in Chapter 4 we will tackle the problem of heterogeneous face recognition focusing on recognition from sketch. In particular, we will show how simple but effective shape features could help in the process of suspect identification [25].
Each of the previous mentioned scenarios has strong connections with AmI. In the following subsections we are going to illustrate the main features of such topics thus summarizing the research contexts, whereas in Chapters 2, 3 and 4 we will describe in details each single research contribute.

1.3.1 Object keypoints analysis to support Augmented Reality

Providing a universally accepted definition about the meaning of Augmented Reality (AR in the following) is quite difficult since in the literature different alternative definitions about such topic are reported and used. Relying upon the Handbook of Augmented Reality [62] we can define AR as a real time direct or indirect view of a physical real-world environment that has been enhanced/augmented by adding virtual computer-generated information to it. AR is both interactive and registered in 3D as well as it combines real and virtual objects. It is important to specify that AR can be potentially applied to all human senses, including not only sight but also taste, hearing, touch, and smell: under this point of view we can define AR as an enrichment of the human sensory perception by building information, generally handled and conveyed electronically, that would not be perceived with the five senses.
Because of the necessary requirements for its use, both in terms of functionality and computing power, AR had initially a limited diffusion and was used in

particular contexts, such as the military field, medical research and education. Nowadays the scenario has radically changed: thanks to the increasing presence of enabling technologies [181], AR can be applied in manifold AmI contexts [47]. Potentially, any object that surrounds us could be enhanced by appropriate virtual information. Among others, the most important application fields are military, emergency services, art and cultural heritage, games, tourism, education, trade, entertainment, industrial systems, design, construction, surgery, etc.

Figure 1.2: Different types of enabling technologies for AR, from left to right: head-mounted device (figure from [173]), smartphone (figure from [171]), special projection device (figure from [174]).

Taking advantage of the Azuma research contribute [12], we can define an AR system to have the following properties: it combines real and virtual objects in a real environment, it has to interact and provide real time experience, it has to register and align real and virtual objects with each other. In other words, AR aims at simplifying the user's life by bringing virtual information not only to his immediate surroundings, but also to any indirect view of the real environment, such as live video stream. Furthermore, it enhances the user's perception of the tangible world and the interaction with it.
It is important not to confuse the concepts of Augmented Reality and Virtual Reality among them: to this aim, Milgram [114] defined a continuum of real-to-virtual environments in which AR is one part of the general area of Mixed Reality (Figure 1.3).
In Augmented Virtuality (where real objects are added to virtual ones) and Vir-

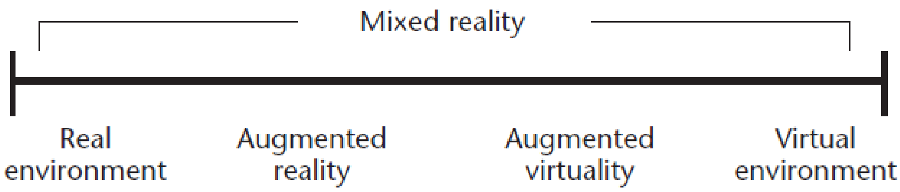


Figure 1.3: The Reality-Virtuality Continuum defined by P. Milgram in [114].

tual environments (or Virtual Reality) the surrounding environment is virtual. On the contrary, in Augmented Reality the surrounding environment is the real one. Certain AR applications also require removing real objects from the perceived environment, in addition to adding virtual objects. For example, an AR visualization of a building that stood at a certain location might remove the building that exists there today. Some researchers refer to the task of removing real objects as *mediated* or *diminished* reality, but we consider it a subset of AR.

Regarding to this thesis, we will focus on AR without dealing with augmented virtuality or virtual environments. In particular, special attention will be paid to the *vision-based* AR, where the analysis of a certain scene is performed taking video frames (or images, generally speaking) coming from cameras and the sensorial perception of the augmented content is given through appropriate viewer devices like head-mounted displays or glasses, smartphone/tablet displays or special projection devices (see Figure 1.2).

In such scenario, object detection and pose estimation represent the main building blocks of any possible application, since only a correct pose recovery makes possible the correct projection of a certain augmented content depending on the user perspective. In the so-called *marker-based* applications some artificial landmarks as 2D-bar codes, beacons, pictorial markers, etc. [62, 126] are employed to simplify such operation. Nonetheless, markers cannot be always used thus making object recognition and pose estimation very challenging. The standard approach in *markerless* scenario takes advantage of keypoint detection and descriptor matching in order to find correspondences between the real time scene/object and an already arranged object model. Such method has been proved to work well in many different contexts but suffers especially when dealing with resource-constrained platforms such as mobile devices. To tackle the problem, we introduce and describe in Chapter 2 a novel technique to reduce the set of detected keypoints only to the main *salient* ones, thus employing the related descriptors for the following matching and pose estimation tasks. Moreover, in Chapter 2 we will prove also that our approach offers an effective solution for different AR scenarios like maintenance and tourist applications.

1.3.2 Segmentation of natural images

Image segmentation is the process of partitioning a digital image into multiple and non-overlapping regions (segments). Segments are typically regions or categories related to different objects or parts of objects. At the end of segmentation process, each pixel is assigned to a different category. When each category has a specific meaning connected to the real world (a car, a cat, a person, etc.) we refer to such process also as semantic segmentation.

Differently from object detection where the ideal output is represented by the minimal bounding box that includes a certain object, in image segmentation the ideal output is composed by all the pixels that constitute the object of interest thus isolating it from the clutter or the background . Points belonging to the same segmented region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions/image parts are instead significantly different with respect to the same characteristic(s) [13, 143]. The main goal of segmentation is to decompose the image into parts in order to allow further and more complex investigation. For example, starting from a segmented image it is straightforward to extract boundaries and shapes of segmented objects as well as classify the image depending on the object classes detected in the image itself.

A huge literature contribution about general purpose segmentation have been proposed in the last three decades, nonetheless some open problems related to segmentation of natural images and images from outdoor environments still remain. Segmentation of natural images is an important preliminary step in many AmI applications such as analysis of green urban environment, recognition of plant species for professional and educational purposes, detection of illness/malformation of plants in agricultural plantation and so on. In such AmI applications, segmentation covers a crucial role since a pixel-level precision is required to locate the object of interest or a part of it (see Figure 1.4). For example, a good segmentation is decisive in plant recognition since accurate leaf shape needs to be extracted whereas in illness detection some specific colored areas which represent a warning about the presence of a disease must be revealed. Once segmentation is completed, proper actions or specific analysis can be carried out: *e.g.* , recognition of the given plant or alerting professionals to fix the detected disease. Unfortunately, segmentation has to be usually performed on images taken under irregular and/or unsupervised illumination conditions, thus making such task quite complex.

In Chapter 3 we will focus our attention on plant species identification by proposing a robust and accurate method for segmenting specular objects acquired under loosely controlled conditions. In particular, we will refer to a concrete case study based on leaves since leaf segmentation plays a crucial role for plant identification,

Figure 1.4: Natural image segmentation examples from CVPPP2015 challenge.

and accurately capturing the local boundary structures is critical for the success of the recognition in state-of-the-art applications. Popular techniques are based on Expectation-Maximization and estimate the color distributions of the background and foreground pixels of the input image. As we show, such approaches suffer in presence of shadows and reflections thus leading to inaccurate detected shapes. Classification-based methods are more robust because they can exploit prior information, however they do not adapt to the specific capturing conditions for the input image. Methods with regularization terms are prone to smooth the segments boundaries, which is undesirable. We will show we can get the best of the EM-based and classification-based methods by first segmenting the pixels around the leaf boundary, and use them to initialize the color distributions of an EM optimization. We show that this simple approach results in a robust and accurate method.

1.3.3 Heterogeneous identification in unconstrained environments

Surveillance and people identification are relevant tasks in Ambient Intelligence discipline [116]. Numerous and different biometric traits can be exploited in order to recognize a person or a class of people, thus originating different biometric systems with different level of reliability depending on the trait [185]:

- biometric systems based on static elements such as fingerprint, face, palmprint, hand veins, iris, ear and DNA;
- biometric systems based on dynamic/behavioral traits such as gait, voice, signature, keystroke and gesture;
- biometric systems based on recognition of chemical or organic aspects such as scent and virus presence.

In the recent years, due to the technology advances and the increased demand for security and reduction of human costs, biometric applications have been constantly increasing. Most common biometric sensing modalities available today fall into one of three categories: *contact*, *contact-less* and *at-a-distance* [162]. In contact-based systems the user is required to touch the sensor (*e.g.* fingerprint or signature acquisition) whereas in contact-less or at-a-distance systems the user can stay at a predetermined distance from the sensor (*e.g.* iris, face or gait recognition).

Contact-based identification systems are probably the most widespread but they require an high level of cooperation by the user itself. Moreover, since a physical contact with the system needs to be carried out in order to acquire biometric samples, such recognition procedure is usually perceived as strongly invasive. Due to these reasons, contact-based systems and related biometric traits are not effective in Ambient Intelligence contexts [162]. In a smart environment the information about people are acquired through cameras or at-a-distance devices and identification must be performed in an unobtrusive and contact-less way, with a minimal or no-cooperation offered by users. Therefore, the main reliable biometric trait that can be exploited in such scenarios results to be face, since its acquisition can be totally transparent to the monitored user while he/she is walking or doing his/her activities.

With reference to surveillance, one emerging trend in Ambient Intelligence and smart environment is face recognition through heterogeneous sources [92] [56]. Standard face recognition is usually performed by comparing a photo against a gallery of images. However, such general approach is not always applicable especially when surveillance has to be carried out nighttime or in unconstrained/crowded

environments. In these scenarios, alternative technologies are employed to capture subject face: thermal images, near infrared images and sketches/identikit provided by eyewitnesses (see Figure 1.5). For this reason, new heterogeneous matching techniques need to be studied and implemented in order to perform matching across different face modalities, thus enabling face recognition in challenging and unusual situations.
In particular, recognition from a face sketch is an interesting problem from the

(a) (b) (c)

Figure 1.5: Subject acquisition from different sources: (a) RGB camera, (b) infrared camera and (c) manual sketch/identikit.

point of view of possible practical applications. Usually, a face sketch is realized by a forensic artist based on the verbal description provided by an eyewitness. In other words, in such scenario the acquiring device is not a camera or a special video surveillance device but a physical person who assists to a criminal event and then provides his memories to the police. In the standard approach, recognition is performed either by broadcasting the identikit to the population with the hope that someone could recognize the depicted subject or by manually comparing the identikit against the already available mug shot photo databases. In Chapter 4 we propose a novel and automatic technique to compare sketches and real photos by extracting common features which enable an heterogeneous, fast and scalable matching.

Chapter 2

Saliency-based Keypoint Selection and Ranking

2.1 Introduction

Keypoints and local descriptors are nowadays largely used for image classification, object detection and recognition, object localization and tracking, image registration and multi-view vision. In the context of object detection/matching, the reference model of a given object can be created by extracting a set of keypoints (*e.g.* , Harris [73], FAST [134], Shi-Tomasi [145], SUSAN [152], Difference of Gaussians [105]) and associating to each of them a local descriptor (*e.g.* , SIFT [105], SURF [14], BRIEF [30]). When the object has to be searched/matched in/against a given test image, a new set of keypoints and related descriptors is extracted and the two sets are matched to consolidate a similarity score. This basic approach works well for some applications but it is not always applicable to real time scenarios especially when low performance computing architectures are involved. In fact, even if time-efficient keypoint detectors and descriptors are used, the potential very large number of resulting keypoints can require heavy computational demands for the matching phase. Decreasing the number of keypoints reduces the matching complexity but this strategy is effective only if the retained keypoints are stable across different instances of the same object and at the same time the associated descriptors are sufficiently discriminant to avoid increasing matching errors.

In this chapter we present a new approach to select good keypoints for object detection and matching based on their *saliency*, under moderate viewpoint and light changes. Keypoint selection is performed through a training stage starting from one or multiple images (views) of the given object. Keypoint saliency is defined in terms of keypoint *detectability* and descriptor *repeatability* and *distinctiveness*. Even though the proposed approach is independent of the keypoint detector and local descriptor, FAST detector [132, 133] and BRIEF descriptor [30] are here used to maximize efficiency thus enabling real time applications.
The chapter includes a summary of systematic experiments conducted on the proposed selection approach and a comparison with state-of-the-art methods. We prove the effectiveness of our technique in a typical real time Augmented Reality scenario, where keypoint matching is used to effectively recover the camera viewpoint thus enabling different possible applications.
The chapter is organized as follows: in Section 2.2 we review the literature about object detection and description whereas a general overview of the proposed approach for detection/matching is presented in Section 2.3. In Section 2.4 we provide a formal definition of keypoint saliency and we introduce the keypoint selection scheme while Section 2.5 presents the experiments carried out to evaluate the approach. Finally, in Section 2.6 we propose different case studies where our saliency evaluation framework could be applied.

2.2 Related work

Local image structures like corners and blobs (here addressed with the equivalent term of *keypoint*) are crucial elements for a variety of computer vision tasks such as tracking, localization, simultaneous location and mapping (SLAM), image matching and recognition. Therefore, detection and description of such structures represent one of the main issues examined by machine vision scientists in the last three decades: a countless amount of different techniques have been introduced in literature. Moreover, due to the increasing need of real time performance to deal with live video stream and mobile devices with restricted computational capabilities, efficiency and speed recently assumed considerable priority becoming a key evaluation factor.
In Subsection 2.2.1 and Subsection 2.2.2, short but thoroughly overviews of the main contributes for keypoint detection and description are presented, respectively.

Finally, in Subsection 2.2.3 studies about keypoint quantitative characterization will be reviewed (main methods are also summarized in Table 2.1 for keypoint detection and Table 2.2 for keypoint description).

2.2.1 Keypoint detection

In image understanding, the main approaches to identify keypoints are multi-scale corner and blob detection. A corner is a point which presents two dominant and different edge directions in its local neighborhood whereas a blob is a region that differs in brightness or color properties compared to other surrounding regions.
In general, localization of blob structures tends to be less accurate than corner detection. Indeed, corners can be identified by a single point while blobs can be only localized by their (often irregular) boundaries and the keypoint is approximately placed in the local extrema inside the blob/region. On the other hand, scale of a corner is ill-defined whereas a blob detector provides useful information about blob detected scale [165]. Even though corners and blobs can be considered as two sides of the same coin, different techniques have been investigate for their detection.
With reference to finding keypoints through blob detection, the main and qualified machine vision methods are based on partial differential equations (PDE). Among others, the main ones are Laplacian-of-Gaussians (LoG), Difference-of-Gaussians (DoG), Harris-Laplacian and Hessian-Laplace [113]. In LoG the image is convolved by a Gaussian kernel at a certain scale to give a scale-space representation and then a scale-normalized Laplacian operator is computed to infer the local maxima/minima whereas DoG represents a good and faster to compute approximation for LoG. Harris-Laplace combines the Harris operator with the scale selection mechanism presented in [102] while Hessian-Laplace detector usually returns more interest regions than Harris-Laplace at a slightly lower repeatability [69].
Beside techniques based on PDE analysis, in the literature we can find template-based blob detectors too. In such approaches, binary comparison and decision tree classification are carried out to investigate interest structures in the image. Among them we have ORB [136], BRISK [98] and FREAK [6] detectors and we will give notions about them in Subsection 2.2.2 since their main contribution is about keypoint description.
With reference to corners and accordingly to the taxonomy recently reported

in [99], corner detection methods can be grouped in three different categories: *gradient-based*, *template-based* and *contour-based* techniques.
Gradient-based methods have been the first in chronological order to be studied. Among them, we can find the very popular Harris corner extraction [73] based on the first order Taylor expansion of the second derivative of the local sum of squared differences (SSD). Furthermore, literature proposes some Harris-inspired methods with different cornerness measures such as the Shi-Tomasi [145] and the KLT detector [164]. In [145] a different interpretation of the eigenvalues based on a threshold is carried out while in [164] the proposed point is selected by analyzing how well the given feature can be tracked. Since the analysis proposed by such methods for gradient-based cornerness evaluation requires a considerable computational effort, different solutions have been introduced in [107] (LOCOCO) and [106] (S-LOCOCO) where integral images and interpolation are employed to reduce the computational load required for Gaussian derivative, cornerness response and non-maximum suppression.
Another way to detect corners in images is comparing the intensity of a given pixel and pixels in its neighborhood. Literature usually refers to such approach with the name of template-based corner detection, where the template is a mask to be centered on the central pixel in order to locate sorrounding pixels. The main approach falling in such category is SUSAN [152] where the intensity of each pixel inside the circular template is subtracted to the intensity of the central pixel and points with absolute intensity difference smaller than a given threshold are considered as corners. Similar template-based approaches have been recently proposed by involving machine learning and decision trees to speed-up the detection process. Among them, FAST [132, 133] represents currently the state-of-the-art. FAST concerns the use of a circular template: instead of computing intensity differences as SUSAN, a point is considered a corner by the FAST approach if and only if at least S contiguous pixels in the circle are darker or brighter than the intensity value of the central pixel and a threshold. To enhance repeatability, FAST has been improved by increasing the thickness of the template thus leading to the FAST-ER method [134]. Another derivation of FAST is the AGAST detector [109] where an adaptive and generic accelerated template-based test is carried out by finding the optimal decision tree in an extended configuration space.
A third more tricky category for corner detection is the previous mentioned contour-based which finds a wider applicability in shape analysis rather than keypoint matching, thus making them not so appropriate for our purpose. Methods falling

in this class try to rely on contour and find interest points by analyzing the maximum curvature in the planar edge curves. Therefore, after performing steps as curve smoothing and curvature estimation, the extracted corner points are mainly located in binary edge maps. Main contour-based approaches for curvature estimation are determinant of gradient correlation matrices [204], anisotropic directional derivatives representation [146], hyperbola fitting [187] and junction detection [193].

Detector type	**Based on**	**Proposed methods**
Corner detection	Gradient	Harris [73], KLT [164], Shi-Tomasi [145], LOCOCO [107], S-LOCOCO [106]
	Template	SUSAN [152], FAST [133], FAST-ER [134], AGAST [109]
	Contour	DoG-curve [204], ANDD [146], Hyperbola fitting [187], ACJ [193]
Blob detection	PDE	LoG, DoG, DoH, Hessian-Laplacian [113], SIFT [105], SURF [14]
	Template	ORB [136], BRISK [98], FREAK [6]

Table 2.1: Summary of the main keypoint detectors at the state-of-the-art.

2.2.2 Keypoint description

One of the best quality techniques nowadays available in literature for keypoint description is Scale-Invariant Feature Transform (SIFT) [105]. SIFT method first includes a rotation and scale invariant keypoint detection through Difference of Gaussians (DoG) (see Subsection 2.2.1). Then, a 128-components vector is built by computing a grid of histograms of oriented gradients. The vector is finally normalized in order to improve invariance to illumination and affine changes. Thanks to its structure, SIFT descriptor is characterized by a high discriminative power and robustness to changes of illumination and viewpoint. On the other hand SIFT is rather slow to compute and match, thus making it not so appropriate for real time applications.

To address real time flaws, another gradient-based descriptor has been introduced: its name is Speeded Up Robust Feature (SURF) [14]. SURF offers significantly lower computational time without affecting recognition accuracy if compared to SIFT. SURF performs keypoint detection by identifying image blobs through the computation of the Hessian matrix determinant while descriptor is obtained by summing Haar wavelet responses at the region of interest.

Different alternatives to decrease time cost of SIFT-based matching are based on dimensionality reduction in order to obtain shorter descriptors. This is done by employing Principal Component Analysis (PCA) or Local Discriminat Analysis (LDA) thus leading to SIFT-like approaches as PCA-SIFT [90] and Gradient Location and Orientation Histogram (GLOH) [112]. PCA-SIFT reduces descriptors from 128 to 36 dimensions whereas dimensionality reduction compromises descriptor distinctiveness. On the other hand, GLOH provides better distinctiveness and uses PCA to shorten descriptor at the price of greater processing time.

Attempts to introduce faster SIFT-like methods by employing dimensionality reduction, values quantization [28, 166, 188] or hashing [142, 156] did not succeed enough mainly because of the required computation of the full descriptor through floating-point calculation *before* doing post-processing optimization, thus having great impact on the spent time. Moreover, distance between floating-point vector is usually determined through the Euclidean distance which negatively affects the overall matching cost.

To deal with the problem of expensive floating-point computation, different approaches involving computation of compact binary strings from image patches have been investigated. The first noticeable contribution falling in such area is Binary Robust Independent Elementary Features (BRIEF) [29, 30]. Single bits of a BRIEF descriptors are obtained by comparing intensities of pixel belonging to the smoothed region of interest: such technique provides a remarkable speed-up not only in terms of descriptor building but also during matching, since the similarity measure between two binary vectors - *i.e.* the Hamming distance - can be quickly computed through a bitwise XOR followed by a bitcount. Furthermore, significant storage saving is favored by adopting binary vectors instead of floating-point structures.

Further approaches have been then examined in order to increase robustness to

scale and rotation of BRIEF without loosing in speed efficiency. Among others, Oriented FAST and Rotated BRIEF (ORB) [136], Binary Robust Invariant Scalable Keypoints (BRISK) [98] and Fast Retina Keypoints (FREAK) [6] are the main contributes. ORB is a fusion of FAST and BRIEF algorithms with peculiar modifications to enhance performance. Specifically, keypoint are detected by using FAST with the addition of accurate orientation component plus Harris corner filter [73] to perform non-maxima suppression of interest points. BRIEF descriptors are then steered according to the orientation of detected keypoints to provide tolerance to viewpoint changes. Similar idea is at the base of BRISK where a combination of AGAST [109] for keypoint detection and a DAISY-affine [163] pattern for description are employed whereas FREAK adopts a geometric pattern that recall the human retina.
As the reader can notice, recent attempts to provide discriminative and robust binary descriptors are mostly based on FAST and BRIEF as core detection and description algorithms respectively, plus encapsulation of scale and rotation information aiming to more viewpoint invariance. In our work we adopt FAST and BRIEF too, but instead of modifying the "nature" of such methods we attempt to discover the "strongest" descriptors by quantitatively characterizing their reliability through the saliency measure illustrated in Section 2.4.

Descriptor type	**Based on**	**Proposed methods**
Non-binary	Gradient	SIFT[105], SURF[14]
	Gradient and dim. reduction	PCA-SIFT [90], GLOH [112]
Binary	Patches/Template	BRIEF [29, 30], ORB [136], FREAK [6], BRISK [98]

Table 2.2: Summary of the main keypoint descriptors at the state-of-the-art.

2.2.3 Keypoint quantitative characterization

The quantitative characterization of keypoint, the definition of keypoint saliency and the effectiveness of local descriptors have been widely investigated in recent years. Usually, the definition of keypoint saliency takes into account different aspects such as robustness with respect to deformations or descriptor distinctiveness. For example, Amit and Geman [7] defined a training procedure to select special points which are likely to be found at certain places in the object but rarely in the background. However they did not consider the robustness of associated local descriptors. The concept of robustness of local object appearance represented as probability density function has been investigated by Fergus *et al.* [55] and Sim and Dudek [147]. Moreover, Pope and Lowe [128] attempted to give an estimation of descriptor detectability and distinctiveness by calculating how often it appears in the learning process. Another memory-based approach was proposed by R.C. Nelson [118] that relies on the combination of an associative memory with a Hough-like evidence technique. Ohba and Ikeuchi [122] proposed an eigen-based selection of robust descriptors according to their variation with respect to deformations and suggested selection of unique descriptors by checking their distinctiveness compared to other descriptors extracted from training images. In the approaches proposed by Agarwal and Roth [2] and Weber *et al.* [186] a clustering algorithm is adopted to select patterns most often appearing during the training stage. Dorkó and Schmid [139] trained a classifier for each object part and proposed a selection of scale invariant feature descriptors to determine the most discriminant ones, whereas Zhang and Kosecká [201] introduced a hierarchical approach where a refinement stage is adopted to select only the most discriminative SIFT features and a simple probabilistic model integrates the evidence from individual matches.

The previously cited approaches allow to effectively determine the most discriminant local appearances for a given object class. Nevertheless, most of them do not explicitly consider the intra-class saliency of local appearances in order to establish a ranking of them. A relevant work that quantitatively characterizes keypoint robustness has been proposed by Comer and Draper [43]: their approach tries to determine if a point is repeatable using a generalized linear model (GLM) which is able to predict which points will repeat according to 17 different attributes. The authors used different keypoint detectors such as Lowe's keypoint detector [105] and Harris-Affine keypoint detector [73, 111].

Differently from [43] our saliency analysis is not based only on the evaluation of keypoint detection response but it also considers associated local descriptor discriminating power. The literature contributions closest to our approach are the interesting (and inspiring) work introduced by Carneiro *et al.* [36, 37] and the classification-based prediction of local descriptors matchability recently introduced by Hartmann *et al.* [74]. Analogously to our approach, [37] and [36] adopted a training phase where geometric and brightness transformations are used to estimate keypoint/descriptor robustness and to define their saliency. Furthermore, as in [74], our aim is to find out in advance best candidates for matching. However, our approach deviates from [37], [36] and [74] in several directions illustrated below.

- We propose a simplified saliency definition and a different matching schema to boost efficiency and enable real time operation on low performance architectures; [37] and [74] use a keypoint classifier to filter out (also from the test image) unwanted keypoints and [37] uses a regressor to estimate probability values given in input to a keypoint-assignment validator. On the contrary we perform keypoint selection only during the model computation (training), and our approach relies on simple NN pairing followed by RANSAC consolidation.
- In [74] a training phase is carried out by learning the probability of a given local descriptors to exceed a matching threshold during NN search. On the contrary, our approach does not rely on matching probability against thresholds.
- We focus on recently introduced keypoint detection (FAST) and local descriptors (BRIEF) while experiments in [37] have been performed by using (more accurate but less efficient) SIFT (as well as [74]) and Phase-based local descriptor.
- In our definition of distinctiveness, with the aim of maximizing keypoint/descriptor inter-class diversity, we consider different keypoints of the same images, while in [37] "negative" examples are randomly selected from a separate set of images. We believe our choice is more effective to filter out from the object model those keypoints/descriptors of the object which are similar to each other, in order to reduce the probability of false assignments.

2.3 Keypoint detection and matching: approach overview

The proposed approach belongs to the field of object detection and matching by keypoint detection and local descriptors comparison. A full overview of the method is reported in Figure 2.1. A training set is composed by a single reference image I^{ref} of the object acquired in neutral viewpoint and lighting conditions and by a set of N generated images $I^1, I^2, ..., I^N$ which depict the same object or scene under different conditions. A generic transformed image I^l is obtained by applying a transformation $Transf_l$, belonging to the set of transformations $\mathcal{T}$, to the reference image as follows:

$$I^l = Transf_l(I^{ref}) \tag{2.1}$$

The nature of $Transf_l$ function is strongly related to the type of transformations characterizing the target application and could be a 2D homography, a 3D projection, a light changing function or a combination of the previous ones.

The training phase starts with the keypoints detection on the reference image I^{ref}. Each keypoint is then mapped on the transformed images (through the known $Transf_l$ functions), thus obtaining a set of reference keypoints and their projections. Keypoint descriptors are then computed for all keypoints and a global analysis is performed to rank the keypoints by saliency and select the m-best ones to create the object model. Matching a test image I^{test} against an object model is carried out by detecting the keypoints and computing descriptors from I^{test} and matching them against the model keypoints.

Considering that we are interested in real time detection/matching, recently proposed FAST and BRIEF have been adopted as detector and descriptor algorithms, respectively (see Section 2.2). FAST keypoint detection involves simple computations considering only the brightness of the surrounding pixels, whereas BRIEF binary descriptors can be easily extracted through straightforward brightness tests and efficiently matched by bitwise operators.

Saliency evaluation relies on the estimation of keypoint distinctiveness, repeatability and detectability properties:

- Distinctiveness quantifies the difference between a given keypoint descriptor and other keypoint descriptors of the same object. Note that distinctiveness

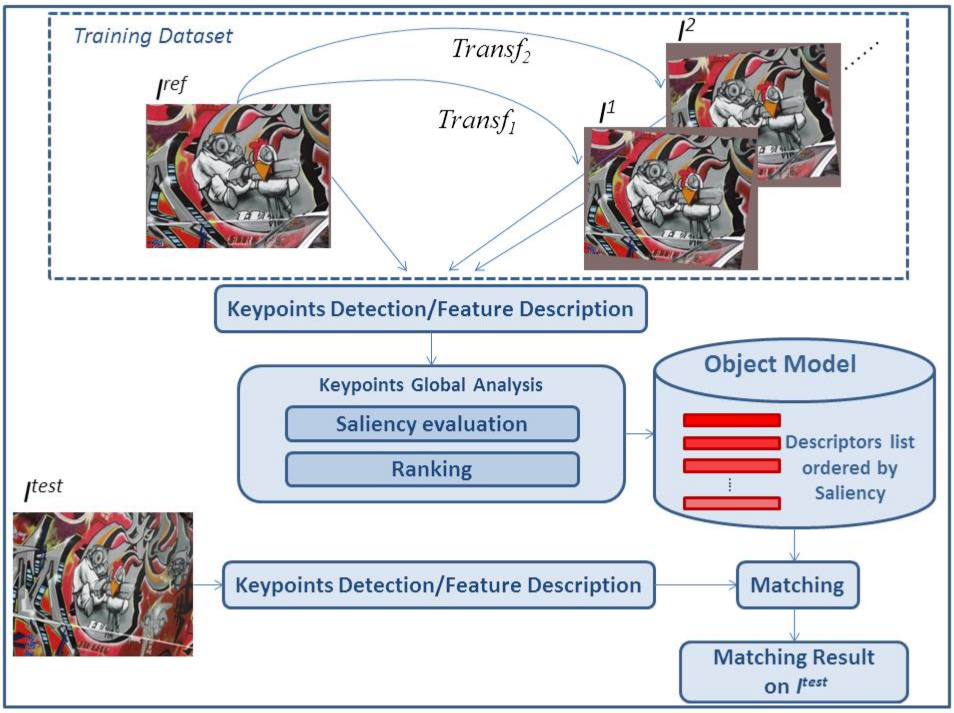


Figure 2.1: Overview of the proposed detection/matching method: a preliminary training stage is performed for saliency-based keypoint ranking and selection. The most salient keypoint descriptors which form the object model are the only keypoints matched against test images.

of a keypoint is strongly related to its local descriptor;

- Repeatability quantifies the difference between a given keypoint descriptor and corresponding descriptors of projected keypoint on transformed images. Hence, repeatability estimates invariance of local descriptor under different conditions (*e.g.* , viewpoint and lighting). Here too, the repeatability of a keypoint is strongly related to its local descriptor;

- Detectability quantifies the aptitude of a given keypoint to be detected under various viewpoint and lighting changes. Unlike distinctiveness and repeatability, detectability of a keypoint is based only on its detection properties (independently of the associated descriptor). For the FAST algorithm this is expressed by a score estimating the corner strength.

A highly distinctive, repeatable and detectable keypoint is an excellent candidate for the matching phase. On the contrary, a point with a low saliency is poorly representative and it could lead to false positive matches. Therefore, focusing on the most salient keypoints not only reduces the computation load but can also improve keypoint matching accuracy. As previously described, saliency evaluation exploits various images taken under different conditions of viewpoint and lighting. To overcome well-known problems of RGB color space when dealing with light changes, according to other authors ([64, 178, 179]) we propose to operate in the Opponent color space [178], defined as follows:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{2.2}$$

The intensity information is encoded by O_3 whereas the color information is represented by O_1 and O_2. Due to the subtraction in O_1 and O_2, such components are shift-invariant with respect to light intensity. Our experimental results confirm that Opponent color space (OCS in the following) is more effective than RGB space.

2.4 Definition of keypoint saliency

In this section we introduce a quantitative characterization of keypoints saliency in terms of repeatability, distinctiveness and detectability. We define with $\boldsymbol{x}_i = (u_i, v_i) \in I^{ref}$ a keypoint selected by the detection algorithm d (in our case the FAST approach) and with s_i the keypoint strength (*i.e.* , amount of cornerness) computed by the algorithm d itself. The set of all keypoints of the reference image $K_d(I^{ref})$ is then:

$$K_d(I^{ref}) = \{(\boldsymbol{x}_i, s_i) : \boldsymbol{x}_i \in I^{ref}, i = 1, \ldots, J\} \tag{2.3}$$

For each $\boldsymbol{x}_i$ we define with $descr : (\mathfrak{R}^2, \mathfrak{R}^S \times \mathfrak{R}^S) \Rightarrow \mathfrak{R}^L$ the function that computes BRIEF descriptor for a keypoint $\boldsymbol{x}_i$ according to the image patch $P(\boldsymbol{x}_i)$ of size $S \times S$ centered on $\boldsymbol{x}_i$.
Given the nature of BRIEF, $\boldsymbol{b}_i = descr(\boldsymbol{x}_i, P(\boldsymbol{x}_i))$ is a binary vector. Therefore, two binary vectors $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ are compared by using the Hamming distance $H(\boldsymbol{b}_i, \boldsymbol{b}_j)$ that can be computed very efficiently through a bitwise XOR operation followed by a bit count.

2.4.1 Distinctiveness

The distinctiveness $D(\boldsymbol{x}_i)$ of a keypoint $\boldsymbol{x}_i \in K_d(I^{ref})$ is proportional to the diversity among the descriptor of $\boldsymbol{x}_i$ and the descriptors of other keypoints $\boldsymbol{x}_j \in K_d(I^{ref})$, $j \neq i$ in the same image. Formally, we estimate the distinctiveness as follows:

$$D(\boldsymbol{x}_i) = \frac{1}{L \cdot (\#K_d(I^{ref}) - 1)} \sum_{\substack{\boldsymbol{x}_j \in K_d(I^{ref}) \\ j \neq i}} H(\boldsymbol{b}_i, \boldsymbol{b}_j) \tag{2.4}$$

being L the descriptor length and $\#K_d(I^{ref}) > 1$ the total number of keypoints detected on I^{ref}. $D(\boldsymbol{x}_i)$ takes a value in the range $[0, 1]$, where 1 denotes maximum distinctiveness.

2.4.2 Repeatability

The repeatability of a keypoint $\boldsymbol{x}_i \in K_d(I^{ref})$ is proportional to the similarity among its descriptor $\boldsymbol{b}_i$ and the descriptors of corresponding keypoints under a set of given transformations. It is defined as follows:

$$R(\boldsymbol{x}_i) = 1 - \frac{1}{L \cdot \#\mathcal{T}} \sum_{\substack{\boldsymbol{x}_i^l = Transf_l(\boldsymbol{x}_i) \\ Transf_l \in \mathcal{T}}} H(\boldsymbol{b}_i, \boldsymbol{b}_i^l) \tag{2.5}$$

being $\#\mathcal{T}$ the cardinality of the set $\mathcal{T}$ and $\boldsymbol{b}_i^l = descr(\boldsymbol{x}_i^l, P(\boldsymbol{x}_i^l))$. $R(\boldsymbol{x}_i)$ takes a value close to 1 when a keypoint is highly repeatable.

2.4.3 Detectability

The detectability of a keypoint depends of the score values returned by the keypoint detection algorithm. If detection is performed with FAST, the score is the corner strength. The detectability of a keypoint $\boldsymbol{x}_i \in K_d(I^{ref})$ is simply an average (normalized in the range $[0, 1]$) over the scores of all keypoints in the original image and its transformed versions:

$$F(\boldsymbol{x}_i) = \frac{1}{\#\mathcal{T}} \sum_{\substack{\boldsymbol{x}_i^l = Transf_l(\boldsymbol{x}_i) \\ Transf_l \in \mathcal{T}}} s_i^l \tag{2.6}$$

where s_i^l is the strength score related to $\boldsymbol{x}_i^l$ and returned by a detection algorithm.

2.4.4 Saliency

Detectability, distinctiveness and repeatability are combined in order to determine the keypoint saliency, as shown in Figure 2.2. More precisely, for each keypoint $\boldsymbol{x}_i \in K_d(I^{ref})$, its saliency is:

$$S(\boldsymbol{x}_i) = \omega_R R(\boldsymbol{x}_i) + \omega_D D(\boldsymbol{x}_i) + \omega_F F(\boldsymbol{x}_i) \tag{2.7}$$

where ω_R, ω_D and ω_F are weights assigned to repeatability, distinctiveness and detectability, respectively. Optimal values for the weights can be computed by trial and error during experimentations by using a validation set. It is worth noting

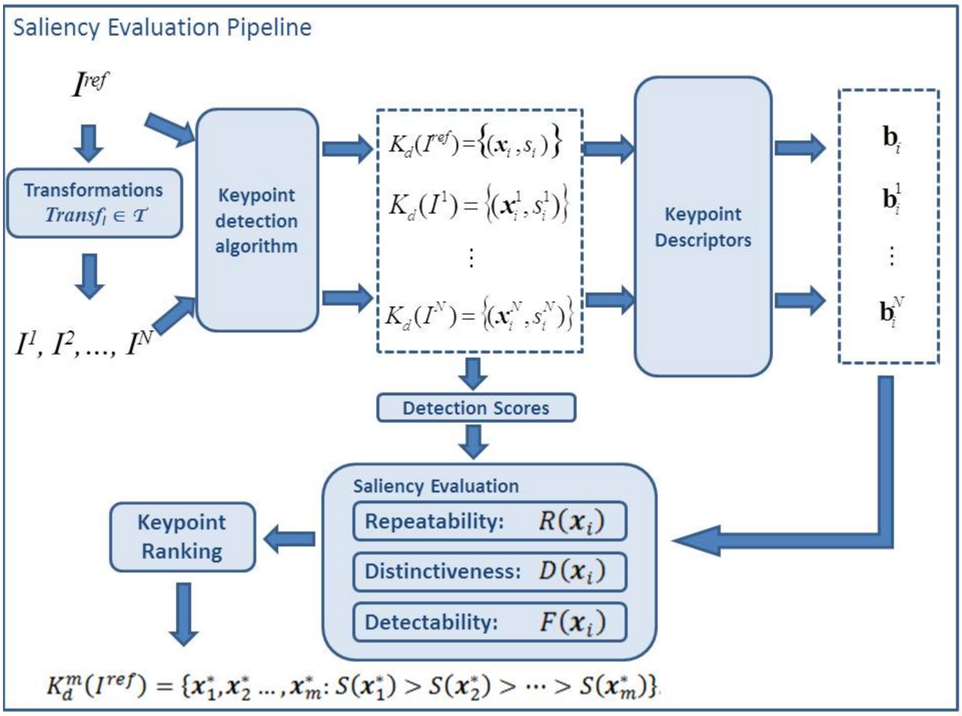


Figure 2.2: Detectability, distinctiveness and repeatability are combined in order to obtain a final value of saliency for each keypoint.

that even if detectability and repeatability can be quite correlated, both the contributions are important: the former is crucial to maximize the probability that the detection algorithm will find the keypoint of interest in new images, the latter to maximize the probability that two corresponding keypoints match under the metrics associated to the chosen descriptors. In other words, while detectability is related to keypoint stability under transformation, repeatability and distinctiveness are related to the discriminant power of descriptors. The keypoints of the image can be ranked according to saliency, resulting in the following ordered set $K_d^*(I^{ref})$:

$$K_d^*(I^{ref}) = \{\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_{\#K_d(I^{ref})}^* : S(\boldsymbol{x}_1^*) \geq \boldsymbol{x}_2^* \geq \ldots \geq S(\boldsymbol{x}_{\#K_d(I^{ref})}^*)\} \quad (2.8)$$

Equivalently, we define the m most salient points as:

$$K_d^m(I^{ref}) = \{\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_m^* : S(\boldsymbol{x}_1^*) \geq \boldsymbol{x}_2^* \geq \ldots \geq S(\boldsymbol{x}_m^*)\} \quad (2.9)$$

The best choice for the value m is related to the specific application in order to achieve at the same time good matching accuracy and speed, as properly shown in Section 2.5.

2.5 Experimental evaluation

Many experiments have been carried out on various real images acquired under different conditions of viewpoint and lighting. In particular:

- a first round of experiments has been carried out on small datasets with the aim of tuning parameters and selecting the most effective color space;
- a second round of experiments is then performed (without further adjusting parameters) on a larger dataset to validate results;
- our approach is then compared with other state-of-the-art techniques including the native ranking of FAST detector and the ranking method proposed in [74]. A direct comparison with the approach by Carneiro *et al.* [37] would be interesting too, but the authors report their results on proprietary datasets (currently not available) and the re-implementation of their method is quite complex and the risk of an inexact implementation is high.

2.5.1 Dataset and evaluation criterion

For the first round of experiments we focus our evaluation on public and commonly used datasets (see Figure 2.3). In particular, Wall and Graffiti datasets

([30, 112, 113, 177]) are typical benchmarks to evaluate keypoint robustness against viewpoint changes while Book dataset ([183, 183]) is employed to analyze keypoint robustness against lighting changes.

For the second round of experiments we use the recently introduced BigBIRD [149] dataset, which contains a larger amount of objects acquired under different viewpoints and lighting conditions. In the following we briefly describe these datasets and the associated ground truth information.

Wall and Graffiti datasets (see Figure 2.3) consist of 6 images, where the first one is acquired under standard conditions and the remaining 5 images are acquired under different viewpoints (with increasing variation). According to other works, the first image of each dataset has been used as reference image whereas the remaining 5 images have been considered for matching. For each transformed image the ground truth homography matrix [177] corresponding to the transformation with respect to the reference image is given, thus allowing the ground truth keypoint correspondence to be easily computed. When working with Graffiti and Wall datasets, for each reference image we generated 80 artificial transformations to be used for the training phase: the variations considered are random homographic transformations within predefined parameter ranges.

Book dataset has been selected from Phos database [172] and contains 45 images taken under different conditions of natural light (overexposure, underexposure, directional).

The first image acquired in standard conditions (normal exposure) has been considered as reference image. Unlike for Wall and Graffiti, here generating artificial light changes (including shadows) for training is complex and could lead to produce unrealistic variations. Therefore the 44 images have been split in two sets: the first set (29 images) has been used for training, while the remaining 15 images for testing.

BigBIRD dataset [149] consists of 125 objects each of them acquired under different poses by varying the camera angle and the rotation plan. We selected a total of 100 objects by leaving out objects with very poor or no texture information. For each object we selected one frontal reference image plus 5 rotated images with increasing rotation angle. As described for Wall and Graffiti, also for each BigBIRD object we exploit ground truth homography matrices to map points between reference and rotated images. Moreover, for our training we use the same number of artificial transformations (80).

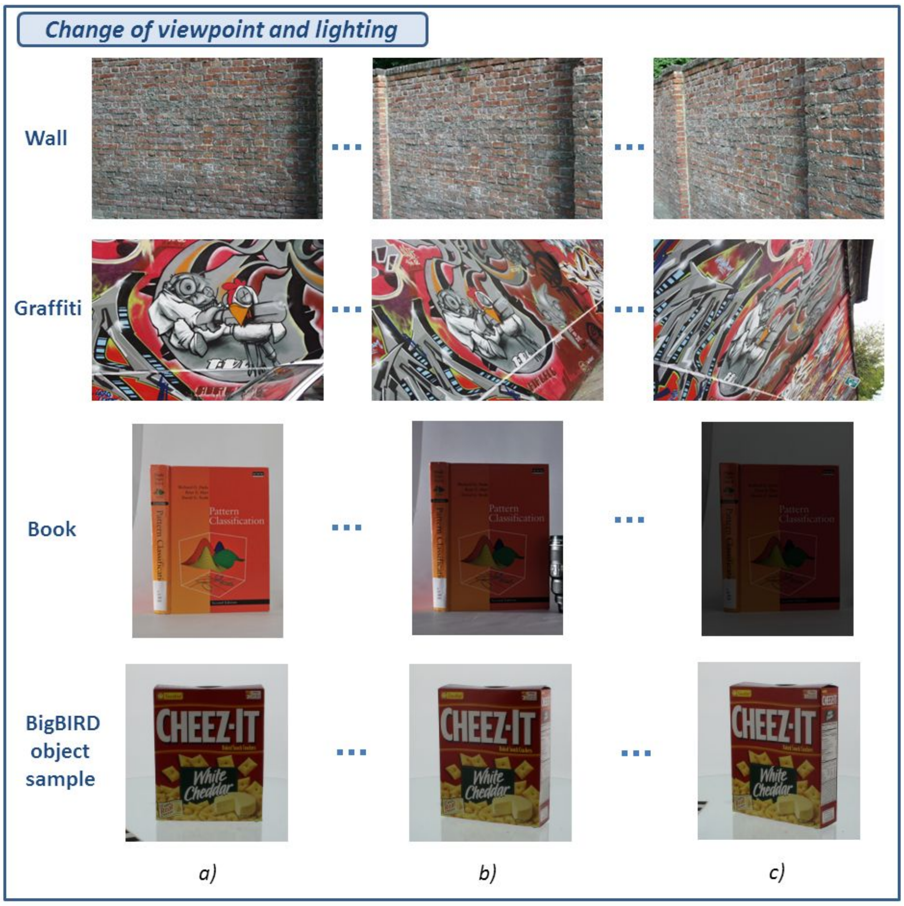


Figure 2.3: Datasets used for matching accuracy evaluation under different conditions of viewpoint and lighting: Wall, Graffiti, Book and BigBIRD object example. Column shows for each dataset: a) the reference images, b) an image with moderate variation, c) an image with high variation.

For each dataset, a training phase has been carried out to detect and rank keypoints for each reference image I^{ref} and the corresponding transformations. According to the previously introduced notation, the m most salient keypoints are denoted by $K_d^m(I^{ref})$.

Then for each test image I^{test} we extract all keypoints $K_d(I^{test})$ and we evaluate the accuracy of our detection/matching approach based on average Hamming distance and recall.

The average Hamming distance is defined as:

$$H^{avg}(I^{test}) = \frac{1}{m} \sum_{\boldsymbol{x}_i \in K_d^m(I^{ref})} H(\boldsymbol{b}_i, \tilde{\boldsymbol{b}}_i) \tag{2.10}$$

where $\tilde{\boldsymbol{x}}_i = Trasf_{test}(\boldsymbol{x}_i)$ is the position corresponding to $\boldsymbol{x}_i$ in the test image according to the known transformation binding I^{test} to I^{ref} and $\tilde{\boldsymbol{b}}_i = descr(\tilde{\boldsymbol{x}}_i, P(\tilde{\boldsymbol{x}}_i))$.

The recall metrics is computed pairing $K_d^m(I^{ref})$ and $K_d(I^{test})$ by nearest-neighbor

Hamming distance and by evaluating the amount of correct pairing. We associate each $\boldsymbol{x}_i \in K_d^m(I^{ref})$ to the keypoint $\boldsymbol{y}_j \in K_d(I^{test})$ such that:

$$H(\boldsymbol{b}_i, \hat{\boldsymbol{b}}_j) = \min_{\boldsymbol{y}_k \in K_d(I^{test})} H(\boldsymbol{b}_i, \hat{\boldsymbol{b}}_k) \tag{2.11}$$

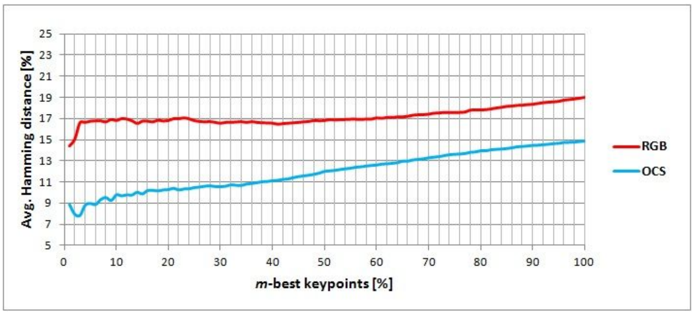
being $\hat{\boldsymbol{b}}_j = descr(\boldsymbol{y}_j, P(\boldsymbol{y}_j))$ the BRIEF descriptor of $\boldsymbol{y}_j$.
Since the mapping function between reference and test images is known we can distinguish between correct and false matches. In particular, according to Mikolajczyk and Schmid [112], a match is considered correct if the spatial overlap between the regions covered by the two keypoint descriptors is larger than a given threshold ε_s. Therefore, recall is defined as:

$$recall = \frac{\#\ correct\ matches}{m} \tag{2.12}$$

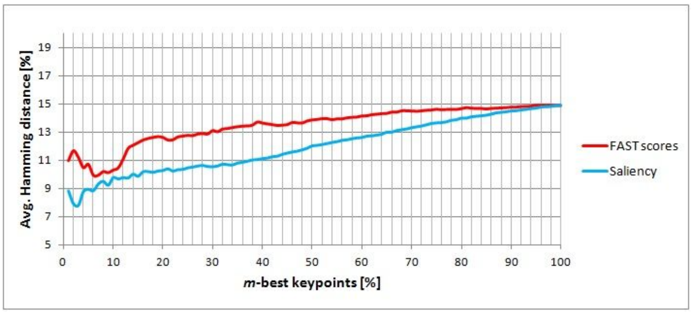
It is worth noting that the two metrics used, even if related, consider different aspects. In fact, when computing the average Hamming distance we do not extract keypoint by FAST approach on I^{test} but we compute the descriptors at the positions corresponding to the projection of I^{ref} keypoints; so this metrics highlights the descriptor matching but discounts potential errors due to keypoint detection. On the contrary, when estimating recall, we compute correspondences by nearest neighbor between the two sets of keypoints (both detected by FAST) and this takes into account false detection and false pairing of keypoints.
Tests have been repeated for different values of m: from 1% to 100% of the total number of keypoints with step 1%. This allows to evaluate the effect of selecting different amounts of keypoints.

2.5.2 Color spaces: OCS vs RGB

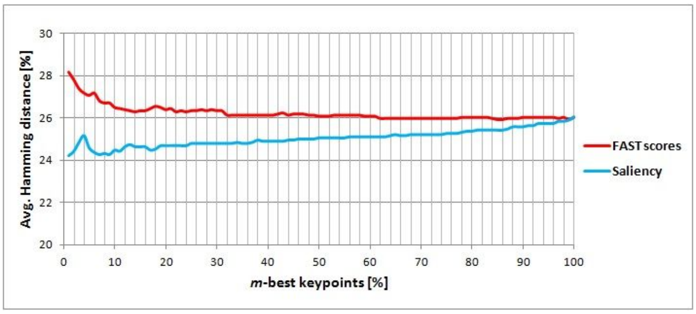
To evaluate the pros/cons of OCS, we used the Book dataset. Figure 2.4 a) shows the Hamming distance reported as percentage with respect to the descriptor length. For both RGB and OCS the increasing trend of the curves proves that the most salient keypoints are the most stable with respect to image variations. As expected, results confirm that by using OCS instead of RGB, BRIEF descriptors result more similar and therefore more invariant to light changes. The good performance obtained by using OCS induced us to use such color space for the rest of the experiments.



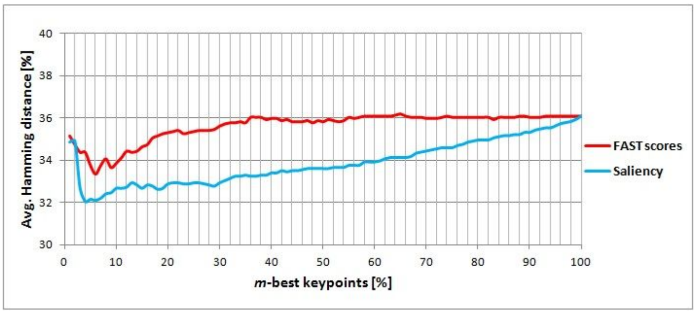
(a)



(b)



(c)



(d)

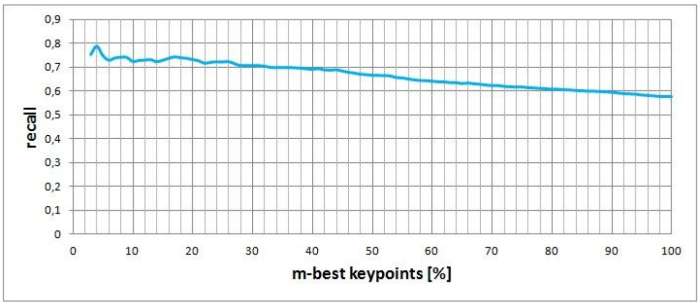
Figure 2.4: Average percentage Hamming distance by varying the percentage of keypoints; (a) the graph refers to average values over all reference-test pairs of Book dataset when using RGB and OCS space; (b) (c) (d) the graphs refer to average values over all reference-test pairs of Book, Wall and Graffiti datasets when using our saliency-based ranking and FAST score ranking (b) (c) (d).

2.5.3 Saliency evaluation: effectiveness of keypoint selection

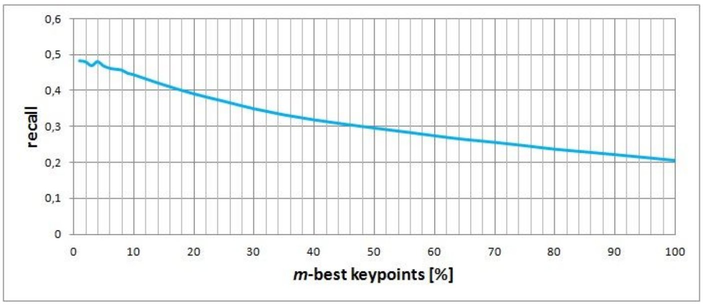
In this set of experiments we compare our saliency-based ranking with respect to FAST score-based ranking (according to the notation used, FAST scores are the s_i introduced in Section 2.4). Results obtained on Wall, Graffiti and Book datasets, reported in Figure 2.4 b), c) and d) respectively, show an increasing trend of the Hamming distances thus proving that the most salient keypoints (those in the first positions of the ranking) are more likely to match under the image variations. Ranking keypoints according to their FAST scores leads to a somewhat analogous trend but in this case the curve is more flat and, especially for small values of m, the reduction in the average Hamming distance is less relevant. Of course, when 100% keypoints are used, ranking is irrelevant and the two curves converge. The preliminary results on Wall, Graffiti and Book allowed us to define optimal values for the basic parameters: $\omega_R = 1$, $\omega_D = 1$ and $\omega_F = 2$. For the rest of experiments we keep these values fixed.

The same type of experiment has been repeated for 100 BigBIRD objects (see Figure 2.6). Here too we note that our ranking is more effective than FAST-score, and we can observe (for our approach) an increased trend of the distances after an initial decrease. It is worth noting that BigBIRD objects typically consist of small objects (*i.e.* , canned food) with respect to Wall, Graffiti and Book images and the number of relevant keypoint is much smaller.

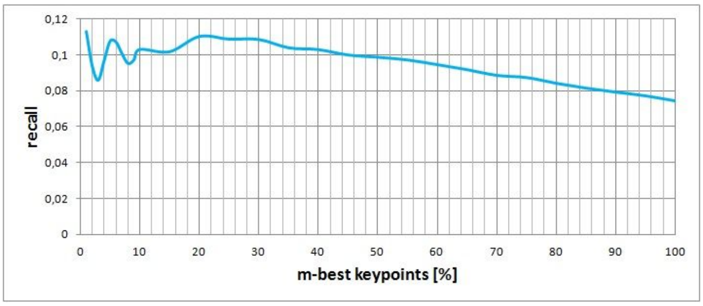
Figure 2.5 shows the recall values on Book, Wall and Graffiti datasets. Here the decreasing trend proves that the most salient descriptors are detectable with more stability thus leading to a lower number of false matches. For example, for Wall dataset, working with the 10%-most salient keypoints results in a recall of about 45%, while using the whole set of keypoint reduces the recall to about 20%. It is worth noting that the recall values are quite different for the three datasets and in particular are higher for Book and lower for Graffiti. This is due to the relative difficulty of the dataset. In particular, Graffiti confirms to be challenging for BRIEF descriptors, leading to low average recall. In [30] a higher recall is reported on this dataset. However, the comparison of our result with [30] is misleading, since in [30] the recall value has been computed by pairing keypoints according to ground truth data.



(a)



(b)



(c)

Figure 2.5: Recall value averaged over all reference-test pairs for Book a) Wall b) and Graffiti c) datasets, plotted as a function of keypoints percentage.

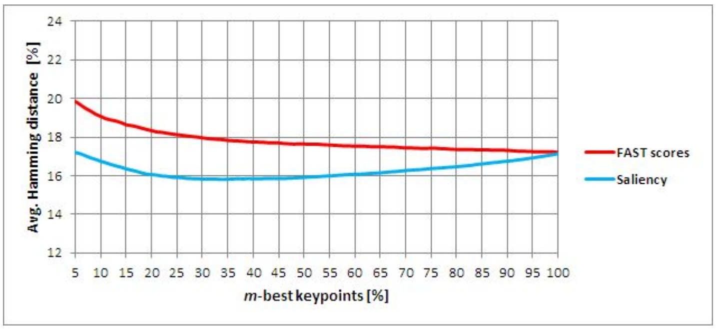


Figure 2.6: Average percentage Hamming distance by varying the percentage of keypoints on 100 BigBIRD objects: the graph refers to average values over all reference-test pairs when using our saliency-based ranking and FAST score ranking.

2.5.4 Saliency evaluation: comparison with the state-of-the-art

In this set of experiments we compare our saliency-based ranking against the FAST score-based ranking and against the ranking method proposed by Hartmann *et al.* in [74] for matchability prediction. In particular, they train a random forest classifier able to return a list of keypoints ordered by matchability score of SIFT descriptors. We use such classifier to produce a list of keypoints for each BigBIRD object. Then, we compare the average Hamming distances of BRIEF descriptors computed for the matchability-based ordered list of keypoints with our saliency-based and FAST score-based ordered lists. To avoid implementation differences for [74] we used the code kindly made available by the authors. Results are reported in Figure 2.7 and show that keypoints ranking according to our saliency yields a more stable set of descriptors.

As the reader can argue, curves in Figure 2.7 are not convergent and trends of

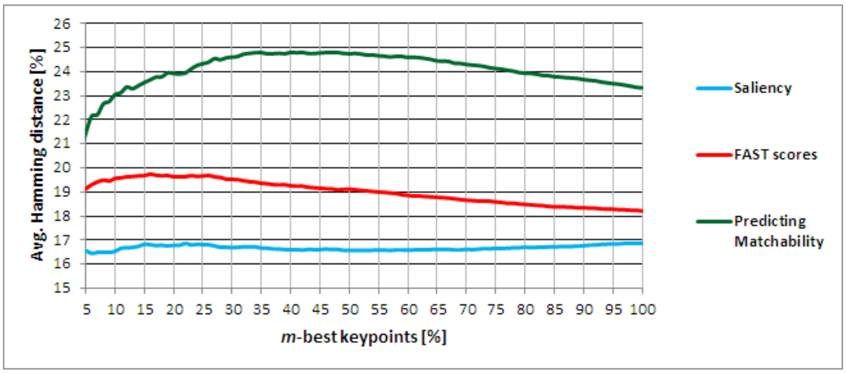


Figure 2.7: Comparison with state-of-the-art: the graph refers to Hamming average distance values over all reference-test pairs of BigBIRD objects when using our saliency-based ranking, FAST score ranking and the very recently proposed ranking based on ”matchability” properties of keypoints.

light blue and red curves are not the same as Figure 2.6. This is due to different reasons: on one hand, [74] was originally designed to operate with SIFT detector thus leading to a different set of keypoints with respect to FAST. On the other hand in this experiment we were forced to truncate the saliency-based and FAST score-based lists in order to make them numerically comparable with the matchability-based list of [74], thus eliminating the tails of the light blue and red curves.

2.5.5 Saliency evaluation: average processing time

In this section we briefly illustrate time performance of the training stage of our approach. For efficiency on test images please refer to Section 2.6.
To carry out the training phase, the following operations are required: FAST detection, BRIEF computation and evaluation of detectability, repeatability and distinctiveness on 80 different transformed images. Considering that each BigBIRD object has an average of 750 detected FAST keypoints, the required average processing time on a workstation Intel i7-2720QM 2.20 GHz with 8 GB of RAM to complete the training phase of a single object is $\sim$ 23 seconds.

2.6 Pose estimation in AR with salient keypoints

2.6.1 A first case study: maintenance application

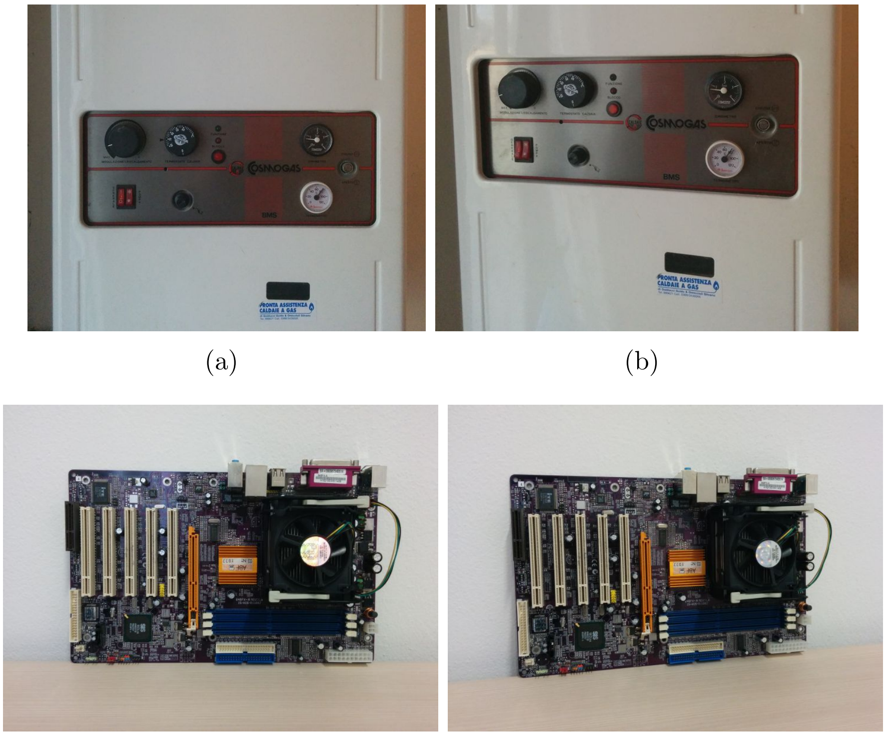
In this section we apply the proposed saliency evaluation to real time pose estimation for vision-based Augmented Reality, with a special attention to maintenance application. In particular, we are interested in detecting "natural" object markers in order to estimate the pose of the object with respect to the camera in order to superimpose information of interest to the captured image. In this section we consider as case of study the AR-based maintenance task of two elements with substantial differences in local appearances and textures: a water heater and a PC motherboard (see Figure 2.8).
In the above mentioned scenarios the user is expected to interact with a mobile device (*e.g.* , tablet or smartphone) whose camera takes live pictures of the object and useful pictorial information is superimposed (at the proper location) to guide maintenance. Since the viewpoint changes are moderate, our approach finds here an ideal application. In particular, the object of interest (*i.e.* , the front panel of a water heater or a motherboard) can be considered as a simple planar object and therefore estimating its pose is equivalent to compute a homographic transformation.
Given a set of keypoint correspondences we can extract the homography matrix through the RANSAC algorithm [57]. We compare three cases:

(A) all FAST keypoints are considered;

(B) only 15% m-best keypoints ranked according to FAST score are considered;

(C) only 15% m-best keypoints ranked according to our saliency-based approach are considered.

In all the above cases, (initial) keypoint correspondences are found by nearest neighbor. In the C case a single reference image is used to generate 80 synthetic viewpoint changes and produce keypoints ranking. Figure 2.10 and Figure 2.11 compare the results on different test images.



(a) (b)

(c) (d)

Figure 2.8: Reference images (a and c) and a test images (b and d) of a water heater and a PC motherboard.

Although RANSAC is somewhat robust against outliers, the advantages of using only robust keypoints is here evident in terms of precision of the recovered viewpoint transformation. We also note how our saliency-based ranking leads to consolidate a higher number of inliers and therefore a better viewpoint estimation with respect to the FAST-score based selection.
We repeated the previous test by random selecting 100 frames from a video taken by moving a tablet in front of the objects. The ground truth pose (for evaluation purposes) has been obtained by manually labeling the four panel corners. Then,

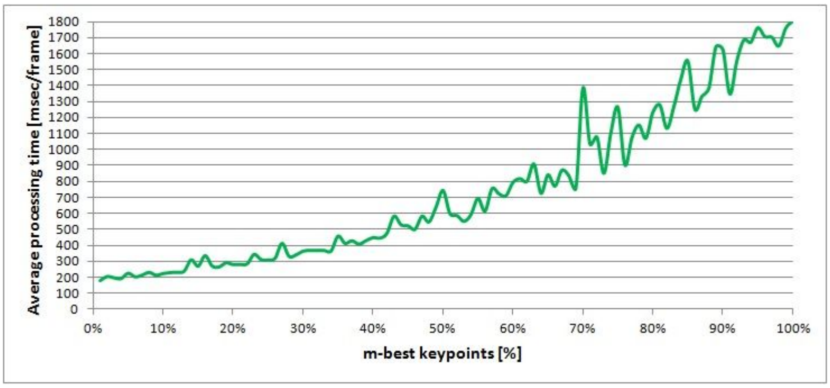


Figure 2.9: Average processing time (milliseconds) required for frame analysis and pose estimation.

automatic pose recovery has been carried out by approaches A, B and C, respectively. The resulting accuracy, here quantified in terms of average corner distance with respect to the ground truth position, increases when using only the most salient keypoints. In particular, with respect to case A: the average distance is 32% lower in B and 56% lower in C for the water heater and 7% lower in B and 50% lower in C for the motherboard.
In terms of computation, the processing time needed to estimate pose for a single frame can be split in: *i*) FAST detection; *ii*) RGB to OCS conversion; *iii*) BRIEF descriptors computation; *iv*) Keypoints matching; *v*) RANSAC homography estimation. The stages whose computing time depends on the number of keypoints are *iv)* and *v)*. Figure 2.9 shows the average processing time for a single frame analysis as function of the keypoints percentage. For this experiment we used a tablet device: Samsung ATIV Smart PC (Intel Atom Processor Z2760 1.5 Ghz). Even if the code (written in C# for .NET) was not highly optimized, by selecting the 10%-best keypoints we can process about 5 frames per second. We are confident that with proper optimization we can significantly improve the frame rate. On the contrary, for this application, if all keypoints are used the efficiency would be about one order of magnitude worse.

2.6.2 A second case study: smart museum tour

As discussed in the introductive chapter, the growth of mobile devices equipped with high quality displays, high resolution cameras and high processing capabilities allows new computer vision applications to be deployed. An example of such new possible application is represented by tourism, arts and intelligent buildings where

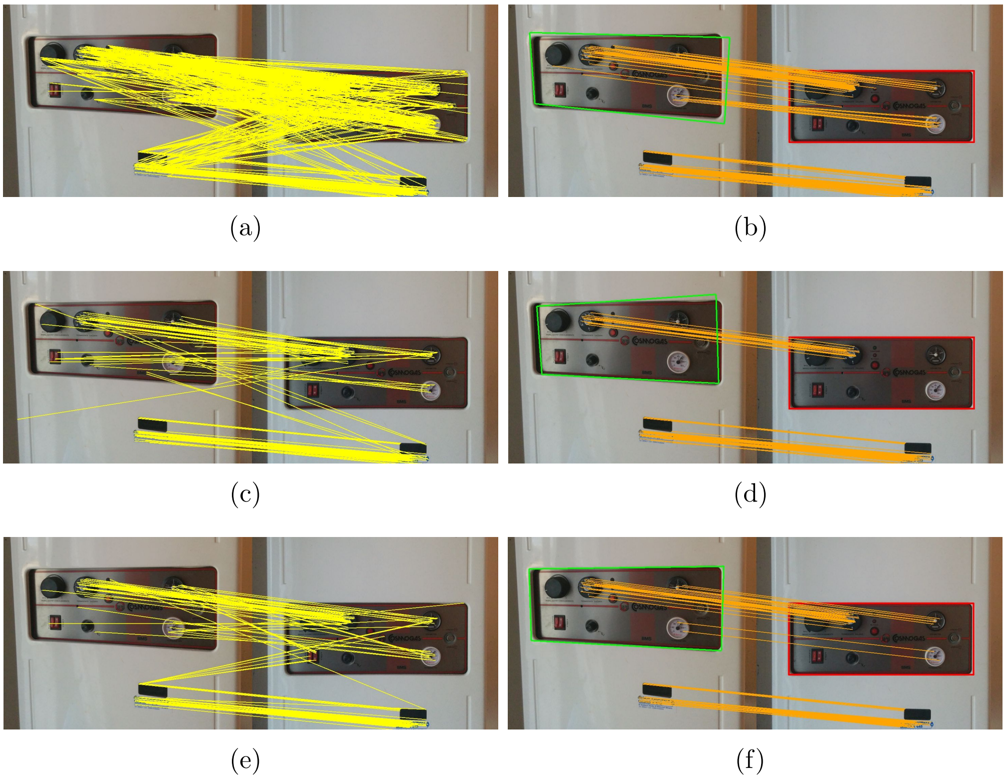
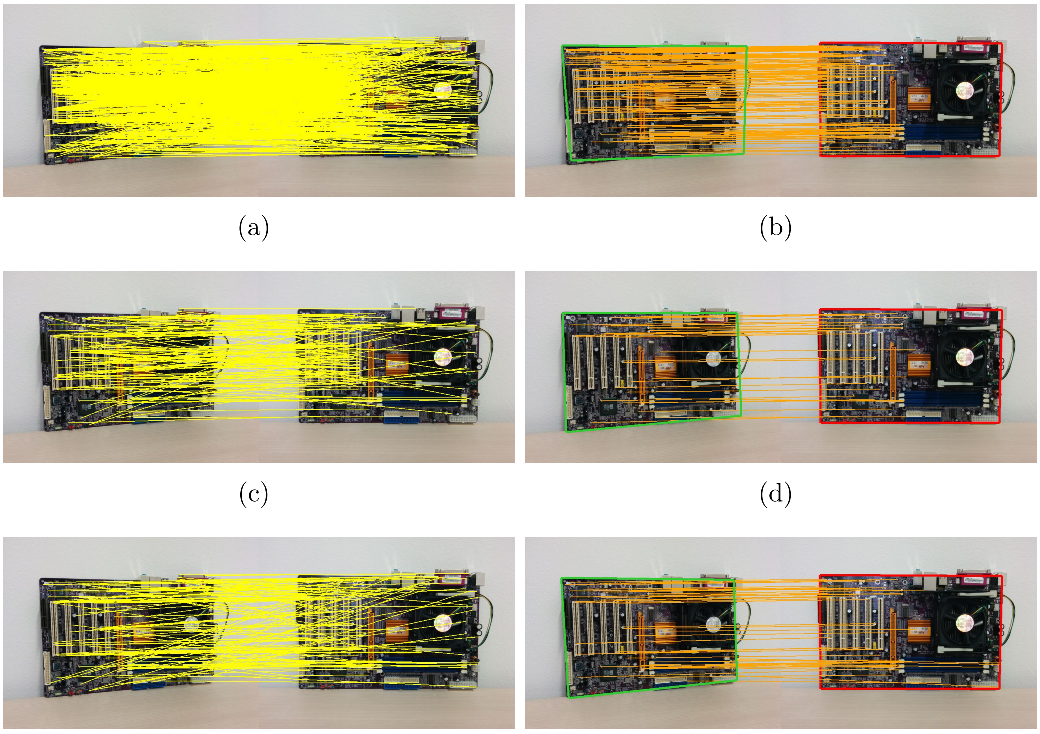


Figure 2.10: Water heater transformation recovery through RANSAC algorithm by taking as input: b) all FAST keypoints; d) 15% m-best keypoints ranked according to FAST score, f) 15% m-best keypoints ranked according to our saliency-based approach. Yellow segments a), c) and e) denote initial keypoint pairing and orange segments b), d) and f) final RANSAC inliers; the green rectangle denotes the homographic transformation inferred by RANSAC.

user-experience could be enhanced through interactive content and Augmented Reality. In this section we design a specific application to perform an augmented museum tour and we extend and apply our saliency framework by including an object recognition phase to be carried out before pose estimation.

A number of AR solutions in the field of cultural heritage and mobile multimedia guides have been recently proposed [35, 48, 49, 115]. Authors of [115] and [48] introduced an exhaustive overview of the main challenges related to conception, implementation, testing and assessment of a smart museum. In PALM-Cities Project [35] technologies such as NFC and QR Codes have been adopted to handle the interaction with the user whereas in [115] an hybrid approach based on markerless tracking plus a rotation sensor is used to allow free movements of the user mobile device.

Here we extend the approach illustrated in Section 2.3 with a (pre)matching phase which is applied to the painting recognition and pose estimation. Similarly to



(a) (b) (c) (d) (e) (f)

Figure 2.11: PC motherboard transformation recovery through RANSAC algorithm by taking as input: b) all FAST keypoints; d) 15% m-best keypoints ranked according to FAST score, f) 15% m-best keypoints ranked according to our saliency-based approach. Yellow segments a), c) and e) denote initial keypoint pairing and orange segments b), d) and f) final RANSAC inliers; the green rectangle denotes the homographic transformation inferred by RANSAC.

[115], in our application the user is expected to enjoy paintings in a markerless environment by interacting with a mobile device (*e.g.* tablet, smartphone or smart glasses) provided with a camera that captures videos of paintings under different conditions (*i.e.* , moderate changes of viewpoint and lighting).

Once a painting has been recognized and its pose has been retrieved the application can properly superimpose to the live camera view useful pictorial or textual information concerning the painting itself (see Figure 2.12 for an example).

An overview of the approach is presented in Figure 2.13: during the training phase we use a single reference image of each painting to compute the painting model including only the most salient keypoints. Models are then stored in a database which is made available to the user's mobile device.

In this study we consider ten famous paintings (see Figure 2.14). For each painting p we downloaded the reference image I_p^{ref} from the web and printed it on paper (A3 format). Paintings were then hanged to the walls of our lab to simulate

Figure 2.12: Example of augmented information geometrically coherent with the painting pose.

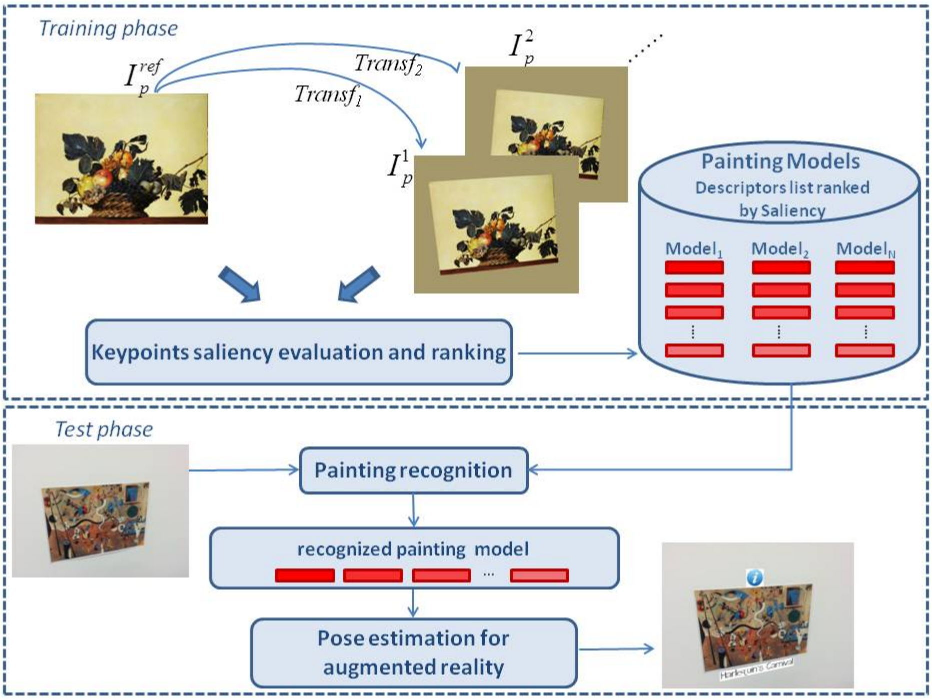


Figure 2.13: Overview of the proposed application based on keypoint saliency evaluation.

Figure 2.14: The ten famous paintings we consider in our study.

a museum room.
For each reference image, we generated 80 artificial transformations to be used for the training phase: the variations considered are random homographic transformations within predefined parameter ranges.
Test was performed using a smart device and capturing videos of each printed painting while moving in front of the painting. For each video we selected 30 frames characterized by different lighting and pose conditions, hence our test set is composed of 300 images (see Figure 2.15 for some examples).
We performed two different experiments: the former to evaluate recognition accuracy and the latter to evaluate the correctness of the estimated pose. In both these experiments our saliency-based ranking is compared to a standard FAST score-based ranking. For each test image I^{test}, the painting recognition phase is implemented as follows:

- all keypoints are extracted (FAST) and their local descriptors (BRIEF) computed;
- for each painting model I_p^{ref}, characterized by its m-most salient keypoints $K_d^m(I_p^{ref})$:
  - we associate each keypoint in $K_d^m(I_p^{ref})$ to the keypoint in I^{test} with smallest Hamming distance (between BRIEF descriptors);
  - we enforce geometrical constraints among keypoint correspondences using RANSAC [57] to filter out outliers;
  - the set of inliers returned by RANSAC is then used to compute a similarity score Φ between I^{test} and I_p^{ref} as follows:

$$\Phi(I^{test}, I_p^{ref}) = \frac{\#\ Ransac\ Inliers}{\# K_d^m(I_p^{ref})} \tag{2.13}$$

- finally, recognition is performed according to maximum similarity.

Tests have been repeated for $n = 300$ and for different values of m ranging from 1% to 100% of the total number of keypoints. This allows to evaluate the effect on recognition of progressive reduction of the keypoint number.
In Table 2.3 we show the recognition rate obtained by considering only the most salient descriptors when our saliency-based ranking and a standard FAST-scores ranking are applied. In general we can observe that our method is more effective

(a) (b) (c)

(d) (e) (f)

Figure 2.15: Samples of the dataset used in our case study: (a) and (d) are the reference images of two different paintings, (b), (c), (e), (f) are test frames acquired live with a tablet.

than the FAST-scores based one. Moreover it turns out that by applying our saliency evaluation a lower percentage of keypoints is sufficient to reach top performance with respect to FAST-scores ranking (only 4% of the keypoints for our approach versus 15% for FAST-scores). A second evaluation has been carried out

***m*-most salient keypoints[%]**	**"Saliency" recognition rate[%]**	**"FAST" recognition rate[%]**
1	74.58	53.75
2	92.5	80.42
3	97.92	90.83
4	98.75	90
5	**98.75**	94.17
10	97.92	95.83
15	96.67	**97.92**
20	97.08	97.92
100	88.75	88.75

Table 2.3: Recognition results for keypoints selection based on saliency and FAST-scores.

to assess correctness and computational load of pose estimation. Since a painting

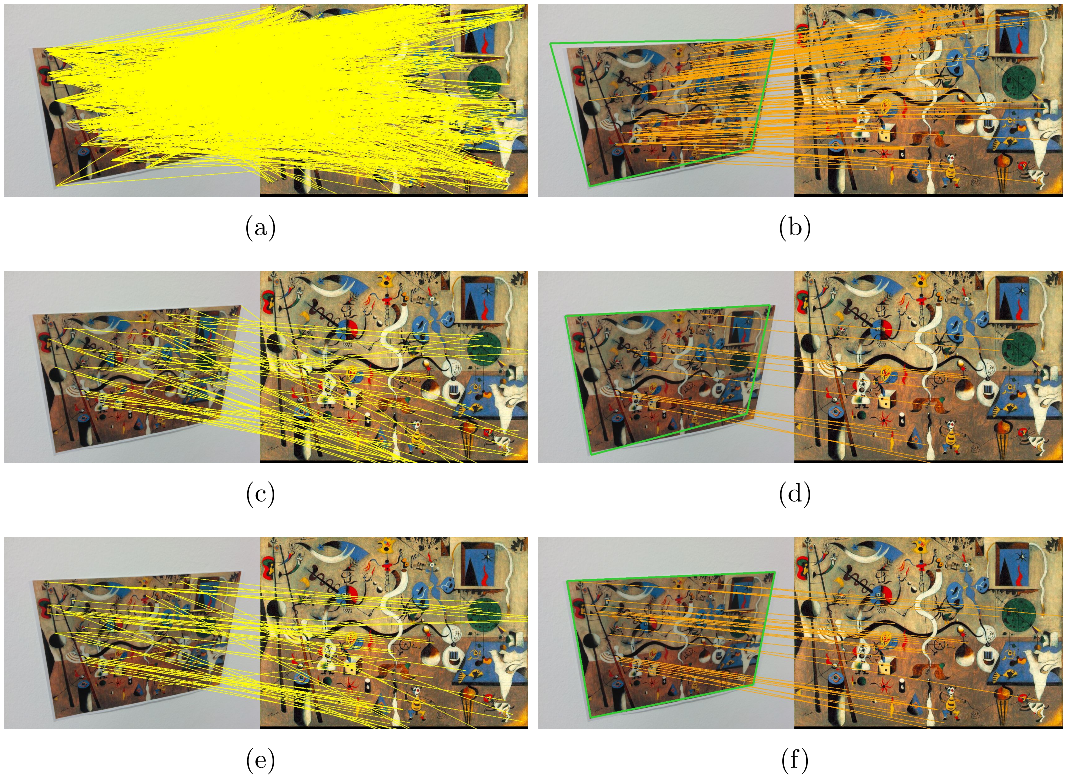
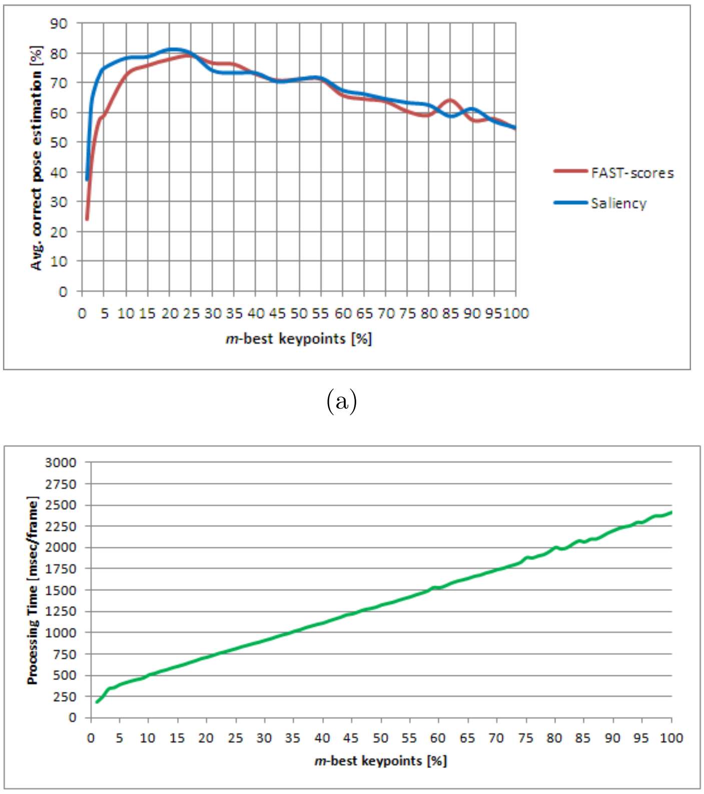


Figure 2.16: Painting transformation recovery through RANSAC algorithm by taking as input: (b) all FAST keypoints; (d) 5% m-best keypoints ranked according to FAST score, (f) 5% m-best keypoints ranked according to our saliency-based approach. Yellow segments (a), (c) and (e) denote initial keypoint pairing and orange segments (b), (d) and (f) final RANSAC inliers; the green rectangle denotes the homographic transformation inferred by RANSAC.

can be considered as a full planar object, pose is computed by estimating a homographic transformation through the RANSAC algorithm given a set of keypoint correspondences with the reference model. Figure 2.16 shows the result of pose estimation for a painting sample by considering three different cases: (a) model including all keypoints, (b) only 5% m-best keypoints selected according to FAST scores and (c) only 5% m-best keypoints selected according to our approach.
Although RANSAC is somewhat robust with respect to outliers, the advantages of using only relevant keypoints are here evident in terms of precision of the recovered viewpoint transformation. We also note how our ranking leads to consolidate a higher number of inliers and therefore a better viewpoint estimation with respect to the FAST-scores based selection. To numerically quantify pose estimation accuracy we manually marked (as ground truth) the four painting corners both for each reference image and each test frame. A pose is then considered correct when the projected painting corners (according to the estimated homography) have a spatial

distance from the corresponding ground truth lower than a prefixed threshold. In the graph of Figure 2.17 a) we show the percentage of "correct pose" estimation averaged over all 300 tests images.
We can easily note that, in both cases, the curves have an increasing trend up to a relatively small value m of best keypoints and then start decreasing. The optimal percentage of keypoint falls in the range $[5\%, 20\%]$ in our approach, and in $[15\%, 30\%]$ when selection is performed according to FAST scores.
Figure 2.17 b) shows the average processing time for a single frame analysis as function of the keypoints percentage. For this experiment we used a Samsung ATIV Smart PC (Intel Atom Processor Z2760 1.5 Ghz) tablet device, the same employed in the previous case study. Even if the code (written in C# for .NET) was not highly optimized, by selecting a percentage of keypoints below 5% we can provide a frame rate from 3 to 5 frame/s, ensuring, at the same time, good accuracy in terms of object recognition and pose estimation.



(a)

(b)

Figure 2.17: (a) Average percentage of correct poses by varying the percentage of the keypoints ranked both according to FAST-scores and our saliency-based approach; (b) Average processing time (milliseconds) required for a single frame analysis including painting recognition and pose estimation on Intel Atom Processor Z2760 1.5 Ghz.

Chapter 3

Leaf Segmentation under Loosely Controlled Conditions

3.1 Introduction

Studying the identity, the evolution and the worldwide distribution of plant species is a core activity in many nature-related fields such as agriculture, ecology and botany. The availability of basic information about plants and their thorough taxonomy is therefore essential. However, such knowledge is often unreachable not only for citizens but also for field experts like scientists and farmers, thus making identification of plants impracticable for common people as well as professionals [85].

Nowadays such situation is going to change. Recently, new tools to make world's herbaria accessible to anyone have been introduced, supported by the increasing presence of digital devices. Moreover, high volumes of data are now online and ready to be exploited by specialist and non-specialist.

One of the basic needs for plant scientists in order to carry out their work is discovering whether a certain species has been already classified and, if so, knowing the name of the species itself [15]. The standard approach does not involve visual recognition systems but it consists in manually navigating taxonomic trees node by node, by answering a multitude of questions, often ambiguous. Such manual technique results in remarkable waste of time for specialists and it is basically infeasible for inexpert users.

In order to speed-up the process and make it simpler, Computer Vision can be of

great help by offering automatic techniques to match a given plant image against a gallery and recognize the species amongst all the possible candidates. By combining vision-based plant identification with commonly used devices such as smartphones/tablets it is possible to offer new instruments to specialists and at the same time help citizen and not-professional stockholders in enjoying knowledge about plant recognition, thus helping them in understanding the surrounding nature.
In Section 3.2 we discuss challenges of leaf segmentation and recognition for plant identification whereas in Section 3.3 we review main contributes at state-of-the-art. Then, in Section 3.4 we detail our proposal to accurately segment leaves. In Section 3.5 we present the dataset we used for experimental evaluation and we show benefits of our approach over the state-of-the-art.

3.2 Leaf segmentation for plant recognition

Extracting accurately the shape of a leaf is a crucial step in image-based plant identification systems. The partial or total absence of textures on leaf surface and the high color variability of leaves belonging to same species make shape as the main recognition element [1, 15, 84, 85, 96]. Common techniques to represent leaf shape are based on multiscale curvature measures computed using differential or integral techniques since curvature is an essential property of a shape [96]. For such reason, accurate leaf segmentation plays a decisive role in the leaf recognition process.
Even though many general segmentation methods [8, 42, 45, 54, 135, 144, 175] have been proposed in the last decades, leaf segmentation presents specific challenges. In particular, a pixel-level precision is required in order to highlight fine scale boundary structures and discriminate similar global shapes. Moreover, even if the input image can typically be taken in controlled conditions, where the leaf is the only visible object over a white background, the user taking the picture is not necessarily an expert and the conditions are often not ideal: the leaf exhibits specular reflections, casts shadows, the background is never exactly white and is usually non-uniform, and the image can be blurry.
Recent leaf recognition applications in loosely conditions [96, 154] rely on the Expectation-Maximization algorithm to separate the color distributions of the foreground and the background pixels. Despite their efficiency, they do not assure

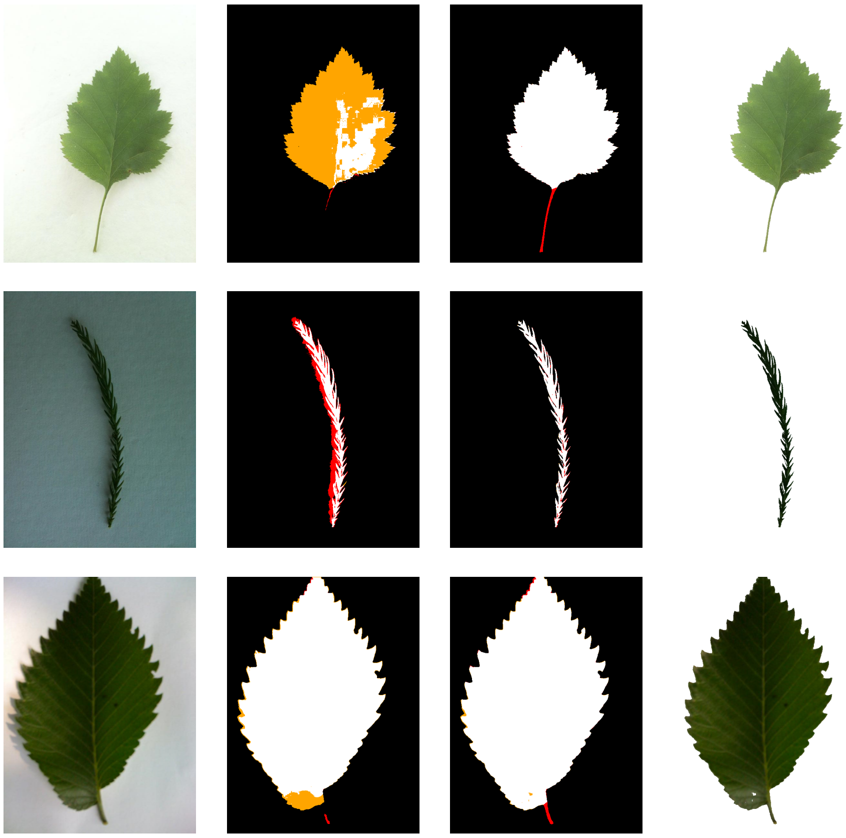


Figure 3.1: Leaves segmentation under loosely controlled conditions (best viewed in color). **First column**: Leaf images with the presence of shadows and irregular light. **Second column**: Segmentation result obtained by the Leafsnap method [96] with included post-processing procedure for stems and false positives suppression. **Third column**: Results obtained by our classification-based approach without any post-processing **Last column**: Input image masked with our segmentation. Red and orange colors are used to mark false positives and false negatives, respectively (ground truth does not include stems). We provide many other visual results in the Appendix A.

robustness to shadows and specular reflections thus leading to incorrect boundaries. In this chapter we introduce a new solution by training a pixel-wise classifier that learns filter responses associated to background and foreground regions in images of leaves. Similar classifiers have been recently used in different fields like medical applications [150] showing great performance for linear and curvilinear structures segmentation.

As shown in Fig. 3.1 and as proved in Section 3.5, we observed benefits when training our classifier by considering as positive (leaf) and non-positive (background) training samples only those pixels located in the neighborhood of the leaf boundary. Our classification-based method is more robust to the presence of shadows

and irregular light thus offering contours that better fit to the real shape.
After applying the classifier to a given input image, we threshold the score map to get segments which belong to foreground or background with a high level of probability. These segments allow us to infer precious color and spatial information about the leaf and provide a support for a suitable initialization of EM algorithm, which yields to a very fine pixel-wise segmentation robust to shadows and specularities.

3.3 Related work

Leaf segmentation represents a core activity for plant identification and research about such topics is constantly rising from the past ten years [1]. Even though great effort has been devoted to object segmentation on images in the Computer Vision history [8], leaves require a precise segmentation and/or boundary detection to effectively describe shape and its local structures. Since a detailed overview of general-purpose segmentation is beyond our scope, we focus here mainly on state-of-the-art of leaf segmentation. Moreover, we provide a review of the emerging results about filters response learning for some specific tasks like detection of curvilinear structures.

3.3.1 Supervised and unsupervised environments

Various leaf segmentation approaches related to different environmental conditions have been proposed. Image binarization with a fixed threshold, also known as "Otsu's method" [125], demonstrated good accuracy for leaf images acquired in supervised setups characterized by uniform light conditions and white background, as those included in the FLAVIA dataset [191], the Swedish leaf dataset [155], *Lab* image category of the Leafsnap dataset and the *scan* image category from the ImageCLEF plant identification challenges [85] (see Figure 3.2).
Very different solutions have been introduced for leaf segmentation on images acquired in unsupervised conditions like those included in recent ImageCLEF challenges [66, 67, 85], with *natural background* or *photo* image categories, where no assumptions are made about the background behind the leaf during image acquisition (see Figure 3.3). A number of automatic [38, 119, 196, 197] and interac-

(a) (b) (c) (d)

Figure 3.2: Example of leaf images acquired in supervised setups: (a) FLAVIA dataset [191], (b) Swedish leaf dataset [155], (c) *Lab* image category of the Leafsnap dataset, (d) *scan* image category from the ImageCLEF plant identification challenges [85].

Figure 3.3: Example of leaf images acquired in unsupervised setups [66].

tive [10, 39, 40, 197] approaches have been presented to solve leaf segmentation in unconstrained setups. In [38] two different semi-automatic and automatic segmentation approaches based on Mean-Shift and K-Means clustering in RGB color space are introduced whereas in [196] a combination of shape, color and texture features are used for plant identification. In [39] polygonal shape models of leaves are employed as prior offering very good support in unsupervised conditions but limiting its applicability to modeled species. A similar approach based on the use of semantic information and guided active contour segmentation has been later presented in [40]. As very recently illustrated in [68], due to the considerable challenge of leaf segmentation and recognition against natural background, user supervision and interaction are recommended during the process to produce reliable input images and initialize the segmentation.

3.3.2 Semi-supervised environments

Regarding to leaf segmentation under semi-supervised conditions (conditions we deal with in this chapter), several automatic approaches have been already presented and tested. Most promising ones [15, 96] are based on the use of EM [21] in color space to estimate foreground and background pixel clusters. As shown in [154], standard EM and its extensions outperform other techniques as graph-based image segmentation [54], Mean Shift [42], GrabCut [135], segmentation by weighted aggregation (SWA) [144], multiscale normalized cut [45]. Particular improvements have been demonstrated by EM when dealing with images taken with mobile devices under various pose and illumination conditions, see the *Field* or *User* image categories of [96]. However, as reported in [154], EM-based methods

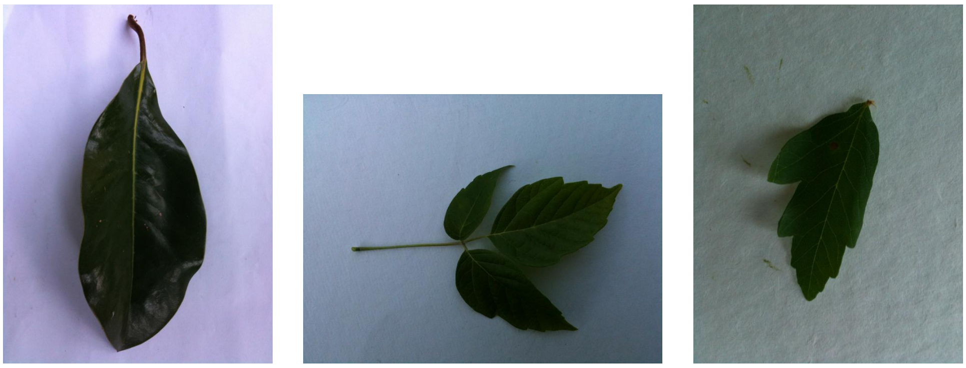


Figure 3.4: Example of leaf images acquired in semi-supervised conditions[96].

do not assure robustness to shadows, specular reflections and requires the adoption of *ad hoc* solutions also for certain particular leaves such as pine leaves, thus proving the weakness of EM initialization.
In this chapter, we show that our classification-based initialization for background and foreground color distribution represents a better solution for the problem at hand. Cascade classifiers [141] exhibit good performance in localization thus allowing a better discrimination of points with similar appearance, as those placed across object contours. Advantages offered by the learning of filter responses have been recently proved in different fields like biomedical images, aerial images and general-purpose contour detection [150]. We aim to apply a similar idea for leaf segmentation, by combining prior knowledge with learning and the adaptability of EM-based methods.

3.4 Proposed segmentation method

Let $I(\boldsymbol{x})$ be an image of a leaf, and $\boldsymbol{x} \in \mathbb{R}^2$ an image pixel location. Leaf segmentation can be carried out by computing the probability distribution of all pixels $\boldsymbol{x}$ and representing it as the mixture of two Gaussians:

$$p(I(\boldsymbol{x})|\boldsymbol{\Theta}) = \sum_{i=1}^{2} \omega_i \, p_i(I(\boldsymbol{x}) \mid \boldsymbol{\Theta}_i) \qquad \boldsymbol{\Theta}_i = \{\boldsymbol{\mu}_i, \Sigma_i\} \quad , \tag{3.1}$$

where $I(\boldsymbol{x})$ is the color of image I at location $\boldsymbol{x}$, and $\boldsymbol{\Theta}_1 = \{\boldsymbol{\mu}_1, \Sigma_1\}$ and $\boldsymbol{\Theta}_2 = \{\boldsymbol{\mu}_2, \Sigma_2\}$ are the parameters of the foreground—the leaf—and background color distributions, respectively. ω_1 and ω_2 weight these two Gaussian distributions.
One way to infer such distributions is to apply K-Means or Expectation-Maximization as in [15, 96], thus computing the parameters and weights of the two Gaussians for the given image and using them for pixel-wise segmentation. [96] considers only the saturation and value color components for EM clustering and a shared covariance matrix is used. However, in practice, some drawbacks appear in this formulation due to challenging leaves like pine needles, false positives detection related to shadows and false negatives detection related to specularities. To tackle such problems, some manually-defined assumptions are made about cluster regions and pixel weights (see Fig. 2 in [96]). Furthermore, post-processing operations are carried-out to remove false positive detections due to shadows and irregular backgrounds, at the risk of hurting the final leaf shape.
To assure more robustness to shadows and specularities, our solution is to pre-train a pixel-wise classifier by learning a function $y(\cdot)$ such that:

$$y(f(\boldsymbol{x}, I)) = \begin{cases} 1 & \text{if } \boldsymbol{x} \text{ is on the leaf surface,} \\ -1 & \text{otherwise,} \end{cases} \tag{3.2}$$

where $f(\boldsymbol{x}, I)$ is a feature vector computed from a neighborhood surrounding $\boldsymbol{x}$ in image I.
By performing a simple thresholding of the score map returned by the classifier we detect segments that belong with high probability to foreground and background. These segments are then exploited to properly initialize a standard EM algorithm thus leading to a final and accurate leaf segmentation. Furthermore, we will show that our learning assures independence from leaf species, since the same classifier is used for all species and no *ad hoc* solutions are required when challenging species

have to be treated.

In the remainder of this section we firstly illustrate our pixel-wise classifier by showing our training that focuses on leaf boundary. Then, we describe how we produce and employ segments to initialize the EM algorithm thus leading to the final segmentation.

3.4.1 Pre-trained pixel-wise classifier

To train our pixel-wise classifier we employ a similar approach to [150]. Given a set of training samples $\{(f_i, y_i)\}_{i=1,\dots,n}$ where $f_i = f(\boldsymbol{x}_i, I_i) \in \mathbb{R}^J$ is the feature vector corresponding to a point $\boldsymbol{x}_i$ in image I_i and $y_i \in \{1, -1\}$ is the label associated to $\boldsymbol{x}_i$, we use GradientBoost and regression trees [75] to approximate $y(f(\boldsymbol{x}, I)$ by a function of the form:

$$\varphi(f(\boldsymbol{x}, I)) = \sum_{k=1}^{K} \alpha_k h_k(f(\boldsymbol{x}, I)) \quad , \tag{3.3}$$

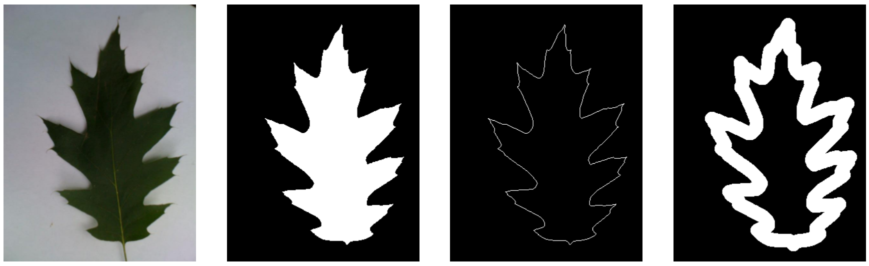
where $h_k : \mathbb{R}^J \to \mathbb{R}$ are weak learners and $\alpha_k \in \mathbb{R}$ are weights.

As shown in Fig. 3.5, we focus our attention on the leaf boundaries by selecting only samples in their neighborhoods. We extract from each training image the leaf contour from the ground truth segmentation and simply thicken this contour with standard morphology dilation. Function φ is built iteratively by minimizing an exponential loss function $\mathcal{L}$ of the form:

$$\mathcal{L} = \sum_{i=1}^{n} L(y_i, \varphi(f(\boldsymbol{x}, I))) \quad , \tag{3.4}$$

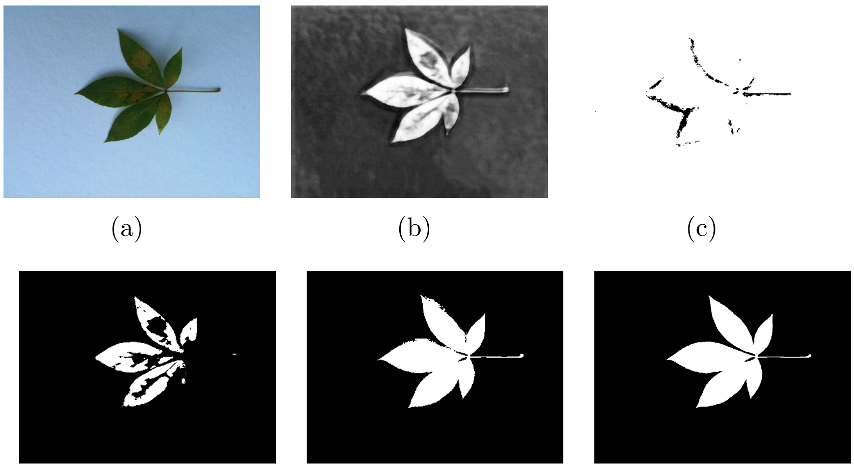
where $L = e^{-y_i \varphi(f(\boldsymbol{x}, I))}$. We also experimented with the log loss function with similar results.

We use a set of convolutional filters learned from the training images as described in [150]. The RGB images are converted into the LUV color space and we learn a different filter bank for each channel.



(a) (b) (c) (d)

Figure 3.5: Example of images used for training: (a) a leaf image, (b) its manually defined ground truth segmentation, (c) the leaf contour extracted from the segmentation, (d) thicker contour obtained by simple dilation. We train our classifier by selecting positive (leaf) and negative (non-leaf) feature samples computed on image (a) that lie on the thicker contour only.



(a) (b) (c)

(d) (e) (f)

Figure 3.6: Segmentation pipeline: (a) Input image I_{test}, (b) score map obtained by applying our pre-trained classifier at each pixel location, (c) pixels belonging to background with high probability (black), (d) pixels belonging to foreground with high probability (white), (e) coarse leaf segmentation obtained using the prior $\mathbf{\Theta}1_{\text{start}}, \mathbf{\Theta}2_{\text{start}}$ built from images (c) and (d), (f) final leaf segmentation after EM optimization from this initialization. Since our training is focused on leaf boundary, high probability background and foreground pixels are more likely to be found near the leaf boundary thus guiding and improving the following final segmentation.

3.4.2 Score map thresholding and segmentation

Our segmentation pipeline is summarized in Fig. 3.6. We apply the classifier described above to each pixel location of a given unknown test image I_{test}. This provides a score map that we then threshold using two different thresholds to detect pixels that belong to foreground and background with a high level of probability. With these pixels at hand we initialize an EM algorithm to estimate foreground and background cluster parameters $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ by working in the saturation-value color space.

This allows us to compute good initial estimates for $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$, the mean and covariances of the colors over the leaf and over the background. This is by contrast with [96], which has to initialize the EM segmentation with the same values for all the images. The other difference with [96] is that we can consider as unlabeled data only the pixels that are in the neighborhood of the detected leaf boundary. This allows to keep focusing on segmenting correctly the pixels around the leaf boundary, and in practice it is enough to get a good segmentation of the other pixels, which are easier to classify.

Moreover, our segmentation method allows speed optimization. Indeed, processing steps from (a) to (e) represented in Figure 3.6 are carried out by working on a downsized version of the original image thus saving processing time to perform classification and further operations. Mathematical model of foreground and background color distributions obtained from the coarse segmentation are then employed to perform the final segmentation of the leaf image at the original size. In this two-class scenario, the posterior function which defines cluster membership takes a linear logistic form that is efficiently computed through as follows:

$$p(z = 1|\boldsymbol{x}) = 1/[1 + \exp(\beta_0 + \boldsymbol{\beta}^T\boldsymbol{x})], \text{ where} \tag{3.5}$$

$$z \in \{1, 2\} \text{ and} \tag{3.6}$$

$$\boldsymbol{\beta} \equiv (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T\Sigma^{-1}, \text{ and} \tag{3.7}$$

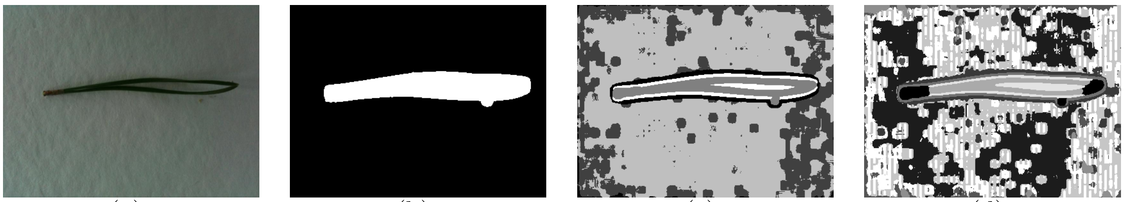
$$\beta_0 \equiv -\frac{1}{2}(\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1) \tag{3.8}$$

$$p(z = 2|\boldsymbol{x}) = 1 - p(z = 1|\boldsymbol{x}) \tag{3.9}$$

3.4.3 Segmentation with *augmented* color vectors

In Subsection 3.4.1 and Subsection 3.4.2 we illustrated our proposed solution for leaf segmentation which involves a classification stage to build a robust starting prior of foreground/background. Classification stage is performed by computing suitable filter responses which constitute our learned features. Then EM is

executed to infer color distributions in saturation value color space focusing on unlabeled 2-dimensional color vectors selected from leaf boundary.
Since EM is based only on color information, the reader can argue what could happen if the classifier is replaced by a better initialization of EM. In particular, it could be possible to augment color vectors by adding filter responses (*i.e.* , the f_i vectors) we already adopt in our classification stage. In such scenario, the classification step would be removed and filter responses plus color information employed to perform a direct and unique EM-based segmentation. In other words, using augmented color vectors we do not exploit any prior information about high probability pixels but we run only standard EM in the $M + 2$ dimensional space (*i.e.* , M components from f_i plus 2 color components) thus directly inferring background and foreground clusters (Gaussians).
In Figure 3.7 different results for the same input image are reported. In particu-



(a) (b) (c) (d)

Figure 3.7: Segmentation with augmented color vectors and different number G of inferred Gaussians: (a) input image, (b) segmentation with $G = 2$ (background/foreground), (c) segmentation with $G = 3$,(d) segmentation with $G = 5$. Different colors are used to highlight pixels belonging to different clusters.

lar, when inferring the two Gaussians related to foreground and background using augmented color vectors we get very "round" and "dilated" segmented shapes (Figure 3.7 b)). The most probable explanation of such behavior is related to the fact that filter responses for each pixel are computed considering pixel-centered patches and pixels placed immediately outside the leaf offer similar responses to pixels placed immediately inside. Therefore, EM tends to assign them to the leaf cluster (Gaussian). On the contrary, pixel-centered patches located in plain background offer very different filter responses and its easier for EM to discriminate them. Furthermore, in such case we deal with vectors of more than 500 components and only 2 are related to color, thus making color information not very influential as before. This behavior is confirmed in Figure 3.7 c) - d) where three and five Gaussians are estimated during EM, respectively. Using more Gaussian it is possible to highlight different regions corresponding to different filter responses.

A different way to exploit EM and augmented feature vectors is by computing the ratio of the probabilities (Eq. 3.5) to belong to each of the clusters to get a continuous score. In Figure 3.8 some examples of such continuous score (second column) are reported, compared with the score map (third column) we get from the classification stage illustrated in Subsection 3.4.1. By using augmented fea-

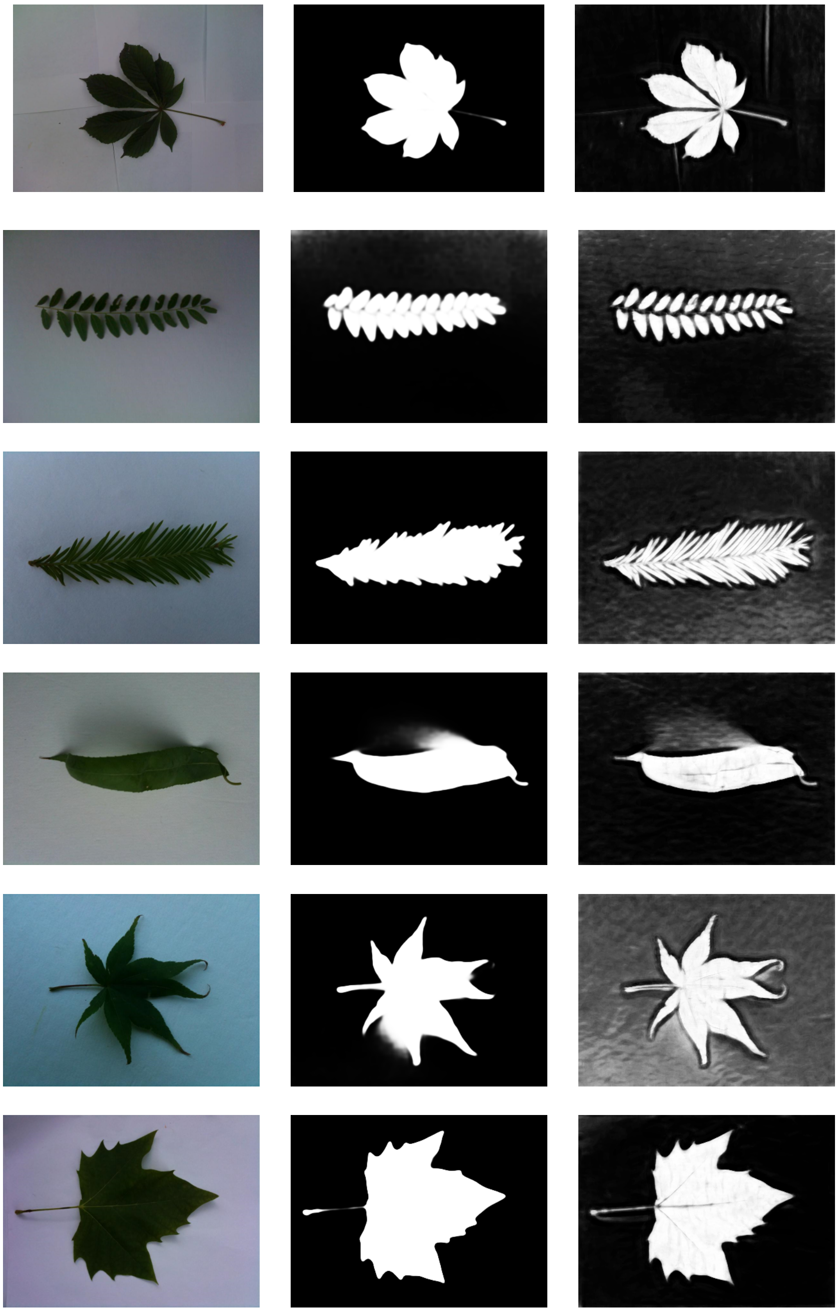


Figure 3.8: Score maps obtained after EM on augmented color vectors by computing the ratio of the probabilities to belong to each of the clusters (second column) and score maps obtained through the classification stage (third column).

ture vector to carry out EM we are not able anymore to capture local structures like serrations, compound leaves and sharp extremities whereas score maps we get from our classification seem more adapt to capture such local structures.
Images in second column of Figure 3.8 could theoretically substitute classification-based score maps in our segmentation pipeline. However, it would lead to a greater processing time for segmentation, due to the large dimension of feature vectors which considerably degrades speed of EM. Therefore, a classification stage as described in Subsection 3.4.1 is not only more suitable to faithfully represent local structures of leaves but it is also faster.

3.5 Experimental evaluation

In this section, we first describe the dataset and the evaluation protocol we used for our experiments. We then compare our method with techniques that demonstrated state-of-the-art performance on loosely controlled conditions, Leafsnap[96] and GrabCut [135]. In particular, we show the benefit given by our classification-based initialization of EM as described in Section 3.4.2. Moreover we evaluate the importance of training the classifier from samples close to the leaf boundaries. We finally provide qualitative results of our segmentation approach.

3.5.1 Leaves dataset and performance metrics

For evaluation we use the *Field* image dataset publicly available online [96]. It is made of 185 different species for a total of 7719 images acquired against solid background and variable light conditions thus simulating typical images that a user could provide for plant recognition.
To train our pixel-wise classifier we randomly select one image for each species and we manually produce segmentation and thicker contour to discriminate between positive and negative training samples placed in the neighborhood of boundary as described in Section 3.4.1.
Since segmentation ground truth is not available and its manual production for thousands of images would require an inestimable amount of time, we considered a subset of the original *Field* dataset. Our testing set is made of 300 images: 150 images for which the EM approach of [96] performs already well thus producing faithful segmentation in accordance with the leaf shape plus 150 more challenging

images for which EM partially or totally fails.
185 training images are randomly selected excluding those images already used to test the classifier. A total of 485 leaves (185 for training and 300 for testing) was therefore manually segmented to produce the ground truth; stems and unrelated components are not part of the ground truth in accordance to the policy employed in [96]. [1]
To compute performance indicators we rely on the publicly available and very popular code of Berkeley Segmentation and Boundary Detection Benchmark [175]. In particular, as in [154], we evaluate the leaf segmentation results by analyzing boundary agreement with ground truth in terms of recall, precision and F-measure since contour is the main recognition cue in typical plant recognition systems.

3.5.2 Segmentation performance

Accuracy measures are reported in Table 3.1 and Table 3.2. Specifically, in Table 3.1 we provide recall, precision, and the F-measure (ODS) which is the harmonic mean of precision and recall to evaluate the trade off between these two measures:

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad . \tag{3.10}$$

Such metrics are computed for the global testing set whereas in Table 3.2 the same results are reported when only the 150 more challenging images are considered to highlight the benefits of our approach.
We compare the two different strategies to train the classifier and initialize the EM segmentation: using samples from the entire image, strategy that we denote *Ours-entire*, and using samples only close to the leaf boundaries, which we denote *Ours-boundary*. Moreover, we report results when only the pre-trained classifier is employed for segmentation (*Classification*).

The benefits of our method already appear clearly in Table 3.1, with a significant raise of the F-measure with respect to the other methods. Even without performing any post-processing to remove false positives, shadow and stems—we remind here that in our ground truth stems are removed, the F-measure for the entire image is better with respect to Leafsnap thus proving the robustness of our

[1] Our manual ground truth segmentation is publicly available at http://smartcity.csr.unibo.it/leaf-segmentation/

	Image segmentation quality			Leaf boundary quality		
	Recall	*Precision*	*F-measure*	*Recall*	*Precision*	*F-measure*
Classification	0.701	0.480	0.570	0.702	0.778	0.738
Leafsnap	0.618	0.858	0.718	0.618	0.929	0.742
Leafsnap*	0.644	0.764	0.699	0.644	0.931	0.762
GrabCut	0.624	0.848	0.719	0.624	0.964	0.757
Ours-entire	0.690	0.800	0.741	0.690	0.940	0.796
Ours-boundary	0.692	0.822	**0.752**	0.693	0.944	**0.799**

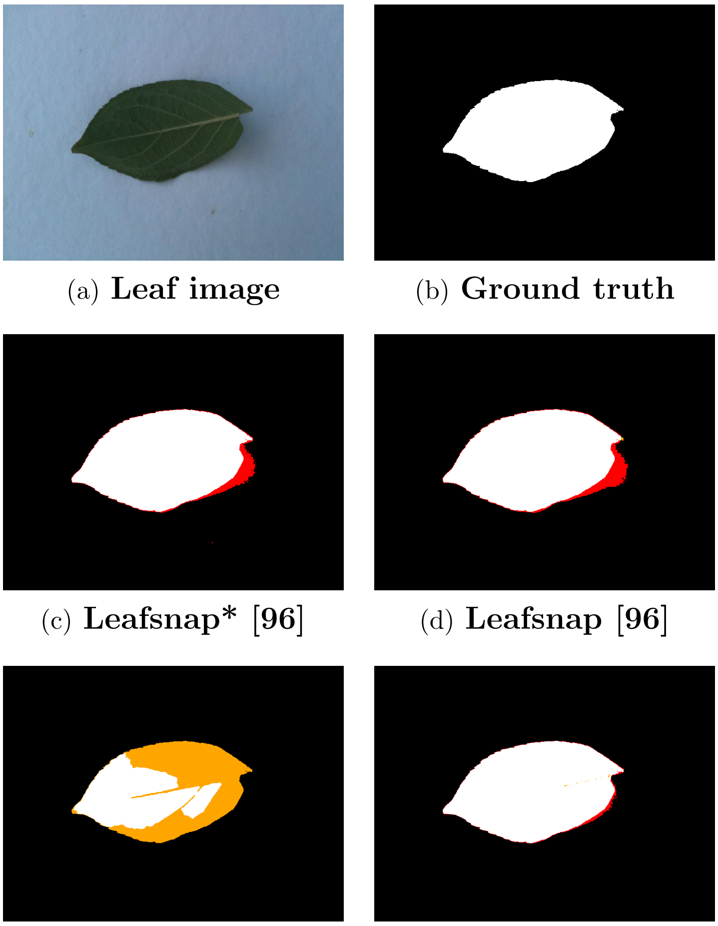
Table 3.1: Recall, precision, and F-measure for the entire testing set (300 images). Our method provides the best trade-off between recall and precision.

	Image segmentation quality			Leaf boundary quality		
	Recall	*Precision*	*F-measure*	*Recall*	*Precision*	*F-measure*
Classification	0.700	0.532	0.604	0.700	0.788	0.742
Leafsnap	0.560	0.777	0.651	0.560	0.884	0.686
Leafsnap*	0.614	0.703	0.656	0.614	0.903	0.731
GrabCut	0.598	0.830	0.695	0.598	0.959	0.737
Ours-entire	0.682	0.772	0.724	0.682	0.923	0.785
Ours-boundary	0.686	0.792	**0.735**	0.686	0.927	**0.788**

Table 3.2: Recall, precision and F-measure on 150 challenging images from the testing set. Our method provides the best trade-off between recall and precision.

method to false positives. Moreover, using only samples placed on leaf boundary to pre-train our classifier (*Ours-boundary*) we outperform all the other methods. Looking at Table 3.2 where only challenging images are considered, the improvements due to our method are confirmed to a greater extent. The results prove that a post-processing based on morphological operations as erosions and dilations hurts quality of boundary especially in terms of recall, thus motivating the adoption of methods already robust to shadows and irregular light.

The behavior of different methods can be qualitatively appreciated looking at Fig. 3.9 where results returned by Leafsnap, Leafsnap without post-processing (marked with *), GrabCut and our method *Ours-boundary* are reported. As the reader can see comparing ground truth details with real segmentations, it is confirmed that post-processing hurts quality of boundaries and should be avoided. On the other hand, GrabCut tends to return round contours. With our method some errors still remain, due to those background pixels that look strongly similar to leaf (and vice-versa). However, our method represents a good compromise since we do not use post-processing but at the same time we assure a good robustness



(a) **Leaf image** (b) **Ground truth**

(c) **Leafsnap* [96]** (d) **Leafsnap [96]**

(e) **GrabCut [135]** (f) **Ours-boundary**

Figure 3.9: Leaves segmentation under loosely controlled conditions with different methods. Note that our approach strongly reduces negative effects of irregular light and shadow regions thus offering a more well defined leaf shape with respect to other methods that do not adapt to specific light conditions. Red and orange colors are used to mark false positives and false negatives, respectively (ground truth does not include stems). Best viewed in color.

to false positives. Furthermore, our contours tend to fit better with the ground truth.

Our non-optimized MATLAB code on a 4-core virtual machine with 16GB of RAM requires about 50 seconds to produce segmentation for one image. The majority of time is required to do classification and produce score map whereas only few milliseconds are required for EM segmentation. Even though at this stage we are not able to guarantee competitive processing time, we are confident that with proper code optimization and the use of more performing hardware like physical machines we can reach much shorter run-times.

Chapter 4

Candidate Photo Selection for Face Recognition from Sketch

4.1 Introduction

People identification has become today a central activity in order to provide custom services to the community and face recognition is one of the most widely used methods to perform such task. Face is a trait that can be acquired in a not-intrusive way and with a low degree of cooperation by users. This is one of the main advantages that make face recognition a relevant topic not only in the field of biometric systems but also in many AmI scenarios [82].

Face is a widely accepted biometric characteristic and usually people have no particular scruple to allow its use for identification. One of the main motivations that could lead a subject to hide his identity is avoiding controls of law enforcements in relation to some kind of criminal or malevolent activity. Unlike other biometric elements, the face of a suspect could be obtained in a secret way not only through the well-known video surveillance cameras but also through a reconstruction provided by an eyewitness who was present when the criminal event occurred. By exploiting a verbal description and the manual skills of a forensic artist, the eyewitness may help to realize a forensic sketch (also known as identikit) of the criminal face in order to support the identification process [65, 161].

Sketch recognition is usually carried out either by broadcasting the image to citizens (hoping that someone could recognize the suspect as a familiar subject) or comparing it with *mug shot* databases owned by the police where generalities and

photos of a large number of subjects have been recorded during years.
The visual comparison with such large databases is of course time consuming and difficult, so that the research community is proposing new techniques able to perform an automatic matching. Unfortunately, many state-of-the-art approaches for face recognition do not allow a direct comparison between sketches and mug shot photos since they are characterized by two distinct modalities of face representation, with uneven richness of details and texture. To overcome this *modality gap*, several approaches of photo-sketch recognition have been proposed in recent years. In spite of their excellent recognition performance on popular databases such as CUFS and CUFSF [169, 170, 184, 203] (Figure 4.1), such approaches do not focus on the aspect of efficiency which is a strict requirement in real applications where the mug shot databases usually contain a very large number of images. In this realistic scenario, finding the most similar mug shot photo to a given sketch might require a long processing since state-of-the-art recognition techniques are rather complex and not well-suited to deal with large-scale galleries.
In this chapter we investigate a possible solution to the problem of dealing with

Figure 4.1: Some examples of real faces (odd columns) and associated sketches (even columns).

large galleries for face sketch recognition. In particular, we present an approach designed to restrict the search space and based on the use of suitable shape features.
The main advantage of our proposal is that shape features can be computed and matched very efficiently, with a great saving of computational effort. Starting from a given sketch, the use of shape features allows to find the most similar photos in the gallery, thus limiting the subsequent (and more computationally demanding) recognition to a reduced but significant portion of data. Moreover, the proposed system overcomes the modality gap since it works on a common representation based on edges which can be easily extracted from both photos and sketches. Thanks to this common representation, shape descriptors of a face are quickly computed and compared.

In the following, after a review of the state-of-the-art about face sketch recognition, we show how to build and combine shape features to obtain a candidate photo selection from a large mug shot gallery thus filtering out clearly misleading subjects. Moreover, we show how our preliminary candidate photo selection avoids unnecessary image processing during fine-grained recognition performed with more complex and robust techniques. Eventually, in the last part of the chapter, we will propose an indexing technique based on multi-space KL and shape descriptors thus completing the path of candidate photo selection. We show that our indexing proposal results in a more general, efficient and scalable method for photo retrieval from sketch on large mug shot galleries.

4.2 Related work

4.2.1 Photo-based *vs* sketch-based face recognition

Face recognition has been a hot research topic especially in the last three decades [82, 207]. Various algorithms and techniques have been introduced considering both 2D and 3D models in order to tackle intrinsic challenges such as variations of light, viewpoint, expression, aging and occlusions due to hair and glasses. Despite peculiar traits of state-of-the-art methods, they share a common idea. In particular, recognition is usually carried out by matching a probe against a gallery and both probe and gallery are composed by homogeneous elements: a probe face image is compared against a set of face images, a 3D model is compared against a gallery of 3D models, and so on.

Differently from common face identification situations, sketch-based face recognition presents a completely different challenge: a drawing has to be matched against real photos and many standard approaches cannot be always applied. Variations of light of viewpoint do not represent big problems in sketch-based recognition, since the face is usually represented in a canonical fashion both in sketches and mug shot photos. Instead, the main hurdle is due to the different richness of details and textures of such images: the ”output” returned by a pencil on a paper is quite divergent from the one returned by a camera. Moreover, the forensic artist could misunderstand the verbal description of the eyewitness, thus introducing grotesque or emphasized face details in the final sketch/identikit.

To deal with the problem of face recognition from sketch, research has been focused on two parallel directions:

1. deep understanding of the psychological mechanisms which regulate and influence the sketch production;
2. study of techniques to directly compare sketches and photos thus overcoming the modality gap.

The second point will be exhaustively examined in Subsection 4.2.2. With reference to the first point, various studies have been realized in psychology [17, 23, 24, 50, 131] which resulted to be precursors of sketch-based recognition approaches in Computer Vision. Providing details about them is beyond the scope of this thesis, but it is worth noting that their focus has been posed on the surprising capability of humans in recognizing objects and faces even when they are the result of caricature or crippling, thus leading to the intuition that our brain exploits exactly such emphasized elements to carry out the recognition.

4.2.2 Generative and discriminative approaches

On the basis of the technique used to tackle the previous mentioned modality gap, state-of-the-art methods for face recognition from sketch can be framed into two categories: *generative* and *discriminative* approaches [20] (a summary is reported in Table 4.1).
Generative approaches perform a preliminary conversion (synthesis) from photo to sketch (or vice versa) and then make a comparison photo-to-photos or sketch-to-sketches. After the initial synthesis it is possible to use traditional methods of face recognition. To this regard, important contributions are those of Wang and Tang [158–160, 184], which proposed several different methods for photo-to-sketch synthesis and recognition like eigentransformation, Markov Random Fields (MRF), Bayesian classifier and the use of separate texture and shape information. Liu *et al.* [103] proposed a pseudo sketches generation from photographs with a method based on local linear geometry preservation and a recognition approach based on a Nonlinear Discriminant Analysis (KNDA), whereas Gao *et al.* [63] showed a sketch synthesis method based on Hidden Markov Models (E-HMM). E-HMM have been used also by Xiao *et al.* in [194]. Other contributions are due to Li *et al.* which

described a generative approach robust to illumination variations [100] as well as Zhang *et al.* [202]. Another scheme for photo-sketch transformation has been illustrated by Liu *et al.* [104] and it is based on the Bayesian Tensor Inference.

Due to some drawbacks of generative approaches, such as the complexity of implementation and deployment, some discriminative approaches have been developed to exploit modality independent features of photos and sketches, making possible a direct matching. Some of them require a specific pre-processing activity in order to standardize sketches and photos and to allow the use of specific descriptors: Uhl and Lobo [167] performed a photometric standardization and a geometric normalization before doing a direct matching with eigenanalysis, whereas Bhatt *et al.* [19] described a decomposition into multi-resolution pyramid in order to conserve high frequency information and use the extended Uniform Circular Local Binary Pattern (UCLBP) for the matching. Yuen and Man [198] proposed a two-phase method based on the use of local and global facial features in the first phase and on a relevance feedback technique in the second phase. Very important contributions come from studies of Klare and Jain [91] which described the use of Scale Invariant Feature Transform (SIFT) to compute invariant descriptor for both sketch and photos. Still Klare and Jain [92] recently published a Prototype Random Subspace (P-RS) framework for heterogeneous face recognition which can be applied also in the photo-sketch recognition scenario. A previous heterogeneous face recognition framework tested on optical image-infrared and sketch-photo pairs was introduced by Lin and Tang [101]. Approach [91] was further investigated by Klare *et al.* [93] leading to the creation of a Local Feature-Based Discriminant Analysis (LFDA) framework, in which both sketches and photos are represented using SIFT and Multiscale Local Binary Patterns (MLBP). Another descriptor, named Weber's local descriptor, was designed by Bhatt *et al.* [20] to perform a direct matching. Other important contributions to direct matching came from Zhang *et al.* in [203] and in [206]: in the first one a new face descriptor based on coupled information-theoretic encoding is discussed; in the second one a comparison of sketch recognition performances between humans and the Principal Component Analysis (PCA) method is presented.

Previous mentioned approaches [20, 92, 93, 167] share the ability to deal with viewed sketches as well as forensic sketches (differences between them are exhaustively described in [93]). Recent studies were also conducted on composite sketches [72, 95]. The effects of matching sketches realized by several artists are investigated by Nizami *et al.* [121] and Zhang *et al.* [205].

In real criminal identification contexts, a photo gallery could be potentially composed of several thousands of items. Among the mentioned state-of-the-art approaches, only a few works report results on so large datasets [20, 72, 93]). Most of the tests are carried out on datasets containing a number of photos exactly corresponding to the number of sketches in a ratio of one-to-one, with no information about the scalability of the system with the increase of the gallery. A mug shot photo candidate selection algorithm, able to limit the set of photos to evaluate for a given query sketch, can significantly reduce the search time.

Approach type	**General idea**	**Proposed methods**
Generative	Synthesis from photo to sketch (or viceversa)	Wang and Tang [158–160, 184], Liu *et al.* [103], Gao *et al.* [63], Xiao *et al.* [194], Li *et al.* [100], Zhang *et al.* [202], Liu *et al.* [104]
Discriminative	Direct matching (no synthesis)	Uhl and Lobo [167], Bhatt *et al.* [19], Yuen and Man [198], Klare and Jain [91] Lin and Tang [101], Klare *et al.* [93], Bhatt *et al.* [20], Zhang *et al.* [203, 206] Klare and Jain [92]

Table 4.1: State-of-the-art approaches for face recognition from sketch.

4.3 Photo and sketch pre-processing

The technique we propose for face recognition from sketch can be framed into the category of discriminative approaches since it does not directly compare original sketches and photos, but attempts to overcome the mentioned modality gap by extracting shape features able to effectively represent both types of image.
In particular, we adopt a preparatory pre-processing to highlight the main facial features, successively used for shape encoding. The proposed pre-processing procedure is structured in two sequential operations. First of all, images are normalized, cropped, resized and aligned according to standard parameters to ensure that descriptors calculated on specific positions of images will refer to the same class of local appearances (eye, nose, chin, eyebrows, etc.). Then, edge detection is executed on the normalized images in order to obtain new images in which only

face edges are highlighted. Such pre-processing activity, as well as shape features computation (Section 4.4), must be performed for each mug shot photo only once to create or to update the face database. The same operations have also to be carried out on the input whenever there is a new sketch of a suspect that has to be identified. The use of such simplified representation is feasible in the context of sketch recognition since both sketches and mug shot images are well controlled, with a frontal pose and a uniform background, thus allowing the extraction of reliable edge images for shape feature extraction.
In the literature many edge detection algorithms have been proposed; one of the most used is Canny edge detector [31]. However, Canny method introduces some artifacts, spurious elements and an unwanted fragmentation effect of faces contours when performed on our sketch and photo images. Because of such issues, we prefer to adopt the Image and Vector Processing Framework (IVPF) [94] as edge detection environment. IVPF provides a series of operational building blocks which execute specific image processing functions. Through a suitable graphic user interface (GUI), the blocks can be parametrized and connected each other to create a processing network that takes/returns images as input/output, respectively. The GUI of IVPF software is depicted in Figure 4.2; in the right-hand side of the GUI it is possible to see some of the provided basic functions.
To carry out contour extraction on real photos and sketches, two distinct processing networks have been adopted in order to properly process these different type of images. An example of such networks is reported in Figure 4.3. They are composed by the same operational building blocks; the main differences are related to the functional parameters of each block. In particular, such parameters have been tuned in order to find the best configuration by considering some sketch and photo test examples. The building blocks which constitute the two networks are described below.

- *ImageFromFile* and *LDToFile*: their function is loading and saving images.

- *GaussianFilter*: this block performs a Gaussian filtering of the image and its main purpose is removing small noise from the input image. The user (network designer) needs to set some parameters such as the windows size and variance.

- *GradientOperator*, *NonMaximaSuppression* and *HysteresisThresholding*: this blocks compute the gradient map by sliding a derivative filter on the input

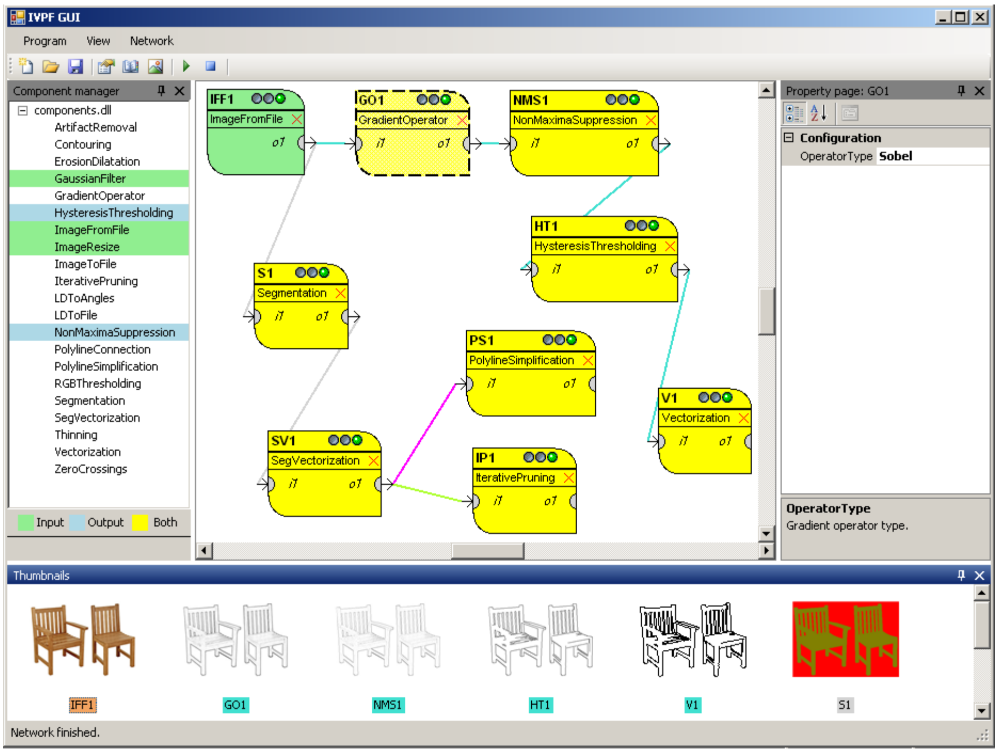


Figure 4.2: GUI of the IVP Framework and a generic example of a processing network.

image, plus a non-maxima suppression and hysteresis thresholding, thus producing thin and not very fragmented contours. These are common steps in contour extraction. The user can choose the derivative filter among different options: Sobel (as we did), Prewitt and Roberts.

- *Vectorization*: such component transforms a binary image in a set of polylines and polygons by exploiting the pixel-chain method [94]. Basically, the contour image is scanned point by point and foreground pixels are connected together with their neighbors according to a given connection policy (end-line connection or junction connection).

- *PolylineConnection*: this final block performs an analysis of polylines in order to create connections and junctions among them. Main parameters that regulate the creation of such connections are the minimal angle, the maximal angle, the distance between polylines, validity of intersection. For further details we address the reader to [94].

The result of the pre-processing procedure is a normalized image containing the main face edges (see Figure 4.4 and Figure 4.5): the edge-based representation allows a direct comparison between photos and sketches on the basis of the shape features described in Section 4.4.

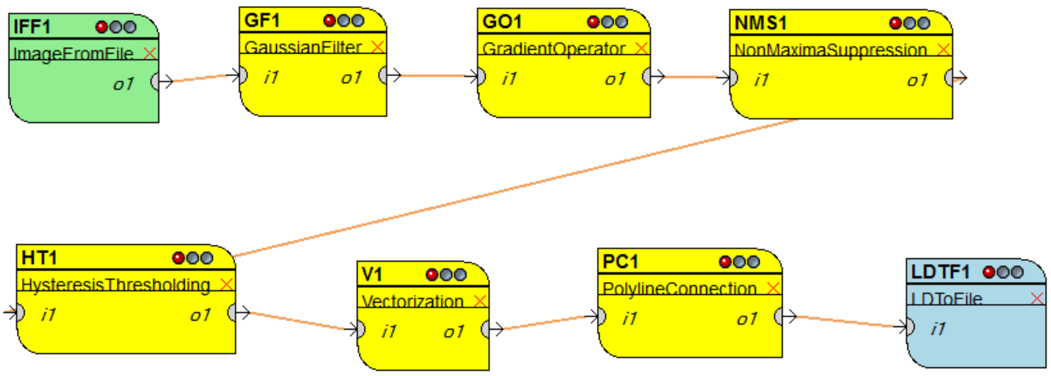


Figure 4.3: IPVF processing network employed for contour extraction on real photos and sketches.

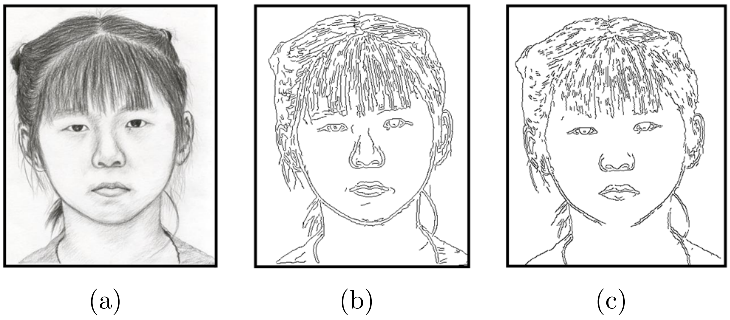


Figure 4.4: Differences between edge extraction from a sketch (a) using the Canny operator (b) and the IVP Framework (c).

4.4 Shape features

With *shape features* we denote a class of descriptors which numerically represent some salient characteristics of a generic shape. Shape features are core descriptors in so-called Sketch-Based Image Retrieval Systems (SBIR) [22, 52, 53, 78]) where image retrieval is carried out by using a generic draw as input query. The proposed approach presents some similarities with classic SBIR systems but also some strategic differences. In particular, previously mentioned SBIR systems are typically designed to distinguish shapes related to objects belonging to different

(a) (b)

Figure 4.5: Edge images for a given sketch (b) - photo (a) pair related to a given subject. The subsequent steps of the algorithm are performed on edge images, thus overcoming the modality gap.

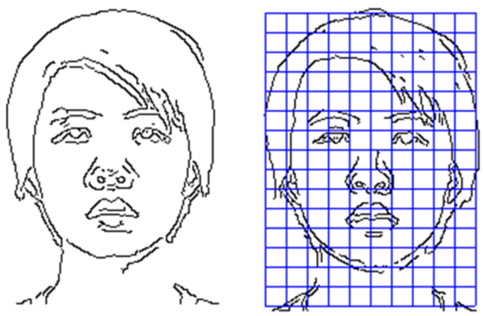
categories (*e.g.* , distinguish a car from an animal or a building). In our case, we must deal with a unique object category (faces) and we need to process information in order to discriminate amongst different elements semantically belonging the same category (one subject from another). Consequently, it is necessary to use specific descriptors able to capture and describe distinctive global and local appearances. Furthermore, SBIR systems typically deal with regular and well-defined shape contours while the proposed approach has to deal with irregular face shapes (such as those illustrated in Figure 4.4 c) and Figure 4.5).
In the literature several shape features have been proposed, many of which are described in an interesting survey by Yang *et al.* [195]. In the following part of the chapter we will show how we employ six different shape features (some of them inspired by [195]) for face representation. For each shape feature, a similarity measure is defined: as described in Section 4.5, given an input sketch to identify each shape feature should be independently evaluated in order to produce a ranking of the most similar photos according to suitable similarity (or distance) measures between related descriptors.

4.4.1 Shape Matrix

Shape Matrix descriptor (SM in the following) is computed by superimposing an $M \times N$ matrix on the face shape (Figure 4.6). In [195] two different version of such feature are presented: Square Model Shape Matrix and Polar Model Shape Matrix. Here we employ the first one: within matrix cells, each pixel (or point) can be uniquely classified as foreground if it belongs to a detected edge or background pixel otherwise. So, for each cell (j, k) it is possible to determine a cell coverage offered by the foreground pixels. Such coverage value can be expressed through a percentage B_{jk} determined as follows:

$$B_{jk} = \frac{\#F_{jk}}{\#P_{jk}} * 100 \tag{4.1}$$

where $\#F_{jk}$ represents the number of foreground pixels in a given cell (j, k) whereas $\#P_{jk}$ is the total amount of pixels of the cell itself.
It is important to highlight that the dimensions of the matrix (M and N) are fixed parameters and the matrix is stretched in order to cover all the face area. In other words, $\#P_{jk}$ can vary from one image to another for the same cell (j, k): by defining B_{jk} as percentage is therefore possible to compare different images.
Once each B_{jk} is computed, we build an $M \times N$ descriptor vector which summarizes the coverage offered by the edges of the whole image. Similarity between a sketch and a photo is determined by calculating the Euclidean distance between the respective SM descriptors.
Even if we use the above described version of Shape Matrix, in order to investigate



(a) (b)

Figure 4.6: Shape Matrix: face edge image (a) and the superimposed $M \times N$ matrix (b).

benefits of such feature we analyzed also some other variants. One of them is computed by employing weights associated to each cell of the matrix. In particular, the application of weights allows to assign distinct importance to different face components. For example, by assigning higher weight to the central cells of the matrix it is possible to confer more relevance to nose, mouth and eyes rather than other components like hair, ears and neck during the following matching phase. Weights are defined in the range [0...1] and are inversely proportional to

the distance from the center of the matrix.
Another different form of such feature can be computed by introducing a coverage threshold Th, thus modifying Eq. 4.1 as follows:

$$B_{jk}^{Th} = \begin{cases} 1, & \text{if } B_{jk} \geq Th \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

In this case, the resulting feature vector is binary thus allowing the use of the Hamming distance for feature matching rather than the Euclidean distance as in the previous versions.

4.4.2 Beam Angle Statistics

Beam Angle Statistics descriptor (BAS in the following) was originally studied in [9] and allows to describe curvature of object boundary, in an invariant fashion respect to translation, rotation, scale and presence of noise. The angles considered to compute such feature are defined by a reference point located on the contour and other sample points located on the same contour.
Let's consider the shape represented in Figure 4.7 and let B be the shape contour.

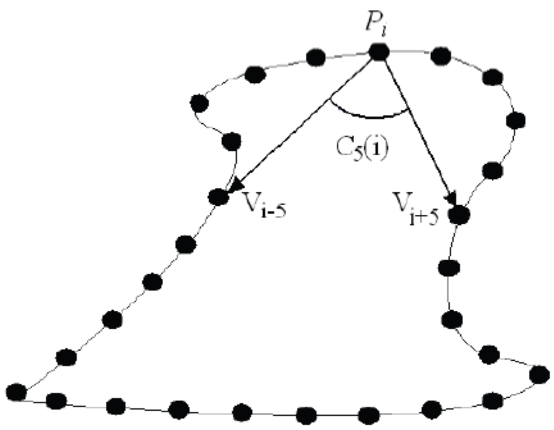


Figure 4.7: Beam Angle Statistics: example of beam angle of order $k = 5$ (figure from [9]).

B can be seen as a connected sequence of N points $P_i = (x_i, y_i)$ such that:

$$B = \{P_1, P_2, ..., P_N\} \tag{4.3}$$

For a given point, it is possible to trace a segment which connects the point to the other ones. The segment connecting the point P_i with the point placed k positions forward it is called *forward beam vector* of order k:

$$V_{i+k} = \overrightarrow{P_i P_{i+k}} \tag{4.4}$$

Similarly, we can define as *backward beam vector* the segment which connects the point P_i with the point located k positions backwards:

$$V_{i-k} = \overrightarrow{P_i P_{i-k}} \tag{4.5}$$

Defined the vectors V_{i+k} e V_{i-k}, the beam angle $C_k(i)$ of order k related to the point P_i is computed as:

$$C_k(i) = (\theta_{V_{i+k}} - \theta_{V_{i-k}}) \tag{4.6}$$

where:

$$\theta_{V_{i+k}} = arctan(\frac{y_{i+k} - y_i}{x_{i+k} - x_i}) \quad \theta_{V_{i-k}} = arctan(\frac{y_{i-k} - y_i}{x_{i-k} - x_i}) \tag{4.7}$$

In this work, BAS is used to describe a face shape by considering only the external contour. Computation of BAS is first done by sampling some of the points located on the face boundary (Figure 4.8). At each point P_i, BAS calculates angles that P_i forms with the forward k points and with the backward k points. These angular values are then respectively subtracted. The complete descriptor consists of k values corresponding to the angular moments of order $1, 2, ..., k$. The left half and the right half of a face are separately processed to make the descriptor more robust with respect to the shape irregularity; so, the final descriptor is obtained as a concatenation of two half descriptors.
Similarity between a sketch and a photo is determined by calculating the Euclidean distance between the respective BAS descriptors.

4.4.3 Local Orientation Histogram

Local Orientation Histogram descriptor (LOH in the following) synthesizes information about local orientations of face edges. Analogously to the SM feature, also in LOH a $M \times N$ regular matrix is superimposed on the image. The core activity to build LOH descriptor is calculating the directional image for each cell of the matrix. A directional image ([108] [110]) is a matrix defined over a discrete grid,

(a) (b)

Figure 4.8: Face edge image (a) and external face contour sampling (b) used for the computation of Beam Angle Statistics and Fourier Descriptors.

superimposed on the gray-scale image, whose elements are in correspondence with the grid nodes. Each element is a vector lying on the xy plane. The vector direction Θ_{ij} (unoriented and lying in the range $[-90°, +90°[$) represents the tangent to the image edges in a neighborhood of the node and its modulus r_{ij} is determined as a weighted sum of contrast (edge strength) and consistency (direction reliability). Examples of directional image are reported in (Figure 4.9).
Each directional image element ij is calculated over a local window where a gradient-type operator is employed to extract several directional estimates (2D sub-vectors). Estimated values are averaged by least-squares minimization to control noise. This technique is more robust than the standard operators used for computing the gradient phase angle.
The more simple and natural approach to extract the local orientation is based on the computation of the image gradient. As we know, the gradient $\nabla(x_i, y_j)$ in the point $[x_i, y_j]$ of a generic image I is a two-dimensional vector $[\nabla_x(x_i, y_j), \nabla_y(x_i, y_j)]$, where ∇_x and ∇_y are the derivatives of I in $[x_i, y_j]$ with reference to the directions x e y. The phase angle of the gradient expresses the direction of maximal change of pixel intensity whereas the direction Θ_{ij} is orthogonal to the phase angle in $[x_i, y_j]$.
Unfortunately, the employment of the gradient is not enough to compute the directional image, for the following reasons:

- nonlinearity and discontinuity around 90°;
- estimating one single orientation represents an analysis too sensible to noise and on the other hand it is not possible compute an average of gradients because of the circularity of angles;

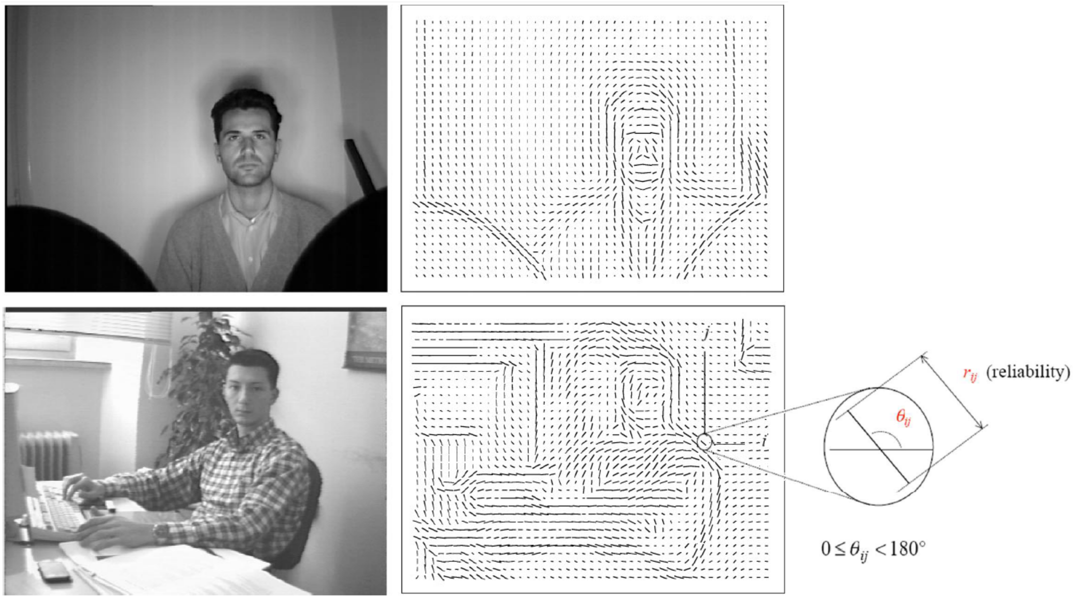


Figure 4.9: Examples of directional image (figure from [108]).

- the concept of average orientation is not well defined (for example, average between 5° and 175° will produce an angle of 90° even if the most correct value would be 0°).

One elegant and simple solution to overcome the problem of angle circularity is doubling the angle itself. Each element of the directional image D is coded by the following vector:

$$d_{ij} = [r_{ij} \cdot \cos 2\theta_{ij}, r_{ij} \cdot \sin 2\theta_{ij}] \tag{4.8}$$

where r_{ij} is the modulus of the vector with orientation Θ_{ij}.
The directional image is computed by splitting the input image by means of $n \times n$ windows. Therefore, each single element of the directional image is obtained as an average of the local orientations inside the considered window. Such average element can be computed by considering separately the two components x and y:

$$\bar{\mathbf{d}} = [\frac{1}{n^2} \sum_{i,j} r_{ij} \cdot \cos 2\theta_{ij}, \frac{1}{n^2} \sum_{i,j} r_{ij} \cdot \sin 2\theta_{ij}] \tag{4.9}$$

To complete the computation of the directional image in the point $[x_i, y_j]$ it is necessary to determine the dominant orientation Θ_{ij} for each window as follows:

$$\theta_{ij} = 90° + \frac{1}{2} atan2(2G_{xy}, G_{xx} - G_{yy}), \tag{4.10}$$

$$G_{xy} = \sum_{h=-8}^{8} \sum_{h=-8}^{8} \nabla_x(x_i + h, y_j + k) \cdot \nabla_y(x_i + h, y_j + k), \tag{4.11}$$

$$G_{xx} = \sum_{h=-8}^{8} \sum_{h=-8}^{8} \nabla_x(x_i + h, y_j + k)^2, \tag{4.12}$$

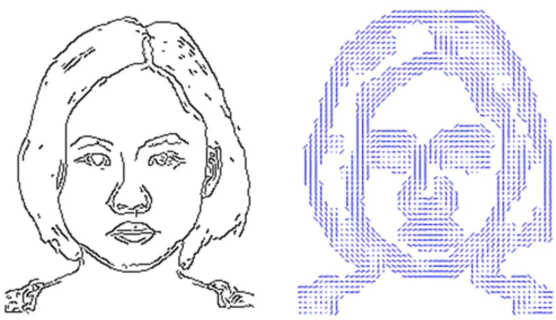
$$G_{yy} = \sum_{h=-8}^{8} \sum_{h=-8}^{8} \nabla_y(x_i + h, y_j + k)^2 \tag{4.13}$$

where ∇_x and ∇_y are the gradient components. Finally, the direction reliability r_{ij} of each orientation is:

$$r_{ij} = \frac{\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}{G_{xx} + G_{yy}} \tag{4.14}$$

The directional image starting from face edges is shown in Figure 4.10. Each directional element is characterized by an orientation between 0° and 180°. This range is discretized in k intervals; each orientation value is discretized to the closest interval. To describe in a compact way all the orientations, LOH involves the construction of a histogram with k bins: the l-th bin is associated to the number of directional elements falling in the l-th interval. Repeating this procedure for each cell and concatenating all of the $M \times N$ histograms bins, we obtain a single descriptor for the whole face.

Finally, as we deal with histograms, the similarity between a sketch S and a photo



(a) (b)

Figure 4.10: Local Orientation Histogram: face edge image (a) and its directional image (b).

P is determined by calculating the Kullback-Leibler divergence $\mathcal{K}$ between the respective LOH descriptors as follows:

$$\mathcal{K}(H_S, H_P) = \sum_{i=1}^{d}(H_S(i) - H_P(i))ln\frac{H_S(i)}{H_P(i)} \tag{4.15}$$

4.4.4 Fourier Descriptors

Fourier Descriptors feature vector (FD in the following) is quite divergent from the other shape descriptors we adopt for our candidate photo selection. As recalled by its name, FD is a feature computed by operating in the frequency domain (or Fourier domain) rather than in the spatial (image) domain. The use of Fourier space provide a valiant alternative to the conventional method since it allows to highlight appearances that could be hardly highlighted and exploited in the spatial domain. There are important applications of FD in the field of shape representation and we will apply such idea to give a description of the face boundary.
In its original version designed by Zhang and Lu [200][199], FD feature is obtained by applying the Fourier transform to the so-called *shape signature*, a one-dimensional function that represents the contour of a shape. In many practical applications, the shape signature is the centroid function, *i.e.* the distance of the contour points from the centroid (x_c, y_c) of the object (see an example in Figure 4.11).
Considering as $(x(t), y(t))$ the coordinates of the N points placed on the shape contour, the centroid function is defined as follows:

$$r(t) = [(x(t) - x_c)^2 + (y(t) - y_c)^2]^{\frac{1}{2}}, \quad t = 0, 1, ..., N-1 \tag{4.16}$$

where

$$x_c = \frac{1}{N}\sum_{t=0}^{N-1} x(t), \quad y_c = \frac{1}{N}\sum_{t=0}^{N-1} y(t) \tag{4.17}$$

Computing the Fourier transform of the centroid function $r(t)$ we get coefficients

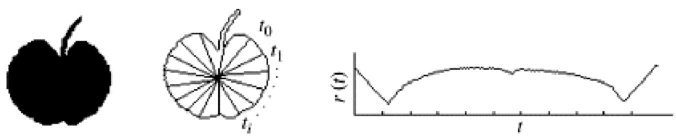


Figure 4.11: Example of centroid function computed from a simple shape (figure from [200]).

that we call Fourier Descriptors. The strength of this feature is that only few coefficients at low frequency are really meaningful to give an effective description of a shape whereas high frequency coefficient can be ignored, as proved in [199]. As shown in Figure 4.12, it is possible to carry out shape reconstruction with a different number of Fourier coefficients, but only a reduced number of low frequency coefficients are required to faithfully reproduce the object shape. Therefore, by applying Fourier transform to the centroid function we can produce a small but expressive amount of values thus building a compact feature vector that effectively describes the object.
Each coefficient a_n is computed as follows:

$$a_n = \frac{1}{N} \sum_{t=0}^{N-1} r(t) \exp\left(\frac{-j2\pi nt}{N}\right) \quad n = 0, 1, ..., N-1 \tag{4.18}$$

a_n is a complex number with a modulus and a phase. In our application, we consider only the modulus thus building a feature vector with values corresponding to low frequencies. Moreover, the coefficients are translation invariant due to the translation invariance of the shape signature. In order to describe the shape, the computed Fourier coefficients have to be further normalized with reference to the first coefficient a_0 so that we can get rotation, scaling and start point invariant feature descriptors [199].
In our proposed version, FD descriptor is computed starting with a point sampling

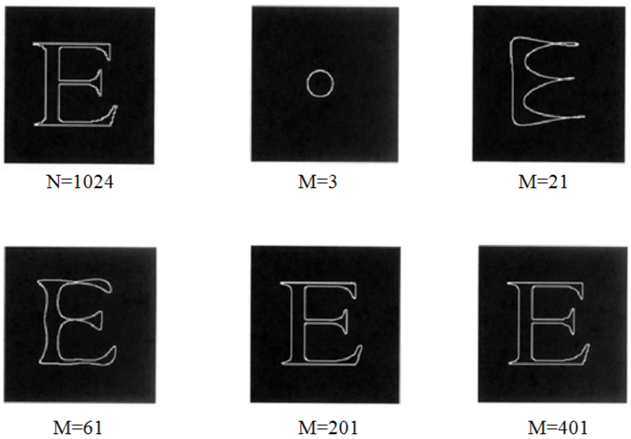


Figure 4.12: Shape reconstruction with a different number of Fourier coefficients.

on the face boundary, analogously to the procedure used for BAS (Figure 4.8). Then we compute a one-dimensional function called *shape signature*: for each sampled point on the face boundary, the normalized Euclidean distance with respect to the face centroid is determined. Conventionally, the centroid is located

at the center of the two eyes.
Such shape signature is then transformed to the frequency domain. Instead of using the standard Fourier transform, we exploit the Discrete Cosine Transform (DCT) that we found more computationally efficient without loosing in accuracy:

$$a_n = \sum_{t=0}^{N-1} r(t) \cos\left[\frac{\pi}{N}\left(t + \frac{1}{2}\right)n\right] \qquad n = 0, 1, ..., N-1 \tag{4.19}$$

Similarity between a sketch and a photo is determined by calculating the Euclidean distance between the respective FD descriptors.

4.4.5 Pixel Decimal Value

Pixel Decimal Value (PDV in the following) is a very simple but effective feature. PDV descriptor is computed by assigning to each pixel a decimal value; such value is obtained by a weighted sum of the pixels in its 8-neighbourhood.
Considering an image of $W \times H$ pixels, the final PDV descriptor will consist of $W \times H$ values, one for each image pixel. The value associated to each pixel is computed as follows: a weight, corresponding to an increasing power of 2 (see Figure 4.13), is assigned to each pixel of the 8-neighborhood. Then, the final value is obtained by summing the weights assigned to the foreground pixels (*i.e.* the black pixels in the edge image).
Similarity between a sketch and a photo is determined by calculating the Euclidean distance between the respective PDV descriptors.

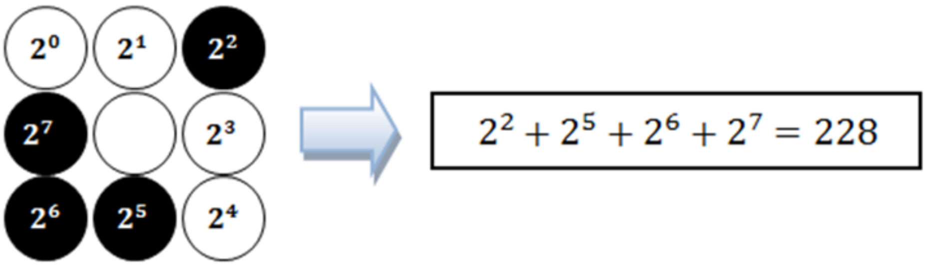


Figure 4.13: An example of Pixel Decimal Value descriptor computation.

4.4.6 Local Binary Pattern

Local Binary Pattern feature (LBP in the following) was originally introduced with the aim of providing a description of images texture [123]. Thanks to its discriminative power, to its computational simplicity, in addition to the robustness with respect to variations in illumination, LBP has soon become a very popular approach in various face recognition applications. In contrast with the common use of LBP in the context of face recognition, in our proposed approach the LBP descriptor is not computed on original sketches and photos but only on their edged images (in accordance with other illustrated shape features).
The basic version of LBP operator assigns to each pixel of the image a binary value depending on the composition of the neighborhood of the pixel itself. Let p_c be a pixel and p a pixel of its 3×3 neighborhood as represented in Figure 4.14. If p has a value greater or equal to p_c then the value 1 is assigned to p, otherwise

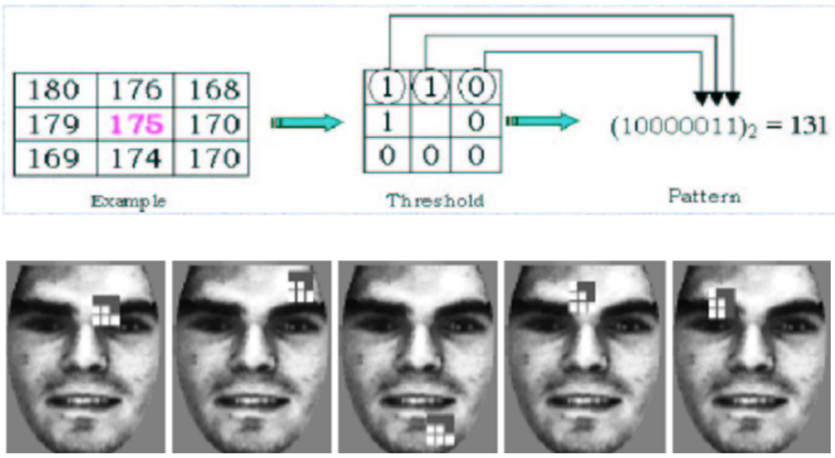


Figure 4.14: LBP computation on a grayscale image of a face (figure from [168]).

0. Collecting such binary digits as a sequence, we get a binary number of 8 digits: the correspondent value in the decimal system represents the final value associated to the reference pixel p_c. Such procedure is similar to PDV feature computation, but differently from PDV here we consider the difference between pixels (and not their belonging to foreground (edge) or background).
Such basic version of LBP has an inconvenience related to the size of the neighborhood, since it would be required to use a fixed or variable size depending on the specific application. For this reason, LBP operator has been extended in order to handle neighborhoods of different sizes. In particular, in the new version a circular neighborhood has been introduced thus replacing the previous illustrated 3×3 mask, as represented in Figure 4.15. When using a circular neighborhood,

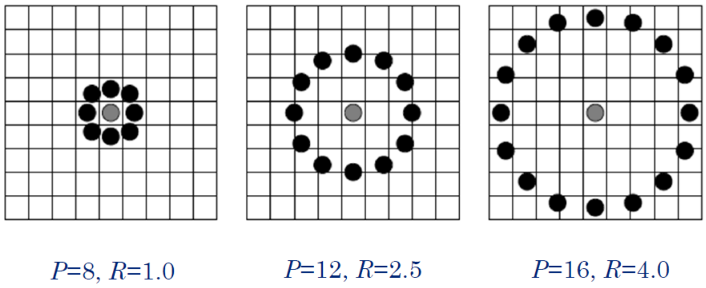


Figure 4.15: Circular neighborhood to compute LBP (figure from [168]).

two new parameters have to be considered: the number of sample points P and the radius R of the circle. Furthermore, interpolation is used to infer the value of a sample point that is placed across multiple pixels. After the definition of parameters P and R, the previously illustrated method is used to associate to a given pixel p_c its corresponding binary value, taking advantage of the interpolation to determine the grayscale value of undefined pixel.

The binary patterns 0-1 that we obtain computing LBP can be classified as *uniform* and *non-uniform*. In Figure 4.16 we can appreciate some examples of uniform patterns. A pattern is uniform when it contains up to two 0-1 transitions (*e.g.* ,

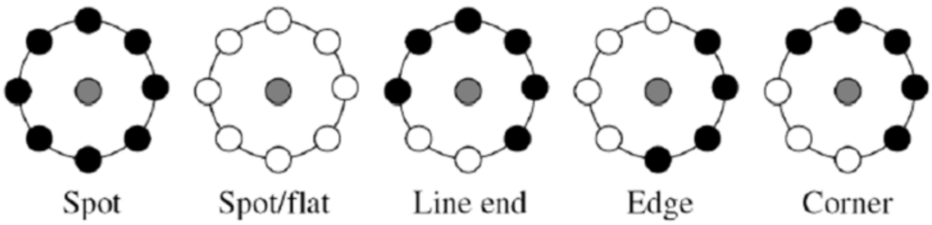


Figure 4.16: LBP: examples of uniform patterns (figure from [168]).

the patterns 10000011, 11110000, 00000000 are uniform). Uniform patterns are important because considering only them it is possible to obtain a large save of memory during processing. Indeed, the number of all possible patterns is 2^P but the number of all possible uniform patterns is only limited to $P(P-1)+2$. To better understand the meaning, we can look at Figure 4.17: considering a grayscale face image, the number of uniform patterns (represented as white dots) is largely smaller than the number of non-uniform ones.

Computing LBP for the entire image leads to a feature vector that can be represented as a histogram. The feature vector is obtained by dividing the image in k^2 sub-windows and for each sub-window we build a histogram where bins take into account the number of occurrences of different uniform patterns. A histogram is therefore composed by $P(P-1)+3$ bins: $P(P-1)$ bins for patterns with 2

(a) (b) (c)

Figure 4.17: LBP: uniform patterns (b) and non-uniform patterns (c) from a grayscale face image (figure from [168]).

transitions, 2 bins for patterns with no transitions and only 1 bin to take account of non-uniform patterns. Histograms from each sub-window are finally normalized and concatenated thus obtaining the final image-related histogram, as illustrated in Figure 4.18.

As the reader should have noticed, previous examples refer to grayscale images of

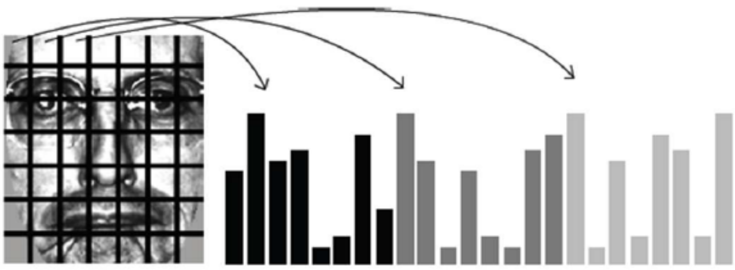


Figure 4.18: LBP: histograms which compose the final feature vector (figure from [168]).

the face. In our sketch recognition scenario, LBP is instead applied to edge images of the face subject, accordingly to other mentioned shape features. Being LPB quite sensitive to image texture and to noise presence, it is necessary to introduce a new parameter: a grayscale threshold. The main goal of such threshold is filter out those pixels that even if not completely white can be anyway considered as background, thus not involving them in the LBP computation as foreground.

As mentioned before, the final feature vector is obtained by concatenating histograms from each sub-window. Even if we have already defined a histogram distance to determine similarity between a sketch and a photo (see Subsection 4.4.3), we employ here the Euclidean distance between LBP descriptors. Indeed, such descriptors are composed by several thousands of elements thus making histogram distance computationally demanding and inefficient.

4.5 Feature fusion for candidate photo selection

In our system we aim at comparing a sketch of an unknown subject with a mug shot photo gallery of already recorded subjects. The main purpose of such comparison is returning a candidate photo list of the most similar photos to the given sketch. Such candidate list can be used either *as is* by police enforcements in order to carry out a rough and manual evaluation of possible suspects or employed as input data for a more accurate and conclusive recognition (as we will see in Section 4.8). In this latter scenario, the system could produce the final recognition result in a shorter time since only a reduced set of photos instead of the whole gallery has to be analyzed.

As illustrated in Section 4.4, we have at our disposal six distinct shape features. Given an input sketch, each single feature can scan the gallery separately thus returning a ranking of the most similar photos according to specific similarity measure. On the other hand, we need to define a fusion technique in order to merge partial results provided by each shape feature and return the final candidate list according to all shape features.

Literature provides different different methods for features fusion [168], as reported in the list below.

- Fusion at feature level: descriptor vectors concerning each single shape features are combined to obtain a unique final vector which describes the subject face. Features fusion is carried out before the matching phase.
- Fusion at score level: different matching algorithms are used for each features thus returning different scores. Scores are then combined to generate the final global score.
- Fusion at rank level: each shape feature provides its own photo ranking based on the reference sketch and then rankings are fused to obtain the final global ranking.
- Fusion at decision level: each feature expresses a "preference" (vote) about the most similar photo to the given sketch and then preferences are combined to get the finale vote for each photo.

Potentially, all the above mentioned techniques are well suited for our purpose, but the most interesting one is the fusion at rank level. Indeed, we want to obtain a ranking of the most similar photos to the given sketch, expecting the correct subject to be in the top positions of the ranking. This allows the creation of a candidate list, where only the top-N results are selected either for further refinements or for the final recognition. Under this point of view, the best fusion technique turns out to be the *borda count* [180].

When adopting borda count, the final ranking is obtained by combining the rankings provided by each single shape feature (an overview of the general idea is reported in Figure 4.19). A decreasing vote is assigned to the first $bCount$ images in each ranking: if we have $bCount$ photos that must be evaluated, the photo that results to be ranked in position j according to a certain shape feature will get a score number equal to $N - j$. Effects of the parameter $bCount$ on the accuracy are evaluated in Section 4.7.

Single rankings are fused with a straightforward sum of scores for each rank position. Moreover, since shape features have different level of reliability, we employ weights to assign distinct levels of importance to each ranking. This is a common strategy in multi biometric systems where weights are experimentally determined according to single classification accuracy of each biometric trait. As reported in Figure 4.19, we experimentally determined BAS, FD and LOH as shape features with higher weights since they proved to be more accurate than the others.

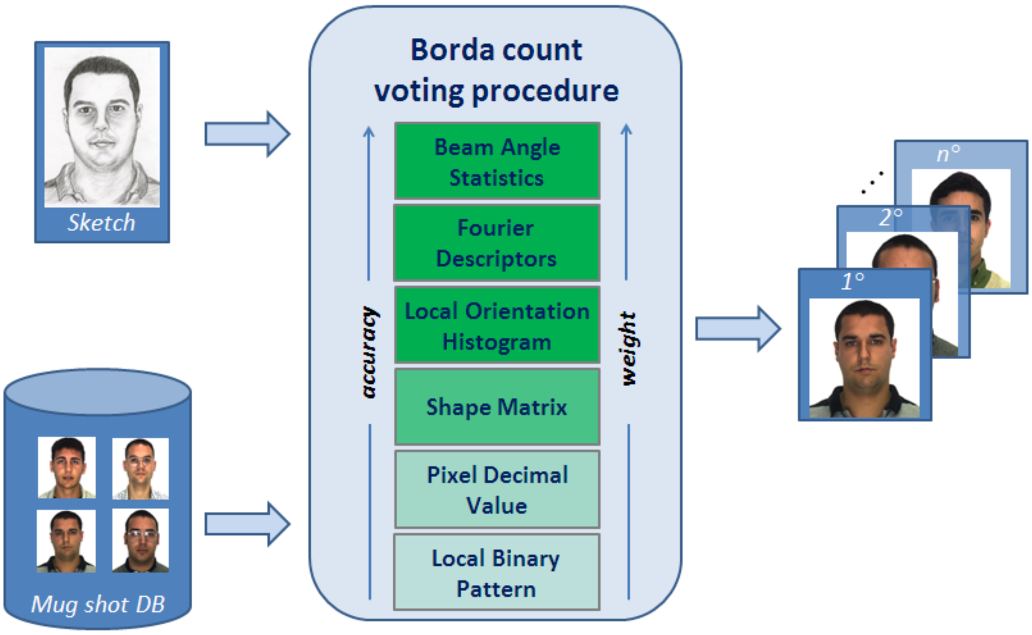


Figure 4.19: Candidate photo selection overview: feature fusion is carried out through borda count voting procedure.

4.6 Sketch and photo datasets

As illustrated in Section 4.2, face recognition from sketch is a relatively new and peculiar research area in the wider field of face recognition. Because of such novelty, finding proper datasets for performance evaluation turns out to be not a so easy task. On one hand, sketches of various subjects are necessary together with real mug shot photos of the same subjects to perform true and false matching. On the other hand, large galleries of mug shot photos are required in order to simulate real scenarios with at least several thousands of potential candidates. Since both sketches and mug shot photos are jealously and secretely kept by law enforcement agencies, for our purpose we must rely on publicly released databases containing photo-sketch pairs.

With reference to publicly available sketches, it is important to distinguish between three categories: *viewed* sketches, *semi-forensic* sketches and *forensic* sketches. Such sketches are produced by means of different procedures:

- viewed-sketches are drawn by a professional artist while looking directly at a person (or at a real photo of the person itself);
- semi-forensic sketches are realized by a professional artist who first looks at a person (or at a real photo of the person itself) and after a time laps of some minutes draws the portrait by relying only on his/her memory;
- forensic sketches are realized by a forensic artist based on the description of an eyewitness by relying on his/her recollection of the criminal event.

Because of their nature, viewed sketches stand out to be good quality images which look quite similar to real photos. On the contrary forensic sketches are picked up from crime investigation activity and turn out to be very challenging. Some instances of forensic sketches are affected by a very poor quality thus making the recognition very challenging also for a human eye. Finally, semi-forensic sketches simulate the forensic context since the sketch artist is not allowed to view the digital image while preparing the sketch.

For our experimental evaluation, we employ 188 viewed sketches from CUHK [184] and 123 from AR [184] (for a total of 311 viewed-sketches), 65 semi-forensic sketches from IIITD datasets [20] and 50 real mug shot photos collected from the web (Table 4.2). In Figure 4.20, the reader can visually appreciate different

examples of the previously mentioned categories.
With reference to mug shot galleries, it is rather difficult to find public face datasets

Dataset name	Description	Number of elements
CUHK [184]	Viewed sketches	188
AR [184]	Viewed sketches	123
IIITD [20]	Semi-forensic sketches	65
Mug shots	Forensic sketches	50

Table 4.2: Viewed, semi-forensic and forensic sketches used in our experiments.

adapt for this scenario. The previous mentioned sketch datasets include exactly those photos which are associated to sketches: if we compose our gallery only with such photos, we would get a gallery made up of $\sim$ 400 elements, therefore not suitable to carry out large scale tests. Taking into account such issue, we created a mixed database of photos, collecting well-controlled and mug shot-like images from various sources: 188 images from CUHK [184] and 123 from AR [184] (paired to viewed sketches), 65 images from IIITD (paired to semi forensic sketches) [20], the 50 real mug shot photos associated to forensic sketches plus photos collected from other datasets. In particular we took 114 photos from CVL [176], 100 from PUT [88], 1194 from FERET [203], 6387 from FRGC [127] for a total of 8221 mug shot-like photos in our gallery (Table 4.3).

Dataset name	Corresponding to sketch ?	Number of elements
CUHK [184]	Yes	188
AR [184]	Yes	123
IIITD [20]	Yes	65
Mug shots	Yes	50
CVL [176]	No	114
PUT [88]	No	100
FERET [203]	No	1194
FRGC [127]	No	6387

Table 4.3: Mug shot photos from different datasets we used to compose our gallery.

(a)

(b)

(c)

Figure 4.20: Different types of sketch (second row), with associated mug shot photos (first row): (a) viewed sketches from CUHK and AR datasets [184] , (b) semi-forensic sketches from IIITD dataset [20] and (c) forensic sketches collected from the web.

4.7 Candidate photo selection performance

In this section we report accuracy of sketch-based photo retrieval from the gallery. In particular, we employ our six shape features (see Section 4.4) to scan the gallery linearly and we build a final candidate set according to rank-level fusion described in Section 4.5.

All the parameters related to shape features have been optimized on a disjoint training set and the same values have been used in all the tests reported in this section. Considering the aim of the proposed technique, the performance are reported in terms of the percentage of the gallery that must be considered to retrieve 95%, 99% and 100% of the real photos associated to the query sketches.

Some preliminary tests have been carried out on the CUHK subset of sketches to evaluate the performance of the proposed technique as a function of the parameter *bCount* representing the number of photos to vote in order to produce the final ranking. The results obtained are summarized in Table 4.4. The results show that increasing the value of *bCount* determines a slight increment of the percentage needed to find 95% of the real photos, but significantly reduces the values related to 99% and 100%. We found that 5000 represents a good choice and such value will be used in the subsequent experiments.

Further tests have been executed to evaluate the behavior of the proposed tech-

bCount	*95%*	*99%*	*100%*
1000	3.68%	32.76%	32.84%
2000	3.97%	24.53%	35.92%
3000	4.79%	20.35%	23.95%
4000	5.02%	13.70%	24.25%
5000	5.60%	12.27%	20.59%
6000	5.53%	15.41%	19.42%
7000	6.83%	16.62%	19.09%
8000	6.83%	16.58%	19.01%

Table 4.4: Percentage of data to consider to retrieve a given percentage (95%, 99% and 100%) of the photo associated to the test sketches, as a function of *bCount*.

nique with different kind of sketches. In Figure 4.21 the percentage of correct photos retrieved as a function of the percentage of database considered is reported for different sets of sketches: CUHK and AR, IIITD and Forensic. The results clearly show that the first set contains easier images, while the difficulty in retrieving the correct photo increases when semi-forensic or forensic sketches are considered, as also highlighted in many works in the literature. Despite of the complexity of the task, the proposed shape features proved to be effective in reducing the search space.

The scalability of the proposed technique has been analyzed by evaluating the per-

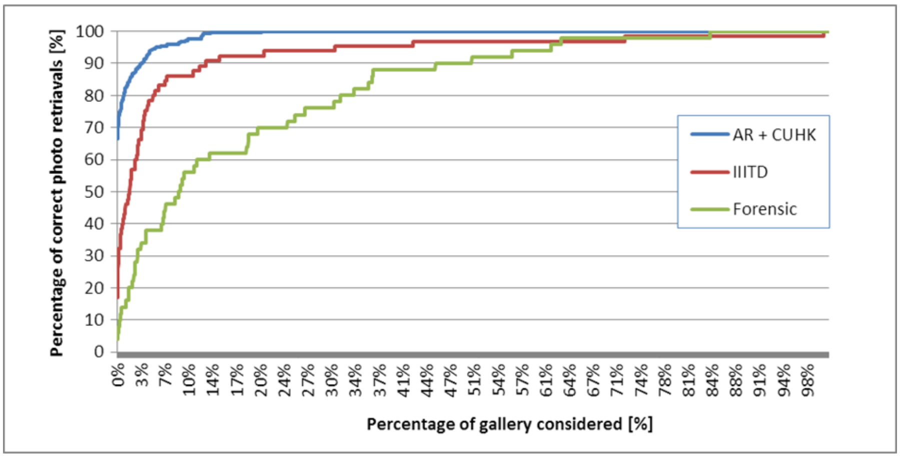


Figure 4.21: Percentage of correct photos retrieved as a function of the percentage of gallery considered for the AR+CUHK, IIITD and Forensic sketches.

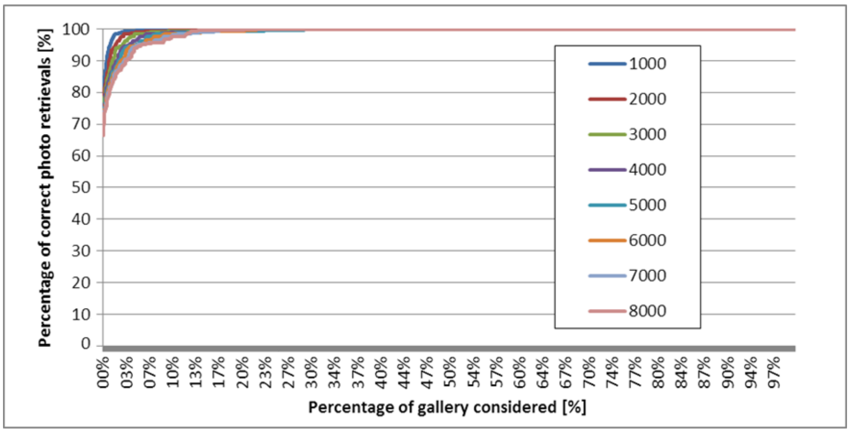


Figure 4.22: Percentage of correct photos retrieved as a function of the percentage of gallery considered. The results are reported for increasing size of the gallery (from 1000 mug shot to 8000).

formance as a function of the gallery size (see Figure 4.22). The results obtained are encouraging: as expected, a smaller gallery produces better results, but overall the performance are not significantly affected by the increment of number of mug shot photos. As to the efficiency of the proposed technique, the average time needed to compute the shape features for a single image is about 131 *msec.*, while matching the descriptors of two images takes about 1 *msec.* on a PC Intel(R)

Xeon(R) CPU E31245 @ 3.30 GHz.

Finally, as qualitative result, we provide visual examples in Figure 4.23 of rankings obtained for different input sketches. In these examples, the real photo pairs are retrieved in the first position, but it is worth highlight also that the subsequent mug shot photos present a visible and clear visual similarity with the input sketches.

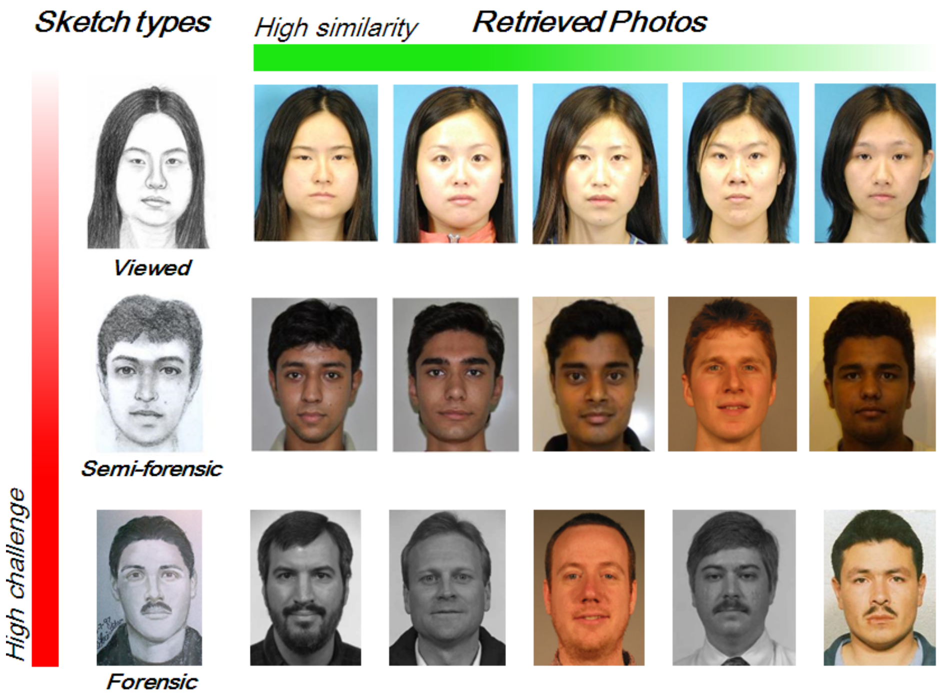


Figure 4.23: An example of the photo ranking provided by our proposed approach: a query sketch is given for each type of sketch (viewed, semi-forensic and forensic) and the five most similar photos according to our algorithm are reported.

4.8 Fine-grained recognition on the candidate set

In Section 4.7 we reported accuracy in face photo retrieval given an unknown sketch and a gallery of photos as input. Starting from the retrieved photos, it could be possible to execute a further step to produce the final recognition result. This can be done manually through a visual comparison or automatically, as we show next, through a fine-grained recognition technique. In other words, the recognition pipeline is made up of two steps: first we employ shape features to filter out photos which do not look like the probe sketch, then we use a suitable

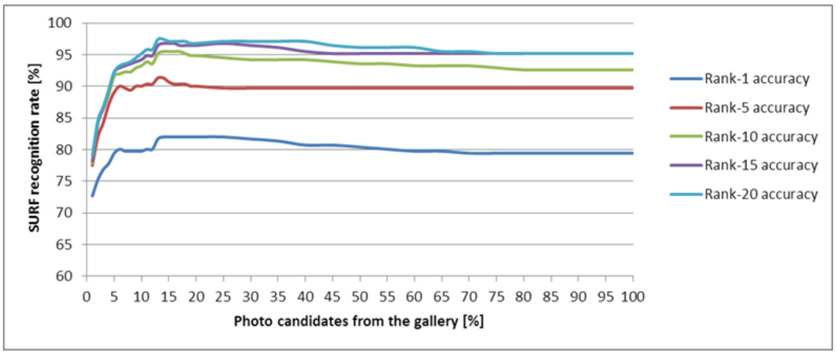
recognition technique to produce the final result by taking into account only the candidate set built in the first step.
Such pipeline provides two important processing benefits. On one hand, the first step can be performed in a relatively short time thanks to the low computational demand of shape features, thus allowing a fast preliminarily scan of the entire gallery. On the other hand, fine-grained recognition can be carried out through a more computationally demanding and accurate technique thus focusing only on a very reduced set of elements (photo candidates) without paying longer processing time that would be instead required if using the same fine-grained technique on the whole gallery. What needs to be evaluated now is the accuracy drop - if any - when combining shape features and fine-grained technique instead of using the fine-grained technique only.
Some techniques we could adopt here have been illustrated in Section 4.2, *e.g.* [20, 72, 91]. Unfortunately, code is not publicly available and their implementation from scratch would be prone to errors. However, in [72, 91] SIFT-like descriptors emerged to be quite effective for the purpose of face recognition from sketch. For these reasons, we adopt Speeded-Up Robust Feature (SURF) descriptors as fine-grained algorithm for a quantitative evaluation.
In the following, we show recognition rate accuracy when considering an increasing percentage of the indexed photos for a given face sketch. The photo gallery and probe sketches are the same described in Section 4.6: 8221 mug shot-like photos and 3 different type of sketches (viewed from CUHK and AR, semi-forensic from IIITD and forensic collected from the web). Given a sketch of an unknown subject that has to be identified, candidate photo selection is first carried out by using our shape features as illustrated in Section 4.5. Fine-grained recognition is then performed through SURF descriptors by taking into account only a reduced percentage of the total retrieved and ordered photos.
Fine-grained recognition accuracy is reported in different charts of Figure 4.24 for viewed, semi-forensic and forensic sketches. Rank-1, Rank-5, Rank-10, Rank-15 and Rank-20 SURF recognition accuracy are showed by varying the considered portion of retrieved and ordered data (as percentage of the entire gallery). Even though the three sketch classes are marked by very different levels of challenge and not comparable recognition rates, it is proved that performing the fine-grained recognition technique on a very reduced set of photos yields to better recognition results rather than considering the entire gallery without any candidate selection (*i.e.* 100% portion of data). In particular, the graphs seem to agree in identifying

approximately at 10% the best portion of retrieved and ordered data within which expect the true photo associated with the given sketch, thus proving effectiveness of our shape features-based candidate selection method.
Shape features allow a fast scan of the gallery since shape descriptors matching is 25 times faster than SURF descriptors matching on the same computing platform (whereas time required to compute six shape descriptors is comparable to the time required to compute a single SURF descriptor) and they are also effective in filtering out those photos which can negatively affect recognition accuracy of a fine-grained algorithm, thus helping good candidates to emerge.

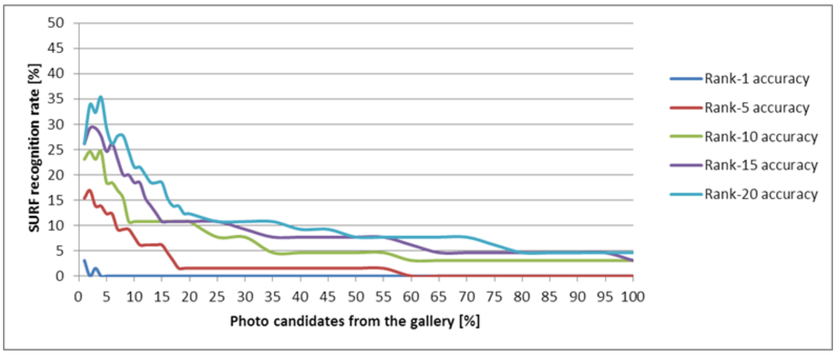
4.9 MKL-based indexing structure for sketch recognition

4.9.1 Indexing structures on large-scale image databases

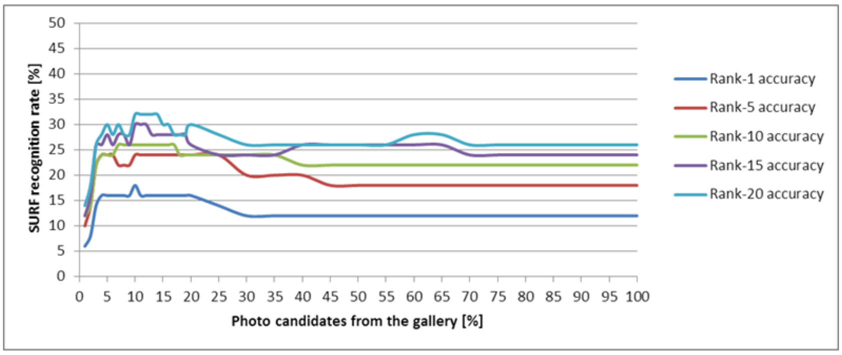
Recognition speed is negligible when a photo gallery includes only few hundreds of samples as in publicly available research-purpose datasets. However, it becomes a critical problem with real mug shot galleries, where a potentially huge number of candidates is represented (*e.g.* , the gallery we built for our experiments and described in Section 4.6). As mentioned in Section 4.2, to the best of our knowledge only few state-of-the-art methods provide performance results with such very large galleries [20, 72, 93]. Moreover, photo indexing techniques capable to reduce search space for sketch recognition are still missing.
A first solution to this problem has been proposed in Section 4.5. In particular, we employed efficient shape features to preliminary scan the gallery and filter out misleading subjects for a given sketch. More specifically, in order to shrink the search space, the sketch is compared against each photo with a linear scan. Shape descriptors similarity is then computed thus producing a candidate list of the most similar subjects. In this section, we extend such idea by proposing an MKL-tree indexing method [59] in order to reduce the search space when a new sketch (query) has to be matched against a gallery of photos. As before, we use shape features to describe both sketches and photos thus overcoming the modality gap. However, instead of employing shape features to perform a linear search on the entire gallery, here we illustrate an MKL-based indexing schema where most



(a)



(b)



(c)

Figure 4.24: Recognition rate using SURF on a) viewed sketches (CUHK and AR), b) semi-forensic sketches (IIITD) and c) forensic sketches. Rank-1, 5, 10, 15 and 20 are reported as a function of the percentage of candidate photos retrieved through shape features analysis.

similar photos are retrieved in a time that is near constant with increasing gallery size, by taking advantage of trees and dimensionality reduction properties.

Efficient and accurate image retrieval through proper indexing structures on large-scale image databases has been widely explored during years yielding to countless approaches. A complete review of such techniques is out of the scope of this document; in the following we focus our attention to indexing techniques for face photo matching with special attention to forensic contexts [81], which are small but challenging subsets in the field of general-purpose image retrieval.

Despite great effort has been spent to create indexing techniques related to other biometrics features such as fingerprint [18, 32], iris [51] and multi biometric patterns [71], only few indexing approaches has been introduced as support for face recognition on large galleries. Bag of Words technique [46, 151] [120], which constitutes the state-of-the-art for many image retrieval systems, showed recognition drawbacks when applied to faces because of low capabilities in handling intraclass and interclass variations [120]. A more reliable and inspiring method has been presented in [192] for celebrity face retrieval where local facial features are employed for a first index-based scan of a web-scale photo gallery and then global features are used to re-rank the previous returned candidate set, thus offering a scalable and accurate system. Furthermore, recently introduced hashing-based indexing [76] has been proposed for face recognition [86, 89] on common available photo datasets but to the best of our knowledge no indexing techniques have been explored for face recognition from sketch where heterogeneous matching between sketch and photos has to be handled.

Another relevant aspect related to face indexing is represented by the dimension of feature descriptors. In face recognition literature, many different discriminative features have been shown to describe faces [79, 97, 124, 157, 189, 190]. However, their high dimensionality makes them not suitable to build state-of-the-art indexing structures as R-trees and k-D trees [70, 77, 182] since they are affected by the *curse of dimensionality* [16]: the algorithm does not scale well to high-dimensional data, typically due to needing an amount of time or memory that is exponential in the number of dimensions of the data. This problem is usually handled by applying a dimensionality reduction technique to decrease the number of data dimensions thus allowing indexing with standard techniques. Different strategies for dimensionality reduction have been introduced to deal with static and globally correlated datasets [61, 83] and for real applications where datasets usually do not comply with such conditions [41, 130].

4.9.2 MKL transform

As mentioned previously, the "dimensionality curse" problem is usually dealt with by applying a dimensionality reduction technique: the feature vectors to be indexed are first reduced to a lower dimensionality by means of the Karhunen-Loève (KL) transform [61] and then indexed with a traditional data structure.
The KL transform is, among all the unitary transformations for dimensionality reduction, the one which guarantees the best Euclidean distance preservation. In other words, it minimizes the mean-square approximation error, defined as the mean distance between the points belonging to the training set and their back-projections from the reduced space.
Given $P = \{\mathbf{x}_i \in \Re^n \mid i = 1, ..., m\}$ a set of m, n-dimensional data points belonging to the dataset, the k-dimensional eigenspace associated to P is denoted as $S_P = [\bar{\mathbf{x}}_P, \Phi_P, \Lambda_P]$, where: $\bar{\mathbf{x}}_P$ is the mean vector, $\mathbf{C}_P$ is covariance matrix of P, $\Lambda_P \in \Re^{k\times k}$ is the matrix of the largest eigenvalues of $\mathbf{C}_P$ and $\Phi_P \in \Re^{n\times k}$ (projection matrix) is the matrix of the eigenvectors of $\mathbf{C}_P$ corresponding to the largest eigenvalues. Starting from this formulation we can define the following notions:

- $\mathbf{y} = \Phi_P^T(\mathbf{x} - \bar{\mathbf{x}}_P)$ is the *projection* of $\mathbf{x} \in \Re^n$ into the eigenspace S_P.
- $\mathbf{x}' = \Phi_P\mathbf{y} + \bar{\mathbf{x}}_P$ is the *back-projection*, into the original space, of a vector $\mathbf{y} \in \Re^k$.
- $d_I(\mathbf{y}_1, \mathbf{y}_2) = \|\mathbf{y}_1 - \mathbf{y}_2\|_2$ is the (Euclidean) *internal-distance* between two vectors $\mathbf{y}_1$, $\mathbf{y}_2$ belonging to the same space.
- $d_{FS}(\mathbf{x}, S_P) = \sqrt{\|\mathbf{x} - \bar{\mathbf{x}}_P\|_2^2 - \|\Phi_P^T(\mathbf{x} - \bar{\mathbf{x}}_P)\|_2^2}$ is the *distance-from-space* of a vector $\mathbf{x} \in \Re^n$ from a space S_P.
- The *reconstruction error* of a vector $\mathbf{x}$ is the approximation resulting from the projection/back-projection operations and it coincides with the distance $\mathbf{x}$ from the space S_P : $d_{FS}(\mathbf{x}, S_P)$; this error can be conceived as a measure of the appropriateness of S_P to represent $\mathbf{x}$.
- $\varepsilon(S_P) = E[d_{FS}(\mathbf{x}, S_P)^2] = \sum_{i=k+1}^{n} \lambda_i$ is the *mean-square* error over all the points in P and it corresponds to the sum of the $n - k$ discarded C_P's eigenvalues.

- $\xi(S_P) = \frac{\sum_{i=k+1}^{n}\lambda_i}{\sum_{i=1}^{n}\lambda_i}$ is *percentage mean-square* obtained by dividing the mean square error by the sum of all the $\mathbf{C}_P$'s eigenvalues.

With this formulation at hand, it is possible to introduce an improved version of the KL transform: the multi-space KL transform (MKL) [33]. The MKL transform is a generalization of the KL transform where more subspaces are created to arrange the data points (see Figure 4.25). Each subspace represents a subset of points having similar characteristics, thus allowing more selective features to be extracted. Let $P = \{\mathbf{x}_i \in \Re^n \mid i = 1, ..., m\}$ a set of m, n-dimensional vectors, then for a

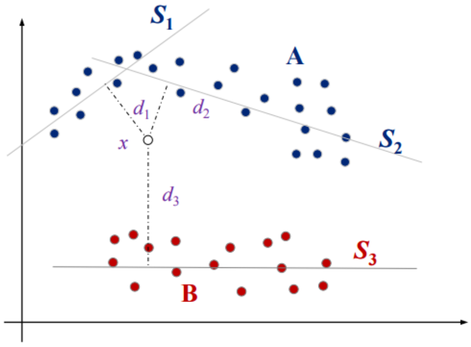


Figure 4.25: Multispace KL transform example: two subspaces (S_1 and S_2) and one subspace (S_3) are used to represent two classes A and B, respectively.

given partition $\mathcal{P} = \{P_1, P_2, ..., P_s\}$ of P and for a given set $K = \{k_1, k_2, ..., k_s\}$ of scalars such that:

- $\bigcup_{i=1,...,s} P_i = P, P_i \cap P_j = \emptyset \quad \forall i, j = 1, ..., s\ , i \neq j.$
- $m_i = card(P_i) \geq \lfloor \frac{m}{s+1} \rfloor \quad \forall i = 1, ..., s.$
- $0 < k_i < min(m_i, n) \quad \forall i = 1, ..., s.$

The MKL transform is defined by the set of subspaces $S = \{S_1, S_2, ..., S_s\}$ where S_i is the eigenspace of reduced dimensionality k_i obtained from the training subset P_i. The KL transform represents a particular case of MKL, where $s = 1$, $\mathcal{P} = \{P\}$ and $K = \{k\}$. A huge number of MKL transforms may be derived from the same initial set P, by varying s, $\mathcal{P}$ and K; we denote with *MKL solution* a triplet $(s, \mathcal{P}, K)$. The partition $\mathcal{P}$ is obtained by means of an *ad hoc* heuristic algorithm aimed to minimize the percentage mean-square recognition error $\xi(s, \mathcal{P}, K)$, defined as the weighted sum of the percentage reconstruction errors related to each KL subspace S_i:

$$\xi(s, \mathcal{P}, K) = \frac{\sum_{i=1}^{s}(m_i \cdot \xi(S_i))}{m} \tag{4.20}$$

4.9.3 MKL-tree and range search

MKL-tree is a disk-based hierarchical structure for n-dimensional points, where data are stored in the leaves, while internal nodes are used to route the search. Nodes correspond to disk blocks and represent the set of objects which are indexed by the corresponding subtree. MKL-tree is a dynamic structure: node representation is updated as a new data point is inserted, in order to give an improved approximation of the corresponding data subset. The tree is height-balanced: all paths of root-leaf have the same length h. Nodes are divided into two categories: internal nodes, containing a representation of their children and a pointer to the corresponding disk block, and leaves, containing representations and pointers to the indexed objects.

A KL subspace of the original space is associated to each node, root excepted. The KL representation of the root is never calculated, since it is not useful to drive the search. Each element of a leaf node consists of the projection into the corresponding KL subspace of an indexed object and of the pointer to the disk block in which the object is stored. Each element of an internal node corresponds to the KL subspace associated to a child node, its mean-square error and the pointer to the disk block in which its child is stored.

The KL subspace associated to each (internal or leaf) node is the subspace that better represents the points in the corresponding subtree (*i.e.* , the subspace that guarantees the minimum reconstruction error for the points stored in the leaves of the subtree). As to the leaves, the related subspace is simply calculated starting from the corresponding data points, whereas in the internal nodes, the subspace is determined by means of a "merging" procedure [58] which creates a representative space starting from the KL subspaces associated to its children. However, in both cases the space associated to a node is characterized by a dimensionality k, far lower than the dimensionality n of the indexed data ($k << n$). The k value may be constant for the whole tree or may vary among different nodes (for instance, it could be low at leaf level and increase when moving toward the root).

The overall space required for storing a node element is $(4 \cdot ((k+1)(n+1)+1))$ bytes, where we consider 4 bytes for storing the pointer to the child and for the

representation of each float value.
Nodes have a capacity, which denotes the maximum number of elements they can contain. This value is usually different for internal nodes (M_I) and leaf nodes (M_L) due to the different structure. Some constraints are imposed to control the minimum loading factor: each internal node, root excepted, must contain at least m_I elements; each leaf node must contain at least m_L elements. The ratio between the maximum and minimum node loading factor is a parameter that must be set at tree-creation time (*e.g.* , $m_I = \frac{M_I}{3}$).
These constraints help in balancing the element distribution among different nodes and allow a higher utilization of the structure to be achieved. The minimum and maximum capacity of the leaves are related to the dimensionality k of the corresponding KL subspace, to the size of disk blocks and the dimensionality n of the original data. In fact, in order to calculate the KL transform, at least $k + 1$ elements must be available in the leaf, thus $m_L > k$. Moreover, the constraint $M_L = 2m_L > 2k$ is necessary to allow leaf splitting. As far as internal nodes are concerned, the only constraint is $M_I = 2m_I$. Figure 4.26 describes the general structure of an MKL-tree: each node of the structure is represented by a KL subspace, denoted by S_i, which is maintained inside its parent node. Internal nodes contain the representation of their children, while leaves contain the projections of the indexed objects into the relative eigenspace. M_I and m_I are the maximum and minimum internal-node capacity, respectively. M_L and m_L are the maximum and minimum leaf capacity, respectively. The root is an internal node not having a minimum capacity. Further details about MKL-tree data structure regarding

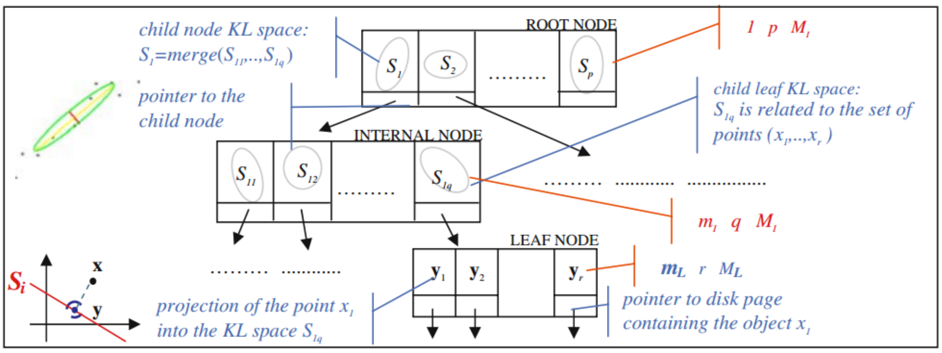


Figure 4.26: The general structure of an MKL-tree (figure from [59]).

node compression and algorithms for bulk loading and compression are reported in [59].
Now we focus our attention on a specific and important managing operation: the similarity search using the indexing structure described above. The MKL-tree

performs approximated queries; as in the search algorithms a pruning criterion is adopted in order to speed up the execution of the queries reducing the number of visited nodes. This criterion, that will be better explained in the following, causes the "loss" of some points that, in the original space, would have fulfilled the search criterion, producing an approximated result. The approximation introduced by this algorithm and its effected will be examined and discussed in Subsection 4.9.4. Besides, unlike what happens in traditional database systems, where queries are typically executed by searching for records which exactly match the searched element, in multimedia database the standard approach is to do a search on a high dimensional space where the concept of exact match has little meaning, and thus concepts of similarity are typically applied. The similarity

```
RANGESEARCH (element p, float ρ, element set R)
begin
        R = ∅
        Set N = root of the tree
        Inizialize an empty heap-queue H
        Insert (N, 0) into H       // insert the root into H with distance 0
        repeat
                Set N = Extract the minimum element from H
                Read the node N from disk and let E be the set of its elements
                if (N is not a leaf)
                        for each element S ∈ E   // S denotes a KL subspace
                                if (not PRUNE(S, p, ρ))
                                        d_S= d_FS(p,S)
                                        Insert (S, d_S) into H
                                endif
                        endfor
                else
                        for each element q ∈ E // q denotes a point
                                Calculate the projection p' of p in the subspace associated to E
                                if d(p', q) ≤ ρ
                                        R = R ∪ {q}
                                endif
                        endfor
                endif
        until H is not empty
end

bool PRUNE(element S, element p, float ρ)
begin
        if ( d_FS(p,S) > 1/2 ε(S) + ρ )
                return true
        else
                if (S is a leaf) and ( d_FB(p,S) >ρ) // d_FB(p,S) is the distance from the approximate
                bounding box
                        return true
                endif
        endif
        return false
```

Figure 4.27: Pseudo-code of the range similarity search.

search algorithm (whose pseudo-code is reported in Figure 4.27) visits the tree

starting from the root, by following the most promising path. To this aim, a heap queue of the nodes to be visited is maintained, ordered according to the distance ($d_F S$) from the searched point. At each step of the search, the first node in the queue is extracted: if it is an internal node, all its children, whose distance from the searched point is lower than a threshold (given by the pruning criterion), are inserted into the queue; if it is a leaf, its elements are evaluated with respect to query radius (or the actual K-NN distance). The search stops as soon the first element in queue does not fulfill the pruning criterion or the heap becomes empty. The pruning rule is based on probabilistic criteria: a node (represented by a subspace S) is visited only if the $d_F B(p, S)$, denoting the smallest distance of the query center p from its approximate bounding box, is less than a threshold (depending on the query radius for range searches or on the minimum distance for nearest-neighbor searches). The approximate bounding box of a KL subspace is defined as the hyper-rectangle that is centered in the origin of the subspace and whose semi-axes lengths are three times the values of the corresponding standard deviation (which coincides with the square root of the related eigenvalue). If the points associated to a node have a Gaussian distribution, this approach guarantees that each point is within the bounding-box (whose size is related to the standard deviation of the distribution) with a probability greater than 99.73%; this statement is not true for other distributions but in any case, the loss of information is very low, as confirmed by the experimental results.
The result of the search procedure is a set of objects that satisfy the search condition in a reduced subspace. Due to the dimensionality reduction performed, some of these objects could not fulfill the search condition in the original space; therefore, a further false positive elimination step is required.

4.9.4 Fine-grained recognition on MKL-based candidate set

MKL-tree achieves high performance thanks to its use of local data approximation at each node. Since each MKL-tree node makes a KL dimensionality reduction of the whole space with the aim of better representing the related points, the precision of the result is increased with respect to a global dimensionality reduction, yielding to improved search performance. Differently from the other access methods based on dimensionality reduction (LDR, GDR), the MKL-tree is a dynamic

structure, in which the reduced data spaces can be updated in order to better characterize new data objects, even drawn from a different distribution. Furthermore, proper compression technique [59] allows to reduce the number of disk block required to store the tree, thus achieving a noticeably reduction of I/O costs.
Retrieval performance and computational costs of MKL-based indexing structure are more deeply illustrated in [59]. In this section, we investigate the use of such indexing structure to retrieve most similar photos to a given sketch, thus enabling fine-grained recognition only on the retrieved subset. Differently from Section 4.8 where the candidate set is obtained by comparing each photo of the gallery to the given sketch by means of shape feature similarity, here the candidate set is retrieved through MKL-tree and range search on it. In other words, we retrieve a candidate set by executing a K-NN sketch-based query. By visiting the tree, the candidate set can be retrieved in a time that is near constant with increasing gallery size. Therefore, the index-based solution offers a shorter retrieval time respect to the standard search illustrated in Section 4.8 where the entire gallery must be parsed at each new sketch-based identification.
One of the main aspect to investigate is how to adapt MKL-tree to our purpose of photo retrieval. The cost for building the MKL-tree has to be paid only once when the tree is created for the first time. Then, it can be updated just in case when new photos are included in the gallery. The main problem to deal with at this phase is due to the heterogeneous composition of shape features. As illustrated in section Section 4.4, we defined different shape features with various meaning (histogram, Fourier transform, space distance, etc.) and they are computed separately for each image. In order to make possible the use of a self-contained indexing structure, we need to work with unique feature vectors associated to photos and sketches, thus enabling the use of range search and K-NN queries in accordance with the described in Section 4.9.3. Therefore, to build our MKL-tree we adopt the following preliminary steps:

i) shape descriptors are computed separately for each photo of the gallery;

ii) shape descriptors are normalized so that each descriptor component falls in the range $[0, ..., 1]$;

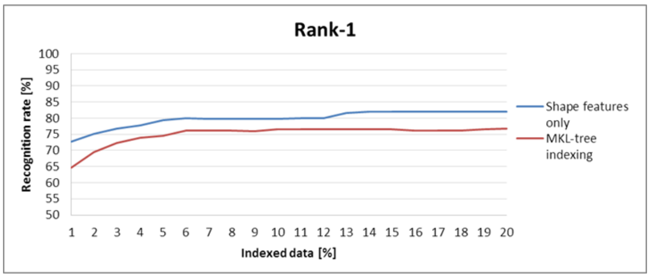
iii) normalized descriptors are weighted accordingly to weights assigned to different shape features (see Section 4.5);

iv) normalized and weighted shape features are concatenated in order to create a unique shape descriptor vector.

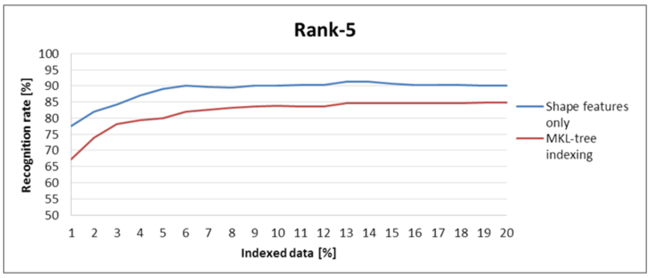
After these preliminary steps, we have at our disposal a feature vector set with 8221 vectors, each one corresponding to a different photo in the gallery. It is worth noting that each vector is composed by 7829 components, thus justifying the use of an indexing structure able to deal with high-dimensional data. Once we create the gallery-related feature set, we carry out the final tree building: we found that a 3 level tree with a KL reduced dimension equal to 80 works well for our purpose. In Figure 4.28 we show fine-grained recognition accuracy on viewed-sketches by performing SURF on two different candidate sets: the one obtained by linearly scanning the gallery thus selecting the most similar photos according to shape features similarity (as explained in Section 4.8) and the one obtained through a sketch-based query performed on a suitable MKL-based indexing tree. As before, we report five different accuracy values (Rank-1, Rank-5, Rank-10, Rank-15 and Rank-20) by varying the amount of retrieved data as percentage of the total available photos. As illustrated in Section 4.8, the best portion of retrieved data within which expect the photo is around 10%: therefore, we consider here a range from 1% to 20% thus showing fine-grained recognition perfomance for such reduced but significant subset.

The charts show that MKL-based indexing leads to a drop of recognition performance. Actually, this is not totally surprising: employing shape features to scan the gallery we perform an exact match by comparing each photo to the given sketch. On the contrary, with MKL-based indexing we adopt dimensionality reduction which introduces an approximation. Moreover, the indexing tree is visited by adopting a range search thus making the matching not exact.

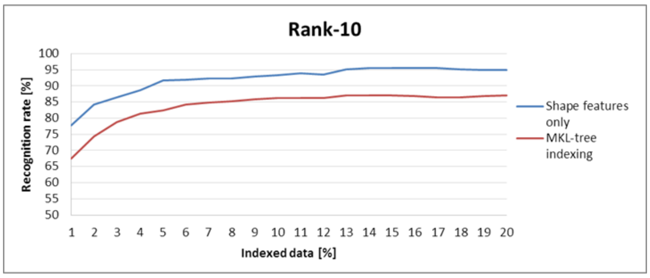
However, we can consider the results offered by our MKL-tree encouraging and worthy of further investigation: the performance drop is not unbearable and MKL indexing leads to fine-grained accuracy that is comparable with the use of shape features only (especially when the candidate set is composed by the 10% of most similar photos to the given sketch). Moreover, tests prove that the average time required to retrieve the candidate set from a skech-based query is ~ 1 *sec.* (using the same platform described in Section 4.7) and it is near constant with increasing gallery size thanks to the property of MKL-tree [59]. On the contrary, using shape features to perform linear scan the time is more than 8 time greater (supposing that photo descriptors have been precomputed and stored thus paying only matching costs) and it is intended to grow proportionally with the gallery size, thus highlighting computational advantages provided by our indexing tree.



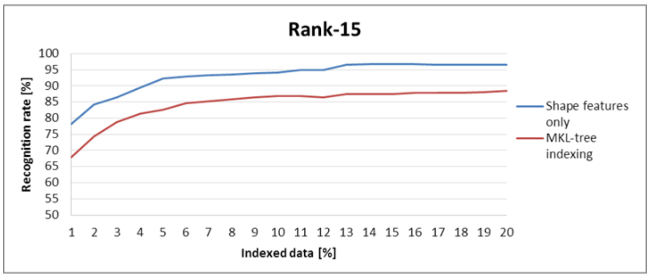
(a)



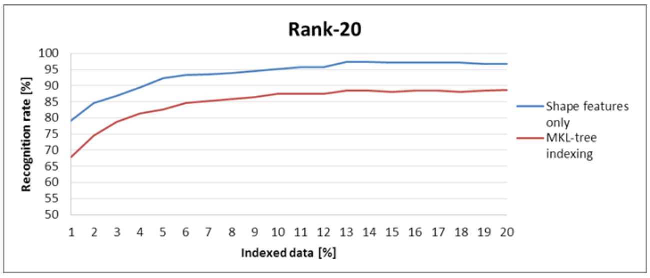
(b)



(c)



(d)



(e)

Figure 4.28: Fine-grained recognition accuracy by varying the percentage of indexed photos considered as candidate set.

Chapter 5

Conclusion

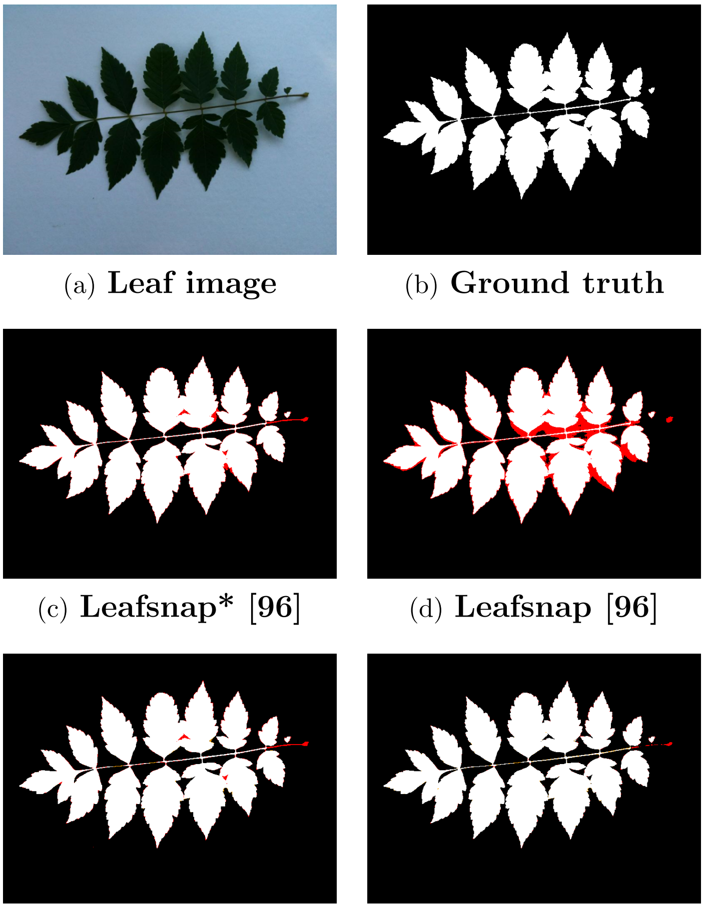
Computers, portable devices and embedded processing units are becoming more and more part of our everyday life and their functionalities are employed by a wide range of population. To make such devices truly at the service of people, it is necessary to develop and embed intelligent (smart) components inside them. What provides intelligence is usually a specific software, that process and understand events in order to take decisions in real time. Generally speaking, the problem has shifted from producing hardware to producing *smart* hardware. In particular, smart hardware and technology placed in the surrounding environment are required to react and interact with objects and people thus adapting to users needs and preferences: this ability is usually denoted and with Ambient Intelligence (AmI). Various core disciplines participate in the creation of the concepts and applications of AmI and researchers from several areas must collaborate: sensor networks, sociology, artificial intelligence and communication networks are some examples.

One of the main enabling science for AmI is Computer Vision. Computer Vision provide the ability of acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information in the forms of decisions. All of this tasks are crucial for many different AmI applications, mainly because an environment can hardly be responsive and proactive without the ability of seeing what is happening around. Computer Vision supplies technologies and solutions for a various applications such as ambient assisted living, human computer interaction, surveillance, recognition and so on.

Among others, main research challenges of Computer Vision are distributed vision algorithms, vision-based people tracking, biometric recognition, human behavior understanding, object detection, recognition, segmentation and tracking. In this thesis we analyzed some of these areas, proposing novel techniques and approaches and applying them to specific Ami-related contexts: keypoints reduction/selection to support Augmented Reality, segmentation of natural images to support plant recognition and face recognition from sketches for people identification. Despite the diversity of such techniques and their possible applications, one important common trait clearly appears: even though all the general discussed topics (image segmentation, face recognition and object detection/tracking) have been hot topics in Computer Vision for many years, there is still work to do because AmI is offering new concrete challenges in real-world contexts. For example, standard techniques for segmentation of natural images tend to fail partially or totally because of images taken in unsupervised or poorly controlled conditions, thus thus requiring novel and improved approaches. Similar situation has been observed for heterogeneous face recognition, where state-of-the-art recognition approaches are not suitable to directly compare different types of image (a sketch and a real photo) and new methods are required. Finally, despite a long research effort in the field of object description and matching, our saliency-based ranking and selection of keypoints demonstrate its effectiveness in terms of both matching accuracy and processing speed making this approach feasible for real time application such as Augmented Reality, which is an important emerging field with strong connection with AmI. Therefore, it is evident that AmI is proposing new research problems to Computer Vision and related solutions need to be investigated with important attention to concrete and real applications.

Appendix A

Leaf segmentation results

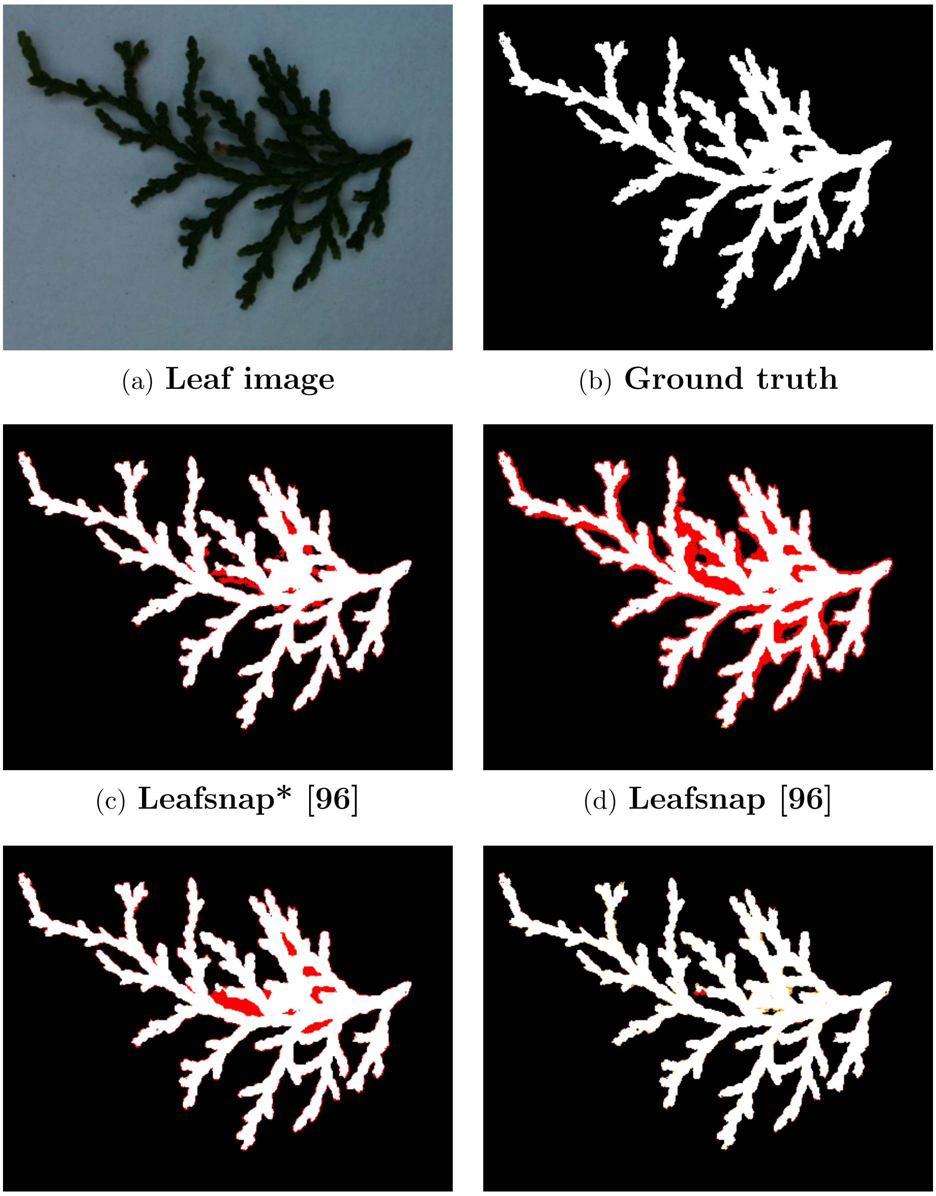


(a) **Leaf image** (b) **Ground truth**

(c) **Leafsnap* [96]** (d) **Leafsnap [96]**

(e) **GrabCut [135]** (f) **Ours-boundary**

Figure A.1: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

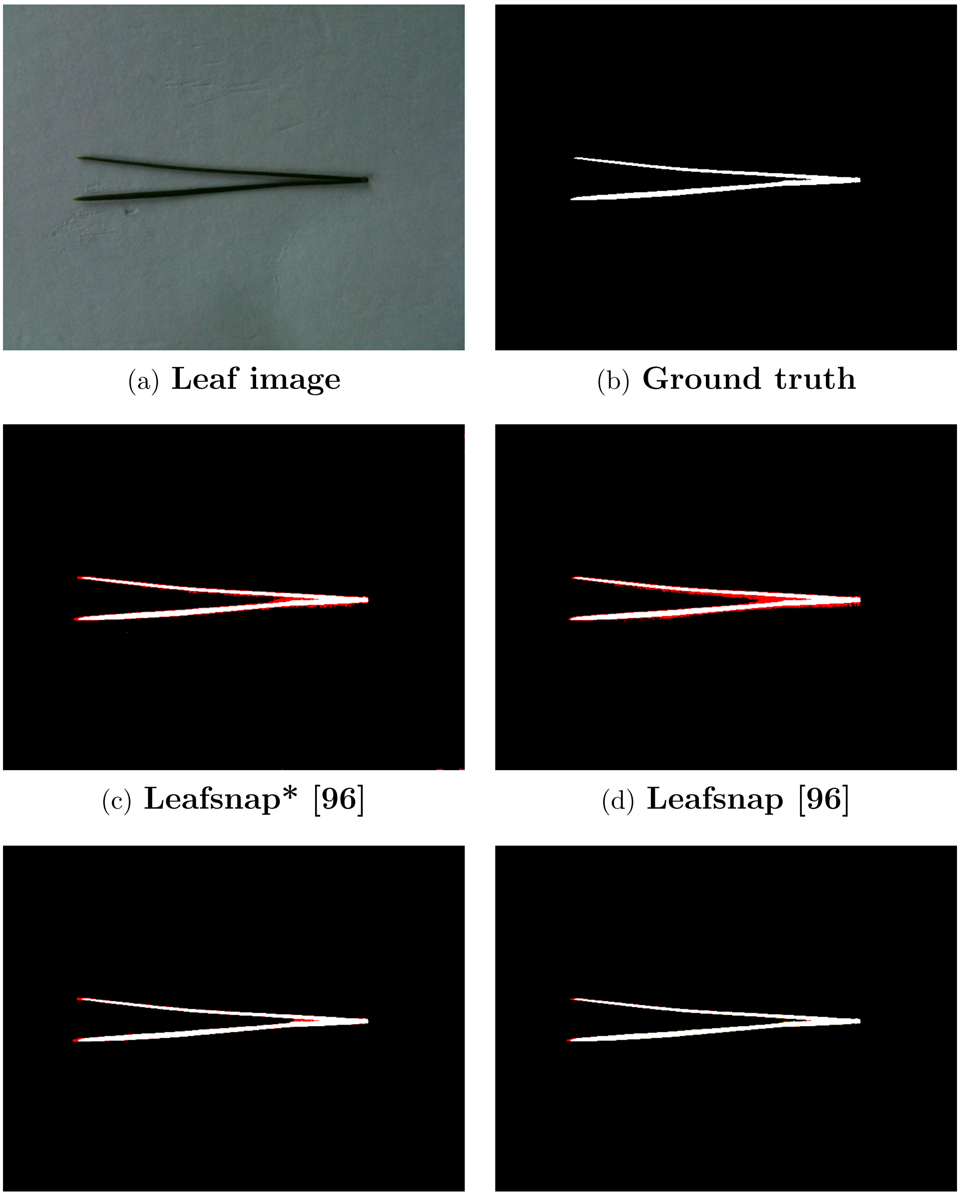


(a) **Leaf image** (b) **Ground truth**

(c) **Leafsnap* [96]** (d) **Leafsnap [96]**

(e) **GrabCut [135]** (f) **Ours-boundary**

Figure A.2: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

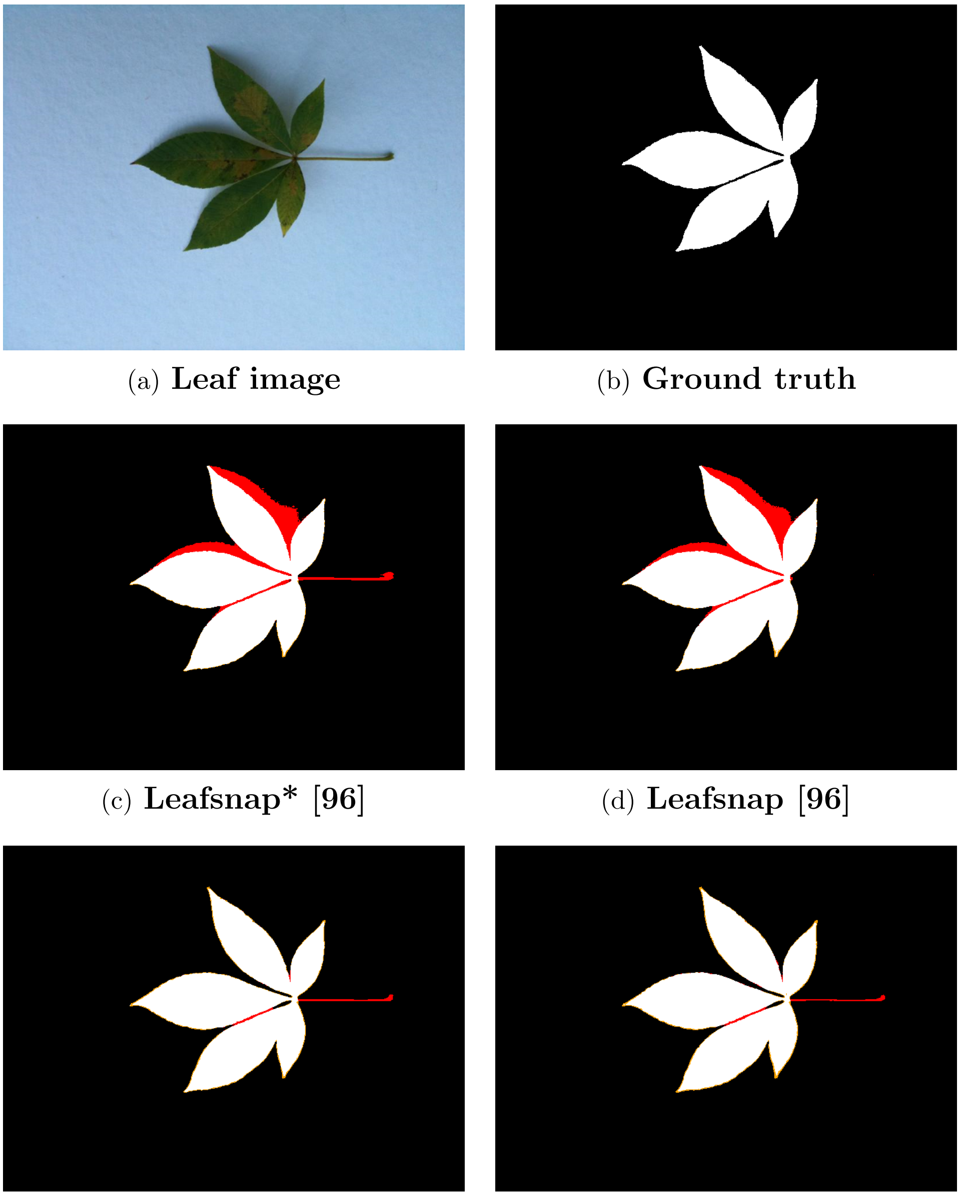


(a) **Leaf image** (b) **Ground truth**

(c) **Leafsnap* [96]** (d) **Leafsnap [96]**

(e) **GrabCut [135]** (f) **Ours-boundary**

Figure A.3: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).



(a) **Leaf image** (b) **Ground truth**

(c) **Leafsnap* [96]** (d) **Leafsnap [96]**

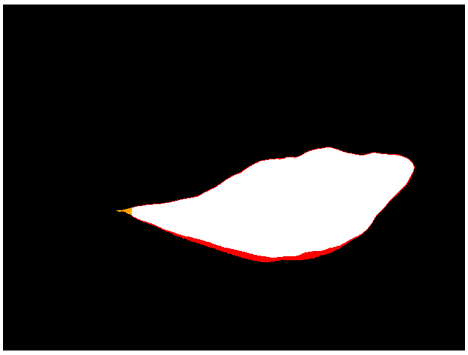
(e) **GrabCut [135]** (f) **Ours-boundary**

Figure A.4: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

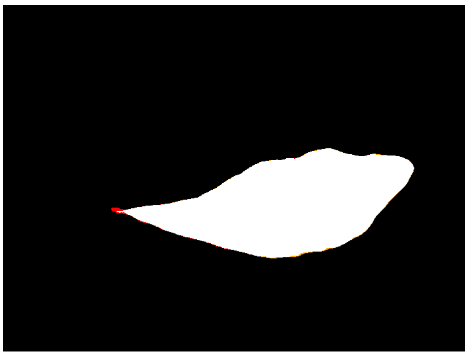
(a) **Leaf image**

(b) **Ground truth**

(c) **Leafsnap* [96]**



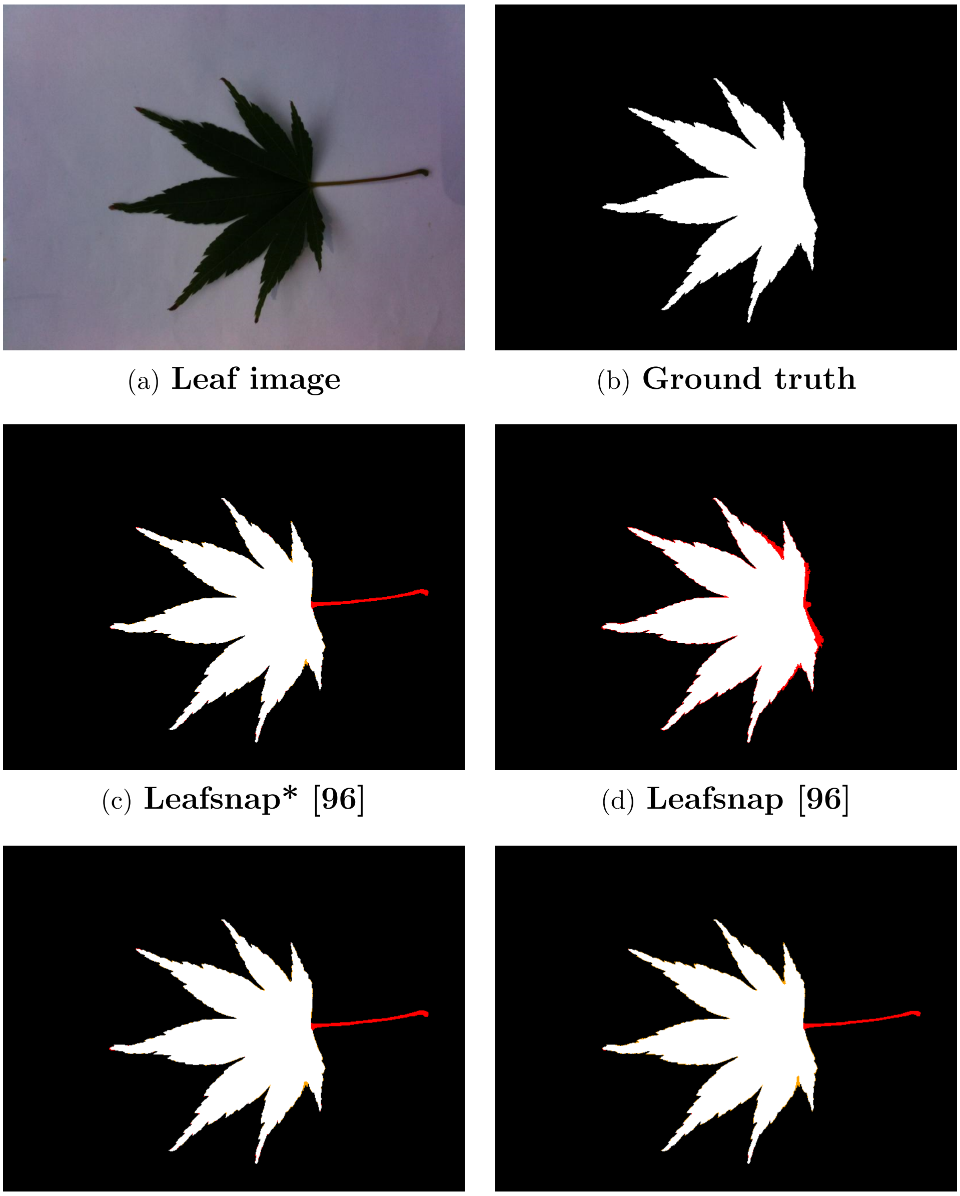
(d) **Leafsnap [96]**



(e) **GrabCut [135]**

(f) **Ours-boundary**

Figure A.5: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).



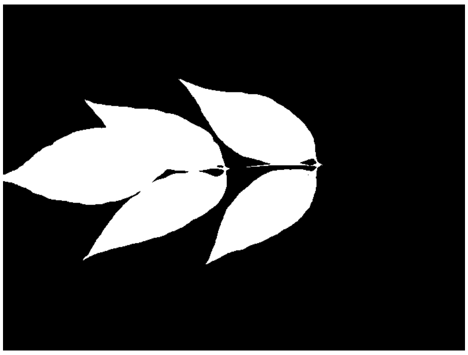
(a) **Leaf image** (b) **Ground truth**

(c) **Leafsnap* [96]** (d) **Leafsnap [96]**

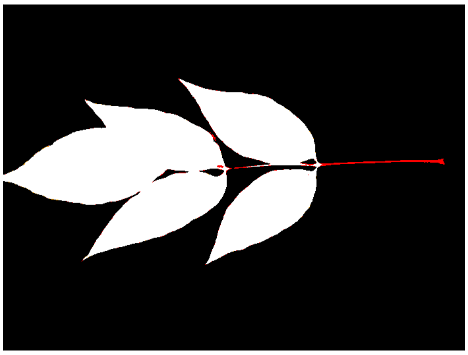
(e) **GrabCut [135]** (f) **Ours-boundary**

Figure A.6: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

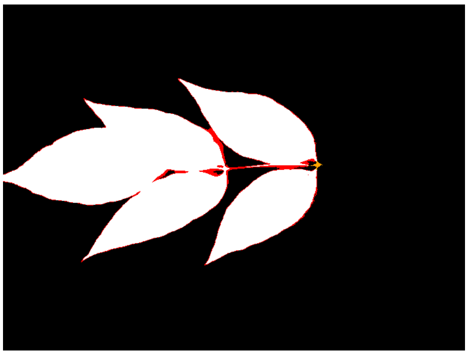
(a) **Leaf image**



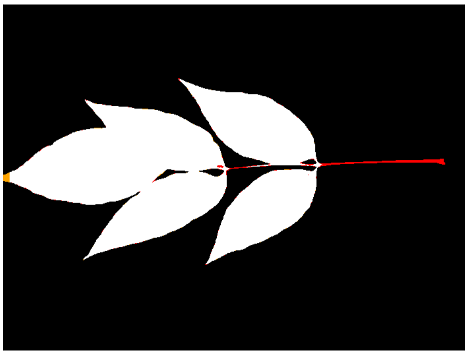
(b) **Ground truth**



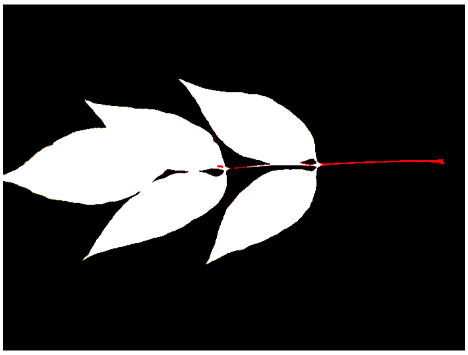
(c) **Leafsnap* [96]**



(d) **Leafsnap [96]**



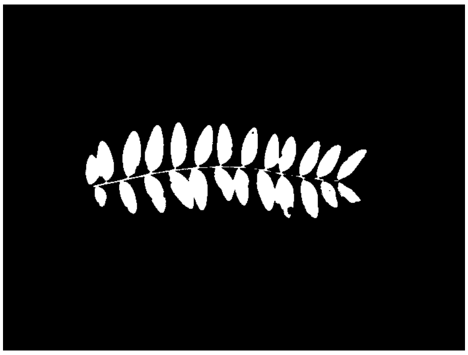
(e) **GrabCut [135]**



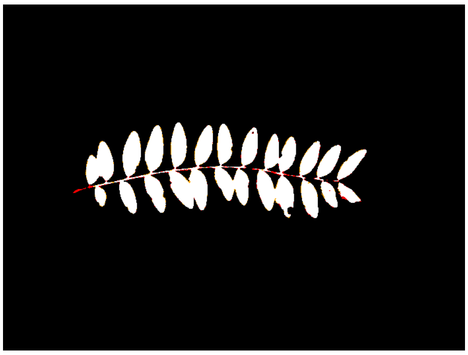
(f) **Ours-boundary**

Figure A.7: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

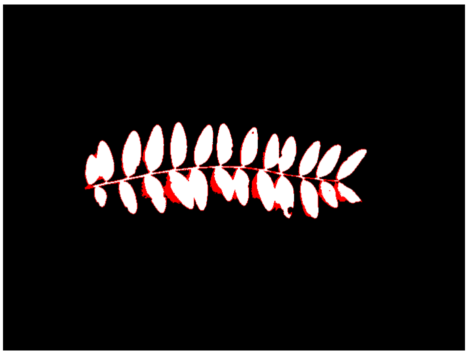
(a) **Leaf image**



(b) **Ground truth**



(c) **Leafsnap* [96]**



(d) **Leafsnap [96]**

(e) **GrabCut [135]**

(f) **Ours-boundary**

Figure A.8: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

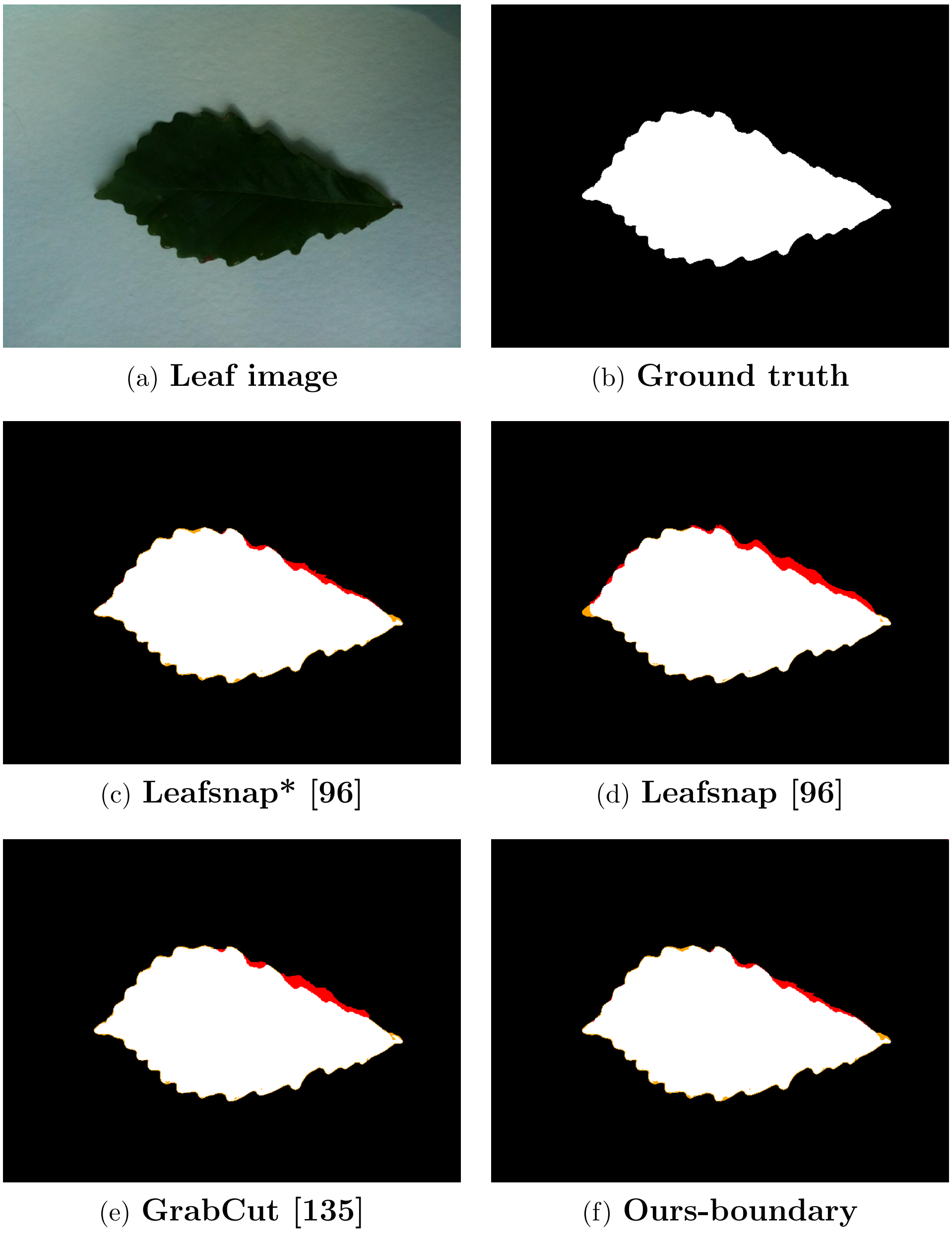
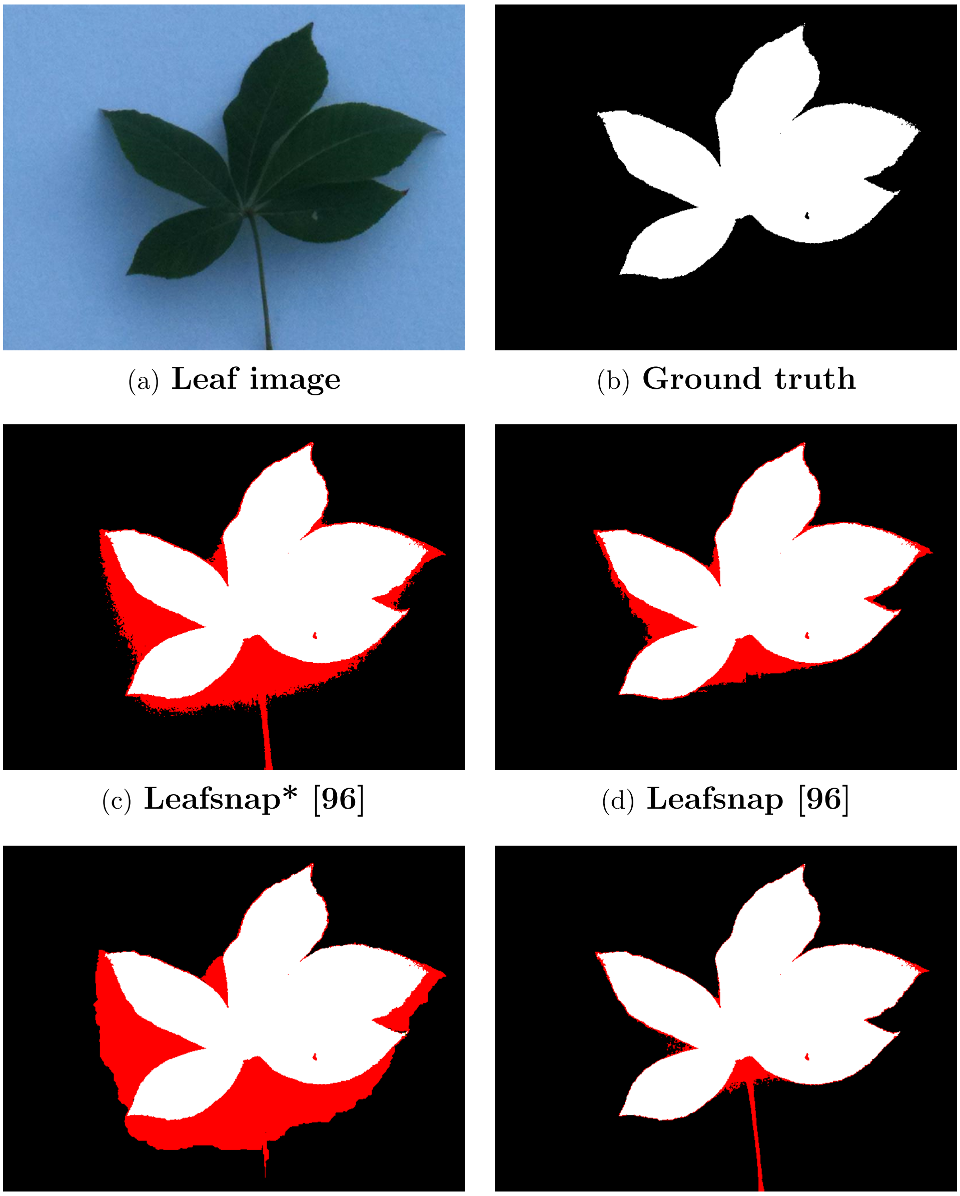


Figure A.9: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

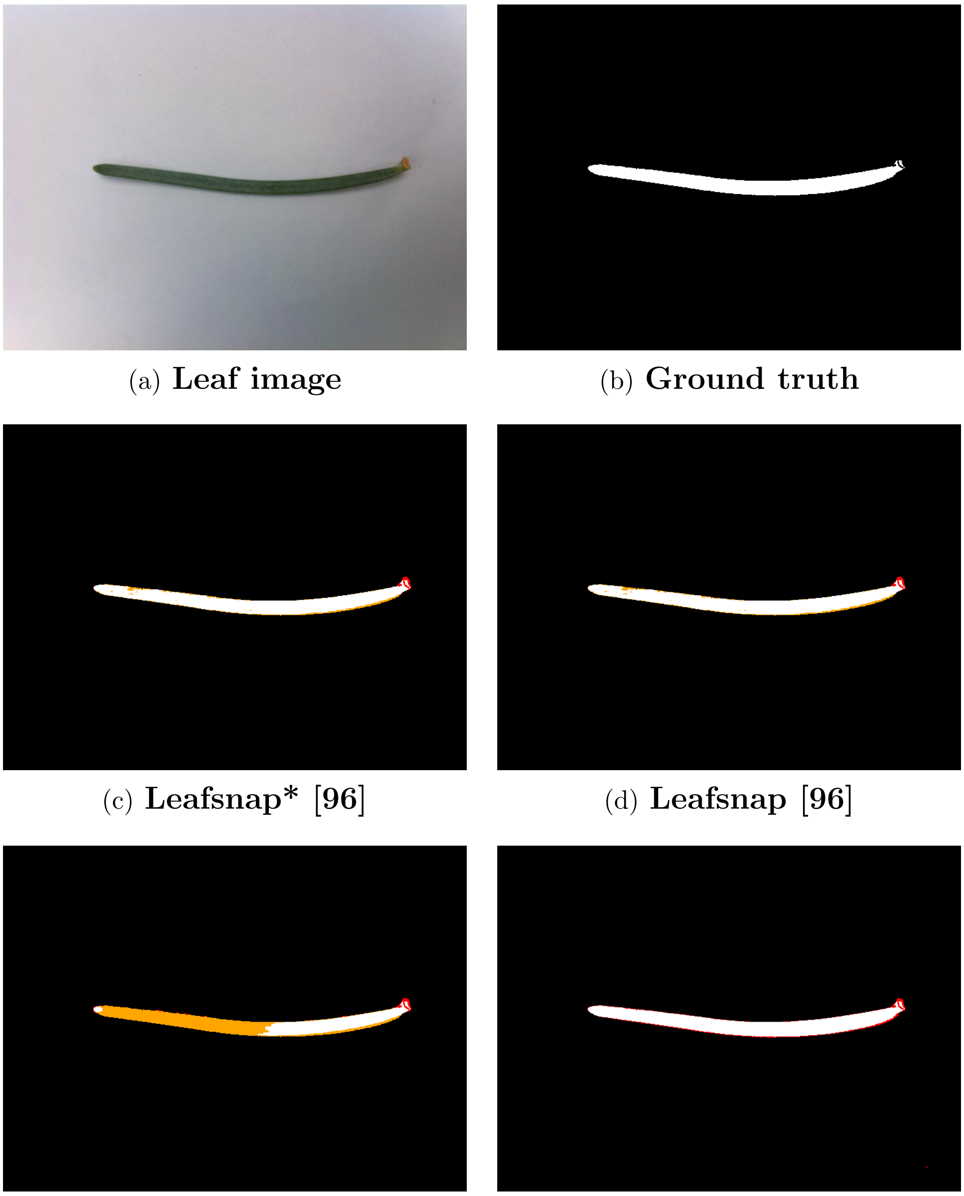


(a) **Leaf image** (b) **Ground truth**

(c) **Leafsnap* [96]** (d) **Leafsnap [96]**

(e) **GrabCut [135]** (f) **Ours-boundary**

Figure A.10: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

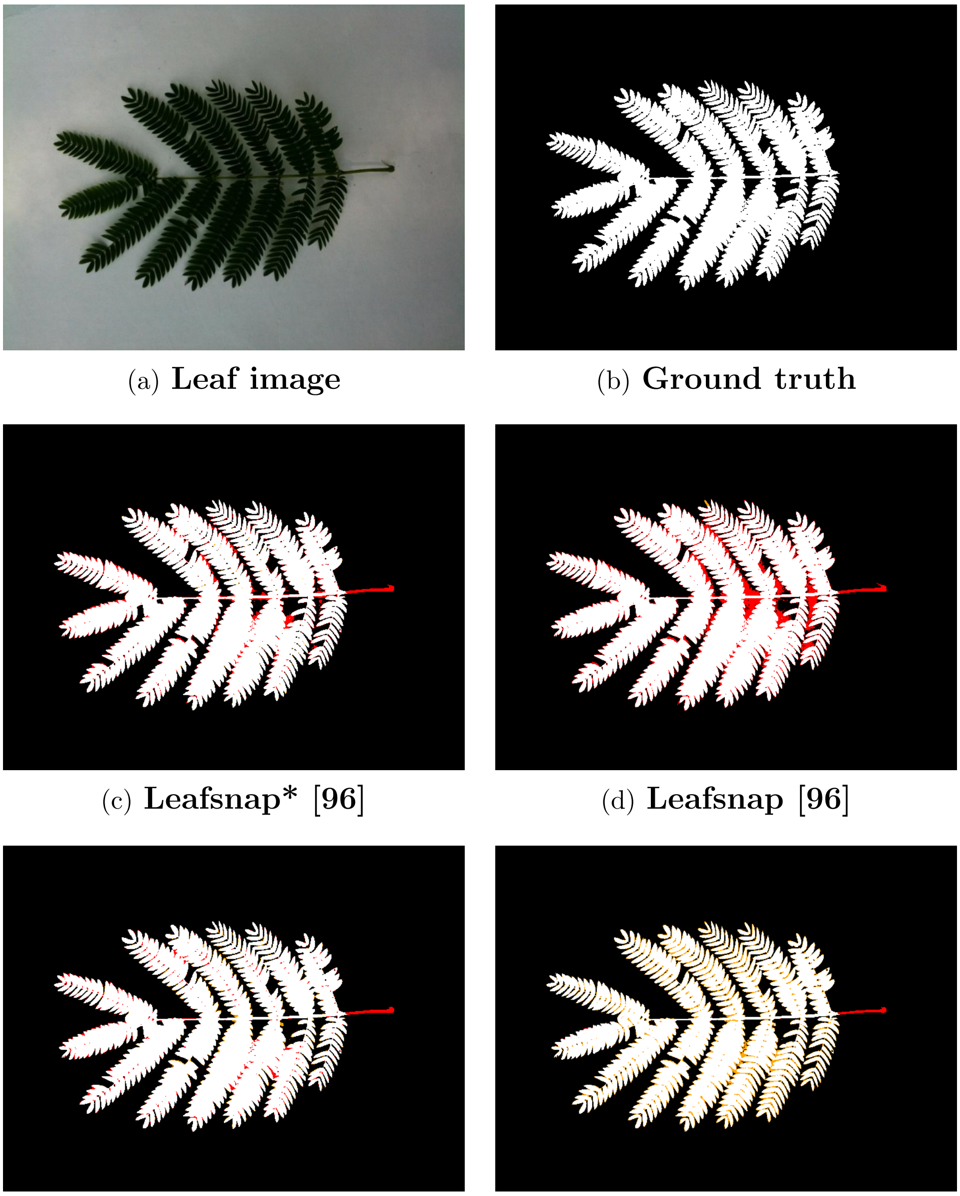


(a) **Leaf image** (b) **Ground truth**

(c) **Leafsnap* [96]** (d) **Leafsnap [96]**

(e) **GrabCut [135]** (f) **Ours-boundary**

Figure A.11: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).



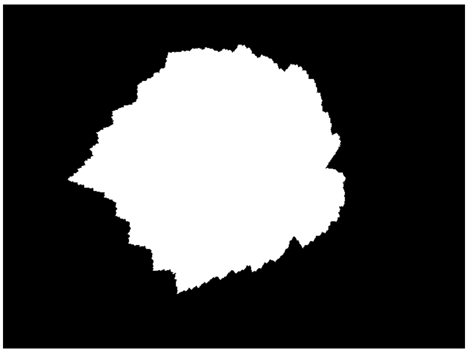
(a) **Leaf image** (b) **Ground truth**

(c) **Leafsnap* [96]** (d) **Leafsnap [96]**

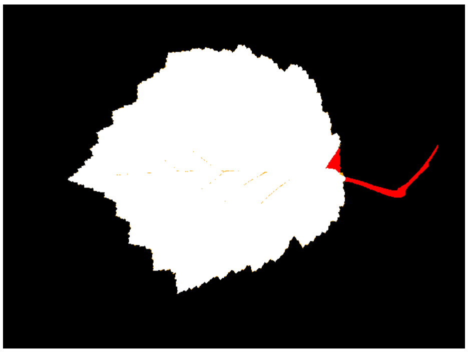
(e) **GrabCut [135]** (f) **Ours-boundary**

Figure A.12: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

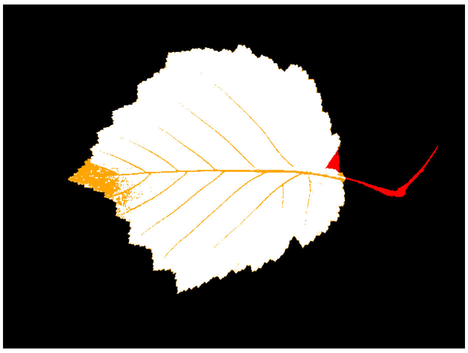
(a) **Leaf image**



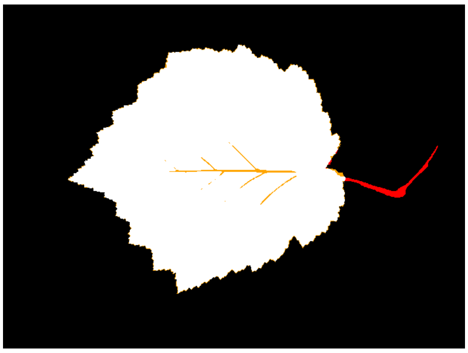
(b) **Ground truth**



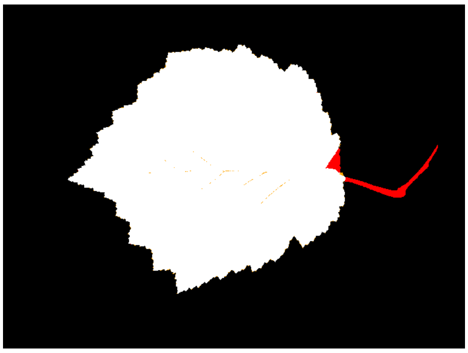
(c) **Leafsnap* [96]**



(d) **Leafsnap [96]**



(e) **GrabCut [135]**



(f) **Ours-boundary**

Figure A.13: Leaves segmentation under loosely controlled conditions with different methods. False positives in red, false negatives in orange (best viewed in color).

Appendix B

List of scientific publications

The author contributed to the following scientific publications during the timeframe of this PhD:

- S. Buoncompagni, A. Franco and D.Maio, "Shape features for candidate photo selection in sketch recognition", 22nd International Conference on Pattern Recognition (ICPR14), Stockholm (Sweden), August 2014, pp. 1728-1733.
  Abstract. *Sketch recognition for forensic applications is a very challenging task and several solutions have recently been proposed. Considering that real mug shot databases can be very large, one important aspect to consider in this scenario is also the efficiency of the search procedure. This work proposes the use of shape features for a preliminary selection of the candidate photos to be successively analyzed by more complex state-of-the-art techniques. The proposed features can be computed and matched in a very short time, and at the same time are able to significantly reduce the search space, thus allowing to speed up the recognition process.*

- S. Buoncompagni, D. Maio, D. Maltoni, S. Papi, "Saliency-based keypoint selection for fast object detection and matching", Pattern Recognition Letters, Vol. 62, September 2015, pp. 32-40.
  Abstract. *In this paper we present a new approach to rank and select keypoints based on their saliency for object detection and matching under moderate viewpoint and lighting changes. Saliency is defined in terms of detectability, repeatability and distinctiveness by considering both the keypoint*

strength (as returned by the detector algorithm) and the associated local descriptor discriminating power. Our experiments prove that selecting a small amount of available keypoints (e.g., 10%) not only boosts efficiency but can also lead to better detection/matching accuracy thus making the proposed method attractive for real-time applications (e.g., augmented reality).

- S. Buoncompagni, D. Maio, D. Maltoni, S. Papi, "Saliency-based keypoint reduction for augmented-reality applications in smart cities", Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9281, pp. 209-217, 2015.
  Abstract. *In this paper we show that Saliency-based keypoint selection makes natural landmark detection and object recognition quite effective and efficient, thus enabling augmented reality techniques in a plethora of applications in smart city contexts. As a case study we address a tour of a museum where a modern smart device like a tablet or smartphone can be used to recognize paintings, retrieve their pose and graphically overlay useful information.*

- S. Buoncompagni, D. Maio, V. Lepetit, "Leaf segmentation under loosely controlled conditions", Proceedings of the British Machine Vision Conference (BMVC15), Swansea (UK), September, 2015.
  Abstract. *We propose a robust and accurate method for segmenting specular objects acquired under loosely controlled conditions. We focus here on leaves because leaf segmentation plays a crucial role for plant identification, and accurately capturing the local boundary structures is critical for the success of the recognition. Popular techniques are based on Expectation-Maximization and estimate the color distributions of the background and foreground pixels of the input image. As we show, such approaches suffer in presence of shadows and reflections thus leading to inaccurate detected shapes. Classification-based methods are more robust because they can exploit prior information, however they do not adapt to the specific capturing conditions for the input image. Methods with regularization terms are prone to smooth the segments boundaries, which is undesirable. In this paper, we show we can get the best of the EM-based and classification-based methods by first segmenting the pixels around the leaf boundary, and use them to initialize the color distributions of an EM optimization. We show that this simple approach results in a robust and accurate method.*

Bibliography

[1] Gaurav Agarwal, Peter Belhumeur, Steven Feiner, David Jacobs, W John Kress, Ravi Ramamoorthi, Norman A Bourg, Nandan Dixit, Haibin Ling, Dhruv Mahajan, et al. First Steps Toward an Electronic Field Guide for Plants. *Taxon*, pages 597–610, 2006.

[2] Shivani Agarwal and Dan Roth. Learning a sparse representation for object detection. In *Computer Vision - ECCV 2002*, pages 113–127. Springer, 2002.

[3] Hamid Aghajan, Juan Carlos Augusto, Chen Wu, Paul McCullagh, and Julie-Ann Walkden. Distributed vision-based accident management for assisted living. In *Pervasive Computing for Quality of Life Enhancement*, pages 196–205. Springer, 2007.

[4] Hamid Aghajan and Andrea Cavallaro. *Multi-camera networks: principles and applications*. Academic press, 2009.

[5] Yasmin Aghajan, Joyca Lacroix, Jingyu Cui, Aart van Halteren, and Hamid Aghajan. Home exercise in a social context: Real-time experience sharing using avatars. In *Intelligent Technologies for Interactive Entertainment*, pages 19–31. Springer, 2009.

[6] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. IEEE, 2012.

[7] Yali Amit and Donald Geman. A computational model for visual selection. *Neural computation*, 11(7):1691–1715, 1999.

[8] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011.

[9] Nafiz Arica and Fatos T Yarman Vural. Bas: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24(9):1627–1639, 2003.

[10] Akhil Arora, Ankit Gupta, Nitesh Bagmar, Shashwat Mishra, and Arnab Bhattacharya. A plant identification system using shape and morphological features on segmented leaflets: Team iitk, clef 2012. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[11] Juan Carlos Augusto. Ambient intelligence: the confluence of ubiquitous/pervasive computing and artificial intelligence. In *Intelligent Computing Everywhere*, pages 213–234. Springer, 2007.

[12] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34–47, 2001.

[13] Lauren Barghout and Lawrence Lee. Perceptual information processing system, July 11 2003. US Patent App. 10/618,543.

[14] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer vision - ECCV 2006*, pages 404–417. Springer, 2006.

[15] Peter N Belhumeur, Daozheng Chen, Steven Feiner, David W Jacobs, W John Kress, Haibin Ling, Ida Lopez, Ravi Ramamoorthi, Sameer Sheorey, Sean White, et al. Searching the world's herbaria: A system for visual identification of plant species. In *Computer Vision–ECCV 2008*, pages 116–129. Springer, 2008.

[16] Richard Bellman, Richard Ernest Bellman, Richard Ernest Bellman, and Richard Ernest Bellman. *Adaptive control processes: a guided tour*, volume 4. Princeton university press Princeton, 1961.

[17] Philip J Benson and David I Perrett. Perception and recognition of photographic quality facial caricatures: Implications for the recognition of natural images. *European Journal of Cognitive Psychology*, 3(1):105–135, 1991.

[18] Bir Bhanu and Xuejun Tan. Fingerprint indexing based on novel features of minutiae triplets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):616–622, 2003.

[19] Himanshu S Bhatt, Samarth Bharadwaj, Richa Singh, and Mayank Vatsa. On matching sketches with digital face images. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–7. IEEE, 2010.

[20] Himanshu S Bhatt, Samarth Bharadwaj, Richa Singh, and Mayank Vatsa. Memetically optimized mcwld for matching sketches with digital face images. *Information Forensics and Security, IEEE Transactions on*, 7(5):1522–1535, 2012.

[21] Jeff A Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[22] Alberto Del Bimbo and Pietro Pala. Visual image retrieval by elastic matching of user sketches. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(2):121–132, 1997.

[23] Vicki Bruce, Elias Hanna, Neal Dench, Pat Healey, and Mike Burton. The importance of "mass" in line drawings of faces. *Applied Cognitive Psychology*, 6(7):619–628, 1992.

[24] Vicki Bruce and Glyn W Humphreys. Recognizing objects and faces. *Visual cognition*, 1(2-3):141–180, 1994.

[25] Simone Buoncompagni, Annalisa Franco, and Dario Maio. Shape features for candidate photo selection in sketch recognition. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1728–1733. IEEE, 2014.

[26] Simone Buoncompagni, Dario Maio, Davide Maltoni, and Serena Papi. Saliency-based keypoint reduction for augmented-reality applications in smart cities. In *New Trends in Image Analysis and Processing–ICIAP 2015 Workshops*, pages 209–217. Springer, 2015.

[27] Simone Buoncompagni, Dario Maio, Davide Maltoni, and Serena Papi. Saliency-based keypoint selection for fast object detection and matching. *Pattern Recognition Letters*, 2015.

[28] Michael Calonder, Vincent Lepetit, Pascal Fua, Kurt Konolige, James Bowman, and Patrick Mihelich. Compact signatures for high-speed interest point

description and matching. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 357–364. IEEE, 2009.

[29] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief: Computing a local binary descriptor very fast. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1281–1298, 2012.

[30] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. *Computer Vision - ECCV 2010*, pages 778–792, 2010.

[31] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[32] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Fingerprint indexing based on minutia cylinder-code. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):1051–1057, 2011.

[33] Raffaele Cappelli and Davide Maltoni. Multispace kl for pattern representation and classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(9):977–996, 2001.

[34] Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. Smart cities in europe. *Journal of urban technology*, 18(2):65–82, 2011.

[35] Andrea Caridi, Mauro Coccoli, and Valentina Volpi. Wolfsoniana smart museum. a pilot plant installation of the palm-cities project. In *UMAP Workshops*, 2013.

[36] Gustavo Carneiro and Allan D Jepson. The distinctiveness, detectability, and robustness of local image features. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 296–301. IEEE, 2005.

[37] Gustavo Carneiro and Allan D Jepson. The quantitative characterization of the distinctiveness and robustness of local image descriptors. *Image and Vision Computing*, 27(8):1143–1156, 2009.

[38] Dalcimar Casanova, Joao Batista Florindo, Wesley Nunes Gonçalves, and Odemir Martinez Bruno. Ifsc/usp at imageclef 2012: Plant identification task. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[39] Guillaume Cerutti, Violaine Antoine, Laure Tougne, Julien Mille, Lionel Valet, Didier Coquin, and Antoine Vacavant. Reves participation-tree species classification using random forests and botanical features. In *Conference and Labs of the Evaluation Forum (CLEF)*, page 1, 2012.

[40] Guillaume Cerutti, Laure Tougne, Julien Mille, Antoine Vacavant, and Didier Coquin. Understanding leaves in natural images–a model-based approach for tree species identification. *Computer Vision and Image Understanding*, 117(10):1482–1501, 2013.

[41] Kaushik Chakrabarti and Sharad Mehrotra. Local dimensionality reduction: A new approach to indexing high dimensional spaces. In *VLDB*, pages 89–100. Citeseer, 2000.

[42] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[43] H Thomson Comer and Bruce A Draper. Interest point stability prediction. In *Computer Vision Systems*, pages 315–324. Springer, 2009.

[44] Diane Cook and Sajal Das. *Smart environments: technology, protocols and applications*. John Wiley & Sons, 2004.

[45] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1124–1131. IEEE, 2005.

[46] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[47] Kevin Curran, Denis McFadden, and Ryan Devlin. The role of augmented reality within ambient intelligence. 2011.

[48] Areti Damala, Pierre Cubaud, Anne Bationo, Pascal Houlier, and Isabelle Marchal. Bridging the gap between the digital and the physical: design and evaluation of a mobile augmented reality guide for the museum visit. In

Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts, pages 120–127. ACM, 2008.

[49] Areti Damala, Isabelle Marchal, and Pascal Houlier. Merging augmented reality based features in mobile multimedia museum guides. In *Anticipating the Future of the Cultural Past, CIPA Conference 2007, 1-6 October 2007,*, pages 259–264, 2007.

[50] Graham Davies, Hadyn D Ellis, and John Shepherd. Face recognition accuracy as a function of mode of representation. *Journal of Applied Psychology*, 63(2):180, 1978.

[51] Somnath Dey and Debasis Samanta. Iris data indexing method using gabor energy features. *Information Forensics and Security, IEEE Transactions on*, 7(4):1192–1203, 2012.

[52] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. A descriptor for large scale image retrieval based on sketched feature lines. In *SBM*, pages 29–36, 2009.

[53] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *Visualization and Computer Graphics, IEEE Transactions on*, 17(11):1624–1636, 2011.

[54] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[55] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE, 2003.

[56] Matteo Ferrara, Annalisa Franco, and Dario Maio. On the use of the kinect sensor for human identification in smart environments. *Journal of Ambient Intelligence and Smart Environments*, 6(4):435–446, 2014.

[57] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[58] Annalisa Franco, Alessandra Lumini, and Dario Maio. Eigenspace merging for model updating. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 156–159. IEEE, 2002.

[59] Annalisa Franco, Alessandra Lumini, and Dario Maio. Mkl-tree: an index structure for high-dimensional vector spaces. *Multimedia Systems*, 12(6):533–550, 2007.

[60] Annalisa Franco, Dario Maio, and Davide Maltoni. Face recognition in ambient intelligence applications. *Handbook of Ambient Assisted Living*, 2012.

[61] Reinosuke FUKUNAGA. *Statistical pattern recognition.* Academic Press., 1990.

[62] Borko Furht. *Handbook of augmented reality.* Springer Science & Business Media, 2011.

[63] Xinbo Gao, Juanjuan Zhong, Jie Li, and Chunna Tian. Face sketch synthesis algorithm based on e-hmm and selective ensemble. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(4):487–496, 2008.

[64] Jan-Mark Geusebroek, Rein Van den Boomgaard, Arnold WM Smeulders, and Hugo Geerts. Color invariance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(12):1338–1350, 2001.

[65] Lois Gibson. *Forensic art essentials: a manual for law enforcement artists.* Academic Press, 2010.

[66] Hervé Goëau, Pierre Bonnet, Alexis Joly, Vera Bakic, Daniel Barthélémy, Nozha Boujemaa, and Jean-François Molino. The ImageCLEF 2013 Plant Identification Task. In *CLEF*, 2013.

[67] Hervé Goëau, Pierre Bonnet, Alexis Joly, I. Yahiaoui, Daniel Barthélémy, Nozha Boujemaa, and Jean-François Molino. The ImageCLEF 2012 Plant Identification Task. In *CLEF*, 2012.

[68] Manuel Grand-Brochier, Antoine Vacavant, Guillaume Cerutti, Camille Kurtz, Jonathan Weber, and Laure Tougne. Tree leaves extraction in natural images: Comparative study of preprocessing tools and segmentation methods. *Image Processing, IEEE Transactions on*, 24(5):1549–1560, 2015.

[69] Kristen Grauman and Bastian Leibe. *Visual object recognition*. Morgan & Claypool Publishers, 2010.

[70] Patrick J Grother, Gerald T Candela, and James L Blue. Fast implementations of nearest neighbor classifiers. *Pattern Recognition*, 30(3):459–465, 1997.

[71] Aglika Gyaourova and Arun Ross. Index codes for multibiometric pattern retrieval. *Information Forensics and Security, IEEE Transactions on*, 7(2):518–529, 2012.

[72] Hu Han, Brendan F Klare, Kathryn Bonnen, and Anubhav K Jain. Matching composite sketches to face photos: A component-based approach. *Information Forensics and Security, IEEE Transactions on*, 8(1):191–204, 2013.

[73] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[74] Wilfried Hartmann, Michal Havlena, and Kaspar Schindler. Predicting matchability. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 9–16. IEEE, 2014.

[75] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning, 2009.

[76] Junfeng He, Regunathan Radhakrishnan, Shih-Fu Chang, and Claus Bauer. Compact hashing with joint optimization of search accuracy and time. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 753–760. IEEE, 2011.

[77] Kaiming He and Jian Sun. Computing nearest-neighbor fields via propagation-assisted kd-trees. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 111–118. IEEE, 2012.

[78] Takashi Hisamori and Gosuke Ohashi. Query-by-sketch interactive image retrieval using rough sets. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 1223–1229. IEEE, 2007.

[79] Gang Hua and Amir Akbarzadeh. A robust elastic and partial matching metric for face recognition. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2082–2089. IEEE, 2009.

[80] Alejandro Jaimes and Nicu Sebe. Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1):116–134, 2007.

[81] Anil K Jain, Brendan Klare, and Unsang Park. Face matching and retrieval in forensics applications. *iEEE MultiMedia*, (1):20–28, 2012.

[82] Anil K Jain and Stan Z Li. *Handbook of face recognition*, volume 1. Springer, 2005.

[83] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[84] Alexis Joly, Hervé Goëau, Pierre Bonnet, Vera Bakić, Julien Barbe, Souheil Selmi, Itheri Yahiaoui, Jennifer Carré, Elise Mouysset, Jean-François Molino, et al. Interactive plant identification based on social image data. *Ecological Informatics*, 23:22–34, 2014.

[85] Alexis Joly, Hervé Goëau, Hervé Glotin, Concetto Spampinato, Pierre Bonnet, Willem-Pier Vellinga, Robert Planque, Andreas Rauber, Robert Fisher, and Henning Müller. Lifeclef 2014: multimedia life species identification challenges. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, pages 229–249. Springer, 2014.

[86] Mehran Kafai, Kave Eshghi, and Bir Bhanu. Discrete cosine transform locality-sensitive hashes for face retrieval. *Multimedia, IEEE Transactions on*, 16(4):1090–1103, 2014.

[87] Kaivan Karimi and Gary Atkinson. What the internet of things (iot) needs to become a reality. *White Paper, FreeScale and ARM*, 2013.

[88] Andrzej Kasinski, Andrzej Florek, and Adam Schmidt. The put face database. *Image Processing and Communications*, 13(3-4):59–64, 2008.

[89] Vandana D Kaushik, J Umarani, Amit K Gupta, Aman K Gupta, and Phalguni Gupta. An efficient indexing scheme for face database using modified geometric hashing. *Neurocomputing*, 116:208–221, 2013.

[90] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506. IEEE, 2004.

[91] Brendan Klare and Anil K Jain. Sketch-to-photo matching: a feature-based approach. In *SPIE Defense, Security, and Sensing*, pages 766702–766702. International Society for Optics and Photonics, 2010.

[92] Brendan F Klare and Anil K Jain. Heterogeneous face recognition using kernel prototype similarities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(6):1410–1422, 2013.

[93] Brendan F Klare, Zhifeng Li, and Anil K Jain. Matching forensic sketches to mug shot photos. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):639–646, 2011.

[94] Jan Klıma and Tomáš Skopal. Shape extraction framework for similarity search in image databases. In *Proceedings of the Dateso 2007 Annual International Workshop on DAtabases, TExts, Specifications and Objects*, pages 89–102, 2007.

[95] Scott Klum, Hu Han, Anubhav K Jain, and Brendan Klare. Sketch based face recognition: Forensic vs. composite sketches. In *Biometrics (ICB), 2013 International Conference on*, pages 1–8. IEEE, 2013.

[96] Neeraj Kumar, Peter N Belhumeur, Arijit Biswas, David W Jacobs, W John Kress, Ida C Lopez, and João VB Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision–ECCV 2012*, pages 502–516. Springer, 2012.

[97] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE, 2009.

[98] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[99] Yali Li, Shengjin Wang, Qi Tian, and Xiaoqing Ding. A survey of recent advances in visual feature detection. *Neurocomputing*, 149:736–751, 2015.

[100] Yung-hui Li, Marios Savvides, and Vijayakumar Bhagavatula. Illumination tolerant face recognition using a novel face from sketch synthesis approach and advanced correlation filters. In *Acoustics, Speech and Signal Processing,*

2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, volume 2, pages II–II. IEEE, 2006.

[101] Dahua Lin and Xiaoou Tang. Inter-modality face recognition. In *Computer Vision–ECCV 2006*, pages 13–26. Springer, 2006.

[102] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[103] Qingshan Liu, Xiaoou Tang, Hongliang Jin, Hanqing Lu, and Songde Ma. A nonlinear approach for face sketch synthesis and recognition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 1005–1010. IEEE, 2005.

[104] Wei Liu, Xiaoou Tang, and Jianzhuang Liu. Bayesian tensor inference for sketch-based facial photo hallucination. In *IJCAI*, pages 2141–2146, 2007.

[105] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[106] Pradip Mainali, Qiong Yang, Gauthier Lafruit, Luc Van Gool, and Rudy Lauwereins. Robust low complexity corner detector. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(4):435–445, 2011.

[107] Pradip Mainali, Qiong Yang, Gauthier Lafruit, Rudy Lauwereins, and LV Gool. Lococo: Low complexity corner detector. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 810–813. IEEE, 2010.

[108] Dario Maio and Davide Maltoni. Real-time face location on gray-scale static images. *Pattern Recognition*, 33(9):1525–1539, 2000.

[109] Elmar Mair, Gregory D Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Computer Vision–ECCV 2010*, pages 183–196. Springer, 2010.

[110] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. *Handbook of fingerprint recognition.* Springer Science & Business Media, 2009.

[111] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *Computer Vision - ECCV 2002*, pages 128–142. Springer, 2002.

[112] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[113] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005.

[114] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[115] Tsutomu Miyashita, Peter Meier, Tomoya Tachikawa, Stephanie Orlic, Tobias Eble, Volker Scholz, Andreas Gapel, Oliver Gerl, Stanimir Arnaudov, and Sebastian Lieberknecht. An augmented reality museum guide. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 103–106. IEEE Computer Society, 2008.

[116] Hideyuki Nakashima, Hamid Aghajan, and Juan Carlos Augusto. *Handbook of ambient intelligence and smart environments*. Springer Science & Business Media, 2009.

[117] Fabian Nater, Helmut Grabner, and Luc Van Gool. Exploiting simple hierarchies for unsupervised human behavior analysis. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2014–2021. IEEE, 2010.

[118] Randal C Nelson. Memory-based recognition for 3-d objects. In *ARPA Image Understanding Workshop*, pages 1305–1310. Citeseer, 1835.

[119] João Camargo Neto, George E Meyer, and David D Jones. Individual leaf extractions from young canopy images using gustafson–kessel clustering and a genetic algorithm. *Computers and Electronics in agriculture*, 51(1):66–85, 2006.

[120] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2161–2168. IEEE, 2006.

[121] H Nizami, Jeremy P Adkins-Hill, Yong Zhang, John R Sullins, Christine McCullough, Shaun Canavan, and Lijun Yin. A biometric database with rotating head videos and hand-drawn face sketches. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, pages 1–6. IEEE, 2009.

[122] Kohtaro Ohba and Katsushi Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(9):1043–1047, 1997.

[123] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[124] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[125] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[126] Devi Parikh and Gavin Jancke. Localization and segmentation of a 2d high capacity color barcode. In *Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on*, pages 1–6. IEEE, 2008.

[127] P Jonathon Phillips, Patrick J Flynn, Todd Scruggs, Kevin W Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. Overview of the face recognition grand challenge. In *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, volume 1, pages 947–954. IEEE, 2005.

[128] Arthur R Pope and David G Lowe. Probabilistic models of appearance for 3-d object recognition. *International Journal of Computer Vision*, 40(2):149–167, 2000.

[129] Richard J Radke. A survey of distributed computer vision algorithms. In *Handbook of Ambient Intelligence and Smart Environments*, pages 35–55. Springer, 2010.

[130] KV Ravi Kanth, Divyakant Agrawal, and Ambuj Singh. Dimensionality reduction for similarity searching in dynamic databases. In *ACM SIGMOD Record*, volume 27, pages 166–176. ACM, 1998.

[131] Gillian Rhodes and Tanya Tremewan. Understanding face recognition: Caricauture effects, inversion, and the homogeneity problem. *Visual Cognition*, 1(2-3):275–311, 1994.

[132] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515. IEEE, 2005.

[133] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[134] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):105–119, 2010.

[135] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

[136] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[137] Albert Ali Salah, Theo Gevers, Nicu Sebe, and Alessandro Vinciarelli. Computer vision for ambient intelligence. *Journal of ambient intelligence and smart environments*, 3(3):187–191, 2011.

[138] Hans Schaffers, Nicos Komninos, Marc Pallot, Miguel Aguas, Esteve Almirall, Tuba Bakici, Jean Barroca, Dave Carter, Michel Corriou, Joana Fernadez, et al. Smart cities as innovation ecosystems sustained by the future internet. 2012.

[139] C Schmid et al. Selection of scale-invariant parts for object class recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 634–639. IEEE, 2003.

[140] Nicu Sebe. Multimodal interfaces: Challenges and perspectives. *Journal of Ambient Intelligence and smart environments*, 1(1):23–30, 2009.

[141] Mojtaba Seyedhosseini, Mehdi Sajjadi, and Tolga Tasdizen. Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2168–2175. IEEE, 2013.

[142] Gregory Shakhnarovich. *Learning task-specific similarity*. PhD thesis, Massachusetts Institute of Technology, 2005.

[143] Linda Shapiro and George C Stockman. Computer vision. 2001. *ed: Prentice Hall*, 2001.

[144] Eitan Sharon, Meirav Galun, Dahlia Sharon, Ronen Basri, and Achi Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810–813, 2006.

[145] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[146] Peng-Lang Shui and Wei-Chuan Zhang. Corner detection and classification using anisotropic directional derivative representations. *Image Processing, IEEE Transactions on*, 22(8):3204–3218, 2013.

[147] Robert Sim and Gregory Dudek. Learning generative models of scene features. *International Journal of Computer Vision*, 60(1):45–61, 2004.

[148] Dario Maio Simone Buoncompagni and Vincent Lepetit. Leaf segmentation under loosely controlled conditions. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editor, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 133.1–133.12. BMVA Press, September 2015.

[149] Ashutosh Singh, Jin Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 509–516. IEEE, 2014.

[150] Amos Sironi, Engin Türetken, Vincent Lepetit, and Pascal Fua. Multiscale centerline detection. Technical report, Institute of Electrical and Electronics Engineers, 2014.

[151] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.

[152] Stephen M Smith and J Michael Brady. Susan - a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[153] Lauro Snidaro, Christian Micheloni, and Cristian Chiavedale. Video security for ambient intelligence. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(1):133–144, 2005.

[154] João VB Soares and David W Jacobs. Efficient segmentation of leaves in semi-controlled conditions. *Machine vision and applications*, 24(8):1623–1643, 2013.

[155] Oskar Söderkvist. Computer vision classification of leaves from swedish trees. 2001.

[156] Christoph Strecha, Alexander M Bronstein, Michael M Bronstein, and Pascal Fua. Ldahash: Improved matching with smaller descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(1):66–78, 2012.

[157] Yaniv Taigman, Lior Wolf, Tal Hassner, et al. Multiple one-shots for utilizing class label information. In *BMVC*, pages 1–12, 2009.

[158] Xiaoou Tang and Xiaogang Wang. Face photo recognition using sketch. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–257. IEEE, 2002.

[159] Xiaoou Tang and Xiaogang Wang. Face sketch synthesis and recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 687–694. IEEE, 2003.

[160] Xiaoou Tang and Xiaogang Wang. Face sketch recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):50–57, 2004.

[161] Karen T Taylor. *Forensic art and illustration*. CRC Press, 2000.

[162] Massimo Tistarelli and Ben Schouten. Biometrics in ambient intelligence. *Journal of Ambient Intelligence and Humanized Computing*, 2(2):113–126, 2011.

[163] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):815–830, 2010.

[164] Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features.* School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[165] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008.

[166] Tinne Tuytelaars and Cordelia Schmid. Vector quantizing feature space with a regular lattice. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[167] Robert G Uhl Jr and Niels da Vitoria Lobo. A framework for recognizing a facial image from a police sketch. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 586–593. IEEE, 1996.

[168] http://bias.csr.unibo.it/franco/SB/DispensePDF/.

[169] http://mmlab.ie.cuhk.edu.hk/archive/cufsf/.

[170] http://mmlab.ie.cuhk.edu.hk/archive/facesketch.html.

[171] https://developer.vuforia.com/library/articles/Solution/Compiling-and-Running-a-Vuforia-iOS-Sample-App.

[172] http://utopia.duth.gr/~dchrisos/pubs/database2.html.

[173] http://wordlesstech.com/augmented-reality-vision-pilots/.

[174] http://www.digit.in/technology-guides/fasttrack-to-augmented-reality/different-types-of-augmented-reality.html.

[175] http://www.eecs.berkeley.edu/research/projects/Cs/vision/grouping/resources.html.

[176] http://www.lrv.fri.uni-lj.si/facedb.html.

[177] http://www.robots.ox.ac.uk/~vgg/research/affine/.

[178] Koen EA Van De Sande, Theo Gevers, and Cees GM Snoek. Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1582–1596, 2010.

[179] Joost Van de Weijer, Theo Gevers, and Andrew D Bagdanov. Boosting color saliency in image feature detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):150–156, 2006.

[180] Merijn Van Erp and Lambert Schomaker. Variants of the borda count method for combining ranked classifier hypotheses. In *IN THE SEVENTH INTERNATIONAL WORKSHOP ON FRONTIERS IN HANDWRITING RECOGNITION. 2000. AMSTERDAM LEARNING METHODOLOGY INSPIRED BY HUMAN'S INTELLIGENCE BO ZHANG, DAYONG DING, AND LING ZHANG*. Citeseer, 2000.

[181] DWF Van Krevelen and R Poelman. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality*, 9(2):1, 2010.

[182] TN Vikram, Chidananda Gowda, DS Guru, and Shalini R Urs. Face indexing and retrieval by spatial similarity. In *2008 Congress on Image and Signal Processing*, pages 543–547. IEEE, 2008.

[183] Vasillios Vonikakis, Dimitrios Chrysostomou, Rigas Kouskouridas, and Antonios Gasteratos. A biologically inspired scale-space for illumination invariant feature detection. *Measurement Science and Technology*, 24(7):074024, 2013.

[184] Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(11):1955–1967, 2009.

[185] James Wayman, Anil Jain, Davide Maltoni, and Dario Maio. *An introduction to biometric authentication systems*. Springer, 2005.

[186] Markus Weber, Max Welling, and Pietro Perona. *Unsupervised learning of models for recognition*. Springer, 2000.

[187] Andrew Willis and Yunfeng Sui. An algebraic model for fast corner detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2296–2302. IEEE, 2009.

[188] Simon Winder, Gang Hua, and Michael Brown. Picking the best daisy. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 178–185. IEEE, 2009.

[189] Laurenz Wiskott, Jean-Marc Fellous, N Kuiger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):775–779, 1997.

[190] John Wright and Gang Hua. Implicit elastic matching with random projections for pose-variant face recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1502–1509. IEEE, 2009.

[191] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pages 11–16. IEEE, 2007.

[192] Zhong Wu, Qifa Ke, Jian Sun, and Heung-Yeung Shum. Scalable face image retrieval with identity-based quantization and multireference reranking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(10):1991–2001, 2011.

[193] Gui-Song Xia, Julie Delon, and Yann Gousseau. Accurate junction detection and characterization in natural images. *International journal of computer vision*, 106(1):31–56, 2014.

[194] Bing Xiao, Xinbo Gao, Dacheng Tao, and Xuelong Li. A new approach for face recognition by sketches in photos. *Signal Processing*, 89(8):1576–1588, 2009.

[195] Mingqiang Yang, Kidiyo Kpalma, and Joseph Ronsin. A survey of shape feature extraction techniques. *Pattern recognition*, pages 43–90, 2008.

[196] B Yanikoglu, Erchan Aptoula, and C Tirkaz. Automatic plant identification from photographs. *Machine vision and applications*, 25(6):1369–1383, 2014.

[197] Berrin A Yanikoglu, Erchan Aptoula, and Caglar Tirkaz. Sabanci-okan system at imageclef 2012: Combining features and classifiers for plant identification. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[198] Pong C Yuen and CH Man. Human face image searching system using sketches. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 37(4):493–504, 2007.

[199] Dengsheng Zhang and Guojun Lu. A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval. *Journal of Visual Communication and Image Representation*, 14(1):39–57, 2003.

[200] Dengsheng Zhang, Guojun Lu, et al. A comparative study of fourier descriptors for shape representation and retrieval. In *Proc. 5th Asian Conference on Computer Vision*. Citeseer, 2002.

[201] Wei Zhang and Jana Košecká. Hierarchical building recognition. *Image and Vision Computing*, 25(5):704–716, 2007.

[202] Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Lighting and pose robust face sketch synthesis. In *Computer Vision–ECCV 2010*, pages 420–433. Springer, 2010.

[203] Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 513–520. IEEE, 2011.

[204] Xiaohong Zhang, Hongxing Wang, Andrew WB Smith, Xu Ling, Brian C Lovell, and Dan Yang. Corner detection based on gradient correlation matrices of planar curves. *Pattern Recognition*, 43(4):1207–1223, 2010.

[205] Yong Zhang, Steve Ellyson, Anthony Zone, Priyanka Gangam, John Sullins, Christine McCullough, Shaun Canavan, and Lijun Yin. Recognizing face sketches by a large number of human subjects: A perception-based study for facial distinctiveness. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 707–712. IEEE, 2011.

[206] Yong Zhang, Christine McCullough, John R Sullins, and Christine R Ross. Hand-drawn face sketch recognition by humans and a pca-based algorithm

for forensic applications. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40(3):475–485, 2010.

[207] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003.