**Università degli Studi di Modena e Reggio Emilia**

DIPARTIMENTO DI INGEGNERIA "ENZO FERRARI"

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

# Hand Gesture recognition on wearable vision devices

*Candidato:*
Lorenzo Baraldi

*Relatore:*
Prof. Rita Cucchiara
*Correlatore:*
Ing. Giuseppe Serra

ANNO ACCADEMICO 2013-2014

# *Abstract in lingua italiana*

**Analisi e Sviluppo di Metodi per il Riconoscimento di Gesti con Dispositivi di Visione Wearable**

Lorenzo Baraldi

Il regolamento della Facoltà di Ingegneria di Modena prevede che le tesi scritte in lingua inglese debbano contenere un'ampia sintesi dei contenuti in lingua italiana: in accordo a questa regola, proponiamo un sunto degli algoritmi, delle tecniche e dei risultati che verranno delineati. Si noti che non si tratta di un sunto esaustivo, e che non è possibile valutare la tesi dalla semplice lettura di queste righe. Per una descrizione più dettagliata e più rigorosa, e per i risultati sperimentali ottenuti, si rimanda al testo in inglese.

Questa tesi tratta dello sviluppo di nuovi algoritmi per la segmentazione delle mani e il riconoscimento dei gesti in video registrati da telecamere ego-centriche, cioè posizionate in maniera solidale con l'utente. Nel nostro caso, dopo aver studiato le proprietà di diverse posizioni e delle relative inquadrature, si è scelto di porre le telecamere sulla testa dell'utente o sul petto. Il fine degli algoritmi e delle tecniche proposte è quello di creare nuove modalità di interazione con opere d'arte, in ambiente museale o in generale di *cultural heritage*: il visitatore, dopo aver indossato una telecamera ego-vision e un dispositivo di elaborazione wearable, può richiedere informazioni sull'opera d'arte che sta visualizzando replicando con le mani gli stessi gesti che utilizzerebbe per chiedere informazioni a una guida. L'idea sottostante gli algoritmi che proponiamo, dunque, è che si possa costruire un nuovo tipo di interfaccia uomo-macchina, non più materiale, come le interfacce a cui siamo abituati, ma del tutto immateriale, e che si basa sul semplice movimento delle mani.

Per poter identificare gesti espressi attraverso le mani, è necessario innanzitutto identificare le mani stesse. Pertanto, nel capitolo 3 proponiamo un nuovo algoritmo di segmentazione delle mani, che parte da una fase di pre-segmentazione in superpixels e viene costruito in modo da essere stabile a diverse condizioni di illuminazione e in modo da produrre maschere di segmentazione coerenti sia dal punto di vista spaziale che da quello temporale. Dopo aver pre-segmentato il frame in superpixels, utilizzando l'algoritmo SLIC [1] (si veda la figura 3.1 per un esempio), i singoli superpixels vengono classificati mediante una pluralità di classificatori Random Forest, ognuno dei quali è costruto in modo da imparare l'apparenza delle mani in una determinata condizione di illuminazione. A tempo di test, vengono selezionati i

classificatori più vicini alle condizioni di illuminazione che si presentano: le loro predizioni vengono poi pesate per produrre una stima finale dell'appartenenza di ciascun superpixel alla mano. Una volta classificato ogni superpixel modo indipendente dagli altri, intervengono le sopracitate tecniche di coerenza spaziale e temporale, che agiscono a livello di pixel. La coerenza temporale fa si che la predizione del singolo pixel sia espressa come somma pesata delle predizioni dei frame precedenti, coprendo in questo modo errori di classificazione estemporanei. D'altro canto, le predizioni così ottenute vengono inserite nell'algoritmo GrabCut [2], che provvede a rivedere i contorni della maschera di segmentazione applicando iterativamente il Graph Cut e basandosi su modelli colore ottenuti tramite misture di gaussiane. La pipeline così ottenuta viene testata su dataset pubblicamente disponibili e su dataset di nostra produzione, dimostrando performances migliori dello stato dell'arte presente in letteratura (si veda la tabella 3.3).

Una volta discusso il problema della segmentazione delle mani, la tesi si concentra sul riconoscimento dei gest all'interno di sequenze di frames di lunghezza fissa. La tecnica che proponiamo, anche in questo caso, supera gli approcci esistenti di riconoscimento dei gesti e delle azioni (tabella 4.1), ed è dunque di interesse anche da un punto di vista di ricerca. L'idea è quella di estrarre punti di interesse nelle regioni delle mani, e tracciarli durante il video. Le traiettorie così ottenute vengono descritte non solo nella loro forma (mediante un Trajectory Descriptor ispirato da [3]), ma anche nel moto e nell'apparenza di un loro intorno, tramite i descrittori HOG, HOF e MBH. Si ottiene così, data una sequenza di frames, un numero variabile di traiettorie, che viene poi codificato mediante un approccio Bag of Words (si veda per un dettaglio la sezione 4.1.3). Il descrittore finale, che dunque è un'istogramma, viene power-normalizzato e quindi classificato tramite un classificatore a massimo margine lineare SVM.

L'ultimo capitolo, infine, si occupa di estendere l'approccio di riconoscimento dei gesti, pensato per sequenze di lunghezza fissa, al caso del riconoscimento di gesti, in tempo reale, da un frame stream proveniente da una telecamera. Per fare ciò, ci ispiriamo ai problemi di *label sequence learning*, proponiamo una versione leggermente modificata dei nostri descrittori e sostituiamo il classificatore lineare SVM con un più complesso classificatore SVM strutturato, a sua volta ispirato agli Hidden Markov Models. In altre parole, si modificano i descrittori in modo da passare da descrittori relativi a una sequenza di frames a descrittori relativi al singolo frames, e si affida all'SVM strutturato il compito di catturare le dipendenze temporali tra più frames. Compito del classificatore strutturato è infatti, in questo caso, associare una label (corrispondente a un gesto o a un *non gesto*, cioè a una situazione in cui non

viene compiuto alcun gesto) a ciascun frames, tenendo conto delle dipendenze temporali tra la label del frame corrente e i descrittori dei frame precenti, e tra la label del frame corrente e le label associate ai frames precedenti. Rispetto al caso del classificatore lineare, si guadagna sia in termini di una migliore descrizione delle dipendenze temporali, sia perchè si ha una classificazione on-line, in contrasto con quella precedente, che necessariamente ha un ritardo di alcuni frames.

Come si è detto l'ultimo capitolo vuole proporre un riconoscimento dei gesti in tempo reale. Per questo scegliamo una piattaforma wearable su cui implementare i nostri algoritmi ed effettuare test, la board Odroid XU, e impieghiamo diverse tecniche di ottimizzazione, sia per sfruttare le potenzialità di un'architettura multicore mediante il multithreading, sia per sfruttare le capacità SIMD dei singoli processori. Inoltre, valutiamo l'uso della GPU embedded nel dispositivo. Impiegando tali tecniche di ottimizzazione passiamo da una prima versione sequenziale da 4.3 frames/s a una versione definitiva da 14 frames/s, capace di estrarre traiettorie di buona qualità in tempo reale. L'implementazione real-time così ottenuta, costituisce di fatto la realizzazione dell'interfaccia uomo-macchina basata su gesti obiettivo della tesi, e viene testata in due applicazioni reali: in una *ego-vision jacket*, in cui la board viene posta all'interno di una giacca da uomo, e si affrontano i conseguenti problemi hardware di surriscaldamento dei processori, e una *gesture-based interface* per GUI e applicazioni desktop, con la quale è possibile controllare, mediante il movimento delle mani, interfacce grafiche pensate per ambienti desktop, come presentazioni Power Point (si veda, a tal proposito, la demo diponibile al seguente URL: `http://www.lorenzobaraldi.com/files/EgoVision_HCI.wmv`).

Il lavoro svolto è inoltre testimoniato da due *papers*, pubblicati in due workshops, e due articoli su rivista, attualmente in fase di revisione.

*To Emanuela*

# *Acknowledgements*

Foremost, I would like to express my sincere gratitude to my advisors Prof. Rita Cucchiara and Dr. Giuseppe Serra for the continuous support of my study, for their patience, motivation, enthusiasm, and knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors for my study.

Besides my advisors, I would like to thank the rest of the Imagelab research group: Costantino Grana, Simone Calderara, Michele Fornaciari, Marco Manfredi, Paolo Santinelli, Augusto Pieracci, Roberto Vezzani, Franesco Solera, Martino Lombardi and Patrizia Varini, for their encouragement, insightful comments, and hard questions. I am also grateful to Prof. Greg Mori, for enlightening me the first glance of research during a visit in Modena.

Special thanks go to my labmates Stefano Alletto and Francesco Paci, for the stimulating discussions, for the days we were working together before deadlines, and for all the fun we have had in the last months. Also I thank my friends Chiara Ferrari, Emanuele Benatti, Xhensila Doda, Davide Setti, and in particular Michela Benedetti, who encouraged me to study computer vision.

Last but not the least, I would like to thank my parents Franco and Elisabetta, for giving birth to me at the first place and supporting me throughout my life, and my sister Alessia.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Egocentric Vision and Wearable devices: an overview

## 1.1 Introduction

Portable head-mounted cameras, able to record dynamic high quality first-person videos, have become a common item among sportsmen over the last five years. These devices represent the first commercial attempts to record experiences from a first-person perspective. This technological trend is a follow-up of the academic results obtained in the late 1990s, combined with the growing interest of the people to record their daily activities. As a recent survey on first person vision [4] recalls, the idea of recording and analyzing videos from first person perspective is not new. To mention some examples: In 1998 Mann proposed the WearCam [5]. Later in 2000, Mayol *et al.* proposed a necklace device [6] , and in 2005 Mayol et al. developed an active shoulder mounted camera [7]. In 2006, the Microsoft Research Center started to use the SenseCam for research purposes [8], while Pentland *et al.* [9] developed a wearable data collector system (InSense). Finally, it is important to highlight the work of Mann, who, since 1978, has been working on his own family of devices.

Up to date, no consensus has yet been reached in the literature with respect to naming this video perspective. *First Person Vision* (FPV) is probabily the most commonly used term, but also *Egocentric Vision* has also recently grown in popularity, and will be used in the rest of this thesis. In the awakening of this technological trend, Google announced the Project Glass in 2012. The company started publishing short previews on the Internet demonstrating the

Glasses FPV recording capabilities. This was coupled by the ability of the device to show relevant information to the user through the head-up display. The main idea of the Project Glass is to use a wearable computer to reduce the time between intention and action. In this thesis, we try to move a step forward, and realize a wearable vision device with better computing capacities and thus able to execute complex computer vision algorithms.

## 1.2 On the positioning of the camera

The ego-centric approach consists in equipping the users with wearable sensors that observe their activities: these devices see what the user sees and look where the user is looking. One of the first questions to solve, of course, is where to put the camera. Positioning an optical device on the human body is quite a problematic task, as occlusion, motion, social issues as well as criteria related to the purpose of the device must be taken into account. Following the work of Mayol *et al.* [10], in this section we give a detailed overview on the best places where to put a wearable camera.

Cameras used for wearable applications fall into two categories for this discussion; static narrow-view devices and omnidirectional devices. Omnidirectional devices include catadioptric, fish-eye and active systems where either the entire field-of-viewis imaged at low resolution, or in the active case the high-resolution narrow-view sensor moved to any orientation. Narrow-view static cameras can only ever see a small part of the user or their environment, and placement is therefore entirely driven by the task. For wide-angle or omnidirectional sensors placement is less constrained and a range of positions are possible.

A variety of solutions appear in the literature. In [11, 12], hat-mounted cameras have been used to look down at the user's hands and reaching space, whereas in [13] cameras are strapped to the wearer's hands themselves. In [14], a hat-mounted camera looks forward, an orientation also used when the camera is attached to a head mounted display [5]. In contrast, [15] uses a camera worn on the chest, in [16] an omnidirectional camera is used above the head, and a wide-angle lens camera mounted at the back in [17].

Mayol *et al.* [10] identify three frames of reference for measurements that a wearable sensor makes:

1. relative to the user body (e.g. sensing the manipulative space in front of the user's chest)

2. relative to the static world (e.g. sensing the ceiling/floor texture to infer user's location)

3. relative to an independent object (e.g. tracking an interesting object)

This task-oriented classification can help us to understand the criteria that should be considered. For working in the user frame alone all that is required is a stable view of the chosen area — often the handling space, and absolute field-of-view may be less important. For sensing the outside environment user occlusion is problematic and absolute field-of-view is more important. Both occlusion and user motion are problems when fixating resolution or processing on a particular part of the environment or independently moving object.

To simulate and compare positions for optical devices around a human body, we must first simulate the human form: [10] use a female example from the Human Animation Working Group[1], consisting of about 1000 markers (points) and 1800 polygons arranged into 16 body-segments which can be independently rotated to simulate any natural pose. They have created a software[2] to allow the simulated optical device to be positioned arbitrarily in space around the body, or for faster automatic tests placed a distance above any of the humanoid's polygons. The utility of such a model is that the variables of position and distance above the body-surface can be varied automatically, allowing tests for a range of device heights over the whole body (which would be tedious at best on a real person).

Determination of occlusion in any direction can be made by emitting a ray from the chosen device centre and checking for intersection with any of the component polygons. Only polygons facing the camera need be considered, and refinements to further reduce the number of tests are widely reported in the ray-tracing literature. For visualization it is also useful to consider emitting rays from the device centre as equivalent to a central projection onto a unit sphere. This yields representations such as Figure 1.1, where the head is clearly visible to the right with the shoulder below it. The proportion of the sphere surface not occluded gives the absolute field-of-view.

In terms of absolute field of view, the head, the position of choice for many researchers (and nature), is favourable, as are the shoulders. This comes as no surprise: however it is important to note which alternative placements are favorable if these locations can not be used (e.g. for social reasons). A further consideration is that raising the sensor away from the body might reduce the occlusion. Figure 1.2 shows the amount of occluision in some of the

[1] http://www.h-anim.org
[2] freely available at http://www.robots.ox.ac.uk/~wmayol/3D/nancy_matlab.html

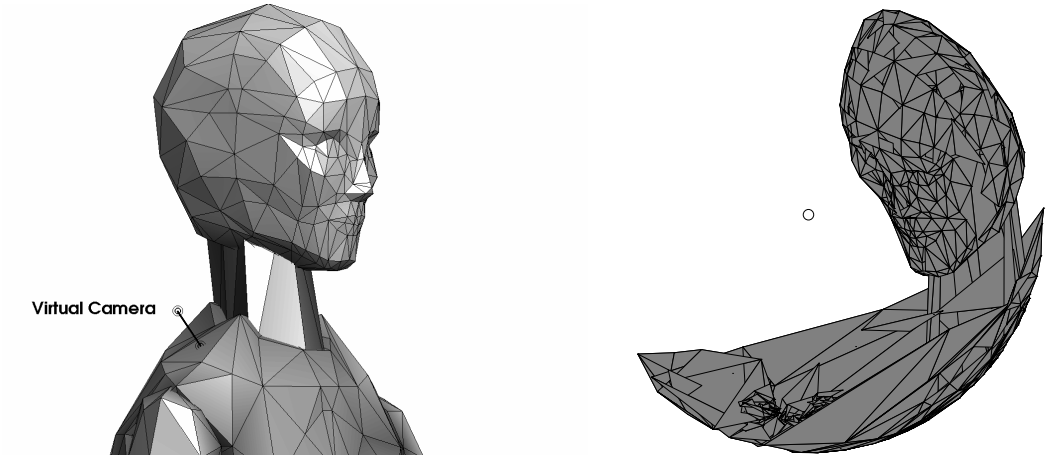



FIGURE 1.1: Left: The device is positioned above the centroid of the polygon, a short distance (37.5mm) along the surface normal. Right: The view from such a camera shows the head and shoulder clearly, but the rest of the body is obscured.

favoured positions and varies the sensor height from resting on the surface (height=0mm) to far above (height=150mm). At all of the chosen positions the occlusion reduces as the height is increased, but the most significant gains are made for the head-mounted positions and shoulder area. Whichever position is chosen, the field of view is always significantly improved if the sensor can be raised a few centimeters from the surface. However, increases beyond 50mm make no significant improvement in the field of view for this model.

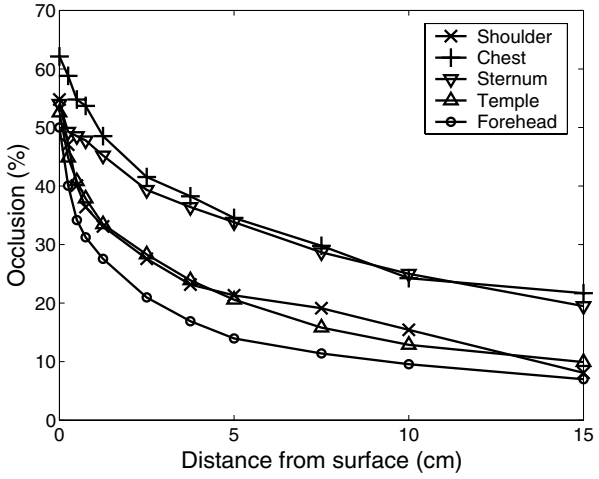



FIGURE 1.2: The proportion of the view obscured against the distance from the body.

The amount of usermotion must also be taken into account. The type of motion that may be encountered can vary enormously and may be task dependent. With an articulated model it

is possible to analyze any motion for which the joint angle evolutions over time are known. In figure 1.3, we show the amount of motion . The arms and legs suffer the highest motions, with most of the torso and head relatively still.

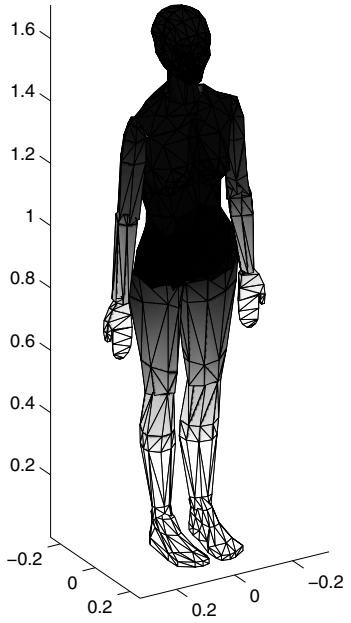



FIGURE 1.3: Amount of motion: arms and legs suffer the most severe motions during walking.

Now, having considered and evaluated occlusion, motion, and distance, it is clear that the best places to put a wearable camera, with respect to these parameters, are head and shoulders. Of course, an head mounted camera will suffer a bigger motion, but can reveal attentional cues, and it is thus the best choice for action and activity recognition. Shoulder mounted cameras, on the other hand, are less invasive, and are still a good choice for social interaction analysis and human-machine interfaces. In this thesis we will explore both options.

## 1.3 Wearable devices

The term *wearable device* refers to electronic technologies or computer that are incorporated into items of clothing and accesories which can be worn on the body. In our case, we are mainly interested in developing wearable *vision* devices, i.e. wearable devices capable to see and process video data. In this section, we will provide a brief description for some of these devices and our experience in using a few of them.

### 1.3.1 Go-Pro

GoPro offers a series of small and high-quality cameras. These cameras can be mounted on various body parts or objects such as a helmet, chest and snow board which makes them very flexible. The GoPro Hero 1 has a fish-eye lens with a 110 view angle. The view angle has increased to 170 in the more recent Hero 2 and 3 cameras. The quality of the video is in HD, with minimal motion blur and other artifacts. The quality of the video is much better outdoors, and the video sometimes becomes very dark indoors. The main advantage of the GoPro camera is its wide field of view and its high quality video. On the other hand, its disadvantage is its bulkiness, and the fact that other individuals in the scene become very aware of it. The battery lasts for about 2 hours during continuous video capture. The data is recorded on a SD card. A 16GB SD card suffices for storing 2 hours of HD video. The price of GoPro is around 200$.

### 1.3.2 Tobii Eye-Tracking Glasses

Tobii offers various kinds of eye-tracking products. Most of their devices are static and monitorbased. However, they have a few mobile eye-tracking systems as well. The system consists of an outward looking camera that captures the scene in front of the user, and an inward looking infrared camera that tracks the subject's right eye. The glasses connect to a pocket size recording device. Before or after data collection, the system needs to be calibrated in order to correctly estimate the gaze point. Calibration is very intensive and becomes very hard for some subjects. The resolution of the video is $640 \times 480$ and the frame rate is 30 fps. The video quality is low and there exist severe motion blur and interlacing effects. On the plus size, the gaze tracking is accurate in comparison to the wearable gaze-tracking devices of other companies. The view field of the camera is around $600 \times 400$. The price of Tobii eye-tracking glasses is around 30,000$.

### 1.3.3 SMI Eye-Tracking Glasses

SMI produced a device similar to Tobii's eye-tracking glasses in January 2012. They have tried to fix some of the issues that exist in the Tobii's system. In particular, their system is easier to calibrate, records a video in HD, and the glasses are more tolerable on the face. The SMI system has two eye-tracking infrared cameras looking at both eyes which results in an

easier calibration in comparison to Tobii's system. An issue that exists in the SMI glasses is that the video is blurred and dark on the frame boundaries. The price of SMI eye-tracking glasses is around 24,000$.

### 1.3.4 Pivothead Glasses

Pivothead has introduced a relatively cheap pair of glasses that have an outward looking camera that captures the scene in front of the user. Obviously this is a cheaper system in comparison to SMI and Tobii because it doesn't track the eyes. The video quality is HD and it can capture for an hour. The only issue with the pivothead glasses is that the camera's field of view is very narrow, even narrower than that of SMI and Tobii systems. The price of Pivothead glasses is around 300$. They also provide a 100$ device which can transmit video to a laptop in realtime.

### 1.3.5 Google Glass

Google Glass is about to become available for public use. The system records a 720p video and takes 5-megapixel images. It can connect to the internet and any bluetooth-capable phone. In addition, Glass has a heads up display (HUD) creating an illusion equivalent to viewing a 25-inch high definition screen from eight feet away. It has 16 GB of RAM, 12 GB of which are usable for apps. Furthermore, it has a microphone, similar to all of the previously mentioned wearable devices.

### 1.3.6 Panasonic HX-A10

The Panasonic HX-A10, rather than relying on mounts and accessories, is equipped with an "earhoo" feature and a remote processing unit, which allows users to attach the camera to their head right out of the box, and no helmet is required. The device itself records at $1920 \times 1080$ video at 60 fps, $1280 \times 720$ at 120 fps, and $640 \times 360$ at 240 fps for slow shots. It has Wi-Fi and can setup a live stream; and battery life for over two hours of filming on a single charge.

The actual lens (F2.5 bright lens, 1/4.1-inch BSI Sensor) part sports a two-foot cable and only weighs in at 4 ounces. It also communicates with a smartphone or tablet while watching in

live view. We used the Panasonic HX-A10 to record two datasets, the Interactive Museum dataset and the Maramotti dataset, taken into the Maramotti Collection (Figure 1.4). The lens helps to capture a wide field of view, which is necessary for observing first-person's hands during daily activities. The fish-eye distortion, however, creates some challenges for using the data.


FIGURE 1.4: We used Panasonic HX-A10 for collecting the Interactive Museum dataset and the Maramotti dataset.

### 1.3.7 Genius WideCam F100

The Genius WideCam F100 is a wide angle HD camera that we used in some of our tests, and is the camera mounted on our wearable vision device. It can record 1080p frames at up to 30 fps and features ultra wide angle lens (up to 120 degrees) and built-in microphones (see figure 1.5).

### 1.3.8 A wearable computing device: the Odroid-XU board

The wearable ego-vision devices we have built embeds a glass-mounted camera and an Odroid-XU developer board, serving as video-processing and network communication unit.


(a) (b)

FIGURE 1.5: Genius WideCam F100.

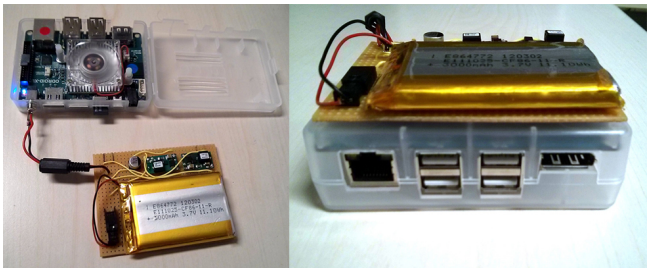


FIGURE 1.6: The Odroid-XU board with battery pack.

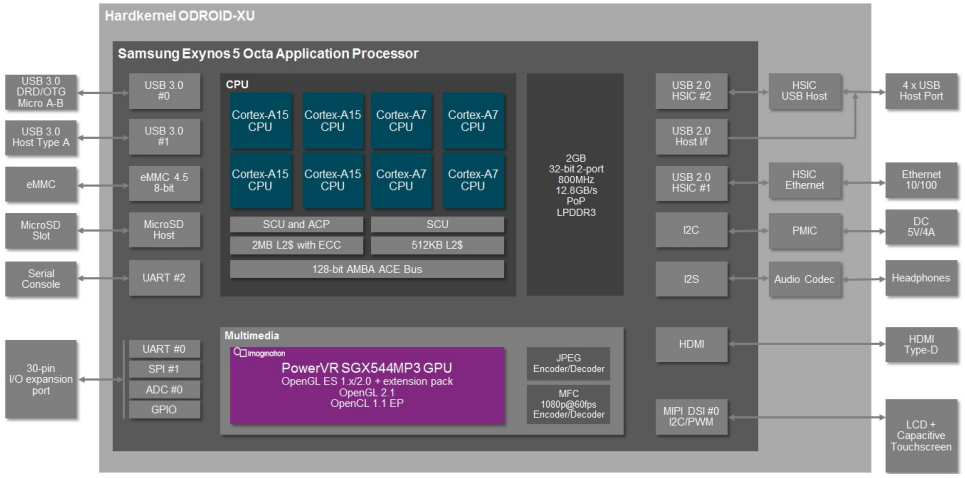



FIGURE 1.7: Hardkernel Odroid-XU block diagram

There are several benefits in using such a portable device: the commercial availability and low costs for prototypes evaluation, the computational power and energy efficiency of the Big-Little architecture. Furthermore it has the possibility of peripheral addition to extend connections and input devices.

Hardkernel have made a name for themselves within the open-source community by delivering high performance development boards at affordable prices. The Odroid-XU platform boasts an Exynos 5 Octa processor – the same SoC found inside the Samsung Galaxy S4, which includes a PowerVR SGX544MP3 GPU clocked at around 600 MHz (see figure 1.7).

The key features and specifications of the ODROID-XU development board include:

- big.LITTLE processing based on the Cortex™-A15 and Cortex™-A7 quad core CPUs
- PowerVR SGX544MP3 GPU (OpenGL ES 2.0, OpenGL ES 1.1, OpenCL 1.1 EP, Renderscript/Filterscript)

- 2 GB LPDDR3 PoP (1600Mbps/pin, 2 x 32bit Bus)
- USB 3.0 Host x 1, OTG x 1, USB 2.0 Host x 4
- HDMI 1.4a output Type-D connector (Micro-HDMI)
- eMMC 4.5 Flash Storage
- Micro-SD socket
- MIPI DSI for LCD display output
- On-board Audio Codec
- Fast 10/100 Ethernet LAN
- WiFi
- 5V/4A power supply

Big.LITTLE is a heterogeneous computing architecture developed by ARM that couples slower, low-power processor cores with more powerful and power-hungry ones. The intention being to create a multi-core processor that can adjust better to dynamic computing needs and use less power than clock scaling alone. In this case we have the Cortex-A7 coupled with the Cortex-A15, which have been designed to be architecturally compatible.

In the Samsung Exynos 5 there is only one way for the different processor cores to be arranged in a big.LITTLE design: the *clustered model* approach. With this approach the operating system scheduler can only see one of the two processor clusters, when the load on one cluster hits a certain point, the system transitions to the other cluster. All relevant data is passed through the common L2 cache, the first core cluster is powered off and the other one is activated. A Cache Coherent Interconnect (CCI) is used.

## 1.4 About this thesis: a new gesture-based interface for cultural heritage

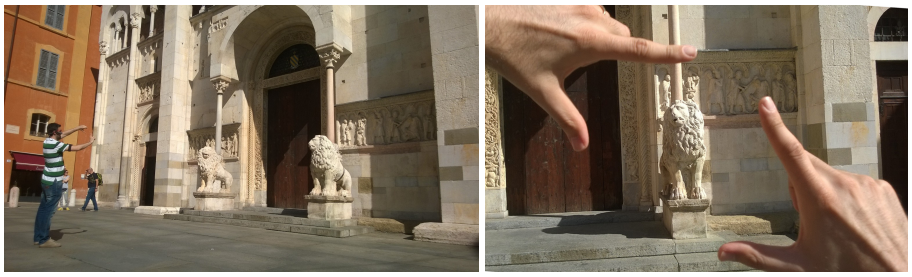
We would like to close this first chapter with a brief introduction to the motivation of this thesis. Our goal is to develop a new gesture-based human-machine interface, specifically designed for cultural heritage. As a matter of fact, in recent years the interest in cultural

heritage has reborn, and the cultural market is becoming a cornerstone in many national economic strategies. In the United States, a recent report of the Office of Travel and Tourism Industries claims that 51% of the 40 million Americans traveling abroad visit historical places; almost one third visit cultural heritage sites; and one quarter go to an art gallery or museum [18]. The same interest is found in Europe, where the importance of the cultural sector is widely acknowledged, South Asia and North Africa. The latest annual research from World Travel and Tourism Council shows that travel and tourism's total contribution to total GDP grew by 3.0% in 2013, faster than overall economic growth for the third consecutive year [19].

Consequently, to deal with an increasing percentage of "digital native" tourists, a big effort is under way to propose new interfaces for interacting with the cultural heritage. In this direction goes the solution "SmartMuseum" proposed by Kuusik *et al.* [20]: by the means of PDAs and RFIDs, a visitor can gather information about what the museum displays, building a customized visit based on his or her interests inserted, prior to the visit, on their website. This project brought an interesting novelty when first released, but it has some limitations. First, being tied to RFIDs does not allow reconfiguring the museum without rethinking the entire structure of the exhibition. Furthermore, researches demonstrated how the use of mobile devices on the long term decreases the quality of the visit due to their users paying more attention to the tool rather than to the work of art itself.

In 2007 Kuflik *et al.* [21] proposed a system to customize visitors experiences in museums using software capable of learning their interests based on the answers to a questionnaire that they compiled before the visit. Similarly to SmartMuseum, one of the main shortcomings of this system is the need to stop the visitor and force him into doing something that he/she might not be willing to do. An interesting attempt to user profiling with wearable sensors was the Museum Wearable [22], a wearable computer which orchestrates an audiovisual narration as a function of the visitors' interests gathered from his/her physical path in the museum. However this prototype does not use any computer vision algorithm for understanding the surrounding environment. For instance the estimation of the visitor location is based again on infrared sensors distributed in the museum space.

Museums and cultural sites still lack of an instrument that provides entertainment, instructions and visit customization in an effective natural way. Too often visitors struggle to find the description of the artwork they are looking at and when they finds it, its detail level could



(a) (b)

FIGURE 1.8: One of the goals of this thesis: natural interaction with artworks.

be too high or too low for their interests. Moreover, frequently the organization of the exhibition does not reflect the visitors' interests leading them to a pre-ordered path which cultural depth could not be appropriate.

To overcome these limitations, we try to enhance visitors' experiences using ego-vision. Ego-vision features glass-mounted wearable cameras able to see what the visitor sees and perceiving the surrounding environment as he does. Our goal is to develop a wearable vision device for museum environments, able to replace the traditional self-service guides and overcoming their limitations and allowing for a more interactive museum experience to all visitors. The aim of our device is to stimulate the visitors to interact with the artwork, reinforcing their real experience, by letting visitors to replicate the gestures (e.g. point out to the part of the painting they're interested in) and behaviors that they would use to ask a guide something about the artwork.

So the main aim of this thesis, from the hardware point of view, is to develop a wearable vision device capable of executing intensive computer vision algorithms, something like the system that appears in Douglas Trumbull's *Brainstorm*, that was capable of recording the sensory and emotional feelings of a subject (Figure 1.9). From the software point of view, we would like to build a new human-computer interface based on gestures (Figure 1.8).

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of conveying meaningful information or interacting with the environment. They constitute one interesting small subspace of possible human motion. A gesture may also be perceived by the environment as a compression technique for the information to be transmitted elsewhere and subsequently reconstructed by the receiver.

FIGURE 1.9: A clip from *Brainstorm* (1983)

Generally, there exist many-to-one mappings from concepts to gestures and vice versa. Hence, gestures are ambiguous and incompletely specified. For example, to indicate the concept *stop*, one can use gestures such as a raised hand with palm facing forward, or, an exaggerated waving of both hands over the head. Similar to speech and handwriting, gestures vary between individuals, and even for the same individual between different instances.

Gestures can be static (the user assumes a certain pose or configuration) or dynamic (with prestroke, stroke, and poststroke phases). Some gestures also have both static and dynamic elements, as in sign languages. Again, the automatic recognition of natural continuous gestures requires their temporal segmentation. Often one needs to specify the start and end points of a gesture in terms of the frames of movement, both in time and in space. Sometimes a gesture is also affected by the context of preceding as well as following gestures. Moreover, gestures are often language- and culture-specific.

In the following of this thesis, we will address all this issues. Our proposed method, in fact, is capable of recognizing both static and dynamic gestures, can process a continuous stream of frames from a wearable camera, and in real-time, without the need to specify the start and end point of the gesture. Moreover, the nature of our classifiers let the user define its own gestures, and therefore its own, personal, way of interacting with the system.

# Chapter 2

# Computer Vision and Machine Learning techniques

To detect hands and recognize gestures, we are going to use the traditional computer vision pipeline, that given an input image, or a frame sequence, detect, extract and codes *descriptors*, and then uses a *learning algorithm* to correctly classify the input. In this chapter, we present some of the most relevant features and machine learning algorithms we are going to exploit in the next sections.

## 2.1 Descriptors

### 2.1.1 Color descriptors

Color is one of the most important features for hand detection, therefore a careful colorspace selection is crucial for this task. One important question is: what is the best colorspace for skin detection, or more generally - is there an optimal colorspace for skin-classification? Surprisingly, many papers on skin detection do not provide strict justification of their colorspace choice, probably because of possibility to obtain acceptable skin detection results on limited dataset with almost any colorspace. In our case, we made an effort to compare different color spaces and ended up using a mixture of diffent spaces. In the following, we present the three color spaces we have elected for our algorithm: RGB, HSV and CIElab.

**RGB colour Space** – RGB is a colorspace originated from CRT (or similar) display applications, when it was convenient to describe color as a combination of three colored rays (red, green and blue). It is one of the most widely used colorspaces for processing and storing of digital image data. However, high correlation between channels, significant perceptual non-uniformity, mixing of chrominance and luminance data make RGB not a very favorable choice for color analysis and colorbased recognition algorithms.

**HSV colour Space** – Hue-saturation based colorspaces were introduced when there was a need for the user to specify color properties numerically. They describe color with intuitive values, based on the artist's idea of tint, saturation and tone. Hue defines the dominant color (such as red, green, purple and yellow) of an area, saturation measures the colorfulness of an area in proportion to its brightness. In HSV, the value is related to the color luminance. The intuitiveness of the colorspace components and explicit discrimination between luminance and chrominance properties made these colorspaces popular in the works on skin color segmentation. Several interesting properties of Hue are: it is invariant to highlights at white light sources, and also, for matte surfaces, to ambient light and surface orientation relative to the light source.

**CIElab** Skin color is not a physical property of an object, rather a perceptual phenomenon and therefore a subjective human concept. Therefore, color representation similar to the color sensitivity of human vision system should help to obtain high performance skin detection algorithm. CIELAB is perceptually uniform (reasonably perceptually unform, to be exact) and was proposed by G. Wyszecki and standardized by CIE (Commission Internationale de L'Eclairage). Perceptual uniformity means that a small perturbation to a component value is approximately equally perceptible across the range of that value. The wellknown RGB colorspace is far from being perceptually uniform, the non-linear transformation to CIELAB try to correct the situation. The price for better perceptual uniformity is complex transformation functions from and to RGB space, demanding far more computation than most other colorspaces

### 2.1.2 Histograms of Oriented Gradient

The HOG descriptor [23] is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions.

In practice this is implemented by dividing the image window into small spatial regions (cells), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. The combined histogram entries form the representation. For better invariance to illumination, shadowing, etc., it is also useful to contrast-normalize the local responses before using them. This can be done by accumulating a measure of local histogram energy over somewhat larger spatial regions (blocks) and using the results to normalize all of the cells in the block.

### 2.1.3 Gabor Filters

Gabor functions were proposed by Hungarian-born electrical engineer Dennis Gabor in 1946. Nowadays, Gabor functions are frequently used for feature extraction, especially in texture-based image analysis (e.g., classification, segmentation or edge detection) and more practically in face recognition. Many of image processing tasks can be seen in terms of a wavelet transform. Informally speaking, the image can be seen under the lens with a magnification given by the scale of a wavelet. In doing so, we can only see just the information that is determined by the shape of the used wavelet. The Gabor functions can also be seen in the words of a wavelet transform. Specifically, Gabor wavelets are created from one particular atom by dilation (and rotation in two-dimensional case). These Gabor wavelets provide a complete image representation. In a two-dimensional case, the absolute square of a correlation between an image and the two-dimensional Gabor function provides a local spectral energy density concentrated around a given position and frequency in a certain direction. A two-dimensional convolution with a circular (non-elliptical) Gabor function is separable to series of one-dimensional ones. J. G. Daugman discovered that simple cells in the visual cortex of mammalian brains can be modeled by Gabor functions. Thus, image analysis by the Gabor wavelets is similar to perception in the human visual system.

In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave:

$$g_{\lambda,\theta,\psi}(x,y) = \exp^{-((x'^2+\gamma^2 y'^2)/2\sigma^2)} \cos\left(2\pi\frac{x'}{\lambda}+\psi\right) \tag{2.1}$$

$$x' = x\cos\theta + y\sin\theta \quad y' = -x\sin\theta + y\cos\theta$$

where $\lambda$ is the amplitude of the filter, $\psi$ is the offset in pase, $\sigma$ is the variance of the gaussian component, $\gamma$ is the ratio of the ellipse, $\theta$ is its orientation.

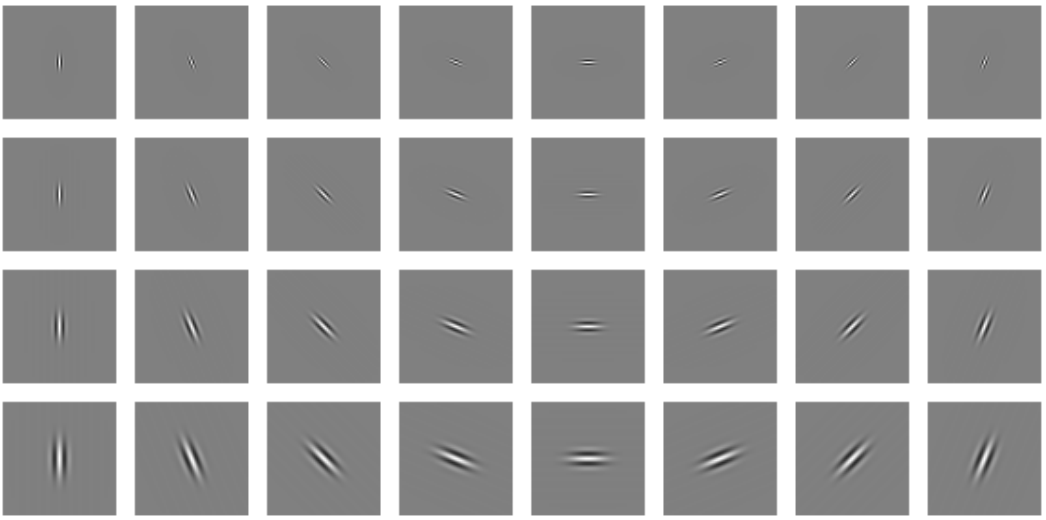


FIGURE 2.1: Examples of Gabor filters.

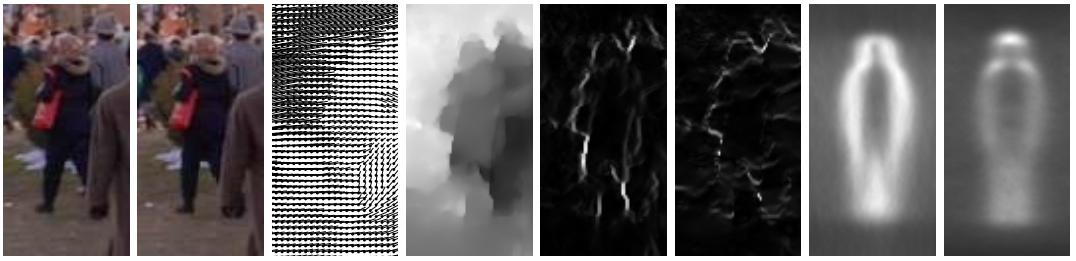
Simple cells in the visual cortex of mammalian brains can be modeled by Gabor functions [24]. Thus, image analysis with Gabor filters is thought to be similar to perception in the human visual system.

### 2.1.4 MBH

Motion Boundary Histograms (MBH) [23] are descriptors that use differential flow to cancel out most of the effects of camera motion and HOG like oriented histogram voting to obtain robust coding.

First note that the image flow induced by camera rotation (pan, tilt, roll) varies smoothly across the image irrespective of 3Ddepth boundaries, and in most applications it is locally essentially translational because significant camera roll is rare. Thus, any kind of local differential or difference of flow cancels out most of the effects of camera rotation. The remaining signal is due to either depth-induced motion parallax between the camera, subject and background, or to independent motion in the scene. Differentials of parallax flows are concentrated essentially at 3D depth boundaries, while those of independent motions are largest at motion boundaries. For human subjects, both types of boundaries coincide with limb and

body edges, so flow differentials are good cues for the outline of a person. However we also expect internal dynamics such as relative limb motions to be quite discriminant for human motions and differentials taken within the subject's silhouette are needed to capture these. Thus, flow-based features can focus either on coding motion (and hence depth) boundaries, or on coding internal dynamics and relative displacements of the limbs.



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

FIGURE 2.2: Illustration of the MBH descriptor. (a,b) Reference images at time t and t+1. (c,d) Computed optical flow, and flow magnitude showing motion boundaries. (e,f) Gradient magnitude of flow field $I_x$, $I_y$ for image pair (a,b). (g,h) Average MBH descriptor over all training images for flow field $I_x$, $I_y$.

Let $I_x$, $I_y$ denote images containing the $x$ (horizontal) and $y$ (vertical) components of optical flow, and $I_w = (I_x, I_y)$ denote the 2D flow image ($w = (x, y)$), and $I_x^x$, $I_x^y$, $I_y^x$, $I_y^y$ denote the corresponding $x$- and $y$-derivative differential flow images. For example, $I_x^y$ is the $y$-derivative of the $x$ component of optical flow.

It is natural to try to capture the local orientations of motion edges by emulating the static-image HOG descriptors. The simplest approach is to treat the two flow components $I_x$, $I_y$ as independent images, take their local gradients separately, find the corresponding gradient magnitudes and orientations, and use these as weighted votes into local orientation histograms in the same way as for the standard gray scale HOG. This is the Motion Boundary Histograms (MBH) descriptor. A separate histogram can be built for each flow component, or the two channels can be combined, e.g. by the winner-takes-all voting method. However, [23] find that separate histograms are more discriminant. As with standard gray scale HOG, it is best to take spatial derivatives at the smallest possible scale ($[1, 0, -1]$ mask) without any form of smoothing.

## 2.2 Bag of Words

The bag-of-words (BoW) methodology is a way to encode a variable number of descriptors in a fixed sized representation. It was first proposed in the text retrieval domain problem for text document analysis, and it was further adapted for computer vision applications [25]. For image analysis, a visual analogue of a word is used in the BoWmodel, which is based on the vector quantization process by clustering low-level visual features of local regions.

To extract the BoW feature from images involves the following steps:

1. automatically detect regions/points of interest,
2. compute local descriptors over those regions/points,
3. quantize the descriptors into words to form the visual vocabulary
4. find the occurrences in the image of each specific word in the vocabulary for constructing the BoW feature (or a histogram of word frequencies).

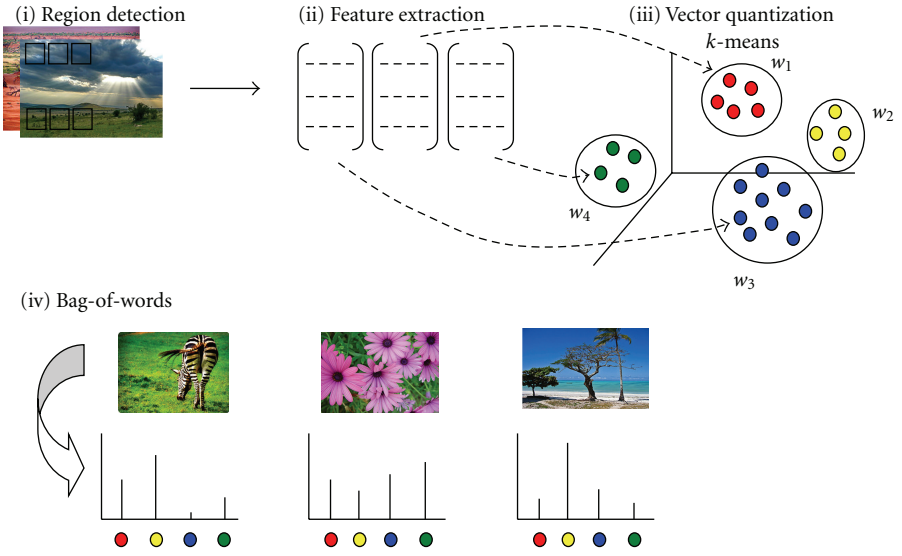



FIGURE 2.3: The Bag of Words approach.

Figure 2.3 describes these four steps to extract the BoW feature from images. The BoW model can be defined as follows. Given a training dataset $D$ containing $n$ images represented by $D = d_1, d_2, ..., d_n$, where $d$ is the extracted visual features, a specific unsupervised learning algorithm, such as k-means, is used to group $D$ based on a fixed number of visual words

$W$ (or categories) represented by $W = w_1, w_2, ..., w_K$, where $K$ is the number of clusters. Then, we can summarize the data in a $K \times N$ cooccurrence table of counts $N_{ij} = n(w_i, d_j)$, where $n(w_i, d_j)$ denotes how often the word $w_i$ occurred in an image $d_i$. The $i$-th column of this table can be used as a global descriptor for the $i$-th image: thanks to the clustering phase a fixed-size descriptor has been obtained from a variable number of points of interests and thus from a variable number of local descriptors.

As stated before, the traditional application of the Bag of Words approach is image classification. Usually SIFT keypoints and descriptors are extracted, and BoW histograms are then passed to a SVM classifier. However, the same principle can be used to classify videos or frame sequences, if we employ spatio-temporal points of interest and descriptors: in the following of this thesis, the BoW technique will be used in combination with spatio-temporal trajectories extracted from frame sequences, each described in its shape and appearance. We will then be able to classify frame sequences instead of images, and to have a descriptor with fixed length even in presence of a variabile number of trajectories per frame. Other techinques will be then proposed to deal with actions with variable length (and thus described by a variable number of frames).

## 2.3 Classifiers

The last step in the traditional computer vision pipeline is classification: in this thesis, we will employ the Support Vector Machine classifier, both in its linear version and in its structured version, and Random Forest classifiers. Support Vector Machines (SVM) have been one of the most popular classifiers in recent years for solving problems in classification, regression, and novelty detection. An important property of support vector machines is that the determination of the model parameters corresponds to a convex optimization problem, and so any local solution is also a global optimum.

### 2.3.1 Linear SVMs

Let's start with the simplest case: linear support vector machines trained on separable data. Label the training data $\{\mathbf{x}_i, y_i\}$ $i = 1, ..., l$, $y_i \in \{-1, 1\}$, $\mathbf{x}_i \in \mathbb{R}^d$. Suppose we have some hyperplane which separates the positive from the negative samples. The points $\mathbf{x}$ which lie on the hyperplane satisfy $\mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{w}$ is normal to the hyperplane, $|b|/\|\mathbf{w}\|$ is

the perpendicular distance from the hyperplane to the origin and $\|\mathbf{w}\|$ is the Euclidean norm of $\mathbf{w}$. Let $\mathbf{w} \cdot \mathbf{x} + b = a$ ($\mathbf{w} \cdot \mathbf{x} + b = -a$) be the hyperplane that touches the closest positive (negative) example. Define the *margin* of a separable hyperplane to be $2a/\|\mathbf{w}\|$. For the linearly separable case, the support vector algorithm simply looks for the separating hyperplane with the largest margin. This can be formulated as follows: suppose that all the training data satisfy the following constraints:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \quad \text{if } y_i = +1 \tag{2.2}$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \quad \text{if } y_i = -1 \tag{2.3}$$

We will now switch to a Lagrangian formulation of the problem. There are two reasons for doing this. The first is that the constraints will be replaced by constraints on the Lagrange multipliers themselves, which will be much easier to handle. The second is that in this reformulation of the problem, the training data will only appear (in the actual training and test algorithms) in the form of dot products between vectors. This is a crucial property which will allow us to generalize the procedure to the nonlinear case.

Thus, we introduce positive Lagrange multipliers $\alpha_i$, $i = 1, ..., l$, one for each of the inequality constraints (2.10). For constraints of the form $c_i \geq 0$, as in our case, the constraint equations are multiplied by positive Lagrange multipliers and subtracted from the objective function, to form the Lagrangian. For equality constraints, the Lagrange multipliers are unconstrained. This gives Lagrangian:

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i \tag{2.4}$$

We must now minimize $L_p$ with respect to $\mathbf{w}$, $b$, and simultaneously require that the derivatives of $L_p$ with respect to all the $\alpha_i$ vanish, all subject to the constraints $\alpha_i \geq 0$. Now this is a convex quadratic programming problem, and those points which satisfy the constraints also form a convex set. Requiring that the gradient of $L_p$ with respect to $\mathbf{w}$ and $b$ vanish give the conditions:

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Longleftrightarrow \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \tag{2.5}$$

$$\frac{\partial L_p}{\partial b} = 0 \Longleftrightarrow \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.6}$$

We can substitute these equalities into (2.4) to give:

$$L_d = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{2.7}$$

Support vector training (for the separable, linear case) therefore amounts to maximizing $L_d$ with respect to the $\alpha_i$, subject to constraints (2.6) and positivity of the $\alpha_i$, with solution given by (2.5). Notice that there is a Lagrange multiplier $\alpha_i$ for every training point. In the solution, those points for which $\alpha_i > 0$ are called *support vectors*.

Since not always the training data is linearly separable, we would like to relax the constraints (2.2) and (2.3), but only when necessary, that is, we would like to introduce a further cost for doing so. This can be done by introducing positive slack variables $\epsilon_i$, $i = 1, ..., l$ in the constraints [26], which then become:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \quad \text{if } y_i = +1 - \epsilon_i \tag{2.8}$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \quad \text{if } y_i = -1 + \epsilon_i \tag{2.9}$$

Thus, for an error to occur, the corresponding $\epsilon_i$ must exceed unity. Combining the two inequalities, and adding an extra cost for errors to the objective function, we can formulate the problem of margin maximization as:

$$\min_{\mathbf{w},b} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{l} \epsilon_i \quad \text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \epsilon_i \geq 0 \tag{2.10}$$

where $C$ is a parameter to be chosen by the user, a larger $C$ corresponding to assigning a higher penalty to error.

### 2.3.2 Structured Learning and Structured SVM

Structured learning deals with the general problem of learning a mapping from input vectors or patterns $\mathbf{x} \in \mathcal{X}$ to discrete response variables $\mathbf{y} \in \mathcal{Y}$, based on a training sample of

input-output pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{X} \times \mathcal{Y}$ drawn from some fixed but unknown probability distribution. Unlike multiclass classification, where the output space consists of an arbitrary finite set of labels or class identifiers, structured classification considers the case where the elements of $\mathcal{Y}$ are structured objects such as sequences, strings, trees, images or graphs.

The objective of a structured classifier is thus to learn functions $f : \mathcal{X} \to \mathcal{Y}$ between the input space and arbitrary discrete output spaces, based on a training sample of input-output pairs, and the approach we pursue is to learn a discriminant function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ over input-output pairs from which we can derive a prediction by maximizing $F$ over the response variable for a specific given input $\mathbf{x}$. Hence, the general form of our hypotheses $f$ is

$$f(\mathbf{x}, \mathbf{y}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}) \tag{2.11}$$

where $\mathbf{w}$ denotes a parameter vector. $F$ can be thought as a compatibility function that measures how compatible pairs $(\mathbf{x}, \mathbf{y})$ are. We can assume $F$ to be linear in some combined feature representation of input and outputs $\Phi(\mathbf{x}, \mathbf{y})$:

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle \tag{2.12}$$

where the specific form of $\Phi(\mathbf{x}, \mathbf{y})$ depends on the nature of the problem.

We then define a loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, where $\Delta(\mathbf{y}, \mathbf{y}')$ quantifies the loss associated with a prediction $\mathbf{y}'$ if the true output value is $\mathbf{y}$. If we assume that input-output pairs $(\mathbf{x}, \mathbf{y})$ are generated according to some fixed distribution $P(\mathbf{x}, \mathbf{y})$, we could describe the goal of supervised learning as to find a function $f$ that minimizes the following risk

$$\mathcal{R}_P^{\Delta}(f) = \int_{\mathcal{X} \times \mathcal{Y}} \Delta(\mathbf{y}, f(\mathbf{x})) dP(\mathbf{x}, \mathbf{y}) \tag{2.13}$$

but since $P$ is unknown, and we only have a finite training set of pairs $S = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y} : i = 1, ..., n\}$, we can describe the performance of a function $f$ on the training set $S$ by the empirical risk,

$$\mathcal{R}_S^{\Delta}(f) = \frac{1}{n}\sum_{i=1}^{n}\Delta(\mathbf{y}_i, f(\mathbf{x_i})) \tag{2.14}$$

First, we consider the separable case in which there exists a function $f$ parameterized by $\mathbf{w}$ such that the empirical risk is zero. If we assume that $\Delta(\mathbf{y}, \mathbf{y}') > 0$ for $\mathbf{y} \neq \mathbf{y}'$ and $\Delta(\mathbf{y}, \mathbf{y}) = 0$, then the condition of zero training error can then be compactly written as a set of non-linear constraints

$$\forall i : \max_{\mathbf{y}\in\mathcal{Y}-\mathbf{y}_i}\{\langle \mathbf{w}, \mathbf{\Psi}(\mathbf{x}_i, \mathbf{y})\rangle\} < \langle \mathbf{w}, \mathbf{\Psi}(\mathbf{x}_i, \mathbf{y}_i\rangle \tag{2.15}$$

Each nonlinear inequalities in 2.15 can be equivalently replaced by $|\mathcal{Y}|-1$ linear inequalities, resulting in a total of $n|\mathcal{Y}| - n$ linear constraints,

$$\forall i, \forall \mathbf{y} \in \mathcal{Y} - \mathbf{y}_i : \langle \mathbf{w}, \delta\mathbf{\Psi}_i(\mathbf{y})\rangle > 0 \tag{2.16}$$

where we have defined the shorthand $\delta\mathbf{\Psi}_i(\mathbf{y}) = \mathbf{\Psi}(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{\Psi}(\mathbf{x}_i, \mathbf{y})$.

If the set of inequalities in 2.16 is feasible, there will typically be more than one solution $\mathbf{w}^*$. To specify a unique solution, we propose to select $\mathbf{w}$ with $||\mathbf{w}|| \leq 1$ for which the score of the correct label $\mathbf{y}_i$ is uniformly most different from the closest runnerup $\hat{\mathbf{y}}_i(\mathbf{w}) = \mathrm{argmax}_{\mathbf{y}\neq\mathbf{y}_i}\langle \mathbf{w}, \mathbf{\Psi}(\mathbf{x}_i, \mathbf{y})\rangle$. This generalizes the maximum-margin principle employed in SVMs to the more general case of structured learning. The resulting hard-margin optimization problem is:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t.} \forall i, \forall \mathbf{y} \in \mathcal{Y} - \{\mathbf{y}_i\} : \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}) \geq 1 \tag{2.17}$$

To allow errors in the training set slack variable are introduced and a soft-margin criterion is used. We introduce one slack variable for every non-linear constraint, which will result in an upper bound on the empirical risk and offers some additional algorithmic advantages. Adding a penalty term that is linear in the slack variables to the objective results in the

quadratic program

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.}\forall i, \forall \xi_i \geq 0, \forall \mathbf{y} \in \mathcal{Y} - \{\mathbf{y}_i\} : \langle \mathbf{w}, \delta\Psi_i(\mathbf{y})\rangle \geq 1 - \xi_i \tag{2.18}$$

However, also slack re-scaling can be used. Furthermore mn-slack and 1-slack versions have been proposed.

### 2.3.3 Label Sequence Learning

A special case of the general scenario of Structured learning is the problem of label sequence learning, or sequence annotation. Here the goal is to predict a label sequence $\mathbf{y} = (y^1, ..., y^T)$ for a given observation sequence $\mathbf{x} = (\mathbf{x}^1, ..., \mathbf{x}^T)$. If we assume that all sequences are of the same length $T$ and that $\mathbf{\Sigma}$ is the set of possible labels for each individual variable $y^t$, hence each sequence of labels is considered to a class of its own, resulting in a multiclass classification problem with $|\mathbf{\Sigma}|^T$ different classes. To model label sequence learning in this manner would of course not be very useful, if one were to apply standard multiclass classification methods.

Inspired by hidden Markov models (HMM) interactions, Altun *et al.* [27] propose to define $\Psi$ to include interactions between input features and labels via multiple copies of the input features as well as features that model interactions between nearby label variables. Before modeling the discriminant function $F$, we need to define the canonical binary representation of labels $\mathbf{y} \in \mathcal{Y}$ by unit vectors

$$\Lambda^c(\mathbf{y}) \equiv (\delta(\mathbf{y}_1, \mathbf{y}), \delta(\mathbf{y}_2, \mathbf{y}), ..., \delta(\mathbf{y}_K, \mathbf{y}))' \in \{0, 1\}^K \tag{2.19}$$

so that $\langle \Lambda^c(\mathbf{y}), \Lambda^c(\mathbf{y}')\rangle = \delta(\mathbf{y}, \mathbf{y}')$. Furthermore, we define the $\otimes$-operation in the following way:

$$\otimes : \mathbb{R}^D \times \mathbb{R}^K \to \mathbb{R}^{D \cdot K},\ (\mathbf{a} \otimes \mathbf{b})_{i+(j-1)D} \equiv a_i \cdot b_j \tag{2.20}$$

Now we can write the discriminant function used in [28]:

$$F(\mathbf{x},\mathbf{y};\mathbf{w}) = \begin{array}{l} \langle \mathbf{w}_a, \sum_{t=1}^{T} \Phi(\mathbf{x}^t) \otimes \Lambda^c(y^t) \rangle \\ +\eta \langle \mathbf{w}_b, \sum_{t=1}^{T-1} \Lambda^c(y^t) \otimes \Lambda^c(y^{t+1}) \rangle \end{array} \tag{2.21}$$

where $\mathbf{w} = (\mathbf{w}'_a, \mathbf{w}'_b)'$ and $\eta \geq 0$ is a scaling factor which balances the two types of contribution. Of course, in this case $\Psi(\mathbf{x},\mathbf{y})$ is

$$\Psi(\mathbf{x},\mathbf{y}) = \begin{pmatrix} \sum_{t=1}^{T} \Phi(\mathbf{x}^t) \otimes \Lambda^c(y^t) \\ \eta \sum_{t=1}^{T-1} \Lambda^c(y^t) \otimes \Lambda^c(y^{t+1}) \end{pmatrix} \tag{2.22}$$

There are at least three ways of extending the given discriminant function. First of all, one can extract features not just from $\mathbf{x}^t$, but from a window around $\mathbf{x}^t$, i.e. replacing $\Phi(\mathbf{x}^t)$ with $\Phi(\mathbf{x}^{t-r}, ..., \mathbf{x}^t, ..., \mathbf{x}^{t+r})$. This has been called the use of *overlapping features* [27]. Secondly, it is also straightforward to include higher order label-label interactions beyond pairwise interactions by including higher order tensor terms, for instance, label triplets $\sum_t \Lambda^c(y^t) \otimes \Lambda^c(y^{t+1}) \otimes \Lambda^c(y^{t+2})$. Thirdly, one can also combine higher order $\mathbf{y}$ features with input features, for example by including terms of the type $\sum_t \Phi(\mathbf{x}^t) \otimes \Lambda^c(y^t) \otimes \Lambda^c(y^{t+1})$.

### 2.3.4 Random Forest

In the last years, the random forest framework has become a very popular and powerful tool for classification and regression problems by exhibiting many appealing properties like inherent multi-class capability, robustness to label noise and reduced tendency to over fitting. They are considered to be close to an ideal learner, making them attractive in many areas of computer vision like image classification, clustering, regression or semantic segmentation. In this work we use the decision forest algorithm to learn hand appearance.

A (binary) *decision tree* is a tree-structured predictor where, starting from the root, a sample is routed until it reaches a leaf where the prediction takes place. At each internal node of the tree the decision is taken whether the sample should be forwarded to the left or right child, according to a binary-valued function. In formal terms, let $X$ denote the input space, let $Y$ denote the output space and let $T$ be the set of the decision trees. In its simplest form a decision tree consists of a single node (a leaf) and is parametrized by a probability distribution $Q \in \mathcal{P}(Y)$ which represents the posterior probability of elements in $Y$ given

any data sample reaching the leaf. We denote this tree as $\text{Lf}(Q) \in T$. Otherwise, a decision tree consists of a node with a left and a right sub-tree. This node is parametrized by a split function $\Phi : X \to \{0, 1\}$, which determines whether to route a data sample $x \in X$ reaching it to the left decision sub-tree $t_l \in T$ (if $\Phi(x) = 0$) or to the right one $t_r \in T$ (if $\Phi(x) = 1$). We denote such a tree as $\text{Nd}(\Phi, t_l, t_r) \in T$. Finally, a decision forest is an ensemble $\mathcal{F} \subseteq T$ of decision trees which makes a prediction about a data sample by averaging over the single predictions gathered from all trees.

Given a decision tree $t \in T$, the associated posterior probability of each element in $Y$ given a sample $x \in X$ is determined by finding the probability distribution $Q$ parametrizing the leaf that is reached by $x$ when routed along the tree. This is compactly presented with the following definition of $P(y|x, t)$, which is inductive in the structure of $t$:

$$P(y|x,t) = \begin{cases} Q(y) & \text{if } t = \text{Lf}(Q) \\ P(y|x,t_l) & \text{if } t = \text{Nd}(\Phi, t_l, t_r) \text{ and } \Phi(x) = 0 \\ P(y|x,t_r) & \text{if } t = \text{Nd}(\Phi, t_l, t_r) \text{ and } \Phi(x) = 1 \end{cases}$$

Finally, the combination of the posterior probabilities derived from the trees in a forest $\mathcal{F} \subseteq T$ can be done by an averaging operation, yielding a single posterior probability for the whole forest:

$$P(y|x, \mathcal{F}) = \frac{1}{|\mathcal{F}|} \sum_{t \in \mathcal{F}} P(y|x, t)$$

# Chapter 3

# Hand segmentation in ego-centric videos

We now focus on the task of pixel-wise hand detection from video recorded with a wearable head-mounted camera. In contrast to a third-person point-of-view camera, such as a mounted surveillance camera or a TV camera, a first-person point-of-view wearable camera has exclusive access to first-person activities and is an ideal viewing perspective for analyzing fine motor skills such as hand-object manipulation or hand-eye coordination. As we said before, the use of ego-centric video has re-emerged as a popular topic in computer vision and has shown promising results in such areas as understanding hand-eye coordination and recognizing activities of daily living. In order to achieve more detailed models of human interaction and object manipulation, and, in our case, to perform hand gesture recognition, it is important to detect hand regions with pixel-level accuracy. Hand detection is, in fact, an important element of such tasks as gesture recognition, hand tracking, grasp recognition, action recognition and understanding hand-object interactions.

The egocentric paradigm presents a new set of constraints and characteristics that introduce new challenges as well as unique properties that can be exploited for the task of first-person hand detection. Unlike static third-person point-of-view cameras typically used for gesture recognition or sign language analysis, the video acquired by a first-person camera undergoes large ego-motion because it is worn by the user. The mobile nature of the camera also results in images recorded over extreme transitions in lighting, such as walking from indoors to outdoors. As a result, the large image displacement caused by body motion makes it very

difficult to apply traditional image stabilization or background subtraction techniques. Similarly, large changes in illumination conditions induce large fluctuations in the appearance of hands. Fortunately, ego-centric videos also have the property of being user-specific, where images of hands and the physical world are always acquired with the same camera for the same user. This implies that the intrinsic colour of the hands does not change drastically over time.

The purpose of this chapter is to identify and address the challenges of hand detection for first-person vision. To this end, we present existing and recent hand detection approaches, and we propose a novel algorithm capable of facing various illumination conditions and different backgrounds. Furthermore, we analyze our proposal on existing datasets to highlight the pros and cons of various widely-used local appearance features. We evaluate the value of modeling global illumination to generate an ensemble of hand region detectors conditioned on the illumination conditions of the scene. Based on our finding, we propose a model using sparse feature selection and an illumination-dependent modeling strategy, and show that it out-performs several baseline approaches.

## 3.1 Previous approaches in Hand Segmentation

The problem of hand detection in ego-vision scenario, has been addressed only recently by the research community. Khan *et al.* 's paper [29], for example, addresses skin detection as a general task for face detection and hand gesture analysis. They used the IHLS colour space for raw pixel based skin detection and evaluated different classifiers: Random Forest, Bayesian Networks, Multilayers Perceptrons, AdaBoost, Naive Bayes, RBF Networks. They demonstrated on a database of 8991 images that Random Forest classification obtains the highest F-score among all the other techniques. Also, they show the effect of increasing the number of trees: the maximum F-score is achieved with 10 trees, while an higher number of tree doesn't increase the F-score.

Jones *et al.* [30] describe the construction of colour models for skin and non-skin classes from a dataset of nearly 1 billion labelled pixels. These classes exhibit a surprising degree of separability which we exploit by building a skin pixel detector achieving a detection rate of 80% with 8.5% false positives. They also compare the performance of histogram and mixture models in skin detection and find histogram models to be superior in accuracy and computational cost. Using aggregate features computed from the skin pixel detector they

build an effective detector for naked people. Their results suggest that colour can be the most powerful cue for detecting people in unconstrained imagery.

On a different note, Hayman *et al.* [31] propose a video stabilization approach for hand segmentation. They present a probabilistic framework for when the camera pans and tilts. A unified approach is developed for handling various sources of error, including motion blur, sub-pixel camera motion, mixed pixels at object boundaries, and also uncertainty in background stabilisation caused by noise, unmodelled radial distortion and small translations of the camera. The second contribution regards a Bayesian approach to specifically incorporate uncertainty concerning whether the background has yet been uncovered by moving foreground objects. They don't assume that a background model is available in advance, instead they generate the background model online, very possibly in the presence of moving objects.

Fathi *et al.* [32] try to solve the hand segmentation problem in the ego-centric perspective, with a foreground segmentation approach. Their method is based on a few assumptions: (1) that the background is static in the world coordinate frame, (2) they define foreground as every entity which is moving with respect to the static background, (3) they assume background objects are usually farther away from the camera than foreground objects and (4) they assume they can build a panorama of the background scene by stitching the background images together using an affine transformation. The fourth assumption is basically assuming that the background is roughly on a plane or far enough from the camera. An object will be moving with respect to the background when it is being manipulated by hands. When the subject finishes a sub-task and stops manipulating the object, the object will become a part of background again.

Their segmentation method is as follows. They first make an initial estimate of background regions in each image by fitting a fundamental matrix to dense optical flow vectors. They make temporally local panoramas of background given their initial background region estimates. Then they register each image into its local background panorama. The regions in the image which do not match the background scene are likely to be parts of foreground. Then they connect the regions in sequence of images spatially and temporally and use graphcut to split them into foreground and background segments. They split the video into short intervals and make local background models for each. The reason is that the background might change over time, for example the subject might finish manipulating an object and leave it on the table, letting it become a part of background. They initially approximately separate foreground and background channels for each image by fitting a fundamental matrix to its optical

flow vectors. They compute the flow vectors to its few adjacent frames. For each interval we choose a reference frame whose initial background aligns the best to other frames.

They build two kinds of temporally local models for background (panoramas): (1) a model based on colour and texture histogram of regions and (2) a model of region boundaries. To build these models, they fit an affine transformation to the initial background SIFT feature correspondences of each frame in the interval, and the reference frame. They stitch these images using affine transformation. After fixing the images to the reference frame coordinate, they build the colour-texture and boundary background models. This is by computing a histogram of values extracted from interval images corresponding to each location in the background panorama.

On the other hand, Li and Kitani in [33] provide an historical overview about approaches for detecting hands from moving cameras. They define three categories: local appearance-based detection, global appearance-based detection, where a global template of hand is needed, and motion-based detection, which is based on the hypothesis that hands and background have different motion statistics. Motion-based detection approach requires no supervision nor training. This approach eventually identifies as hand an object manipulated by the user, since it moves together his hands. In addition they proposed a model with sparse feature selection which was shown to be an illumination-dependent strategy. To solve this issue, they trained a set of random forests indexed by a global colour histogram, each one reflecting a different illumination condition. Recently Bagdanov *et al.* [34] propose a method to predict the status of the user hand by jointly exploiting depth and RGB imagery.

All the presented previous works present good characteristics, but lack of generality, since they take into account only few aspects to model user hand appearance and they are not integrated with a gesture recognition system. We therefore present a novel method for hand segmentation that can be used as basis for ego-vision applications. The proposed Hand detection approach is based on Random Forest classifiers learned by colour and gradient features which are computed on superpixels. In order to improve the detection accuracy we present two strategies that incorporate temporal and spatial coherence: temporal smoothing and spatial consistency.

## 3.2 Proposed approach

Ego-vision applications require a fast and reliable segmentation of the hands; thus we propose to use random forest classifiers, as they are known to efficiently work even with large inputs [35]. Since using a per-pixel basis in label assignment has show to be inefficient [30], we adopt segmentation method which assign labels to superpixels, as suggested in [36]. This allows a complexity reduction of the problem and also gives better spatial support for aggregating features that could belong to the same object. Summarizing, our method extract superpixels from the input image, and then classifies each superpixel as belonging to the hand or not.

To extract superpixels for every frames we use the Simple Linear Iterative Clustering (SLIC) algorithm, proposed in [1] as it is memory efficient and highly accurate segmentation method. Moreover, being the SLIC algorithm an adaptation of $k$-means for superpixels, the generated superpixels tend to be circular (see Fig. 3.1). It features two main characteristics:

1. The number of distance calculations in the optimization is dramatically reduced by limiting the search space to a region proportional to the superpixel size. This reduces the complexity to be linear in the number of pixels $N$– and independent of the number of superpixels $k$.

2. A weighted distance measure combines colour and spatial proximity, while simultaneously providing control over the size and compactness of the superpixels.

SLIC is simple to use and understand. By default, the only parameter of the algorithm is $k$, the desired number of approximately equally-sized superpixels. For colour images in the CIELAB colour space, the clustering procedure begins with an initialization step where $k$ initial cluster centers $C_i = [l_i \quad a_i \quad b_i \quad x_i \quad y_i]^T$ are sampled on a regular grid spaced $S$ pixels apart. To produce roughly equally sized superpixels, the grid interval is $S = \sqrt{N/k}$. The centers are moved to seed locations corresponding to the lowest gradient position in a $3 \times 3$ neighborhood. This is done to avoid centering a superpixel on an edge, and to reduce the chance of seeding a superpixel with a noisy pixel. Next, in the assignment step, each pixel $i$ is associated with the nearest cluster center whose search region overlaps its location. This is the key to speeding up the algorithm because limiting the size of the search region significantly reduces the number of distance calculations, and results in a significant speed advantage over conventional $k$-means clustering where each pixel must be compared with all

FIGURE 3.1: Images segmented using SLIC into superpixels of size 64, 256, and 1024 pixels (approximately).

cluster centers. Since the expected spatial extent of a superpixel is a region of approximate size $S \times S$, the search for similar pixels is done in a region $2S \times 2S$ around the superpixel center.

Once each pixel has been associated to the nearest cluster center, an update step adjusts the cluster centers to be the mean $[l \quad a \quad b \quad x \quad y]^T$ vector of all the pixels belonging to the cluster. The $L_2$ norm is used to compute a residual error between the new cluster center locations and previous cluster center locations. The assignment and update steps can be repeated iteratively until the error converges. Finally, a post-processing step enforces connectivity by re-assigning disjoint pixels to nearby superpixels. A real-time implementation of the SLIC algorithm is also available in [37].

Having extracted SLIC superpixels, we represent superpixels by features to encode colour and gradient information. As pointed out by previous works, HSV and LAB colour spaces have been proven to be robust for skin detection. In particular, we describe each superpixel with:

- mean and covariance matrix of its pixel values, both in HSV and Lab colour spaces
- a 32-bin colour histogram, again both in HSV and Lab colour spaces
- Gabor features (see 2.1.3) obtained with 27 filters (nine orientations and three different scales: $7 \times 7$, $13 \times 13$, $19 \times 19$)

- a simple histogram of gradients with nine bins, computed inside each superpixel (this could be interpreted as a superpixel version of the HOG descriptor).

The gradient features are included in order to discriminate between objects with a hand-like colour distribution and hands. Beyond describing each superpixel with colour and texture, we propose three approaches to enhance the performance of our method: (1) *illumination invariance*, by which we make our method invariant to variable illumination conditions; (2) temporal smoothing, by which we impose the temporal coherence of the segmentation; (3) spatial consistency, that is, we require that segmentation masks are spatially consistent. In the next subsections we will describe each of these techniques in detail.

### 3.2.1 Illumination invariance

The main purpose of this technique is to deal with different illumination conditions. We cluster the training images running the $k$-means algorithm on a global HSV histogram. Hence, we train a Random Forest classifier for each cluster. By using a histogram over all three channels of the HSV colour space, each scene cluster encodes both the appearance of the scene and its illumination. Intuitively, this models the fact that hands viewed under similar global appearance will share a similar distribution in the feature space.

Given a feature vector $\mathbf{l}$ of a superpixel $\mathbf{s}$ and a global appearance feature $\mathbf{g}$ (the HSV histogram), the posterior distribution of $\mathbf{s}$ is computed by marginalizing over different clusters $c$:

$$P(\mathbf{s}|\mathbf{l},\mathbf{g}) = \sum_{c=1}^{k} P(\mathbf{s}|\mathbf{l},c)P(c|\mathbf{g}) \tag{3.1}$$

where $k$ is the number of clusters, $P(\mathbf{s}|\mathbf{l},c)$ is the output of the cluster-specific classifier and $P(c|\mathbf{g})$ is a conditional distribution of a cluster $c$ given a global appearance feature $\mathbf{g}$. In test phase, the conditional $P(c|\mathbf{g})$ is approximated using an uniform distribution over the five nearest clusters.

It is important to highlight that the optimal number of classifiers depends on the characteristics of the dataset: a training dataset with several different illumination conditions, taken both inside and outside, will need an higher number of classifiers than one taken indoor. In


FIGURE 3.2: Comparison before (left image) and after (right image) Temporal smoothing.

addition, we model the hand appearance not only considering illumination variations, but also including semantic coherence in time and space.

### 3.2.2 Temporal smoothing

Temporal smoothing aims to improve the foreground prediction of a pixel in a frame by a weighted combination of its previous frames, since past frames should affect the results prediction for the current frame. This technique is inspired by [38].

The smoothing filter for a pixel $\mathbf{x}_t^i$ of a frame $t$ can thus be defined as follows:

$$P(\mathbf{x}_t^i = 1) = \sum_{k=0}^{\min(d,k)} w_k \big( P(\mathbf{x}_t^i = 1 | \mathbf{x}_{t-k}^i = 1) \cdot P(\mathbf{x}_{t-k}^i = 1 | \mathbf{l_{t-k}}, \mathbf{g_{t-k}}) + P(\mathbf{x}_t^i = 1 | \mathbf{x}_{t-k}^i = 0) \cdot \cdot P(\mathbf{x}_{t-k}^i = 0 | \mathbf{l_{t-k}}, \mathbf{g_{t-k}}) \big) \quad (3.2)$$

where $P(\mathbf{x}_{t-k}^i = 1 | \mathbf{l_{t-k}}, \mathbf{g_{t-k}})$ is the posterior probability that a pixel in frame $t-k$ is marked as hand part and $d$ is a number of past frames used. This likelihood can be defined as the probability $P(\mathbf{s} | \mathbf{l_{t-k}}, \mathbf{g_{t-k}})$, being $\mathbf{x}_t^i$ part of $\mathbf{s}$. In the same way, $P(\mathbf{x}_{t-k}^i = 0 | \mathbf{l_{t-k}}, \mathbf{g_{t-k}})$ is defined as the probability $1 - P(\mathbf{s} | \mathbf{l}, \mathbf{g_{t-k}})$.

While $P(\mathbf{x}_t^i = 1 | \mathbf{x}_{t-k}^i = 1)$ and $P(\mathbf{x}_t^i = 1 | \mathbf{x}_{t-k}^i = 0)$ are prior probabilities estimated from the training set as follows:

$$P(\mathbf{x}_t^i = 1 | \mathbf{x}_{t-k}^i = 1) = \frac{\#(\mathbf{x}_t^i = 1, \mathbf{x}_{t-k}^i = 1)}{\#(\mathbf{x}_{t-k}^i = 1)} \quad (3.3)$$

FIGURE 3.3: Comparison before (left image) and after (right image) Spatial Consistency.

$$P(\mathbf{x}_t^i = 1 | \mathbf{x}_{t-k}^i = 0) = \frac{\#(\mathbf{x}_t^i = 1, \mathbf{x}_{t-k}^i = 0)}{\#(\mathbf{x}_{t-k}^i = 0)} \tag{3.4}$$

where $\#(\mathbf{x}_{t-k}^i = 1)$ and $\#(\mathbf{x}_{t-k}^i = 0)$ are the number of times in which $\mathbf{x}_{t-k}$ belongs or not to a hand region, respectively; $\#(\mathbf{x}_t^i = 1, \mathbf{x}_{t-k}^i = 1)$ is the number of times that two pixels at the same location at frame $t$ and $t-k$ belong to a hand part; similarly, $\#(\mathbf{x}_t^i = 1, \mathbf{x}_{t-k}^i = 0)$ is the number of times that a pixel in frame $t$ belongs to a hand part and pixel in the same position in frame $t-k$ does not belong to a hand region.

Based on our preliminary experiments we set $d$ equal to three. Figure 3.2 shows an example where temporal smoothing deletes blinking regions (i.e. the tea box brand and jar shadows on the right).

### 3.2.3 Spatial consistency

Given pixels elaborated by the previous steps, we want to exploit spatial consistency to prune away small and isolated pixel groups that are unlikely to be part of hand regions and also aggregate bigger connected pixel groups. To this aim, we exploit the GrabCut algorithm [2], which essentially works as follows:

1. It takes as input an initial trimap by of background, foreground and unknown pixels.
2. It creates an initial image segmentation, where all unknown pixels are tentatively placed in the foreground class and all known background pixels are placed in the background class.
3. Gaussian Mixture Models (GMMs) are created for initial foreground and background classes.

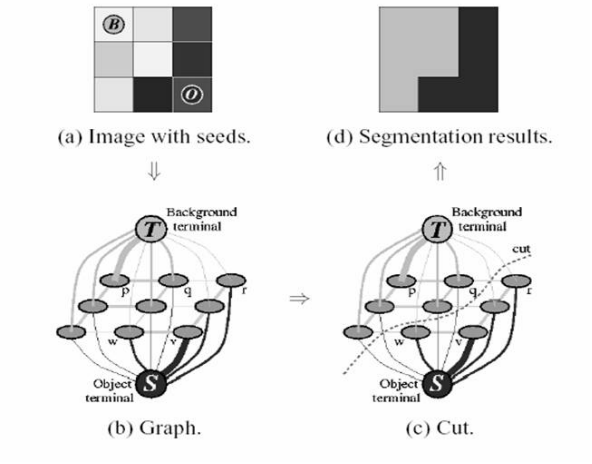



FIGURE 3.4: The GrabCut algorithm.

4. Each pixel in the foreground class is assigned to the most likely Gaussian component in the foreground GMM. Similarly, each pixel in the background is assigned to the most likely background Gaussian component.

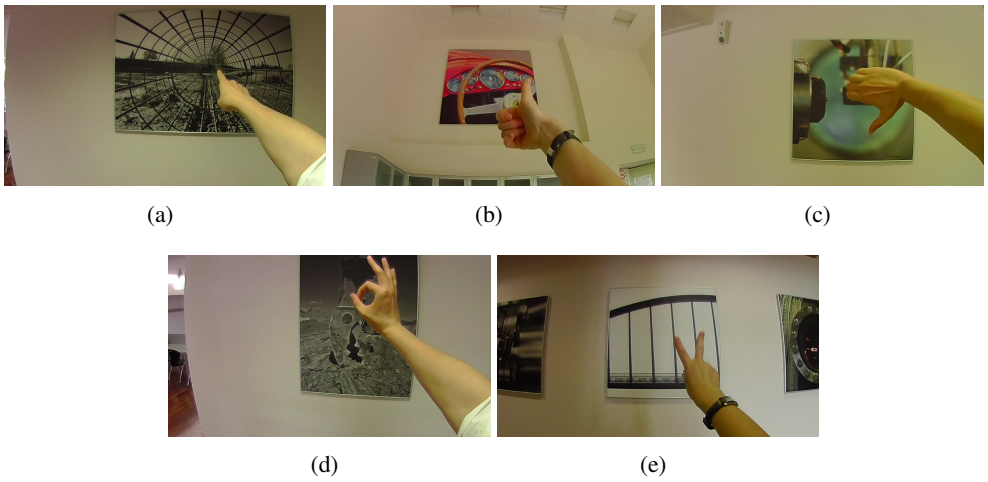
5. The GMMs are thrown away and new GMMs are learned from the pixel sets created in the previous set.

6. A graph is built and Graph Cut is run to find a new tentative foreground and background classification of pixels (Fig. 3.4).

7. Steps 4-6 are repeated until the classification converges

For every pixel $\mathbf{x}$, we extract its posterior probability $P(\mathbf{x}_t^i)$ and use it as input for the GrabCut algorithm. Each pixel with $P(\mathbf{x}_t^i) \geq 0.5$ is marked as foreground, otherwise it's considered as part of background. After the segmentation step, we discard all the small isolated regions that have an area of less than 5% of the frame and we keep only the three largest connected components.

In Figure 3.3 an example with and without applying the Spatial Consistency method is depicted; notice this technique allows to better aggregate superpixels that are near the principal blob region.



(a) (b) (c)

(d) (e)

FIGURE 3.5: The EGO-HSGR dataset contains hands performing five gestures: a) point out; b) like; c) dislike; d) ok; e) victory.

## 3.3 Experimental results

Having fully described our method, we now want to evaluate its performance on two datasets: EDSH and EGO-HSGR. The recent publicly available EDSH dataset [33] consists in egocentric videos acquired to analyze performance of several hand detection methods. It consists of three videos (EDSH_1, used as train video, and EDSH_2 and EDSH_kitchen used as test videos) that contain indoor and outdoor scenes with large variations of illumination, mild camera motion induced by walking and climbing stairs. All videos are recorded at a resolution of 720p and a speed of 30 fps. Also, the dataset includes segmentation masks of hands.

We also generated a new dataset which contains 12 videos of indoor scenes (EGO-HSGR). Videos have been recorded with a Panasonic HX-A10 Wearable Camcorder (see 1.3.6) at a resolution of $800 \times 450$ with a 25FPS in two different locations: a library and department's exhibition area.

The aim of this dataset is to reproduce an environment similar to a museum for human and object interaction: paintings and posters are hung on the walls, true masterpieces or either its virtual images; the visitor walks and sometimes stops in front of an object of interest performing some gestures to interact with next generation wearable devices. We identify five different gestures that are used commonly: *point out*, *like*, *dislike*, *ok* and *victory*. These can

| **Features** | **EDSH_2** | **EDSH_kitchen** |
|---|---|---|
| HSV | 0.752 | 0.801 |
| + LAB | 0.754 | 0.808 |
| + LAB hist. | 0.755 | 0.823 |
| + HSV hist. | 0.755 | 0.823 |
| + Grad hist. | 0.758 | 0.828 |
| + Gabor | **0.761** | **0.829** |

TABLE 3.1: Performance by incrementally adding new features.

be associated to different action or used for record social experience. Fig. 3.5 shows some frame examples.

To evaluate performance of our pixel-level hand detector a subset of six videos are used (three for training and two for testing). Segmentation masks are provided every 25 frames for a total of 700 annotations. The F-score (harmonic mean of the precision and recall rate) is used to quantity hand detection performance.

In the following subsections, we are going to evaluate every single aspect of our method: the performance of the features, and of the proposed techniques: temporal consistency, spatial coherence and illumination invariance. We will also compare our approach to existing methods.

### 3.3.1 Features performance

First, we examine the effectiveness of our features to discriminate between hand and non-hand superpixels. Table 3.1 shows performance in terms of F-measure on EDSH dataset with different feature combinations: firstly we describe each superpixel with mean and covariance matrix of its pixel values in HSV colour space, then we do the same using LAB colour space and we add colour histograms. Lastly, we include a histogram of gradients and Gabor feature. In order to analyze how visual features impact on the performance, in this experiment we do not include the temporal and spatial context information by using a single random forest classifier. Note that although colour information plays a fundamental role for hand detection, some ambiguities between hands and other similar coloured object still remain; these can be reduced by adding features based on gradient histograms. In fact, the usage of the full descriptor slightly improves the performance.

| Features | EDSH_2 | EDSH_kitchen |
|---|---|---|
| II | 0.789 | 0.831 |
| II + TS | 0.791 | 0.834 |
| II + TS + SC | **0.852** | **0.901** |

TABLE 3.2: Performance comparison considering Illumination Invariance (II), Time Smoothing (TS) and Spatial Consistency (SC).

| | EDSH_2 | EDSH_kitchen |
|---|---|---|
| Video stabilization [31] | 0.211 | 0.213 |
| Single pixel colour [30] | 0.708 | 0.787 |
| Collection of random forest [33] | 0.835 | 0.840 |
| **Our method** | **0.852** | **0.901** |

TABLE 3.3: Hand segmentation comparison with the state-of-the-art.

### 3.3.2 Temporal Smoothing and Spatial Consistency

In this experiment we validate the proposed techniques that take into account illumination variations, time dependence and spatial consistency. Table 3.2 shows the F-measure scores obtained on EDSH dataset incrementally adding Illumination Invariance (II), Time Smoothing (TS) and Spatial Consistency (SC). Note that there is a significant improvement in performance when all these three techniques are applied together. In particular, illumination invariance substantially increases the performance with respect to results obtained using only visual features and a single random forest classifier, while the improvement introduced by temporal smoothing is less pronounced. The main contribution is given by Spatial Consistency, that prunes away small and isolated pixel groups and merge spatially nearby regions, increasing the F-measure score of about six percentage points. The proposed technique is also tested in our EGO-HSGR dataset obtaining an F-measure score of $0.908$ and $0.865$ for the EGO-HSGR_4 and EGO-HSGR_5 videos.

### 3.3.3 Comparison to related methods

In Table 3.3 we compare our results to several approaches on EDSH dataset: a single-pixel colour approach inspired by [30], a video stabilization approach based on background modeling using affine alignment of image frames inspired by [31] and the approach based on random forest, by Li *et al.* [33]. The single-pixel approach is a random regressor trained

only using single-pixel LAB colour values. The background modeling approach aligns sequences of 15 frames estimating their mutual affine transformations; pixels with high variance are considered to be foreground hand regions. As can be seen, although the single-pixel approach is conceptually simple, is still quite effective. In addition, we observe that the low performance of the video stabilization approach is due to large ego-motion because the camera is worn by the user. The method recently proposed by [33] is more similar to our approach, but the use of superpixels, the selection of a new set of local features and the introduction of temporal and spatial consistency allow us to outperforms that results.

We have addressed one of the main issues involved in hand gesture recognition: hand segmentation. The proposed algorithm will be exploited in the next chapter, where our gesture recognition approach will be introduced and fully described.

# Chapter 4

# Gesture recognition in the cultural heritage scenario

Having presented our hand segmentation approach, we move a step forward and introduce a more complex problem: detecting static and dynamic hand gestures. As we said in the introductory chapter, the target scenario of our research is the cultural heritage domain. In fact, our goal is to deploy new gesture-based human machine interfaces to enhance museum experience.

In this chapter we provide algorithms that perform gesture analysis to recognize user interaction with artworks. We also propose to use scalable and distributed wearable devices capable of communicating with each other and with a central server. In particular the connection with the central server allows our wearable devices to grab gestures of past visitors for improving gesture analysis accuracy.

The proposed gesture recognition algorithm is compared to the current state of the art on benchmark datasets showing superior performance, and is tested in real and virtual museum environments. We further demonstrate that our gesture recognition approach can achieve acceptable accuracy results even with a few training samples performed by the visitor, and can benefit from distributed training in which gestures performed by other visitors are exploited.

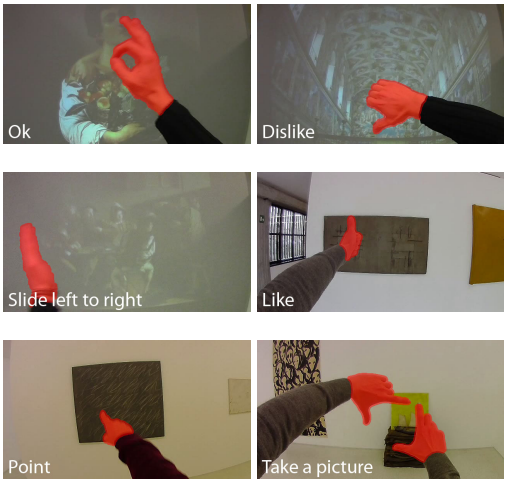



FIGURE 4.1: Results from the proposed hand segmentation and gesture recognition algorithms. Hand segmentation results are highlighted in red and detected gestures are reported in the bottom part of each frame.

## 4.1 Proposed Method

To our knowledge, the study of gesture recognition in the ego-centric paradigm has not yet been addressed. Even though not related to ego-vision domain, several approaches to gesture and human action recognition have been proposed. Sanin *et al.* [39] developed a new and more effective spatio-temporal covariance descriptor to classify gestures in conjunction with a boost classifier. Lui *et al.* [40, 41] used tensors and tangent bundle on Grassmann manifolds to classify human actions and hand gestures. Kim *et al.* [42] extended Canonical Correlation Analysis to measure video-to-video similarity to represent and detect actions in video. However, all these approaches are not appropriate for the ego-centric perspective, as they do not take into account any of the specific characteristics of this domain, such as fast camera motion and background cluttering.

Gesture recognition systems should recognize both static and dynamic hand movements. Therefore, we propose to describe each gesture as a collection of dense trajectories extracted around hand regions. Feature points are sampled inside and around the user's hands and tracked during the gesture; then several descriptors are computed inside a spatio-temporal volume aligned with each trajectory, in order to capture its shape, appearance and movement at each frame. These descriptors are coded, using the Bag of Words approach and power normalization, in order to obtain the final feature vectors, which are then classified using a linear SVM classifier. A summary of our approach is presented in Figure 4.2.

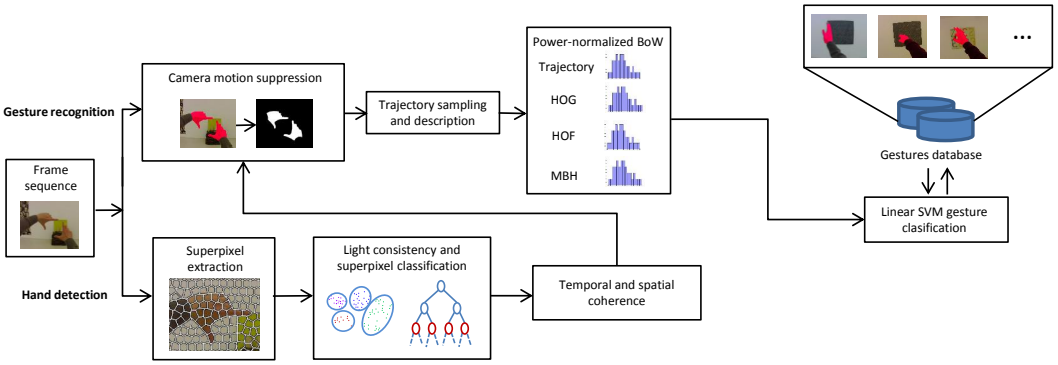



FIGURE 4.2: Outline of the proposed Gesture Recognition method.

In the next subsections we will introduce the main components of our approach: camera motion suppression, trajectory sampling and description, the power normalized Bag of Words and the final classification. We will also explain how these steps exploit the hand segmentation approach we have presented in the previous chapter.

### 4.1.1 Feature points sampling

We start by describing how we densely sample feature points for generating trajectories. To this aim, we consider a grid spaced by $W$ pixels. Sampling is carried out on each spatial scale separately. This guarantees that feature points equally cover all spatial positions and scales. Experimental results showed that a sampling step size of $W = 5$ pixels is dense enough to give good results over all datasets. There are at most 8 spatial scales in total, depending on the resolution of the video. The spatial scale increases by a factor of $1/\sqrt{2}$. Our goal is to track all these sampled points through the video. However, in homogeneous image areas without any structure, it is impossible to track any point. We remove points in these areas. Here, we use the criterion that the points are removed if the eigenvalues of the auto-correlation matrix are very small. We set a threshold $T$ on the eigenvalues for each frame $I$ as

$$T = 0.001 \times \max_{i \in I} \min(\lambda_i^1, \lambda_i^2) \tag{4.1}$$

where $(\lambda_i^1, \lambda_i^2)$ are the eigenvalues of point $i$ in the image $I$. Experimental results showed that a value of 0.001 represents a good compromise between saliency and density of the sampled points.

### 4.1.2 Camera motion suppression

A key step in our approach is to remove camera motion. To this end, the homography between two consecutive frames is estimated running the RANSAC algorithm on features points sampled as described.

An homography (or *projective transformation*) is defined by the following equation, where $x_1$ and $x_2$ are 2-d points expressed in homogeneous coordinates:

$$x_1 = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} x_2$$

and, because two homography matrices that differ for a scale are equivalent, this matrix has only 8 degrees of freedom. RANSAC is accomplished to produce a good homography with the following steps:

1. Randomly selecting a subset of feature points
2. Fitting an homography to the selected subset
3. Determining the number of outliers
4. Repeating steps 1-3 for a prescribed number of iterations, and select the iteration with the minimum amount of outliers

This would be the standard way to fit an homography between two frames. However, in first-person camera views hands movement is not consistent with camera motion and this generates wrong matches between the two frames. For this reason we introduce a segmentation mask that disregards feature matches belonging to hands. In fact, without the hand segmentation mask, many feature points from the user's hands would become inliers, degrading the homography estimation. As a consequence, the trajectories extracted from the video would be incorrect. Instead, computing an homography using feature points from non-hand regions allows us remove all the camera movements.

### 4.1.3 Trajectory Sampling and description

Having removed camera motion between two adjacent frames, trajectories can be extracted. The second frame is warped with the estimated homography, the optical flow between the first and the second frame is computed, and then feature points around the hands of the user are sampled and tracked following what [3] does for human action recognition.

Feature points are tracked on each spatial scale separately. For each frame $I_t$, its dense optical flow field $\omega_t = (u_t, v_t)$ is computed w.r.t. the next frame $I_{t+1}$, where $u_t$ and $v_t$ are the horizontal and vertical components of the optical flow. Given a point $P_t = (x_t, y_t)$ in frame $I_t$, its tracked position in frame $I_{t+1}$ is smoothed by applying a median filter on $\omega_t$:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega_t) \tag{4.2}$$

where $M$ is the median filtering kernel. The size of the median filter kernel $M$ is $3 \times 3$ pixels. As the median filter is more robust to outliers than bilinear interpolation, it improves trajectories for points at motion boundaries that would otherwise be smoothed out. Once the dense optical flow field is computed, points can be tracked very densely without additional cost. Another advantage of the dense optical flow is the smoothness constraints which allow relatively robust tracking of fast and irregular motion patterns. To extract dense optical flow fields, we use Farneback's algorithm [43] which embeds a translation motion model between neighborhoods of two consecutive frames. Polynomial expansion is employed to approximate pixel intensities in the neighborhood.

Points of subsequent frames are concatenated to form trajectories: $(P_t, P_{t+1}, P_{t+2}, ...))$. As trajectories tend to drift from their initial locations during the tracking process, we limit their length to $L$ frames in order to overcome this problem (based on our preliminary tests, we set $L = 30$). For each frame, if no tracked point is found in a $W \times W$ neighborhood, a new point is sampled and added to the tracking process so that a dense coverage of trajectories is ensured.

As static trajectories do not contain motion information, we prune them in a post-processing stage. Trajectories with sudden large displacements, most likely to be erroneous, are also removed. Such trajectories are detected, if the displacement vector between two consecutive frames is larger than 70% of the overall displacement of the trajectory.

Furthermore, in contrast to what [3] does, trajectories are restricted to lie inside and around the user's hands: at each frame the hand mask is dilated, and all the feature points outside the computed mask are discarded.

Then, the spatio-temporal volume aligned with each trajectory is considered, and Trajectory descriptor, HOG, HOF and MBH are computed around it. The trajectory descriptors encodes local motion patterns. Given a trajectory of length $L$, we describe its shape by a sequence $(\Delta P_t, ..., \Delta P_{t+L-1})$ of displacement vectors $\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - xt, y_{t+1} - y_t)$. The resulting vector is normalized by the sum of displacement vector magnitudes:

$$T = \frac{(\Delta P_t, .., \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\Delta P_j\|} \tag{4.3}$$

As we use trajectories with a fixed length of $L = 30$ frames, we obtain a 60 dimensional descriptor.

Besides the trajectory shape information, we also design descriptors to embed appearance and motion information. We compute descriptors within a space-time volume aligned with a trajectory to encode the motion information. The size of the volume is $N \times N$ pixels and $L$ frames long. To embed structure information, the volume is subdivided into a spatio-temporal grid of size $n_\sigma \times n_\sigma \times n_\tau$. We compute a descriptor (e.g., HOG, HOF or MBH) in each cell of the spatio-temporal grid, and the final descriptor is a concatenation of these descriptors. The default parameters for our experiments are $N = 32$, $n_\sigma = 2$, $n_\tau = 3$.

As we have already said in Chapter 2, HOG focuses on static appearance information, whereas HOF captures the local motion information. We compute the HOG and HOF descriptors along the dense trajectories. For both HOG and HOF, orientations are quantized into 8 bins with full orientation and magnitudes are used for weighting. An additional zero bin is added for HOF (i.e., in total 9 bins). It accounts for pixels whose optical flow magnitudes are lower than a threshold. Both descriptors are normalized with their $L_2$ norm. The final descriptor size is 96 for HOG (i.e., $2 \times 2 \times 3 \times 8$) and 108 for HOF (i.e., $2 \times 2 \times 3 \times 9$).

The motion boundary histogram (MBH) descriptor computes derivatives separately for the horizontal and vertical components of the optical flow. The descriptor encodes the relative motion between pixels, and since MBH represents the gradient of the optical flow, locally constant camera motion is removed and information about changes in the flow field (i.e., motion boundaries) is kept. MBH is more robust to camera motion than optical flow, and thus more discriminative. We employ MBH s motion descriptor for trajectories. The MBH

descriptor separates optical flow $\omega = (u, v)$ into its horizontal and vertical components. Spatial derivatives are computed for each of them and orientation information is quantized into histograms. The magnitude is used for weighting. We obtain a 8-bin histogram for each component (i.e., MBHx and MBHy). Both histogram vectors are normalized separately with their $L_2$ norm. The dimension is 96 (i.e., $2 \times 2 \times 3 \times 8$) for both MBHx and MBHy. For both HOF and MBH descriptor computation, we reuse the dense optical flow that is already computed to extract dense trajectories.

While HOF and MBH are averaged on five consecutive frames, a single HOG descriptor is computed for each frame. In this way we can better describe how the hand pose changes in time. After this step, we get a variable number of trajectories for each gesture. In order to obtain a fixed size descriptor, the Bag of Words approach is exploited: we train four separate codebooks, one for each descriptor. Each codebook contains 500 visual words and is obtained running the $k$-means algorithm in the feature space.

Since BoW histograms in our domain tend to be sparse, they are power normalized to unsparsify the representation, while still allowing for linear classification. To perform power-normalization [44], the following function is applied to each bin $h_i$:

$$f(h_i) = \text{sign}(h_i) \cdot |h_i|^{\frac{1}{2}} \tag{4.4}$$

We have also observed that power-normalization greatly improves the final accuracy. The feature vector is then obtained by the concatenation of its four power-normalized histograms. Eventually, gestures are recognized using a linear SVM 1-vs-1 classifier. SVM 1-vs-all and SVM Multiclass classifiers were also tested.

## 4.2 Experimental Results

We now evaluate the performance of the proposed approach. To compare the performance of our gesture recognition algorithm with existing approaches, we test it on the Cambridge-Gesture database [45], which includes nine hand gesture types performed on a table, under different illumination conditions. To better investigate the effectiveness of the proposed approach in videos taken from the ego-centric perspective and in a museum setting, we also propose two far more realistic and challenging dataset which contains seven gesture classes,

| **Method** | **Set1** | **Set2** | **Set3** | **Set4** | **Overall** |
|---|---|---|---|---|---|
| TCCA [42] | 0.81 | 0.81 | 0.78 | 0.86 | 0.82 |
| PM [40] | 0.89 | 0.86 | 0.89 | 0.87 | 0.88 |
| TB [41] | **0.93** | 0.88 | 0.90 | 0.91 | 0.91 |
| Cov3D [39] | 0.92 | **0.94** | 0.94 | 0.93 | 0.93 |
| **Our method** | 0.92 | 0.93 | **0.97** | **0.95** | **0.94** |

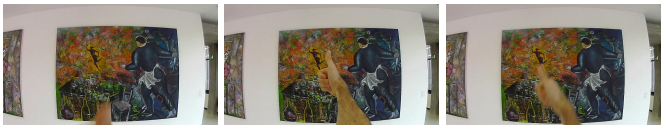
TABLE 4.1: Recognition rates on the Cambridge dataset.

performed by five subjects in an interactive exhibition room which functions as a virtual museum and in a real museum.

The Cambridge Hand Gesture dataset contains 900 sequences of nine hand gesture classes. Although this dataset does not contain ego-vision videos it is useful to compare our results to recent gesture recognition techniques. In particular, each sequence is recorded with a fixed camera, placed over one hand, and hands perform leftward and rightward movements on a table, with different poses. The whole dataset is divided in five sets, each of them containing image sequences taken under different illumination conditions. The common test protocol, proposed in [45], requires to use the set with normal illumination for training and the remaining sets for testing, thus we use the sequences taken in normal illumination to generate the BoW codebooks and to train the SVM classifier. Then, we perform the test using the remaining sequences.

Table 4.1 shows the recognition rates obtained with our gesture recognition approach, compared with the ones of tensor canonical correlation analysis (TCCA) [42], product manifolds (PM) [40], tangent bundles (TB) [41] and spatio-temporal covariance descriptors (Cov3D) [39]. Results show that proposed method outperforms the existing state-of-the-art approaches.

We then propose the Interactive Museum dataset, a gesture recognition dataset taken from the ego-centric perspective in a virtual museum environment. It consists of 700 video sequences, all shot with a wearable camera, in an interactive exhibition room, in which paintings and artworks are projected over a wall, in a virtual museum fashion (see figure 4.3). The camera is placed on the user's head and captures a $800 \times 450$, 25 frames per second 24-bit RGB image sequence. In this setting, five different users perform seven hand gestures: *like*, *dislike*, *point*, *ok*, *slide left to right*, *slide right to left* and *take a picture*. Some of them (like the *point*, *ok*, *like* and *dislike* gestures) are statical, others (like the two *slide* gestures) are dynamical. This dataset is very challenging since there is fast camera motion and users have not been trained



(a) *Like* gesture in the *museum* setting

(b) *Dislike* gesture

(c) *Ok* gesture in the *museum* setting, in low light

(d) *Point* gesture

(e) *Slide right to left* gesture in the *museum* setting, while another visitor walks in

(f) *Slide left to right* gesture in the *demo room* setting performed by user 3

(g) *Take a picture* gesture in the *demo room* setting

FIGURE 4.3: Sample gestures from the Interactive Museum dataset and the Maramotti dataset.

| User | No segmentation | With segmentation |
|---|---|---|
| Subject 1 | 0.91 | **0.95** |
| Subject 2 | 0.87 | **0.87** |
| Subject 3 | 0.92 | **0.95** |
| Subject 4 | **0.96** | 0.94 |
| Subject 5 | 0.91 | **0.96** |
| Average | 0.91 | **0.93** |

TABLE 4.2: Gesture recognition accuracy on the Interactive Museum dataset with and without hand segmentation.

before recording their gestures, so that each user performs the gestures in a slightly different way, as would happen in a realistic context. We have publicly released our dataset[1].

Since Ego Vision applications are highly interactive, their setup step must be fast (i.e. few positive examples can be acquired). Therefore, to evaluate the proposed gesture recognition approach, we train a 1-vs-1 linear classifier for each user using only two randomly chosen gestures per class as training set. The reported results are the average over 100 independent runs.

In Table 4.2 we show the gesture recognition accuracy for each of the five subjects, and we also compare with the ones obtained without the use of the hand segmentation mask for camera motion removal and trajectories pruning. Results show that our approach is well suited to recognize hand gestures in the ego-centric domain, even using only two positive samples per gesture, and that the use of the segmentation mask can improve recognition accuracy.

On a different note, to test our approach in a real setting, we created a dataset with videos taken in the Maramotti modern art museum, in which paintings, sculptures and *objets d'art* are exposed. As in the previous dataset, the camera is placed on the user's head and captures a 800 $\times$ 450, 25 frames per second image sequence. The Maramotti dataset contains 700 video sequences, recorded by five different persons (some are the same of the Interactive Museum dataset), each performing the same gestures as before in front of different artworks. We have publicly released this dataset too[2].

[1] http://imagelab.ing.unimore.it/files/ego_virtualmuseum.zip
[2] http://imagelab.ing.unimore.it/files/ego_maramotti.zip

TABLE 4.3: Gesture recognition accuracy on the Maramotti dataset.

| **User** | **Single user's Gestures** | **Augmented** |
|---|---|---|
| User A | 0.54 | **0.65** |
| User B | 0.52 | **0.72** |
| User C | 0.68 | **0.68** |
| User F | 0.56 | **0.79** |
| User G | 0.53 | **0.72** |
| **Average** | 0.57 | **0.71** |

Figure 4.3 show some examples of gestures performed in the two datasets. In the Interactive Museum dataset, users perform gestures in front of a wall over which the works of art are projected. This setting is quite controlled: the illumination is constant, the art works are in low light, while hands are well illuminated. On the other hand, in the Maramotti dataset, users perform gestures in front of real artworks inside a museum. This is a realistic and very challenging environment: the illumination changes, other visitors are present and sometimes walk in. In both cases there is significant camera motion, because the camera moves as the users move their heads or arms. It is also important to underline that users have not been trained before recording their gestures, so each user performs the gestures in a slightly different way, as would happen in a realistic context.

In Table 4.3 we show the results of our gesture recognition approach on the Maramotti dataset. As can be seen, in this case the challenging and real environment causes a drop in accuracy. This is mainly due to the illumination changes, to the presence of other visitors, and to the fact that often the artworks are better illuminated than hands. Since our wearable vision devices is fully connected to a central server, we show how the use of other visitors' gestures can improve the recognition accuracy. In our scenario each visitor coming to the museum performs, in the initial setup phase, two training gestures for each class. These training gestures from past visitors, manually checked, are used to augment the training set, so no erroneous data is accumulated into the model. In particular, in our test "Augmented" (Table 4.3) each ego-vision wearable device uses two randomly chosen gestures performed by its user as training, plus gestures performed by the remaining four users supplied by their devices to the central server. Results show that this distributed approach is effective and leads to a significant improvement in accuracy.

We described a novel approach to cultural heritage fruition based on ego-centric vision devices. Our work is motivated by the increasing interest in ego-centric vision and by the growth of the cultural market, which encourages the development of new interfaces to interact with the cultural heritage. We presented a gesture and painting recognition model that can deal with static and dynamic gestures and can benefit from a distributed training. Our gesture recognition and hand segmentation results outperform the state-of-the-art approaches on Cambridge Hand Gesture and CMU EDSH datasets. Finally, we ran an extensive performance analysis of our system on a wearable board.

In the next chapter, a real-time version of the described algorithm will be proposed, and we will describe the necessary optimizations steps to make such implementation real-time.

# Chapter 5

# A real-time implementation for the Odroid-XU developer board

Another important contribution of this thesis is a real-time implementation of our gesture recognition approach, ready to be used on an embedded device and suitable for real-world applications. To achieve this goal, we will need to modify and improve the classification module of our algorithm, and choose the Odroid-XU developer board as our target plaform. To make our implementation capable of running in real-time, we will also have to exploit several optimization techniques. At the end of this chapter we will also present some real-world applications of the proposed algorithm.

In the previous chapter, we classified frame sequences containing a gesture using a linear and power-normalized SVM classifier. In particular, we extracted trajectories with fixed length $L = 30$ from the frame sequence, we then applied a standard BoW approach, and the output of the BoW was our final feature vector, with fixed size and ready to be classified. We now ask ourselves how to modify this approach in order to take a frame stream (with unknown length) as input, instead of a fixed sized sequence. To this aim, we propose two approaches: a naive sliding window approach and a more complex label sequence learning approach. In the following, we will use the latter as our default implementation.

## 5.1 How to treat a frame stream

The simplest way to extend the proposed approach to frame streams is to exploit a sliding window. Let's suppose we have a window with size $W$, where $W$ is greater than the maximum gesture duration, and that the window step is $s$. Thus, at iteration $i$, the window will contain the following set of frames:

$$\{I_{1+i\cdot s}, I_{2+i\cdot s}, ..., I_{W+i\cdot s}\} \tag{5.1}$$

each window is then classified using the linear SVM classifier. Of course, in this case we have an additional class, the *non-gesture* one. We define as *non-gesture* any window that does not contain a complete gesture.

However, this implementation leads to, at least, three drawbacks: first, we would be always $s$ frames late, and therefore for a real-time gesture recognition we would have to choose a small $s$ and accept to classify the same frame lots of times. Secondly, the classifier wouldn't learn a clear concept of *non-gesture*, since the *non-gesture* class would contain partial gestures. Third, an SVM classifier doesn't learn label/label dependencies.

For this reasons, we will now treat the problem of classifying a frame stream as a *label sequence learning* problem. We start observing that our trajectory descriptors can be thought as a concatenation of single-point descriptors. Formally, let's consider a trajectory $T_i$ starting at frame $t_i$ and ending at frame $t_i+L-1$: we can express $T_i$ as $\left[P^i_{t_i}, P^i_{t_{i+1}}, ..., P^i_{t_i+L-1}\right]$, where $P^i_j$ is the point of trajectory $T_i$ at frame $j$. If we denote with $D(T_i)$ the descriptor of trajectory $T_i$, then $D(T_i)$ can be expressed as the concatenation of $d(P^i_{t_i}), d(P^i_{t_{i+1}}), ..., d(P^i_{t_i+L-1})$, where $d(P^i_j)$ is a descriptor computed around $P^i_j$, at frame $j$. Proving this statement is straightforward, given the nature of the descriptors employed.

Let's now consider a single frame $j$ of a frame stream, and suppose $\{T_1, T_2, ..., T_n\}$ are the trajectories crossing this frame. In this on-line version, we build a BoW histogram on each frame, being the input of the BoW clustering the set of descriptors of the trajectory points crossing the frame, i.e. $\{d(P^1_j), d(P^2_j), ..., d(P^n_j)\}$. Therefore, the corresponding BoW histogram won't represent a collection of trajectories anymore, but a collection of points, each one belonging to a different trajectory, and all related to the same frame.

Shortly, in the previous chapter the temporal dimension was encoded directly into the BoW input, now this will be up to the sequence classifier.

### 5.1.1 Classification

In Section 2.3.3 we described how a Struct SVM classifier can be used for label sequence learning. Now, we exploit the Structural SVM implementation by Altun *et al.* [27] to classify this modified version of our feature vectors.

First of all, we modify the source code provided by Joachims [46] in order to include higher order label-label and label-features interactions, so that the generic $\Psi(\mathbf{x}, \mathbf{y})$ takes the following form:

$$\Psi_{\epsilon,\tau}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \sum_{t=1}^{T} \Phi(\mathbf{x}^t) \otimes \Lambda^c(y^{t-\epsilon}) \otimes ... \otimes \Lambda^c(y^t) \\ \eta \sum_{t=1}^{T} \Lambda^c(y^{t-\tau}) \otimes ... \otimes \Lambda^c(y^t) \end{pmatrix} \tag{5.2}$$

where $\epsilon$ is the order of label-label dependencies, and $\tau$ is the order of label-features dependencies.

Furthermore, we investigate the use of different loss functions and their impact on recognition rates and confusion matrices. We define a generic loss function that exploits a per-token loss function $\delta(y^t, y'^t)$: $\Delta(\mathbf{y}, \mathbf{y}') = \sum_{t=1}^{T} \delta(y^t, y'^t)$, and propose two different per-token loss functions:

$$\delta_0(y, y') = \begin{cases} 0 & \text{if } y = y' \\ 1 & \text{otherwise} \end{cases} \tag{5.3}$$

$$\delta_1(y, y') = \begin{cases} 0 & \text{if } y = y' \\ 3 & \text{if } y = y_{ng}, y' \neq y_{ng} \vee y \neq y_{ng}, y' = y_{ng} \\ 1 & \text{otherwise} \end{cases} \tag{5.4}$$

where $y_{ng}$ is the *non-gesture* label. 5.3 is the default 0-1 loss function included in Joachims's code, and 5.4 is a slightly modifed version that aims at penalizing the case when a non-gesture frame is confused with a gesture frame or viceversa.

We train the Structural SVM classifier using four different gesture sequences from the Maramotti dataset, each five minutes long, and perform test on three sequences taken in front of different artworks. Using our gesture recognition algorithm we get a feature vector for each frame,

| $\tau$ | $\epsilon$ | **Accuracy %** |
|---|---|---|
| 1 | 0 | 92.2 % |
| 1 | 1 | 92.4 % |
| 2 | 0 | 92.4 % |
| 2 | 1 | 93.0 % |
| **2** | **2** | **93.8 %** |

TABLE 5.1: Accuracy results with different orders of dependencies of transitions and emissions

| | **Like** | **Dislike** | **Point** | **Ok** | **Slide LR** | **Slide RL** | **T. a pic** | **No ges.** |
|---|---|---|---|---|---|---|---|---|
| **Like** | 80% | 0 % | 0 % | 10% | 0 % | 0 % | 0 % | 10% |
| **Dislike** | 0 % | 91% | 0 % | 0 % | 0 % | 0 % | 0 % | 9% |
| **Point** | 0 % | 0 % | 91% | 9% | 0 % | 0 % | 0 % | 0 % |
| **Ok** | 0 % | 0 % | 0 % | 100% | 0 % | 0 % | 0 % | 0 % |
| **Slide left to right** | 0 % | 0 % | 0 % | 0 % | 90% | 10% | 0 % | 0 % |
| **Slide right to left** | 0 % | 0 % | 0 % | 0 % | 0 % | 100% | 0 % | 0 % |
| **Take a picture** | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 100% | 0 % |
| **No gesture** | 1% | 0 % | 0 % | 1% | 1% | 0 % | 3% | 93% |

TABLE 5.2: Confusion matrix using the $\delta_0$ per-token loss function. Percentages are rounded to the nearest integer and computed considering sequences of adjacent frames in which a gesture is performed or not.

| | **Like** | **Dislike** | **Point** | **Ok** | **Slide LR** | **Slide RL** | **T. a pic** | **No ges.** |
|---|---|---|---|---|---|---|---|---|
| **Like** | 90% | 0 % | 10% | 0 % | 0 % | 0 % | 0 % | 0 % |
| **Dislike** | 9% | 91% | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % |
| **Point** | 0 % | 0 % | 91% | 9% | 0 % | 0 % | 0 % | 0 % |
| **Ok** | 0 % | 0 % | 0 % | 100% | 0 % | 0 % | 0 % | 0 % |
| **Slide left to right** | 0 % | 0 % | 0 % | 0 % | 90% | 10% | 0 % | 0 % |
| **Slide right to left** | 0 % | 0 % | 0 % | 0 % | 0 % | 100% | 0 % | 0 % |
| **Take a picture** | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 100% | 0 % |
| **No gesture** | 0 % | 0 % | 0 % | 1 % | 0 % | 0 % | 1 % | 97% |

TABLE 5.3: Confusion matrix using the $\delta_1$ per-token loss function. Percentages are rounded to the nearest integer and computed considering sequences of adjacent frames in which a gesture is performed or not.

which is then classified using the Structural SVM classifier. Of course, since sequences contain frames in which the user is not performing any gesture, the classifier has to deal with a *non-gesture* class too.

Table 5.1 shows the accuracy results obtained with various orders of label-label and label-features dependencies, using the $\delta_0$ per-token loss function. As can be seen, high accuracy results can be achieved even with $\tau = 1$ and $\epsilon = 0$, whereas using second order dependencies there is a slight increase in accuracy. On the other hand, this would imply slower training and testing.

Moreover, the $\delta_0$ per-token loss function leads to a confusion matrix where some *non-gesture* examples are classified as gestures (see Table 5.2). Since false positives can be a major problem in human-machine interfaces, we propose the $\delta_1$ loss function, which increases the penalty when *non-gesture* examples are misclassified and viceversa. Table 5.3 shows that this loss function significantly reduce the confusion between gesture and *non-gesture* sequences.

## 5.2 Implementation

As stated in Section 1.3.8, on the Odroid-XU board only one of the two clusters can run at the same time. Since our application is CPU intensive, we choose the A15 cluster, which includes four cores and is the most performant cluster. This is done trough the following command:

```
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

which basically activates the `performance` governor as the controller for the big.LITTLE switching mechanism. The `performance` governor will simply turn off the A7 cluster and turn on the A15 cluster.

Then, having written a sequential implementation of our algorithm using the OpenCV library and our modified version of Joachims's SVM-HMM [46] [1], we compile it using `gcc` and the following options:

```
-g -mfpu=neon-vfpv4 -ftree-vectorize -mfloat-abi=hard -mtune=cortex-a15 -marm
```

The `-g` flag tells the compiler to produce debugging information in the operating system's native format, `-mfpu=neon` specifies what floating-point hardware (or hardware emulation)

[1] This is joint work with Francesco Paci.

is available on the target, `-ftree-vectorize` enables the auto-vectorizer (which will try to use SIMD instructions, when possible), `-mfloat-abi=hard` allows generation of floating-point instructions and uses FPU-specific calling conventions, and `-mtune=cortex-a15` tunes the performance of the compiler for the A15 target.

### 5.2.1 Optimizations

This first sequential version runs at 4.3 frames/s on $160 \times 120$ frames, and therefore needs 234 ms to elaborate each frame. Having profiled the application with Valgrind [47], we observe that the main bottleneck of this implementation is the multi-scale Farneback's optical flow, which requires 141 ms for each frame to run (see Figure 5.1 for a section of the Valgrind call graph). Farneback's algorithm is executed two times for each frame, since the optical flow is calculated both on the original frame on the warped one, so we exploit OpenMP parallel sections to run these two calls simultaneously on two different threads. Then, we use Neon intrinsics in order to exploit the SIMD capabilities of our CPU and thus gain a better performance on each thread.

A complete explanation of the optimizations made is beyond the purpose of this chapter, but, as an example, let's consider this for loop, inside the OpenCV implementation of Farneback's optical flow[2], which is responsible of 51 of the 234 ms:

```
for( ; x < width*5; x++ ) {
        float s0 = srow[m][x]*kernel[0];
        for( i = 1; i <= m; i++ )
                s0 += (srow[m+i][x] + srow[m-i][x])*kernel[i];
        vsum[x] = s0;
}
```

Once optimized with Neon SIMD, the previous code becomes:

```
int xstart = x;
float kernelext0[4];
fill_n(kernelext0,4,kernel[0]);
for( ; x < width*5-4; x+=4 ) {
           vst1q_f32(vsum+x, vmulq_f32(vld1q_f32(srow[m]+x),
                vld1q_f32(kernelext0)));
}
float a[width*5];
float kernelext[4];
```

[2]Source code is available at: https://github.com/Itseez/opencv/blob/0224a20ff6d0cf051cf818efb364048a2dcb716d/modules/video/src/optflowgf.cpp

```
for( i = 1; i <= m; i++ ) {
        fill_n(kernelext, 4, kernel[i]);
           for (x=xstart; x<width*5-4; x+=4) {
                        vst1q_f32(a+x, vaddq_f32(vld1q_f32(srow[m+i]+x),
                        vld1q_f32(srow[m-i]+x)));
                        vst1q_f32(vsum+x, vmlaq_f32(vld1q_f32(vsum+x),
                        vld1q_f32(a+x), vld1q_f32(kernelext)));
           }
}
```

As can be seen, the two for loops have been inverted and four floats are processed at each iteration now. The new code block takes only 20 ms to run.

Having included several other OpenMP/SIMD optimizations, our code runs at 10 frames/s on the A15 cluster, still not enough for real-time. Moreover, our algorithm needs an high frame rate in order to compute significant trajectories. Being Farneback's algorithm our main bottleneck, we could turn our attention to the PowerVR GPU, and use the OpenCL implementation of Farneback's algorithm included in OpenCV, for instance. Unfortunately, Hardkernel has not yet released an Ubuntu kernel that supports the PowerVR GPU[3], so the only way we have to increase speed is to further reduce the frame size and keep only one level of the spatial pyramid. Having reduced the frame size to $113 \times 85$ the code can run at 14 fps, which is quite a good result, since trajectories can still be extracted with good accuracy and the overall recognition performance is only slightly affected: in fact, we have observed a $5\%$ drop in recognition accuracy.

## 5.3 Some applications

During the process of adapting our approach to build a real-time gesture recognizer we have implemented and tested two applications, both of which have required a careful tuning and test phase: an *ego-vision jacket*, which basically is a jacket that embeds our developer board and a camera, placed on the chest, and a *gesture-based interface* for desktop applications.

**Ego-Vision Jacket**: Embedding a wearable camera and a board in a jacket has required some tailoring work: the lens of the camera has been placed on a butthole, sewed on the chest, and the board in a custom designed pocket, made with breathable fabric. Furthermore, a battery has been developed to make the board completely wireless.

[3]See: http://forum.odroid.com/viewtopic.php?f=61&t=2236.

Of course here the main technical issue is overheating, since the board has to stay inside a pocket: for this reason, we extensively measured the CPU temperature with different workload conditions and external temperatures. Results during a two hours execution of our algorithm shows that in fact the board temperature is remarkably higher than in normal conditions, but the embedded fan is still able to maintain the cores well below their maximum allowed temperature, that is 80 °C (see Fig. 5.2).

Our *ego-vision jacket* is the first prototype of a future high technological jacket we are going to develop, which will include other sensors, like a GPS antenna, and which will be fully connected to the internet (via EDGE/UMTS), to local area networks (through the Ethernet port or through the Wifi module) and to the user's smartphone via Bluetooth. We plan to use such jackets, in conjunction with augmented-reality algorithms, to enhance historical city visits.

**Gesture-based interface** This is basically a gesture-based controller for Power Point presentations or others desktop applications, that gives the user the ability to control a GUI using his gestures. Commands are passed through a simple socket, and then transformed in mouse or keyboards events. We exploit a client-server model, where our client is the Odroid-XU board, and the computer hosting the presentations acts as a server.

The board automatically connects to the remote server, via the following lines:

```
int sockfd;
struct sockaddr_in serv_addr;
char buffer[1];
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0) {
  cerr << "Error opening socket";
  exit(EXIT_FAILURE);
}
memset(&serv_addr, 0, sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = inet_addr(ip_address);
serv_addr.sin_port = htons(atoi(port_number));

if (connect(sockfd, (struct sockaddr*) &serv_addr, sizeof(serv_addr)) < 0) {
  cerr << "Unable to connect";
  exit(EXIT_FAILURE);
}
```

where `ip_address` and `port_number` are the server IP address port number. Once a gesture is recognized, the client sends the corresponding command on the socket, using

the `send` primitive. In our implementation commands are coded with an unsigned char, thus allowing 255 different commands. Similarly, the server creates a socket and listens on `port_number`. It then transforms the command into a sequence of mouse/keyboard events using `xdotool`. For example, if a gesture corresponds to pressing the Enter key, the following command is executed:

```
xdotool key KP_Enter
```

or, if the gesture corresponds to a mouse click, the server calls:

```
xdotool click 1
```

Of course, more complex mouse/keyboard actions can be defined. In our demo (see Figure 5.3), we used a set of three gestures: the *Point* gesture, to trigger an animation, and two *Slide* gestures, to move backwards and forwards in a Power Point presentation.

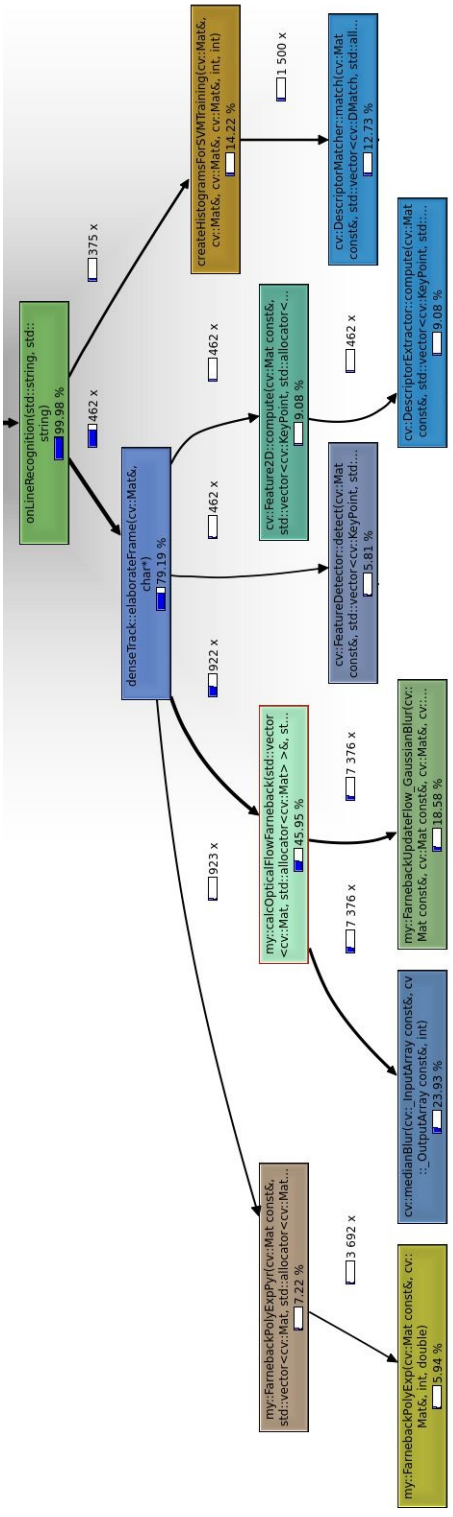



FIGURE 5.1: A section of the Valgrind Call Graph of the first sequential version. As can be seen, the optical flow takes more than the 50% of the execution time.

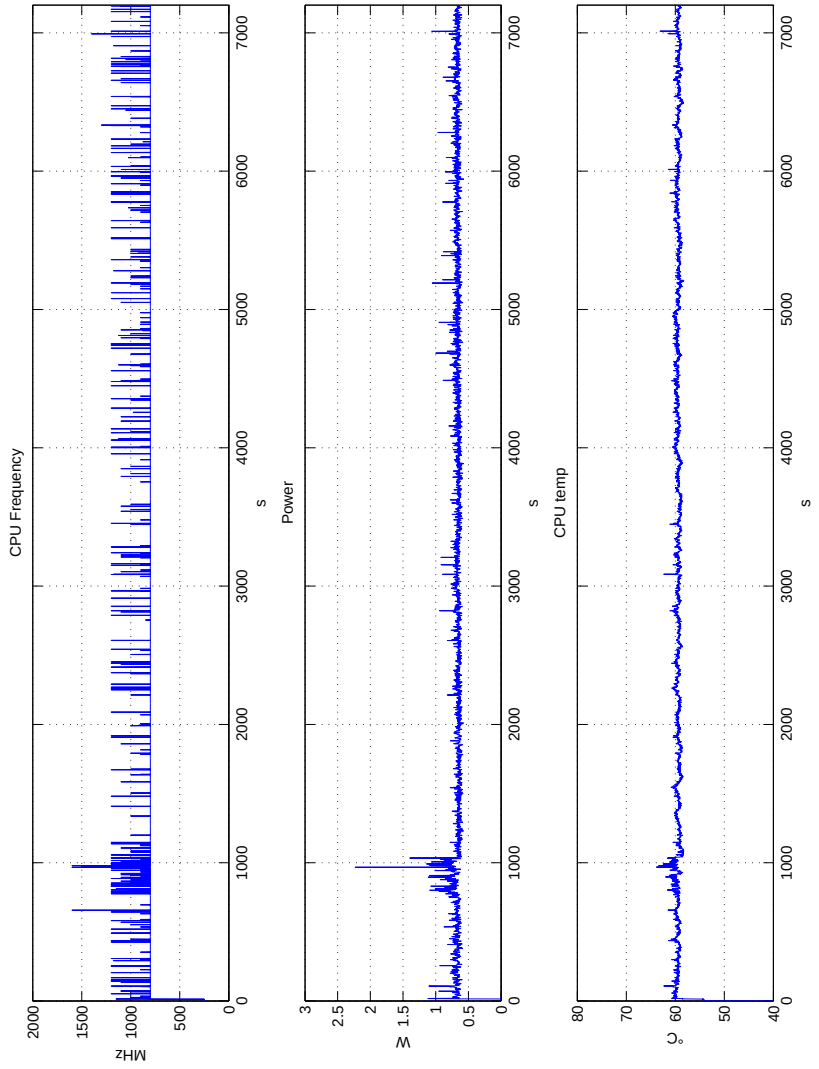



FIGURE 5.2: CPU Frequency, Power consumption and CPU Temperature during a two hours execution of our algorithm, inside the pocket of Ego-vision Jacket.


(a) The *point* gesture reveals a description of the current artwork.


(b) *Slide* gestures let the user move forward and backwards.

FIGURE 5.3: Gestures let the user control a virtual museum interface. A demo video is available at http://www.lorenzobaraldi.com/files/EgoVision_HCI.wmv

# Chapter 6

# Conclusion and future work

The work described in this thesis has been concerned with the development of an image segmentation algorithm, specifically designed for hands, and a new hand gesture recognition approach, inspired by dense trajectories. We have proposed a superpixel-based strategy for hand segmentation, where superpixels are classified with color, texture and gradient information, then temporal and spatial consistency are exploited. Moreover, we have proposed a gesture recognition approach that extracts trajectories inside and around the user's hands, and describes them with appearance and motion features. We have extensively tested both the algorithms in several different conditions, on standard datasets as well as on new and more challenging datasets we have developed, where they showed state-of-the-art performances. In addition to the design and development of these algorithms, we put a big effort into optimizing them and building a real-time gesture recognizer on a wearable computing device. We also showed that it can be used as a new kind of human-machine interface.

## 6.1 Recommendation for Future Work

Although the results presented here have demonstrated the effectiveness of the approach, it could be further developed in a number of ways.

First of all, the spatial and temporal consistency strategies work at the pixel level, while the classification part of the hand segmentation works at the superpixel level. This is a little contradictory, since the power and spatial support of superpixels are lost in the middle of

the pipeline: one, therefore, could develop a spatial consistency approach based on superpixels (or supervoxels), and propose a modified version of GrabCut capable of performing a GraphCut on superpixels. This would imply not only a cleaner approach, but also a more computationally efficient one.

The gesture recognition approach could be enhanced and extended too. For example, depth wearable cameras could be used, in conjunction with an extended version of our descriptors: we believe that this could bring some improvement and make the approach even more stable to light variations. Finally, global shutter cameras could reduce the effect of blur, which can significantly deteriorate the recognition accuracy, being our method based on keypoints tracking.

In closing, we witness that there is still much work to be done in order to develop new and effective human machine interfaces based on ego-vision. Still, this thesis is a preliminary and important contribution in this direction.

# Appendix A

# Publications

The research activity conducted in the context of this thesis has led to some publications in international journals and workshops. These are summarized below and included in the following pages.

1. **L. Baraldi**, S. Alletto, G. Serra, R. Cucchiara. "Interacting with Art: Ego-vision for Enriched Cultural Experience", *Machine Vision and Applications*. (Submitted)

2. **L. Baraldi**, F. Paci, G. Serra, L. Benini, R. Cucchiara. "Gesture Recognition using Wearable Vision Sensors to Enhance Visitors' Museum Experiences", *IEEE Sensors Journal*. (Submitted after minor revision)

3. **L. Baraldi**, F. Paci, G. Serra, L. Benini, R. Cucchiara. "Gesture Recognition in Ego-Centric Videos using Dense Trajectories and Hand Segmentation", in *IEEE Computer Vision and Pattern Recognition (CVPR) Embedded Vision Workshop (EVW)*, 2014.

4. G. Serra, M. Camurri, **L. Baraldi**, M. Benedetti, R. Cucchiara. "Hand Segmentation and Gesture Recognition in EGO-Vision", in *Proc. of ACM Multimedia International Workshop on Interactive Multimedia on Mobile and Portable Devices (IMMPD)*, 2013.

**Noname manuscript No.**
(will be inserted by the editor)

# Interacting with Art: Ego-vision for Enriched Cultural Experience

**Lorenzo Baraldi · Stefano Alletto · Giuseppe Serra · Rita Cucchiara**

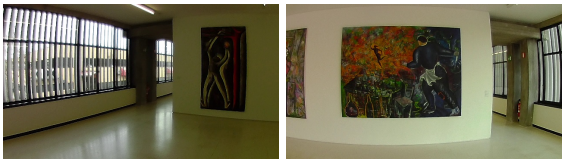
Received: date / Accepted: date

**Abstract** Everyone has been to a museum once. We all quarrelled with those self-service audio-guides at least once, striving to find a correspondence between what we were looking at and what the guide was trying to explain. Now imagine if we had a device capable of seeing what we saw to which we could have asked for the information we wanted by just pointing with a finger to the artwork. In this paper we provide a novel approach to cultural heritage experience: by the means of an ego-vision embedded platform we develop an approach that aims to a new, more entertaining and multimedia way of accessing museum knowledge. Our proposal deals with two main challenges: gesture recognition and painting recognition.We propose the use of dense trajectories sampled around the hand region to perform self-gesture recognition, understanding the way a user naturally interacts with an artwork. We extensively test our approach on two publicly available datasets and we further extend our experiments to both virtual and real museum scenarios where our method shows robustness when challenged with real-world data.

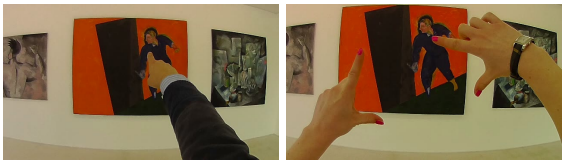
**Keywords** Wearable vision, interactive museum, embedded systems, gesture recognition, natural interfaces.

## 1 Introduction

The rebirth of interest in cultural heritage is an unequivocal fact. Thanks to the internet and to social media, the cultural market is becoming a key-point in many national economics strategies: the importance of the cultural sector is traditionally straightforward in Europe, but it is equally high also in other parts of the world. In fact, the Office of Travel and Tourism Industries in USA affirms that half of the Americans travelling abroad visit historical places; almost one-third visit cultural heritage sites; and one-quarter go to an art gallery or museum [1]. To deal with the new behaviours of tourists and art lovers which are more and more digital natives, the cultural places, archaeological sites, museums and exhibitions must deal with the need of new multimedia technologies. There is a big effort in proposing and providing new natural, attractive and immersive interfaces for interacting with the cultural heritage. Visitors need new enhanced experiences in cultural heritage [6]: for example singular and social experiences using mobile and smartphones [32] or in



(a) First-person viewpoint in museum experience



(b) Natural interaction with artwork

Fig. 1: Sample gestures from a real-world contemporary art museum.

L. Baraldi, S. Alletto, G. Serra, R. Cucchiara
Università degli Studi di Modena e Reggio Emilia
Dipartimento di Ingegneria "Enzo Ferrari"
Tel.: +390592056265
Fax: +390592056129
E-mail: {name.surname}@unimore.it

the crowd using crowd sourcing [26]. Further efforts are made in the digitalization of available cultural heritage objects like paintings or sculptures [13, 16].

The multimedia research community is largely involved in designing new devices that go beyond standard paper books or audio guides, such as touch interfaces [4], interactive interfaces [20, 36] and smartphones with games [5].

In this paper, we present our research in a new emerging technology, namely ego-vision for cultural experience in smart museums. Ego-vision features glass-mounted wearable cameras able to see what the visitor sees and perceiving the surrounding environment as he does. Since small low cost cameras with high processing capabilities are becoming available, these systems will be largely spread in the near future. For this reason, we propose their use in enhanced cultural experiences, making users interact with physical or digital cultural products to improve their fruition of information and to share knowledge on the social community. The technology that we present, being based on scalable wearable devices capable of communicating both with each other and the rest of the world, will be open to further integration with the internet-of-things and sensors such as RFID, GPS and gyroscopes. However, the research in ego-vision is still in its beginning and presents several challenges such as processing video recorded from an unconstrained prospective. In this work, we provide techniques that performs (i) painting recognition in a museum to achieve content-awareness, (ii) self-gesture analysis in order to recognize user interaction with artworks providing an natural and enriched interface to cultural heritage. Figure 1 shows an example of both the setting (Figure 1a) of our experiments and a real-world gesture analysis scenario (Figure 1b).

Our main novelties and contributions are the enabling technologies for multimedia interaction. In particular we propose robust algorithms for gesture analysis based on trajectory and shape information classification and painting recognition, tested both in real-world and virtual museums (see Figure 2. These methods have been compared to the current state of the art techniques over two public datasets showing our superior performance. Moreover, thanks to the very limited training requirements of our method, it can be extended to understand many other gestures not originally included. We also propose a painting recognition technique capable of classifying artworks despite the challenging conditions of real world museums like ever-changing lighting conditions, fast camera motion and partial occlusions.

The paper is structured as follows: in the next section, some related works for both multimedia technologies in cultural experiences and ego-vision systems are reported. In section 3 we present the details of the proposed solutions for painting recognition and self-gesture analysis. In section 4 results are shown and in conclusion some final considerations and proposals for future work are described.

## 2 Related Work

The present section explores the current state of the art in the two main areas of interest touched by our work: enhanced cultural experience in museums and egocentric vision.

Museums are traditionally spaces that have, by their very nature, an abundance of information available to visitors. In many cases, objects are accompanied by textual descriptions, usually too short or long to be adequate to the cultural interests of all visitors. Because of this, visitor access to museum collections can be often unsatisfactory or not appealing, especially to new generations. Personalization of multimedia museum content is one answer to this problem [27]. Personalization offers visitors a customized presentation of appropriate information related to the visitor's tastes and preferences. For these reasons, many solutions have been recently proposed for interactive user-profile based guides. An example is the work "SmartMuseum" [22]: by means of PDAs and RF-IDs, a visitor can gather information about what the museum displays, building a customized visit based on his or her interests inserted, prior to the visit, on their website. This approach brought an interesting novelty when first released, but it has some very limiting flaws. First of all, being tied to RFIDs does not allow reconfiguring the museum without rethinking the entire structure of the knowledge on which the project is based. Furthermore, researches demonstrated how the use of PDAs devices on the long term decreases the quality of the visit due to their users paying more attention to the tool rather than to the work of art itself. Similarly in [21], authors proposed to customize museums experiences with machine learning techniques applied on the answer to questionnaires that the users should compile before the visit. In both proposals the main flaw is the need to invasive interaction, asking the visitors to do something that probably they would not want to do. One of the valuable attempts to user profiling with wearable sensors was the "Museum Wearable" [31], a wearable computer which orchestrates an audio-visual narration as a function of the visitors' interests gathered from his/her physical path in the museum. However this prototype does not use any visual understanding algorithms for understanding the surrounding environment. For instance the estimation of the visitor location is based again on infrared sensors distributed

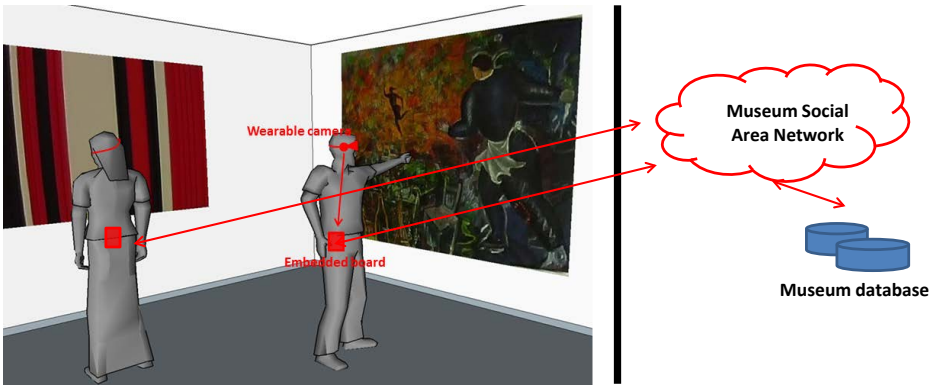



Fig. 2: An example of the interaction architecture of our proposal.

in the museum space. On a different note, the work by Yanulevskaya et al. [35] proposes a framework to automatically classify abstract paintings based on the emotional response they trigger in the visitors. Using statistical analysis and eye tracking devices, they calculate a score of each painting based on the emotions it provokes.

The ego-vision scenario has been addressed only recently by the research community and mainly to understand human activities and to recognize hand regions. Pirsiavash et al. [29] detected activities of daily living using an approach that involves temporal pyramids and object detectors tuned for objects appearance during interactions and spatial reasoning. Sundaram et al. [33] proposed instead to use Dynamic Bayesian Networks to recognize activities from low resolution videos, without performing hand detection and preferring computational inexpensive methods. Fathi et al. [11] used a bottom-up segmentation approach to extract hand held objects and trained object-level classifier to recognize objects; furthermore they also proposed an activity detection algorithm based on object state changes [10].

Regarding hand detection, Khan et al. in [17] studied color classification for skin segmentation. They pointed out how color-based skin detection has many advantages and potentially high processing speed, and demonstrated that Random Forest is one of the best classifiers for skin segmentation. Fathi et al. [11] proposed a different approach to hand detection, based on the assumption that background is static in the world coordinate frame, thus foreground objects are detected as to be the moving regions respect to the background. This approach is shown to be a robust tool for skin detection and hand segmentation in indoor environments, even if it performs poorly with more unconstrained scenarios. Li et al. [23] proposed a method with sparse feature selection which was shown to be an illumination-dependent strategy. To solve this issue, they trained a set of Random Forests indexed by a global color histogram, each one reflecting a different illumination condition.

The gesture analysis domain in ego-vision is rather unexplored. Even though not related to ego-vision domain, several approaches to gesture and human action recognition have been proposed. Kim et al. [18] extended Canonical Correlation Analysis to measure video-to-video similarity in order to represent and detect actions in video. Lui et al. [25,24] used tensors and tangent bundle on Grassmann manifolds to classify human actions and hand gestures. Sanin et al. [30] developed a new and more effective spatio-temporal covariance descriptor to classify gestures in conjunction with a boost classifier. However, all these approaches are not appropriate for the ego-centric perspective, as they do not take into account any of the specific characteristics of this domain, such as fast camera motion, hand presence and background cluttering.

## 3 Ego-vision for cultural heritage

We present the techniques we propose for improving cultural experience. The first component of our method is the egocentric artwork recognition, which is used to provide knowledge to the user without the need to explicitly input the painting identifier for which the information is desired.

The second one is recognizing the gestures of the user, hence the *self-gesture recognition*. In this regard, adapting to personal requests is a key aspect, in fact people in different cultures have very different ways of express through gestures. Our method can indeed learn from a very limited set of examples and it is robust to lighting changes and ego-motion.

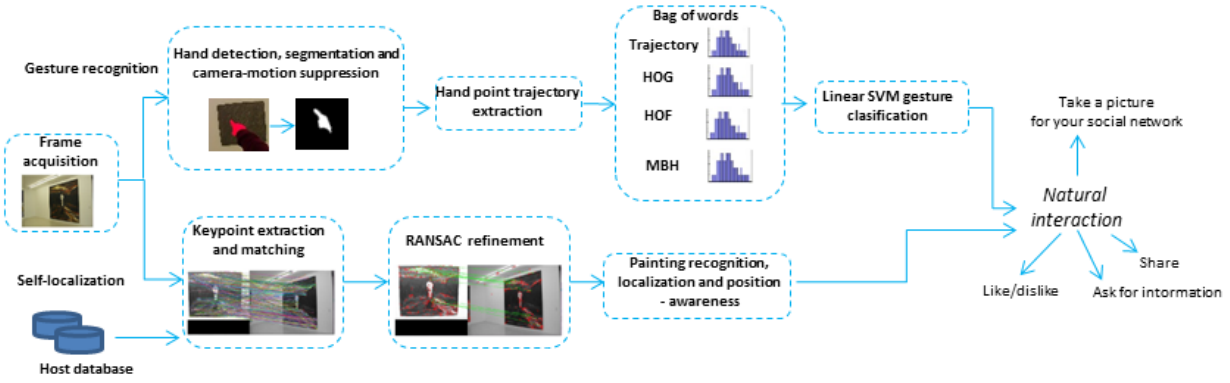



Fig. 3: Schematization of the proposed self-gesture and painting recognition methods.

Figure 3 provides a more detailed schematization of the workflow of the two components of our method, which will be deeper examined in the following of this section.

### 3.1 Egocentric painting recognition

The ability to understand what painting is the user looking at without the need of directly asking it to him or interact with some early deployed tag is one of the key features of our approach. In fact, this allows for a wide range of applications from user localization to social tagging and sharing. Aiming at the recognition of an artwork in a museum introduces a useful constraint: the number of paintings is finite and know in advance. Furthermore, a database containing painting data and features like descriptors and keypoints can be available, greatly reducing the required processing time. While object-recognition is a field in which some very interesting results are available, the real-world ego-vision setting we deal with makes this task full of challenges. For example, ego-vision presents extremely camera motion and different lighting conditions due to peculiar painting needs. Furthermore, paintings in museums are often protected by reflective glasses or occluded by other visitors, requiring in the necessity of a method capable of dealing with this difficulties too.

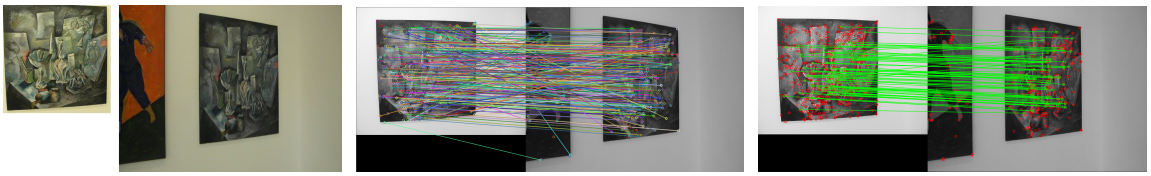
An important step to any recognition attempt in ego-vision is to understand whether the task should be performed or not. In fact, a typical ego-vision characteristic is that the camera wearer can have very fast head motion, e.g. when he is looking around for something. The high head motion can cause a significant blur in the video sequence resulting in an extremely low quality video. If not addressed properly, this situation can degrade the descriptor extraction on which our recognition recognition process is based at the point that its results can loose significance.

To deal with this challenge typical of the ego-vision scenario, previous to the recognition process we evaluate the amount of blurriness and decide whether to proceed with the frame or to skip it. The idea behind our approach is to compute the amount of gradient in the frame and to learn a threshold that discriminates a fast head movement due to the user looking around from the normal blur caused by small and inevitable motions. We define a blur function which recognizes the blur degree in a frame $I$, according to a threshold $\theta_B$:

$$Blur(I, \theta_B) = \sum_I \sqrt{\nabla S_x^2(I) + \nabla S_y^2(I)}, \quad (1)$$

where $\nabla S_x(I)$ and $\nabla S_y(I)$ are the $x$ and $y$ components of Sobel's gradient in the image and $\theta_B$ is the threshold under which the frame is discarded due to excessive motion blurriness, a parameter which can be learned by computing the average amount of gradient in a sequence. This step, that can be done in real-time, effectively allows to remove those frames that could prevent our painting recognition method to work properly.

Figure 4a shows an example of a room view captured by a first person perspective: the painting on the wall is heavily distorted by perspective preventing template matching techniques. In order to find a match between the framed artwork and its counterpart in the museum database, SIFT keypoints are extracted from the whole image. The need to proceed with this approach instead of sampling from a detected area derives from the difficulties that arise when trying to detect paintings from a fist-person perspective. Is it a painting or is that the window? Detection based on shape resulted in a false positive rate that prevented any further evaluation, hence we rely on sampling over the



(a) The painting template stored in the museum database (left) and the scene captured from the first person camera view (right).

(b) SIFT matching between the two artworks.

(c) Remaining keypoints after processing the matches with the RANSAC algorithm.

Fig. 4: Painting recognition in our approach.

whole image. Figure 4b shows the SIFT matching between the current frame and the template painting from the database. In order to improve the match quality, we process the matched keypoints using the RANSAC algorithm (Figure 4c). To further improve the matching results, a first thresholding step is performed: using a threshold over the distances between SIFT descriptors $\theta_S$, we remove the matches which have a distance greater than $\theta_S$. This allows to tune our method to different situations and primarily influences the recognition performance.

In order to understand whether the current frame really contains a painting or the computed matches refer to different elements (e.g. windows or architectonic details) a second thresholding step is applied. Using the threshold $\theta_d$ over the ratio between matches that survived the previous pruning steps (RANSAC and $\theta_S$) and the original amount of keypoints in the current frame, it can be decided whether the current frame contains an artwork or not. Adjusting this threshold can render the method more robust to noise and clutter situations or increase its detection range. A detailed analysis of the impact of these thresholds on the proposed method is presented in experimental section. A summarization of the painting recognition method is presented in Algorithm 1.

**Algorithm 1:** Painting recognition

**input** : Current frame, template database
**output**: Painting identifier

Compute current frame keypoints and local descriptors;;
**for** *each painting template* **do**
  read SIFT descriptors;
  calculate matching keypoints;
  apply RANSAC algorithm to discard outliers;
  remove matches with distance greater than $\theta_S$;
  compute ratio between remaining matches and total keypoints of the current frame;

Extract the painting with the highest $\frac{matches}{keypoints}$ ratio;
**if** $\frac{matches_{max}}{keypoints_{max}} > \theta_d$ **then**
  **return** Recognized painting identifier;
**else**
  **return** No painting detected

## 3.2 Gesture Recognition

Gesture recognition systems should recognize both static and dynamic hand movements. Therefore, we propose to describe each gesture as a collection of dense trajectories extracted around hand regions. Feature points are sampled inside and around the user's hands and tracked during the gesture; then several descriptors are computed inside a spatio-temporal volume aligned with each trajectory, in order to capture its shape, appearance and movement at each frame. These descriptors are coded, using the Bag of Words approach and power normalization, in order to obtain the final feature vectors which are then classified using a linear SVM classifier.

To describe information of shape, appearance and movement of the hand trajectory we rely on the following descriptors according to [34]: Trajectory descriptor, histograms of oriented gradients (HOG), of optical flow, and motion boundary histograms. The first one directly captures trajectory shape, while HOG [7] are a spacial descriptor representing the orientation of image grandients and thus encode the static appearance of the region surronding the trajectory. HOF and MBH [8] are based on optical flow and are used to capture motion information enforcing the temporal aspect of our method.

### *3.2.1 Camera motion removal*

In order to estimate hand motion, it is first necessary to remove the camera motion which is, semantically, noise. To do so, the homography transform between two consecutive frames is estimated running the RANSAC [12]

algorithm on densely sampled features points. SURF [3] features and sample motion vector are extracted from Farneback's optical flow [9] to get dense matches between frames.

In ego-vision, however, it is often the case where camera and hand motions are not consistent, resulting in wrong matches between the frames and degrading the consequent homography estimation. This introduces the need for an additional step based on a totally decoupled feature. We use a hand segmentation mask that allows us to remove the matches belonging to the user's hand, which would have resulted in incorrect trajectories. Computing the homography based only of non-hand keypoints allow to have a motion model consistent with the ego-motion of the camera which can, consequently, be removed.

#### *3.2.2 Gesture Description*

After the suppression of camera-motion, hand trajectories can be extracted. Using the previously estimated homography, the second frame is warped and the optical flow between the two frames is then recomputed in order to estimate the motion resulting from the hand movement. Feature points around the hand region are sampled and tracked in a way similar to what [34] does for human action recognition. We build a spatial pyramid with four layers, such that each layer has half the area of the previous one, and at each spatial scale we apply a threshold on the minimal eigenvalue of the covariance matrix of image derivatives. Each resulting keypoint $P_t = (x_t, y_t)$ is then tracked by the means of median filtering with kernel $M$ in a dense optical flow field $\omega = (u_t, v_t)$:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega)|_{(\bar{x_t}, \bar{y_t})}, \qquad (2)$$

where $(\bar{x_t}, \bar{y_t})$ is the rounded position of $P_t$.

Differently to [34], our trajectories are calculated under the constraint that they lie inside and around the user's hand: at each frame the hand mask is dilated and all keypoints still outside it are discarded.

A spatio-temporal volume aligned with each trajectory is then considered, as a collection of $32 \times 32$ patches, around the keypoint. Then, Trajectory descriptor, HOG, HOF and MBH are computed inside the volume. We introduce a difference in how to weight the temporal volume of each component of our feature vector: while HOF and MBH are averaged on five consecutive frames, a single HOG descriptor is computed for each frame. This allows us to describe the changes in the hand pose at a finer temporal granularity. This step results in a variable number of trajectory descriptors for each gesture. In order to obtain a fixed size descriptor, we exploit the Bag of Words approach training four separate codebooks, one for each descriptor. Each codebook contains 500 visual words and is obtained running the $k$-means algorithm in the feature space.

Since the histograms obtained from the Bag of Words in our domain tend to be sparse, they are power normalized to unsparsify the representation, while still allowing for linear classification. To perform power-normalization [28], the function:

$$f(h_i) = \text{sign}(h_i) \cdot |h_i|^{\frac{1}{2}} \qquad (3)$$

is applied to each bin $h_i$ in our histograms.

The final descriptor is then obtained by the concatenation of its four power-normalized histograms. Eventually, gestures are recognized using a linear SVM 1-vs-1 classifier.

#### *3.2.3 Hand Segmentation*

A key step in our gesture recognition process is the hand segmentation: this allows to distinguis between camera and hand motions, effectively pruning the trajectories that do not regard the user hand. Our method disregards all the semantic noise resulting from other motions in the scene, obtaining a descriptor that captures hand movements and shape as if the video sequence were captured by a fixed camera.

In order to compute the aforementioned segmentation mask, we extract superpixels at each frame using the SLIC algorithm [2]. It performs a $k$-means based local clustering of pixel in a space generated by $(labxy)$ where $(lab)$ are the coordinates of the LAB color space and $(xy)$ are the spatial coordinates on the image. Superpixels are represented with several features: histograms in the HSV and LAB color spaces (that have been proved to be good features for skin representation [17]), Gabor filters and a simple histogram of gradients, to discriminate between objects with a similar color distribution.

In order to achieve robustness to varying illumination conditions, instead of using a single classifier we train a collection of Random Forest classifiers indexed by a global HSV histogram. This results in distributing the training images among the classifiers using the $k$-means clustering on the feature space. At test time, the predictions from the five nearest classifier are averaged to make the final prediction.

Furthermore, semantic coherence over space and time is taken into account. Since past frames should affect the prediction for the current frame, a smoothing filter is applied: the prediction for each frame is replaced with a combination of the classifier results from past frames.

Then, in order to remove small and isolated pixel groups and also to aggregate bigger connected pixel groups, the GrabCut algorithm is applied to exploit spatial consistency.

## 4 Experimental Results

To investigate the performance of our method we record and publicly release two datasets: Interactive Museum and Maramotti Collection. The Interactive Museum is a dataset taken from the ego-centric perspective in a virtual environment where users can interact with digital artworks using gestures[1]. The Maramotti Collection dataset is a completely unconstrained real-world museum dataset recorded at the Maramotti Museum of contemporary art[2]. This dataset features different lighting conditions due to different museum rooms having different illumination requirements and partial artwork occlusions due to the presence of the user hands performing gestures. This dataset challenges both our painting and gesture recognition algorithms on a real-world scenario.

To further compare the performance of the proposed gesture recognition algorithm with existing approaches, we test it on the Cambridge-Gesture database [19], which includes nine hand gesture types performed on a table, under different illumination conditions. To evaluate the hand segmentation approach, we test it on the publicly available CMU EDSH dataset [23] which consists of three ego-centric videos with indoor and outdoor scenes and large variations of illuminations.

### 4.1 Painting Recognition

We test our painting recognition method on a subset of the real world Maramotti Collection museum dataset, which subset contains more than 13000 frames at $960 \times 540$ resolution annotated with the current visible painting. This scenario challenges our method in several ways: being a real museum lighting conditions can vary greatly from one room to another due to different artworks needs. Furthermore, being the same dataset used for gesture recognition, artworks can be partially occluded by the user hand during the recognition process. We evaluate our results in terms of detection precision and recall and classification accuracy: this allows to loosely decouple the detection performance from the classification phase, effectively reflecting the fact that our method is based on two different steps. Figure 5 shows the results of our experiments under varying detection and distance thresholds. It can be seen how deep can be the impact of different threshold values on the method performance: using a detection threshold $\theta_d = 0$ immediately produces a 100% recall due to, de facto, not performing any detection but accepting everything as a painting. This produces good classification accuracy performance if the distance threshold is small, but quickly degrades increasing the number of keypoints that are considered good matches (Fig. 5a). On the other hand, a detection threshold too high ($\theta_d = 0.15$) effectively produces a high amount of false negatives resulting in a high precision but very low recall and subsequently low classification accuracy due to the lack of detections (Fig. 5d). In this scenario, increasing the distance threshold $\theta_S$ increases the number of available keypoints to use in the detection increasing both accuracy and recall, with an accuracy upper bound that is significantly lower that the recall one. With these results in mind, our method fixes $\theta_S = 150$ and $\theta_d = 0.025$.

### 4.2 Gesture Recognition

The Cambridge Hand Gesture dataset contains 900 sequences of nine hand gesture classes. Although this dataset does not contain ego-vision videos it is useful to compare our results to recent gesture recognition techniques. In particular, each sequence is recorded with a fixed camera, placed over one hand, and hands perform leftward and rightward movements on a table, with different poses. The whole dataset is divided in five sets, each of them containing image sequences taken under different illumination conditions. The common test protocol, proposed in [19], requires to use the set with normal illumination for training and the remaining sets for testing, thus we use the sequences taken in normal illumination to generate the BoW codebooks and to train the SVM classifier. Then, we perform the test using the remaining sequences.

Table 1 shows the recognition rates obtained with our gesture recognition approach, compared with the ones of tensor canonical correlation analysis (TCCA) [18], product manifolds (PM) [25], tangent bundles (TB) [24] and spatio-temporal covariance descriptors (Cov3D) [30]. Results show that proposed method is able to overcome the existing state-of-the-art approaches.

We then present experiments on the Interactive Museum dataset: it consists of 700 video sequences, all shot with a wearable camera, in an interactive exhibition room, in which paintings and artworks are projected over a wall, in a virtual museum fashion. The camera is

[1] `http://imagelab.ing.unimore.it/files/ego_virtualmuseum.zip`

[2] `http://imagelab.ing.unimore.it/files/ego_maramotti.zip`

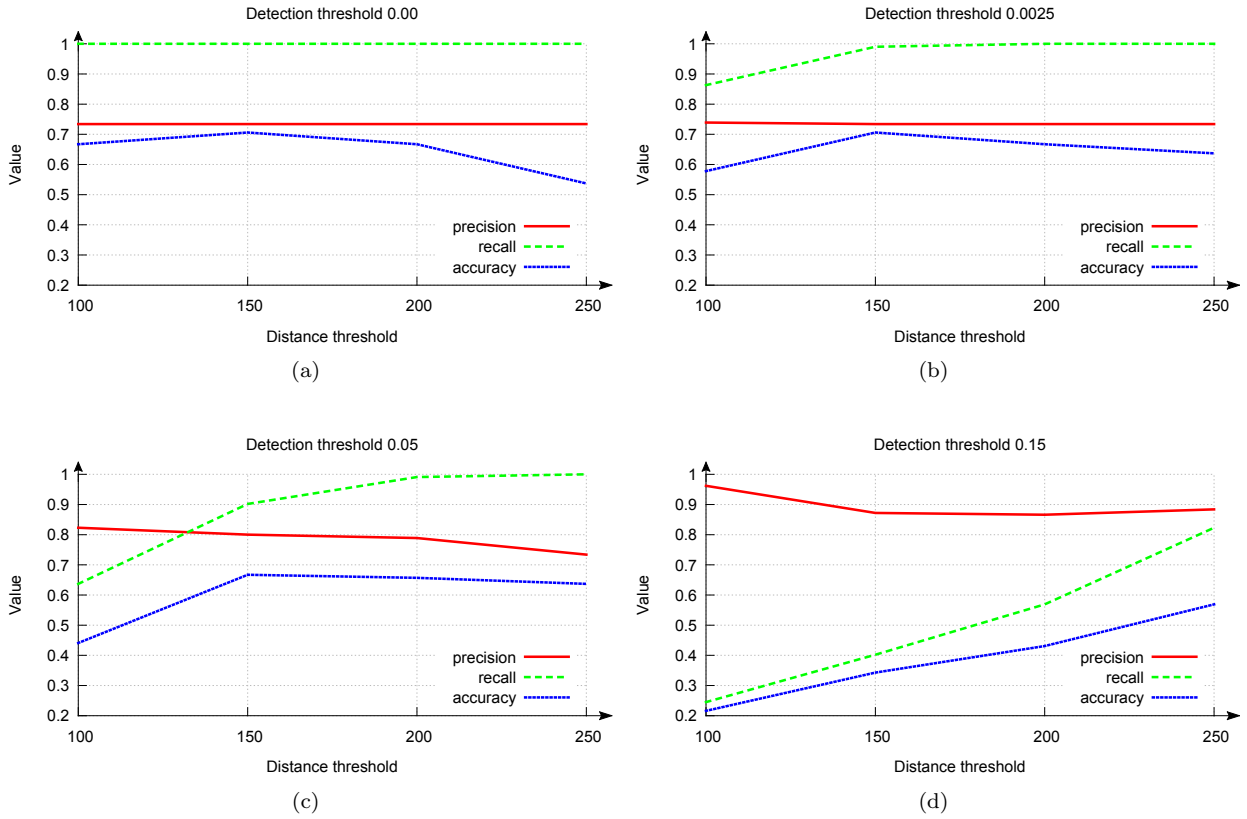



Fig. 5: Results of our painting recognition evaluation in terms of precision and recall of the detection step and accuracy of the classification phase.

| Method | Set1 | Set2 | Set3 | Set4 | Overall |
|---|---|---|---|---|---|
| TCCA [18] | 0.81 | 0.81 | 0.78 | 0.86 | 0.82 |
| PM [25] | 0.89 | 0.86 | 0.89 | 0.87 | 0.88 |
| TB [24] | **0.93** | 0.88 | 0.90 | 0.91 | 0.91 |
| Cov3D [30] | 0.92 | **0.94** | 0.94 | 0.93 | 0.93 |
| **Our method** | 0.92 | 0.93 | **0.97** | **0.95** | **0.94** |

Table 1: Recognition rates on the Cambridge dataset.

placed on the user's head and captures a 800 × 450, 25 frames per second 24-bit RGB image sequence. In this setting, five different users perform seven hand gestures: *like*, *dislike*, *point*, *ok*, *slide left to right*, *slide right to left* and *take a picture*. Some of them (like the *point*, *ok*, *like* and *dislike* gestures) are statical, others (like the two *slide* gestures) are dynamical. This dataset is very challenging since there is fast camera motion and users have not been trained before recording their gestures, so that each user performs the gestures in a slightly different way, as would happen in a realistic context.

Furthermore, we show an evaluation on our second more challenging dataset recorded in the Maramotti Collection contemporary art museum. It features more than 27000 frames annotated with gesture labels of the 7 gestures previously described. This dataset presents a more challenging environment due to its setting a real-world museum, featuring the inclusion of random visitors in the scene or different lighting conditions.

Since ego-vision applications are highly interactive, their setup step must be fast (i.e. few positive examples can be acquired). Therefore, to evaluate the proposed gesture recognition approach, we train a 1-vs-1 linear classifier for each user using only two randomly chosen gestures per class as training set. The reported results are the average over 100 independent runs.

In Table 2 we show the gesture recognition accuracy for each of the five subjects, and we also compare with the ones obtained without the use of the hand segmentation mask for camera motion removal and trajectories pruning. Results show that our approach is well suited to recognize hand gestures in the ego-centric do-

| User | Virtual Room |
|---|---|
| Subject 1 | 0.95 |
| Subject 2 | 0.87 |
| Subject 3 | 0.95 |
| Subject 4 | 0.94 |
| Subject 5 | 0.96 |
| Average | 0.93 |

Table 2: Gesture recognition accuracy on the Interactive Museum dataset with and without hand segmentation.

| User | Maramotti |
|---|---|
| Subject 1 | 0.94 |
| Subject 4 | 0.52 |
| Subject 5 | 0.68 |
| Subject 6 | 0.56 |
| Subject 7 | 0.54 |
| Average | 0.53 |

Table 3: Gesture recognition accuracy on the Maramotti Collection museum.

main, even using only two positive samples per gesture, and that the use of the segmentation mask can improve recognition accuracy. Furthermore, table 3 shows the accuracy results of our gesture recognition method applied to the Maramotti Collection museum, a real-world scenario where the resulting accuracy reflects all the challenges of a testing a method on an unconstrained environment.

### 4.3 Hand Segmentation

The CMU EDSH dataset consists of three ego-centric videos (EDSH1, EDSH2, EDSHK) containing indoor and outdoor scenes where hands are purposefully extended outwards to capture the change in skin color. As this dataset does not contain any gesture annotation, we use it to evaluate only the hand segmentation part.

We validate the techniques that we have proposed for temporal and spatial consistency. In Table 4 we compare the performance of the hand segmentation algorithm in terms of F1-measure, firstly using a single Random Forest classifier, and then incrementally adding illumination invariance, the temporal smoothing filter and the spatial consistency technique via the GrabCut algorithm application. Results shows that there is a significant improvement in performance when all the three techniques are used together: illumination invariance increases the performance with respect to the results obtained using only a single random forest classifier, while temporal smoothing and spatial consistency correct incongruities between adjacent frames, prune away small and isolated pixel groups and merge spatially nearby regions, increasing the overall performance.

| Features | EDSH2 | EDSHK |
|---|---|---|
| Single RF classifier | 0.761 | 0.829 |
| II | 0.789 | 0.831 |
| II + TS | 0.791 | 0.834 |
| II + TS + SC | **0.852** | **0.901** |

Table 4: Performance comparison considering Illumination Invariance (II), Temporal Smoothing (TS) and Spatial Consistency (SC).

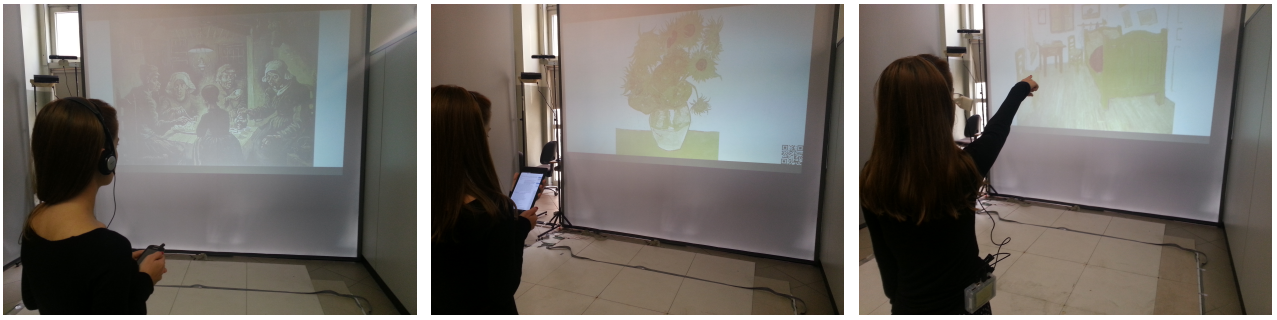
| Method | EDSH2 | EDSHK |
|---|---|---|
| Hayman and Eklundh [14] | 0.211 | 0.213 |
| Jones and Rehg [15] | 0.708 | 0.787 |
| Li and Kitani [23] | 0.835 | 0.840 |
| **Our method** | **0.852** | **0.901** |

Table 5: Hand segmentation comparison with the state-of-the-art.

Then, in Table 5 we compare our segmentation method with different techniques: a video stabilization approach based on background modeling [14], a single-pixel color method inspired by [15] and the approach proposed in [23] by Li et al. , based on a collection of Random Forest classifiers. As can be seen, the single-pixel approach, which basically uses a random regressor trained only using the single pixel LAB values, is still quite effective, even if conceptually simple. Moreover, we observe that the video stabilization approach performs poorly on this dataset, probably because of the large ego-motions these video present. The method proposed by Li et al. is the most similar to our approach, nevertheless exploiting temporal and spatial coherence we are able to outperform their results.

### 4.4 User experience evaluation

The system we provide allows for a novel kind of interaction with the artwork and hence requires some sort of user experience evaluation. Exploiting the setting of our virtual museum, we compare our method to two of the most common interfaces to museum knowledge: audio guides and QR-CODE based guides. The first one requires the user to input a number displayed next to the artwork and provides an audio description of the painting (Fig. 6a). Similarly, the second one requires



(a) (b) (c)

Fig. 6: Examples of evaluated interfaces in our virtual-museum environment.

| Interface | Score |
|---|---|
| Audio-Guide | 0.389 |
| QR-CODE | 0.401 |
| Our Method | 0.422 |

Table 6: Results of the user experience evaluation of different museum interaction interfaces.

the user to frame a QR-CODE sited close to the artwork with a tablet. A written description of the artwork is then displayed on the device's screen (Fig. 6b). To be compared with the aforementioned tools, we made our method recognize the current painting and provide an audio description of it at the detection of the "point" gesture (Fig. 6c). Using a Likert scale, we asked a set of 20 test subjects to answer the question "How natural did the interaction feel?" in our virtual museum setting with a score between 1 and 5, where 1 was "Unnatural" and 5 "Extremely natural". Most of the test subjects agree that our solution improves the fruition since it does not require the user to divert his attention from the painting, as showed by Table 6.

## 5 Conclusion

We described a novel approach to cultural heritage fruition. With the devices and the algorithms we proposed we overcame some of the limitations of self-service museum guides. We proposed a new technique of hand gesture recognition in ego-centric videos: our model can deal in real-time with static and dynamic gestures and can achieve high accuracy results even when trained with a few positive samples, which allows for an easy personalization to the different ways each user performs the same gestures. We also showed how our gesture recognition and hand segmentation results outperform the state-of-the-art approaches on Cambridge Hand Gesture and CMU EDSH datasets. In addition, we proposed a technique for automatic painting recognition that does not require any user interaction nor predetermined museum hardware such RF-IDs. While we use this recognition technique to provide the user content based on what he is really looking at, we recognize how the painting recognizion can be a key step for many further applications. For example, automatic painting recognition can be used to locate the user inside the museum providing services like customized visiting paths. We evaluated our new museum interface providing evidence that test subjects prefer to use an ego-vision gesture-based approach instead of the more common audio guides or QR-CODE based applications, showing great promise for future works.

**Acknowledgements** This work was partially supported by the PON R&C project DICET-INMOTO (Cod. PON04a2_D) and the CRMO project "Vision for Augmented Experiences". The authors would like to thank Collezione Maramotti for granting the use of their space in order to test our system in a realistic scenario.

## References

1. How the americans will travel 2015. tech rep. http://tourism-intelligence.com/
2. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34**(11), 2274–2282 (2012)
3. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Proc. of ECCV. Springer (2006)
4. Blöckner, M., Danti, S., Forrai, J., Broll, G., De Luca, A.: Please touch the exhibits!: Using nfc-based interaction for exploring a museum. In: Proc. of MobileHCI, pp. 71:1–71:2 (2009)
5. Coenen, T., Mostmans, L., Naessens, K.: Museus: Case study of a pervasive cultural heritage serious game. J. Comput. Cult. Herit. **6**(2), 8:1–8:19 (2013)
6. Cucchiara, R., Bimbo, A.D.: Visions for augmented cultural heritage experience. IEEE Multimedia **21**(1), 74–82 (2014)

7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. of CVPR, vol. 1, pp. 886–893. IEEE (2005)
8. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Proc. of ECCV. Springer (2006)
9. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Image Analysis, pp. 363–370. Springer (2003)
10. Fathi, A., Rehg, J.M.: Modeling actions through state changes. In: Proc. of CVPR (2013)
11. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: Proc. of CVPR (2011)
12. Fischler, M.A., Bolles, R.C.: 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)
13. Guidi, G., Frischer, B., Russo, M., Spinetti, A., Carosso, L., Micoli, L.: Three-dimensional acquisition of large and detailed cultural heritage objects. Machine Vision and Applications **17**(6), 349–360 (2006)
14. Hayman, E., Eklundh, J.O.: Statistical background subtraction for a mobile observer. In: Proc. of ICCV (2003)
15. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection (1999)
16. Khan, F., Beigpour, S., van de Weijer, J., Felsberg, M.: Painting-91: a large scale database for computational painting categorization. Machine Vision and Applications **25**(6) (2014)
17. Khan, R., Hanbury, A., Stoettinger, J.: Skin detection: A random forest approach. In: Proc. of ICIP (2010)
18. Kim, T.K., Cipolla, R.: Canonical correlation analysis of video volume tensors for action categorization and detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31**(8), 1415–1428 (2009)
19. Kim, T.K., Wong, K.Y.K., Cipolla, R.: Tensor canonical correlation analysis for action classification. In: Proc. of CVPR (2007)
20. Kortbek, K.J., Grønbæk, K.: Interactive spatial multimedia for communication of art in the physical museum space. In: Proc. of ACM Multimedia (2008)
21. Kuflik, T., Stock, O., Zancanaro, M., Gorfinkel, A., Jbara, S., Kats, S., Sheidin, J., Kashtan, N.: A visitor's guide in an active museum: Presentations, communications, and reflection. Journal on Computing and Cultural Heritage (JOCCH) **3**(3), 11 (2011)
22. Kuusik, A., Roche, S., Weis, F., et al.: Smartmuseum: Cultural content recommendation system for mobile users. In: Proc. of ICCIT (2009)
23. Li, C., Kitani, K.M.: Pixel-level hand detection in egocentric videos. In: Proc. of CVPR (2013)
24. Lui, Y.M., Beveridge, J.R.: Tangent bundle for human action recognition. In: In proc. of Automatic Face and Gesture Recognition and Workshops (2011)
25. Lui, Y.M., Beveridge, J.R., Kirby, M.: Action classification on product manifolds. In: Proc. of CVPR (2010)
26. Oomen, J., Aroyo, L., Marchand-Maillet, S., Douglass, J.: Personalized access to cultural heritage: Multimedia by the crowd, for the crowd. In: Proc. of ACM Multimedia (2012)
27. Pechenizkiy, M., Calders, T.: A framework for guiding the museum tours personalization. In: Proc. of PATCH'07 (2007)
28. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Proc. of ECCV (2010)
29. Pirsiavash, H., Ramanan, D.: Detecting activities of daily living in first-person camera views. In: Proc. of CVPR (2012)
30. Sanin, A., Sanderson, C., Harandi, M.T., Lovell, B.C.: Spatio-temporal covariance descriptors for action and gesture recognition. In: Proc. of Workshop on Applications of Computer Vision (2013)
31. Sparacino, F.: The museum wearable: Real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences. (2002)
32. Suh, Y., Shin, C., Woo, W., Dow, S., Macintyre, B.: Enhancing and evaluating users' social experience with a mobile phone guide applied to cultural heritage. Personal Ubiquitous Compututing **15**(6), 649–665 (2011)
33. Sundaram, S., Cuevas, W.W.M.: High level activity recognition using low resolution wearable vision. In: Proc. of CVPR (2009)
34. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action Recognition by Dense Trajectories. In: Proc. of CVPR (2011)
35. Yanulevskaya, V., Uijlings, J., Bruni, E., Sartori, A., Zamboni, E., Bacci, F., Melcher, D., Sebe, N.: In the eye of the beholder: Employing statistical analysis and eye tracking for analyzing abstract paintings. In: Proc. of ACM Multimedia, MM '12 (2012)
36. Zabulis, X., Grammenos, D., Sarmis, T., Tzevanidis, K., Padeleris, P., Koutlemanis, P., Argyros, A.: Multicamera human detection and tracking supporting natural interaction with large-scale displays. Machine Vision and Applications **24**(2) (2013)

# Gesture Recognition using Wearable Vision Sensors to Enhance Visitors' Museum Experiences

Lorenzo Baraldi, Francesco Paci, Giuseppe Serra, Luca Benini, Rita Cucchiara

***Abstract*—We introduce a novel approach to cultural heritage experience: by means of ego-vision embedded devices we develop a system which offers a more natural and entertaining way of accessing museum knowledge. Our method is based on distributed self-gesture and artwork recognition, and does not need fixed cameras nor RFIDs sensors. We propose the use of dense trajectories sampled around the hand region to perform self-gesture recognition, understanding the way a user naturally interacts with an artwork, and demonstrate that our approach can benefit from distributed training. We test our algorithms on publicly available datasets and we extend our experiments to both virtual and real museum scenarios where our method shows robustness when challenged with real-world data. Furthermore, we run an extensive performance analysis on our ARM-based wearable device.**

***Keywords*—Wearable vision, interactive museum, embedded systems, gesture recognition, natural interfaces.**

## I. Introduction

In recent years the interest in cultural heritage has reborn, and the cultural market is becoming a cornerstone in many national economic strategies. In the United States, a recent report of the Office of Travel and Tourism Industries claims that 51% of the 40 million Americans traveling abroad visit historical places; almost one third visit cultural heritage sites; and one quarter go to an art gallery or museum [1]. The same interest is found in Europe, where the importance of the cultural sector is widely acknowledged, South Asia and North Africa. The latest annual research from World Travel and Tourism Council shows that travel and tourism's total contribution to total GDP grew by 3.0% in 2013, faster than overall economic growth for the third consecutive year [2].

Consequently, to deal with an increasing percentage of "digital native" tourists, a big effort is underway to propose new interfaces for interacting with the cultural heritage. In this direction goes the solution "SmartMuseum" proposed by Kusik *et al.* [3]: by the means of PDAs and RFIDs, a visitor can gather information about what the museum displays, building a customized visit based on his or her interests inserted, prior to the visit, on their website. This project brought an interesting novelty when first released, but it has some limitations. First, being tied to RFIDs does not allow reconfiguring the museum without rethinking the entire structure of the exhibition. Furthermore, researches demonstrated how the use of mobile devices on the long term decreases the quality of the visit due to their users paying more attention to the tool rather than to the work of art itself.

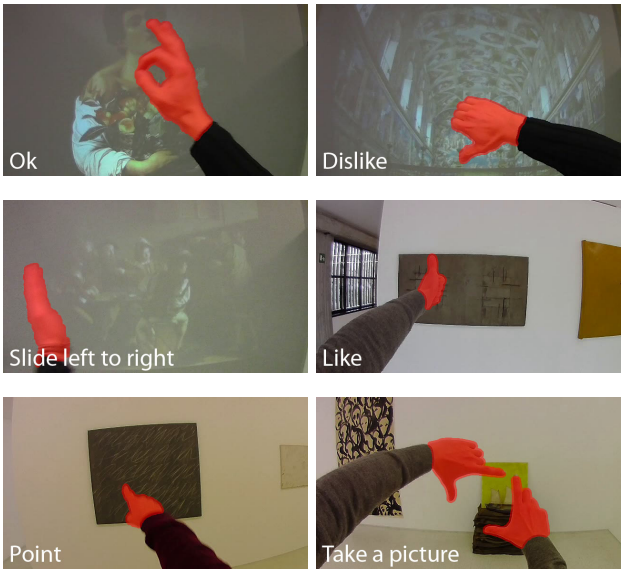



Fig. 1: Natural interaction with artworks: visitors can get specific content or share information about the observed artwork through simple gestures. Hand segmentation results are highlighted in red and detected gestures are reported in the bottom part of each frame.

In 2007 Kuflik *et al.* [4] proposed a system to customize visitors experiences in museums using software capable of learning their interests based on the answers to a questionnaire that they compiled before the visit. Similarly to SmartMuseum, one of the main shortcomings of this system is the need to stop the visitor and force him into doing something that he/she might not be willing to do. An interesting attempt to user profiling with wearable sensors was the Museum Wearable [5], a wearable computer which orchestrates an audiovisual narration as a function of the visitors' interests gathered from his/her physical path in the museum. However this prototype does not use any computer vision algorithm for understanding the surrounding environment. For instance the estimation of the

L. Baraldi, G. Serra and R. Cucchiara are with the Dipartimento di Ingegneria "Enzo Ferrari", University of Modena and Reggio Emilia, Italy (e-mail: baraldi.lorenzo@gmail.com; giuseppe.serra@unimore.it; rita.cucchiara@unimore.it).

F. Paci and L. Benini are with the Dipartimento dell'Energia Elettrica e dell'Informazione, University of Bologna, Italy (e-mail: f.paci@unibo.it; luca.benini@unibo.it).

L. Benini is also with Departement of Information Technology and Electrical Engineering, ETHZ, Zürich (e-mail: lbenini@iis.ee.ethz.ch).

Manuscript received xxx; revised xxxx.

visitor location is based again on infrared sensors distributed in the museum space.

Museums and cultural sites still lack of an instrument that provides entertainment, instructions and visit customization in an effective natural way. Too often visitors struggle to find the description of the artwork they are looking at and when they finds it, its detail level could be too high or too low for their interests. Moreover, frequently the organization of the exhibition does not reflect the visitors' interests leading them to a pre-ordered path which cultural depth could not be appropriate.

To overcome these limitations, we present a solution to enhance visitors' experiences based on a new emerging technology, namely *ego-vision* [6]. Ego-vision features glass-mounted wearable cameras able to see what the visitor sees and perceiving the surrounding environment as he does. We developed a wearable vision device for museum environments, able to replace the traditional self-service guides and overcoming their limitations and allowing for a more interactive museum experience to all visitors. The aim of our device is to stimulate the visitors to interact with the artwork, reinforcing their real experience, by letting visitors to replicate the gestures (e.g. point out to the part of the painting they're interested in) and behaviors that they would use to ask a guide something about the artwork.

In this work, we provide algorithms that perform gesture analysis to recognize user interaction with artworks, and artwork recognition to achieve content-awareness. The proposed solution is based on scalable and distributed wearable devices capable of communicating with each other and with a central server and hence does not require fixed cameras. In particular the connection with the central server allows our wearable devices to grab gestures of past visitors for improving gesture analysis accuracy, to get information and specific content of the observed artwork through the automatic recognition module, and to share visitor's feelings and photos on social networks. The main novelties and contributions of this paper are:

- A distributed architecture that improves museum visitors' experience. It is composed by ego-vision wearable devices and a central server, and it is capable of recognizing users' gestures and artworks.
- A gesture recognition approach specifically developed for the ego-vision perspective. Unlike standard gesture recognition techniques, it takes into account camera motion and background cluttering, and does not need markers on hands. It shows superior performance when compared on benchmark dataset, and can achieve good accuracy results even with a few training samples. We further demonstrate that it can benefit from distributed training in which gestures performed by past visitors are exploited.
- A novel hand segmentation approach that considers temporal and spatial consistency, and that is capable of adapting itself to different illumination conditions. It achieves the state-of-the-art results in the ego-vision EDSH dataset. Moreover, we show that when combined with our gesture recognition approach, it can improve the overall system accuracy.
- A performance evaluation of our algorithms on an ARM big.LITTLE heterogeneous platform for embedded devices which shows that our system can run in near real-time.

The rest of this article is structured as follows: in the next section we report related works for ego-vision. In Section III we give a detailed description of our system, focusing on self gesture recognition and artwork recognition. In Section IV our algorithms are compared with the state of the art and we present two novel datasets taken in real and virtual museum environments.

## II. Related Work

Only recently the ego-vision scenario has been addressed by the research community. The main effort has focused on understanding human activities and detecting hand regions. Pirsiavash *et al.* [7] detected activities of daily living using temporal pyramids and object detectors tuned for objects appearance during interactions and spatial reasoning. Sundaram *et al.* [8] proposed instead to use Dynamic Bayesian Networks to recognize activities from low resolution videos, without performing hand detection and preferring computational inexpensive methods. Fathi *et al.* [9] used a bottom-up segmentation approach to extract hand held objects and trained object-level classifier to recognize objects; furthermore they also proposed an activity detection algorithm based on object state changes [10].

Regarding hand detection, Khan *et al.* in [11] studied color classification for skin segmentation. They pointed out how color-based skin detection has many advantages and potentially high processing speed, invariance against rotation, partial occlusion and pose change. The authors tested Bayesian Networks, Multilayers Perceptrons, AdaBoost, Naive Bayes, RBF Networks and Random Forest. They demonstrated that Random Forest classification obtains the highest F-score among all the other techniques. Fathi *et al.* [9] proposed another approach to hand detection, based on the assumption that background is static in the world coordinate frame, thus foreground objects are detected as the moving regions with respect to the background. An initial panorama of the background is required to discriminate between background and foreground regions: this is achieved by fitting a fundamental matrix to dense optical flow vectors. This approach is shown to be a robust tool for skin detection and hand segmentation in limited indoor environments, even if it performs poorly with more unconstrained scenarios.

Li *et al.* [12] provide a historical overview of approaches for detecting hands from moving cameras. They define three categories: local appearance-based detection, global appearance-based detection, where a global template of hand is needed, and motion-based detection, which is based on the hypothesis that hands and background have different motion statistics. Motion-based detection approaches require no supervision nor training. On the other hand, these approaches may identify as hand an object manipulated by the user, since it moves together with his hands. In addition they proposed a method with sparse feature selection which was shown to be an illumination-dependent strategy. To solve this issue, they trained a set of

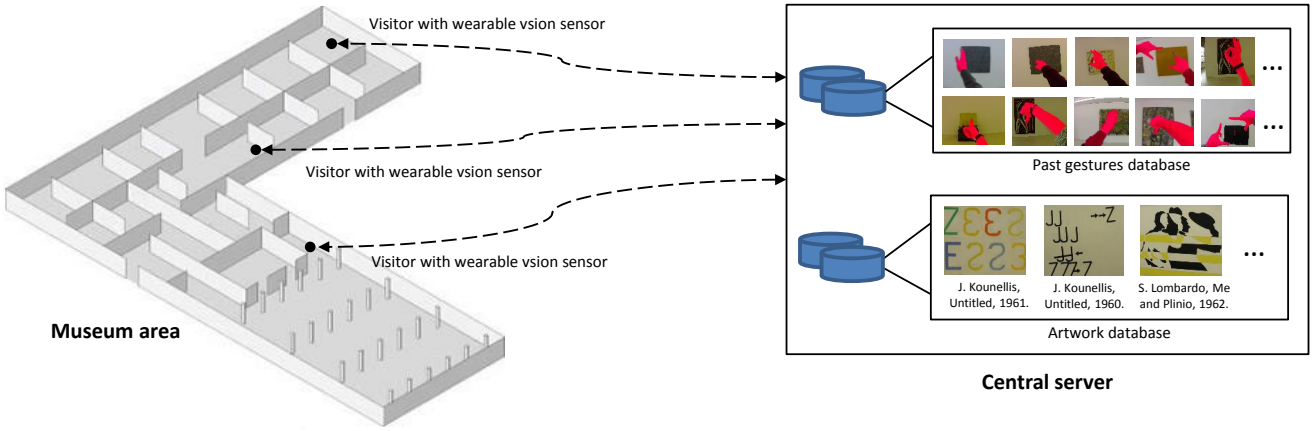



Fig. 2: Schema of the proposed distributed system. Each wearable vision sensor can communicate with a central server to send captured hand gestures and to retrieve gestures from other users and painting templates for artwork recognition. The central server contains two databases: the gesture database, which includes gestures performed by past visitors, and the artwork database, which contains artwork templates.

Random Forests indexed by a global color histogram, each one reflecting a different illumination condition.

Several approaches to gesture and human action recognition have been proposed. Sanin *et al.* [13] developed a new and more effective spatio-temporal covariance descriptor to classify gestures in conjunction with a boost classifier. Lui *et al.* [14], [15] used tensors and tangent bundle on Grassmann manifolds to classify human actions and hand gestures. Kim *et al.* [16] extended Canonical Correlation Analysis to measure video-to-video similarity to represent and detect actions in video. However, all these approaches are not appropriate for the ego-centric perspective, as they do not take into account any of the specific characteristics of this domain, such as fast camera motion and background cluttering. To our knowledge, the study of gesture recognition in the ego-centric paradigm has been partially addressed by P. Mistry *et al.* [17]. Their work presents a natural interface to interact with the physical world and embeds a projector to show results of that interaction. However they use colored markers on user's fingers to recognize gestures and they require a backpacked laptop as computational unit. Although our work could seem similar to this last approach, we move a step forward with respect to [17]: we proposed a fully automatic gesture recognition approach based on appearance and motion of the hands. Our approach can deal with background cluttering and camera motion and does not require any markers on fingers. In addition we provide an embedded solution that the user can easily wear.

## III. Proposed Architecture

Our cultural heritage system consists of a central server and a collection of wearable ego-vision devices, that embed a glass-mounted camera and an Odroid-XU developer board, serving as video-processing and network communication unit. There are several benefits in using such a portable device: the commercial availability and low costs for prototypes evaluation, the computational power and energy efficiency of the big.LITTLE architecture, the possibility of peripheral addition to extend connections and input devices. In particular, the developer board [18] we use embeds the ARM Exynos 5 SoC, that hosts a Quad big.LITTLE ARM processor (Cortex A15 and A7) [19]. To make it a portable demo device a battery pack of 3000 mAh has been added (see Figure 4).

This wearable device hosts the two main components of our system. The first one is the software that makes it capable of recognizing the gestures performed by its user and can customize itself, learning the way its user reach out for information. Adapting to personal requests is a key aspect in this process, in fact people in different cultures have very different ways of express through gestures. Our method is robust to lighting changes or ego-motion and can learn from a very limited set of examples gathered during a fast setup phase involving the user. The second component of our architecture is the artwork recognition, which allows not only to understand what the user is observing but also to infer the user's position.

The cooperation of ego-vision devices with the central server is two-fold. First, to increase gesture recognition accuracy, wearable devices receive gesture examples performed by past visitors and then send gestures for future users to augment the training set; second, the server also features a database of all the artworks in the museum, which is used for painting recognition and for obtaining detailed text, audio and video content. A schema of the proposed system is presented in Figure 2.

### A. Gesture recognition

Gestures can be characterized by both static and dynamic hand movements. Therefore, we consider a video sequence captured by a glass mounted camera, in which a gesture


Fig. 3: One user interacting with wearable camera.

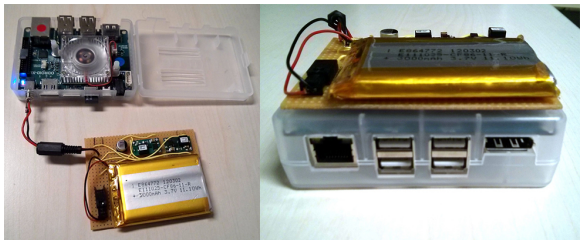


Fig. 4: The Odroid-XU board with battery pack.

may be performed, and describe it as a collection of dense trajectories extracted around hand regions. When the user's hands appear, feature points are sampled inside and around the hands and tracked during the gesture; then several descriptors are computed inside a spatio-temporal volume aligned with each trajectory to capture its shape, appearance and movement at each frame. We use the following descriptors, according to [20]: Trajectory descriptor, histograms of oriented gradients (HOG), of optical flow (HOF), and motion boundary histograms (MBH). The first one directly captures trajectory shape, while HOG [21] are based on the orientation of image gradient and thus encode the static appearance of the region surrounding the trajectory. HOF and MBH [22] are based on optical flow and are used to capture motion information enforcing the temporal aspect of our method. These descriptors are coded, using the Bag of Words approach and power normalization, to obtain the final feature vectors, which are then classified using a linear SVM classifier. Figure 5 provides a more detailed outline of the workflow of the proposed gesture analysis module.

*1) Camera motion removal:* To estimate the hand motion, it is first necessary to remove the camera motion, which is, semantically, noise. To do so, the homography transform between two consecutive frames is estimated running the RANSAC [23] algorithm on densely sampled features points: SURF [24] features and sample motion vector are extracted from the Farneback's optical flow [25] to get dense matches between frames. The choice of this particular optical flow algorithm is induced by our preliminary tests, in which Farneback's optical flow showed the best performance when compared to other popular optical flow algorithms, such as TV-L1 [26] and SimpleFlow [27].

In ego-vision, however, it is often the case where camera and hand motions are not consistent, resulting in wrong matches between the frames and degrading the consequent homography estimation. This introduces the need for an additional step based on a totally decoupled feature. We use a hand segmentation mask that allows us to remove the matches belonging to the user's hands, which could have resulted in incorrect trajectories. Computing the homography based only on non-hand keypoints allows to have a motion model consistent with the ego-motion of the camera which can, consequently, be removed.

*2) Gesture Description:* After the suppression of camera motion, trajectories can be extracted. Using the previously estimated homography, each frame of the sequence is warped and the Farneback's optical flow between each couple of adjacent frames is recomputed to estimate the motion resulting from the hand movement. Feature points around the hand region are sampled and tracked in a way similar to [20]. We build a spatial pyramid with four layers, such that each layer has half the area of the previous one, and at each spatial scale we apply a threshold on the minimal eigenvalue of the covariance matrix of image derivatives to obtain dense keypoints. We also ensure that keypoints are not duplicated among different spatial layers, and that a minimum distance between each couple of points is preserved. Each keypoint $P_t = (x_t, y_t)$ is then tracked by the means of median filtering with kernel $M$ in a dense optical flow field $\omega = (u_t, v_t)$:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega)|_{(\bar{x}_t, \bar{y}_t)} \quad (1)$$

where $(\bar{x}_t, \bar{y}_t)$ is the rounded position of $P_t$. Differently from [20], our trajectories are calculated under the constraint that they lie inside and around the user's hand: at each frame the hand mask is dilated and all keypoints still outside are discarded.

A spatio-temporal volume aligned with each trajectory is then build, as a collection of $32 \times 32$ patches around the keypoint. Then, Trajectory descriptor, HOG, HOF and MBH are computed inside the volume. We introduce a difference in how to weight the temporal volume of each component of our feature vector: while HOF and MBH are averaged on five consecutive frames, a single HOG descriptor is computed for each frame. This allows us to describe the changes in the hand pose at a finer temporal granularity. This step results in a variable number of descriptors for each video sequence. To obtain a fixed size descriptor, we exploit the Bag of Words approach training four separate codebooks, one for each descriptor. Each codebook contains $K$ visual words (in the experiments we fix $K = 500$) and is obtained running the $k$-means algorithm on the training data.

Since the histograms obtained from the Bag of Words in our domain tend to be sparse, they are power normalized to unsparsify the representation, while still allowing for linear classification. To perform power-normalization [28], the function:

$$f(h_i) = \text{sign}(h_i) \cdot |h_i|^{\frac{1}{2}} \quad (2)$$

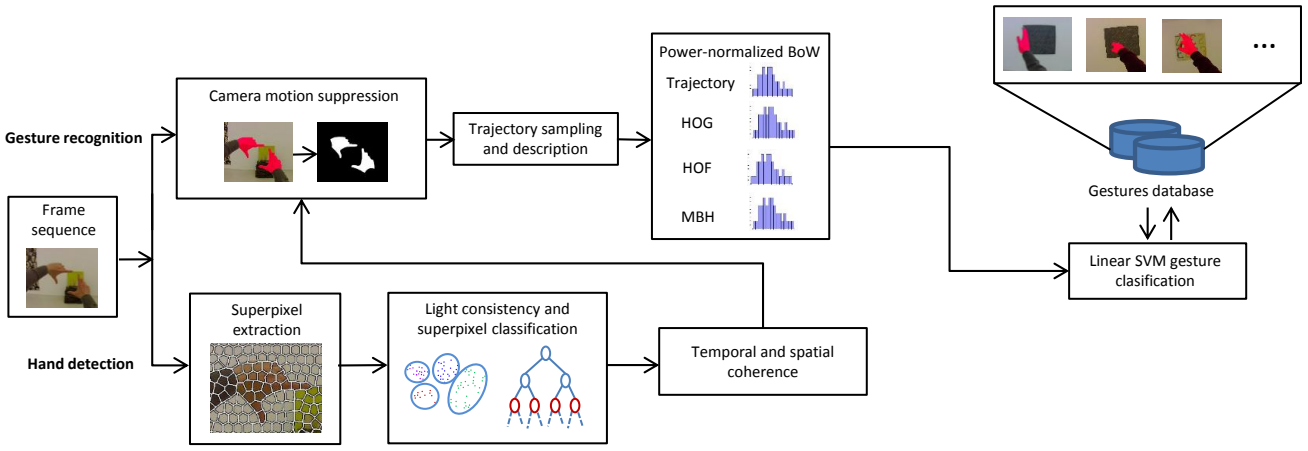



Fig. 5: An outline of the proposed gesture recognition module. It is roughly composed by three steps: the first step consists of hand segmentation and feature extraction, the second step performs BoW coding, the third step is the classification enhanced by past visitors' gestures.

is applied to each bin $h_i$ in our histograms.

The final descriptor is then obtained by the concatenation of its four power-normalized histograms. Finally, gestures are recognized using a linear SVM 1-vs-1 classifier.

### B. Hand Segmentation

As stated before, a hand segmentation mask is used to distinguish between camera and hand motions, and to prune away all the trajectories that do not belong to the user's hand. In this way, our descriptor captures hands movement and shape as if the camera was fixed, and disregards the noise coming from other moving regions that could be in the scene.

At each frame we extract superpixels using the SLIC algorithm [29], that performs a $k$-means-based local clustering of pixels in a 5-dimensional space, where color and pixel coordinates are used. Superpixels are then represented with several features: histograms in the HSV and LAB color spaces (that have been proven to be good features for skin representation [11]), Gabor filters and a simple histogram of gradients, to discriminate between objects with a similar color distribution.

*1) Illumination invariance:* To deal with different illumination conditions, we cluster the training images running the $k$-means algorithm on a global HSV histogram. Hence, we train a Random Forest classifier for each cluster. By using a histogram over all three channels of the HSV color space, each scene cluster encodes both the appearance of the scene and its illumination. Intuitively, this models the fact that hands viewed under similar global appearance will share a similar distribution in the feature space. Given a feature vector $\mathbf{l}$ of a superpixel $\mathbf{s}$ and a global appearance feature $\mathbf{g}$, the posterior distribution of $\mathbf{s}$ is computed by marginalizing over different clusters $c$:

$$P(\mathbf{s}|\mathbf{l},\mathbf{g}) = \sum_{c=1}^{k} P(\mathbf{s}|\mathbf{l},c)P(c|\mathbf{g}) \tag{3}$$

where $k$ is the number of clusters, $P(\mathbf{s}|\mathbf{l},c)$ is the output of the cluster-specific classifier and $P(c|\mathbf{g})$ is a conditional distribution of a cluster $c$ given a global appearance feature $\mathbf{g}$. In test phase, the conditional $P(c|\mathbf{g})$ is approximated using an uniform distribution over the five nearest clusters. It is important to highlight that the optimal number of classifiers depends on the characteristics of the dataset: a training dataset with several different illumination conditions, taken both inside and outside, will need an higher number of classifiers than one taken indoor. In addition, we model the hand appearance not only considering illumination variations, but also including semantic coherence in time and space.

*2) Temporal coherence:* To improve the foreground prediction of a pixel in a frame, we replace it with a weighted combination of its previous frames, since past frames should affect the prediction for the current frame.

We define a smoothing filter for a pixel $x_t^i$ from frame $t$ as:

$$\begin{aligned} P(x_t^i = 1) = \sum_{k=0}^{\min(t,d)} & w_k (P(x_t^i = 1|x_{t-k}^i = 1) \cdot \\ & \cdot P(x_{t-k}^i = 1|\mathbf{l}_{t-k},\mathbf{g}_{t-k}) + P(x_t^i = 1|x_{t-k}^i = 0) \\ & \cdot P(x_{t-k}^i = 0|\mathbf{l}_{t-k},\mathbf{g}_{t-k})) \end{aligned} \tag{4}$$

where $d$ is the number of past frames used, and $P(x_{t-k}^i = 1|\mathbf{l}_{t-k},\mathbf{g}_{t-k})$ is the probability that a pixel in frame $t-k$ is marked as hand part, equal to $P(\mathbf{s}|\mathbf{l}_{t-k},\mathbf{g}_{t-k})$, being $x_t^i$ part of $\mathbf{s}$. In the same way, $P(x_{t-k}^i = 0|\mathbf{l}_{t-k},\mathbf{g}_{t-k})$ is defined as $1 - P(\mathbf{s}|\mathbf{l}_{t-k},\mathbf{g}_{t-k})$. Last, $P(x_t^i = 1|x_{t-k}^i = 1)$ and $P(x_t^i = 1|x_{t-k}^i = 0)$ are prior probabilities estimated from the training set as follows:

$$P(x_t^i = 1|x_{t-k}^i = 1) = \frac{\#(x_t^i = 1, x_{t-k}^i = 1)}{\#(x_{t-k}^i = 1)}$$

$$P(x_t^i = 1|x_{t-k}^i = 0) = \frac{\#(x_t^i = 1, x_{t-k}^i = 0)}{\#(x_{t-k}^i = 0)} \quad (5)$$

where $\#(x_{t-k}^i = 1)$ and $\#(x_{t-k}^i = 0)$ are the number of times in which $x_{t-k}^i$ belongs or not to a hand region, respectively; $\#(x_t^i = 1, x_{t-k}^i = 1)$ is the number of times that two pixels at the same location in frame $t$ and $t-k$ belong to a hand part; similarly $\#(x_t^i = 1, x_{t-k}^i = 0)$ is the number of times that a pixel in frame $t$ belongs to a hand part and the pixel in the same position in frame $t-k$ does not belong to a hand region. Based on our preliminary experiments we set $d$ equal to three.

*3) Spatial consistency:* Given pixels elaborated by the previous steps, we want to exploit spatial consistency to prune away small and isolated pixel groups that are unlikely to be part of hand regions and also aggregate bigger connected pixel groups. For every pixel $x$, we extract its posterior probability $P(x_i^t)$ and use it as input for the GrabCut algorithm [30]. Each pixel with $P(x_i^t) \geq 0.5$ is marked as foreground, otherwise it's considered as part of background. After the segmentation step, we discard all the small isolated regions that have an area of less than 5% of the frame and we keep only the three largest connected components.

### C. Artwork recognition

The second component of our system is artwork recognition: a matching is established between the framed artwork and its counterpart on the system database. The real-world ego-vision setting we are dealing with makes this task full of challenges: paintings in a museum are often protected by reflective glasses or occluded by other visitors and even by user's hands, requiring a method capable of dealing with these difficulties too.

For this reason, we follow common approaches of object recognition based on interest points and local descriptors [31], [32], that have been proved to be able to capture sufficiently discriminative local elements and are robust to large occlusions.

First of all, SIFT keypoints are extracted from the whole image. The need to proceed with this approach instead of sampling from a detected area derives from the difficulties that arise when trying to detect paintings from a first person perspective. Detection based on shape resulted in high false positive rate, hence we rely on sampling over the whole image. To improve the match quality, we process the matched keypoints using the RANSAC algorithm. The ratio between the remaining matches and the total number of keypoints is then thresholded, allowing to recognize if the two images refer to the same artwork even in presence of partial occlusions. In addition, to avoid occlusions with user's hands we perform artwork recognition on the frames captured before the recognized gesture using a temporary buffer.

Fig. 6: Sample images from the Cambridge Hand Gesture dataset.

## IV. Experimental evaluation

To evaluate the performance of our gesture recognition and hand segmentation algorithms we first compare them with existing approaches. In particular we test our gesture module on the Cambridge-Gesture database [33], which includes nine hand gesture types performed on a table, under different illumination conditions. Whereas to evaluate the hand segmentation approach, we test it on the publicly available CMU EDSH dataset [12] which consists of three ego-centric videos with indoor and outdoor scenes and large variations of illuminations.

Furthermore, to investigate the effectiveness of the proposed approach in videos taken from the ego-centric perspective and in a museum setting, we also propose and release publicly two realistic and challenging datasets recorded in in an interactive exhibition room, which functions as a virtual museum, and a real museum of Modern Art. Finally, we perform a performance evaluation of the proposed algorithms on one of our wearable devices.

### A. Cambridge Hand Gesture dataset

The Cambridge Hand Gesture dataset contains 900 sequences of nine hand gesture classes. Although this dataset does not contain ego-vision videos it is useful to compare our results with recent gesture recognition techniques. In particular, each sequence is recorded with a fixed camera, placed over one hand, and hands perform leftward and rightward movements on a table, with different poses (see Figure 6). The whole dataset is divided in five sets, each of them containing image sequences taken under different illumination conditions. The common test protocol, proposed in [33], requires to use the set with normal illumination for training and the remaining sets for testing, thus we use the sequences taken in normal illumination to generate the BoW codebooks and to train the SVM classifier. Then, we perform the test using the remaining sequences.

Table I shows the recognition rates obtained with our gesture recognition approach, compared with the ones of tensor canonical correlation analysis (TCCA) [16], product manifolds (PM) [14], tangent bundles (TB) [15] and spatio-temporal covariance descriptors (Cov3D) [13]. Results show that the proposed method is effective in recognizing hand gestures, and that it outperforms the existing state-of-the-art approaches.

### B. EDSH Hand Segmentation dataset

The CMU EDSH dataset consists of three ego-centric videos (EDSH1, EDSH2, EDSHK) containing indoor and outdoor

TABLE I: Recognition rates on the Cambridge dataset.

| Method | Set1 | Set2 | Set3 | Set4 | Overall |
|---|---|---|---|---|---|
| TCCA [16] | 0.81 | 0.81 | 0.78 | 0.86 | 0.82 |
| PM [14] | 0.89 | 0.86 | 0.89 | 0.87 | 0.88 |
| TB [15] | **0.93** | 0.88 | 0.90 | 0.91 | 0.91 |
| Cov3D [13] | 0.92 | **0.94** | 0.94 | 0.93 | 0.93 |
| **Our method** | 0.92 | 0.93 | **0.97** | **0.95** | **0.94** |

(a) *Dislike* gesture.

(b) *Point* gesture.

(c) *Slide left to right* gesture.

Fig. 7: Gestures from the Interactive Museum dataset.

scenes where hands are purposefully extended outwards to capture the change in skin color. As this dataset does not contain any gesture annotation, we use it to evaluate only the hand segmentation part.

We validate the techniques that we have proposed for temporal and spatial consistency. In Table II we compare the performance of the hand segmentation algorithm in terms of F1-measure, firstly using a single Random Forest classifier, and then incrementally adding illumination invariance, the temporal smoothing filter and the spatial consistency technique via the GrabCut algorithm application. Results shows that there is a significant improvement in performance when all three techniques are used together: illumination invariance increases the performance with respect to the results obtained using only a single Random Forest classifier, while temporal smoothing and spatial consistency correct incongruities between adjacent frames, prune away small and isolated pixel groups and merge spatially nearby regions, increasing the overall performance.

Then, in Table III we compare our segmentation method with different techniques: a video stabilization approach based on background modeling [34], a single-pixel color method inspired by [35] and the approach proposed in [12] by Li *et al.*, based on a collection of Random Forest classifiers. As can be seen, the single-pixel approach, which basically uses a random regressor trained only using the single pixel LAB values, is still quite effective, even if conceptually simple. Moreover, we

TABLE II: Performance comparison considering Illumination Invariance (II), Temporal Coherence (TC) and Spatial Consistency (SC).

| Features | EDSH2 | EDSHK |
|---|---|---|
| Single RF classifier | 0.761 | 0.829 |
| II | 0.789 | 0.831 |
| II + TC | 0.791 | 0.834 |
| II + TC + SC | **0.852** | **0.901** |

TABLE III: Hand segmentation comparison with the state-of-the-art

| Method | EDSH2 | EDSHK |
|---|---|---|
| Hayman and Eklundh [34] | 0.211 | 0.213 |
| Jones and Rehg [35] | 0.708 | 0.787 |
| Li and Kitani [12] | 0.835 | 0.840 |
| **Our method** | **0.852** | **0.901** |

observe that the video stabilization approach performs poorly on this dataset, probably because of the large ego-motions these video present. The method proposed by Li *et al.* is the most similar to our approach, nevertheless exploiting temporal and spatial coherence we are able to outperform their results.

### C. Virtual and Real museum environments

We propose two new gesture recognition datasets taken from the ego-centric perspective in virtual and real museum environments. The Interactive Museum dataset consists of 700 video sequences, all shot with a wearable camera, taken in a interactive exhibition room, in which paintings and artworks are projected over a wall in a virtual museum fashion (see Figure 7). The camera is placed on the user's head and captures a 800 $\times$ 450, 25 frames per second 24-bit RGB image sequence. Five different users perform seven hand gestures: *like*, *dislike*, *point*, *ok*, *slide left to right*, *slide right to left* and *take a picture*. Some of them (like the *point*, *ok*, *like* and *dislike* gestures) are statical, others (like the two *slide* gestures) are dynamical. We have publicly released the dataset[1].

Since ego-vision applications are highly interactive, their setup step must be fast (i.e. few positive examples can be acquired). Therefore, to evaluate the proposed gesture recognition approach, we train a 1-vs-1 linear classifier for each user using only two randomly chosen gestures per class as training set.

In Table IV we show the gesture recognition accuracy for each of the five subjects of the Interactive Museum dataset. To validate the proposed technique, that combines gesture recognition and hand segmentation, we also show the results obtained without the use of the hand segmentation mask. As can be seen, our approach is well suited to recognize hand gestures in the ego-centric domain, even using only two positive samples per gesture, and the use of the segmentation mask for camera motion removal and trajectories pruning can

[1] http://imagelab.ing.unimore.it/files/ego_virtualmuseum.zip

TABLE IV: Gesture recognition accuracy on the Interactive Museum dataset with and without hand segmentation.

| User | No segmentation | With segmentation |
|---|---|---|
| User A | 0.91 | **0.95** |
| User B | **0.96** | 0.94 |
| User C | 0.91 | **0.96** |
| User D | 0.87 | **0.87** |
| User E | 0.92 | **0.95** |
| **Average** | 0.91 | **0.93** |

TABLE V: Gesture recognition accuracy on the Maramotti dataset.

| User | Single user's Gestures | Augmented |
|---|---|---|
| User A | 0.54 | **0.65** |
| User B | 0.52 | **0.72** |
| User C | 0.68 | **0.68** |
| User F | 0.56 | **0.79** |
| User G | 0.53 | **0.72** |
| **Average** | 0.57 | **0.71** |

improve recognition accuracy. The reported results are the average over 100 independent runs.

On a different note, to test our approach in a real setting, we created a dataset with videos taken in the Maramotti modern art museum, in which paintings, sculptures and *objets d'art* are exposed. As in the previous dataset, the camera is placed on the user's head and captures a 800 $\times$ 450, 25 frames per second image sequence. The Maramotti dataset contains 700 video sequences, recorded by five different persons (some are the same of the Interactive Museum dataset), each performing the same gestures as before in front of different artworks. We are currently waiting for the permission to release this dataset from the Maramotti museum.

Figures 7 and 8 show some examples of gestures performed in the two datasets. In the Interactive Museum dataset, users perform gestures in front of a wall over which the works of art are projected. This setting is quite controlled: the illumination is constant, the art works are in low light, while hands are well illuminated. On the other hand, in the Maramotti dataset, users perform gestures in front of real artworks inside a museum. This is a realistic and very challenging environment: the illumination changes, other visitors are present and sometimes walk in. In both cases there is significant camera motion, because the camera moves as the users move their heads or arms. It is also important to underline that users have not been trained before recording their gestures, so each user performs the gestures in a slightly different way, as would happen in a realistic context.

In Table V we show the results of our gesture recognition approach on the Maramotti dataset. As can be seen, in this case the challenging and real environment causes a drop in accuracy. This is mainly due to the illumination changes, to the presence of other visitors, and to the fact that often the artworks are better illuminated than hands. Since our wearable vision devices is fully connected to a central server, we show how the use of other visitors' gestures can improve the recognition accuracy. In our scenario each visitor coming to the museum performs, in the initial setup phase, two training gestures for each class. These training gestures from past visitors, manually checked, are used to augment the training set, so no erroneous data is accumulated into the model. In particular, in our test "Augmented" (Table V) each ego-vision wearable device uses two randomly chosen gestures performed by its user as training, plus gestures performed by the remaining four users supplied by their devices to the central server. Results show that this distributed approach is effective and leads to a significant improvement in accuracy.

### *D. Performance evaluation*

In this section we present our gesture recognition approach performance and optimizations. They are evaluated on the Hardkernel Odroid-XU board, already introduced in Section III. The tests we further present are performed on the Maramotti dataset. To evaluate the performance of our gesture recognition application, we split our algorithm in five main sub-modules (already deeply explained in the previous sections): Hand Segmentation, Camera motion removal, Trajectory extraction, Trajectory description, Power-normalized BoW and SVM-based Classification. To reach good performance on the Odroid-XU embedded device we applied different optimization techniques. Firstly compiler optimization has been used to speed-up code execution adding -O3 to compilation flags. Then we used Neon optimized instructions, by including neon library in source code and using these flags at compile time: -mfpu=neon-vfpv4 -mfloat-abi=hard -mtune=cortex-a15 -marm. Several low level "for cycles" have been balanced on different processors using OpenMP parallel regions. In Figure 9 we show the impact of each sub-module, separately, to elaborate 38 frames, that is the average gesture length within the Maramotti dataset. On the bottom part of each column we report the number of times each sub-module is called.

As can be seen, the Hand Segmentation is by far the most time consuming sub-module compared to the others. This is also due to the number of times each of sub-module is called:

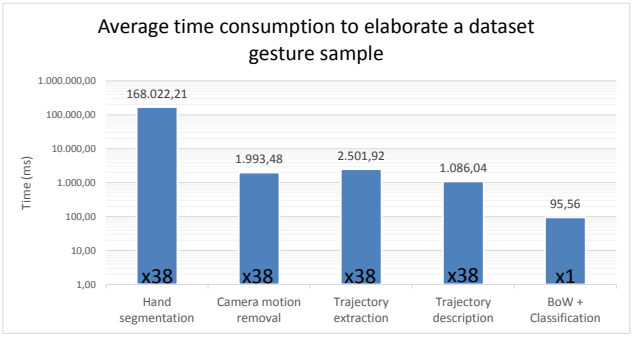



Fig. 9: Average time consumption of each sub-module to elaborate a gesture sample from the Maramotti dataset.

(a) *Like* gesture.

(b) *Ok* gesture, in low light.

(c) *Slide right to left* gesture, while another visitor walks in.

(d) *Take a picture* gesture.

Fig. 8: Gestures from the Maramotti dataset.

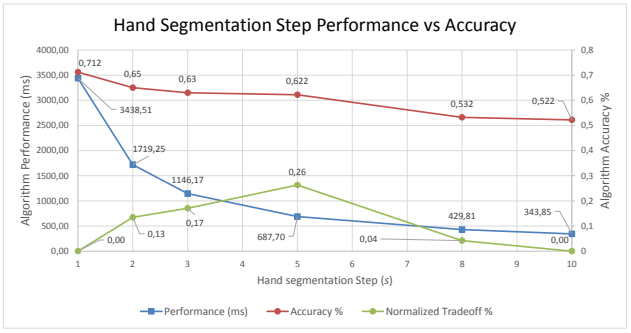



Fig. 10: Performance-accuracy trade-off of the proposed gesture recognition approach with different Hand Segmentation frame steps.

while Classification and Power-normalized BoW are executed just one time per gesture, the others are called one time per frame.

Therefore, we studied the performance-accuracy tradeoff of hand segmentation introducing a frame step between subsequent elaborations. The idea is to benefit of the hand segmentation not on each frame, but to introduce a gap between segmentation processing of the video stream and see how this impact on the gesture recognition accuracy. In this case, the hand segmentation mask is computed every $s$ frames. Trajectories and descriptors are still computed using all frames, but new keypoints are sampled only when the hand segmentation mask is available.

Figure 10 summarizes the whole gesture recognition algorithm performance and accuracy, applying different hand segmentation frame steps. We evaluated it as an average of the five Maramotti subjects, and the execution step of the Hand Segmentation is evaluated on the average length of the dataset samples (38 frames).

Three lines are shown in the graph: accuracy, performance and the normalized tradeoff. This last line has been computed as plain multiplication of normalized accuracy by normalized

TABLE VI: Gesture recognition performance with different step sizes.

| Step size | ms per frame | Frame/second |
|---|---|---|
| s = 1 | 3438.51 | 0.29 |
| s = 5 | 687.70 | 1.45 |
| s = 10 | 343.85 | 2.91 |

performance. The best normalized tradeoff is given by a step size of 5 frames. The average hands segmentation accuracy decreases of 9% (from 71.2% to 62.2%) in a tradeoff with a speed-up of 5x. This is a good result for performance, because paying a 9% accuracy loss we reduce the execution time from *3438.51 ms* to *687.70 ms*. In Table VI we show a summary of the performances obtained with different step sizes. As can be seen, the best computational performance on Odroid-XU platform is reached when using a step size of 10, and paying an accuracy loss of about 19%. Based on this analysis, we can state that our gesture recognition with hand segmentation is sufficiently accurate for real-life deployment and runs with an acceptable computation performance on ARM-based embedded devices.

## V. Conclusion

We described a novel approach to cultural heritage fruition based on ego-centric vision devices. Our work is motivated by the increasing interest in ego-centric vision and by the growth of the cultural market, which encourages the development of new interfaces to interact with the cultural heritage. We presented a gesture and painting recognition model that can deal with static and dynamic gestures and can benefit from a distributed training. Our gesture recognition and hand segmentation results outperform the state-of-the-art approaches on Cambridge Hand Gesture and CMU EDSH datasets. Finally, we ran an extensive performance analysis of our system on a wearable board.

## Acknowledgments

This work was partially supported by the FP7 project PHIDIAS (g.a. 318013), the FP7 ERC project MULTITHER-

MAN (g.a. 291125), the PON R&C project DICET-INMOTO (Cod. PON04a2_D) and the CRMO project "Vision for Augmented Experiences". The authors would like to thank Collezione Maramotti for granting the use of their space in order to test our system in a realistic scenario.

## REFERENCES

[1] "How the americans will travel 2015," http://tourism-intelligence.com.

[2] "Economic Impact of Travel & Tourism 2014," World Travel and Tourism Council, 2014.

[3] A. Kuusik, S. Roche, F. Weis *et al.*, "Smartmuseum: Cultural content recommendation system for mobile users," in *ICCIT'09: Fourth International Conference on Computer Sciences and Convergence Information Technology*, 2009, pp. 477–482.

[4] T. Kuflik, O. Stock, M. Zancanaro, A. Gorfinkel, S. Jbara, S. Kats, J. Sheidin, and N. Kashtan, "A visitor's guide in an active museum: Presentations, communications, and reflection," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 3, no. 3, p. 11, 2011.

[5] F. Sparacino, "The museum wearable: real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences," in *In Proc. of Museums and the Web*, 2002, pp. 17–20.

[6] T. Kanade and M. Hebert, "First-person vision," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2442–2453, Aug 2012.

[7] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *Proc. of CVPR*, 2012.

[8] S. Sundaram and W. W. M. Cuevas, "High level activity recognition using low resolution wearable vision," in *Proc. of CVPR*, 2009.

[9] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *Proc. of CVPR*, 2011.

[10] A. Fathi and J. M. Rehg, "Modeling actions through state changes," in *Proc. of CVPR*, 2013.

[11] R. Khan, A. Hanbury, and J. Stoettinger, "Skin detection: A random forest approach," in *Proc. of ICIP*, 2010.

[12] C. Li and K. M. Kitani, "Pixel-level hand detection in ego-centric videos," in *Proc. of CVPR*, 2013.

[13] A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell, "Spatio-temporal covariance descriptors for action and gesture recognition," in *Proc. of Workshop on Applications of Computer Vision*, 2013.

[14] Y. M. Lui, J. R. Beveridge, and M. Kirby, "Action classification on product manifolds," in *Proc. of CVPR*, 2010.

[15] Y. M. Lui and J. R. Beveridge, "Tangent bundle for human action recognition," in *In proc. of Automatic Face & Gesture Recognition and Workshops*, 2011.

[16] T.-K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 8, pp. 1415–1428, 2009.

[17] P. Mistry and P. Maes, "Sixthsense: A wearable gestural interface," in *ACM SIGGRAPH ASIA 2009 Sketches*. ACM, 2009, pp. 11:1–11:1.

[18] "Odroid-XU dev board by Hardkernel," http://www.hardkernel.com.

[19] "Samsung Exynos5 5410 ARM CPU," http://www.samsung.com/global/business/semiconductor/minisite/Exynos/products5octa_5410.html.

[20] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action Recognition by Dense Trajectories," in *Proc. of CVPR*, 2011.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[22] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 428–441.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[24] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proc. of ECCV*, 2006.

[25] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*. Springer, 2003, pp. 363–370.

[26] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Pattern Recognition*. Springer, 2007, pp. 214–223.

[27] M. Tao, J. Bai, P. Kohli, and S. Paris, "Simpleflow: A non-iterative, sublinear optical flow algorithm," in *Computer Graphics Forum*, vol. 31, no. 2pt1. Wiley Online Library, 2012, pp. 345–353.

[28] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. of ECCV*, 2010.

[29] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274–2282, 2012.

[30] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[31] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *in Proc ICCV*, 2003.

[32] A. D. Bagdanov, L. Ballan, M. Bertini, and A. Del Bimbo, "Trademark matching and retrieval in sports video databases," in *Proc. of ACM International Workshop on Multimedia Information Retrieval (MIR)*, 2007.

[33] T.-K. Kim, K.-Y. K. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *Proc. of CVPR*, 2007.

[34] E. Hayman and J.-O. Eklundh, "Statistical background subtraction for a mobile observer," in *Proc. of ICCV*, 2003.

[35] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," in *Proc. of CVPR*, 1999.

# Gesture Recognition in Ego-Centric Videos using Dense Trajectories and Hand Segmentation

Lorenzo Baraldi[1], Francesco Paci[2], Giuseppe Serra[1], Luca Benini[2,3], Rita Cucchiara[1]

[1]Dipartimento di Ingegneria "Enzo Ferrari"
Università di Modena e Reggio Emilia, Italy
baraldi.lorenzo@gmail.com
{name.surname}@unimore.it

[2]DEI
Univeristà di Bologna, Italy
{f.paci, luca.benini}@unibo.it

[3]Integrated Systems Laboratory
ETH Zürich, Switzerland
lbenini@iss.ee.ethz.ch

## Abstract

*We present a novel method for monocular hand gesture recognition in ego-vision scenarios that deals with static and dynamic gestures and can achieve high accuracy results using a few positive samples. Specifically, we use and extend the dense trajectories approach that has been successfully introduced for action recognition. Dense features are extracted around regions selected by a new hand segmentation technique that integrates superpixel classification, temporal and spatial coherence. We extensively test our gesture recognition and segmentation algorithms on public datasets and propose a new dataset shot with a wearable camera. In addition, we demonstrate that our solution can work in near real-time on a wearable device.*

## 1. Introduction

Ego-centric vision is a paradigm that joins in the same loop humans and wearable devices to augment the subject vision capabilities by automatically processing videos captured with a first-person camera. We are interested in investigating the usage of ego-vision algorithms and devices to enhance new human-machine interfaces that could integrate information from the local environment with web and social media. These interfaces could help users to generate and share content in real-time, and could offer a customized experience, more suited for the user's specific cognitive needs and interests. For instance, ego-vision wearable systems could help understand what visitors of a museum are observing or doing, and determine their degree of interest, collecting data to enhance and customize visitors' experience.

Moreover, the recent growth of computational capability of embedded devices has made possible to exploit wearable and low-power devices as target platforms for ego-centric real-time applications. For this reason, applications and algorithms designed for ego-vision must be suited for portable input and elaboration devices, that often present a more constrained scenario, with different power needs and performance capabilities.

In this paper, we propose a hand gesture recognition approach that could be used in future human-machine interfaces. We take into account both static gestures, in which the meaning of the gesture is conveyed by the hand pose, and dynamic gestures, in which the meaning is given by motion too. It should be noted that gestures are somehow personal. In fact, they can vary from individual to individual and even for the same individual between different instances. Our method uses a monocular camera placed on the user's body to recognizes his gestures. The video stream processing is achieved with an ARM based embedded device that can be worn by users.

Our main contributions are:

- A novel gesture recognition algorithm for ego-vision applications that uses trajectories, appearance features and hand segmentation to classify static and dynamic hand movements, and that can achieve high accuacy results even when trained with a few positive samples.
- A performance analysis of the proposed method on an x86 based workstation and an ARM based embedded device that demonstrates that our algorithm can work in near real-time on a wearable device.

## 2. Related Work

The ego-vision scenario has been addressed only recently by the research community and mainly to understand human activities and to recognize hand regions. Pirsiavash *et al.* [15] detected activities of daily living using an approach that involves temporal pyramids and object detectors tuned for objects appearance during interactions and spatial

reasoning. Sundaram *et al*. [17] proposed instead to use Dynamic Bayesian Networks to recognize activities from low resolution videos, without performing hand detection and preferring computational inexpensive methods. Fathi *et al*. [5] used a bottom-up segmentation approach to extract hand held objects and trained object-level classifier to recognize objects; furthermore they also proposed an activity detection algorithm based on object state changes [4].

Regarding hand detection, Khan *et al*. in [8] studied color classification for skin segmentation. They pointed out how color-based skin detection has many advantages and potentially high processing speed, and demonstrated that Random Forest is one of the best classifiers for skin segmentation. Fathi *et al*. [5] proposed a different approach to hand detection, based on the assumption that background is static in the world coordinate frame, thus foreground objects are detected as to be the moving regions respect to the background. This approach is shown to be a robust tool for skin detection and hand segmentation in indoor environments, even if it performs poorly with more unconstrained scenarios. Li *et al*. [11] proposed a method with sparse feature selection which was shown to be an illumination-dependent strategy. To solve this issue, they trained a set of Random Forests indexed by a global color histogram, each one reflecting a different illumination condition.

To our knowledge, the study of gesture recognition in the ego-centric paradigm has not yet been addressed. Even though not related to ego-vision domain, several approaches to gesture and human action recognition have been proposed. Kim *et al*. [9] extended Canonical Correlation Analysis to measure video-to-video similarity in order to represent and detect actions in video. Lui *et al*. [13, 12] used tensors and tangent bundle on Grassmann manifolds to classify human actions and hand gestures. Sanin *et al*. [16] developed a new and more effective spatio-temporal covariance descriptor to classify gestures in conjunction with a boost classifier. However, all these approaches are not appropriate for the ego-centric perspective, as they do not take into account any of the specific characteristics of this domain, such as fast camera motion, hand presence and background cluttering, as well as the limited computational power of wearable platforms.

## 3. Proposed Method

Gesture recognition systems should recognize both static and dynamic hand movements. Therefore, we propose to describe each gesture as a collection of dense trajectories extracted around hand regions. Feature points are sampled inside and around the user's hands and tracked during the gesture; then several descriptors are computed inside a spatio-temporal volume aligned with each trajectory, in order to capture its shape, appearance and movement at each frame. These descriptors are coded, using the Bag of Words approach and power normalization, in order to obtain the final feature vectors, which are then classified using a linear SVM classifier. A summary of our approach is presented in Figure 1.

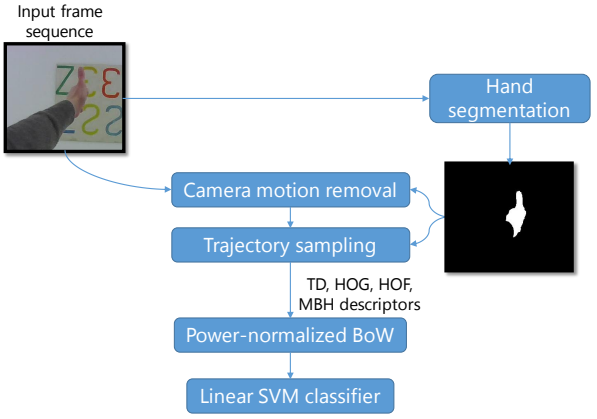



Figure 1: Outline of the proposed Gesture Recognition method.

### 3.1. Camera motion removal

To describe shape, appearance and movement of each trajectory we use the Trajectory descriptor, histograms of oriented gradients, of optical flow, and motion boundary histograms, following [18]. The Trajectory descriptor captures trajectory shape, HOG are based on the orientation of image gradient and encode the static appearance of the region surrounding the trajectory, HOF and MBH are based on optical flow and capture motion information.

In order to remove camera motion, the homography between two consecutive frames is estimated running the RANSAC algorithm on densely sampled features points. SURF features and sample motion vector are extracted from the optical flow to get dense matches between frames.

However, in first-person camera views hands movement is not consistent with camera motion and this generates wrong matches between the two frames. For this reason we introduce a segmentation mask that disregards feature matches belonging to hands. In fact, without the hand segmentation mask, many feature points from the user's hands would become inliers, degrading the homography estimation. As a consequence, the trajectories extracted from the video would be incorrect. Instead, computing an homography using feature points from non-hand regions allows us remove all the camera movements.

### 3.2. Gesture Description

Having removed camera motion between two adjacent frames, trajectories can be extracted. The second frame is warped with the estimated homography, the optical flow be-

tween the first and the second frame is recomputed, and then feature points around the hands of the user are sampled and tracked following what [18] does for human action recognition. Feature points are densely sampled at several spatial scales and tracked using median filtering in a dense optical flow field. In contrast to [18], trajectories are restricted to lie inside and around the user's hands: at each frame the hand mask is dilated, and all the feature points outside the computed mask are discarded.

Then, the spatio-temporal volume aligned with each trajectory is considered, and Trajectory descriptor, HOG, HOF and MBH are computed around it. While HOF and MBH are averaged on five consecutive frames, a single HOG descriptor is computed for each frame. In this way we can better describe how the hand pose changes in time. After this step, we get a variable number of trajectories for each gesture. In order to obtain a fixed size descriptor, the Bag of Words approach is exploited: we train four separate codebooks, one for each descriptor. Each codebook contains 500 visual words and is obtained running the $k$-means algorithm in the feature space.

Since BoW histograms in our domain tend to be sparse, they are power normalized to unsparsify the representation, while still allowing for linear classification. To perform power-normalization [14], the following function is applied to each bin $h_i$:

$$f(h_i) = \text{sign}(h_i) \cdot |h_i|^{\frac{1}{2}} \tag{1}$$

The final feature vector is then obtained by the concatenation of its four power-normalized histograms. Eventually, gestures are recognized using a linear SVM 1-vs-1 classifier.

### 3.3. Hand Segmentation

The proposed gesture recognition approach uses a hand segmentation mask to distinguish between camera and hand motions, and to prune away all the trajectories that do not belong to the user hand. In this way, our descriptor captures hands movement and shape as if the camera was fixed, and disregards the noise coming from other moving regions that could be in the scene.

For computing hand segmentation masks, at each frame we extract superpixels using the SLIC algorithm [3], that performs a $k$-means-based local clustering of pixels in a 5-dimensional space, where color and pixel coordinates are used. Superpixels are represented with several features: histograms in the HSV and LAB color spaces (that have been proved to be good features for skin representation [8]), Gabor filters and a simple histogram of gradients, to discriminate between objects with a similar color distribution.

In order to deal with different illumination conditions we also train a collection of Random Forest classifiers indexed by a global HSV histogram, instead of using a single classifier. Hence, training images are distributed among the classifiers by a *k*-means clustering on the feature space. At test time, the predictions from the five nearest classifier are averaged to make the final prediction.

Furthermore, semantic coherence in time and space is taken into account. Since past frames should affect the prediction for the current frame, a smoothing filter is applied, so that the prediction for each frame is replaced with a combination of the classifier results from past frames. Then, to remove small and isolated pixel groups and also to aggregate bigger connected pixel groups, the GrabCut algorithm is applied to exploit spatial consistency.

## 4. Experimental Results

To compare the performance of the proposed gesture recognition algorithm with existing approaches, we test it on the Cambridge-Gesture database [10], which includes nine hand gesture types performed on a table, under different illumination conditions. To better investigate the effectiveness of the proposed approach in videos taken from the ego-centric perspective and in a museum setting, we also propose a far more realistic and challenging dataset which contains seven gesture classes, performed by five subjects in an interactive exhibition room which functions as a virtual museum. Furthermore, to evaluate the hand segmentation approach, we test it on the publicly available CMU EDSH dataset [11] which consists of three ego-centric videos with indoor and outdoor scenes and large variations of illuminations. We implemented two different versions of our approach, one targeted for x86 based workstations and a lightweight version for ARM based embedded devices. We present performance evaluations on these two implementations and evaluate the accuracy-performance tradeoff of the embedded version.

### 4.1. Gesture Recognition

The Cambridge Hand Gesture dataset contains 900 sequences of nine hand gesture classes. Although this dataset does not contain ego-vision videos it is useful to compare our results to recent gesture recognition techniques. In particular, each sequence is recorded with a fixed camera, placed over one hand, and hands perform leftward and rightward movements on a table, with different poses. The whole dataset is divided in five sets, each of them containing image sequences taken under different illumination conditions. The common test protocol, proposed in [10], requires to use the set with normal illumination for training and the remaining sets for testing, thus we use the sequences taken in normal illumination to generate the BoW codebooks and to train the SVM classifier. Then, we perform the test using the remaining sequences.

| Method | Set1 | Set2 | Set3 | Set4 | Overall |
|---|---|---|---|---|---|
| TCCA [9] | 0.81 | 0.81 | 0.78 | 0.86 | 0.82 |
| PM [13] | 0.89 | 0.86 | 0.89 | 0.87 | 0.88 |
| TB [12] | **0.93** | 0.88 | 0.90 | 0.91 | 0.91 |
| Cov3D [16] | 0.92 | **0.94** | 0.94 | 0.93 | 0.93 |
| **Our method** | 0.92 | 0.93 | **0.97** | **0.95** | **0.94** |

Table 1: Recognition rates on the Cambridge dataset.

(a) *Dislike* gesture

(b) *Point* gesture

Figure 2: Sample gestures from the Interactive Museum dataset.

Table 1 shows the recognition rates obtained with our gesture recognition approach, compared with the ones of tensor canonical correlation analysis (TCCA) [9], product manifolds (PM) [13], tangent bundles (TB) [12] and spatio-temporal covariance descriptors (Cov3D) [16]. Results show that proposed method outperforms the existing state-of-the-art approaches.

We then propose the Interactive Museum dataset, a gesture recognition dataset taken from the ego-centric perspective in a virtual museum environment. It consists of 700 video sequences, all shot with a wearable camera, in an interactive exhibition room, in which paintings and artworks are projected over a wall, in a virtual museum fashion (see figure 2). The camera is placed on the user's head and captures a 800 $\times$ 450, 25 frames per second 24-bit RGB image sequence. In this setting, five different users perform seven hand gestures: *like*, *dislike*, *point*, *ok*, *slide left to right*, *slide right to left* and *take a picture*. Some of them (like the *point*, *ok*, *like* and *dislike* gestures) are statical, others (like the two *slide* gestures) are dynamical. This dataset is very challenging since there is fast camera motion and users have not been trained before recording their gestures, so that each user performs the gestures in a slightly different way, as would happen in a realistic context. We have publicly released our dataset[1].

Since Ego Vision applications are highly interactive, their setup step must be fast (i.e. few positive examples can be acquired). Therefore, to evaluate the proposed gesture recognition approach, we train a 1-vs-1 linear classifier for each user using only two randomly chosen gestures per class as training set. The reported results are the average over 100 independent runs.

[1] http://imagelab.ing.unimore.it/files/ego_virtualmuseum.zip

| User | No segmentation | With segmentation |
|---|---|---|
| Subject 1 | 0.91 | **0.95** |
| Subject 2 | 0.87 | **0.87** |
| Subject 3 | 0.92 | **0.95** |
| Subject 4 | **0.96** | 0.94 |
| Subject 5 | 0.91 | **0.96** |
| Average | 0.91 | **0.93** |

Table 2: Gesture recognition accuracy on the Interactive Museum dataset with and without hand segmentation.

In Table 2 we show the gesture recognition accuracy for each of the five subjects, and we also compare with the ones obtained without the use of the hand segmentation mask for camera motion removal and trajectories pruning. Results show that our approach is well suited to recognize hand gestures in the ego-centric domain, even using only two positive samples per gesture, and that the use of the segmentation mask can improve recognition accuracy.

### 4.2. Hand Segmentation

The CMU EDSH dataset consists of three ego-centric videos (EDSH1, EDSH2, EDSHK) containing indoor and outdoor scenes where hands are purposefully extended outwards to capture the change in skin color. As this dataset does not contain any gesture annotation, we use it to evaluate only the hand segmentation part.

We validate the techniques that we have proposed for temporal and spatial consistency. In Table 3 we compare the performance of the hand segmentation algorithm in terms of F1-measure, firstly using a single Random Forest classifier, and then incrementally adding illumination invariance, the temporal smoothing filter and the spatial consistency technique via the GrabCut algorithm application. Results shows that there is a significant improvement in performance when all the three techniques are used together: illumination invariance increases the performance with respect to the results obtained using only a single random forest classifier, while temporal smoothing and spatial consistency correct incongruities between adjacent frames, prune away small and isolated pixel groups and merge spatially nearby regions, increasing the overall performance.

Then, in Table 4 we compare our segmentation method with different techniques: a video stabilization approach

| Features | EDSH2 | EDSHK |
|---|---|---|
| Single RF classifier | 0.761 | 0.829 |
| II | 0.789 | 0.831 |
| II + TS | 0.791 | 0.834 |
| II + TS + SC | **0.852** | **0.901** |

Table 3: Performance comparison considering Illumination Invariance (II), Temporal Smoothing (TS) and Spatial Consistency (SC).

| Method | EDSH2 | EDSHK |
|---|---|---|
| Hayman and Eklundh [6] | 0.211 | 0.213 |
| Jones and Rehg [7] | 0.708 | 0.787 |
| Li and Kitani [11] | 0.835 | 0.840 |
| **Our method** | **0.852** | **0.901** |

Table 4: Hand segmentation comparison with the state-of-the-art.

based on background modeling [6], a single-pixel color method inspired by [7] and the approach proposed in [11] by Li *et al*., based on a collection of Random Forest classifiers. As can be seen, the single-pixel approach, which basically uses a random regressor trained only using the single pixel LAB values, is still quite effective, even if conceptually simple. Moreover, we observe that the video stabilization approach performs poorly on this dataset, probably because of the large ego-motions these video present. The method proposed by Li *et al*. is the most similar to our approach, nevertheless exploiting temporal and spatial coherence we are able to outperform their results.

### 4.3. Performance Evaluations

We have first implemented and tested our algorithm on an Intel based workstation, with a i7-2600 CPU that runs at 3.40 GHz, and then developed a lightweight version that reaches good performance even on low-power devices. On the workstation implementation we did not perform any code optimization whereas the embedded implementation has been optimized using OpenMP and tested on a Odroid-XU developer board [1]. This board embeds the ARM Exynos 5 SoC, hosting a Quad big.LITTLE ARM processor (Cortex A15 and A7), codename 5410 [2].

To evaluate execution times on both architectures, we divide our algorithm in four modules: the *Trajectory sampling* module, which includes trajectories extraction and description, the *Power-normalized BoW* module, that exploits the Bag of Words approach and power normalization to build the final feature vectors, the *Classification* module, that performs linear SVM classification, and the *Hand Segmentation* module, that runs our segmentation algorithm.

The embedded version has been implemented in C++ and each described module has been optimized, using OpenMP parallel regions. This allows to exploit the computational power of the Exynos processor, that embeds two clusters A15 and A7, that runs from 250 $MhZ$ to 1.6 $GhZ$. The performance evaluation has been done at maximum CPU frequency using the A15 cluster. Figure 3 shows the execution time for each module on both devices.

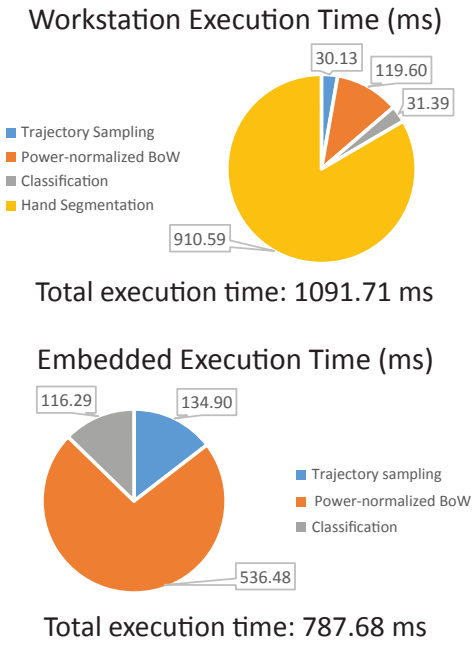



Figure 3: Performance comparison between Workstation and Embedded implementations on 15 frames trajectories.

As can be seen, *Trajectory Sampling*, *Power-normalized BoW* and *Classification* modules tested on the workstation reach around 4x to 5x speedup, compared to the embedded ones. Thus for the embedded implementation we removed the hand segmentation module that has the worst accuracy/performance contribution to the whole algorithm. Result shows that the workstation implementation can elaborate almost 15 frames per second, while the embedded one reaches around 19 $fps$, when the *Hand segmentation* module is disabled. This is a good result that means that we reach a near real-time frame rate, for both the two versions. Moreover comparing these results with Figure 3 and Table 2 it is possible to correlate the accuracy loss, that is around 2%. Hence we trade off a modest accuracy loss for being able to reach near real time performance.

## 5. Conclusion

We described a novel approach to hand gesture recognition in ego-centric videos. Our work is motivated by the increasing interest in ego-centric human-machine interfaces and by the growth of computational capabilities of wearable devices, which encourages the development of real-time computer vision algorithms. We presented a model that can deal with static and dynamic gestures and can achieve high accuracy results even when trained with a few positive samples. Our gesture recognition and hand segmentation results outperform the state-of-the-art approaches on Cambridge Hand Gesture and CMU EDSH datasets. Finally, we demonstrated that our algorithm can work in near real-time on a wearable Odroid board.

## References


[1] Odroid-XU developement board by Hardkernel. http://www.hardkernel.com. 5

[2] Samsung Exynos5 5410 ARM SoC. http://www.samsung.com/global/business/semiconductor/minisite/Exynos/products5octa_5410.html. 5

[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012. 3

[4] A. Fathi and J. M. Rehg. Modeling actions through state changes. In *Proc. of CVPR*, 2013. 2

[5] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *Proc. of CVPR*, 2011. 1, 2

[6] E. Hayman and J.-O. Eklundh. Statistical background subtraction for a mobile observer. In *Proc. of ICCV*, 2003. 4, 5

[7] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. 1999. 4, 5

[8] R. Khan, A. Hanbury, and J. Stoettinger. Skin detection: A random forest approach. In *Proc. of ICIP*, 2010. 2, 3

[9] T.-K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1415–1428, 2009. 2, 3, 4

[10] T.-K. Kim, K.-Y. K. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *Proc. of CVPR*, 2007. 3

[11] C. Li and K. M. Kitani. Pixel-level hand detection in ego-centric videos. In *Proc. of CVPR*, 2013. 2, 3, 4, 5

[12] Y. M. Lui and J. R. Beveridge. Tangent bundle for human action recognition. In *In proc. of Automatic Face & Gesture Recognition and Workshops*, 2011. 2, 3, 4

[13] Y. M. Lui, J. R. Beveridge, and M. Kirby. Action classification on product manifolds. In *Proc. of CVPR*, 2010. 2, 3, 4

[14] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. of ECCV*, 2010. 3

[15] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Proc. of CVPR*, 2012. 1

[16] A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell. Spatio-temporal covariance descriptors for action and gesture recognition. In *Proc. of Workshop on Applications of Computer Vision*, 2013. 2, 3, 4

[17] S. Sundaram and W. W. M. Cuevas. High level activity recognition using low resolution wearable vision. In *Proc. of CVPR*, 2009. 1

[18] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *Proc. of CVPR*, 2011. 2, 3

# Hand Segmentation for Gesture Recognition in EGO-Vision

Giuseppe Serra
giuseppe.serra@unimore.it

Marco Camurri
marco.camurri@yahoo.it

Lorenzo Baraldi
baraldi.lorenzo@gmail.com

Michela Benedetti
michela.benedetti89@gmail.com

Rita Cucchiara
rita.cucchiara@unimore.it

Dipartimento di Ingegneria "Enzo Ferrari"
Università degli Studi di Modena e Reggio Emilia
Via Vignolese 905, 41125 Modena - Italy

## ABSTRACT

Portable devices for first-person camera views will play a central role in future interactive systems. One necessary step for feasible human-computer guided activities is gesture recognition, preceded by a reliable hand segmentation from egocentric vision. In this work we provide a novel hand segmentation algorithm based on Random Forest superpixel classification that integrates light, time and space consistency. We also propose a gesture recognition method based Exemplar SVMs since it requires a only small set of positive sampels, hence it is well suitable for the egocentric video applications. Furthermore, this method is enhanced by using segmented images instead of full frames during test phase. Experimental results show that our hand segmentation algorithm outperforms the state-of-the-art approaches and improves the gesture recognition accuracy on both the publicly available EDSH dataset and our dataset designed for cultural heritage applications.

## Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: [Segmentation, Scene Analysis, Applications]; I.4.6 [**Segmentation**]: [Pixel classification]

## Keywords

Hand segmentation, Gesture Recognition, Exemplar SVM, Random Forest, Ego-vision

## 1. INTRODUCTION AND RELATED WORK

The recent progresses in sensor development and mobile computing, and the increasing availability of wearable computers (e.g. Google Glass and Vuzix SmartGlass) has raised the interest of the research community toward the new field of egocentric vision.


*IMMPD'13,* October 22 2013, Barcelona, Spain.

http://dx.doi.org/10.1145/2505483.2505490.

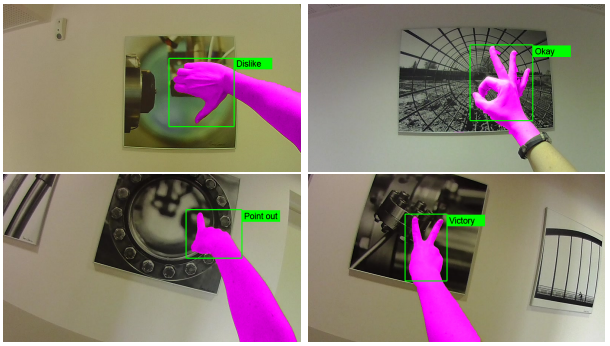



**Figure 1: Sample results of the proposed hand segmentation and gesture recognition algorithm.**

Egocentric vision, or ego-vision, is a paradigm that joints in the same loop human and wearable devices to augment the human vision capabilities by automatically process videos acquired with a first-person camera. Initial efforts have been made on the definition of methodologies to automatically understand human actions, objects and people interactions in egocentric vision. Systems that perceive what you perceive, see what you see and understand what you can understand or more will be used for many human augmentation applications. Some examples could be systems which recognize people around you, which understand dangerous situations, and provide assistance for activities such as surgery, sport, entertainment and more. The egocentric paradigm presents many new challenges, such as background clutter [14], large ego-motion and extreme transitions in lighting, but it also has some unique advantages. Egocentric videos are recorded from the same person over a continuous temporal space, there is no need to place multiple fixed cameras on the environment; furthermore objects and gestures are less likely to be occluded.

In this paper, we are interested in exploring the usage of ego-vison devices for cultural heritage domain: the museum experience, for example, could be enhanced by developing innovative human-machine interfaces such as new kinds of self-guided tour that can integrate information from the local

environment, Web and social medias. Furthermore these interfaces can help users to generate and share content in real time. In this scenario, hand detection and gesture recognition play a fundamental role, since this kind of applications should substitute other physical controller devices. Since gestures are strictly related to a specific scenario or application, it necessary to build a set of new classifiers using information gathered during a fast setup phase involving the user.

This problem, in ego-vision scenario, has been addressed only recently by the research community. Khan and Stoettinger in [8] studied color classification for skin segmentation and pointed out how color-based skin detection has many advantages, like potentially high processing speed, invariance against rotation, partial occlusion and pose change. The authors tested Bayesian Networks, Multilayers Perceptrons, AdaBoost, Naive Bayes, RBF Networks and Random Forest. They demonstrated that Random Forest classification obtains the highest F-score among all the other techniques. Fathi et al. [4] proposed a different approach to hand detection, exploiting the basic assumption that background is static in the world coordinate frame. Thus foreground objects are detected as to be the moving region respect to the background. An initial panorama of the background is required to discriminate between background and foreground regions: this is achieved by fitting a fundamental matrix to dense optical flow vectors. This approach is shown to be a robust tool for skin detection for hand segmentation in a limited indoor environment but it performs poorly with more unconstrained scenes. Li and Kitani [9] provide an historical overview about approaches for detecting hands from moving cameras. They define three categories: local appearance-based detection, global appearance-based detection, where a global template of hand is needed, and motion-based detection, which is based on the hypothesis that hands and background have different motion statistics. Motion-based detection approach requires no supervision nor training. On the other hand, this approach eventually identifies as hand an object manipulated by the user, since it moves together his hands. In addition they proposed a model with sparse feature selection which was shown to be an illumination-dependent strategy. To solve this issue, they trained a set of random forests indexed by a global color histogram, each one reflecting a different illumination condition. Recently Bagdanov et al. [2] propose a method to predict the status of the user hand by jointly exploiting depth and RGB imagery.

All the presented previous works present good characteristics, but lack of generality, since they take into account only few aspects to model user hand appearance and they are not integrated with a gesture recognition system. In this paper we present a novel method for hand segmentation and gesture recognition that can be used as basis for ego-vision applications. Hand detection is based on Random Forest classifiers learned by color and gradient features which are computed on superpixels. In order to improve the detection accuracy we present two strategies that incorporate temporal and spatial coherence: temporal smoothing and spatial consistency. Hand detection masks is then used as input for the gesture recognition step in order to reduce misclassification. We propose to use Exemplar SVMs to recognize gestures since it requires a only small set of positive samples, hence it well suitable for the ego-vision application domain. Experimental results show that our hand segmentation algorithm outperforms the state-of-the-art approaches on publicly available EDSH dataset [9] and demonstrate that segmentation step improves the gesture recognition accuracy. Fig. 1 shows sample results of the proposed hand segmentation and gesture recognition algorithm.
Here, we mainly provide our contributions:

- we define a novel hand segmentation algorithm that differently from [9], uses a superpixel features, and integrate not only illumination invariance but also temporal and spatial consistency improves the state-of-the-art results in the publicly available EDSH dataset.

- we develop of a gesture recognition algorithm based on Exemplar SVM technique that, even with a few positive samples, permits to reach competitive results.

# 2. METHOD OVERVIEW

## 2.1 Hand segmentation

Ego-vision applications require a fast and reliable segmentation of the hands; thus we propose to use random forest classifiers, as they are known to efficiently work even with large inputs [3]. Since using a per-pixel basis in label assignment has show to be inefficient [7], we adopt segmentation method which assign labels to superpixels, as suggested in [16]. This allows a complexity reduction of the problem and also gives better spatial support for aggregating features that could belong to the same object.

To extract superpixels for every frames we use the Simple Linear Iterative Clustering (SLIC) algorithm, proposed in [1] as memory efficient and highly accurate segmentation method. The SLIC super-pixel segmentation algorithm is a k-means-based local clustering of pixels in a 5D space, where Lab color values and pixel coordinates are used. A parallel implementation of the SLIC super-pixel algorithm is available in [13].

We represent superpixels by features to encode color and gradient information. As pointed out by previous works, HSV and LAB color spaces have been proven to be robust for skin detection. In particular, we describe each superpixel with mean and covariance matrix of its pixel values, and a 32-bin color histogram both in HSV and Lab color spaces. To discriminate between objects with a similar color distribution of skin we include following gradient information: Gabor feature obtained with 27 filters (nine orientations and three different scales: $7\times7$, $13\times13$, $19\times19$) and a simple histogram of gradients with nine bins.

### 2.1.1 *Illumination invariance, Temporal and Spatial Consistency*

In order to deal with different illumination conditions we train a collection of random forest classifiers indexed by a global HSV histogram with 32 bins, as described in [9]. Hence, training images are distributed among the classifiers by a k-means clustering on the feature space. By using a histogram over all three channels of the HSV color space, each scene cluster encodes both the appearance of the scene and its illumination. Intuitively, it models the fact that hands viewed under similar global appearance will share a similar distribution in the feature space. Given a feature vector $\mathbf{l}$ of a superpixel $\mathbf{s}$ and a global appearance feature $\mathbf{g}$, the posterior distribution of $\mathbf{s}$ is computed by marginalizing over different scenes $c$:

**Figure 2: Comparison before (left image) and after (right image) Temporal smoothing.**

**Figure 3: Comparison before (left image) and after (right image) Spatial Consistency.**

$$P(\mathbf{s}|\mathbf{l},\mathbf{g}) = \sum_{c} P(\mathbf{s}|\mathbf{l},c)P(c|\mathbf{g}), \tag{1}$$

where $P(\mathbf{s}|\mathbf{l},c)$ is the output of a global appearance-specific classifier and $P(c|\mathbf{g})$ is a conditional distribution of a scene $c$ given a global appearance feature $\mathbf{g}$. In test phase, the conditional $P(c|\mathbf{g})$ is approximated using an uniform distribution over the five nearest models learned at training. It is important to underline that the optimal number of classifiers depends on the characteristics of the dataset: a training dataset with several different illumination conditions, taken both inside and outside, will need an higher number of classifiers than one taken indoor.

In addition to [9], we model the hand appearance not only considering illumination variations, but also including semantic coherence in time and space.

### 2.1.2 *Temporal smoothing*

We exploit temporal coherence to improve the foreground prediction of a pixel in a frame by a weighted combination of its previous frames, since past frames should affect the results prediction for the current frame.

The smoothing filter for a pixel $\mathbf{x}_t^i$ of a frame $t$ (inspired by [10]) can thus be defined as follows:

$$\begin{aligned} P(\mathbf{x}_t^i = 1) = \sum_{k=0}^{d} w_k \big( & P(\mathbf{x}_t^i = 1|\mathbf{x}_{t-k}^i = 1) \cdot \\ & \cdot P(\mathbf{x}_{t-k}^i = 1|\mathbf{l_{t-k}}, \mathbf{g_{t-k}}) + P(\mathbf{x}_t^i = 1|\mathbf{x}_{t-k}^i = 0) \cdot \\ & \cdot P(\mathbf{x}_{t-k}^i = 0|\mathbf{l_{t-k}}, \mathbf{g_{t-k}}) \big) \end{aligned} \tag{2}$$

where $P(\mathbf{x}_{t-k}^i = 1|\mathbf{l_{t-k}}, \mathbf{g_{t-k}})$ is the posterior probability that a pixel in frame $t-k$ is marked as hand part and $d$ is a number of past frames used. This likelihood can be defined as the probability $P(\mathbf{s}|\mathbf{l_{t-k}}, \mathbf{g_{t-k}})$, being $\mathbf{x}_t^i$ part of $\mathbf{s}$. In the same way, $P(\mathbf{x}_{t-k}^i = 0|\mathbf{l_{t-k}}, \mathbf{g_{t-k}})$ is defined as the probability $1 - P(\mathbf{s}|\mathbf{l}, \mathbf{g_{t-k}})$.

While $P(\mathbf{x}_t^i = 1|\mathbf{x}_{t-k}^i = 1)$ and $P(\mathbf{x}_t^i = 1|\mathbf{x}_{t-k}^i = 0)$ are prior probabilities estimated from the training set as follows:

$$P(\mathbf{x}_t^i = 1|\mathbf{x}_{t-k}^i = 1) = \frac{\#(\mathbf{x}_t^i = 1, \mathbf{x}_{t-k}^i = 1)}{\#(\mathbf{x}_{t-k}^i = 1)}$$

$$P(\mathbf{x}_t^i = 1|\mathbf{x}_{t-k}^i = 0) = \frac{\#(\mathbf{x}_t^i = 1, \mathbf{x}_{t-k}^i = 0)}{\#(\mathbf{x}_{t-k}^i = 0)}$$

where $\#(\mathbf{x}_{t-k}^i = 1)$ and $\#(\mathbf{x}_{t-k}^i = 0)$ are the number of times in which $\mathbf{x}_{t-k}$ belongs or not to a hand region, respectively; $\#(\mathbf{x}_t^i = 1, \mathbf{x}_{t-k}^i = 1)$ is the number of times that two pixels at the same location at frame $t$ and $t-k$ belong to a hand part; similarly, $\#(\mathbf{x}_t^i = 1, \mathbf{x}_{t-k}^i = 0)$ is the number of times that a pixel in frame $t$ belongs to a hand part and pixel in the same position in frame $t-k$ does not belong to a hand region. Figure 2 shows an example where temporal smoothing deletes blinking regions (i.e. the tea box brand and jar shadows on the right).

### 2.1.3 *Spatial Consistency*

Given pixels elaborated by the previous steps, we want to exploit spatial consistency to prune away small and isolated pixel groups that are unlikely to be part of hand regions and also aggregate bigger connected pixel groups.

For every pixel $\mathbf{x}$, we extract its posterior probability $P(\mathbf{x}_t^i)$ and use it as input for the GrabCut algorithm [15]. Each pixel with $P(\mathbf{x}_t^i) \geq 0.5$ is marked as foreground, otherwise it's considered as part of background. After the segmentation step, we discard all the small isolated regions that have an area of less than 5% of the frame and we keep only the three largest connected components.

In Figure 3 an example with and without applying the Spatial Consistency method is depicted; notice this technique allows to better aggregate superpixels that are near the principal blob region.

## 2.2 Hand status recognition

Given a subregion of the image space whose pixel are likely to be a segmented hand, we want now to recognize the hand configuration, picking from a finite set of possible gestures. Thus, we use the Exemplar SVM (ESVM) approach proposed by Malisiewicz et al. [11]. This method involves two steps: first, for each class an independent training of a finite and small set of positive examples (the "exemplars") versus a huge set of negative examples is performed. After the training stage each independent SVM classifier is tuned to detect its specific positive exemplar, thus a further step to aggregate classifiers of each class is necessary. In the second stage, since the outputs of each classifier are not comparable, a calibration by fitting a probability distribution to a held-out set of negative and positive samples is performed, as described in [12]. The calibration can be interpreted as a simple rescaling and shifting of the decision boundary, and does not affect the ordering of the score, allowing to compare the outputs of multiple independently-trained Exemplar-SVMs. Thus, multiple Exemplar-SVMs can be combined to provide a model for each class.

Since Ego Vision applications are highly interactive, their setup step must be fast (i.e. few positive examples can be acquired) but they allow an *a priori* massive collection of negative examples (e.g. for a museum application, a large footage without hands can be early acquired to train the classifier). Thus, ESVMs is well suitable for our application and was preferred to Latent SVM (LSVM) proposed by Felzenszwalb et al. [5] that have shown good performance in image classi-

| Features | EDSH_2 | EDSH_kitchen |
|---|---|---|
| HSV | 0.752 | 0.801 |
| + LAB | 0.754 | 0.808 |
| + LAB hist. | 0.755 | 0.823 |
| + HSV hist. | 0.755 | 0.823 |
| + Grad hist. | 0.758 | 0.828 |
| + Gabor | **0.761** | **0.829** |

**Table 1: Performance by incrementally adding new features.**

fication competitions. Although LSVM could well model the hand deformability property, it's more complex, it requires a more balanced set of negative and positive examples during the training stage and exhibits similar performance w.r.t. ESVM method [11].

# 3. EXPERIMENTAL RESULTS

To evaluate the performance of proposed method we tested it on two datasets: EDSH and EGO-HSGR. The recent publicly available EDSH dataset [9] consists in egocentric videos acquired to analyze performance of several hand detection methods. It consists in three videos (EDSH_1, used as train video, and EDSH_2 and EDSH_kitchen used as test videos) that contain indoor and outdoor scenes with large variations of illumination, mild camera motion induced by walking and climbing stairs. All videos are recorded at a resolution of 720p and a speed of 30FPS. The dataset includes segmentation masks of hands, but it is not comprehensive of gesture annotations.

In order to analyze the performance of our method to recognize gestures, we generated a new dataset which contains 12 videos of indoor scenes (EGO-HSGR); it includes segmentation masks and gesture annotations. Videos have been recorded with a Panasonic HX-A10 Wearable Camcorder at a resolution of $800 \times 450$ with a 25FPS in two different locations: a library and department's exhibition area.

The aim of this dataset is to reproduce an environment similar to a museum for human and object interaction: paintings and posters are hung on the walls, true masterpieces or either its virtual images; the visitor walks and sometimes stops in front of an object of interest performing some gestures to interact with next generation wearable devices. We identify five different gestures that are used commonly: *point out*, *like*, *dislike*, *ok* and *victory*. These can be associated to different action or used for record social experience. Fig. 4 shows some frame examples.

To evaluate performance of our pixel-level hand detector a subset of six videos are used (three for training and two for testing). Segmentation masks are provided every 25 frames for a total of 700 annotations. For gesture analysis we extract all the keyframes and we manually annotated them distinguishing between gestures. The F-score (harmonic mean of the precision and recall rate) is used to quantity hand detection performance, while gesture recognition is evaluated in terms of mAP (mean Average Precision).

## 3.1 Features performance

First, we examine the effectiveness of our features to discriminate between hand and non-hand superpixels. Table 1 shows performance in terms of F-measure on EDSH dataset with different feature combinations: firstly we describe each superpixel with mean and covariance matrix of its pixel values in HSV color space, then we do the same using LAB color space and we add color histograms. Lastly, we include a histogram of gradients and Gabor feature. In order to analyze how visual features impact on the performance, in this experiment we do not include the temporal and spatial context information by using a single random forest classifier. Note that although color information plays a fundamental role for hand detection, some ambiguities between hands and other similar colored object still remain; these can be reduced by adding features based on gradient histograms. In fact, the usage of the full descriptor slightly improves the performance.

| Features | EDSH_2 | EDSH_kitchen |
|---|---|---|
| II | 0.789 | 0.831 |
| II + TS | 0.791 | 0.834 |
| II + TS + SC | **0.852** | **0.901** |

**Table 2: Performance comparison considering Illumination Invariance (II), Time Smoothing (TS) and Spatial Consistency (SC).**

| | EDSH_2 | EDSH_kitchen |
|---|---|---|
| Hayman et al. [6] | 0.211 | 0.213 |
| Jones et al. [7] | 0.708 | 0.787 |
| Li et al. [9] | 0.835 | 0.840 |
| **Our method** | **0.852** | **0.901** |

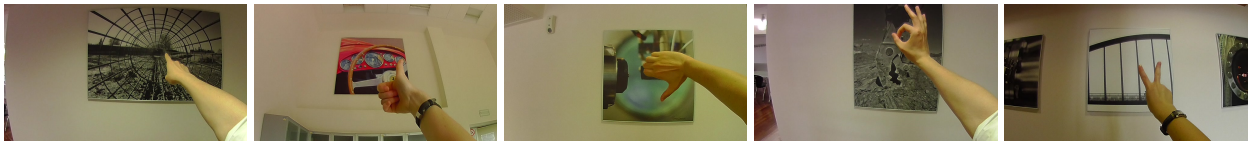
**Table 3: Hand segmentation comparison with the state-of-the-art.**

## 3.2 Temporal Smoothing and Spatial Consistency

In this experiment we validate the proposed techniques that take into account illumination variations, time dependence and spatial consistency. Table 2 shows the F-measure scores obtained on EDSH dataset incrementally adding Illumination Invariance (II), Time Smoothing (TS) and Spatial Consistency (SC). Note that there is a significant improvement in performance when all these three techniques are applied together. In particular, illumination invariance substantially increases the performance with respect to results obtained using only visual features and a single random forest classifier, while the improvement introduced by temporal smoothing is less pronounced. The main contribution is given by Spatial Consistency, that prunes away small and isolated pixel groups and merge spatially nearby regions, increasing the F-measure score of about six percentage points. The proposed technique is also tested in our EGO-HSGR dataset obtaining an F-measure score of 0.908 and 0.865 for the EGO-HSGR_4 and EGO-HSGR_5 videos.

## 3.3 Comparison to related methods

In Table 3 we compare our results to several approaches on EDSH dataset: a single-pixel color approach inspired by [7], a video stabilization approach based on background modeling using affine alignment of image frames inspired by [6] and an approach based on random forest, proposed by [9]. The single-pixel approach is a random regressor trained only using single-pixel LAB color values. The background modeling approach aligns sequences of 15 frames estimating



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

**Figure 4: Our dataset consists of five gestures: a) point out; b) like; c) dislike; d) ok; e) victory.**

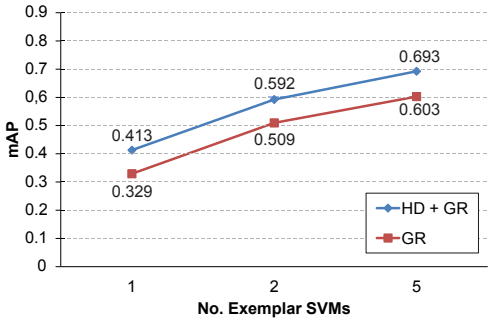



**Figure 5: Mean Average Precision for different number of trained Exemplar SVMs.**

| Gestures | GR | HD + GR |
|---|---|---|
| Dislike | 0.626 | 0.761 |
| Point out | 0.382 | 0.458 |
| Like | 0.817 | 0.754 |
| Ok | 0.698 | 0.951 |
| Victory | 0.490 | 0.541 |

**Table 4: Average precision with five trained Exemplar SVMs per Gesture.**

their mutual affine transformations; pixels with high variance are considered to be foreground hand regions. As can be seen, although the single-pixel approach is conceptually simple, is still quite effective. In addition, we observe that the low performance of the video stabilization approach is due to large ego-motion because the camera is worn by the user. The method recently proposed by [9] is more similar to our approach, but the use of superpixels, the selection of a new set of local features and the introduction of temporal and spatial consistency allow us to outperforms that results.

### 3.4 Hand Recognition

In order to evaluate the use of Exemplar SVM for gesture recognition we test our approach using a different number of trained classifiers on EGO-HSGR dataset. Since Ego-vision scenario requires a fast initial setup for the user, we analyze our approach with a very few positive samples. Figure 5 shows the mean Average Precision in two different settings: we apply our gesture recognition algorithm based on Exemplar SVM on the dataset frames directly (GR); we use the Exemplar SVM on the same frames processed by our hand segmentation algorithm (HD + GR). As expected, the mean Average Precision increases proportionally with the number of trained Exemplars in both settings. The performance obtained using our hand segmentation approach outperforms the gesture recognition without hand segmentation. In Table 4 we present the Average Precision per gesture obtained with five trained Exemplar SVM. Notice that using our hand detection technique provides a gain in performance for all gestures, except *Like*. This is due to fact *Like* gesture is more sensitive to erroneous hand segmentation that negatively effects the recognition step.

## 4. CONCLUSION

The work in this paper gives some initial but very promising results for the feasibility in adopting ego-vision to recognize human actions by first-person camera view. The proposed approach shows interesting results in term of accuracy for hand segmentation and gesture recognition which are novel in the panorama of multimedia-oriented computer vision dealing with the new paradigm of ego-vision.

Although the problem could become very challenging for a computer vision approach (e.g. cameras are not fixed and are freely moving, quality of images are poor and noisy due to the wearable camera), it can open new interesting scenarios for multimedia applications. For example user actions can be self-interpreted and integrated, since sensors are directly attached to the user.

## 5. REFERENCES


[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.

[2] A. D. Bagdanov, A. Del Bimbo, L. Seidenari, and L. Usai. Real-time hand status recognition from rgb-d imagery. In *Proc. of International Conference on Pattern Recognition (ICPR)*, 2012.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] A. Fathi, X. Ren, and J. Rehg. Learning to recognize objects in egocentric activities. In *Proc. of CVPR*, 2011.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[6] E. Hayman and J. Eklundh. Statistical background subtraction for a mobile observer. In *Proc. of ICCV*, 2003.

[7] M. Jones and J. Rehg. Statistical color models with application to skin detection. In *Proc. of CVPR*, 1999.

[8] R. Khan, A. Hanbury, and J. Stoettinger. Skin detection: A random forest approach. In *Proc. of ICIP*, 2010.

[9] C. Li and K. M. Kitani. Pixel-level hand detection in ego-centric videos. In *Proc. of CVPR*, 2013.

[10] K.-H. Liu, M.-F. Weng, C.-Y. Tseng, Y.-Y. Chuang, and M.-S. Chen. Association and temporal rule mining for post-filtering of semantic concept detection in video. *IEEE Transactions on Multimedia*, 10(2):240–251, 2008.

[11] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[12] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[13] C. Y. Ren and I. Reid. gslic: a real-time implementation of slic superpixel segmentation. Technical report, University of Oxford, Department of Engineering Science, 2011.

[14] X. Ren and C. Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *Proc. of CVPR*, 2010.

[15] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[16] J. Tighe and S. Lazebnik. Superparsing - scalable nonparametric image parsing with superpixels. *International Journal of Computer Vision*, pages 329–349, 2013.

# Bibliography


[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.

[2] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[3] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action Recognition by Dense Trajectories. In *Proc. of CVPR*, 2011.

[4] Alejandro Betancourt, Pietro Morerio, Carlo S. Regazzoni, and Matthias Rauterberg. The evolution of first person vision methods: A survey. URL http://arxiv.org/abs/1409.1484.

[5] Steve Mann. 'wearcam'(the wearable camera): personal imaging systems for long-term use in wearable tetherless computer-mediated reality and personal photo/videographic memory prosthesis. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pages 124–131. IEEE, 1998.

[6] WW Mayol, BJ Tordoff, and David W Murray. Wearable visual robots. *Personal and Ubiquitous Computing*, 6(1):37–48, 2002.

[7] Walterio W Mayol, Andrew J Davison, Ben J Tordoff, and David W Murray. Applying active vision and slam to wearables. In *Robotics Research*, pages 325–334. Springer, 2005.

[8] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. Sensecam: A retrospective memory aid. In *UbiComp 2006: Ubiquitous Computing*, pages 177–193. Springer, 2006.

[9] Mark Blum, Alex Pentland, and Gerhard Troster. Insense: Interest-based life logging. *IEEE MultiMedia*, 13(4):40–48, 2006.

[10] WW Mayol, B Tordoff, and DW Murray. On the positioning of wearable optical devices. Technical report, Technical Report OUEL224101. Oxford University, 2001.

[11] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1371–1375, 1998.

[12] Bernt Schiele and Alex Pentland. Attentional objects for visual context understanding. In *SUBMITTED TO ISWC'99*, 1999.

[13] Naohiko Kohtake, Jun Rekimoto, and Yuichiro Anzai. Infostick: an interaction device for inter-appliance computing. In *Handheld and ubiquitous computing*, pages 246–258. Springer, 1999.

[14] Hisashi Aoki, Bernt Schiele, and Alex Pentland. Realtime personal positioning system for a wearable computer. In *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*, pages 37–43. IEEE, 1999.

[15] Thad Starner, Jake Auxier, Daniel Ashbrook, and Maribeth Gandy. The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *Wearable computers, the fourth international symposium on*, pages 87–94. IEEE, 2000.

[16] Wasinee Rungsarityotin and Thad E Starner. Finding location using omnidirectional video on a wearable computing platform. In *Wearable Computers, The Fourth International Symposium on*, pages 61–68. IEEE, 2000.

[17] Brian Clarkson and Alex Pentland. Unsupervised clustering of ambulatory audio and video. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3037–3040. IEEE, 1999.

[18] How the americans will travel 2015. http://tourism-intelligence.com.

[19] Economic Impact of Travel & Tourism 2014. World Travel and Tourism Council, 2014.

[20] Alar Kuusik, Sylvain Roche, Frédéric Weis, et al. Smartmuseum: Cultural content recommendation system for mobile users. In *ICCIT'09: Fourth International Conference on Computer Sciences and Convergence Information Technology*, pages 477–482, 2009.

[21] Tsvi Kuflik, Oliviero Stock, Massimo Zancanaro, Ariel Gorfinkel, Sadek Jbara, Shahar Kats, Julia Sheidin, and Nadav Kashtan. A visitor's guide in an active museum: Presentations, communications, and reflection. *Journal on Computing and Cultural Heritage (JOCCH)*, 3(3):11, 2011.

[22] Flavia Sparacino. The museum wearable: real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences. In *In Proc. of Museums and the Web*, pages 17–20, 2002.

[23] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[24] S Marčelja. Mathematical description of the responses of simple cortical cells*. *JOSA*, 70(11):1297–1300, 1980.

[25] Anna Bosch, Xavier Muñoz, and Robert Martí. Which is the best way to organize/classify images by content? *Image and vision computing*, 25(6):778–791, 2007.

[26] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20 (3):273–297, 1995.

[27] Yasemin Altun, Ioannis Tsochantaridis, Thomas Hofmann, et al. Hidden markov support vector machines. In *ICML*, volume 3, pages 3–10, 2003.

[28] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, and Yoram Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(9), 2005.

[29] R. Khan, A. Hanbury, and J. Stoettinger. Skin detection: A random forest approach. In *Proc. of ICIP*, 2010.

[30] M. Jones and J Rehg. Statistical color models with application to skin detection. In *Proc. of CVPR*, 1999.

[31] E. Hayman and J. Eklundh. Statistical background subtraction for a mobile observer. In *Proc. of ICCV*, 2003.

[32] A. Fathi, Xiaofeng Ren, and J.M. Rehg. Learning to recognize objects in egocentric activities. In *Proc. of CVPR*, 2011.

[33] C. Li and K. M. Kitani. Pixel-level hand detection in ego-centric videos. In *Proc. of CVPR*, 2013.

[34] Andrew D. Bagdanov, Alberto Del Bimbo, Lorenzo Seidenari, and Lorenzo Usai. Real-time hand status recognition from rgb-d imagery. In *Proc. of International Conference on Pattern Recognition (ICPR)*, 2012.

[35] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[36] Joseph Tighe and Svetlana Lazebnik. Superparsing - scalable nonparametric image parsing with superpixels. *International Journal of Computer Vision*, pages 329–349, 2013.

[37] Carl Yuheng Ren and Ian Reid. gslic: a real-time implementation of slic superpixel segmentation. Technical report, University of Oxford, Department of Engineering Science, 2011.

[38] Ken-Hao Liu, Ming-Fang Weng, Chi-Yao Tseng, Yung-Yu Chuang, and Ming-Syan Chen. Association and temporal rule mining for post-filtering of semantic concept detection in video. *IEEE Transactions on Multimedia*, 10(2):240–251, 2008.

[39] Andres Sanin, Conrad Sanderson, Mehrtash T Harandi, and Brian C Lovell. Spatio-temporal covariance descriptors for action and gesture recognition. In *Proc. of Workshop on Applications of Computer Vision*, 2013.

[40] Yui Man Lui, J Ross Beveridge, and Michael Kirby. Action classification on product manifolds. In *Proc. of CVPR*, 2010.

[41] Yui Man Lui and J Ross Beveridge. Tangent bundle for human action recognition. In *In proc. of Automatic Face & Gesture Recognition and Workshops*, 2011.

[42] Tae-Kyun Kim and Roberto Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1415–1428, 2009.

[43] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. Springer, 2003.

[44] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. of ECCV*, 2010.

[45] Tae-Kyun Kim, Kwan-Yee Kenneth Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *Proc. of CVPR*, 2007.

[46] Thorsten Joachims. SVM$^{hmm}$ – Sequence Tagging with Structural Support Vector Machines. http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html.

[47] Nicholas Nethercote and Julian Seward. Valgrind: a framework for heavyweight dynamic binary instrumentation. In *ACM Sigplan Notices*, volume 42, pages 89–100. ACM, 2007.