Order number : 4402

UNIVERSITY OF BORDEAUX I

DOCTORAL SCHOOL OF MATHEMATICS AND COMPUTER SCIENCE

Indexing of Activities in Wearable Videos : Application to Epidemiological Studies of Aged Dementia

---

by **Svebor KARAMAN**

A dissertation thesis submitted to the University Bordeaux 1 in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (PhD)

IN COMPUTER SCIENCE

Defended the : 12th December 2011

After the reviews of :

Mr Atilla Baskurt	Professor, INSA Lyon
Mr Alberto Del Bimbo	Professor, University of Firenze

The dissertation is approved by the following members of the Final Oral Committee :

Ms Régine André-Obrecht	Professor, University Paul Sabatier	Committee chair
Mr Atilla Baskurt	Professor, INSA Lyon	Thesis referee
Ms Jenny Benois-Pineau	Professor, University Bordeaux 1	Thesis supervisor
Mr Jean-Pierre Cocquerez	Professor, UTC	Examiner
Mr Rémi Mégret	Lecturer, IPB	Thesis co-supervisor
Mr Yann Gaëstel	Doctor of Science, University Bordeaux 2	Invited member

– 2011 –

A mes parents,

A Yasmine

Thanks

First of all, I would like to thank my advisors Jenny Benois-Pineau and Rémi Mégret for their support during my thesis. Their ideas and tremendous support had a major influence on this thesis. They have taught me how to appreciate the good scientific work that helps other researchers to build on it.

I would also like to thank my reviewers. It is an honor for me that Pr. Alberto Del Bimbo and Pr. Attila Baskurt have accepted to review my PhD thesis.

I am truly thankful to all the IMMED project partners (Régine André-Obrecht, Julien Pinquier, Patrice Guyot, Vladislavs Dovgalecs, Hazem Wannous, Yann Gaëstel, Jean-François Dartigues) for the inspiring IMMED meeting we had.

I would like to thank the healthy volunteers and the patients who have contributed to the recording of this unique corpus of videos.

Many thanks to my office colleagues (Daniel Szolgay, Rémi Vieux, Hugo Boujut, Claire Morand, Laétitia Letoupin) for making the day-to-day life at lab such a pleasure.

I would also like to thank the other members of the team (Aurélie Bugeau, Jean-Philippe Domenger, Ronan Sicre, Christian Kaes, Yifan Zhou, Mathieu Brulin) for the interesting talks we may had, and colleagues of other teams in the lab (Rémi Laplace, Thomas Rocher, Renaud Tabary, Vincent Filou, Damien Dubernet, Cyril Cassagne, Nicolas Aucouturier) for the talks and laughs we have shared.

I would like to thank my friends (Elodie, Alexiane, Maxime, Yasmine) who have supported me during this thesis and especially for the writing of this manuscript.

I thank my parents, grandparents, brother and sister for always being here for me.

Résumé

Le travail de recherche de cette thèse de doctorat s'inscrit dans le cadre du suivi médical des patients atteints de démences liées à l'âge à l'aide des caméras videos portées par les patients. L'idée est de fournir aux médecins un nouvel outil pour le diagnostic précoce de démences liées à l'âge telles que la maladie d'Alzheimer. Plus précisément, les Activités Instrumentales du Quotidien (IADL : Instrumental Activities of Daily Living en anglais) doivent être indexées automatiquement dans les vidéos enregistrées par un dispositif d'enregistrement portable.

Ces vidéos présentent des caractéristiques spécifiques comme de forts mouvements ou de forts changements de luminosité. De plus, la tâche de reconnaissance visée est d'un très haut niveau sémantique. Dans ce contexte difficile, la première étape d'analyse est la définition d'un équivalent à la notion de « plan » dans les contenus vidéos édités. Nous avons ainsi développé une méthode pour le partitionnement d'une vidéo tournée en continu en termes de « points de vue » à partir du mouvement apparent.

Pour la reconnaissance des IADL, nous avons développé une solution selon le formalisme des Modèles de Markov Cachés (MMC). Un MMC hiérarchique à deux niveaux a été introduit, modélisant les activités sémantiques ou des états intermédiaires. Un ensemble complexe de descripteurs (dynamiques, statiques, de bas niveau et de niveau intermédiaire) a été exploité et les espaces de description joints optimaux ont été identifiés expérimentalement.

Dans le cadre de descripteurs de niveau intermédiaire pour la reconnaissance d'activités nous nous sommes particulièrement intéressés aux objets sémantiques que la personne manipule dans le champ de la caméra. Nous avons proposé un nouveau concept pour la description d'objets ou d'images faisant usage des descripteurs locaux (SURF) et de la structure topologique sous-jacente de graphes locaux. Une approche imbriquée pour la construction des graphes où la même scène peut être décrite par plusieurs niveaux de graphes avec un nombre de nœuds croissant a été introduite. Nous construisons ces graphes par une triangulation de Delaunay sur des points SURF, préservant ainsi les bonnes propriétés des descripteurs locaux c'est-à-dire leur invariance vis-à-vis de transformations affines dans le plan image telles qu'une rotation, une translation ou un changement d'échelle.

Nous utilisons ces graphes descripteurs dans le cadre de l'approche Sacs-de-Mots-Visuels. Le problème de définition d'une distance, ou dissimilarité, entre les graphes pour la classification non supervisée et la reconnaissance est nécessairement soulevé. Nous proposons une mesure de dissimilarité par le Noyau Dépendant du Contexte (Context-Dependent Kernel : CDK) proposé par H. Sahbi et montrons sa relation avec la norme classique L_2 lors de la comparaison de graphes triviaux (les points SURF).

Pour la reconnaissance d'activités par MMC, les expériences sont conduites sur le premier corpus au monde de vidéos avec caméra portée destiné à l'observation des d'IADL et sur des bases de données publiques comme SIVAL et Caltech-101 pour la reconnaissance d'objets.

Summary

The research of this PhD thesis is fulfilled in the context of wearable video monitoring of patients with aged dementia. The idea is to provide a new tool to medical practitioners for the early diagnosis of elderly dementia such as the Alzheimer disease. More precisely, Instrumental Activities of Daily Living (IADL) have to be indexed in videos recorded with a wearable recording device.

Such videos present specific characteristics i.e. strong motion or strong lighting changes. Furthermore, the tackled recognition task is of a very strong semantics. In this difficult context, the first step of analysis is to define an equivalent to the notion of "shots" in edited videos. We therefore developed a method for partitioning continuous video streams into viewpoints according to the observed motion in the image plane.

For the recognition of IADLs we developed a solution based on the formalism of Hidden Markov Models (HMM). A hierarchical HMM with two levels modeling semantic activities or intermediate states has been introduced. A complex set of features (dynamic, static, low-level, mid-level) was proposed and the most effective description spaces were identified experimentally.

In the mid-level features for activities recognition we focused on the semantic objects the person manipulates in the camera view. We proposed a new concept for object/image description using local features (SURF) and the underlying semi-local connected graphs. We introduced a nested approach for graphs construction when the same scene can be described by levels of graphs with increasing number of nodes. We build these graphs with Delaunay triangulation on SURF points thus preserving good properties of local features i.e. the invariance with regard to affine transformation of image plane: rotation, translation and zoom.

We use the graph features in the Bag-of-Visual-Words framework. The problem of distance or dissimilarity definition between graphs for clustering or recognition is obviously arisen. We propose a dissimilarity measure based on the Context Dependent Kernel of H. Sahbi and show its relation with the classical entry-wise norm when comparing trivial graphs (SURF points).

The experiments are conducted on the first corpus in the world of wearable videos of IADL for HMM based activities recognition, and on publicly available academic datasets such as SIVAL and Caltech-101 for object recognition.

List of Main Notations

Introduction, page 9

ADL Activities of Daily Living, page 9

IT Information Technologies, page 9

Video Monitoring with a Wearable Camera, page 13

IMMED Indexing Multimedia Data from Wearable Sensors for diagnostics and treatment of Dementia, page 13

MRI Magnetic Resonance Imaging, page 13

Human Activities Recognition Overview, page 27

DBN Dynamic Bayesian Network, page 30

HMM Hidden Markov Models, page 30

Image and Video Content Description, page 33

ART Angular Radial Transform, page 37

CBIR Content-Based Image Retrieval, page 33

CLD The MPEG-7 Colour Layout Descriptor, page 36

CSS Curvature Scale Space, page 37

DCD The MPEG-7 Dominant Color Descriptor, page 36

HOG Histogram of Oriented Gradients, page 38

HSV Hue, Saturation, Value color space, page 34

LUV Color space which attempt to perceptual uniformity, page 34

RGB Red, Green, Blue color space, page 34

SIFT Scale Invariant Feature Transform, page 39

SURF Speed-Up Robust Features, page 39

SVM Support Vector Machines, page 42

YUV Color space defined in terms of Luminance (Y) and Chrominance (u,v), page 34

Hidden Markov Models: Applications to Video Analysis, page 45

q_i i^{th} state of the HMM, page 47

s_t State at time t, page 46

A Transition matrix, page 47

CHMM Coupled Hidden Markov Model, page 59

EM Expectation-Maximization algorithm, page 53

GMM Gaussian Mixture Models, page 48

HHMM Hierarchical Hidden Markov Model, page 56

HMM Hidden Markov Model, page 45

SHMM Segmental Hidden Markov Model, page 60

Design of a two-level Hierarchic Hidden Markov Model for Activities Segmentation, page 63

A^0 Transition matrix of the top-level HMM, page 64

A^1 Transition matrix of a bottom-level HMM, page 64

Audio Audio descriptor, page 71

H_c Cut histogram, page 70

H_{tpe} Histogram of the log-energy of translation parameter, page 69

Loc Localization histogram, page 72

RM Residual motion descriptor, page 71

Experiments on Activities Recognition, page 75

FN False Negatives, page 77

FP False Positives, page 77

TN True Negatives, page 77

TP True Positives, page 77

Objects Recognition in Images and Videos, page 105

$\alpha_{i,k}$ Affinity between feature f_i and visual word v_k, page 111

f_i A feature vector, page 111

v_k A visual word, page 111

BoVW Bag-of-Visual-Words, page 106

BoW Bag-of-Words, page 105

CDK Context-Depenedent Kernel, page 116

ERR Equal Error Rate, page 118

PMK Pyramid Match Kernel, page 115

RMK Relaxed Matching Kernel, page 120

SPMK Spatial Pyramid Matching Kernel, page 119

tf-idf Term Frequency-Inverse Document Frequency, page 106

Delaunay Graph Words, page 123

E The set of edges, page 124

G A graph feature, page 124

X Feature points sets, page 124

Experiments on Object Recognition, page 131

MAP Mean Average Precision, page 133

Contents

Part I

Thesis Context

Introduction

In this introductory part we will define the context of this PhD thesis. We will define the medical aspect of the project in which this thesis is included and review related works combining Information Technology and healthcare. We will then detail the specifities of this project and describe the recording device and the characteristics of the recorded video streams.

Chapter 1

Introduction

1.1 Context

In the context of the aging of the European population, the development of home care services and technologies is crucial in order to help seniors maintaining their independence and stay at home longer and to help medical practitioners in their studies of aging and age related dementia for the elaboration of adapted therapeutic treatments.

The large scale medical study PAQUID [PHA+08] [HPL+06] which has involved 3777 subjects over 15 years, has shown that the earliest signs of dementia can be observed as functional difficulties in daily living activities up to 10 years before the clinical diagnostic defined by the cognitive methods of reference [MDF+84] [Hac94]. The capabilities of a patient to accomplish the activities of daily living (ADL) is estimated by the answers given by the patient and their relatives to a survey [LB69] [BGCG+92]. This approach is a first step towards the inclusion of ADL related symptoms in the elaboration of a patient's diagnosis. However, it captures an indirect observation of such symptoms. An observation method at the patient's home could potentially bring additional information that would be valuable for the doctors in charge of the diagnosis to further refine their analysis. The observation of the patients in their ecological and familiar environments at home would indeed allow a correct interpretation of the signs that appear in the questionnaires.

In order to enable the doctors to observe and analyze one patient's daily living, the use of a camera worn by the patient is an original approach. The observations of daily living activities would give an objective evaluation of potential difficulties in the activities helping the doctor in his diagnosis, enabling the setup of adapted reeducation techniques and the evaluation of therapeutic efficiency.

In this chapter we will first review how Information Technologies (IT) have been applied to healthcare and specifically for aging population. We will then focus only on the few projects which have used wearable videos for this purpose. We will finally detail the thesis objectives and the outline of this PhD manuscript.

1.2 Information Technologies for healthcare and the aging population

The application of IT to healthcare often aims at monitoring the evolution of some characteristics of the patient. In order to gather these characteristics, a setup of sensors has to be defined. The sensors used in the applications of IT to healthcare can be divided in two main categories: wearable sensors and stationary sensors.

Wearable sensors Most approaches of monitoring, as reviewed in [SCB+06], make use of accelerometers. Accelerometry is a low-cost, flexible, and accurate method for the analysis of posture and movement, with applications in fall detection and gait analysis. The accelerometers have been used for ADL recognition in [HBS07] and [Huy08]. These sensors require to be attached to several body parts such as wrists and knees [KSS03], to capture the motion of the patient.

The project MIThrill [DSGP03] is a wearable computing platform which integrates many sensors such as EKG (electrocardiography), EMG (electromyography), GSR (galvanic skin response) and temperature electrodes. These approaches can capture a lot of information on physical characteristics of the patient, but at the cost of a pervasive equipment.

Stationary sensors Stationary sensors require an installation at the patient's home, hence the application to many patients would induce the equipment at a very large scale with associated high costs of technical deployment. Stationary sensors can be cameras, infrared or passive infrared sensors, which can detect the presence of the patient in a specific area [CHD97]. We can also consider RFID tags as stationary sensors as they have to be arranged on specific objects or locations [SHVLS08].

Another option is to create a smart environment where sensors are installed and let the patient evolve in this environment [ZBTV07]. But this solution is hardly applicable to ADL analysis as the patient would have to evolve in an unknown environment. This will therefore add difficulties to the potential problems the patient may encounter while executing ADL, thus making an evaluation of the capacities or troubles of the patient much more ambiguous for doctors, as compared to an observation in an ecological and familiar environment.

1.3 Wearable videos for healthcare

The use of a wearable camera for healthcare is rather original. One of the first project involving the use of wearable imaging device is the SENSECAM project [HWB+06], which aims to provide wearable image lifelog as a memory aid. The camera is worn around the neck and captures pictures at several seconds of interval during the day. Finally, the events of the day are summarized as automatic life-logs [BKW+07].

A wearable camera has also been used in the WearCam project [PNB+07], where the camera is mounted on the head of children to help early diagnosis of autism. The automatic analysis of the child gaze during the execution of specific movements is interpreted for the diagnosis.

1.4 Thesis objectives

In this thesis, we address the problem of the automatic indexing of ADL in the video stream acquired from wearable cameras. This overall goal covers challenges on several levels such as the high variability of the visual content, the expected high semantic level of such analysis and the special nature of temporal sequence of images. We will therefore pursue two main objectives: first, propose models and methods for structuring the video stream into meaningful activities; second, propose approaches for extracting useful features from the video to improve the semantic level of the analysis.

This PhD work is tighlty related to the IMMED project (ANR-09-BLAN-0165-02) involving IT partners such as: the LaBRI[1], the IMS[2] and the IRIT[3]; and medical researchers from the INSERM U897[4].

1.5 Manuscript outline

We present here the organization of the manuscript. The chapter 2 briefly reviews the project history and details the wearable recording device and the characteristics of the video content recorded by this device when being worn by patients. This chapter concludes the introduction part of the manuscript.

In the second part we will focus on the indexing of activities of daily living. In chapter 3, we review the methods which have been proposed in the literature for the task of human activities recognition and see which model may be adapted for our task. We will then review the state-of-the-art in images and videos description in chapter 4, defining in our context which descriptors may be helpful for the video content description. We then analyze the properties of different models that may be used for the activities modeling in the formalism of Hidden Markov Models in chapter 5. The proposed model and the approaches for information extraction from the video content are introduced in chapter 6. The experiments are detailed and analyzed in chapter 7. These experiments conclude the second part of the manuscript.

The final part of this manuscript tackles the problem of object recognition with the aim of recognizing objects of daily life in our videos. After reviewing the state-of-the-art in object recognition in chapter 8, we will present our proposal in chapter 9. The experiments for evaluating our proposed approach are presented in chapter 10 which ends this last part.

The final conclusions and perspectives of this work are finally presented.

[1] http://www.labri.fr
[2] http://www.ims-bordeaux.fr
[3] http://www.irit.fr/
[4] http://www.isped.u-bordeaux2.fr/

Chapter 2

Video Monitoring with a Wearable Camera

Introduction

The idea of this research from a medical point of view is to use the video recording in the same way as an assessment such as MRI or radiography and to get this observations in a stress-less and friendly environment for the patient, while at home. In a target usage scenario the doctor will ask the paramedical staff to visit the patient with the recording system. Then, the recorded video is automatically processed and indexed by our method off-line. Finally, the doctor will use the video and indexes produced by our analysis to navigate in it and search for the activities of interest. Visual analysis of the latter serves to diagnose the disease or assess the evolution of the patient's condition. The typical recording scenario consists of two stages. A small bootstrap video for estimation of patient's localization in his home environment is recorded at the beginning of the recording session. Indeed, when a paramedical assistant comes to visit a patient for the first time, the patient "visits" his house when recording. Then, the patient is asked to realize some of the activities which are a part of clinical evaluation protocols in assessing dementia progress. These activities define the targeted events to be detected by our method.

We will here briefly review the evolution of the projects which have led to the IMMED (Indexing Multimedia Data from Wearable Sensors for diagnostics and treatment of Dementia) project and describe the general processing flow principle. We will then detail the evolution of the recording device towards the current prototype. We will then describe the characteristics of the videos recorded from this device and present the annotation and visualization tools.

2.1 Projects history

The research work on wearable video monitoring started with the exploratory project PEPS S2TI CNRS "Monitoring Vidéo Embarqué"[1]. This project which was the first in France to explore this problem, have enabled the definition of the constraints on the recording device such as the set of possible positions for the camera, on the type of camera,

[1]Projet "Monitoring Vidéo Embarqué": http://www.labri.fr/projet/AIV/projets/peps/

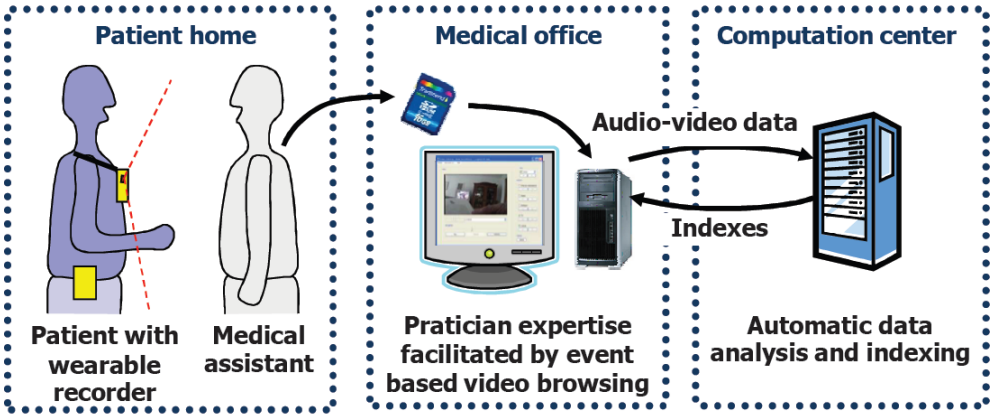


Figure 2.2.1: Global processing flowchart.

especially the type of lens which should be used in order to observe efficiently the ADL, and on the requirements for storage or transmission of the video stream. This short term project (2007-2008) have lead to the definition of a first prototype [MSBP+08] presented in section 2.3. This first prototype has shown the high acceptability of the device by the patients.

The IMMED project ANR-09-BLAN-0165-02 started in late 2008 and is funded by the ANR (Agence Nationale de la Recherche). The goal of this project is the development and the validation of a complete system for the diagnosis and monitoring of dementia by the use of a wearable camera. There are three main objectives to this project:

- The development of an audio and video recording device with ergonomic constraints adapted to the medical application;
- The development of methods for the automatic analysis of such video content enabling an easy visualization by the medical practitioners;
- The validation of these technologies by the integration in a clinical study and the definition of the first diagnosis guide adapted to this new paradigm.

The work of this thesis focuses on the second objective, the definition of methods for automatically indexing the video content in terms of ADL.

2.2 Processing flow principle

The general principle of the system is presented in Figure 2.2.1. The activities of the patient are acquired as audio and video data using a wearable device, as described in next section, under the supervision of a medical assistant. This data is stored on a SD-card which is transferred to the browsing station. A bootstrap annotation of the data is done on the beginning of the video in order to facilitate the automatic analysis of the rest of the video, see section 2.5.1. The video data is transferred through a secure connection to the computation center that indexes the whole video in order to detect the events of interest. The indexes are sent back to the browsing station to help a medical specialist visualize,

see section 2.5.2, and analyze patients' behavior and spot meaningful impairments in their daily activities.

2.3 The video recording device

The video acquisition device should be easy to put on, should remain in the same position even when the patient moves hectically, and it has to bring as less discomfort as possible to an aged patient.

Using the first prototype, which was presented in [MSBP+08], the monitored person was equipped with an onboard camera and microphone which were integrated into a bag that is attached to the shoulder and the hip. Two setups have been tested in this project: a first setup is located close to the manipulation area and a second one on the shoulder. The video and audio signals were transmitted wirelessly to a base recording station via an analog 2.4GHz transmitter within a 100m range, which is enough for capturing the actions inside a house. The recording station received the analog signal, digitized and compressed it through an acquisition card and stored the compressed video on a hard drive. The quality of the video obtained from this process may be higly altered by the wireless transmission. The noise can be induced by long distance transmission if the patient moves far away from the station but could also depend on the material used for the home construction as some materials may block the wireless signals.

In the current prototype, a vest was adapted to be the support of the camera. The camera, the battery and the storage device are all merged in the same sensor which is a GoPro[2] camera. The camera is fixed near the shoulder of the patient with hook-and-loops fasteners which allow the camera's position to be adapted to the patient's morphology. This position combined with the wide angle lens of the camera offers a large view field similar to the patient's one. With the camera being light and the vest distributing the weight on all the upper body, the acceptance of the device is very good. The volunteers have felt no discomfort while wearing it and were able to perform their activities as if the device was not present. An illustration of the device is given in Figure 2.3.1.

2.4 The video characteristics

The videos obtained from wearable cameras are quite different from the standard edited videos on one hand and from video surveillance videos on the other hand. Indeed, edited videos which are usually a target of video indexing methods have clean motion and are assembled from video shots with discontinuities on the shot borders. In our case, the video is recorded as a long continuous sequence, as in surveillance applications. The latter deal with stationary cameras or with regular motions, such as PTZ. In a "wearable" video the motion can be locally strong since the camera follows the ego-motion of the patient. This strong motion may produce blur in frames, as shown in Figure 2.4.1a. Moreover, the patient may face a light source, leading to sharp luminosity changes, as shown in Figure 2.4.1b and 2.4.1c. The camera has a wide angle objective in order to capture a large part of the patient's environment.

[2]GoPro camera: http://www.gopro.com/

Figure 2.3.1: The recording device (red circle) fixed on the vest adapted to be the support of the camera

(a) Motion blur due to strong motion.

(b) Low lighting while in dark environment.

(c) High lighting while facing a window.

Figure 2.4.1: Examples of frames acquired with the wearable camera.

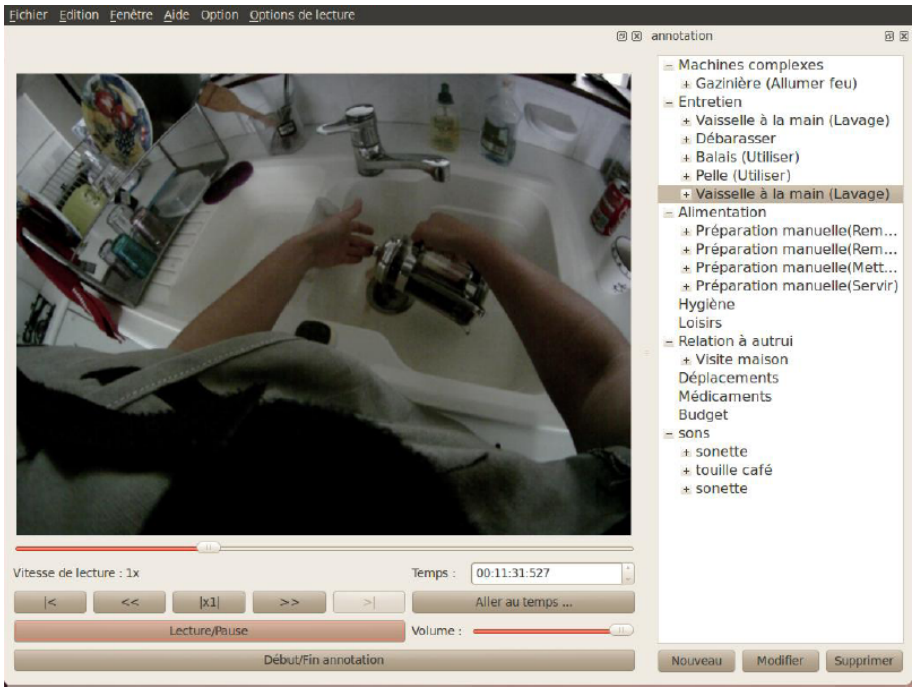


Figure 2.5.1: Example of an annotated video.

Furthermore, the variability of the data is very strong: the same activities are not performed by different patients in the same environment as this is the case in "smart homes" [ZBT+09].The patients evolve in their own home when recording.

2.5 Video annotation and visualization

In this section we will present the tool we provided to the medical partners of the project for the annotation and visualization of the videos.

2.5.1 Video annotation

Video indexing methods require a learning phase before being able to automatically detect localization (section 6.4.3.1) and recognize activities (chapter 6). This data are very much patient dependent, as home environments do not contain a large amount of invariants. Hence, a protocol has to be defined to annotate a minimum amount of data to allow the learning phase. The difficulty in here is that the annotation interface will be used by a medical practitioner who is not accustomed to advanced Information and Communication Technology (ICT). Hence the interface prototype developed comprises the activities and also localization elements. In the protocol, the medical assistant will annotate the first minutes of the video which will contain a tour of the patient's house. Therefore, the video annotation tool (Figure 2.5.1) should be easy to use and crossplatform.

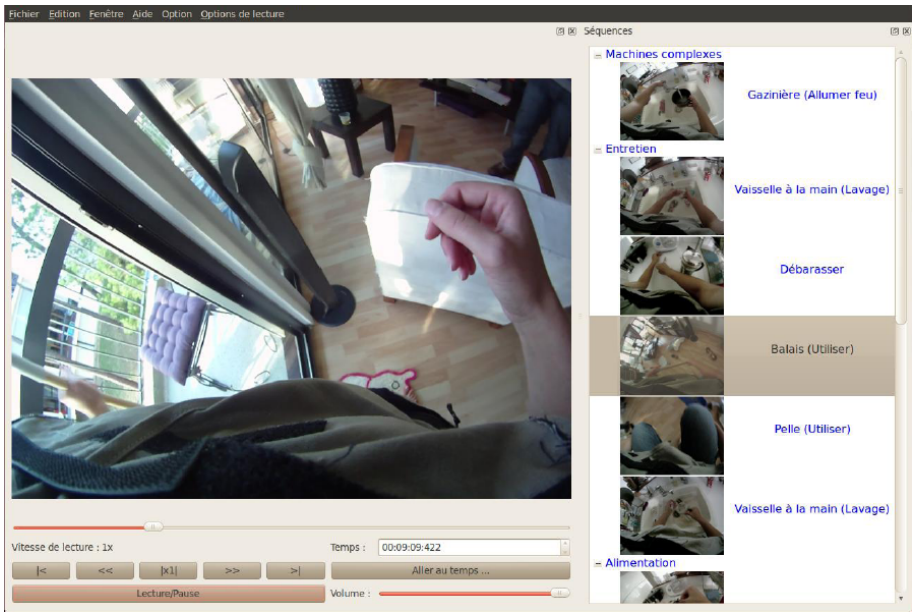


Figure 2.5.2: The visualization tool.

2.5.2 Video visualization

The video footage at the patient's house provides a long sequence shot. The sequence can be of one hour up to half-a-day duration, which is too long for a medical practitioner to watch entirely. Moreover, activities are of interest only when the autonomy of the patient may be evaluated. Hence the navigation is proposed via pre-defined set of activities, but also in a purely sequential manner to ensure all sequences of interest are viewed. The visualization tool is presented in Figure 2.5.2. The methods used for the automatic indexing of the activities will be presented in chapter 6.

2.6 Instrumental activities of daily living

The set of activities have been defined by the doctors. The taxonomy has evolved during the project, we give the final taxonomy in Table 2.6.1. This set covers the activities of interest for the doctors. However, all these activities might not have been executed during the recordings and moreover should have been executed several times in order to be used in the automatic analysis. The three levels of activities will not be used, only the "General name" and "Goal" level will help us define the target activities for the automatic recognition process.

Conclusion

In this chapter we have presented the project IMMED in which this work takes place. The processing flow principle shows clearly how the methods presented in this manuscript will be integrated in this context. The characteristics of the videos and the high semantic level of the activities defines the difficult problem we will tackle in this manuscript.

Table 2.6.1: Hierarchical taxonomy.

General name	Goal	Basic action
Complex machines	Coffee machine	Use
	Toaster	Use
	Microwave	Use
	Oven	Turn on, Cook, Turn off
	TV	Turn on, Remote, Turn off
	Gas cooker	Turn on, Cook, Turn off
	Dishwasher	Detergent, Program, Fill, Empty
	Washing machine	Detergent, Program, Fill, Empty
Cleaning	Empty	Use
	Hoover	Use
	Broom	Use
	Shovel	Use
	Bed	Make
	Bin	Use, Empty
	Wash dishes	Wash, Dry up, Storage
	Wash clothes	Hand-washed, Ordering, Hang out, Iron
Food	Drink	
	Eat	
	Cook	Cut, Serve, Fill, Lay the table
Hygiene	Clothes	Dress up, Button up, Lace up
	Body	Wash hands, Brush teeth, Dry hands
	Aesthetic	Perfume, Comb
Leisure	Gardening	Water, Cut, Plant, Harvest
	Pet	Play, Stroke
	Read	
	Watch TV	
	Computer	Use
	Knitting	
Relationship	Phone	Answer, Use
	Home visit	
Moving	Free	Up/Down the stairs, Walk, Open door
	With tools	Get up from bed/chair, Helped walk
Medicine	Medicine	Fill/Use pillbox
Budget	Budget	Pay, Check change

Conclusion

This introductory part has settled the context of this thesis. We have given the objectives and described the specifities of the videos we are working with. The next part will focus on the indexing of activities of daily living.

Part II

Indexing of Activities of Daily Living

Introduction

In this part we will review how human activities have been analyzed in the literature and how the information of images or videos can be extracted. We will then detail specific models for human activities modeling and proposed our method. Finally, we will evaluate our approach on the videos we have recorded.

Chapter 3

Human Activities Recognition Overview

Introduction

Human activities recognition has many applications, we can cite for example behavior recognition, content-based video analysis, security and surveillance, interactive applications and environments, animation and synthesis. We will review in this chapter methods which have been proposed for human activities recognition in a general scope and then more specifically in the context of wearable videos.

3.1 Human activities modeling

The task of human activities modeling can be separated into two problems according to the complexity of the activity being modeled. We will therefore make the distinction between actions and activities:

- actions are simple motion patterns such as walking, bending, etc.
- activities are much more complex sequences of actions which may involve one or several people such as "meet and shake hands".

In this section we will review methods proposed in the context of stationary cameras as video-based activity recognition has been investigated much more extensively in this context than in the wearable video context. This will allow us to define the first step towards actions or activities modeling which is the extraction of low level features.

Low level features Since videos consist of a large amount or raw information in the form of spatio-temporal pixel intensity variations, this raw information is not directly relevant for the task of understanding and identifying activities occurring in the video. We will review a few popular low-level features which are optical flow, point trajectories, background subtracted blob or shape, and filter responses.

Optical flow The optical flow corresponds to the apparent motion of individual pixels on the image plane. It is used as an approximation of the true physical motion. The optical flow gives a description of the regions in the image which are moving and of the corresponding velocity. The assumption of invariance of color or intensity of a pixel during its displacement between one video frame and the next is often made. In practice, the optical flow may suffer from noise and illumination changes. We refer to [BB95] for a survey on optical flow computation techniques.

Point trajectories Trajectories of moving objects, e.g. humans, can been good features to infer the corresponding activity. Rather than the raw trajectories in image plane, alternative representations (less sensitive to translations, rotations and scale changes) may be used, such as: trajectory velocities, spatio-temporal curvature, etc. A survey of these approaches can be found in [CS95]. The authors of [CS95] oppose motion-based approach to structure-based approach, stating that motion is more important.

Background subtracted blob and shape Background subtraction isolates the moving parts of a scene by segmenting it into background and foreground. From this binarization, the foreground may be treated like a blob considering the entire shape (a description of the region being computed, such as moments [Hu62]) or only the shape contour is taken into account [Fre61]. Finally, skeletal approaches represent the shape as a set of 1D curves [BN78].

Filter responses Similarly to local interest points in images that we will introduce in section 4.3, spatio-temporal filter responses can be defined. These features aim at detecting regions that present strong variation both in the spatial and temporal domains, these approaches will be detailed in section 4.4.

Actions modeling Methods for actions modeling can be categorized into three major classes: non-parametric, volumetric and parametric time-series approaches. We will briefly review the different characteristics and applications of these approaches.

Non-parametric approaches In non parametric methods, a set of features is extracted for each frame of the video and then matched to a template. The template can be a 2D template as in [BD01], where after a background subtraction, the sequence of background subtracted blobs is aggregated into a single static image: the "motion energy image". A "motion history image" can be created by giving higher weights to blobs extracted from newer frames. These templates are discriminative enough with regard to the recognition of simple action as "sitting down", "bending", "crouching", etc. The template can also be a 3D template i.e. a spatio-temporal template. For example, in [BGS+05], Blank et al. proposed a binary space-time volume built by stacking together background subtracted blobs. Other examples of non-parametric approaches can be found in [TCSU08].

Volumetric approaches Volumetric approaches consider a video as a 3D volume of pixel intensities. These approaches often rely on low-level features using spatio-temporal filtering previously categorized as "filter responses". The volumetric approach presented

in [KSH05], which relies on 3D filter banks and boosting will be detailed in 4.4. Volumetric features were used as an input for a fully automated deep model in [BMW+11]. This approach involves two steps; first, based on the extension of *Convolutional Neural Networks [LKF10]* to 3D, spatio-temporal features are learned; second, a *Recurrent Neural Network [GSS03]* classifier is trained for the task of human actions recognition. The results on the KTH data set show that the method outperforms most of the state-of-the-art methods.

Parametric methods Previous approaches were defining a template model of action and tried to match newly extracted features to this model. When addressing more complex actions such as dancing, or different actions of a tennis game, these approaches are limited. One of the most popular parametric model for such task is the Hidden Markov Model (HMM), which has been applied to gait pattern recognition in [LS06] and tennis shots recognition in [YOI92] for example. The HMMs will be detailed in chapter 5.

Activities modeling Modeling more complex activities requires higher level representation and reasoning methods. We can categorize these methods in three classes: graphical models, syntactic approaches and knowledge and logic-based approaches.

Graphical models The most popular graphical models are Dynamic Belief Networks (DBNs) and Petri nets. DBNs encode complex conditional dependence relations among several random variables. Usually, the structure of a DBN is provided by a domain expert but this is often difficult in real-life systems which involve a very large number of variables with complex interdependencies. Petri nets are bipartite graphs consisting of two types of nodes: places and transitions. They are an intuitive tool for expressing complex activities, particularly useful to model sequencing, concurrency, synchronization and resource sharing [DA94]. However, they are often built using a priori knowledge and cannot usually deal with uncertainty of lower level modules i.e. a missing or false detection.

Syntactic approaches Inspired by language modeling, activities recognition can be modeled by a set of production rules of lower level events. For example, context-free grammar (CFG) were used in [RA06] for the recognition of human activities and multiperson interactions. However, deterministic grammars cannot deal with errors at lower levels. Stochastic grammars have been developed to cope with this limitation and were for example applied to model a blackjack game with several participants in [ME02].

Knowledge and logic-based approaches Logic-based methods are intuitive as they rely on the definition of logical rules describing the activities. These logic rules require an explicit and extensive definition by a domain expert and do not address the problem of uncertainty in the observation of lower levels. To overcome this limitation, a combination of logical and probabilistic models was presented in [TD08]. The set of rules in these approaches are defined empirically for each specific deployment. To facilitate portability and interoperability between different systems, ontologies standardize the set of rules associated to an activity. Ontologies have been specified for some domains of visual surveillance such as meeting videos [HS04].

3.2 Human activities recognition in egocentric videos

Only a few works have used wearable cameras for human activities recognition. One of the first works, presented in [ASP99], was actually aiming at location recognition using color histogram similarity. The wearable camera was attached to the front of a cap. In [CMP00], Brian Clarkson et al. presented experiments on the recognition of a person's situation from a wearable camera and microphone. The types of situations considered in these experiments are coarse locations (such as work, in a subway or in a grocery store) and coarse events (such as in a conversation or walking down a busy street).

In [MM05], a shoulder-mounted wearable active camera was used for hand activity recognition. A skin color classification was run as a pre-processing stage. Considering two classes, skin and background, the corresponding model color histograms are built for the U and V channels. The skin color is detected by computing the conditional probabilities for each pixel to be assigned to one of the two models and then a spatial filter is applied to remove high frequency noise. The events: single hand (HS), hands resting on table (HR), handling a tennis ball (HB), hands operating a keyboard (HK) and hands operating a calculator (HC); are detected if their probability according to the overall area of skin, the object classification and spatial distribution, is greater than a threshold.

A Virtual Reality (VR) environment was used in [SPL+07] as a general framework to understand how the execution of an activity is related to the situated space and the object detection. The situated space is organized in four different subspaces: the world space (all known objects), the observable space (objects can be seen), the manipulable space (objects can be reached) and the object manipulation. The activities are cooking recipes, eating or cleaning and are modeled by Hidden Markov Models (HMM).

More recently, an activity recognition method using low resolution wearable vision was proposed in [SC09]. The vision section aims at recognizing manipulation motion. The authors extract hand detection as the pixels corresponding to residual motion by computing the difference between the current frame and the previous frame with motion compensation. Then, a temporal template using only the red chrominance component is used to model actions and the matching between temporal template is done by normalized cross correlation. A Dynamic Bayesian Network (DBN) that infers locations, objects and activities from a sequence of actions is introduced. The highest level of the DBN is modeled by an HMM.

Conclusion

In this chapter, we have reviewed how human activities can be modeled in videos, starting from the requirement of low-level features extraction to the definition of actions and activities models. Most of the works we have cited were applied to stationary videos, but several approaches presented are not applicable in our context of wearable videos. For example, low-level features such as background subtracted blobs and shapes can be hardly transposed to wearable videos as the background is almost constantly changing while the patient moves. The specificities of our videos require the use of adapted descriptors, we will explore popular descriptors in the literature in chapter 4.

Many models proposed for activities recognition integrate a priori knowledge to define the structure of the model. In our context, it is really difficult to define a generic set

of rules or sub-events to describe the IADLs. The next chapters will lead us to a model adapted to the activities met in our applicative domain.

Chapter 4

Image and Video Content Description

Introduction

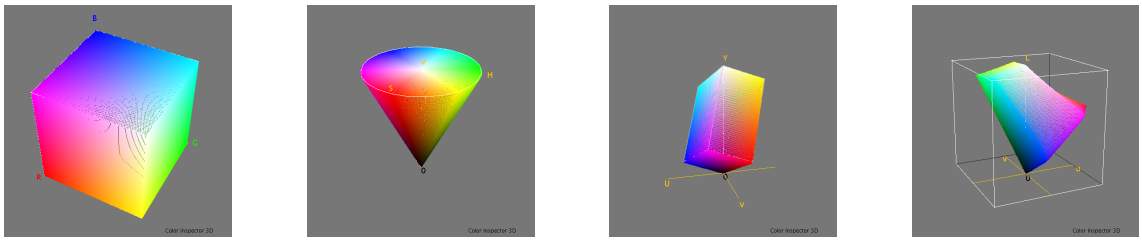
In order to analyze the content of images, state-of-the-art methods use features which describe some specific characteristics of the image. In this section we review most of them. They can be organized in the following categories: color, shape and local keypoints features. Finding an accurate description of the images is the first step towards all image analysis related applications such as Content-Based Image Retrieval (CBIR), image classification, object recognition or scene understanding. In this section we will first describe the color features which are mainly part of the MPEG-7 standard. The second part is dedicated to shape features which are specifically relevant for the task of object recognition. The last category of features that we will introduce is the local keypoint features. They are widely used in recent approaches, giving the best results on many different data sets for image classification.

4.1 Color features

Color information being an important part of human visual perception, computer vision researchers have defined features which aim at capturing this information within few numerical values. In this section, we present a set of color features and detail how each of them describes the images and videos. Color features are quite effective, meaningful and are pretty easy and fast to extract. In literature, they can be generally qualified as statistical or spectral features.

Foreword on image color spaces

Before going further in the description of color features, it is necessary to define the notion of color spaces. A color space is a mathematical model that enables the representation of colors, usually as a tuple of color *components*. There exists several models of this type, some motivated by the application background, some by the perceptual background of the human vision system.



(a) RGB (b) HSV (c) YUV (d) Luv

Figure 4.1.1: Graphical representation of different color spaces. Figures created with the 3D Color Inspector plugin for ImageJ.

The most commonly used color space is the *RGB* space, where a color is defined by the additive amount of the primary colors *R*ed, *G*reen and *B*lue. The design of this color space is closely related to the way the colors are reproduced on hardware devices such as computer screens, televisions, etc. A classic representation of the RGB color space is a cube, where each axis corresponds to the amount of red, green or blue components, see Figure 4.1.1a.

The *HSV* (for *H*ue, *S*aturation, *V*alue) color space was designed in an attempt to describe the perceptual color relationships more accurately than RGB, while remaining simple. It is defined by a unique, non-linear mapping of the RGB space. The colors in HSV are traditionally represented in a 3D-cone, see Figure 4.1.1b. The Hue takes values from 0 to 360 representing the color wheel. The Saturation represented by the distance from the center of a circular cross-section of the cone, corresponds to the purity of the color (pure red, green, yellow...). The Value component corresponds to the brightness/darkness of the color. It is located on the color cone at the corresponding distance from the pointed end of the cone. Saturation and Value usually take values in the interval $[0, 1]$.

The emergence of the color in television has motivated the usage of color spaces that separate the pixel luminance (brightness) and chrominance (color) values, such as *YUV*, see Figure 4.1.1c. Such a definition of the color enabled the cohabitation of black and white and color for analog television. YUV is also the standard in video encoding, since the chrominance component can be encoded using a reduced bandwidth without any loss of perceptual quality.

Finally, some efforts have been made in order to build color spaces that attempt perceptual uniformity. One such color space is *Luv*, see Figure 4.1.1d. Luv was designed so that the perceptual color difference can be computed in the Luv space using the euclidean distance.

4.1.1 Color histogram

Color histograms aim at representing the distribution of colors within the image or a region of the image. Each bin of a histogram h represents the frequency of a color value within the image or region of interest. It usually relies on a quantization of the color values, which may differ from one color channel to another. Each bin of the color histogram in one channel counts the number of pixels which color value, in the current channel, falls in the range of the bin. The quantization of the different channels is usually chosen to give a

finer resolution for the luminance channel of color spaces such as YUV and Luv. This can be related to the human perception as there are much more rod cells, able to capture the intensity of light but not color, than cone cells, dedicated to color. Histograms are invariant under geometrical transformations within the region of the histogram computation. Formally, given a color image I defined on a spatial domain Ω, see (4.1.1), the normalized histogram h_j models the marginal distribution of the values of channel j.

$$\begin{array}{rccl} I: & \Omega \subset \mathbb{R}^2 & \longrightarrow & \mathbb{R}^3 \\ & \mathtt{x} = (x,\, y) & \longmapsto & (I_1(x,y),\, I_2(x,y),\, I_3(x,y)) \end{array} \tag{4.1.1}$$

$$\begin{aligned} h_j(i) &= \frac{1}{|\Omega|} \sum_{\mathtt{x} \in \Omega} \delta_i(I_j(\mathtt{x})), \qquad \sum_{i=1}^{N_b} h_j(i) = 1 \\ \text{where } \delta_i(I_j(\mathtt{x})) &= \begin{cases} 1 & \text{if } lb(i) \leq I_j(\mathtt{x}) < ub(i) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4.1.2}$$

This is expressed in (4.1.2) where $lb(i)$ and $ub(i)$ are respectively the lower and upper bounds of the bin i of the histogram and N_b is the total number of bins.

4.1.2 Color moments

Color moments are another way of representing the color distribution of an image or a region of an image. The first order moment (4.1.3) is the mean which provides the average value of the pixels of the image.

$$E_j = \frac{1}{|\Omega|} \sum_{\mathtt{x} \in \Omega} I_j(\mathtt{x}) \tag{4.1.3}$$

The standard deviation (4.1.4) is a second order moment representing how far color values of the distribution are spread out from each other. It is computed as the square root of the variance of the distribution. The variance being computed as the mean of the squares of the deviations of the color values from the first order moment.

$$\sigma_j = \sqrt{\frac{1}{|\Omega|} \sum_{\mathtt{x} \in \Omega} (I_j(\mathtt{x}) - E_j)^2} \tag{4.1.4}$$

For higher order moment, we introduce the definition of the k^{th} central moment of channel j in (4.1.5).

$$\mu_{kj} = E\left[\left(I_j(\mathtt{x}) - E\left[I_j(\mathtt{x}) \right] \right)^k \right] \tag{4.1.5}$$

The third order moment, named skewness (4.1.6), can capture the asymmetry degree of the distribution. It will be null if the distribution is centered around the mean.

$$s_j = \frac{\mu_{3j}}{\sigma^3} \tag{4.1.6}$$

The fourth order moment, called kurtosis (4.1.7), is a measure of «peakedness» of the distribution or of the presence or absence of «heavy tails». This measure is merely used in the context of texture analysis. The kurtosis of a normal distribution being 3, a measure called «excess kurtosis» is defined as the kurtosis introduced in (4.1.7) minus 3, which therefore gives a value of 0 for a normal distribution.

$$\kappa_j = \frac{\mu_{4j}}{\sigma^4} \tag{4.1.7}$$

Using color moments, a color distribution can be represented in a very compact way selecting 3 moments for each of the 3 color channels, yielding a 9 dimensional vector [JLZZ02], [LZF03].

4.1.3 Dominant Color Descriptor

The Dominant Color Descriptor (DCD) was introduced in the MPEG-7 standard [MSS02]. The DCD provides a compact representation of salient colors within the image or the region considered. The DCD is defined as:

$$DCD = \{(c_i, p_i, v_i), s\},\ i = (1,\ 2,\ ...,\ N) \tag{4.1.8}$$

Where N is the number of dominant colors, c_i a vector of the color components values in a specific color space. The percentage of pixels in the image corresponding to the color c_i is $p_i \in [0, ..., 1]$, with the constraint $\Sigma_i p_i = 1$. The color variance v_i is an optional parameter. The last parameter s is a single value representing the spatial coherency of colors in the image. The most common use of the DCD is to retrieve images with similar colors in large databases. As it can be seen, the DCD compresses the color histogram, but it does not define the way the color domain has been quantized.

4.1.4 Color Layout Descriptor

The Color Layout Descriptor (CLD) is a compact representation of the spatial color distribution based on a partitioning of an image into 8x8 blocks. A representative color, for example the average color, is computed for each block. Let us denote $A_{m,n}$ the average color value of the block on the m^{th} row and n^{th} column, and N the number of partitions on each dimension (here $N = 8$). For each channel of the image, a Discrete Cosine Transform (DCT) is then applied to this set of 64 values:

$$DCT(p,q) = \frac{2}{N} C(p) C(q) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A_{m,n} cos\left[\frac{(2m+1)p\pi}{2N}\right] cos\left[\frac{(2n+1)q\pi}{2N}\right]$$
$$\text{for} \begin{cases} 0 \le p \le N-1 \\ 0 \le q \le N-1 \end{cases}, \text{where C(x)} = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } x = 0 \\ 1 & \text{for } x > 0 \end{cases} \tag{4.1.9}$$

A few low frequency coefficients of the DCT are then selected and quantized through a zigzag-scanning pattern. The CLD is invariant to changes in resolution (scale) but not invariant with respect to rotation or translation. Experiments conducted at the LaBRI, for example within the TRECVID challenge, have shown the strong discriminative power of the CLD.

We have briefly reviewed some of the mainly used color descriptors. We could also cite some others, more descriptors are described in [MOVY01]. The Scalable Color Descriptor is computed in the HSV color space. 16 bins are defined for Hue, 4 for Saturation and 4 for Value. The Color Structure Descriptor counts the number of times a color is present while scanning the image with a structuring element, typically a 8x8 square element. The color features presented here are most of the time used as global descriptors but are efficient in various contexts where color is the most important feature. However, they only describe one of the characteristics of the image. For the application of object recognition, the use of other kinds of information could be necessary. The next section presents a selection of shape features.

4.2 Shape features

Shape description relies on the extraction of accurate contours of shapes within the image or region of interest. An image segmentation is usually fulfilled as a pre-processing stage. In order for the descriptor to be robust with regard to affine transformations of an object, quasi perfect segmentation of shapes of interest is supposed. A correct image segmentation is really hard to obtain from our videos. In our work, none of the shape descriptors would be relevant. Nevertheless, we will briefly introduce here three representations of shapes within images.

4.2.1 Curvature Scale Space

The main idea of the Curvature Scale Space (CSS) descriptor [MS98] is that a shape is well described by its inflection points, the curvature zero-crossings points. The CSS descriptor describes the evolution of the set of inflection points, when a progressive smoothing is applied to the contour until it reaches convexity. The CSS descriptor is one of the shape descriptors of the MPEG-7 standard.

4.2.2 Angular Radial Transform

The Angular Radial Transform (ART) is a moment-based description method adopted as a region-based shape descriptor in the MPEG-7 standard. The ART can describe complex objects that could be connected or disconnected region shapes. It uses a complex orthogonal unitary transform of the unit disk that consists in the complete orthogonal sinusoidal basis functions in polar coordinates. The ART coefficients, F_{nm} of order n and m, are defined by:

$$F_{nm} = \int_0^{2\pi} \int_0^{1} A_m(\theta)\, R_n(\rho)\, f(\rho,\theta)\rho d\rho d\theta \tag{4.2.1}$$

where $f(\rho, \theta)$ is the image to describe, transformed into polar coordinates, and:

$$A_m(\theta) = \frac{1}{2\pi} exp(jm\theta)$$
$$R_n(\rho) = \begin{cases} 1 & if\ n = 0 \\ 2cos(\pi n \rho) & if\ n \neq 0 \end{cases} \tag{4.2.2}$$

The ART descriptor is defined as a set of normalized magnitudes of the set of F_{nm} coefficients which make this descriptor invariant to rotation around the center. In the MPEG-7 standard, twelve angular and three radial functions are used, yielding a descriptor vector of 35 dimensions as the value for $n = 0$ and $m = 0$ is constant.

4.2.3 Shape context

The approach introduced in [BMP02] called «shape context» describes the shape as a set of points sampled from the contours of the object. Then, for each point, a coarse histogram in a log-polar space of all the others points positions is computed. The matching between shapes is an instance of a square assignment (or weighted bipartite matching) problem that can be solved by the Hungarian method [PS98] or by the more efficient algorithm of [JV87].

4.2.4 Histogram of Oriented Gradients

Similarly to the shape context description, the Histogram of Oriented Gradients (HOG) introduced in [DT05], focus on the description of the edges of an object. The main idea of the HOG descriptors is to evaluate well-normalized histograms of image gradient orientations in a dense grid. The HOG descriptor is computed on a dense grid of uniformly spaced cells and makes use of overlapping local contrast normalizations for improved performance.

The first step of the descriptor calculation is the gradient computation. The gradient is estimated using point discrete derivative mask for each pixel. Defining local cells, such as a 3×3 neighborhood, each pixel casts a weighted vote (according to gradient magnitude) in an orientation histogram. The histogram channels are evenly spread over 0 to 180 degrees (for unsigned gradient) or 0 to 360 degrees (for signed gradient). For the application to human detection, Dalal and Triggs [DT05] found that unsigned gradient with 9 histogram channels performed the best. Introducing blocks which are a group of spatially connected cells, a local normalization of the gradient strengths can be applied in order to cope with changes in illumination and contrast.

The HOG descriptor has been widely applied for the task of human detection using either SVM classifier [DT05] or cascade classifier [ZYCA06], and remains to this day a very popular approach in the literature for this specific task.

Most color and shape descriptors introduced here aimed at describing the whole image or at least a rather large region. The HOG descriptor is defined as a set of local histograms of gradients computed for each cell of a dense grid. In image analysis, for the last few years, the focus has been mainly set on local features presented in the next section.

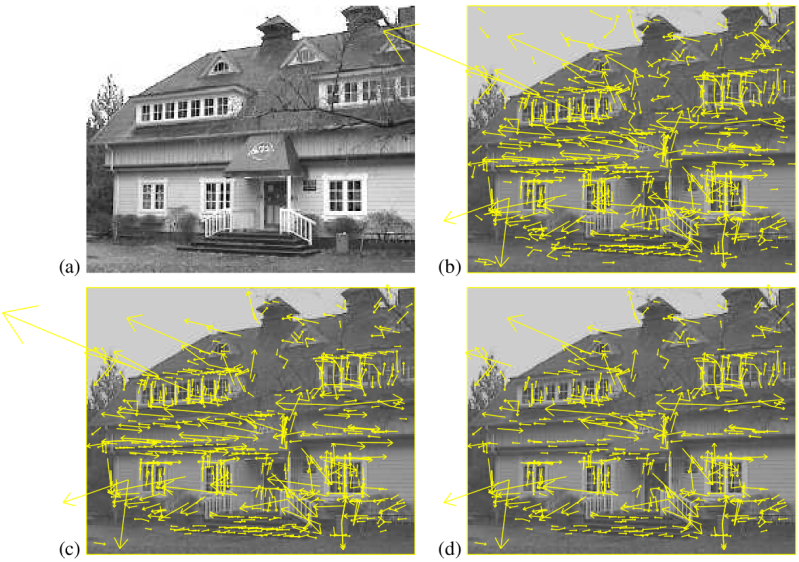


Figure 4.3.1: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures. Image from [Low04].

4.3 Local features

Another category of descriptors for image content is the local descriptors category. The main idea behind the use of local descriptors is to look for some areas which have really specific local characteristics rather than trying to describe a whole region or image. The local descriptors are computed at some locations in the image according to an interest point detector. The interest point detector usually searches for strong changes in the two-dimensional space of the image. This interest point is then described by a local feature. The first local feature we will present is the Scale Invariant Feature Transform (SIFT) descriptor proposed by Lowe in [Low04]. Similar works have been presented afterward, improving for example the computational cost with the Speed-Up Robust Features (SURF) in [BETVG08]. The following sections are overviews of SIFT and SURF descriptors, more details are given in the Appendix.

4.3.1 SIFT

The SIFT features proposed by Lowe have many properties that make them suitable for matching differing images of an object or scene. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. A large number of features can be extracted from typical images with efficient algorithms. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high

probability against a large database of features, providing a basis for object and scene recognition. The four major steps of SIFT images features computation are the following:

1. **Scale-space extrema detection**: The first stage of computation is to build a scale pyramid, obtained by convolutions of the image by variable-scale Gaussian, then to search over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. **Keypoint localization and filtering**: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability, i.e. unstable keypoints with low contrast or which are located along an edge and may be poorly determined are filtered out. The different stages of keypoint selection are shown in Figure 4.3.1.

3. **Orientation assignment**: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed according to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. **Keypoint descriptor**: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that permits significant levels of local shape distortion and change in illumination, see Figure 4.3.2.

4.3.2 SURF

SIFT features have been efficiently used in many image related applications. However, computing all the convolutions between images and the Gaussian filter at all the scales may have a high computational cost. Using integral images, Herbet Bay et. al proposed the Speed-Up Robust Features (SURF) in [BETVG08]. The four main ideas of the method introduced by Bay are:

1. **Integral images:** Every entry of an integral image is the sum of all pixels values contained in the rectangle between the origin and the current position.

2. **Detection of keypoint:** The detection of SURF keypoint relies on a Hessian-matrix approximation where the second order Gaussian partial derivatives are computed with box filters. Due to the use of box filters and integral images, Bay can apply box filters of any size at exactly the same speed directly. Therefore, the scale space is analyzed by up-scaling the filter size rather than iteratively reducing the image size, see Figure 4.3.3.

3. **Orientation assignment:** SURF keypoint are assigned an orientation to ensure rotation invariance. The dominant orientation is estimated by calculating the sum of Haar Wavelet responses within a sliding orientation window.

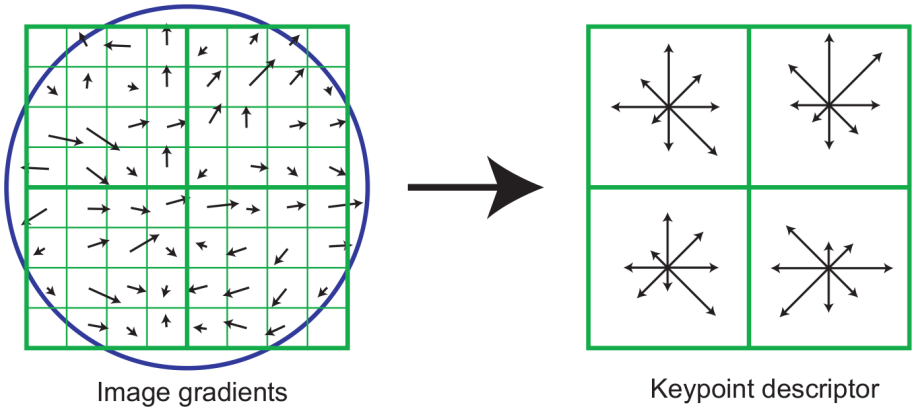


Figure 4.3.2: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 sub-regions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

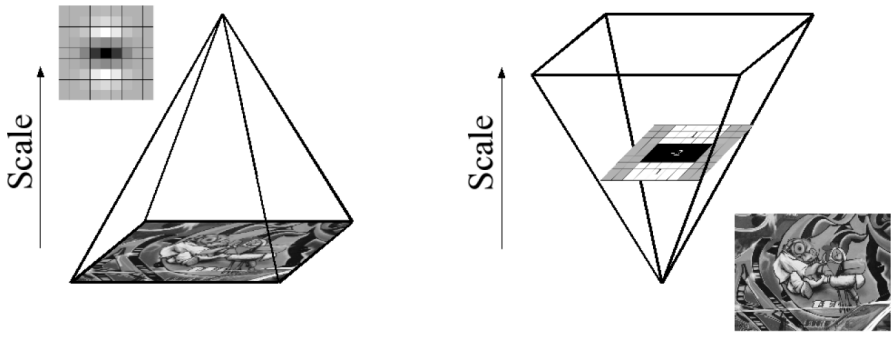


Figure 4.3.3: Instead of iteratively reducing the image size (left), the use of integral images allows the up-scaling of the filter at constant cost (right). Image from [BETVG08].

Figure 4.3.4: Detail of the Graffiti scene showing the size of the oriented descriptor window at different scales.

4. **Keypoint description:** In an oriented square window centered at the keypoint, which is split up into 4x4 sub-regions, each sub-region yields a feature vector based on the Haar wavelet responses.

4.4 Video descriptors

A video is often treated as a succession of frames i.e. image descriptors are applied to each frame separately. However, this kind of approach discards the temporal component of videos. The first works on video descriptors aim at extending efficient image descriptors towards video description by integrating the temporal dimension in existing frameworks.

One way of extending the power of interest points is to use an efficient framework applied to image and add a specific approach for video on top of it. This is the idea developed in [BBDBS09]. Ballan et al. have used the Bag-of-Visual-Words (BoVW) framework to represent each frame. The BoVW framework is presented in 8.1. The main ideas are to build a visual dictionary by clustering a large set of interest points; then to compute a visual word frequency vector for an image; finally, the image comparison corresponds to the computation of a distance between two histograms. The proposed method thus represents video clips as phrases (strings), which are the concatenation of the visual word frequency vectors of consecutive frames. The Needleman-Wunsch distance [NW70], which performs a global alignment that accounts for the structure of strings, is used to compare these phrases by defining a string kernel which is used in the Support Vector Machines (SVM) framework introduced in [DWV99]. The evaluation on action recognition on football videos and a subset of TRECVID 2005 contest clearly shows the good properties of the

method compared to a baseline BoVW approach, and that the SVM classifier outperforms a kNN classifier.

Other approaches have defined extensions of interest points detection including the temporal component. The spatio-temporal interest points, introduced in [Lap05] by Laptev, extends the idea of Harris and Förstner interest points operators and detect local structures in space-time where the image values have significant local variations in both space and time. As presented in the previous section, interest points detectors often rely on the convolution of an image with a Gaussian kernel, here the Gaussian kernel has independent spatial and temporal variances. Considering partial derivatives, the corners are detected similarly as in the spatial domain. More precisely Laptev has introduced a normalized spatio-temporal Laplace operator, and interest points are detected as extrema over both spatial and temporal scales. The problem of scale adaption is solved by splitting the space-time and scale dimensions, and iteratively optimizing over the subspaces until convergence. The events detected by these spatio-temporal points are spatial corners, the velocity vector of which is reversing direction. For example, it can be the moment when hands touch each other when clapping, the extreme positions of a hand waving or when legs are crossing in a gait pattern.

In [DRCB05], an approach defining «cuboids» is introduced. The detection is based on a response function obtained by a convolution of frames with a 2D Gaussian smoothing kernel and a Gabor function [GK95] applied temporally. The cuboid is extracted as a spatio-temporally windowed pixel values at the local maxima of the response function, the cuboid size being defined to contain most of the volume of data that has contributed to the response function. The cuboid is represented by a flattened vector of the brightness gradient in the three channels. Having a set of cuboids, a k-means clustering is applied to define types of cuboids. A video clip is only characterized by the histogram if the cuboids types present within it. The applications to three data sets: facial expressions, mouse behavior and human activity show that the method performs better than other state-of-the-art approaches (Efros et al. [EBMM03] and Zelnik-Manor and Irani [ZMI01]).

A real-time event detector using a cascade of filters based on volumetric features has been presented in [KSH05]. The volumetric features are computed on dense optical flow measurements, separated on its horizontal and vertical components. The features, which calculate the volume or the volumetric difference in X, Y and time, are computed over a volume window of 64×64 pixels by 40 frames in time. In fact, one million possible features are used scaling down to a 4×4 pixels by 4 frames spatio-temporal volume. Extending the idea of integral images used in [BETVG08], the authors introduced the «integral video» structure which contains at location (x, y) and time t, the sum of all pixels at locations less than or equal to (x, y, t), enabling fast computation of the features. These features are then integrated in a cascaded classifier. A scale adaptive window constructed by varying width and height, is used to scan the video with a classical sliding procedure. This may yield multiple detections in small area in space-time which is used as an indicator of the quality of detections: the more detections, the more likely the event.

Conclusion

In this chapter, we have introduced a wide set of image and video descriptors. The first descriptors gather information about the color distribution within an image. We have seen specifically that the Color Layout Descriptor (CLD) gathers a general spatial organization of the color in an image and shows robustness and efficiency in difficult tasks. The shape features have been briefly introduced. Yet, it would need an efficient segmentation of the image which is practically unfeasible in our context.

We have then introduced the two most popular local features, the interest points SIFT and SURF. They have been widely used and have shown discriminative power in many different tasks. However, the interest points can hardly be used directly for activities recognition in our videos. It is necessary to define intermediate tasks that can be fulfilled by using these features.

Finally, a set of features dedicated to video content description was introduced. However, most of them are applied to action recognition in videos from external viewpoint. The detections often correspond to specific body parts in characteristic position, for example for gait pattern detection or for hand waving recognition. In our first person view recordings, it seems difficult to make full use of these descriptors as most of the body remains mostly unseen.

Chapter 5

Hidden Markov Models: Applications to Video Analysis

Introduction

In chapter 3 we presented various models for recognition of humans activities. Some of them used efficiently Hidden Markov Models for this task. This chapter will first introduce the Markov property and simple models such as Markov chains, then presents the classical formulation of Hidden Markov Models (HMM) and more complex models using the HMM framework. The HMMs have been applied to a wide variety of topics, in this chapter we will mainly focus on applications to video analysis.

Many physical principles can be seen as deterministic processes. A deterministic process always produces the same output from given starting conditions and same external inputs. A stochastic process, or random process, is often described by a set of the different possible states of the process. The changes of state of the system are called transitions, and the probabilities associated with various state-changes are called transition probabilities. The indeterminacy evolution of the process is described by the mean of the transition probabilities between the different possible states of the process.

A very important property is the Markov property, from Andrey Markov [MN88] who contributed tremendously to the theory of stochatstic processes. The Markov property corresponds to the memoryless property of a stochastic process. A stochastic process has the Markov property if the conditional probability distribution of future states of the process, given the present state and the past states, depends only upon the present state i.e how the current state was obtained is not important. This property is therefore interesting in applications to video where events appear as a time sequence by enforcing the locality in time of such events.

Let us denote $Q = \{q_1, \ldots, q_n\}$ the state space i.e. the n possible states of the process, $q(t)$ the selected state at time t and $S = \{s_1, s_2, \ldots\}$ be a random process in the discrete space Q, the Markov property can be written as:

$$P(s_t = q(t)|s_{t-1} = q(t-1), \ldots, s_o = q(0)) = P(s_t = q(t)|s_{t-1} = q(t-1)) \qquad (5.0.1)$$

Markov Chains are discrete-time random processes endowed with the Markov property. Given a finite state space Q defined as previously, the transition between state q_i and q_j at time t can be denoted by $p_{ij}(t) = P(s_{t+1} = q_j | s_t = q_i)$. If this transition is time-independent, we call the Markov chain *time-homogeneous* and denote the transition probability by p_{ij}. One example could be a simple weather model, let $Q = \{q_1 = sunny, q_2 = rainy\}$ be the set of possible states and the transition matrix A, where p_{ij} is the probability that given the weather in current day is of type i, it will be followed by a day of type j:

$$A = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{pmatrix}. \tag{5.0.2}$$

Based on current state s_t, which corresponds to the weather on current day, a Markov chain can estimate the weather on next day according to the transition probabilities. The transitions are only dependent on the current states, the weather on all previous days have no influence at all. The Markov chains are applicable to a random variable which changes through time in a context where the system state is fully observable. In this case the state is directly observable, it is the weather on current day.

Some systems are only partially observable, the observations are related to the state of the system but not sufficient to precisely determine it. In this case, the transitions probabilities alone do not enable a correct representation of the system. Following previous example, we want to establish the weather model of an unknown location. In this location, consider we know someone, say Bob, who talks to us on the phone every day saying which of the three following activities he has perform this day: «walk», «shop» or «clean». In this case, the observations differ from the states: the observations are the activities while the states are still the weather conditions. The current state, i.e. the weather of the current day, being unkown, a Markov chain cannot be applied. We say that the states are hidden. Modeling this situation can be done using a Hidden Markov Model (HMM) that we will detail in section 5.1. Markov chains and HMM are generative models, i.e a full probabilistic model of all variables. In the case of HMM, the variables are the transitions probabilities between states and observations probabilities for each state. We say that a state emits an observation, in our case the state «sunny» will induce a given proportion of activities «walk», «shop» and «clean» which is different from the state «rainy». The probability of emitting any observation for each state has to be included in the parametric model. The HMM enables the estimation of these hidden parameters.

In video applications of the HMM, information is extracted from the analysis of images in the form of descriptors. These descriptors constitute the observations of an HMM. The objective is therefore to infer the state from the observations. In simple cases, the state corresponds exactly to an event to be retrieve e.g. the weather condition. However, when the events to be detected are complex, they can hardly be modeled by a single state. A hierarchical approach can be applied to model simple events at bottom level, and more and more complex events while going up the hierarchy. This aspect will be presented in the Hierarchical Hidden Markov Models section 5.2. When using several descriptors, the question of how to combine these different modalities arises. One approach is called the Coupled Hidden Markov Models, see section 5.3, where each modality is treated separately

in its own HMM. The Segmental Hidden Markov Models introduce the possibility to have observations which are not always of the same duration and will be presented in section 5.4.

5.1 Classical hidden Markov models

The Hidden Markov Model (HMM) is as statistical model which was presented in [Rab89] in the context of speech recognition. In this section, we will first introduce the HMM formalism. The practical use of a HMM model for data analysis considers the model itself and its parameters, a temporal sequence of states corresponding to the evolution of the process which is not observable and the set of observations that can be obtained from the data. The main problems that have to be solved and the solutions to these problems will be presented in the following sections. Finally, applications of the classical Hidden Markov Models will be detailed in the last section.

5.1.1 Hidden Markov model formalism

States and transitions An HMM is composed of m states: $Q = \{q_1, ..., q_m\}$. The state at current time t is denoted s_t. Each state is connected to other states but not necessary all of them. The transition matrix $A = (a_{ij})$ contains all the transitions probabilities between all states of the HMM, a_{ij} is the transition probability between state q_i and q_j and the diagonal of the matrix contains the loop probabilities. Formally:

$$A = \{a_{ij}\} \text{ where } a_{ij} = P(s_{t+1} = q_j \,|\, s_t = q_i),\ 1 \leq i,\ j \leq m \tag{5.1.1}$$

which induces the stochastic contraint

$$\sum_{j=1}^{m} a_{ij} = 1,\ 1 \leq i \leq m\ \forall i, j\ a_{ij} \geq 0 \tag{5.1.2}$$

Each state has a probability to start the observation sequence. These probabilities are stored in a vector denoted π.

Observations HMM can be formulated for a discrete or continuous observation space. In the discrete case, the observations are part of a known alphabet of size M, we can define the alphabet $E = \{e_1, ..., e_M\}$. The emission probability matrix contains the probability of each symbol e_k to be emitted by each state q_i. Formally:

$$B = \{b_i(e_k)\} \text{ where } b_i(e_k) = P(e_k(t) \,|\, s_t = q_i),\quad \begin{matrix} 1 \leq i \leq m \\ 1 \leq k \leq M \end{matrix} \tag{5.1.3}$$

with the stochastic contraint

$$\sum_{k=1}^{M} b_i(e_k) = 1,\ 1 \leq i \leq m\ \forall i, k\ b_i(e_k) \geq 0 \tag{5.1.4}$$

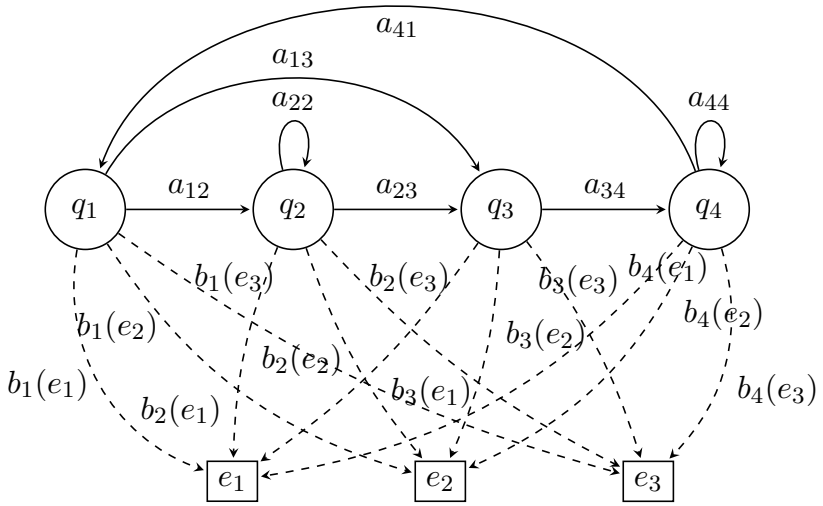


Figure 5.1.1: Example of a HMM with 4 states and discrete observation space with an alphabet of size 3.

The figure 5.1.1 is an illustration of a 4 states HMM defined over an alphabet of size 3.

In the continuous case, the observations are not a single symbol but a vector of values. An observation model, taking the form of a probability density function has to be defined. Obtain a proper model of the data with a single probability function may be difficult. Therefore, a Mixture Model f is often defined as the probability density function:

$$f(o) = \sum_{k=1}^{K} w_k \, f_k(o) \tag{5.1.5}$$

where f_k is a component density of the mixture and w_k is the weight of this component (with the constraints $0 \leq w_k \leq 1$ and $\sum_k w_k = 1$). For the case of Gaussian Mixture Model (GMM), the probability density function of the observation of each state q_i is defined by the following:

$$b_i(o) = \sum_{l=1}^{L} w_{il} \, f_l(o; \mu_{il}, \Sigma_{il}), \; 1 \leq i \leq m \tag{5.1.6}$$

$$\text{where} f_l(o; \mu_{il}, \Sigma_{il}) = \frac{1}{|2\pi\Sigma_{il}|^{\frac{1}{2}}} exp(-\frac{1}{2}(o - \mu_{il})^T \Sigma_{il}^{-1}(o - \mu_{il})) \tag{5.1.7}$$

where o is the observation vector, w_{il} is the weight of component l for state q_i, f_l is a density function, μ_{il} is the mean vector and Σ_{il} is the covariance matrix of model l for state q_i. The weights w_{il} have to satisfy the constraints:

$$\sum_{l=1}^{L} w_{il} = 1, \; 1 \leq i \leq m \tag{5.1.8}$$

$$w_{il} \geq 0,\ 1 \leq i \leq m, 1 \leq l \leq L$$

Under these conditions, the probability density function is normalized:

$$\int_{-\infty}^{\infty} b_i(x)dx = 1,\ 1 \leq i \leq m \tag{5.1.9}$$

Typical problematics The complete HMM model is defined as: $\lambda = (A, B, \pi)$ in the discrete case, where the transition matrix is A, the initial probabilities of each state is π and the observations probability matrix is B, or $\lambda = (A, b, \pi)$ in the continuous case, where b is the probability density function of the observations. Three main problems have to be solved:

- computing the probability of an observation sequence given a model: $P(o_1 \dots o_n|\lambda)$
- inferring the best state sequence given an observation sequence: $\underset{q_1 \dots q_n}{argmax} P(s_1 \dots s_n = q_1 \dots q_n|o_1 \dots o_n, \lambda)$
- estimating the model parameters that maximize the probability of observation of a sequence: $\underset{\lambda}{argmax} P(o_1 \dots o_n|\lambda)$

5.1.2 Probability of an observation sequence

Having an observation sequence $O = o_1 o_2 ... o_T$ and a HMM λ, how can we efficiently compute the probability of the observation sequence given the model $P(O|\lambda)$? This can be useful if we have several competing models and want to choose which model best matches the given observations.

The full computation of all the possible state sequences is however unfeasible, the calculation of $P(O|\lambda)$ would involve $2T \times m^T$ calculations. For example, this computation for $m = 10$ (states of the HMM) and $T = 100$ (observations) would be on the order of $2 \times 100 \times 10^{100} \approx 10^{102}$ computations.

Fortunately, a much more efficient approach exists: the forward-backward procedure [BE67] [BS68]. This procedure involves the definition of two variables: the forward variable $\alpha_t(i)$ and the backward variable $\beta_t(i)$. The forward variable $\alpha_t(i)$ is the probability of the partial observation sequence from the start until time t and state q_i at time t given the model λ. The backward variable $\beta_t(i)$ is the probability of the partial observation sequence from time $t + 1$ to the end and state q_i at time t given the model λ. Both these variables are computed inductively. We will formally define $\alpha_t(i)$ in (5.1.10) and its computation and illustrate the induction within the computation, then provide the same information for $\beta_t(i)$ in 5.1.14. An illustration of forward variable computation is given in Figure 5.1.2.

$$\alpha_t(i) = P(o_1 o_2 ... o_t, s_t = q_i|\lambda) \tag{5.1.10}$$

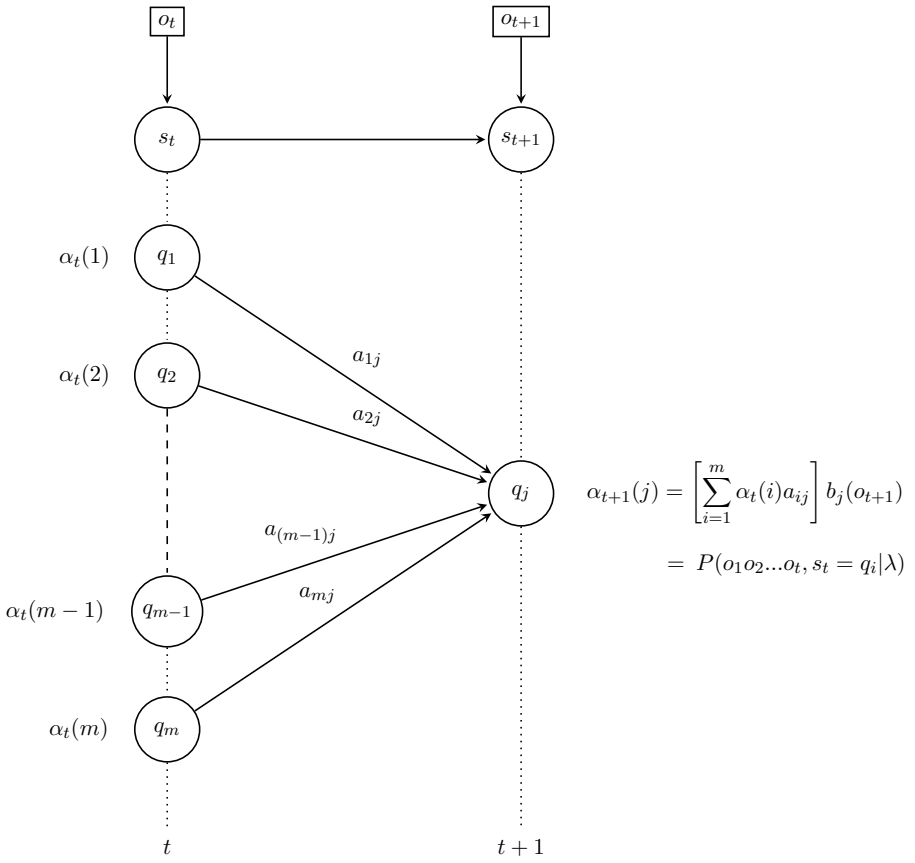


Figure 5.1.2: Illustration of forward variable computation.

Initialization :

$$\alpha_1(i) = \pi_i b_i(o_1), \qquad 1 \le i \le m \tag{5.1.11}$$

Induction :

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{m} \alpha_t(i)a_{ij}\right] b_j(o_{t+1}), \quad \begin{matrix} 1 \le j \le m \\ 1 \le t \le T-1 \end{matrix} \tag{5.1.12}$$

Termination :

$$P(O|\lambda) = \sum_{i=1}^{m} \alpha_T(i) \tag{5.1.13}$$

Only the computation of the forward variable is necessary for computing $P(O|\lambda)$. But, since the computation of the backward variable is necessary for the two other problems and is similar, we will present it here. An illustration of backward variable computation is given in Figure 5.1.3.

$$\beta_t(i) = P(o_{t+1} o_{t+2} ... o_T | s_t = q_i, \lambda) \tag{5.1.14}$$

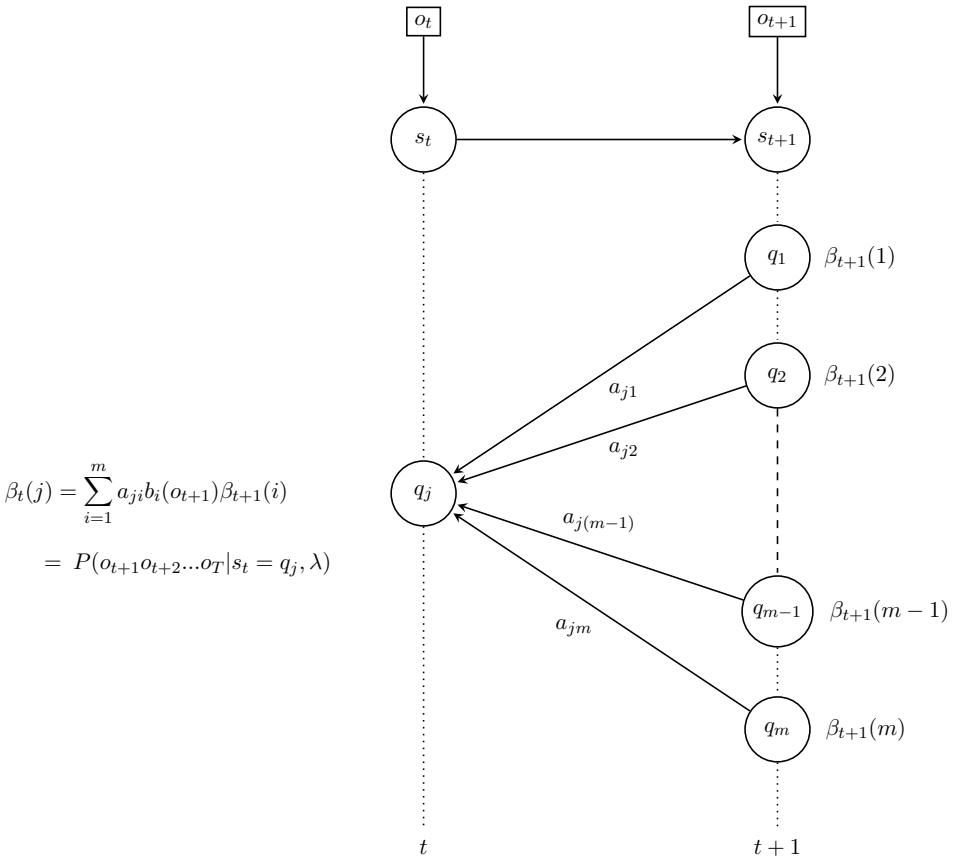


Figure 5.1.3: Illustration of backward variable computation.

Initialization :

$$\beta_T(i) = 1, \qquad 1 \leqq i \leqq m \tag{5.1.15}$$

Induction :

$$\beta_t(i) = \sum_{j=1}^{m} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \qquad \begin{matrix} 1 \leqq i \leqq m \\ t = T-1, T-2, ..., 1 \end{matrix} \tag{5.1.16}$$

5.1.3 Best states sequence

When addressing the problem of finding the best state sequence the first issue is to give a definition of such a sequence. The first solution is to define the sequence as the sequence of states which are individually more likely at each time. The second solution is to use the sequence of states which is globally more likely.

Computing states that are individually more likely The variable $\gamma_t(i)$ defines the probability to be in state q_i at time t according to the sequence of observations O and the

model λ:

$$\gamma_t(i) = P(s_t = q_i|O,\lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{m}\alpha_t(i)\beta_t(i)} \tag{5.1.17}$$

The best state sequence can be defined by selecting at each time the most probable state:

$$s_t = \underset{1\leq i\leq m}{argmax}[\gamma_t(i)],\ 1 \leq t \leq T \tag{5.1.18}$$

The problem with this definition is that it is not directly related to the probability of the estimated sequence considered globally. For instance, it could potentially give an invalid sequence if one of the selected states has a zero probability transition to the next best state.

Computing the global best state sequence To compute the best state sequence using this second definition we use the Viterbi algorithm [Vit67],[FJ73]. To take into account the sequence probability, we need to define the variable $\delta_t(i)$ which represents the best score (the highest probability) of a path finishing at state q_i at time t given the t first observations:

$$\delta_t(i) = \underset{s_1,s_2,\ldots,s_{t-1}}{max} P(s_1 s_2 ... s_{t-1}, s_t = q_i|o_1 o_2 ... o_t, \lambda) \tag{5.1.19}$$

and by induction:

$$\delta_{t+1}(i) = \underset{1\leq j\leq m}{max}[\delta_t(j)a_{ji}] \times b_i(o_{t+1}) \tag{5.1.20}$$

To complete the procedure to find the best state sequence, the best path which has reached state q_i at time t has to be stored for each time and each state. Thanks to the Markov property, only the value of the previous state needs to be stored to complete the procedure. Let us denote $\psi_t(i)$ this value. The computation of all the values for all states and all times is done by a recursive procedure:

Initialization :

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(o_1), & 1 \leq i \leq m \\ \psi_1(i) &= 0 \end{aligned} \tag{5.1.21}$$

Recursion :

$$\begin{aligned} \delta_t(i) &= \underset{1\leq j\leq m}{max}[\delta_{t-1}(j)a_{ji}] \times b_i(o_t), & \begin{matrix}1 \leq i \leq m \\ 2 \leq t \leq T\end{matrix} \\ \psi_t(i) &= \underset{1\leq j\leq m}{argmax}[\delta_{t-1}(j)a_{ji}] & \begin{matrix}1 \leq i \leq m \\ 2 \leq t \leq T\end{matrix} \end{aligned} \tag{5.1.22}$$

Termination :

$$\begin{aligned} P^* &= \underset{1\leq i\leq m}{max}[\delta_T(i)], \\ s_T^* &= \underset{1\leq i\leq m}{argmax}[\delta_T(i)] \end{aligned} \tag{5.1.23}$$

Finally, the best state sequence retrieval needs to follow the best path backwards:

$$s_t^* = \psi_{t+1}(s_{t+1}^*),\ t = T-1, T-2, ..., 1 \tag{5.1.24}$$

An illustration of the Viterbi algorithm is given in Figure 5.1.4.

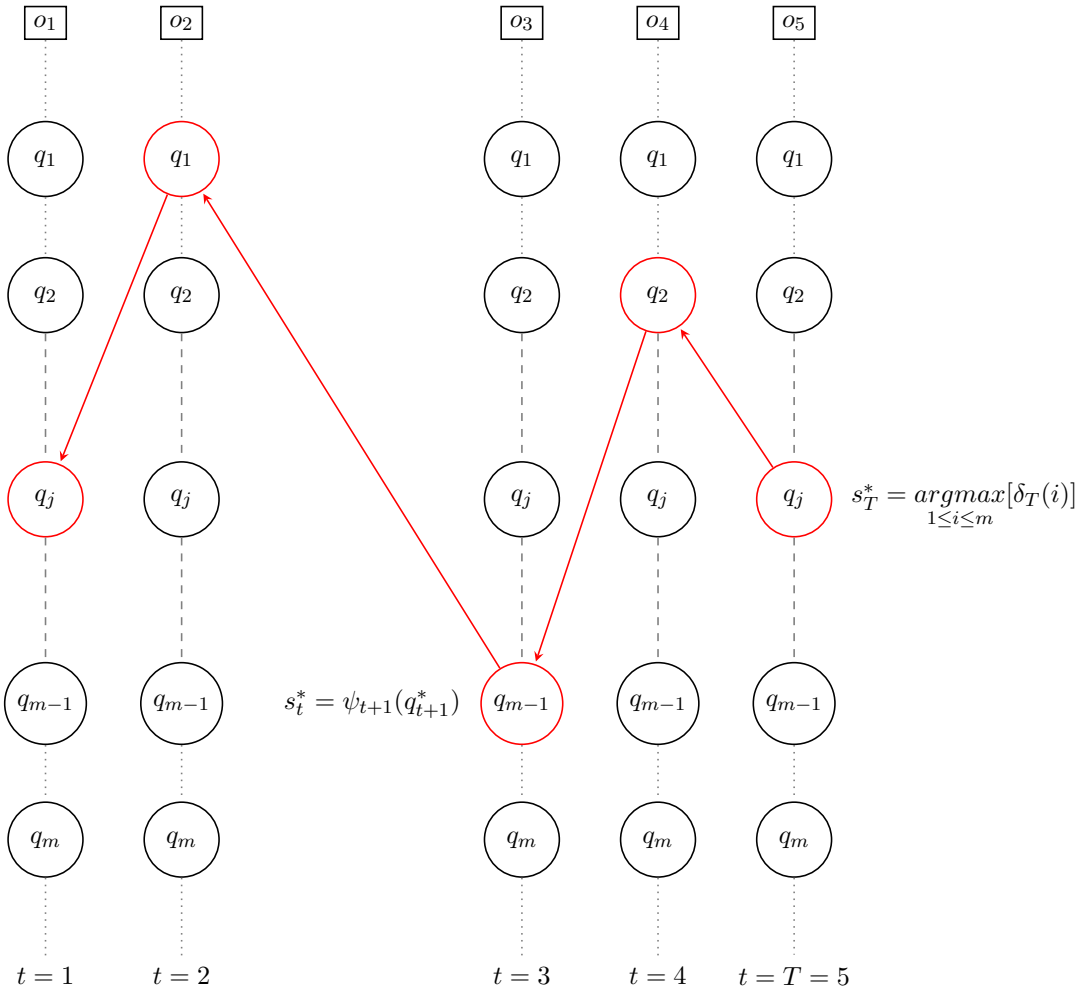


Figure 5.1.4: Illustration of Viterbi algorithm.

5.1.4 Estimating the model parameters

The parameters of a HMM can be fixed if prior knowledge on the data exist. However, in many applications, prior knowledge cannot be integrated for the parametrization of the HMM. Therefore a learning procedure is needed. Adjusting the model parameters to maximize the probability of an observation sequence is a difficult problem. There is no optimal way for this estimation. However, an iterative procedure such as the Baum-Welch method [BPSW70] (which is equivalent to the Expectation-Maximization (EM) method [DLR77]) can enable the estimation of the HMM parameters to produce a model

where $P(O|\lambda)$ is locally maximized. To describe the Baum-Welsh procedure we need to introduce another variable $\xi_t(i,j)$ which represents the probability of being in state q_i at time t, and state q_j at time $t+1$, given the model and the observations sequence:

$$\xi_t(i,j) = P(q_t = q_i, q_{t+1} = q_i | O, \lambda) \tag{5.1.25}$$

Using the previously defined forward and backward variables, we can write $\xi_t(i,j)$ as (5.1.26), where the numerator is $P(q_t = q_i, q_{t+1} = q_j, O|\lambda)$.

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \tag{5.1.26}$$

The variable $\gamma_t(i)$ representing the probability of being in state q_i at time t defined in (5.1.17) can be rewritten according to $\xi_t(i,j)$:

$$\gamma_t(i) = \sum_{j=1}^{m} \xi_t(i,j) \tag{5.1.27}$$

By summing $\gamma_t(i)$ over time (excluding the final time T), we obtain a quantity which represents the expected number of transitions from state q_i and by summing in the same way $\xi_t(i,j)$ we obtain the expected number of transitions from state q_i to state q_j. Using these values, the parameters of the HMM can be re-estimated as follows for the discrete case:

$$\overline{\pi_i} = \text{expected number of times in state } q_i \text{ at time } (t=1) = \gamma_1(i) \tag{5.1.28}$$

$$\overline{a_{ij}} = \frac{\text{expected number of transitions from state } q_i \text{ to } q_j}{\text{expected number of transitions from state } q_i} \tag{5.1.29}$$

$$= \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

$$\overline{b_i(e_k)} = \frac{\text{expected number of times in state } q_i \text{ observing } e_k}{\text{expected number of times in state } q_i} \tag{5.1.30}$$

$$= \frac{\sum_{t=1}^{T}{}_{s.t.\ O_t=e_k}\ \gamma_t(i)}{\sum_{t=1}^{T}\gamma_t(i)}$$

In the continuous case, using GMM, the re-estimation of the observations model requires the definition of an adapted γ variable. The variable $\gamma_t(i,l)$ is the probability of

being in state q_i at time t with the l^{th} mixture component accounting for O_t. It generalizes the previous definition of $\gamma_t(i)$ given for a discrete density:

$$\gamma_t(i,l) = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{m}\alpha_t(i)\beta_t(i)}\right]\left[\frac{w_{il}\, f(O,\mu_{il},\Sigma_{il})}{\sum_{l=1}^{L} w_{il}\, f(O,\mu_{il},\Sigma_{il})}\right] \tag{5.1.31}$$

The parameters of the observation model can therefore by re-estimated by the following rules:

$$\overline{w_{il}} = \frac{\sum_{t=1}^{T}\gamma_t(i,l)}{\sum_{t=1}^{T}\sum_{l=1}^{L}\gamma_t(i,l)} \tag{5.1.32}$$

$$\overline{\mu_{il}} = \frac{\sum_{t=1}^{T}\gamma_t(i,l)\cdot O_t}{\sum_{t=1}^{T}\gamma_t(i,l)} \tag{5.1.33}$$

$$\overline{\Sigma_{il}} = \frac{\sum_{t=1}^{T}\gamma_t(i,l)\cdot(O_t-\mu_{il})(O_t-\mu_{il})^T}{\sum_{t=1}^{T}\gamma_t(i,l)} \tag{5.1.34}$$

These re-estimation rules can easily be interpreted. The weight $\overline{w_{il}}$ is evaluated as the ratio between the estimated number of times being in state q_i with mixture l over the total number of times being in state q_i. The average and covariance of the l^{th} component of the i^{th} state is recomputed by reweighting with $\gamma_t(i,l)$.

The Baum-Welsh algorithm will produce a locally optimal model by iteratively running the previous procedure using $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$ at each step instead of $\lambda = (A, B, \pi)$. The Baum-Welsh algorithm is equivalent to an EM procedure and produces at each step of the process a set of HMM parameters which satisfies the stochastic constraints defined in (5.1.1) and (5.1.3).

5.1.5 Hidden Markov models applications

Since their most known applications for speech recognition in the early 1970's, HMMs have been applied to many fields such as handwriting and gesture recognition [CKV08], bio-informatics and video. The video applications of the HMMs have been first developed for low-level temporal structuring like the method for video segmentation in [BW98]. The aim of this work is to retrieve the shots within a video taking into account a number of possible transitions as cut, fade, dissolve and camera motion like pan and zoom. J.S.

Boreczky and L.D. Wilcox [BW98] use a luminance histogram distance, audio features based on cepstral coefficients and motion features as observations of a HMM. The HMM states represent the shot, the camera motion (pan and zoom) and the transitions between shots (fade, cut and dissolve). The results clearly outperformed the baseline method based on thresholding.

HMMs have also been widely applied for event classification in sports videos, since many sports have well established rules giving highly structured videos. The events in sports such as tennis or baseball are very constrained, and collecting a data-set of such videos is easy as these sports are widely broadcasted. For example, Yamoto et al. [YOI92] and later Petkovitc [PJZ01] et al. have applied HMMs for classifying tennis events into one of 6 strokes.

Most of the works using HMMs for sports video analysis need to perform a shot segmentation as a preprocessing step. But as presented by N. Harte, D. Lennon and A. Kokaram in [HLK09], HMMs can perform simultaneously segmentation and recognition. In this work, the data-set is composed of videos from a scientific study on the retainment of primary reflexes from infancy in dyslexic children. Among fourteen exercises, the authors focus on the recognition of one particular exercise in which the head of the child is successively rotated to the left and to the right. The features are based on motion vectors extracted from the video and the events modeled are rotation, the child pose setup, a pause and no-rotation.

In the follow-up of HMM applications in structuring of complex video content, Hierarchical HMM have been introduced. They will be presented in the following section.

5.2 Hierarchical hidden Markov models

The Hierarchical Hidden Markov Model (HHMM) makes possible the use of hierarchical structure within an HMM. In [FST98] the hierarchical structure is defined using the bottom-level states as emitting states and higher-level states as internal states to model the structure. In this formulation, a state is denoted by q_i^d ($d \in \{1, ..., D\}$) where i is the state index and d is the hierarchy index i.e. the state level. The number of sub-states of an internal state q_i^d is denoted by $|q_i^d|$. The possible transitions are in this model both horizontal and vertical. A transition matrix $A^{q_i^d} = (a_{jk}^{q_i^d})$ is defined for the sub-states of each internal state q_i^d, where $a_{jk}^{q_i^d} = P(q_j^{d+1}, q_k^{d+1})$ is the probability of making a horizontal transition from sub-states j and k of q_i^d. The vertical transitions $\Pi^{q^d} = \{\pi^{q^d}(q_i^{d+1})\} = \{P(q_i^{d+1}|q^d)\}$ can be seen as the probability of entering state q_i^{d+1} from its parent state q^d and is related to the initial probabilities of the classical HMM as the probability that state q^d will initially activate the state q_i^{d+1}. Each production state (state at lower level) is parametrized by an observation model B^{q^D}. An example of Hierarchic HMM is presented in Figure 5.2.1. The HHMMs were applied to activities recognition from movement trajectories in [NPVB05] using shared structures. The results show improvement in recognition performance over flat HMM. However, one of the main drawbacks of these fully hierarchical models is the high number of parameters which induce the need of a large amount of learning data.

With regard to the complexity of some structures, specifically in complex content such as video, the classical "flat" HMMs are limited. Introducing a hierarchy of HMMs is one

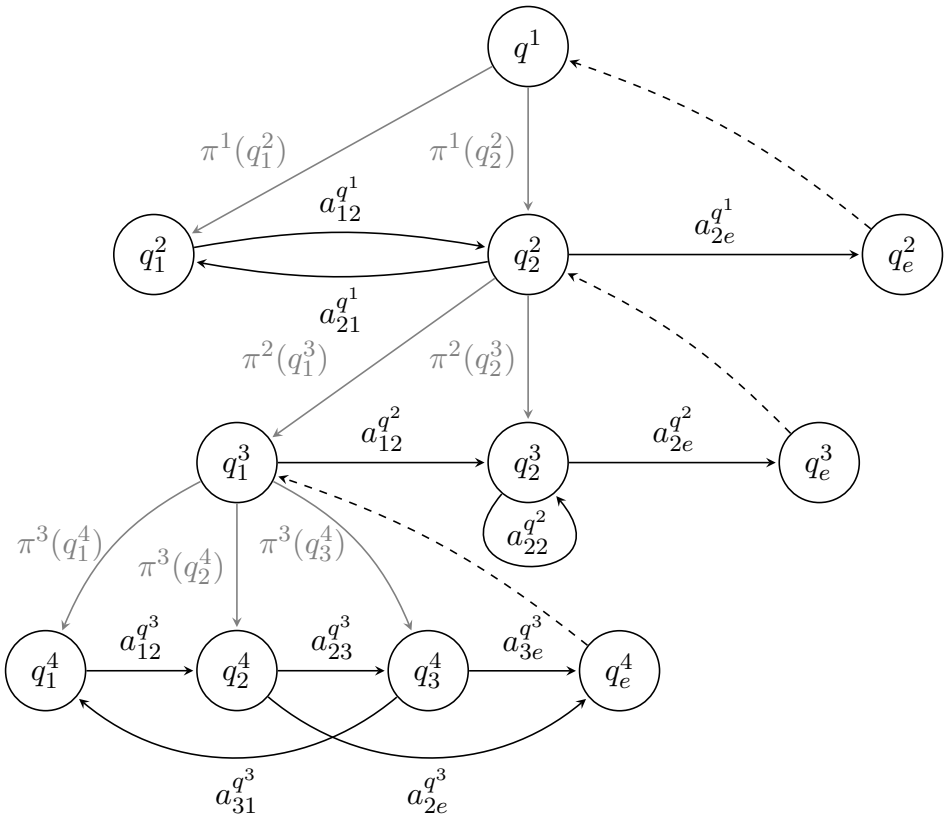


Figure 5.2.1: Example of a Hierarchical HMM.

solution to deal with this limitation. The method presented in [KGG+03] combines audio and visual cues for tennis video structuring. The video stream is automatically segmented into shots by detecting cuts and dissolve transitions; thus the basic temporal unit is the video shot. There are four principal view classes in a tennis video: global, medium, close-up and audience. In a global view, most of the image corresponds to the tennis court whereas close-up views are often a camera tracking a player. To define a mesure permitting to identify a global view, the first step is the selection of a keyframe K_{ref} representative of a global view. Then the visual similarity measure between a keyframe K_t and K_{ref}, denoted $v(K_t, K_{ref})$ or just v_t, uses two features: a vector of dominant colors F and its spatial coherency C, and the activity A that reflects camera motion during a shot. Formally, defining weights w_1, w_2 and w_3 respectively for the spatial coherency, the distance function and the activity, the visual similarity measure is defined as:

$$v(K_t, K_{ref}) = w_1|C_t - C_{ref}| + w_2\, d(F_t,\, F_{ref}) + w_3|A_t - A_{ref}| \tag{5.2.1}$$

The audio is represented by a binary vector describing which audio classes, among speech, applause, ball hits, noise and music, are present in the shot. HMMs are used to merge these audio-visual information and model each of these four events: first missed serve, rally, replay, and break. Using only prior information about tennis content and

editing rules, a higher-level HMM represents the hierarchical structure of a tennis match i.e. the points, games and sets.

Y. A. Ivanov and A. F. Bobick have proposed in [IB00] a probablistic syntactic approach to the detection and recognition of temporally extended activities and interactions between multiple agents. The main idea of this work is to divide the recognition problem into two levels. Independent probabilistic event detectors propose candidate detections of low-level features which are used as the input stream for a stochastic context-free grammar parsing mechanism. A Stochastic Context-Free Grammar (SCFG) is a probabilistic extension of a Context-Free Grammar. This extension is implemented by adding a probability measure to every production rule. The Context-Free property of the grammar means that rules are conditionnaly independent. Therefore, the probability of generating a particular complete derivation is the product of the probabilities of the rules involved in the derivation. In this method, low-level detectors with the ability of generating detection events and some characterization of the detection confidence are needed. For the gesture recognition task presented in this approach, HMMs are used as low-level event detectors. The HMMs are used to modeled simple hand trajectories such as "up-down", "left-right", etc. Each HMM picks out the part of the trajectory which is the most similar to the primitive gesture it has been trained on, estimating the likelihood of the corresponding model. The outputs of the HMMs are used to build a set of discrete events which are the input of the parser. The parser attempts to find the most likely interpretation of the event set. The approach is more efficient in domains where the atomic events can be clearly defined.

5.3 Coupled hidden Markov models

Many interesting systems are composed of multiple interacting processes. In the classical HMMs framework, three classes of solutions could be used to deal with multiple process. The Direct Identification Model is performed by one HMM using an observation stream build as the concatenation of all the process streams [AOJS96],[AB95]. The second solution, than can be named «Separated Identification Model», would process each stream separately and then use expert rules, a combination of probabilities or fuzzy scores to take the final decision [FD96]. The last solution would be to build a «Product Model» of separate HMMs. The observation will be here again the concatenation of the separate streams as in [Jou95].

The first attempts to build a Coupled HMM are the master-slave HMM proposed in [BDMGO92]. Cognitive research has shown that both acoustic and articulatory information are important for the auditive recognition process for a human. Being in the context of automatic speech recognition from videos, R. André-Obrecht et al., in [AOJS96], focus on a master-slave HMM to fuse the acoustic and articulatory information. The labial signals are composed of the three main characteristics of lip gestures: horizontal width, vertical height and area of the internal lip opening. The acoustic is represented by 8 Mel Frequency scale Cepstrum Coefficients (MFCC) and the energy. The derivatives of both labial and acoustic characteristics are computed. The authors compare two kinds of models: the reference model M_{ref} and the Master-Slave model $M_{art} - M_{acous}$. In the M_{ref} model, a HMM model an elementary unit, called the pseudo-diphone, which is a steady part of a phone or a transition between two phones. Each word of the application

is described with these units.

The Master-Slave model $M_{art} - M_{acous}$ defines two different models. The master M_{art} model for the articulatory analysis is composed of three states and three pdfs and takes the labial coefficients as input. The slave model M_{acous} is defined similarly as the M_{ref} model but its parameters (transition matrix and pdfs) are probabilistic functions of the state ot the master model. The observations of the M_{acous} model are only the acoustic parameters. The experiments confirm the perception results as using both acoustic and labial parameters increase the performance of the M_{ref} model. The comparison between the M_{ref} model and the Master-Slave model $M_{art} - M_{acous}$ leads to two conclusions. The Master-Slave model gives better results than the M_{ref} model alone when using both acoustic and labial parameters. However, when adding a "cocktail party" noise to the acoustic signal, the performance of the Master-Slave model is still greater on the training set but are poorer on the test set. The authors conclude that it is due to the higher number of parameters in the Master-Slave model which would necessitate a larger learning database to achieve good performances. This is one common problem when dealing with more complex models than the classical HMM.

This approach of Master-Slave model is based on the idea of coupling several channels of data. When modeling data coming from more than one channel, one can try to use a single state variable while using a multivariate pdf on the output variables. But this approach won't model the interactions between the processes creating the different channels of data. As it has been seen previously in the work from R. André-Obrecht et al., modeling this interaction can be very important. This has been generalized as the Coupled Hidden Markov Model (CHMM) in [BOP97].

The problem of coupling can lead to two main ideas: the source separation problem and the sensor fusion problem. An example for the source separation problem may be the audio recording of voices from unrelated conversations at a cocktail party. The sensor fusion problem is much more complex: in this situation multiple channels carry complementary information about different components of a system. This is the case of the previously presented analysis of speech using both audio and viusal features extracted from lip-tracking. Using a dynamic programming method the computation of the forward and backward variables in a CHMM of C chains can be achieved in $O(TN^{2C})$. Since the posterior probability mass of an HMM is not distributed evenly among all possible state sequences, Brand et al. introduce a N-heads dynamic programming approach in $O(T(CN)^2)$ using only the best state sequences, avoiding computation cost along the low-probability sequences. Considering a two-chain CHMM, each state sequence is double-tracked with a head in one HMM and a sidekick in the opposite one. A sequence of {head,sidekick} pairs is a «path» and the subsequence leading up to any particular {head,sidekick} pair the «antecedent path». Therefore, for the computation of the forward-backward analysis, at each step the aim is to obtain the MAP mass {head,sidekick} pairs given all antecedent paths. This is done by choosing in each chain the MAP state given all antecedent paths, this will be the sidekick for the heads in other chains. Then for each head, calculate the new path posterior given antecedent paths and sidekicks. With full coupling of a large number of HMMs the computation may have a high cost and the approximation of the propose method will be weaker. However, the authors suggest that most of the systems need to be coupled only through a limited set of pairwise interactions.

Nevertheless, the coupled HMMs require stronger computational overload compared

to other HMMs formalisms. Furthermore, when the environment represents a complex audio-visual scenery, it is difficult to define a priori master-slave or a limited set of pariwise interactions. The early fusion in the observation space still remains seducing.

The experiments on real data presented in this paper are the recognition of T'ai Chi Ch'uan gestures. Using a stereo vision blob-tracker for 3D hand-tracking for both hands, the N-heads Coupled HMMs method yeild the highest quality models in the least time, with robustness to initial conditions, good discrimination properties and good generalization to novel examples. The model introduced by Brand et al. was used in [ORP98] for a video surveillance task. In videos capturing a pedestrian scene, the aim is to detect interactions such as «follow, reach and walk together», «approach, meet and go on», «approach, meet and continue together», «change direction to meet, approach, meet and go together» and «change direction to meet, approach, meet and go on separetely». In this task, CHMMs clearly outperforms classical HMMs.

5.4 Segmental hidden Markov models

The Segmental Hidden Markov Models (SHMM) were introduced in [GY93]. The authors aim to deal with one of the major drawback in the use of HMMs i.e. invalidity of the «Independence Assumption». Therefore, they review some methods which have tried to handle correlation between observation vectors i.e. to define a segment of observations. We can for example cite the Variable Frame Rate (VFR) analysis [PP91]. The idea of which is to assume that a segment is well represented by the first observation. Other approaches, like the Stochastic Segment Model [OR89] have also been studied. All the methods including the one introduced by [GY93] have either low impact on performances or need a large number of parameters. Moreover, the estimation assuming no bounds on segment length yeilds $2^T - 1$ possible segmentations to search over. When learning the models parameters both the Viterbi algorithm and Baum-Welsh algorithm have a complexity of $O(T^2)$. This can be reduced by considering a maximum segment length t_{max} to $O(Tt_{max})$.

The model in [ODK96] also addresses the problem of variable length sequences of observation vectors by generalizing the previous segmental approaches. The fundamental difference between SHMMs and HMMs is that in SHMMs a hidden state is associated to a complete sequence of observations $O_{1:t}$, called segment, instead of a unique feature vector o_t. Therefore, a general segment model defines a joint model for a random-length sequence of observations generated by the hidden state q_i of the SHMM:

$$p(o_t, \ldots, o_{t+l}, l|i) = p(o_t, \ldots, o_{t+l}|l, i)p(l|i) = b_{i,l}(O_{t:t+l})p(l|i) \tag{5.4.1}$$

Hence, $p(l|i)$ is a duration distribution, giving the likelihood of segment length l for state q_i and $b_{i,l}(O_{t:t+l})$ is the emission probability distribution over a segment, conditioned on the segment length and the hidden state.

From a generative point of view, SHMMs can be seen as a Markovian process where a hidden state emits a sequence of observations, whose length is governed by a duration model before transiting to another state. In the case of HMMs: at a given time the process is in a given state and generates one observation symbol and then transits to another state.

For SHMMs, at a given moment, the stochastic process enters a state and remains there according to a probability given by the state duration model. A sequence of observations is generated, instead of a single observation. Then, the process transits to a new state with a transition probability, as in HMMs, and so on until the complete sequence of observations is generated.

The application to video has been for example shown for tennis video parsing approach [DGG08] where, thanks to SHMMs, different modalities can be processed with their native sampling rates and models. This has also been applied to baseball videos structuring with a Multi-Channel Segmental Hidden Markov Model (MCSHMM) in [DF08] which integrates both hierarchical and parallel structure within the model. A set of mid-level semantic structures, as rudimentary semantic building blocks, is defined. They should be frequent, repeatable and relatively well-defined. In this work, it corresponds to cameras views and play types. The video mining problem can be defined as an inference problem. The objective is to infer mid-level semantic structures from visual features. Once again, despite the gain in performance, these models have a much higher computational cost and number of parameters than the flat HMM.

Conclusion

In this chapter, we have introduced the Hidden Markov Model (HMM) formalism as well as more complex models than the classical HMM. Each of the elaborate models extends the possibility of the classical HMM in a specific direction. The Hierarchical HMMs (HHMMs) aim at capturing more complex events that cannot be modeled by a single state. The Coupled HMMs (CHMMs) are used for modeling multimodal events. Finally, the Segmental HMMs (SHMMs) tackle the problem of modeling variable length observations.

Each of these methods has shown interesting properties but this was not without a significant additional cost in terms of number of parameters and computation. The Multi-Channel Segmental Hidden Markov Model (MCSHMM) proposed in [DF08] is a combination of all the elaborate models presented in this chapter. The Multi-Channel HMMs can actually give multiple mid-level semantic labels but use the coupling approach of the CHMM for exploring the interaction between the semantic structures.

In next chapter, we will specifically focus on the formalism used for indexing of instrumental activities in our videos, which is a hierarchical two level Hidden Markov Model (HMM). We had to take into account both the complexity of the data we have to analyze and the lack of large amount of data which makes very complex HMMs unsuitable. For example, instead of using a SHMM we will define a pre-segmentation. Therefore, without adding any complexity to the HMM model, we will not use frames as observations but segments.

Chapter 6

Design of a two-level Hierarchic Hidden Markov Model for Activities Segmentation

Introduction

In this chapter, we first introduce in 6.1 the proposed two-level Hierarhic Markov Model which is the core of our video structuring method. The video structuring consists in labeling units of the video as one of the IADL defined by the medical practitioners. In order to define meaningful units of the video to be labeled, we will introduce a temporal motion-based segmentation in 6.3. This segmentation yields segments that can be interpreted as viewpoints. The video structuring relies on efficient descriptors of the video content, we will present these descriptors in section 6.4.

6.1 The two-level structure

In order to take into account both the complexity of our data and the lack of large amount of training data for learning purposes, we propose the following model. If we abstract our problem of recognition of daily activities in the video to its simplest core, we can draw an equivalence between an activity and a hidden state of an HMM. The connectivity of the HMM can, at this level, be defined by the spatial constraints of the patient's environment when it is known. The easiest way is to design a fully connected HMM and train the inherent state-transition probabilities from the labeled data. Unfortunately, the ADL we consider are very much heterogeneous and often very complex, therefore the suggested equivalence between an activity and a hidden state cannot hold together.

Hence, we propose a two-level Hierarchical HMM (HHMM). The activities that are meaningful to the medical practitioners are encoded in the top-level HMM, the set of possible states is thus defined accordingly. We also introduce a reject state "None" to model non-meaningful observations from doctors' point of view. Thus defined, the top-level HMM contains the transitions between "semantic" activities including the reject class. A bottom-level HHM models an activity with m non-semantic states, as in [SPL$^+$07]. We fix the parameter m as 3, 5 or 7 for ADL states and 1, 3, 5 or 9 for the reject class "None"

in our experiments. The overall structure of the HMM is presented in Figure 6.1.1, with 3 states at the bottom level. Dashed circled states are non emitting states.

6.1.1 Top-level HMM

In this work, the actions of interest are the IADLs defined by the medical practitioners. The set of activities evolves during the experiments in chapter 7, starting from a limited set in the first experiments to the complete set of activities in the final "large scale" experiment. The top-level HMM represents the relations between these actions. We denote the set of states at this level as $Q^0 = \{q_1^0, \ldots, q_{n_0}^0\}$ and transitions matrix $A^0 = (a_{i,j}^0)$, where n_0 is the number of activities. In this work, constraints were specified over the transitions between these activities in the first experiment. But such restrictions are very difficult to know a priori when addressing a larger set of activities and when analyzing a large set of videos where the physical constraints of each patient's house are different. Moreover, the IADLs a patient is asked to fulfill depend very much on his condition and their sequencing cannot be fixed for all patients in the same way. Hence, in the remaining experiments we design the top level as a fully connected HMM on all the remaining experiments. We consider equiprobable transitions from activities states to one another, hence $\forall i, j : a_{i,j}^0 = \frac{1}{n_0}$. The states of the top-level HMM are denoted in Figure 6.1.1 as "Act" for the sake of simplicity.

6.1.2 Bottom-level HMM

For each activity in the top-level HMM a bottom-level HMM is defined with the set of states $Q_i^1 = \left\{q_{i_1}^1, \ldots, q_{n_1^i}^1\right\}$ with $n_1^i = 3, 5 \text{ or } 7$ for IADL states and $n_1^i = 3, 5, 7 \text{ or } 9$ states for the reject class "none" in our experiments. The state transition matrices A_i^1, for $i = 1, \ldots, n_0$ also correspond to a fully connected HMM: $a_{i\,k,l}^1 \neq 0$, at initialization, for $k = 1, \ldots, n_1^i$ and $l = 1, \ldots, n_1^i$. For the video stream not to be over-segmented the loop probabilities $a_{i\,k,k}^1$ have to be initialized with greater values than other transition probabilities: $a_{i\,k,k}^1 > a_{i\,k,l}^1 \forall k \neq l$, this will be explicitly defined in our experimental study, see chapter 7. Activities are more likely to involve several successive observations rather than just one, this explains the choice for such a higher loop probability. At the bottom level, each non semantic state models the observation vector o by a Gaussian Mixture Model (GMM) which has been introduced in section 5.1.1, see equation (5.1.7). The GMM and the transitions matrix of all the bottom-level HMMs are learned using the classical Baum Welsh algorithm [Rab89] with labeled data corresponding to each activity.

6.2 Implementation, training and recognition

HMM is a well studied subject for today, and a lot of implementations of HMMs are available in open source software. In our implementation of the designed two-level HHMM, we used the HTK library [YY94]. This probably is the mostly used software for HMMs.

For training the bottom-level HMMs we use the Baum-Welsh algorithm. We have presented a detailed description of this algorithm in section 5.1.4. We consider the continuous HMM to model observations probability with GMM, see (5.1.7). The observations will be detailed in section 6.4. In the Baum-Welsh algorithm, an initialization is needed.

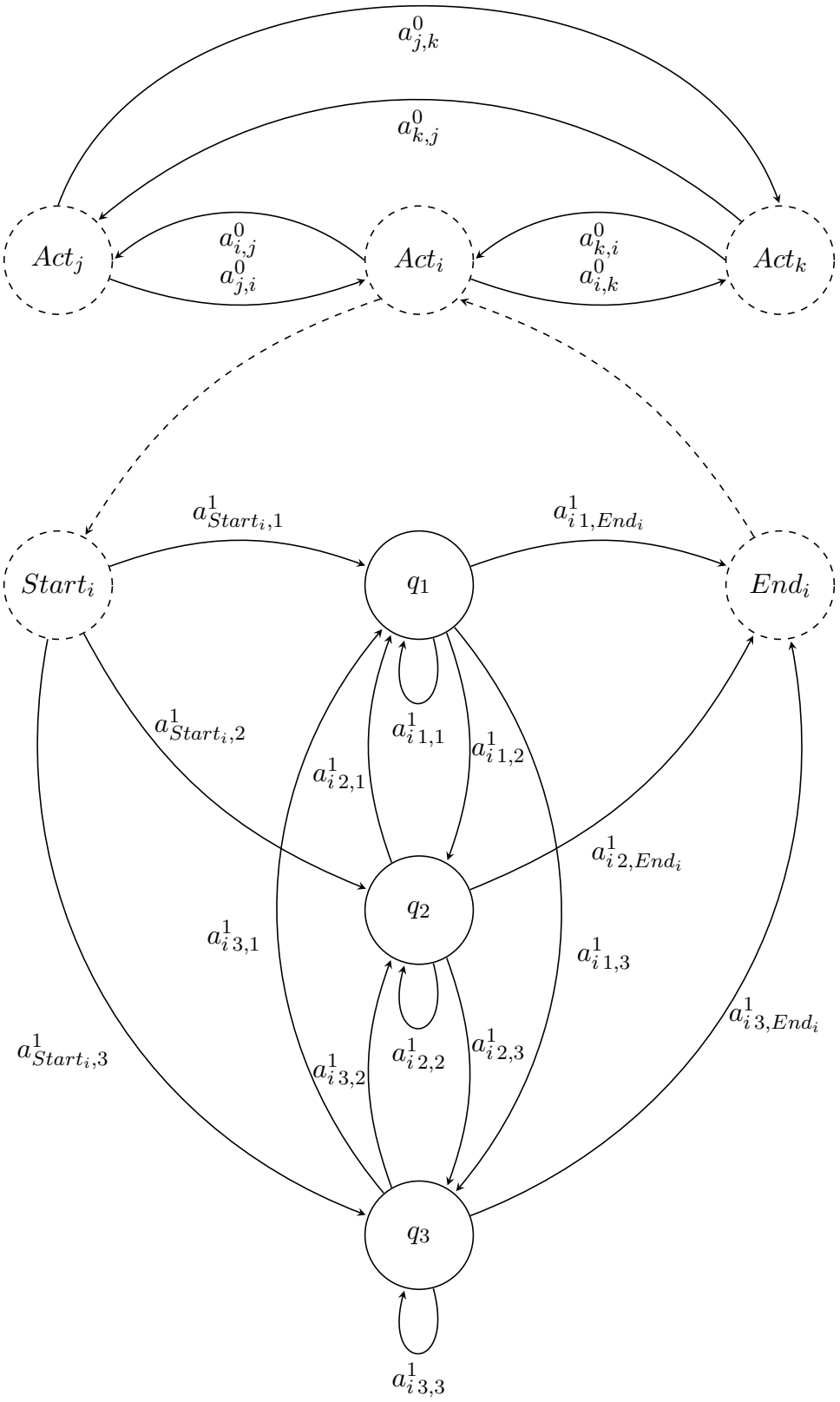


Figure 6.1.1: The HHMM structure.

The number of states m is fixed and will not be changed during the learning process. The transition probabilities are initialized with greater values for loop probabilities as stated in previous section, the exact values are precised in each experiment presented in chapter 7. We used a fixed number of Gaussian components for the observation model. The HTK Baum-Welsh training implementation may discard low-weight Gaussian component in a mixture. Precisely, the component l of the GMM is discarded if the re-estimated weight $\overline{w_{il}}$, see (5.1.32), is lower than a minimal "threshold" weight. The initialization of the GMM can be done as a "flat-start" i.e. setting all means and variances to be equal to the global mean and variance. However, since the Baum-Welsh would only find a local optimum and that the amount of learning data in our context is not very large, a more detailed initialization is possible by using iterative Viterbi alignments.

For the recognition, the Viterbi algorithm is used. The Viterbi alogrithm has been detailed in section 5.1.3. The HTK implementation makes efficient use of the "token passing" paradigm to implement a beam pruned Viterbi search. Details on the HTK library can be found in the HTK Book [YEK+97].

6.3 Temporal pre-segmentation into "viewpoints"

The video structuring will rely on an analysis unit. We want to establish a minimal unity of analysis which is more relevant than the video frames. The objective is to segment the video into the different viewpoints that the patient provides by moving throughout his home. In contrast to the work in [HWB+06] where the segmentation is based on a fixed key-framing of the video, our goal is to use the motion of the patient as the criterion for segmentation. This viewpoint segmentation of our long uninterrupted video sequences may be considered as an equivalent to shots in edited video sequences.

We will present the designed motion based segmentation of the video. The viewpoints based segmentation relies on the estimation of the global motion. Therefore we will first define the complete affine model in 6.3.1. Using this model, the computation of corners trajectories is presented in 6.3.2 and finally the definition of the segments is given in 6.3.3.

6.3.1 Global motion estimation

Since the camera is worn by the person, the global motion observed in image plane can be called the ego-motion. We model the ego-motion by the first order complete affine model and estimate it with a robust weighted least squares by the method reported in [KBP+05]. The parameters of (6.3.1) are computed from the motion vectors extracted from the compressed video stream where one motion vector $\overrightarrow{d_i} = (dx_i, dy_i)$ is extracted per i-th image block and is supposed to follow the model

$$\begin{pmatrix} dx_i \\ dy_i \end{pmatrix} = \begin{pmatrix} a_1 \\ a_4 \end{pmatrix} + \begin{pmatrix} a_2 & a_3 \\ a_5 & a_6 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \tag{6.3.1}$$

with (x_i, y_i) being the coordinates of a block center.

The models parameters $(a_1, a_2, a_3, a_4, a_5, a_6)$ are stored in the column vector θ and are computed by the following matrix product:

$$\theta = (a_1, a_2, a_3, a_4, a_5, a_6)^T = \left(H^T W H\right)^{-1} H^T W H \tag{6.3.2}$$

Let N be the number of blocks, Z is the column vector of size $2N$ containing the motion compensation vectors:

$$Z = (dx_1, ..., dx_N, dy_1, ..., dy_N)^T \tag{6.3.3}$$

Let H be the observations matrix of size $2N \times 6$:

$$H = \begin{pmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & x_N & y_N \end{pmatrix} \tag{6.3.4}$$

and W be the diagonal weights matrix defined by the Tukey operator [DBP01] of size $2N \times 2N$. The weights enable minimizing the errors that can arise for example on blocks near the image border which often induce chaotic motion.

$$W = \begin{pmatrix} w_1 & 0 & \cdots & 0 & 0 \\ 0 & w_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & w_{2N-1} & 0 \\ 0 & 0 & \cdots & 0 & w_{2N} \end{pmatrix} \tag{6.3.5}$$

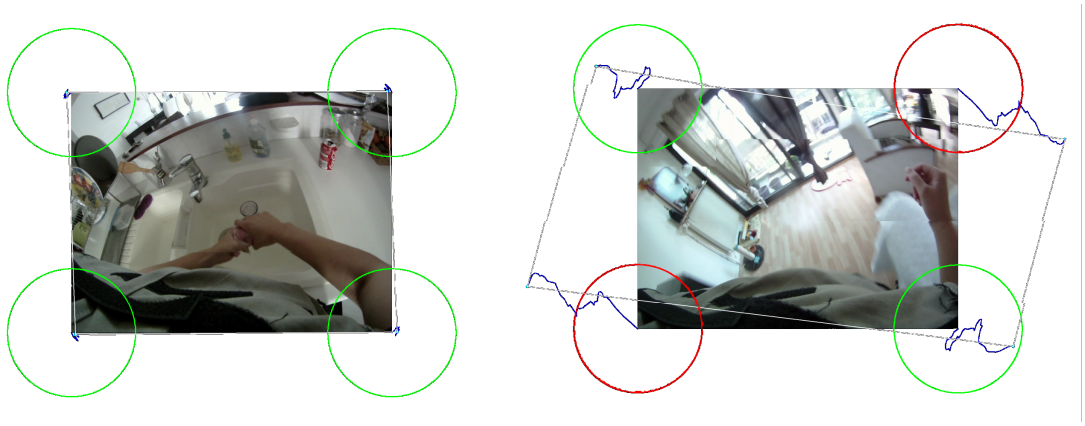
6.3.2 Corners trajectories

To split the video stream into segments, we compute the trajectories of each corner using the global motion estimation previously presented. As the motion model enables the computation of the motion vector $\overrightarrow{d_i} = (dx_i, dy_i)$ of any point (x_i, y_i) between frame t and $t+1$ we can write:

$$\begin{pmatrix} x_i^{(t+1)} \\ y_i^{(t+1)} \end{pmatrix} = \begin{pmatrix} x_i^{(t)} \\ y_i^{(t)} \end{pmatrix} + \begin{pmatrix} dx_i^{(t)} \\ dy_i^{(t)} \end{pmatrix} \tag{6.3.6}$$

From (6.3.1), we can write the integrated motion in a matrix from as:

$$\begin{pmatrix} x_i^{(t+1)} \\ y_i^{(t+1)} \\ 1 \end{pmatrix} = \begin{pmatrix} a_2 + 1 & a_3 & a_1 \\ a_5 & a_6 + 1 & a_4 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i^{(t)} \\ y_i^{(t)} \\ 1 \end{pmatrix} \tag{6.3.7}$$



(a) Corner trajectories while the person is static. (b) Corner trajectories while the person is moving.

Figure 6.3.1: Example of corners trajectories.

For each frame the distance between the initial and the current position of a corner is calculated. We denote w as the image width and s as a threshold on the frame overlap rate. A corner is considered as having reached an outbound position once it has had at least once a distance greater than $s \times w$ from its initial position in the current segment. These boundaries are represented by green and red (when the corner has reached an outbound position) circles in Figure 6.3.1.

6.3.3 Definition of segments

Each segment S_k corresponds to a temporal interval $S_k = [t_k^{min}, t_k^{max}]$ which aims to represent a single "viewpoint". This notion of viewpoint is clearly linked to the threshold s, which defines the minimal proportion of an image which should be contained in all the frames of the segment.
We define the following rules:

- a segment should contain 5 frames at minimum
- a segment should contain 1000 frames at maximum
- the end of the segment is the frame corresponding to the time when at least 3 corners have reached, at least once, an outbound position. This condition is obvious from simple geometrical consideration, e.g strong motion once on the left and then on the right.

The key frame is then chosen as the temporal center of the segment, see examples in Figure 6.3.2.

Hence, the proposed viewpoints based segmentation using the estimated motion model serves for three goals:

- the segments define the analysis unit i.e. the minimal unit to be labelled by our HHMM;

Figure 6.3.2: An example of key frame (center) with the beginning (left) and ending (right) frames of the segment.

- estimated motion parameters are used for the computation of dynamic features which will be used in the HHMM description space;
- the key frames extracted from motion-segmented "viewpoints" are the basis for extraction of spatial features which will also serve for the HHMM description space.

We will now focus on the definition of all features and the design of the global description space candidates. The feature vectors of the description space will be the observations for the HHMM.

6.4 Observations for hierarchical hidden Markov model

The description space aims to describe the different modalities that can be extracted from the video stream. We first introduce descriptors that characterize the motion within the video recorded in 6.4.1, then define the audio analysis in 6.4.2 and finally present static descriptors that gather the context of the patient's environment in 6.4.3. The fusion of all these features will be presented in 6.4.4.

6.4.1 Motion description

The motion contains interesting information that can be used to characterize an activity. Therefore, a set of descriptors for several properties of the motion will be defined. This choice corresponds to the need to distinguish between various activities of a patient which are naturally static (e.g. reading) or dynamic (e.g. hoovering).

6.4.1.1 Global and instant motion

The ego-motion is estimated by the global motion analysis presented in section 6.3.1. The parameters a_1 and a_4 are the translation parameters in (6.3.1). We limit our analysis to these parameters, since as in the case of wearable cameras, they better express the dynamics of the behavior, and pure affine deformation without any translation is practically never observed.

The instant motion histogram is defined for a frame f_t as the histogram of the log-energy of each translation parameter H_{tpe}, as expressed in (6.4.1), defining a step s_h and using a log scale. This histogram characterizes the instant motion; which is computed

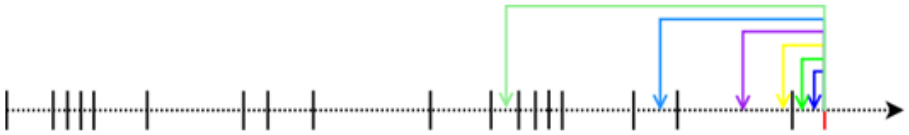


Figure 6.4.1: The number of cuts (black lines) is summed to define the value of each bin. In this example: $H_c[1]$=0, $H_c[2]$=0, $H_c[3]$=1, $H_c[4]$=1, $H_c[5]$=2, $H_c[6]$=7.

for each frame. This feature is designed to distinguish between "static" activities e.g. "knitting" and dynamic activities, such as "sweeping".

$$H_{tpe,j}[i] = \begin{cases} 1 \text{ iff } a_j(t) \in B_i \\ 0 \text{ otherwise} \end{cases} \quad \text{with the bins } B_i \text{ defined as}$$

$$\begin{array}{lll} a_j \in B_1 & \text{iff} \quad log\,(a_j^2) < i \times s_h & \text{for } i = 1 \\ a_j \in B_i & \text{iff} \quad (i-1) \times s_h < log\,(a_j^2) < i \times s_h & \text{for } i = 2\,..\,N_e - 1 \\ a_j \in B_{N_e} & \text{iff} \quad log\,(a_j^2) \geq (i-1) \times s_h & \text{for } i = N_e \end{array} \tag{6.4.1}$$

where $j = 1, 4$.

The feature for a video segment S_k is an averaged histogram on all its frames: $\overline{H_{tpe,j}}$, $j = 1, 4$ for horizontal and vertical translations parameters, respectively a_1 and a_4. The global instant motion feature is the concatenation of both: $\overline{H_{tpe}} = \left(\overline{H_{tpe,1}}, \overline{H_{tpe,4}}\right)$.

$$\overline{H_{tpe,j}}[i] = \frac{Card\{t \in S_k | a_j(t) \in B_i\}}{t_k^{max} - t_k^{min}} \tag{6.4.2}$$

with the bins B_i defined as in (6.4.1), and $j = 1, 4$.

We denote $H_{tpe}(x) = H_{tpe,1}$ the histogram of the log-energy of horizontal translation, and $H_{tpe}(y) = H_{tpe,4}$ the histogram of the energy of vertical translation observed in image plane. The number of bins is chosen empirically and equally with regards to x and y, $N_e = 5$, the threshold s_h is chosen in such a way that the last bin corresponds to the translation of the image width or height respectively.

6.4.1.2 Historic of global motion

Another element to distinguish static and dynamic activities is the motion history. On the contrary to the instant motion, we design it to characterize long-term dynamic activities, such as walking ahead, vacuum cleaning, etc. This is estimated by computing a "cut histogram" H_c. We design it as a histogram of N_c bins. Each i-th bin contains the number $H_c(i)$ of cuts (according to the motion based segmentation presented in section 6.3) that happened in the last 2^i frames, see Figure 6.4.1. The number of bins N_c is defined as 8 in our experiments providing a history horizon of 256 frames, which represent almost 9 seconds of our 30 fps videos. Such a cut histogram is associated to each frame in the video. The descriptor $\overline{H_c}$ associated to a segment is the average of the cut histograms of the frames belonging to the segment.

6.4.1.3 Local motion

All the previous motion descriptors focus on the global motion which is very important as it provides a characterization of the ego-motion. However, the residual motion may reveal additional information, such as the occurrence of a manual activity or the presence of a moving object or a person in the visual field of the patient. In order to describe the residual motion we introduce a descriptor which is computed on each block of a 4x4 grid segmenting the image. The value representing each block b of width N and height M is computed, as presented in (6.4.3) as the Root Mean Square (RMS) of the difference $\overrightarrow{\Delta d_{k,l}} = (\Delta dx_{k,l},\, \Delta dy_{k,l})^T$ between motion vector extracted from compressed video stream and the one obtained from the estimated model (6.3.1). The residual motion descriptor RM of the whole frame has therefore a dimensionality of 16.

$$RM_b = \sqrt{\frac{\sum_{k=1,l=1}^{k=N,l=M} (\Delta dx_{k,l}^2,\, \Delta dy_{k,l}^2)}{NM}} \tag{6.4.3}$$

6.4.2 Audio

The audio descriptors were developped by the IMMED project partners at IRIT. In this section, we will breifly define the characteristics extracted from the audio stream.

The particularity of the contribution in the design of the audio description space consists in the use of low-level audio descriptors. Indeed, in the home environment with ambient TV audio track, noise produced by different objects the patient is manipulating, conversations with the persons, etc, all are good indicators of the patient's activity and its location. In order to characterize the audio environment, different features are extracted. Energy is used for silence detection. 4 Hz energy modulation and entropy modulation give voicing information, being specific to the presence of speech. The number of segments per second and the segment duration, resulting from a "Forward-Backward" divergence algorithm [AO88], are used to find harmonic sound, like music. Then, a first set of features is characteristic of these particular sounds: speech, music, noise, silence and periodic sounds [PAO06].

Two other features are added and describe the water flow and the vacuum cleaner use. This indexation method results from the audio research on the IMMED project and is beyond the scope of this manuscript. Breifly, an original low level descriptor call spectral cover is used in a system based on thresholds and allows to recognize some specific sound events.

Finally, the complete set of audio descriptors is composed of 7 possible events: speech, music, noise, silence, periodic sounds, water flow, vacuum cleaner. As the last two descriptors were developed during the project, we used a subset of the complete set of audio descriptors in some experiments.

6.4.3 Static descriptors

Static descriptors aim to characterize the instantaneous state of the patient within his/her environment. The first static descriptor is the estimated localization, the second defines the local spatial and color environment using the MPEG-7 [MSS02] descriptor "Color Layout".

6.4.3.1 Localization

The localization descriptor was developped by the IMMED project partners at IMS. We briefly define the main ideas of the proposed localization estimation method. The reader is referred to [DMB11] for a full description of the approach.

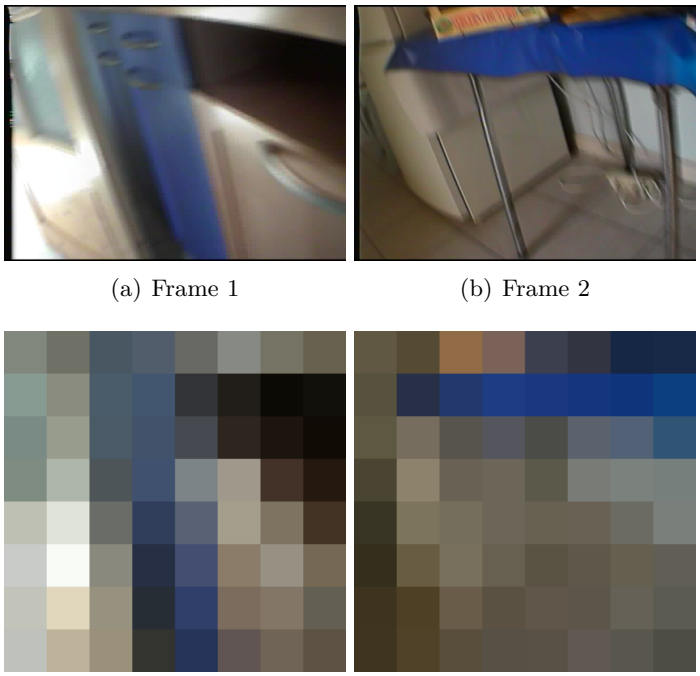
The IMS partners use the method of Bag of Visual Words [LSP06] to represent an image as a histogram of visual words. Low level visual information contained within an image is captured using local features SURF [BETVG08] descriptors. Descriptors are quantized into visual words using a pre-built vocabulary which is constructed in a hierarchical manner [JT05]. The Bag of Words vector is built by counting the occurrence of each visual word. Due to a rich visual content, the dimensionality of such histograms is very high (we used a 1111 word dictionary in our context). Kernel PCA [SSM98] is used to reduce these descriptors to several hundreds dimensions [BDLR$^+$06] by embedding them into the subspace of highest variance in the Reproducing Kernel Hilbert Space associated to the kernel. The intersection kernel was used in the experiments. Initial non-linear relationships in the original space are therefore represented by a simple scalar product between the embeddings. A linear kernel SVM [Bur98] classifier is then applied on the embeddings in a one-vs-all approach to produce the location vector, representing a 1 for the detected class, and 0 for other classes. This estimation is used as a feature in the activities recognition process.

We defined a set of 6 generic localization classes according to the different rooms existing in the first video recordings available. The generic localizations we defined are: “bathroom”, “bedroom”, “kitchen”, “living room”, “outside” and “other”. Therefore the localization feature Loc will be built as an histogram of 6 bins where each bin contains the output of the SVM classifier for the corresponding class. The descriptor $\overline{Loc}$ associated to a segment is the average of the localization features of the frames belonging to the segment.

Each room being specific for each patient, a localization bootstrap is recorded at each patient’s house. This video footage is composed of a brief visit of the house and is annotated by the medical partners according to the defined taxonomy of localizations. The frames of this video are used to learn the model of each class.

6.4.3.2 Color Layout Descriptor

Using the extracted key frames representing each segment, a simple description of the local spatial and color environment is expected. In this choice we seek for the global descriptors which characterize the color of frames while still preserving some spatial information. The MPEG-7 Color Layout Descriptor (CLD) presented in 4.1.4, proved to be a good compromise for both of them [QBPM$^+$08]. It is computed on each key frame and the classical choice [MSS02] of selecting 6 parameters for the luminance and 3 for each chrominance was adopted. This descriptor gives a coarse but yet discriminative visual summary of the local environment. Examples of CLD computed on frames extracted from videos recording with the first recording device prototype are presented in Figure (6.4.2).



(a) Frame 1 (b) Frame 2

(c) CLD 1 (d) CLD 2

Figure 6.4.2: Frames extracted from our videos and corresponding CLD.

6.4.4 Descriptors fusion

Hence, for the description of the content recorded with wearable cameras we designed three description subspaces : the "dynamic" subspace has 34 dimensions (5 for each H_{tpe}, 8 for H_c and 16 for RM) and contains the descriptors D = $(H_{tpe}(x), H_{tpe}(y), H_c, RM)$; the "audio" subspace contains a maximum of $k = 7$ audio descriptors p = $(p_1, ..., p_k)$; the "static" subspace contains 18 coefficients, more precisely l = 12 CLD coefficients C = $(c_1, ..., c_l)$ and m = 6 localization coefficients L = $(l_1, ..., l_m)$.
We design the global description space in an "early fusion" manner concatenating all descriptors in an observation vector in R^n space with n = 59 dimensions when all descriptors are used. Thus designed, the description space is inhomogeneous. We will study the completeness and redundancy of this space in a pure experimental way with regard to the indexing of activities in chapter 7, by building a variety of possible partial fusions.

Conclusion

In this chapter we have introduced the proposed two-level Hierarchical Hidden Markov Model (HHMM), where the top level defines the transitions between activities while each bottom-level HMM models one activity. A bottom-level HMM is composed of m non-semantic states, each of them relying on a Gaussian Mixture Model (GMM) for modeling a part of the activity in the description space.

Since the video recording device is worn by the patient, the global motion in the image plane can be seen as the ego-motion. We have presented a complete affine model for the

estimation of the global motion and introduced a motion based temporal segmentation using this model. The segments thus defined will be used as the analysis unit in our experiments.

To describe the content of the video, we have introduced the set of 6 descriptors we will use in the next chapter for the experiments. This set can be decomposed in 3 subsets:

- the "dynamic" subset, containing the 3 motion descriptors. The first one H_{tpe} characterizes the strength on the global and instant motion, the second H_c describing the long-term dynamic and finally the RM descriptor which captures the residual motion.

- the "audio" subset, containing the audio descriptors describing the 7 classes of events: speech, music, noise, silence, periodic sounds, water flow and vacuum cleaner.

- the "static" subset, containing 2 descriptors describing the visual environment of the patient. The first descriptor is the *Loc* histogram which aims to define the possible current location of the patients in the 6 classes: "bathroom", "bedroom", "kitchen", "living room", "outside" and "other". The second one being the CLD descriptor capturing the spatial and color environment of the patient.

We defined description space candidates using an early fusion of some or all descriptors. The observations for the HHMM will be feature vectors of these description spaces.

The next chapter presents the series of experiments we have conducted throughout this thesis. We will investigate the possible bottom-level HMMs configurations, the contribution and potential drawbacks of the proposed segmentation and the efficiency of the proposed descriptors with regard to the task of activities recognition.

Chapter 7

Experiments on Activities Recognition

Introduction

This chapter presents the results of the experiments we have conducted for the task of activities recognition. We will first detail the video corpus in 7.1, and then introduce the evaluation metrics in 7.2. The detail of four experiments and corresponding results are presented in 7.3.

7.1 Video corpus

During the project several recording sessions have been conducted in increasing order of complexity of both environment and activities and also of data volume. We only list here the videos recorded with the final prototype, see section 2.3. This final prototype was used to record videos in MPEG4/AVC format with a resolution of 1280x960 pixels at a frame rate of 30fps. We can distinguish three types of recording session:

- Healthy volunteers recording in a laboratory environment. This corpus contains 13 videos recorded by 3 volunteers for a total of 2 hours of content. The activities present in this corpus represent a small set of possible ADL such as "washing the dishes", "making coffee", "reading", "discuss" and "working on computer".

- Healthy volunteers recording at home. This corpus contains 9 videos recorded by 7 volunteers for a total of 5 hours of content. The set of activities is more complete, some ADL such as "cooking", "hoovering" or "sweeping" which were not performed it the laboratory corpus are present here.

- Patients recording at home. This corpus contains 27 videos recorded by 24 volunteers for a total of 7 hours of content. The set of activities is the widest of all corpora, it contains most of activities present in the doctors paper survey, for example "gardening" and "teeth brushing" which were not present in previous corpora.

The characteristics of each corpus is listed in Table 7.1.1.

Table 7.1.1: Corpora characteristics.

Recording session	Number of videos	Duration				Activities
		Min	Max	Avg	Total	
Healthy volunteers recording in a laboratory environment	13	3 min	17 min	7 min 50 s	1 h 42 min	"Discussing", "Writing", "Washing the dishes", "Making coffee", "Making tea", "Looking in the fridge", "Reading", "Working on computer", "Photocopying", "Drying hands"
Healthy volunteers recording at home	14	6 min	43 min 16 s	30 min 21 s	4 h 52 min	"Cooking", "Moving around", "Hoovering", "Sweeping", "Clearing the table", "Making the bed", "Cleaning shovel", "Washing dishes", "Cleaning garbage", "Body hygiene", "Aesthetic hygiene", "Gardening", "Reading", "Watching TV", "Using computer", "Using coffee machine", "Using cooker", "Using microwave", "Medecine", "Using phone", "Home visit"
Patients recording at home	30	12 min 51 s	33 min 14 s	22 min 18 s	6 h 36 min	"Food manual preparation", "Displacement free", "Cleaning hoover", "Cleaning broom", "Cleaning clear", "Cleaning bed", "Cleaning shovel", "Cleaning dustbin", "Cleaning dishesbyhand", "Hygiene body", "Hygiene beauty", "Hygiene clothes", "Leisure gardening", "Leisure reading", "Leisure watch", "Leisure computer", "Complex machines coffee maker", "Complex machines gas cooker", "Complex machines washing machine", "Complex machines microwave", "Medicines medicines", "Relationship phone", "Relationship home visit"

7.2 Evaluation metrics

We here describe the quality metrics for the evaluation of algorithms and methods. In our system, each observation given as an entry of the HHMM is labeled according to one of the activities. According to the label given manually, each of these automatic labeling can be considered as positive or negative responses. When a classification method has to be evaluated, this information is usually summarized in a confusion matrix, which is a form of contingency table showing the differences between the true and false predictions for a set of labeled examples (usually referred to as ground truth). An example of such a matrix is given below in Table 7.2.1, where the predicate $pred()$ denotes predicted negative (N) or positive (P) label for a given class c and $test()$ denotes the real label (ground truth).

Table 7.2.1: An example of confusion matrix for binary classification

	$test(c) = P$	$test(c) = N$
$pred(c) = P$	True Positives (TP)	False Positives (FP)
$pred(c) = N$	False Negatives (FN)	True Negatives (TN)
column totals	c_P	c_N

Given a classifier and an instance, there are four possible outcomes:

- if the instance is positive and it is classified as positive, it is counted as a *true positive*;
- if it is classified as negative, it is counted as a *false negative*;
- if the instance is negative and it is classified as negative, it is counted as a *true negative*;
- if it is classified as positive, it is counted as a *false positive*.

Given a classifier and a set of instances (the test set), a two-by-two confusion matrix (also called a contingency table) can be constructed representing the misclassifications of the set of instances. This matrix forms the basis for many common metrics that are summarized in Table 7.2.2.

Table 7.2.2: Common performance metrics calculated from a confusion matrix

$FP_{rate} = \frac{FP}{c_N}$	$TP_{rate} = \frac{TP}{c_P}$
$precision = \frac{TP}{TP+FP}$	$recall = \frac{TP}{TP+FN}$
$accuracy = \frac{TP+TN}{c_P+c_N}$	$F - Score = \frac{2}{1/precision+1/recall}$

The precision/recall measures are often used in the literature. Precision decreases when the number of false positives (FP) increases. Recall decreases as the number of false negatives (FN) increases. This usually means that when recall increases precision decreases. To counterbalance these two opposite tendencies, another metric, the so-called F-measure or F-score was defined [VR79].

Table 7.2.3: An example of confusion matrix for multiclass classification

	$test(A) = P$	$test(B) = P$	$test(C) = P$
$pred(A) = P$	TP_A	e_{BA}	e_{CA}
$pred(B) = P$	e_{AB}	TP_B	e_{CB}
$pred(C) = P$	e_{AC}	e_{BC}	TP_C

The previous measures were introduced in the case of binary classifiers. However, in our case each observation is classified by the HHMM according to one of learnt activities. Therefore, we now introduce similar measures in the context of multiclass classifiers. Considering we have only 3 classes A, B and C for simplicity, we can define the confusion matrix as in Table 7.2.3, where TP_c is the number of true positive classifications for class c and e_{cd} is the number of misclassifications as class d when the ground truth was c.

We can now define the metrics recall, precision, F-score for one class in the multiclass evaluation context. The false-positive predictions FP_c of class c can be expressed as:

$$FP_c = \sum_{d \neq c} e_{dc} \tag{7.2.1}$$

and the false-negative predictions FN_c as:

$$FN_c = \sum_{d \neq c} e_{cd} \tag{7.2.2}$$

The precision, recall and F-score measures are then computed similarly as in the binary classification. The true-negative predictions TN_c is the sum of all predictions where class c does not appear either as ground truth or estimation:

$$TN_c = \sum_{d \neq c, b \neq c} e_{db} + \sum_{d \neq c} TP_d \tag{7.2.3}$$

As the number of classes grows the true negative predictions number can grow easily as well, this induces that a confusion matrix of per class accuracy will easily have high values on the diagonal. We will therefore only present confusion matrix of precision, recall and F-score. These measures are only valid when evaluated for each class separately, we will therefore compute the mean precision, recall and F-score as a global measure. We will also use the global accuracy which is defined as the total number of true positives over the total number of observations Tot: $\frac{1}{Tot} \sum_c TP_c$. The metrics precision, recall, F-score will be presented by confusion matrices in the following evaluation of our analysis of semantic activities in videos. Global accuracy will be given and the corresponding chronogram will be plotted.

7.3 Model validation

In this work we conducted four series of experiments:

1. Experiment on a video recording from a person at home with low semantic activities and constraint transitions between activites. We also used ground truth on the localization for learning purpose.

2. Experiment on a set of videos recorded in the lab from healthy volunteers with a small set of semantic activities.

3. Real world application on a set of videos recorded in the home of different people with a large number of activities.

4. Large scale application on a set of videos recordedin the home of more than thirty different people with a large number of activities.

In these experiments the goals are:

1. To choose the optimal HMM architecture from a limited set of possible (1, 3, 5, 7) non-semantic states.

2. To choose the optimal and evaluate the discriminative power of each description space

3. To evaluate contribution and drawbacks of temporal pre-segmentation of videos

4. To evaluate system performance on available real world data.

In the follow up we will present these four series of experiments each time specifying the goal of the experiment, the evaluation protocol, the training and test data sets and giving analysis of the results. Let us denote N_m the number of possible states configuration, N_d the number of descriptors that can be used and N_t the number of possible thresholds we want to evaluate we have a total number of possible configuration T_c:

$$T_c = N_m \times N_t \times (2^{N_d} - 1)$$

If we consider $N_m = 4$, $N_d = 6$ and $N_t = 19$, i.e. a threshold evolving in the range $[0.05; 0.95]$ by steps of 0.05, the total number of configurations equals 4788. Therefore, in each experiment we will only use a subset of this total number of configurations but the influence of all parameters will be studied over the complete set of experiments.

7.3.1 Model validation on a single video recording

Goal of the experiment The goal of this first experiment was to validate the approach for activities recognition on the first wearable video recording available. The recording was done with the FishEye prototype, see section 2.3, at a patient's home. The patient was not asked to perform specific activities but was free to do any activities he would usually do in his house. Therefore this video recording has less semantic activities.

Table 7.3.1: Configuration for best recognition results

Configuration	Mean F-score	Mean recall	Mean precision	Global accuracy
HtpeHcLoc 1 state HMMs	**0.65**	0.67	0.81	**0.82**
HtpeCLDLoc 5 states HMMs	0.59	**0.68**	0.62	0.73
HcLoc 1 state HMMs	0.61	0.65	**0.87**	0.81

Training and test data sets Since this was the only video available at that time we had to train the model on data extracted from this same video. For learning purposes, we use 10% of the total number of frames for each activity. These frames were used to train the localization estimator and the activities HMMs. In this experiment, we used the ground truth localization to train the HMMs. The other features H_{tpe}, H_c and CLD were extracted automatically. The features RM and $Audio$ were not used. The temporal segmentation was performed with a fixed threshold of 0.2 and the only constraint on segment definition was that a segment should last at least 5 frames. The tests were done over the segment observations. The description spaces where built by all possible early fusions yeilding 4 description spaces of isolated descriptors, 6 description spaces of 2 descriptors, 4 description spaces of 3 descriptors and the single complete description space of 4 descriptors. The dataset was composed of 3974 frames used for learning and 310 segments for recognition corresponding to 33 minutes of video.

Evaluation protocol We considered the number of elementary states m=1,3 or 5, for each activity HMM. The initial looping probabilities A^1_{ii} were set to 0.9 for 1 state HMM, 0.8 for 3 states HMM and to 0.7 for 5 states HMM. The 7 different activities were "moving in home office", "moving in kitchen", "going up/down the stairs", "moving outdoors", "moving in the living room", "making coffee" and "working on computer". The possible transitions where limited according to the constraints of the patient's house as presented in Figure 7.3.1. The 7 different activities present in the video were annotated, enabling us to evaluate the performances according to the measures: precision, recall, F-score and global accuracy presented in 7.2. The best recognition performances are presented in Table 7.3.1 where configurations are presented in the left column, corresponding confusion matrices are shown in Figure 7.3.2, lines and columns representing previously listed activities.

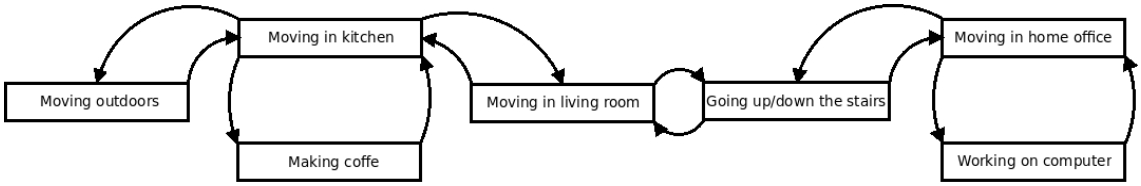
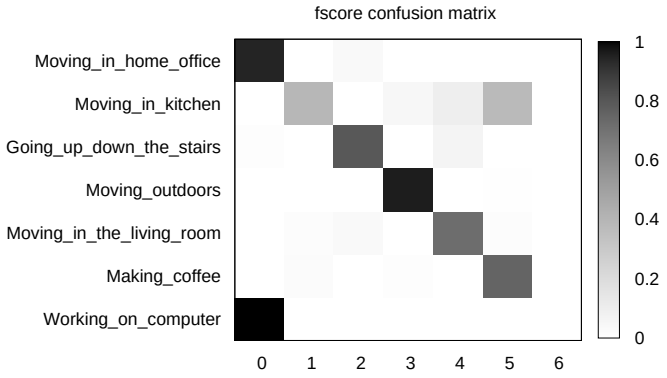
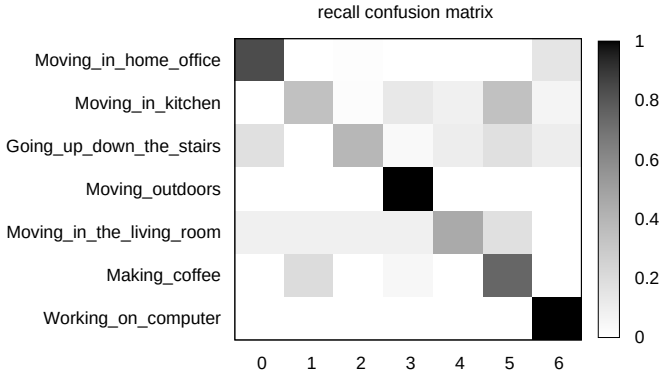


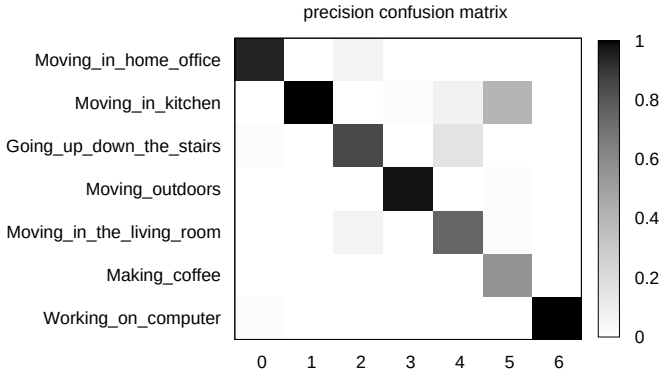
Figure 7.3.1: Constraints on transitions between activities according to the patient's home.



(a) F-score



(b) recall



(c) precision

Figure 7.3.2: Confusion matrices for best recognition results

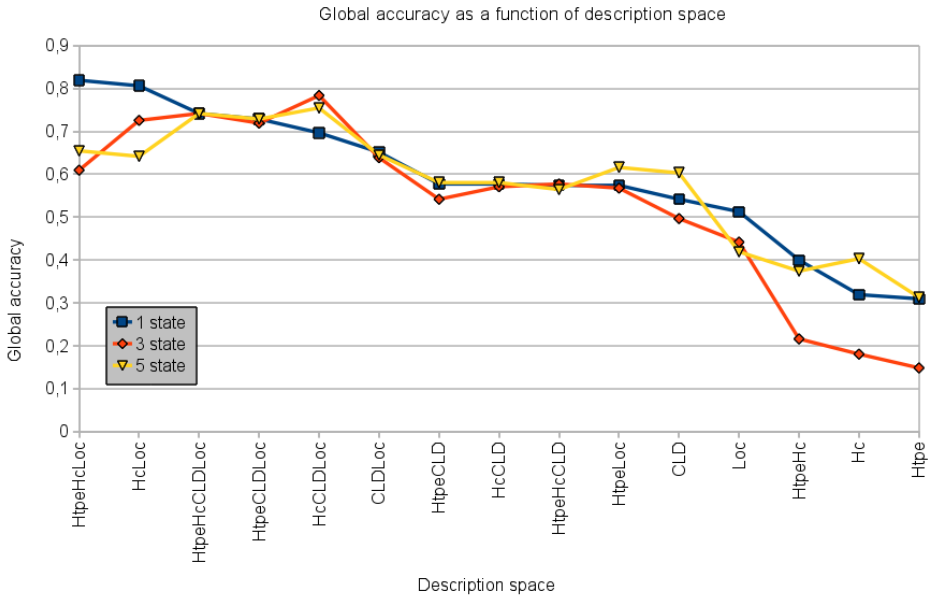


Figure 7.3.3: Global accuracy as a function of description space for the 3 configurations of HMMs studied (sorted by decreasing accuracy for 1 state results).

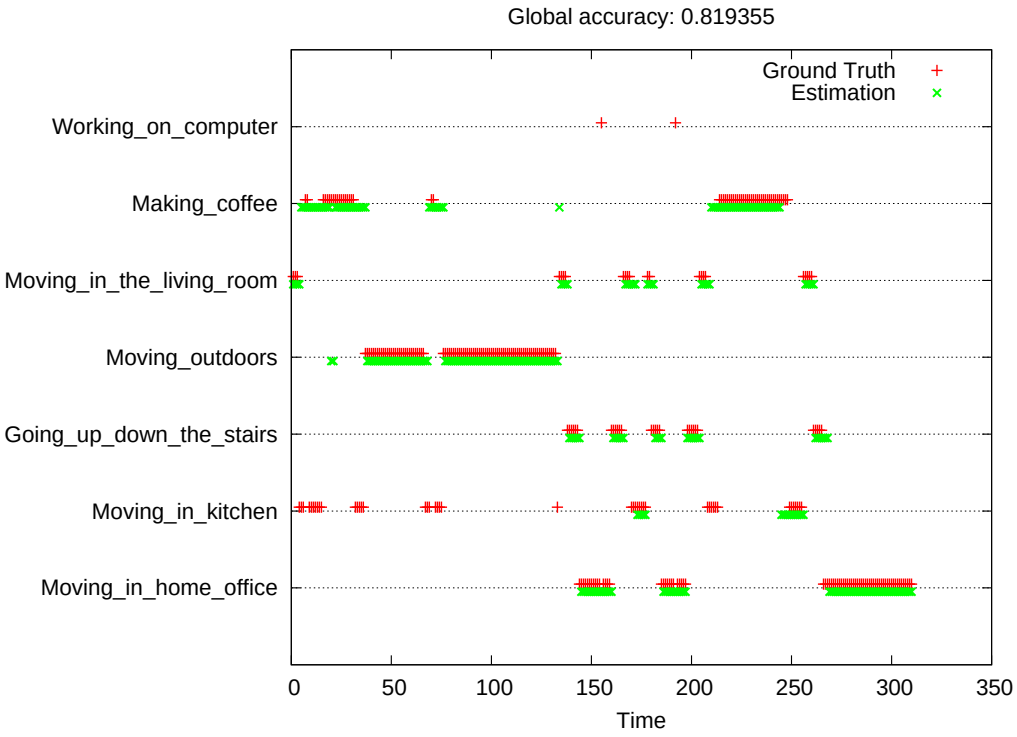


Figure 7.3.4: Chronogram for best global accuracy result

Analysis of results The results presented in the precision confusion matrix, Figure 7.3.2c, are very good. The activities "moving in kitchen" and "working on computer" have a precision of 1. The activities "moving in home office", "going up/down the stairs", "moving outdoors" and "moving in the living room" have high precision and low confusions which only appears with activities which are usually preceding or following the target activity. The main confusion appears between "making coffee" and "moving in the kitchen", since these two activities are located in the same environment and involves similar visual and motion description this is not surprising. The best results in terms of recall, see Figure 7.3.2b, leads to similar conclusions. High recall values are obtained for activities "Moving in home office", "Moving outdoors" and "Working on computer". The main confusions appear once again between "making coffee" and "moving in the kitchen" and also between "moving in the living room" and "going up/down the stairs". The results in terms of F-score show high confusion between "making coffee" and "moving in the kitchen" and also between "working on computer" and "moving in home office".

In Figure 7.3.3, we analyze the performances of the description spaces according to the global accuracy measure for the 3 different HMM states configuration studied. The description spaces are ordered according to decreasing accuracy for the 1 state configuration. The isolated descriptors perform the poorest, while the best performances are obtained with complex fused description spaces. Interestingly, the description space corresponding to a full fusion $H_{tpe}H_cCLDLoc$ shows high performances for all HMM states configurations. This confirms our intuition that the proposed descriptors are relevant for the task and that early fusion is efficient for fusing several modalities. Moreover, the Figure 7.3.3 also shows that similar results are obtained for the 3 HMM states configurations studied. The best results obtained for the single state configuration for the two description spaces $H_{tpe}H_cLoc$ and H_cLoc are highly linked with the fact that the activities of this experiment have low semantic and that the learning and testing process have been done on the same video content. When working with more semantic activities and having the learning and testing observations extracted from different videos, the single state configuration will be strongly limited and therefore excluded from the possible HMMs configurations.

Conclusions and perspectives An overview of these results allows us to conclude that the recognition performances are very good, this can be seen in the chronogram presented in Figure 7.3.3 selected for the best global accuracy configuration. Most of the confusions appear between classes which are usually following each other or are similarly represented in our description space. This reveals that the visual description space is still limited to distinguish activities which take place in the same environment and involve similar motion activity. The activity "working on computer" is also specific because the temporal segmentation gives only 2 segments in the video even if this activity represents thousands of frames. The temporal segmentation has been refined in the following experiments according to this result by specifying a maximal duration of a segment, as explained in 6.3.3. This experiment shows that the approach is valid for the task of activities recognition. The test on a single video was a proof of concept, therefore the next experiment will be run on a mid-volume corpus.

7.3.2 Experiments in a controlled environment

Goal of the experiment The recording at patients' home involves several constraints such as possible medical problems, patients and medical partners availability which induce that it is difficult to obtain a large amount of data for evaluation. At the time of this experiment, no rich corpus of data from wearable video settings has been publicly released. We can reference the dataset [DHMV09] for a very limited task of behavior in the kitchen, where subjects are cooking different recipes. Therefore, we have conducted a set of recording session in a laboratory environment to enable evaluation of our method on a larger scale corpus. This corpus of 2 hours of videos contains heterogeneous activities, for this experiment we used only a part of it to ensure multiple occurrences of activities for the supervised learning. The dataset used for this experiment comprises 6 videos shot in the same laboratory environment, containing a total of 81435 frames which represent more than 45 minutes. In these videos 6 activities of interest appear: "working on computer", "reading", "making tea", "making coffee", "washing the dishes", "discussing" and we added a reject class called "NR", for "not relevant". It represents all the moments which do not contain any of the activities of interest. The activities of interest are a subset of those present in the survey the doctors were using until now.

Training and test data sets We use a cross validation, the HMMs models of activities were learnt on all but one video and tested on this excluded video. Both learning and testing were performed on segments observations. The segment definition relies on the usual constraints, a segment should contain a minimum of 5 frames and a maximum of 1000 frames. For description of the content recorded with wearable cameras we used the two motion descriptors H_{tpe}, containing 10 coefficients and describing the global instant motion, and H_c, containing 8 coefficients which characterize the history of motion; the visual descriptor CLD which contains $l = 12$ coefficient C=(c_1, ... ,c_l) and the *Audio* descriptor which contains $k = 5$ audio descriptors p=(p_1, ... ,p_k), describing the presence of the events: speech, music, noise, silence or periodic sounds. The RM and *Localization* features were not used in this experiment. We designed the global description space in an "early fusion" manner concatenating all descriptors in an observation vector o in R^n space with $n = 35$ dimensions when all descriptors were used. We excluded isolated descriptors from the possible description space configurations. Therefore the description spaces candidates are composed of either 2, 3 ou 4 descriptors.

Evaluation protocol We considered the number of elementary states m=3, 5 or 7, for each activity HMM. The initial looping probabilities A^1_{ii} were set to 0.8 for 3 states HMM and 0.7 for 5 states HMM and to 0.6 for 7 states HMM. The segmentation threshold was evolving in the range $[0.05; 0.95]$ by steps of 0.05. The 7 different activities ("working on computer", "reading", "making tea", "making coffee", "washing the dishes", "discussing" and reject class "NR") present in the videos were annotated, enabling us to evaluate the performances according to the measures presented in 7.2. We will investigate the influence of the segmentation threshold over the global accuracy. We will also give the chronogram of one of the best cross validation runs in terms of global accuracy as the test video with the closest global accuracy to this average value.

Analysis of results We will first discuss the influence of the segmentation parameters, then we will study the description space and finally analyze the activities recognition results.

Segmentation analysis

The influence of the segmentation threshold is linked to the complexity of the model. For the 3 states HMM configuration, the influence is less significant than we expected but Figure 7.3.5(a) shows that the accuracy starts to decrease for threshold values higher than 0.45. However, Figures 7.3.5(b) for 5 states HMMs and 7.3.5(c) for 7 states HMMs show how a higher threshold induces poorer results. Indeed, the higher the threshold is, the more the probability of having a segment containing different activities increases. For instance, the activity "making coffee" and "washing the dishes" may follow each other in a short time. Moreover, the higher the threshold is the less data are available for the HMMs training, some learning activity sequences may be reduced to less observations than the number of states. Therefore, when no valid data sequence is available the model cannot be learnt and the recognition cannot be run. This explains the fall to zero in some curves when there is not enough data to train the HMM. This is why in the next experiments on real world data, we will apply the training on sequences of frames rather than on segments to ensure enough data for the learning process.

Study of the description space The description space is defined as one of the possible combinations of the descriptors, excluding the configurations with only one descriptor. Figure 7.3.5 presents the average accuracy for different combinations of descriptors as a function of the segmentation threshold parameter. As many configurations with 5 and 7 states have performances falling to zero, we will only study the performances in the 3 states configuration. Considering first description spaces being the fusion of only two descriptors, we can see that the poorest performance is obtained using only the motion descriptors H_cH_{tpe}. Combining Audio and Motion gives better results (H_cAudio and $H_{tpe}Audio$) which indicates the positive contribution of the audio descriptor. But the best results are obtained when fusing with the CLD: H_cCLD and $H_{tpe}CLD$. Considering now all possible fusions, we can see that all the six best performances for a threshold lower than 0.2 contains the CLD descriptor. The CLD descriptor seems to improve the results for low segmentation thresholds. This is rather normal since the larger the segment is, the less meaningful the CLD of the key frame will be, regarding the content of the segment. The full description space $H_cH_{tpe}CLDAudio$ performs really well, especially for the 0.15 threshold. Being more complex this description space also needs more training data, therefore with higher thresholds the performance falls.

Activities recognition In order to evaluate the ADL recognition we have chosen one of the best recognition results presented in Figure 7.3.6. The "read" and "discuss" activities are present and detected for this video. Most detections for these activities are correct or appear near an annotated event. The main confusions are observed for the activities "make coffee" and "make tea". These activities are similar in terms of environment as well as motion and audio characteristics. Moreover, since these activities are semantically close, this is not a big issue for the final task where doctors will watch

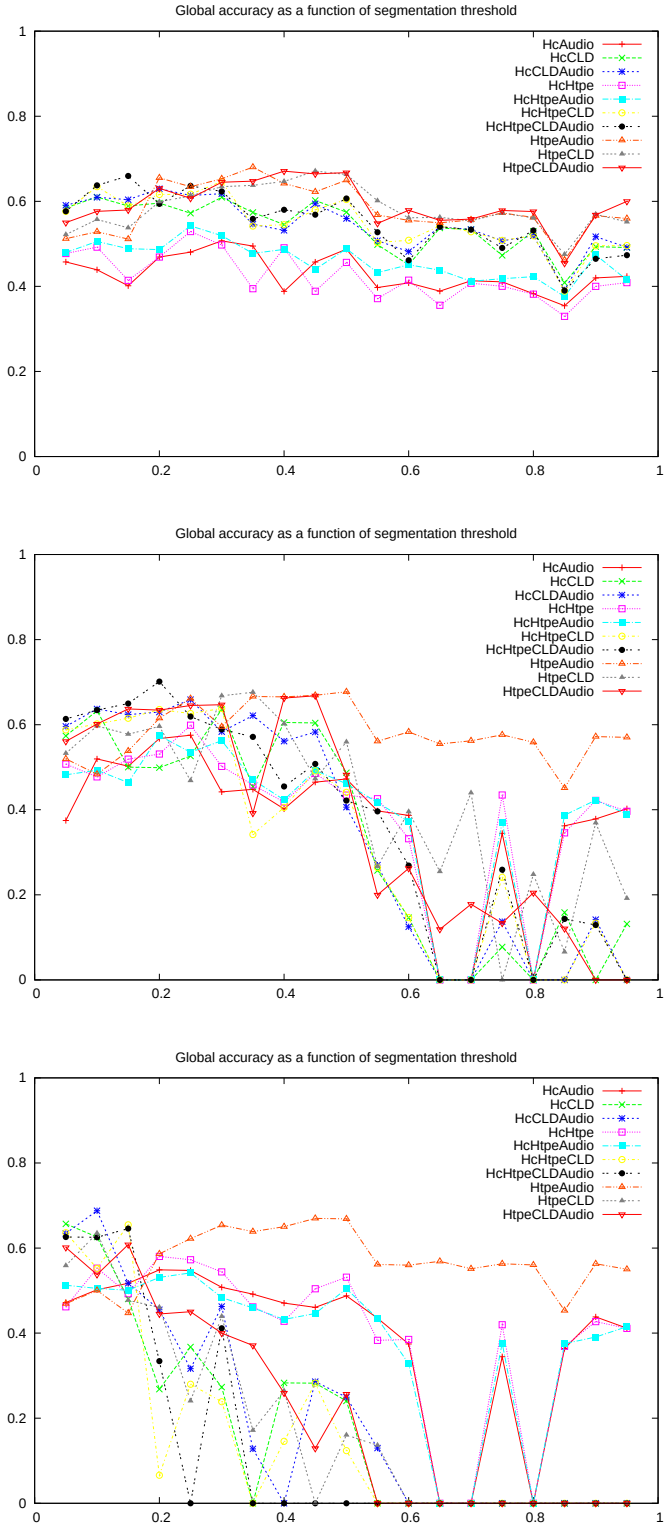


Figure 7.3.5: Description space choice and segmentation threshold influence over accuracy.

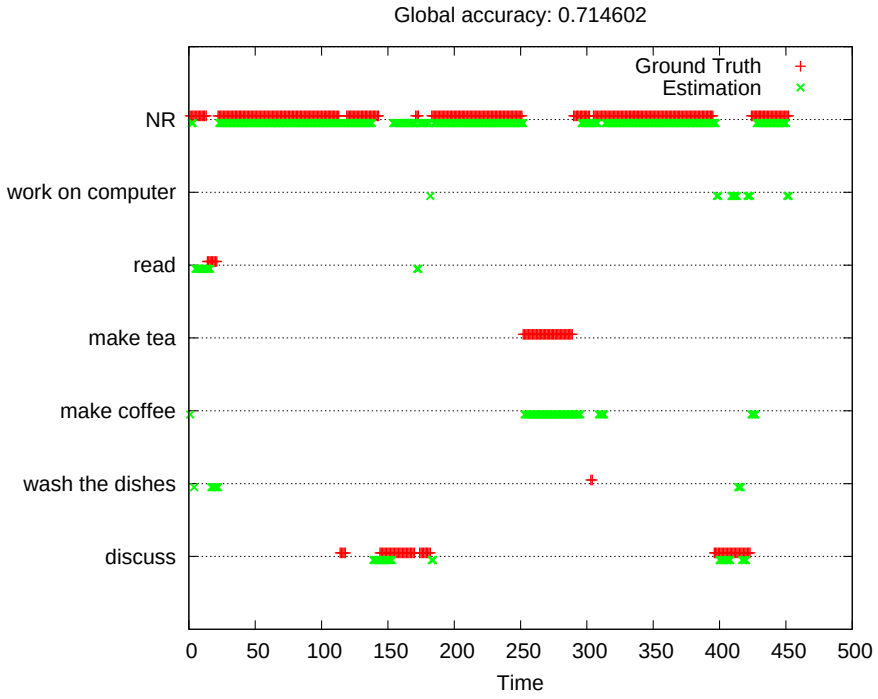


Figure 7.3.6: Chronogram for description space $H_c H_{tpe} CLDAudio$ with a segmentation threshold of 0.15.

the videos using the index. The activity "wash the dishes" is missed but concerns only one segment.

Conclusions and perspectives In this experiment, we have evaluated our approach in a mid-volume corpus. The results are rather good but also show some confusion between activities where the global descriptors can be close for different activities. The next experiment will therefore be the first to include the local motion descriptor RM. The videos used in this experiment were recorded in a controlled environment e.g. a laboratory; the next experiment will be run on a similar corpus in terms of volume but with real world data.

7.3.3 Real word application

Goal of the experiment The aim of this experiment was to run a first evaluation of the proposed method in a real world application. The experiments were conducted on 5 videos recorded with 5 different persons in 5 different environments, i.e. their own homes. Each video is of an average duration of 40 minutes and contains approximately 10 activities; not all activities are present in each video. Each video represents an amount of 50 to 70 thousands frames, which induces hundreds to a thousand segments according to our motion-based temporal segmentation presented in section 6.3. The results of this experiment will be used to select a subset of configurations for a larger scale experiment.

Training and test data sets The HMMs were learned using 4 videos and the last video was used for testing. This was done in a cross validation approach and the results are presented in terms of global accuracy of activities averaged over the cross validation process. The learning was done over a sub sampling of smoothed data extracted on frames. The smoothing substitute the value of each frame descriptor by the average value on the 10 surrounding frames, then one out of ten samples is selected to build ten times more learning sequences. The testing has been done on frames or segments of the last video. The segment definition relies on the usual constraints, a segment should contain a minimum of 5 frames and a maximum of 1000 frames. All descriptors (H_c, H_{tpe}, RM, $Audio$, CLD, Loc) are used in this experiment. The description spaces are built using each descriptor separately and with all possible combinations of descriptors where order is not considered. Therefore, a total of 63 different description spaces are taken into account. We design the global description space in an "early fusion" manner concatenating all descriptors in an observation vector.

Evaluation protocol In the experiments presented here, the bottom level HMM of each activity contains 3 or 5 states. For one evaluation all activities have the same number of states, except the "None" which may be modeled with more or fewer states, here 9 or only one. The initial looping probabilities A^1_{ii} were set to 0.8 for both 3 and 5 states HMM. All HMMs observation models are 5 Gaussians mixtures except the "None" one state-HMM which has only one Gaussian. The segmentation threshold is fixed to 0.2. The set of activities contains 17 different activities: "Cooking", "Moving around", "Hoovering", "Sweeping", "Clear the table", "Make the bed", "Cleaning shovel", "Washing dishes", "Cleaning garbage", "Body hygiene", "Aesthetic hygiene", "Gardening", "Reading", "Watching TV", "Using computer", "Using coffee machine", "Using cooker", "Using microwave", "Medicine", "Use the phone", "Home visit" and reject class "None". They were annotated in the videos, enabling us to evaluate the performances according to the global accuracy measure, which is a ratio between the number of correct estimations and the total number of observations, see section 7.2. We will analyze the performances of recognition using either frames or segments. We will distinguish the best set of description space, this selection will be used in the large scale experiment.

Analysis of results

Evaluation of the influence of temporal segmentation The proposed temporal segmentation reveals three main advantages. First, the amount of data to process in the recognition process is reduced by a factor between 50 and 80 since one observation is defined for a segment and not for a frame. Second, the key frames may be used as a summary of the whole video which is relevant as it gathers the evolution of the patient in successive places. Finally, the evaluation of recognition performance presented in Figure 7.3.7 shows that the results are better when the recognition process is run on segments. The description spaces are ordered by decreasing performances on segments recognition. The best results are always obtained with segments as observations and other results are similar using frames or segments.

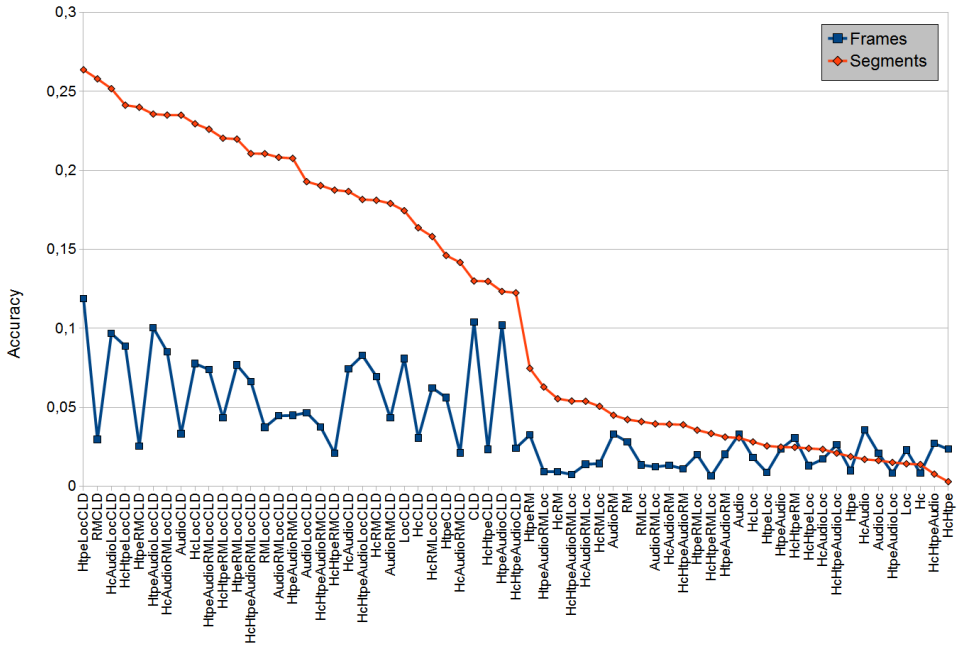


Figure 7.3.7: Global accuracy evaluation of recognition using frames (blue curve and square points) and segments (red curve and diamond shaped points) over all the description spaces fusion tested (sorted by decreasing accuracy with respect to segments approach).
NB: For a better readability of the figure, results are shown for a selected configuration (3statesNone1State) of the HMM but are similar for other configurations.

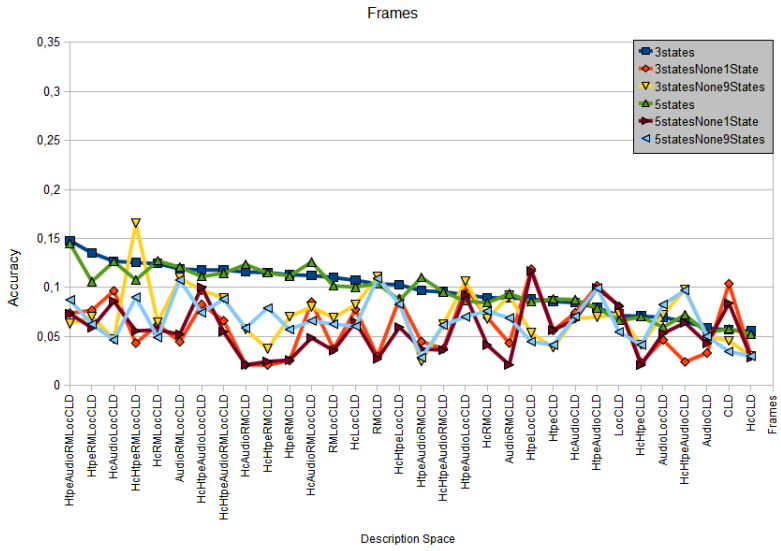
Description space evaluation Figure 7.3.7 also shows which configurations are the most successful for the task. All the 33 best configurations are actually all the configurations including the CLD descriptor. We will therefore in the following only consider configurations which include CLD, and evaluate all possible combination of it with the other descriptors. The results are presented in Figure 7.3.8, where Figure 7.3.8a represents the recognition performance in terms of accuracy when considering frames observations while performances in Figure 7.3.8b are evaluated over segments observations. The descriptions spaces ordered by decreasing performances according to the 3 states configuration which is the one obtaining the best average results. Once again, a significant gain in performance can be observed when using segments instead of frames observations. Here, the best performance is obtained for the fusion $AudioCLD$ with configurations where the reject class is modeled with a single state, the reject class modeling will be studied in the next paragraph. As a general trend, the 3 states configuration gives the best results even if for some description spaces, other configurations may obtain slightly better peformances. The best performances for the 3 states configuration are obtained for description spaces $H_{tpe}LocCLD$, $RMCLD$, $H_cAudioLocCLD$ and $H_cH_{tpe}LocCLD$, showing that all descriptors are relevant.

Reject class model evaluation We have also investigated the influence of modeling the reject class "None" in a different way than all the IADL HMMs. We have performed experiments when modeling this "None" class by a single state HMM or by a much more complex 9 states HMM. From the same Figure 7.3.8, we can see as a general trend that performances with the reject class being modeled as a single state are clearly poorer and using 9 states seems to have less influence on the performances. However, this configuration with 9 states for the "None" class shows good performances in high dimensionality description spaces built upon video segments.

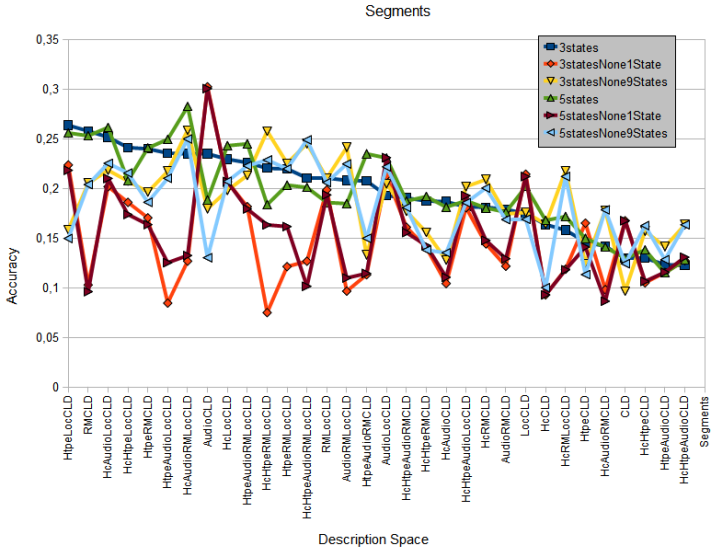
Conclusions and perspectives This first experiment on real world data has shown the difficulty of our task with a significative drop of performances compared to the experiment run in controlled environment. This can be explained as this real world corpus is much more difficult and therefore induces the need for a much larger corpus of learning sequences. This is the aim of the next experiment. As the 3 states configuration have shown the best average performance this configuration is selected for the large scale application.

7.3.4 Large scale application

Goal of the experiment The goal of this experiment was to evaluate the performances on real world data. The previous experiments have shown the complexity of our data. To efficiently learn complex activities as those involved in our videos, a large amount of training data is necessary. The corpus which has grown since the last experiment to 26 videos recorded by 24 people is now more suitable for this complex learning task. From the beginning of these experiments no normalization was applied to the descriptors. Some of them are already normalized (H_{tpe}, $Audio$) but other were used without a normalization preprocessing in the previous experiments. We will therefore investigate the influence of a normalization procedure in these experiments. We have chosen to use the min-max normalization procedure. Min-max normalization subtracts the minimum value of a



(a) Results using frames as observations



(b) Results using segments as observations

Figure 7.3.8: Global accuracy evaluation of recognition using segments over CLD and all possible fusion with CLD description spaces using frames (a) or segments (b) as observations. The curves represent 6 different HMM configuration: 3 states (blue curve and square points), 3 states with “None” class being model with only one state (red curve with circle points), 3 states with “None” class being model with 9 states (yellow curve with triangle pointing down points), 5 states (green curve and triangle pointing up points), 5 states with “None” class being model with only one state (purple curve with triangle pointing right points), 5 states with “None” class being model with 9 states (pale blue curve with triangle pointing left points). The curves are sorted by decreasing accuracy for the 3 states results.

Table 7.3.2: Number of training sequences.

	Number of training sequences		
	Min	Average	Max
All activities	10	229	3112
All activities, except "none"	10	114	401

dimension from each value of this dimension and then divides the difference by the range of the dimension i.e. the difference between the maximum value and the minimum value. These new values evolve therefore in the range $[0, 1]$.

Training and test data sets Here we also use a cross validation, the HMMs models of activities were learnt on all but one video and tested on this excluded video. This was done in a cross validation approach and the results are presented in terms of global accuracy of activities averaged over the cross validation process. The learning was done over a sub sampling of smoothed data extracted on frames. The smoothing substitutes the value of each frame descriptor by the average value on the 10 surrounding frames, then one of ten samples is selected to build ten times more learning sequences. The testing has been done on frames or segments of the last video. The segment definition relies on the usual constraints, a segment should contain a minimum of 5 frames and a maximum of 1000 frames. The description spaces are the 12 best configurations from previous experiments: $H_{tpe}AudioLocCLD$, $H_{tpe}AudioRMLocCLD$, $H_cLocCLD$, $H_{tpe}AudioRMCLD$, $H_cH_{tpe}RMLocCLD$, $H_cH_{tpe}LocCLD$, $H_{tpe}RMCLD$, $AudioCLD$, $H_cAudioLocCLD$, $H_cAudioRMLocCLD$, $H_{tpe}LocCLD$ and $RMCLD$.

We give an overview of the number of training sequences for this experiment in Table 7.3.2. We can see that even in this "large scale" experiment, some activties have still a few training sequences. The activity with only 10 training sequences is "Relationship home visit". However, the average number of training sequences shows that for most activities there is a sufficient number of them. When considering all activities, the reject activity "none" is learned with the largest number of training sequences. When excluding the reject activity, the maximum number of training sequences is for the activity "Cleaning dishes by hand". The time for learning all activities HMMs for one description space configuration, for one test video i.e. on the 25 other videos is about one hour, see details in Table 7.3.3. The testing is instantaneous, the computation time is only one second.

Evaluation protocol We considered the number of elementary states fixed to $m = 3$, for each activity HMM. The initial looping probabilities A^1_{ii} were set to 0.8. The segmentation threshold is fixed to 0.2. The different activities "Food manual preparation", "Displacement free", "Cleaning hoover", "Cleaning broom", "Cleaning clear", "Cleaning bed", "Cleaning shovel", "Cleaning dustbin", "Cleaning dishes by hand", "Hygiene body", "Hygiene beauty", "Hygiene clothes", "Leisure gardening", "Leisure reading", "Leisure watch", "Leisure computer", "Complex machines coffee maker", "Complex machines gas cooker", "Complex machines washing machine", "Complex machines microwave", "Medicines medicines", "Relationship phone", "Relationship home visit" present in the videos were annotated. We will present the configurations giving the best results in

Table 7.3.3: Training and testing computation time for the 12 selected description space, each learning is run on 25 videos.

	Computation time (s)			
	Learning			Testing
Descriptors	Min	Average	Max	Average
HtpeAudioLocCLD	516	600	641	1
HtpeAudioRMLocCLD	1074	1219	1303	1
HcLocCLD	1548	1772	1912	1
HtpeAudioRMCLD	2083	2366	2558	1
HcHtpeRMLocCLD	2642	2987	3232	1
HcHtpeLocCLD	3158	3584	3881	1
HtpeRMCLD	3668	4148	4485	1
AudioCLD	4127	4696	5065	1
HcAudioLocCLD	4634	5281	5684	1
HcAudioRMLocCLD	5180	5891	6344	1
HtpeLocCLD	5664	6456	6944	1
RMCLD	6132	6977	7496	1

terms of F-score, recall, precision and global accuracy. We will give the confusion matrices of the corresponding configurations, selected on a run of the cross validation process with a performance close to the average performance. We will give the chronogram of one of the cross validation run with the closest global accuracy to the best average value. We will compare the performances when using, or not, the normalization procedure.

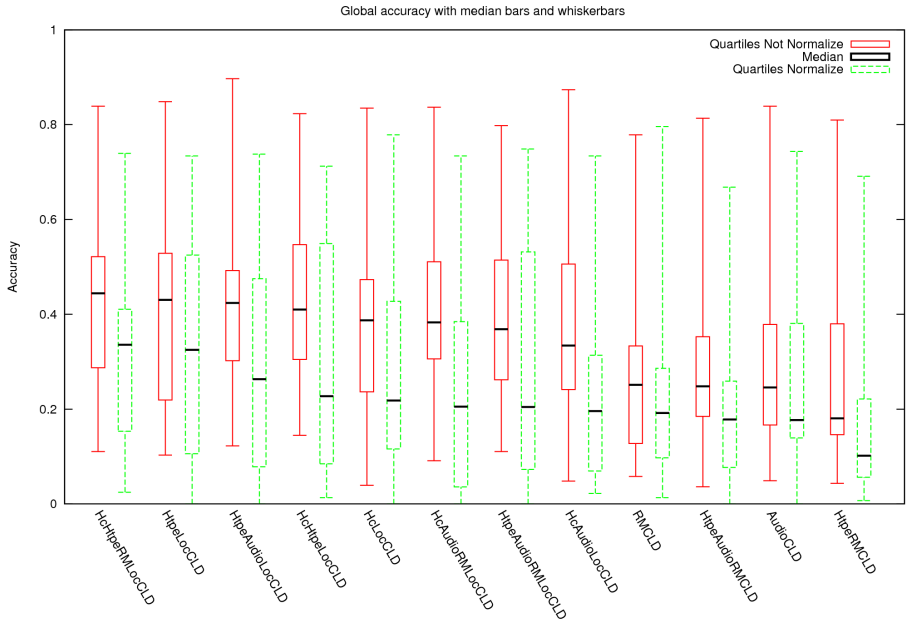
Analysis of results The configurations giving the best recognition results are presented in Table 7.3.4. First of all, we can observe that all the proposed descriptors are relevant for the task as they all appear at least once in the best recognition performances. We can also note that the configuration which obtains the best recognition performance according to one measure has also rather good performances when considering the other measures. The Figure 7.3.9 shows a box plot depicting global accuracy performances of each description space through five values: the minimum performance, lower quartile (Q1), median (Q2), upper quartile (Q3), and maximum performance over all the cross validation runs. This figure shows that the recognition performance evolves in a wide range over the cross validation runs. It also shows that the normalized description spaces gives poorer performances, this can be explained as the normalization process is done for each video separately according to the maximum and minimum values of each dimension in order to adapt the dynamic to each specific patient. This induces a change in the dynamics of each dimension which is not independent of the video.

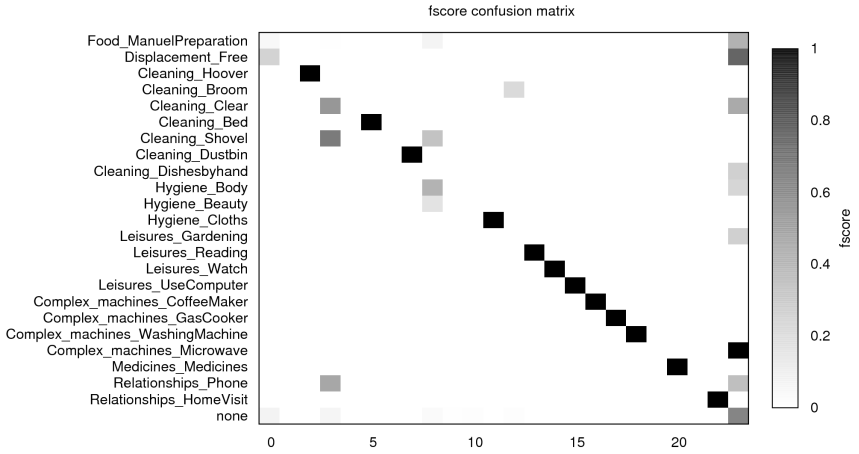
To analyze more precisely the performances at the activity level, confusion matrices are plotted in Figure 7.3.10. In Figure 7.3.10a, which represent the F-score confusion matrix, we can observe that most of the confusion appears between “cleaning” activities. The reject activity “none” is involved in many confusions regarding all measures presented. Similar conclusions can be drawn from the Figure 7.3.10b for precision and 7.3.10c for recall, where the reject activity “none” is involved in most of the confusions while confusions

Table 7.3.4: Configuration for the best recognition results

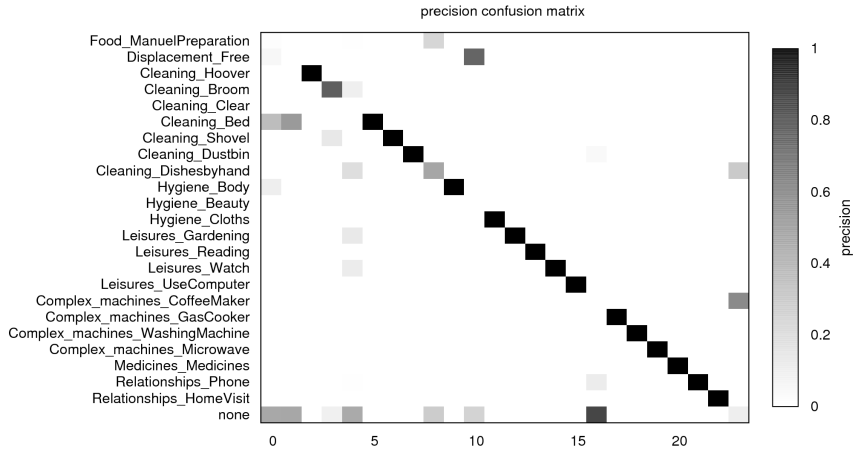
Configuration	Mean F-score	Mean recall	Mean precision	Global accuracy
$H_{tpe}AudioLocCLD$	**0.50**	0.61	0.70	0.39
$H_cAudioLocCLD$	0.48	**0.62**	0.65	0.36
$H_{tpe}LocCLD$	0.50	0.60	**0.70**	0.39
$H_cH_{tpe}RMLocCLD$	0.44	0.61	0.60	**0.41**

Figure 7.3.9: Global accuracy with regards to description spaces for each set of descriptor, performance with normalization is shown on the right and without normalization on the left.

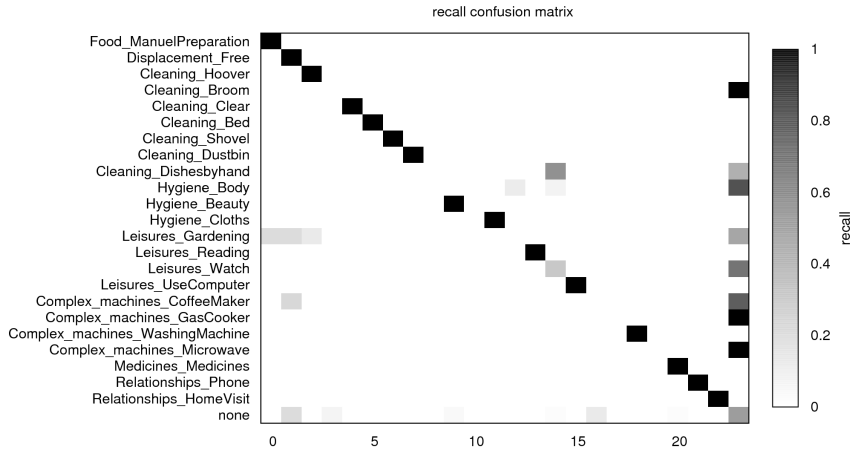




(a) F-score



(b) Precision



(c) Recall

Figure 7.3.10: Confusion matrices for best average recognition results. The confusion matrix presented is the result of one fold of the cross-validation giving the closest value to the average performance.

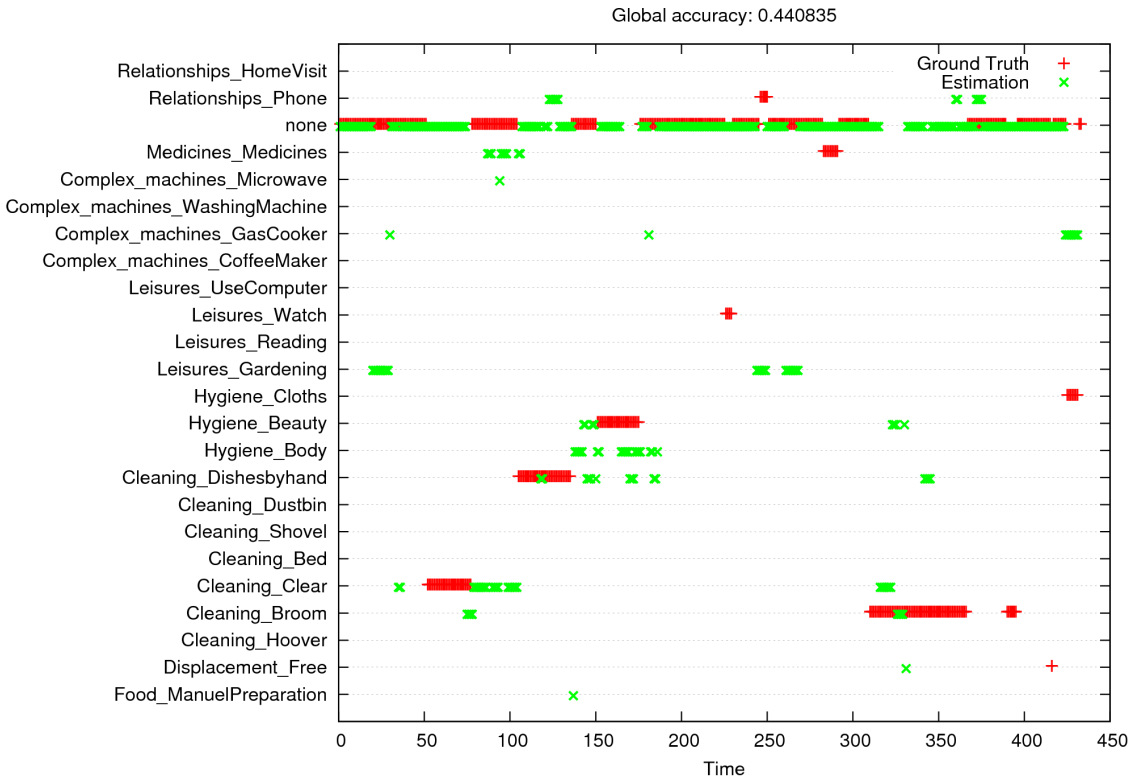


Figure 7.3.11: Chronogram for description space $H_c H_{tpe} RMLocCLD$ selected for a video where the global accuracy is close to the average value

between semantic activities often appears between activities which have similar semantics such as "Cleaning" and "Hygiene" activities.

In Figure 7.3.11, the chronogram of activities is plotted for a video where the global accuracy is close to the average value using description space $H_c H_{tpe} RMLocCLD$. We can here again observe the confusion between activities of the same categories as for example the confusion between "Cleaning clear" and "Cleaning broom" or between "Hygiene beauty" and "Hygiene body".

Conclusions and perspectives This large scale experiment has shown the improvements on performances when a sufficient amount of learning data are available. Since the recording sessions are going on we can expect that with the increasing amount of learning data, the performances increase will continue. The set of descriptors proposed in this manuscript captures most of the information contained in the video stream. However, some activities are still hard to distinguish with this description space such as "Leisure watch" and "Leisure use computer" which may correspond to similar motion, audio and global visual content. Therefore, we will propose an object recognition method in the next chapter as many IADLs are linked to the use of an object. Since we are building a description space with several modalities: motion, audio and visual features, the early fusion may not extract fully the discriminative power of each modalities. More precisely, some

modalities may be more discriminant for some kinds of activities but be mostly seen as noise for others. Therefore, an intermediate or late fusion where modalities are weighted specifically for each activity may also enhance the performances.

Conclusion

Thus, we have conducted experiments for the task of activities recognition with a controlled environment corpus and a real world corpus of same volume i.e. approximately same number of videos and activities. We then conducted a larger scale experiment on real world data with a much larger number of people recording many activities in many different environments. Several conclusions can be drawn from the analysis of these experiments:

- In order to have a sufficient amount of training data, the learning process should be done on frames observations and not on segments.
- In order to avoid an over segmentation during the recognition stage, the observations should be computed on segments.
- The most complete descriptions spaces including most of the descriptors presented in this report give the best performances. These descriptors captures the information in both video domain, by analyzing the motion and the visual content, and audio domain.

The global performances of the method in a controlled environment with a mid-volume corpus are rather good with a best global accuracy score of 0.71. However, they drop significatively in an unconstrained environment with the same volume of data with a best global accuracy score of 0.3. This can be explained by an insufficient amount of training data with regard to the variability of the content of the video scenes. Finally, on a larger scale corpus we observe a strong improvements of performances as the data space is better covered for training with a best global accuracy score of 0.41.

From the computation cost point of view the HMM-based indexing is a non-symmetric tool requiring a strong operational workload for the training stage. In the actual state of the work, the recognition step is also performed off-line and some improvements in the data processing protocol are ongoing. The off-line process is not an obstacle for potential use in clinical practice, as the medical practitioner has to get the final indexing result and not to observe the patient in real-time.

This allows us to conclude that in the context of the IMMED project where data collection will be continued during the next 12 months period, the proposed approach will bring satisfactory results as the training data volume increases. Nevertheless, in view of results we got for the choice of best description space, in the sense of its completeness, we have to further investigate for meaningful features which would be strongly correlated and sufficiently discriminative for activity recognition. Thus, in the following we present our results for extraction of such feature that is object recognition.

Conclusion

In this part we proposed a solution for activities recognition in wearable videos in the framework of Hidden Markov Models (HMM) formalism. After the analysis of the large state-of-the-art of HMM, we proposed a two-level Hierarchical HMM which we considered most adapted for our problem. We have also introduced a pre-segmentation method based on the global motion estimation.

Different description spaces comprising low-level features such as color features of frames, ego-motion, local residual motion and mid-level features results of partial interpretation of the data such as localization in home environment and audio events have been explored. A large scale experimental study on all available corpora allowed for optimal choice of feature combinations but also showed the limits of available description.

Thus, we strongly believe that the incorporation of more semantic feature in the proposed framework could improve the performance. Many activities involve interactions with objects and therefore the recognition of objects can help inferring the activities [PFP+04] [PFKP05]. For this reason, in the following part of this PhD manuscript we will tackle an ambitious task of object recognition in frames which we think will be helpful in the feature.

Part III

Objects Recognition

Introduction

The problem of object recognition in images and videos is one of the hottest in the community. It is a topic of international competitions of leading researchers in the field such as PascalVOC[1] and TRECVID[2]. A large variety of methods have been proposed since the last decades on the bases of local features [Low04] and machine learning approaches. It is clear today that we are stepping into a very large road which still has not led to the right goal. In the variety of methods proposed to achieve the object recognition task, the recent trend which seems promising to us is on the consideration of spatial context in the recognition process [LSP06], [SARK08].

Thus in this part of the manuscript we will propose an approach for object recognition with a new structural and statistical feature keeping good properties of invariance with regard to affine transformation of image plan preserving angles, we call graph words. To efficiently introduce our approach we will first review the state-of-the-art of bag-of-features approaches and related extensions, and further the development of our approach based on this philosophy.

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/
[2] http://trecvid.nist.gov/

Chapter 8

Objects Recognition in Images and Videos : Insight of the State-of-the-Art

Introduction

The SIFT and SURF features previously introduced in chapter 4 have been widely used for representing and matching images in many applications such as automatic stitching, image retrieval, object and location recognition. In the context of image retrieval in databases or object recognition with learning from a large-scale database image to image matching within the whole database is untractable. Therefore an efficient approach was introduced in the last years called the Bag-of-Visual-Words (BoVW) framework [SZ03]. We will present, in next section, the main ideas of this approach. The last sections present some approaches taking the strength of the framework and trying to compensate some of its drawbacks.

8.1 Bag-of-Visual-Words

The state-of-the-art in image or object categorization and recognition has been highly influenced by the paper [SZ03] published by Sivic and Zisserman. In this paper, they have proposed to apply many techniques that have proven to be efficient for text retrieval in the context of object matching within videos. The Bag-of-Words framework will first be presented for the application to text documents. Then, the main steps for its application to images will be reviewed.

8.1.1 Bag-of-Words for text documents

In text retrieval [Lew98], documents are parsed in words. Each word is represented by its stem, for example the stem «walk» stands for the possible variations «walking» or «walks». Then a stop list is used to reject the most common words such as «the» and «an» since they are not discriminant. A unique identifier v_i is associated to each stem.

Each document d is represented by a vector W giving the frequency of occurrence of the words the document contains: $W_d = (t_{v_1}, ..., t_{v_i}, ..., t_{v_k})$.

These values may be weighted, for example by the term frequency-inverse document frequency (tf-idf) weighting [MS83]. Each component of the vector representing the document is the weighted word frequency computed as the product of two terms as in eq. (8.1.1): the *word frequency* $\frac{n_{id}}{n_d}$ and the log *inverse document frequency* $log\,\frac{N}{n_i}$, where n_{id} is the number of occurrences of word v_i in document d, n_d is the total number of words in the document, n_i is the number of occurrences of term v_i in the whole database and N is the number of documents in the whole database. The word frequency term weights words occurring often in a document while the inverse document frequency term down weights words that appear often in the database and are therefore less discriminative.

$$t_{v_i} = \frac{n_{iI}}{n_I}\, log\, \frac{N}{n_i} \tag{8.1.1}$$

Another interesting technique is the use of inverted file which enables fast retrieval. An inverted file has an entry for each word in the corpus followed by a list of documents in which the word occurs. Finally, a text is retrieved by computing its vector of word frequencies and returning the documents with the closest vectors.

8.1.2 Bag-of-Words for images

A general presentation of the framework applied to images is presented in the first part of this section and the approach proposed by Sivic and Zisserman for images is detailed in its remainder. The limit of each module and possible alternatives will be discussed after.

8.1.2.1 Overview

The Bag-of-Visual-Words framework has four main stages: building a visual dictionary, quantifying the features, choosing an image representation using the dictionary and comparing images according to this representation. These steps are explained in the following paragraphs.

Visual dictionary When using images which are only composed of pixel values in a color space, it is necessary to define an equivalent to words in the text context. The images are represented by a set of features describing the content of some regions of interest extracted from the image. Local features such as SIFT and SURF introduced in section 4 are relevant and widely used for image representation. Local features computed over the same object or part of an object contained in different images have many variations due to change of illumination, orientation and so on. They can be seen as hand written words of variations of a stem. According to this analogy, it is necessary to create a set of «visual words» that we can call a «visual dictionary» and denote it by V. Generally, a set of randomly selected features is used to build a visual dictionary by clustering. Similarly to the method in text domain, the most common and rare words can be deleted from the dictionary to enhance the performance.

Feature quantization Feature vectors are often real vectors of high dimension therefore computing distances between many feature vectors is expensive. Moreover, feature vectors computed on the same part of an object or a scene in slightly different illumination or viewing angle conditions will not have exactly the same values. Therefore to enhance the computational performances and have robust representation of an image the feature vectors are quantized according to the visual dictionary V. Usually the quantization step consists in assigning each feature f_i of an image to its closest word v_i in the dictionary V. This process can be referred to as the «coding step».

Image representation According to the visual dictionary of k words, each image I of the data set can now be represented by a k-vector of visual word frequencies W_I. Usually, the vector is normalized by the number of features within the image. Therefore, W_I is a normalized histogram representing the distribution of visual words for the image I. The normalization process enables to compare images which may have different number of features. The visual word can also be weighted by the tf-idf weighting scheme presented in (8.1.1).

Image comparison All images are now represented by histograms, therefore the comparison of two images can be simply done be comparing two histograms. When the number of images in the database grows, this comparison may become computationally expensive. The document vector is very sparse, the comparison of all these sparse vectors is not efficient for retrieval. Since in many applications the task is to retrieve similar images without necessary ranking all the images, another useful tool from the text retrieval domain can be applied. In the classical file structure words are stored in the document they appear. On the contrary, an inverted file structure has an entry for each word where all occurrences of the word in all documents are stored. Therefore, querying the database for similar images can be achieve by simply retrieving the content of the inverted file at entries corresponding to the visual words of the query image. Hence, the ranking process will be applied on a small subset of the database.

8.1.2.2 Sivic and Zisserman proposal

Sivic and Zisserman [SZ03] have used two interest regions detectors: Shape Adapted (SA) [MS02] and Maximally Stable (MS) [MCUP04] regions but both are represented by SIFT features. In this paper, the vector quantization is carried out by K-means clustering. The number of clusters is chosen empirically to maximize retrieval results on the ground truth sets, about 6k clusters are used for Shape Adapted regions, and about 10k clusters for Maximally Stable regions. At the retrieval stage images are ranked by their normalized scalar product (cosine of angle) between the query vector W_q and all images vectors W_I in the database.

In their experiments Sivic and Zisserman have first tested this approach on scene matching. The data set is composed of 164 frames from 48 shots taken from 19 different locations from the movie Run Lola Run. Each frame is used in turn as a query region. The retrieval performance are evaluated using the average normalized rank of relevant images [MMMP02]. The performance shows the efficiency of the method as the retrieval ranking is perfect for 17 of the 19 locations.

The second experiment is on object retrieval, the query object is defined by the user as a sub-part of any frame. A key frame is extracted every second of the movie. Descriptors are computed and quantized using the dictionary for each frame. For this object retrieval task, frames are first ranked according to the weighted frequency vector alone, and then re-ranked according to a spatial consistency measure. A search area is defined by the 15 nearest neighbors of each match, and each region which also matches within this area casts a vote for that frame. Matches with no support are rejected and the total number of votes determines the rank of the frame. The results are once again really good as most of the highly ranked frames contains the query object.

Some ideas presented by Sivic et al. are not always reproduced in more recent works. For example they have used the classical weighting scheme from text retrieval which is the «term frequency-inverse document frequency» presented earlier. The adaptation to images is straightforward, each component of the vector $W_I = (t_{v_1}, ..., t_{v_i}, ..., t_{v_k})$ representing image I corresponds to the weighted word frequency computed as the product of two terms as in (8.1.1): the *word frequency* $\frac{n_{iI}}{n_I}$ and the *inverse document frequency* $log\,\frac{N}{n_i}$, where n_{iI} is the number of occurrences of visual word v_i in image I, n_I is the total number of visual words in image I, n_i is the number of occurrences of visual word v_i in the whole database and N is the number of images in the whole database.

Another way of increasing the discriminative power of the words produced by the clustering is to use a stop list. The idea of the stop list is to remove from the vocabulary words which are very frequent and those who are very rare. The stop list used by the Sivic and Zisserman was determined empirically. They considered the top 5% and the bottom 10% as stop words.

8.2 Bag-of-Visual-Words limitations and improvements

The BoVW framework was clearly a breakthrough in the domain of image recognition or retrieval. However, this framework had some limitations that have been discussed and challenged since the paper of Sivic and Zisserman [SZ03]. We will review in this section the different limitations and recently proposed improvements of each step of the procedure.

8.2.1 Dictionary building process

In the initial BoVW framework proposed by Sivic and Zisserman [SZ03], the visual dictionary was built by a k-means clustering. This method has been widely used since [CDF+04, LM01, WCM05]. The name «k-means» was first used in [Mac67], we will briefly review the most common algorithm also referred to as Lloyd's algorithm [Llo57].

K-means algorithm Given a set of n observations $X = \{x_1, x_2, \ldots, x_n\}$, the aim of the k-means algorithm is to define k ($k \leq n$) set of observations $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares:

$$\underset{S}{argmin} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \tag{8.2.1}$$

where μ_i is the mean of points in set S_i. The initial k means $(\mu_1^{(1)}, \ldots, \mu_k^{(1)})$ are usually selected randomly in the data set and then the algorithm proceeds by alternating between two steps:

- The assignment step: each observation in the data set is assigned to the closest mean.

$$S_i^{(t)} = \left\{ x_j \,|\, \left\| x_j - \mu_i^{(t)} \right\| \leq \left\| x_j - \mu_l^{(t)} \right\| \forall l \in [1, k] \right\} \tag{8.2.2}$$

- The update step: calculate the new mean to be the centroid of each observations cluster.

$$\mu_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j \tag{8.2.3}$$

The k-means iterative process stops when the assignment no longer changes. The k-means algorithm has no guarantee to converge to the global optimum and the result may depend on initially selected means. Moreover, the local optimal solution found may be arbitrarily bad compared to the global optimal solution.

Therefore an improvement of the algorithm, known as *k-means++,* has been proposed in [AV07]. The intuition of the k-means++ algorithm is to spread initial cluster centers away form each other. The first cluster is chosen uniformly at random from the data points, after which each subsequent cluster center is chosen from the remaining data points with probability proportional to its distance squared to the point's closest cluster center. Once the initialization is done, the standard k-means algorithm is applied. The k-means++ algorithm guarantees an approximation ratio $O(log\, k)$.

Vocabulary tree One very interesting use of the k-means algorithm in the context of the BoVW framework was proposed in [NS06], where the algorithm is used to build a hierarchic structure that the authors called the *vocabulary tree.* The vocabulary tree defines a hierarchical quantization that is built by hierarchical k-means clustering. A large set of representative descriptor vectors are used in the unsupervised training of the tree. Instead of k defining the final number of clusters or quantization cells, k defines the branch factor (number of children of each node) of the tree. First, an initial k-means process is run on the training data, defining k cluster centers. The training data is then partitioned into k groups, where each group corresponds to the set of the descriptor vectors closest to a particular cluster center. The same process recursively defines quantization cells by splitting each quantization cell into k new parts. The tree is determined level by level, up to some maximum number of levels L, and each division into k parts is only defined by the distribution of the descriptor vectors that belong to the parent quantization cell. The process is illustrated in Figure 8.2.1.

The tree directly defines the visual vocabulary and an efficient search procedure in an integrated manner. This is different from defining a visual vocabulary non-hierarchically, and then devising an approximate nearest neighbor search in order to find visual words efficiently. The vocabulary tree gives both higher retrieval quality and efficiency compared to the initial BoVW framework of [SZ03].

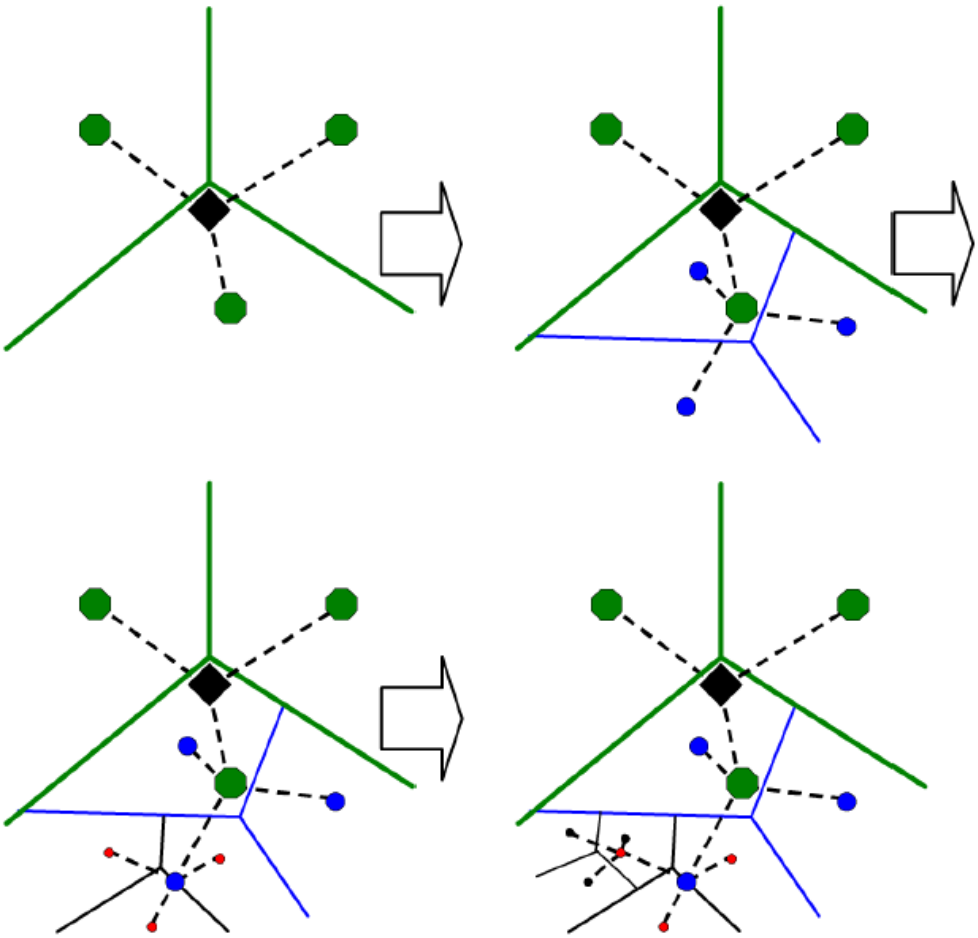


Figure 8.2.1: An illustration of the process of building the vocabulary tree. The hierarchical quantization is defined at each level by k centers (in this case $k = 3$) and their Voronoi regions. Image from [NS06].

8.2.2 Visual words quantization

Given an image I, let F be a set of features computed at N locations identified with their indices $i = 1, ..., N$. In previous sections, according to the analogy with text, we have considered that, during the «coding step», a feature f_i was associated with a single visual word v_i. However, in the visual domain, the visual words are much more ambiguous than in the text domain. Therefore, a feature f_i may be represented by a codeword being either a single scalar value i.e. the identifier of the visual word or by a vector representing weights of several words which are similar to the feature. Let α_i be the codeword assigned to f_i by the coding operator q. For the sake of generalization, we will in the following consider it as a vector $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,k})^T \in \mathbb{R}^K$ that relates feature f_i to each visual word v_k of the dictionary by an affinity $\alpha_{i,k}$. The quantization can be formalized as in 8.2.4.

$$\alpha_i = q\,(f_i),\ i = 1,\ ...,\ N \tag{8.2.4}$$

The coding step or quantization is the process of transforming the input feature f_i into a representation α_i that has some desirable properties such as compactness, sparseness (i.e. most components are 0) or statistical independence. An efficient coding step should be able to tackle the codewords ambiguity which encompass several different situations presented in Figure 8.2.2, where the data sample represented by a square is close to two codewords and the one depicted by a diamond is far from the closest codeword. Different coding operators have been proposed in the literature but often in a whole image retrieval or recognition framework without much discussion of the influence of this choice. In [BBLP10], Boureau et al. have studied the influence of three of the most widely used coding operators: hard quantization, soft quantization and sparse coding. In the following paragraphs we will review the principles of these process and their performances.

Hard quantization Hard quantization is the classical formulation of the bag-of-words framework [SZ03]. The coding operator q minimizes the distance to a code book, i.e. each feature is assigned to the closest codeword in the dictionary, which is usually build by an unsupervised algorithm such as K-means. Let v_k denote the k-th codeword. This process can be formalized as:

$$\alpha_{i,j} = \begin{cases} 1 & \text{if } j = \underset{j}{argmin} \parallel f_i - v_j \parallel_2^2 \\ 0 & \text{otherwise} \end{cases} \tag{8.2.5}$$

Soft quantization From the previous definition it is clear that there is a strong quantization by assigning a continuous feature to a single representative. This drawback has been studied by Gemert et al. in [GVSG10]. They explore soft quantization techniques and evaluate the influence on classification performances when using low to high dimensional features or small to very large vocabulary. The idea of soft quantization is to tackle the ambiguity of a visual word that hard quantization simply ignores. The drawbacks of

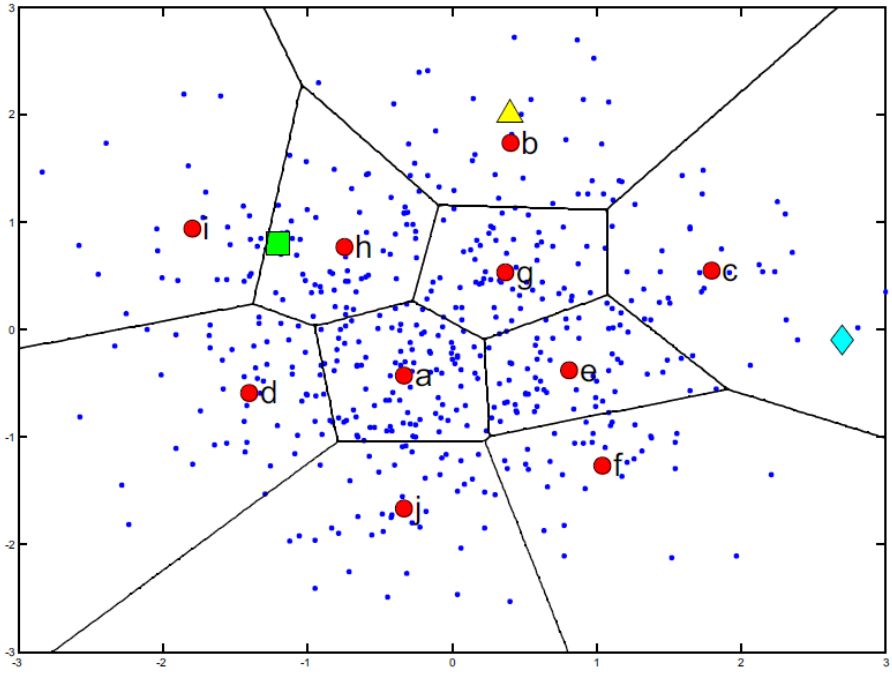


Figure 8.2.2: An example illustrating visual word ambiguity in the code book model. The small dots represent image feature vectors. The labeled red circles are visual words found by unsupervised clustering. The triangle represents a data sample that is well suited to the code book model. Visual word uncertainty is exemplified by the square, whereas visual word plausibility is illustrated by the diamond. Figure from [GVSG10].

hard quantization are twofold: (i) when a data sample is close to several codewords, only the closest is considered and (ii) a codeword is assigned to the closest codeword no matter how far it can be. The first aspect is referred to as *word uncertainty* and the second as *word plausibility*. Instead of using histograms to estimate the probability density function the authors proposed to use a kernel density estimation [BSI08], [SG86]. Defining a Gaussian-shaped kernel $K_\sigma(x)$ in (8.2.6), three models are studied in this paper. The *kernel codebook KCB* defined in (8.2.7) will weight each word by the average kernel density estimation for each data sample. The *codeword uncertainty UNC* defined in (8.2.8) normalizes the amount of probability mass to a total of 1 which is distributed over all relevant codewords. Finally, the *codeword plausibility PLA* defined in (8.2.9) will give a higher weight to more relevant data samples but is not able to select multiple codeword candidates. The weight distributions obtained by these three kernel based approaches and the standard hard quantization techniques on the example of data samples presented in Figure 8.2.2 are depicted in Figure 8.2.3. The results on a classification task obtained on the data sets Scene-15, Caltech-101, Caltech-256 and Pascal VOC 2007/2008 shows that the *codeword uncertainty UNC* outperforms all other methods using either low or high dimensional feature vectors or small or very large visual vocabulary.

$$K_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}\, exp(-\frac{1}{2}\frac{x^2}{\sigma^2}) \tag{8.2.6}$$

$$\alpha_{i,j}^{KCB} = K_\sigma(\, D\,(v_j,\, f_i)) \tag{8.2.7}$$

$$\alpha_{i,j}^{UNC} = \frac{K_\sigma(\, D\,(v_j,\, f_i))}{\sum_{k=1}^{|V|} K_\sigma(\, D\,(v_k,\, f_i))} \tag{8.2.8}$$

$$\alpha_{i,j}^{PLA} = \begin{cases} K_\sigma(\, D\,(v_j,\, f_i)) & \text{if } v_j = \underset{v_j \in V}{argmin}(\, D(v_j,\, f_i)) \\ 0 & \text{otherwise,} \end{cases} \tag{8.2.9}$$

Sparse Coding Sparse coding [OF97] uses a linear combination of a small number of codewords to approximate the feature f_i. These codewords are represented by a dictionary $V = (v_1, \ldots, v_k)$ in matrix form $V \in \mathbb{R}_{d\times K}$ where d is the dimension of the feature space. The linear weights correspond to the vector $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,k})^T \in \mathbb{R}^K$. Yang et al. [YYGH09] have obtained state-of-the-art results by using sparse coding and max pooling. We will here only briefly define the sparse coding process as an overview of sparse coding is beyond the scope of this manuscript.

$$\begin{aligned} \hat{\alpha_i} \;= \underset{\alpha_i}{argmin} L(\alpha_i, V) \qquad &\triangleq \| f_i - V\alpha_i \|_2^2 + \lambda \|\alpha_i\|_1 \\ &= \left\| f_i - \sum_k \alpha_{i,k} v_k \right\|_2^2 + \lambda \|\alpha_i\|_1 \end{aligned} \tag{8.2.10}$$

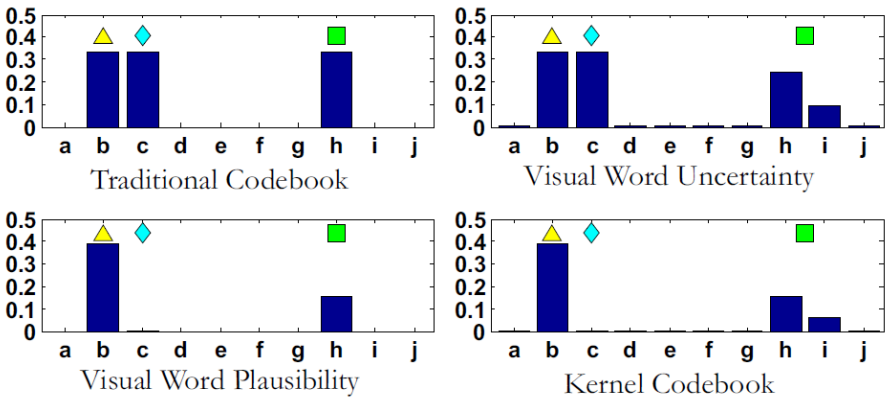


Figure 8.2.3: Summary of different types of codeword ambiguity. Figure from [GVSG10]

where $\|\alpha\|_1$ denotes the L_1 norm of α, λ is a parameter that controls the sparsity, and V is a dictionary trained by minimizing the average of $L(\alpha_i, V)$ over all samples, alternatively over V and the α_i. Yang et al. and Boureau et al. [BBLP10] have shown that sparse coding outperforms both hard quantization and soft quantization in several data sets such as Scene-15, Caltech-101 and Catlech-256.

The initial hard quantization process has clear limitations. The recent improvements which consist of distributing each feature over a small number of representative codewords clearly enhance the quantization step of the Bag-of-Words framework.

8.2.3 Visual words distribution comparison

Histograms comparison The BoVW approach represents the distribution of visual words in an image by an histogram. The comparison of two images therefore relies only on the comparison of their representative histograms. Let W_A and W_B be the visual words distribution histograms of two images A and B respectively, both being normalized and of same dimensionality K. Let us denote $W^{(i)}$ the value of the i-th bin of the histogram. Many metrics can be defined to compare histograms, we will review those which have often been used in BoVW frameworks.

- The L2 metric:

$$d_{L_2}(W_A, W_B) = \sum_{i=1}^{K} (W_A^{(i)} - W_B^{(i)})^2 \tag{8.2.11}$$

- The Chi-Square metric:

$$d_{\chi^2}(W_A, W_B) = \sum_{i=1}^{K} \frac{(W_A^{(i)} - W_B^{(i)})^2}{W_A^{(i)} + W_B^{(i)}} \tag{8.2.12}$$

- The L1 metric:

$$d_{L_1}(W_A, W_B) = \sum_{i=1}^{K} \left| W_A^{(i)} - W_B^{(i)} \right| \tag{8.2.13}$$

In practical applications, most of the approaches use the histogram intersection or the L1 distance or equivalently the associated similarity[1], the histogram intersection:

$$\mathcal{I}(W_A, W_B) = \sum_{i=1}^{K} min(W_A^{(i)}, W_B^{(i)}) \tag{8.2.14}$$

8.2.3.1 Feature distribution comparison

Not all approaches recently developed in the context of image retrieval use the quantization according to a single visual dictionary. One interesting approach presented in [GD05] makes use of a multiresolution quantization process.

Pyramid Matching kernel Instead of quantifying features as visual words according to a visual dictionary, the Pyramid Match Kernel (PMK), presented in [GD05], maps unordered features sets to a multi-resolution histograms and then compares the histograms with a weighted histogram intersection measure in order to approximate the similarity of the best partial matching between the two feature sets. This Kernel function is positive-definite, making it appropriate to use with learning methods that guarantee convergence to a unique optimum for positive-definite kernels, for example SVM (Support Vector Machines). The method does not assume a parametric model and can handle sets of unequal cardinality.

An image can be represented by a feature set F of m_F d-dimensional feature vectors in an input space $\mathtt{X}$:

$$\mathtt{X} = \left\{ F \,|\, F = \left\{ \left[f_1^1, ..., f_d^1 \right], ..., \left[f_1^{m_F}, ..., f_d^{m_F} \right] \right\} \right\} \tag{8.2.15}$$

The feature extraction process builds a vector of concatenated histograms $\Psi(F)$, see (8.2.16). The number of levels in the pyramid L is set to $log_2 d$. Each histogram $H_i(F)$ have bins of side length 2^i, each subsequent histogram has bins that double in size (in all d dimensions) compared to the previous one. At the finest-level, each data point fall in its own bin of the histogram H_{-1} while all data points falls potentially into a single bin at the coarsest level L.

$$\Psi(F) = [H_{-1}(F), H_0(F), ..., H_L(F)] \tag{8.2.16}$$

In this multi-resolution histogram space, the pyramid match kernel K_Δ (8.2.17) measures the similarity between points sets $\mathtt{y}$ and $\mathtt{z}$ as a weighted sum of the number of newly matched pairs of features N_i found at each level i of the pyramid formed by Ψ. The weight w_i is proportional to how similar two points can be, as determined by the bin size. Matches made within large bins are weighted less than those found in smaller bins. In practice the choice made by Grauman and Darell [GD05] is to set the weight to $\frac{1}{2^i}$ to reflect the (worst case) similarity of points matched at level i.

[1] $\mathcal{I}(W_A, W_B) = \frac{1}{2} \sum_{i=1}^{K} W_A^{(i)} + W_B^{(i)} - \left| W_A^{(i)} - W_B^{(i)} \right| = 1 - d_{L_1}(W_A, W_B)$ if $\sum_{i=1}^{K} W_A^{(i)} = \sum_{i=1}^{K} W_B^{(i)} = 1$

$$K_\Delta(\Psi(\mathtt{y}), \Psi(\mathtt{z})) = \sum_{i=0}^{L} w_i N_i \tag{8.2.17}$$

To compute the number of newly matched pairs, the kernel makes use of a histogram intersection function $\mathcal{I}$ (8.2.14) which measures the «overlap» between two histograms of r bins at the same level. Then N_i is computed as the difference between successive histogram levels intersections (8.2.18). The kernel never computes distances between the vectors in each set. It simply counts the number of newly matched pairs by comparing the intersection measure at successive scales, see Figure 8.2.4.

$$N_i = \mathcal{I}(H_i(\mathtt{y}),\, H_i(\mathtt{z})) - \mathcal{I}(H_{i-1}(\mathtt{y}),\, H_{i-1}(\mathtt{z})) \tag{8.2.18}$$

The PMK allows computing a relaxed [Ved09] metric between two sets, which is less strict on quantization boundaries than non pyramidal comparison. However, this method does not take into account the spatial relations between the features extracted from the image. The next section will detail some approaches that integrate spatial information for distribution comparisons.

Context Dependent Kernel Let us denote two sets of interest regions $S_A = \left\{r_1^A, \ldots, r_n^A\right\}$ and $S_B = \left\{r_1^B, \ldots, r_m^B\right\}$ extracted from two images A and B respectively, where a region r_i^I of image I is defined by its coordinates $(x_i^I,\, y_i^I)$ and a feature f_i^I: $r_i^I = (x_i^I,\, y_i^I,\, f_i^I)$. The previous approaches have compared the two sets S_A and S_B by either quantifying these features as visual words and comparing their visual distribution W_A and W_B as histograms or by comparing a pyramid of histograms quantization in the feature space. Both of these approaches discard all spatial relations, e.g. proximity, between the interest regions extracted.

In [SARK08] and [SAK10], Sahbi et al. have introduced a kernel which takes into account both feature similarity «alignment quality» and spatial alignment in a «neighborhood» criteria. The «Context-Depenedent Kernel» (CDK) is defined as the fixed-point of an energy function which balances a «fidelity» term, i.e. the alignment quality in terms of features similarity, a «context» criterion, i.e. the neighborhoods spatial coherence of the alignment and an «entropy» term. The proposed alignment model is model-free, i.e. it is not based on any a priori alignment model such as homography and can therefore capture inter-object transformations.

Considering any pair of regions (r_i^I, r_j^J) of two images I and J, let us denote D the matrix of dissimilarity in the feature space: $D_{r_i^I, r_j^J} = d(r_i^I, r_j^J) = \left\| f_i^I - f_j^J \right\|_2$. Let $\mathcal{N}(r_i^I)$ be the set of neighbors of r_i^I. Let us denote P the proximity matrix defined according to the neighborhood criterion:

$$P_{r_i^I, r_j^J} = \begin{cases} 1 & \text{if I} = \text{J and r}_\text{j}^\text{J} \in \mathcal{N}(r_i^I) \\ 0 & \text{otherwise} \end{cases} \tag{8.2.19}$$

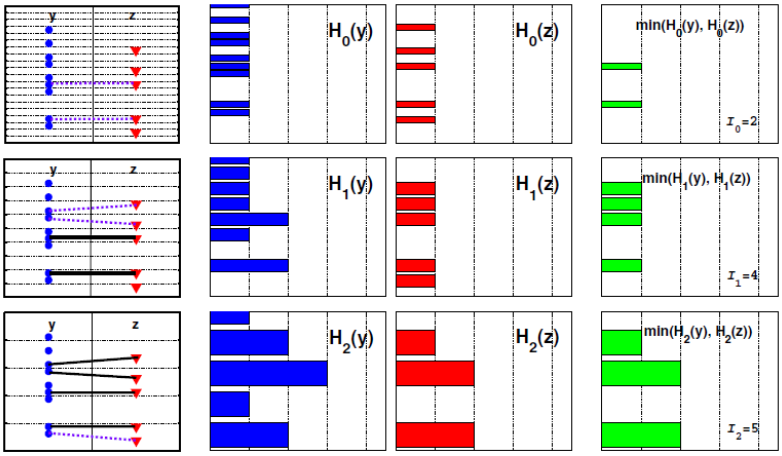


Figure 8.2.4: A pyramid match determines a partial correspondence by matching points once they fall into the same histogram bin. In this example, two 1-D feature sets are used to form two histogram pyramids. Each row corresponds to a pyramid level. In (a), the set `y` is on the left side, and the set `z` is on the right. Light dotted lines are bin boundaries, bold dashed lines indicate a pair matched at this level, and bold solid lines indicate a match already formed at a finer resolution level. In (b) multi-resolution histograms are shown, with bin counts along the horizontal axis. In (c) the intersection pyramid between the histograms in (b) are shown. K_Δ uses this to measure how many new matches occurred at each level. $\mathcal{I}_i$ refers to $\mathcal{I}(H_i(y), H_i(z))$. Here, $\mathcal{I}_i = 2, 4, 5$ across levels, and therefore the number of new matches found at each level are $N_i = 2, 2, 1$. The sum over N_i, weighted by $w_i = 1, 1/2, 1/4$, gives the pyramid match similarity. Image from [GD05]

The Context-Dependent Kernel K is the unique solution of the energy function minimization problem and is the limit of:

$$K^{(t)} = \frac{G(K^{(t-1)})}{\left\|G(K^{(t-1)})\right\|_1}$$

where

$$G(K) = exp(-\frac{D}{\beta} + \frac{\alpha}{\beta} P K^{(t-1)} P)$$

and

$$K^{(0)} = \frac{exp(\frac{-D}{\beta})}{\left\|exp(\frac{-D}{\beta})\right\|_1}$$

Where exp represents the coefficient-wise exponential and $\|M\|_1 = \sum_{ij} |M_{ij}|$ represents the L_1 matrix norm. The two parameters β and α can be seen respectively as weights for features distance and spatial consistency propagation. The CDK convergence is fast, in [SAK10] only one iteration was applied. The CDK was evaluated on the Olivetti face database, the Smithsonian leaf set, the MNIST digit database and ImageClef@ICPR set showing significant improvements of equal error rate (ERR) compared to Context-Free Kernels.

8.2.3.2 Distribution comparison using visual words and spatial information

Yet efficient, the Bag-of-Words approach does not cover one important part of an image or an object: the spatial organization. Representing an image by a global histogram induces lack of spatial information and relations between interest regions. In the past few years, several methods have tried to overcome this limitation of the BoW framework. We will here report two approaches that uses local histograms instead of global histograms: visual phrases and the spatial pyramid matching kernel.

Visual Phrases In [AMC10], Albatal et al. note two important limitations of the BoW framework as visual words are much more ambiguous than text words and that in the global histogram representation, all information related to topological organization of the regions of interest in the image are lost. They have proposed a method to create groups of regions in the image to form areas which are spatially larger than the individual regions and have the same robust visual properties. As shown in [ZZN+08], grouping several regions may describe and distinguish classes of objects better than individual regions. Let a region of interest r_p^I be a quadruple $(x_p, y_p, \rho_p, f_p) \in M$, with: x_p and y_p the coordinates of the region center in the image I, ρ_p is the radius of the region, f_p the visual feature of the region and M the domain of possible values for regions of interest. Let R^Idenote the set of all regions of interest in an image I.

Albatal et al. have proposed to use a «Single Link» clustering function, with a topological proximity criterion based on the Euclidean distance between the regions of interest, see (8.2.20). This criterion defines two regions as close if the Euclidean distance between their centers is less or equal than the sum of their radii. This type of clustering does not depend on the starting point and ensure that the created groups are disjoint i.e. each cluster defines a visual phrase.

$$c(r_p^I,\, r_n^I) \equiv \sqrt{(x_p - x_n)^2 + (y_p - y_n)^2} \leq \rho_p + \rho_n \tag{8.2.20}$$

Each Visual Phrase is then represented as a histogram of N dimensions with N the number of words in a chosen visual dictionary V. Visual Phrases construction process gives the following properties to the visual phrases:

- Invariance to scale changes: the topological proximity criterion does not change with scale. If the scale becomes larger, distances between regions and radii of regions will expand proportionally;
- Invariance to rotations: rotation does not affect the distance between regions nor regions radii;
- Invariance to translations: translation does not change the criterion of proximity;
- Invariance to brightness changes: this property is hold by the regions of interest, Viusal Phrase would simply inherit it.

The approach is evaluated on an automatic annotation task on the VOC2009 collection. First, in the learning step, Visual Phrases are extracted and labeled according to objects boundaries in the training images. A supervised discriminative algorithm is applied to learn an annotation model per class, which is able to give a score to each Visual Phrase that indicates whether it represents part of an object or not. Finally, the score of a new image is calculated using the scores obtained by its Visual Phrases according to the annotation model. The evaluation shows that using Visual Phrases only yields poorer results than the baseline (BoW on the whole images) but according to the authors mainly because Visual Phrases account only for the description of the objects while the baseline integrate also information about the background. A late fusion of recognition score for each image enhance the performance above baseline's initial results.

Spatial Pyramid Matching One of the successful approach to overcome the lack of spatial information within the BoW framework is the Spatial Pyramid Matching Kernel (SPMK) approach introduced in [LSP06]. The method is using the Pyramid Match Kernel [GD05] in order to compare images signatures according to a visual vocabulary but applying the pyramid construction to the coordinates of the features in the image space. The features are quantized into K discrete types according to a visual vocabulary V obtained by traditional clustering techniques in feature space. Only features discretized to the same channel k can be matched. For a pair of images X and Y to compare, each channel k gives two sets of two-dimensional vectors, X_k and Y_k, representing the coordinates of features of type k found in images X and Y respectively.

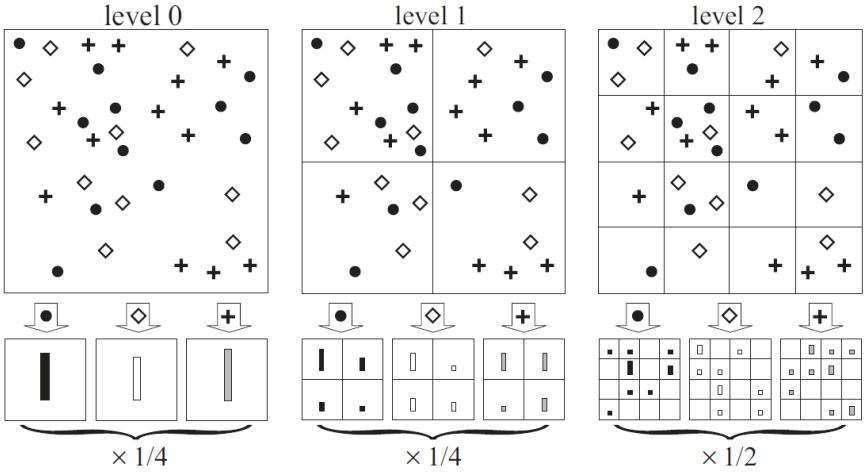


Figure 8.2.5: Toy example of constructing a three-level pyramid. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, the image is subdivided at three different levels of resolution. Next, for each level of resolution and each channel, we count the features that fall in each spatial bin. Image from [LSP06].

The kernel computation is illustrated in Figure 8.2.5. The final kernel (8.2.21) is the sum of the separate channel kernels. For $L = 0$ the approach is reduced to a standard bag-of-words approach. Experiments on three publicly available images databases show significant improvements using the Spatial Pyramid Matching approach. However, since locations are expressed in absolute coordinates, the representation is unsuitable in the case of spatial displacement of the object of interest unless exhaustive search is done using a spatial sub-window.

$$K^L(X, Y) = \sum_{k=1}^{K} K_\Delta^L(X_k, Y_k) \tag{8.2.21}$$

8.2.3.3 Relaxed Matching Kernels

A. Vedaldi has generalized in his thesis [Ved09] several kernel matching approaches for object recognition, including the Pyramid Match Kernel and the Spatial Pyramid Match Kernel, as «Relaxed Matching Kernels» that we will denote by RMK. The PMK is a RMK in the feature space domain while SPMK is a RMK in the spatial domain. The usual trade-off when choosing the resolution of the visual dictionary is that an excessively fine quantization causes features from two images to never match (over-fitting), while an excessively coarse quantization yields non-discriminative histograms (bias). The main idea of RMK is to overcome this trade-off by using different resolutions of vocabulary by building a sequence of R relaxed (coarser) dictionaries $V_0, V_1, \ldots, V_{R-1}$, where each word is obtained by merging words which are close in the finer vocabulary. The result of this

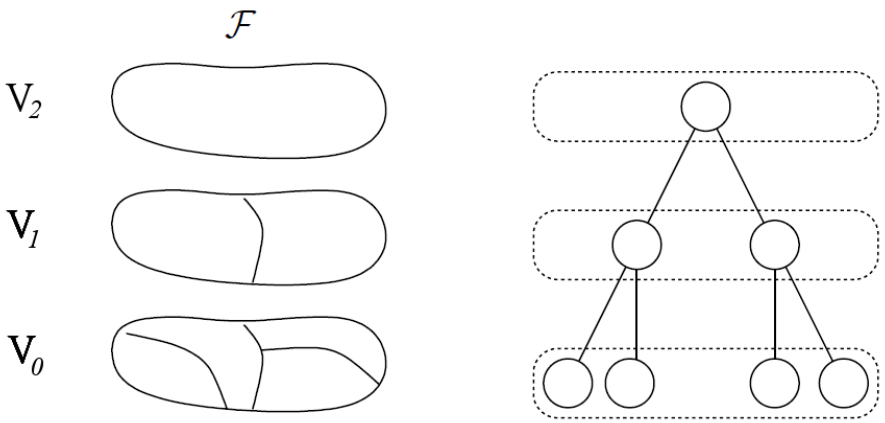


Figure 8.2.6: RMK construction: agglomerative tree.
Left. The feature space F and a sequence of three relaxations V_0, V_1 and V_2. *Right.* The agglomerative tree represents the merging operations transforming a relaxation to the next. Each relaxation V_r corresponds to a cut of the tree (dotted boxes). Figure from [Ved09] .

process is an agglomerative tree, see Figure 8.2.6. Each relaxed dictionary can be seen as a cut of the tree, having the property of preserving the mass of the dictionary:

$$\sum_{i=1}^{|V_0|} h_0^{(i)} = 1 = \sum_{j=1}^{|V_r|} h_r^{(j)} \tag{8.2.22}$$

Then, given two images I and J, a similarity measure S_r (8.2.23) is defined considering a base kernel k (8.2.24) and a multiscale approach computes a weighted score (8.2.25) of multiple relaxations as a positive combination of the BoF similarities at the various levels.

$$S_r = k\left(h_{I,r},\, h_{J,r}\right) \tag{8.2.23}$$

The base kernel, which may be the l_1 kernel k_1, the χ^2 kernel k_{χ^2} or the Hellinger's kernel k_H, compute the similarity at one relaxation level:

$$\begin{aligned} k_1(h_{I,r},\, h_{J,r}) &= \sum_{i=0}^{|V_r|} min\left(h_{I,r}^{(i)},\, h_{J,r}^{(i)}\right) \\ k_{\chi^2}(h_{I,r},\, h_{J,r}) &= 2\sum_{i=0}^{|V_r|} \frac{h_{I,r}^{(i)}\, h_{J,r}^{(i)}}{h_{I,r}^{(i)} + h_{J,r}^{(i)}} \\ k_H(h_{I,r},\, h_{J,r}) &= \sum_{i=0}^{|V_r|} \sqrt{h_{I,r}^{(i)}\, h_{J,r}^{(i)}} \end{aligned} \tag{8.2.24}$$

The weights w_r are positive and establish the relative importance of relaxations levels. This formulation yields a proper Mercer (positive definite) kernel : the «Relaxed Matching Kernel» (RMK).

$$K(h_I, h_J) = \sum_{r=0}^{R-1} w_r S_r \tag{8.2.25}$$

The «Relaxed Matching Kernel» gives a generalization of recent efficient approaches such as PMK and SPMK. The main idea behind relaxed matching is that, the difficulty of fixing some parameters of many methods can be overcame by a relaxed kernel exploring several parameters candidates and using a weighted scheme over the kernel similarities measurement to define the similarity between two images. The relaxations can be done for example on the bin width in PMK, on the size of dictionary in usual BoVW framework or on spatial grid definition in SPMK. Therefore, in this context two images are more likely to be similar if they match at all levels of the relaxations.

Conclusions

This non-exhaustive review of the state-of-the-art in the field of object recognition and image retrieval have mostly detailed the BoVW framework. The approach proposed in [SZ03] have influenced most of the recent works in this field. The method has been decomposed in four main stages: the construction of the visual dictionary, the feature quantization, the image representation and finally the image comparison. We have then reviewed some more recent work which have proposed improvements on the original method.

One of the limitation is the lack of spatial information in the final image representation. The reviewed approaches have proposed the construction of a set of local histograms according to a fix or data driven segmentation of the image. However, the integration of the spatial information being done at the last stage of the framework is also dependent of the quality of the quantization for single features.

We therefore have the feeling that incorporating spatial information before quantization could be interesting. This idea will be developed in the next chapter by presenting new semi-structural features for content description.

Chapter 9

Delaunay Graph Words

Introduction

The most successful approaches on object recognition rely on the Bag-of-Visual-Words framework which has been presented in the previous chapter. In this framework, images are represented by their distribution of visual words without taking into account any spatial information. Recent performance improvements when testing on academic databases have been achieved by adding spatial information within the Spatial Pyramid Match Kernel [LSP06]. However, this approach relies on a fix partitioning of the image, which induces its non invariance to affine transformations. This approach turned to be efficient when applied to most academic databases because they often represented centered objects within their usual contexts. However, in our first person video recordings and considering daily living objects that may be moved from one room to another, this approach does not seem applicable. Another integration of spatial information was presented in [PCI+07], where after applying a BoVW approach for retrieval the top ranked images where re-ranked by applying a LO-RANSAC [CMO04] algorithm with affine model transformations.

At the other end of the spectrum of methods adressing the problem of object recognition, the spatial information has often been incorporated by graph representation. The most common idea is to build a graph model of an object, the recognition process consisting in matching the prototype to a candidate one. In [RLYB10], a pseudo-hierarchical graph matching has been introduced. Using local interest points, the pseudo-hierachical aspect relies on progressively incorporating "smaller" model features (in terms of scale) as the hierarchy increases. The edges of the graph were defined accordingly to a scale-normalized proximity criterion. The model graph is matched to a new scene by a relaxation process starting from a graph model including only points of highest scale and adding smaller model features during the matching process. In [LKHH10], the graph model was defined according to locally affine-invariant geometric constraint. Each point is represented as an affine combination of its neighboring points. Defining an objective function taking into account both feature and geometric matching costs, the matching is solved by linear programming. These approaches are efficient for object matching, however when dealing with a large amount of image candidates, the matching process is too costly to be applied to all images.

We believe that integrating spatial information with local interest points in a BoVW

can be an elegant approach to overcome both the limitation of the BoVW framework and of object matching in case of large scale retrieval. Therefore, we will present new semi-structural approach for content description, by a bag of graph words.

9.1 New semi-structural features for content description

We present a method for integration of spatial information within the features by building local graphs upon local interest points and aim to integrate these new semi-local features in a BoVW framework. We will first detail the graph feature construction and then introduce our layered approach. In order to integrate these features in a BoVW framework we define a dissimilarity measure taking into account both nodes attributes and graph topology and then introduce our clustering approach.

The experiments on our videos necessitating a large amount of annotations, we will first evaluate the performances on academic databases that are relevant with regard to the final task. This aims to validate our approach in experiments similar as those presented in the literature before applying it to our videos. The application to videos has not been finalized yet, we will only present results on academic databases in this thesis dissertation.

9.1.1 Graph feature construction

Let us consider a graph $G = (X, E)$ with X a set of nodes corresponding to some feature points $x_{k,k=1,..,K}$, in image plane and $E = \{e_{kl}\}_{,k=1,..,K,l=1,..,K}$, where $e_{kl} = (x_k, x_l)$, a set of edges connecting these points. We call such a graph a "graph feature". We will build these features upon sets of neighboring feature points in image plane. Hence we propose a spatial embedding of local features with graphs. In order to build such graphs two questions have to be addressed:

- the choice of feature points sets X;
- the design of connectivity as edges E.

To define the feature point sets X upon which graphs will be built we are looking for a set of feature points that we call the "seeds". Around them, other feature points will be selected to build each graph feature. Selected seeds have to form a set of SURF points which are more likely to be detected in various instances of the same object. SURF points are detected where local maxima of the response of the approximated Hessian determinant are reached [BETVG08]. The amplitude of this criterion is a good choice for selecting the seeds, as SURF points with higher response correspond to more salient visual structures and are therefore more likely to be more repeatable. Hence, the seeds considered for building the graphs will be the SURF points with highest responses. Considering a fixed number of seeds N_{Seeds}, we can define the set of seeds S:

$$S = \{s_1, \ldots, s_{N_{Seeds}}\} \tag{9.1.1}$$

Given S, our aim is to add partial structural information of the object while keeping the discriminative power of SURF key points. We will therefore define graphs over the

seeds and their neighboring SURF points. Finding the k spatial nearest SURF neighbors of each seed s_i gives the set of neighbors P_i:

$$P_i = \{p_1, \ldots, p_k\} \tag{9.1.2}$$

Hence the set of nodes X^{G_i} for each graph G_i is defined as the seed s_i and the neighbors P_i, see (9.1.3). For the edges we use the Delaunay triangulation [She96] which is invariant with regard to affine transformations of image plane preserving angles: translation, rotation and scaling. Furthermore, regarding the future extensions of this work to video, the choice of Delaunay triangulation is also profitable for its good properties in tracking of structures [MBPB01]. The set of all vertices used for building the graph G_i is X^{G_i}, the union of the seed and its neighborhood:

$$X^{G_i} = \left\{x_1^{G_i}, \ldots, x_k^{G_i}\right\} = P_i \bigcup \{s_i\} \tag{9.1.3}$$

For a graph G, a Delaunay triangulation is computed on the points of X^G, building triangles according to the Delaunay constraint i.e. maximizing minimal angle of the triangulation. An edge $e_{ij} = (x_i^G, x_j^G)$ is defined between two vertices of the graph G if an edge of a triangle connects these two vertices.

9.1.2 The nested layered approach

The choice of the number of nodes in a graph feature obviously depends on various factors such as image resolution, complexity of visual scene, its sharpness... This choice is difficult a priori. Instead we propose a hierarchy of "nested" graphs for the same image, capturing structural information increasingly and illustrate it in Figure 9.1.1. Let us introduce a set of L "layers". We say that the graph G_i^l at layer l and the graph G_i^{l+1} at layer $l+1$ are nested if the set of nodes of graph G_i^l is included in the set of nodes of graph G_i^{l+1}: $X_i^l \subset X_i^{l+1}$. Note that, so defined, the number of graphs at each layer is the same. Furthermore, in the definition (by construction) of graph features a node can belong to more than one graph of the same layer. We still consider these graph features as separate graphs.

Introducing this layered approach, where each layer adds more structural information, we can define graphs of increasing size while moving from one layer to the next one. Each layer has his own set of neighbors around each seed s_i and the Delaunay triangulation is run separately on each layer. To avoid a large number of layers, the number of nodes added at each layer should induce a significant change of structural information. To build a Delaunay triangulation, at least two points have to be added to the seed at the second layer. Adding one more node may yield three triangles instead of just one, resulting in a more complete local pattern. Therefore, the number of nodes added from one layer to the upper one is fixed to three. We define four layers, the bottom one containing only one SURF point, the seed, and the top one containing a graph built upon the seed and its 9 nearest neighbors, see examples in Figure 9.1.2.

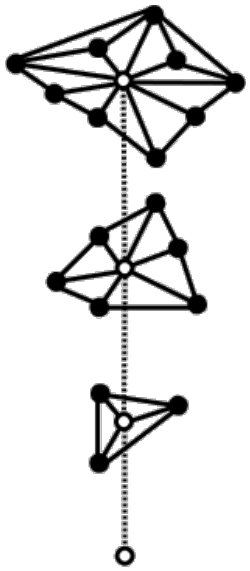


Figure 9.1.1: The nested approach. Bottom to top: SURF seed depicted as the white node, 3 neighbors graph where neighbors are in black, 6 neighbors graph and 9 neighbors graph at the top level.

(a) SURF features (b) 3-nearest neighbors graphs (c) 6-nearest neighbors graphs (d) 9-nearest neighbors graphs

Figure 9.1.2: SURF and graph features on a cropped image of the object "ajaxorange" from SIVAL database.

9.1.3 Graph comparison

In order to integrate these new graph features in a Bag-of-Visual-Words framework a dissimilarity measure and a clustering method have to be defined. In this section, we define the dissimilarity measure. We are dealing with attributed graphs, where nodes can be compared with respect to their visual appearance. Although it could be possible to take into account similarities of node features only or the topology of the graph only, more information can be obtained by combining both information for defining a dissimilarity measure between local graphs. To achieve this we will investigate the use of the Context Dependent Kernel (CDK) presented in [SARK08]. The definition of the CDK relies on two matrices: D which contains the distances between node features, and T which contains the topology of the graphs being compared. Considering two graphs A and B with respective number of nodes m and n, let us denote C the union of the two graphs:

$$\begin{aligned} & C = A \oplus B \\ & \text{with} \begin{cases} x_i^C = x_i^A & \text{for} \quad i \in [1..m] = I_A \\ x_i^C = x_{i-m}^B & \text{for} \quad i \in [m+1..m+n] = I_B \end{cases} \end{aligned} \tag{9.1.4}$$

with I_A and I_B, the sets of indices of each graph nodes.

The feature correspondence square matrix D of size $(m+n) \times (m+n)$ contains the "entrywise" L2-norm (i.e., the sum of the squared values of vector coefficients) of the difference between SURF features:

$$\begin{aligned} D &= (d_{ij})_{ij} \\ \text{where } d_{ij} &= \left\| x_i^C - x_j^C \right\|_2 \end{aligned} \tag{9.1.5}$$

The square topology matrix T of size $(m+n) \times (m+n)$ defines the connectivity between two vertices $x_i^C$and x_j^C. In this work we define a crisp connectivity as we set T_{ij} to one if an edge connects the vertices x_i^C and x_j^C and 0 otherwise. Hence, only sub matrices where both lines and columns in I_A or I_B are not entirely null. More precisely, we can define sub matrices T_{AA} and T_{BB} corresponding to the topology of each graph A and B respectively, while sub matrices T_{AB} and T_{BA} are entirely null, vertices of graphs A and B are not connected.

$$\begin{aligned} T &= (T_{ij})_{ij} \\ \text{where } T_{ij} &= \begin{cases} 1 & \text{if edge } (x_i^C,\, x_j^C) \text{ belongs to A or B} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{9.1.6}$$

The CDK denoted K is computed by an iterative process consisting of the propagation of the similarity in the description space according to the topology matrix.

$$
\begin{aligned}
K^{(0)} &= \frac{exp(\frac{-D}{\beta})}{\left\|exp(\frac{-D}{\beta})\right\|_1} \\
K^{(t)} &= \frac{G(K^{(t-1)})}{\left\|G(K^{(t-1)})\right\|_1} \\
G(K) &= exp(-\frac{D}{\beta} + \frac{\alpha}{\beta} T K^{(t-1)} T)
\end{aligned}
\tag{9.1.7}
$$

Where exp represents the coefficient-wise exponential and $\|M\|_1 = \sum_{ij} |M_{ij}|$ represents the L_1 matrix norm. The two parameters β and α can be seen respectively as weights for features distance and topology propagation. Similarly to the definition of sub matrices in topology matrix T we can define sub matrices in the kernel matrix K. The sub matrix $K^{(t)}_{AB}$ represents the strength of the inter-graph links between graphs A and B once the topology has been taken into account. We can therefore define the dissimilarity measure that will be used for clustering:

$$
\begin{aligned}
s(A,B) &= \sum\nolimits_{\{i \in I_A,\, j \in I_b\}} K^{(t)}_{ij} \in [0,1] \\
\rho(A,B) &= s(A,A) + s(B,B) - 2s(A,B) \in [0,1]
\end{aligned}
\tag{9.1.8}
$$

This dissimilarity measure will be applied separately on each layer. However, for the bottom layer, since there is no topology to take into account for isolated points we will use directly the "entrywise" L2-norm of the difference between SURF features denoted by d. This corresponds to an approximation of the dissimilarity measure used for graphs features by considering a graph with a single point. We prove this point following for a pair of graph points A and B with features vectors x_A and x_B. The CDK K will be constructed as :

$$
\begin{aligned}
K^{(0)} = \frac{exp(-\frac{D}{\beta})}{\left\|exp(-\frac{D}{\beta})\right\|_1} \quad \text{where} \quad D &= \begin{pmatrix} 0 & d \\ d & 0 \end{pmatrix} \\
\text{Let } E = exp(-\frac{D}{\beta}) \quad &= \quad \begin{pmatrix} 1 & e^{-d/\beta} \\ e^{-d/\beta} & 1 \end{pmatrix} \\
K^{(0)} = \frac{E}{\|E\|_1} \quad &= \quad \frac{E}{2(1+e^{-d/\beta})}
\end{aligned}
\tag{9.1.9}
$$

When considering the dissimilarity measure from (9.1.8) and apply (9.1.9) we obtain :

$$
\begin{aligned}
\rho(A,B) &= s(A,A) + s(B,B) - 2s(A,B) \\
\rho(A,B) &= \frac{2 - 2e^{-d/\beta}}{2(1+e^{-d/\beta})} = \frac{1 - e^{-d/\beta}}{1+e^{-d/\beta}}
\end{aligned}
\tag{9.1.10}
$$

Now,

$$\text{Let } t = \frac{d}{\beta}, \text{ if } \beta \gg d :$$
$$\rho(A,B) = \frac{2e^{-t}}{(1+e^{-t})^2} \mid_{t=0} \; t + o(t) \approx \frac{d}{2\beta} \tag{9.1.11}$$

Hence for degenerated graph features (points) the dissimilarity measure $\rho(A,B)$ is proportional to their distance in the L2 metric description space. Therefore the comparison of degenerated graphs points with L2 norm of the difference of their feature vectors is justified.

9.2 Visual dictionaries

The state-of-the-art approach for computing the visual dictionary of a set of features is the use of the K-means clustering algorithm [SZ03] with a large number of clusters, often several thousands. The code-word is either the center of a cluster or a non-parametric representation like a K-Nearest Neighbors (K-NN) voting approach. These approaches are not suitable for the graph-features because using the K means clustering algorithm implies iteratively moving the cluster centers with interpolation and defining a mean graph is a difficult task. Morevoer, a fast K-NN requires an indexing structure which is not available in our graph feature space since it is not a vector space. Therefore, we present in the following the method we choose for building the code book which is a two pass agglomerative hierarchical clustering [Ser96]. The model of a cluster is chosen to be thus its median instead of the mean.

9.2.1 Clustering method

In order to quantize a very large database, it can be interesting to use a two pass clustering approach as proposed in [GCPF08], as it enables a gain in terms of computational cost. Here, the first pass of the agglomerative hierarchical clustering will be run on all the features extracted from training images of one object. The second pass is applied on the clusters generated by the first pass on all objects of the database. To represent a cluster, we use the following definition of the median:

$$median = \underset{G \in V}{argmin} \sum_{i=1}^{m} \|v_i - G\| \tag{9.2.1}$$

With V – a cluster and v_i – members of a cluster, G the candidate median and $\|\cdot\|$ being a distance or dissimilarity measure in our case. For the first pass, the dissimilarities between all the features, of the same layer, extracted from all the images of an object are computed. For the second pass, only the dissimilarities between all the medians of all object clusters are computed. Each layer being processed independently, we obtain a visual dictionary for each layer of graphs with 1, 3,. . . ,N_{max} nodes.

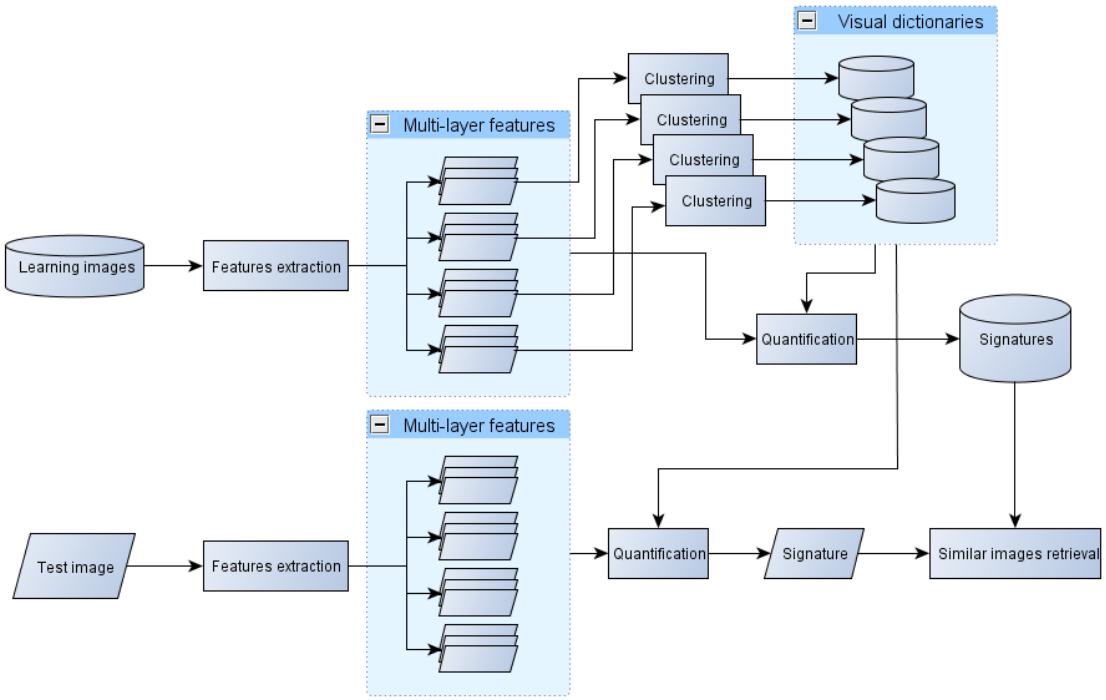


Figure 9.2.1: Flowchart of the Bag-of-Words framework applied to our multilayer features.

9.2.2 Visual signatures

The usual representation of an image in a BoVW approach is to compute a histogram of all the visual words of the dictionary within the image. Each feature extracted from an image is assigned to the closest visual word of the dictionary. We use this representation without rejection, a feature is always assigned to a word in the dictionary. The signatures are then normalized to sum to one by dividing each value by the number of features extracted from the image. Once the visual signatures of images have been computed, one can define the distance between two images as the distance between their visual signatures. In preliminary experiments we have compared results when using Hamming distance, Euclidean distance and L_1 distance for this task. The L_1 distance giving better results, final results are presented using this measure only.

An overview of the proposed method is illustrated in Figure 9.2.1.

Conclusion

In this chapter, we have presented new graph features built upon SURF points as nodes and expressing spatial relations between local key points. The multi-layer approach using growing neighborhoods in several layers enables to capture the most discriminative visual information for different types of objects.

Chapter 10

Experiments on Object Recognition

Introduction

The final goal of the proposed method will be the application to our videos. However, the videos from our corpora are very difficult due to strong motion and lighting changes. Moreover, working with our videos would induce a heavy annotation cost in order to obtain a amount of annotated objects which would be significant to evaluate our method.

Therefore, we propose to evaluate our approach on publicly available data sets. The choice of the data sets are guided by the need of annotated object as this work only focus on object recognition and not detection, and also by the representativity towards the final task i.e. recognition of daily living objects in videos. We present the two selected data sets in 10.1.

We will evaluate our method on a retrieval task i.e. having a query example of an object, retrieve similar objects within the database. The details on the evaluation protocol are given in 10.2.

Finally, the results of the experiments will be analyzed by comparing first the SURF BoW approach with our proposition of graph words in 10.3 and then comparing both with the results of the proposed nested approach in 10.4.

10.1 Data sets

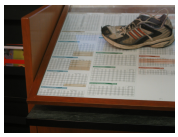
To evaluate our method we needed to find public data sets with labeled objects. The experiments were therefore conducted on two publicly available data sets with object annotations. The first one, the SIVAL (Spatially Independent, Variable Area, and Lighting) data set [RGZ+05] includes 25 objects, each of them being present in 60 images taken in 10 various environment and different poses yielding a total of 1500 images. This data set is quite challenging as the objects are depicted in various lighting conditions and poses. It has also been chosen as the longer term perspective of this work is the recognition of objects of the every day life that may appear in different places of a house, for example a hoover that may be moved in all the rooms in one's house. The second one is the well known Caltech-101 [FFFP06] data set, composed of 101 object categories. The categories

(a) DirtyWorkGloves

(b) StriperNotebook

(c) GlazedWoodPot

(d) GoldMedal



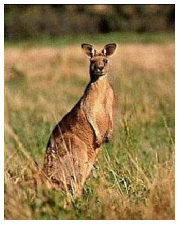
(e) DirtyRunningShoe

(f) Mandolin

(g) Schooner

(h) Canon

(i) Airplanes



(j) Kangaroo

Figure 10.1.1: Excerpts from image data sets. SIVAL (a)-(e), Caltech-101 (f)-(j)

are different types of animals, plants or objects. A snippet of both data sets is shown in Figure 10.1.1a and b.

10.2 Evaluation protocol

We separate learning and testing images by a random selection. On each data set, 30 images of each category are selected as learning images for building the visual dictionaries and for the retrieval task. Some categories of Caltech-101 have several hundred of images when others have only a few more than 30. The testing images are therefore a random selection of the remaining images up to 50. We only take into account the content of a bounding box of each object as this work only deals with object recognition and not localization yet. SURF key points of 64 dimensions are extracted within the bounding box, the numbers of seeds for the graphs building process is fixed to 300. The second layer corresponds to graphs built upon the seeds and their 3 nearest neighbors, the third layer with the 6 nearest neighbors and the fourth and last layer with the 9 nearest neighbors. For the CDK, α is set to 0.0001, β to 0.1 (ensuring K is a proper kernel) and the number of iterations is fixed to 2, as H. Sahbi [SARK08] has shown that the convergence of the CDK is fast. The first pass clustering compute 500 clusters for each object. The final dictionary size varies in the range 50-5000. Each layer will yield its own dictionary. We compare our method with standard BoVW approach. For that purpose, we use all the SURF features available on all images of the learning database to build the BoVW dictionary. The visual words are obtained by performing k-means clustering on the set of all these descriptors. Each visual word is characterized by the center of a cluster.

The graph features are not built using all available SURF points, therefore to analyze the influence of this selection, signatures are computed for the set of SURF which have been selected to build the different layers of graphs. These configurations will be referred to as SURF3NN, SURF6NN and SURF9NN corresponding respectively to all the points upon which graphs with 3, 6 and 9 nearest neighbors have been defined. In this case, as

for graphs, the dictionaries are built with our two-pass clustering approach.

For each query image and each database image, the signatures are computed for isolated SURF and the different layers of graphs. We have investigated the combination of isolated SURF and the different layers of graphs by an early fusion of signatures i.e. concatenating the signatures. For SIVAL that concatenation has been done with the signature from the selected SURF corresponding to the highest level whereas for Caltech-101 we used the classical BoW SURF signature. Finally, the L_1-distance between histograms is computed to compare two images.

The performance is evaluated by the Mean Average Precision (MAP) measure. For each test image, all images in the learning set are ranked from the closest (in terms of L_1 distance between visual signatures) to the furthest. The average precision AP aims to evaluate how well the target images, i.e images of the same class as the query, are ranked amongst the n retrieved images:

$$AP = \frac{\sum_{k=1}^{n} P(k) \times rel(k)}{c_p}$$

where $rel(k)$ equals 1 when the k^{th} ranked image is a target image and 0 otherwise and c_p is the total number of target images as defined in Table 7.2.1.The average precision is evaluated for each test image of an object, and the MAP is the mean of these values for all the images of an object in the test set. For the whole database we measure the performance by the average value of the MAP i.e. we do not weight the MAP per class by the number of query which would give more consideration to categories where more testing images are present.

10.3 SURF based BoW vs Graphs Words

First of all, it is interesting to analyze if the graph words approach obtains similar performances compared to the classical BoVW approach using only SURF features. This is depicted in Figure 10.3.1, Figure 10.3.3 and 10.3.4 where isolated SURF points are depicted as dotted lines, single layer of graph words are dashed lines and the combination of SURF and different graphs layers are plotted as continuous lines. At first glance, we can see that for SIVAL isolated SURF features perform the poorest, separated layers of graphs perform better and the combination of different layers of graphs and the SURF features upon which the highest layer has been computed obtain the best performances. Our clustering approach seems to give worse results for very small size of dictionaries but better results for dictionaries larger than 500 visual words, which are the commonly used configurations in BoVW approaches. Each layer of graph words performs much better than the SURF upon which they are built. The introduction of the topology in our features have a significant impact on the recognition performance using the same set of SURF features.

The average performance hides however some differences in the performance of each feature on some specific objects. To illustrate this we select two object categories where graph features and SURF features give different performances in Figure 10.3.3 and Figure 10.3.4. For the object “banana” from SIVAL, the isolated SURF features outperform

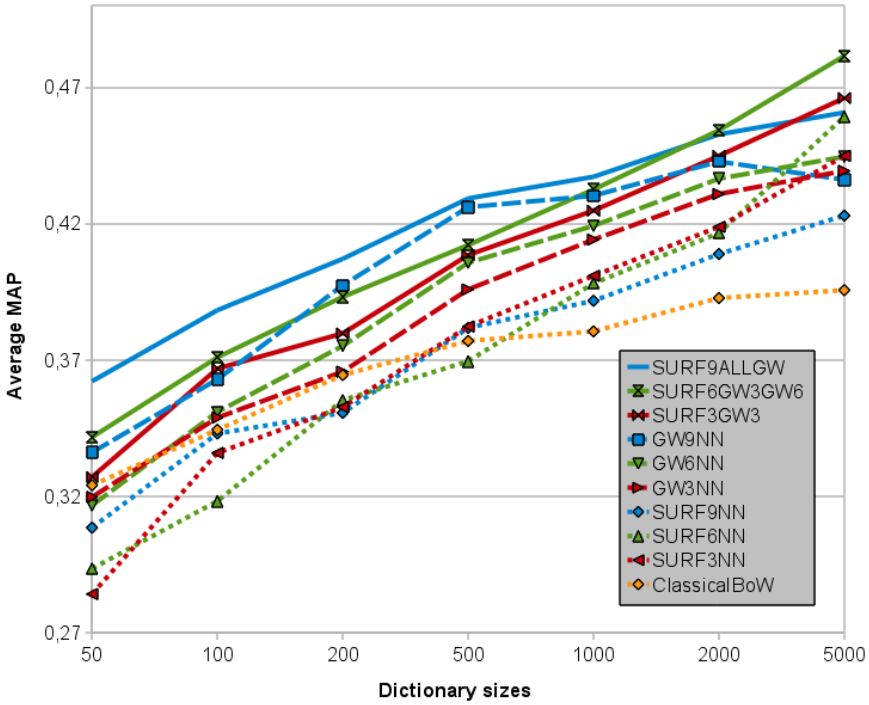


Figure 10.3.1: Average MAP on the whole SIVAL data set. Isolated SURF features are the dotted curves, single layer Graphs Words are drawn as dashed curves and the multilayer approach in solid curves.

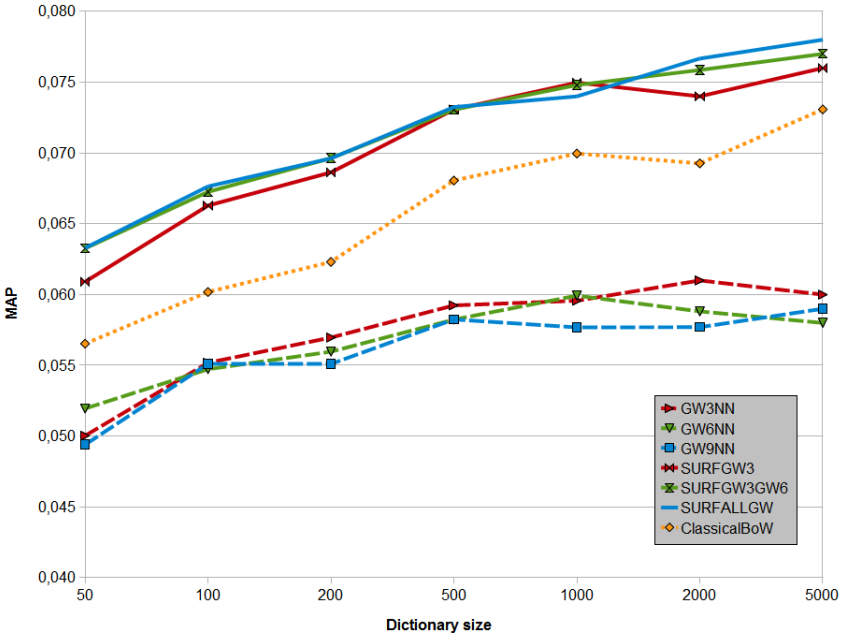


Figure 10.3.2: Average MAP on the whole Caltech-101 data set. Isolated SURF features are the dotted curves, single layer Graphs Words are drawn as dashed curves and the multilayer approach in solid curves.

the graph approach, see Figure 10.3.3. This can be explained as the "banana" object represent a small part of the bounding box and is a poorly textured object. In some environments the background is highly textured, this characteristics induce many SURF points detected in the background and these SURF points may have a higher response than those detected on the object. This will lead to the construction of many graph features on the background and less on the object, see Figure 10.3.5. On the other hand, for the "Faces" category from Caltech-101 the graphs features perform better, see Figure 10.3.4. Here, the object covers most of the bounding box and many SURF points are detected. In this situation, the graph features capture a larger part of the object than isolated SURF points, making them more discriminative, see Figure 10.3.6.

This unequal discriminative power of each layer leads naturally to the use of the combination of the different layers in a single visual signature.

10.4 The multilayer approach

The combination of graphs and SURF features upon which the graphs have been built is done by the concatenation of the signatures of each layer. The three curves in solid lines in Figure 10.3.1 correspond to the multilayer approach using only the two bottom layers (SURF + 3 nearest neighbors graphs) in red, the three bottom layers (SURF + 3 nearest neighbors graphs + 6 nearest neighbors) in green and all the layers in blue. For SIVAL, the improvement in the average MAP is clear, and each addition of layer improves the results. The average performance of the combination always outperforms the performance

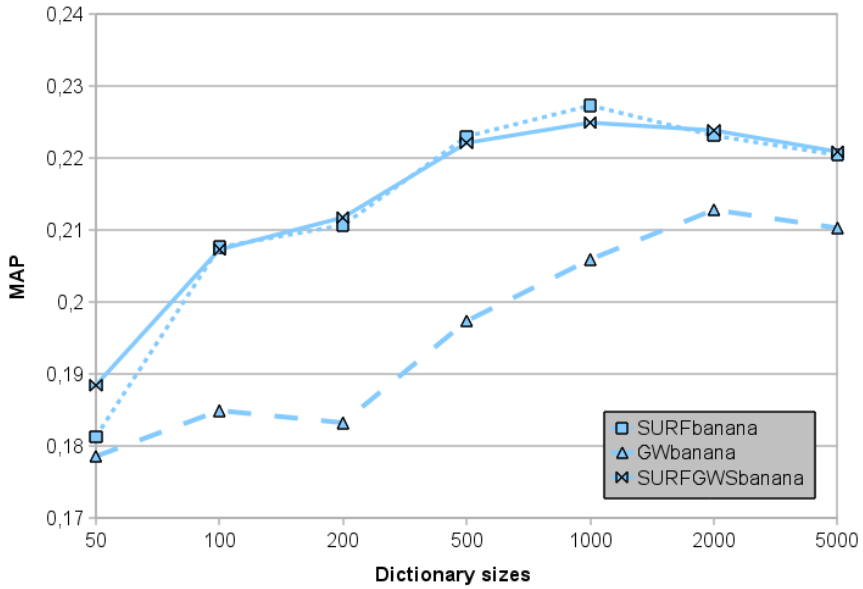


Figure 10.3.3: MAP for the object "banana" from SIVAL where isolated SURF features (dotted curves) outperforms graphs (dashed curves). The multilayer approach is the solid curve.

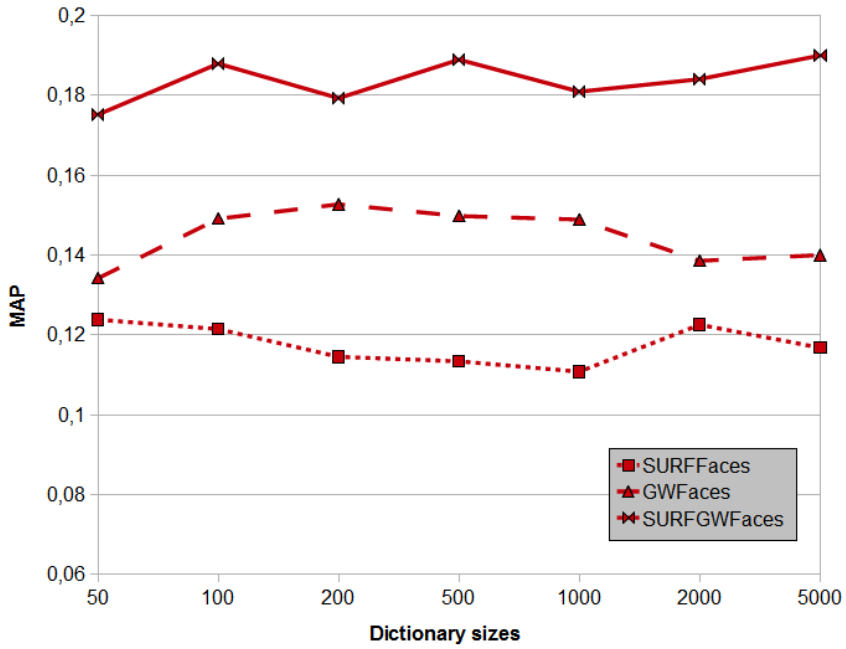


Figure 10.3.4: MAP for category "Faces" from Caltech-101 where graphs (dashed curves) outperforms isolated SURF features (dotted curves). The multilayer approach is the solid curves.

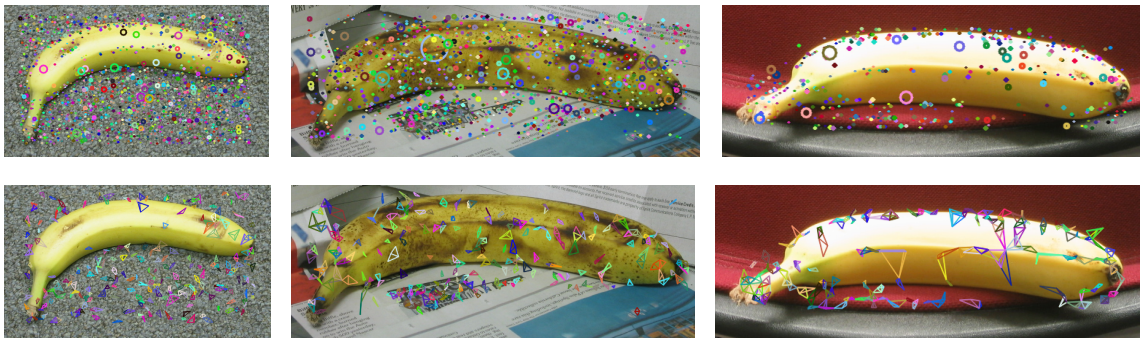


Figure 10.3.5: 3 images from category “Banana” from SIVAL. Top: SURF features within the bounding box. Bottom: graphs features.

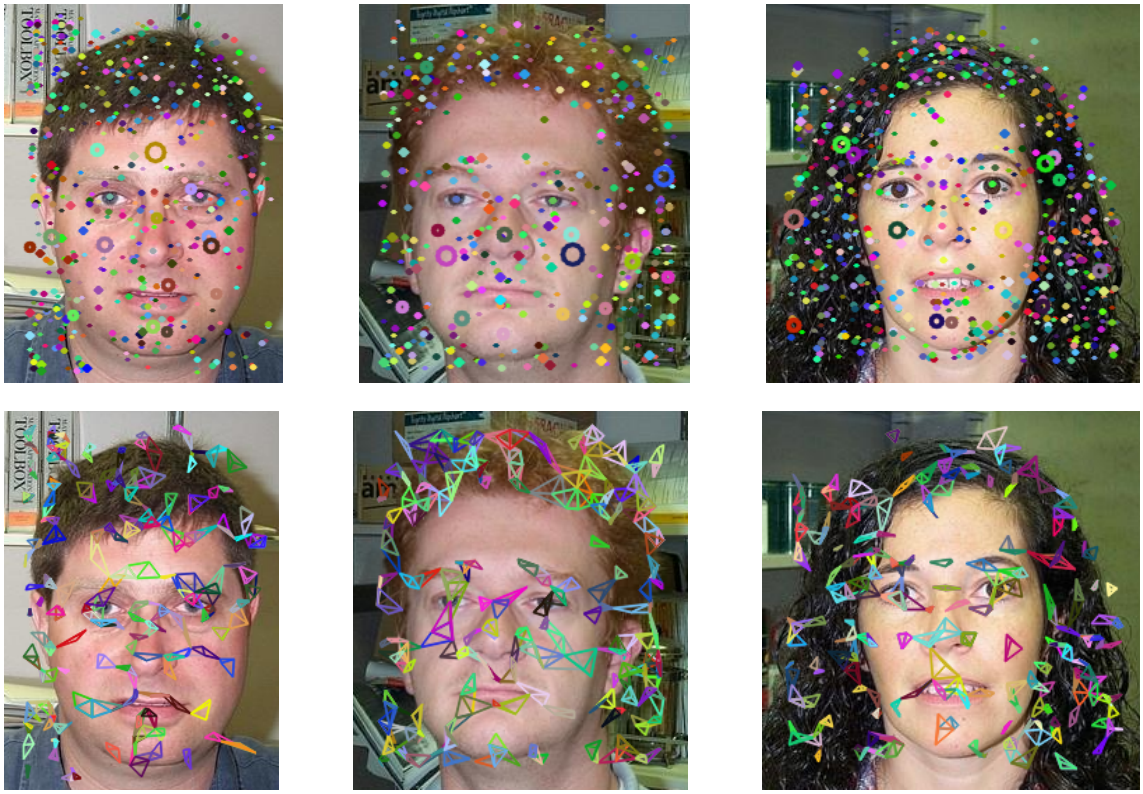


Figure 10.3.6: 3 images from category “Faces” from Caltech-101. Top: SURF features within the bounding box. Bottom: graphs features.

of each layer taken separately.

For Caltech-101, see Figure 10.3.2, the average MAP values of all methods are much lower which is not surprising as there are much more categories and images. Single layer of graphs gives results in the range 0.050-0.061 while the classical BoVW framework on SURF features performances are within 0.057-0.073 of average MAP values. The combination of all layers outperforms here again SURF or graphs used separately with average MAP values in the range of 0.061-0.077. The performance of single layers of graphs can be explained as the fixed number (300) of seeds selection induces for Caltech-101 a strong overlapping of graphs as the average number of SURF points within the bounding box is only 427 when it was more than a thousand for SIVAL. This may give less discriminant graph words as it will be harder to determine separable clusters in the clustering process. This combined with the hard quantization used in the experiment can explained these results.

The detailed results presented in Figure 10.3.3 and Figure 10.3.4 show that the combination of the visual signatures computed on each layer separately performs better or at least as well as the best isolated feature.

Conclusion

In this chapter we have evaluated the proposed graph word approach on two public databases: SIVAL and Caltech-101. The recognition performance was shown to improve by using both visual and topological information inside the graph features. Using growing spatial neighborhood clearly improves the results while each layer taken separately yields smaller improvements. Moreover, this approach introduces spatial information within the features themselves and is therefore complementary and compatible with other recent improvements of the BoW framework that takes geometry into account, such as the Spatial Pyramid Matching [LSP06].

The future of the work on object recognition is the application of the method to the recognition of objects in videos. The approach could be enhanced by refining some steps of the graphs construction and comparison. For instance, the selection of seeds could be performed by an adaptive method and the topology matrix be defined with a soft connectivity. In order to be efficient when processing a large amount of images, i.e. in videos, an indexing structure should be used as that would speed up the recognition process. A graph embedding procedure could be applied in order to use indexing structure already existing for features represented by vectors.

Conclusion

In this part, we proposed a solution for object recognition that combines the advantages of the BoVW approach and semi-structural representations. After reviewing and analyzing the strength and limits of the BoVW approach, we have identified the high potential of incorporating spatial context in this framework. As opposed to representations that operate a quantization on low level local features, we proposed to exploit more structural features: local Delaunay graph words which extend local features with contextual topological information.

This construction can keep the invariance properties with regard to planar affine transformations. It also takes the advantage of the BoVW approach, by providing a final descriptor that is efficient to index while incorporating information from medium level features.

The experiments show the potential of our proposition in challenging image databases that were chosen to include as much as possible the variability that can be encountered in wearable videos.

Thanks to this encouraging properties, we believe that our proposition introduces a promising paradigm that can be used in future works to improve the quality of object recognition when applied to videos.

Chapter 11

Conclusions and perspectives

Hence, in this PhD tightly related to the ANR-09-BLAN-0165-02 IMMED project, we proposed solutions for video indexing according to the objective of recognition of activities in videos recorded with wearable cameras.

The recognition of IADL, which are activities of high semantic level, required powerful models. The HHMs, through the Markov property, are adapted for the video analysis. Hence, we proposed two-level HHMM to model complex activities. The proposed model is advantageous compared to a full Hierachical HMM since it does require training of a lower number of parameters. In our method, the bottom-level HMM models non semantic elementary activities with emitting states. The upper-level HMM models IADL as states and transitions in between. Furthermore, the "temporal" dimension of video sequences was taken into account by a motion-based pre-segmentation of the video stream into a set of segments. This solution is an alternative to segmental HMMs requiring heavy computational overload.

Many content descriptors from the literature reviewed in this manuscript were hardly applicable in our context. Therefore we introduced a set of descriptors carrying more information from the video stream such as motion, audio and visual clues. The experiments conducted in a controlled environment and on a large-scale real world data set have shown the efficiency of the method and the discriminative power of the proposed descriptors and their combinations when enough representative learning data are available.

The interest of our approach resides in combining both low-level descriptors and mid-level descriptors resulting from a pre-analysis of the stream (e.g. in audio or in localization) . To go further in this direction to incorporate high-level semantic features, we considered the detection of semantic objects. Their presence in video is strongly correlated with activities.

We have therefore investigated the existing approaches for object recognition with the aim to develop a robust method with regard to the challenging conditions of our videos. The state-of-the-art methods for the task of object recognition can be categorized in two main classes: i) BoVW related approaches which define a visual dictionary and represent the image by the distribution of these visual words and ii) structural approaches mainly using graph matching methods. The BoVW framework is very efficient when dealing with large amount of data thanks to the indexing possibilities given by the visual words distribution representation but is operating on very local features and most of the time discards spatial information. The structural approaches describe an object as a whole

graph which is very distinctive, but the application of graph matching to large data sets is untractable.

We thus introduced new structural features on the basis of feature points. We proposed to build local graphs by Delaunay triangulation, hence preserving invariance of local features with regard to affine transformations of the image plane, and integrated these new semi-local features in a BoVW framework yielding the definition of "graphs words". A layered approach, where each layer incorporates more structural information, has been introduced. The application to two academic data sets has shown the higher discriminative power of the proposed approach compared to a standard BoVW.

We believe that the definition of mid-level features (such as object detection) that captures partial semantics from the scene will help defining powerful contextual information and hence lead to better recognition of activities. While adding more features, an evolution of the framework may become necessary. The definition of coherent and homogenous description subspace should be done, and these newly created modalities could be modeled separately.

The direct perspectives of this PhD research are obviously to adapt the object recognition method to wearable video content and its incorporation in the whole HMM framework. Here several problems have to be addressed: how medical practitioners observe video content? What are the salient features for them? These questions on content saliency pose interesting perspectives for bringing the object recognition techniques such as those presented to yield more semantic value on the analysis.

The second short-term perspective is in the question: can the global HMM framework evolve towards a "flow of concepts", leaving aside low-level features? In order to do this we have to define coherent and homogeneous description subspaces such as "dynamic", "static", etc. Then the fusion can be done by e.g. HMM in the decision space i.e. the results of preliminary classifications.

The combination of our methodology with external observations and other sensors fusion seems promising and is the subject of a starting European project Dem@Care, which was initiated on the basis of the concept we proposed in this PhD.

In a further future, our system could evolve in a more proactive way. It would not only observe and analyze the elderly people's activities but also interact with them and predict their further actions using the history of recognized activities, thus becoming an advising tool.

Appendix

In this appendix we detail the methods for SIFT and SURF interest points detection and description.

Details on SIFT

We will here detail each step of the SIFT points detection and description process.

Scale-space extrema detection The first stage of keypoint detection is to identify locations and scales that can be repeatably assigned under differing views of the same object. Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales. Lowe uses the Gaussian function as the scale-space kernel. The scale space of an image is defined as a function, $L(x, y, \sigma)$, that is produced from the convolution of a variable-scale Gaussian, $G(x, y, \sigma)$, with an input image $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{11.0.1}$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} exp^{-(x^2+y^2)/2\sigma^2} \tag{11.0.2}$$

where $*$ is the convolution operation in x and y, and σ is the scale parameter. To efficiently detect stable keypoint locations in scale space, Lowe used scale-space extrema in the difference-of-Gaussian function convolved with the image, $D(x, y, \sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{11.0.3}$$

The maxima and minima of the scale-normalized Laplacian of Gaussian, $\sigma^2\nabla^2 G$, produce the most stable image features compared to a range of other possible image functions, such as the gradient, Hessian, or Harris corner function [MS02]. The difference-of-Gaussian function D provides a close approximation to the scale-normalized Laplacian of Gaussian [Lin94]. As illustrated in Figure 11.0.1, the difference-of-Gaussian function D can

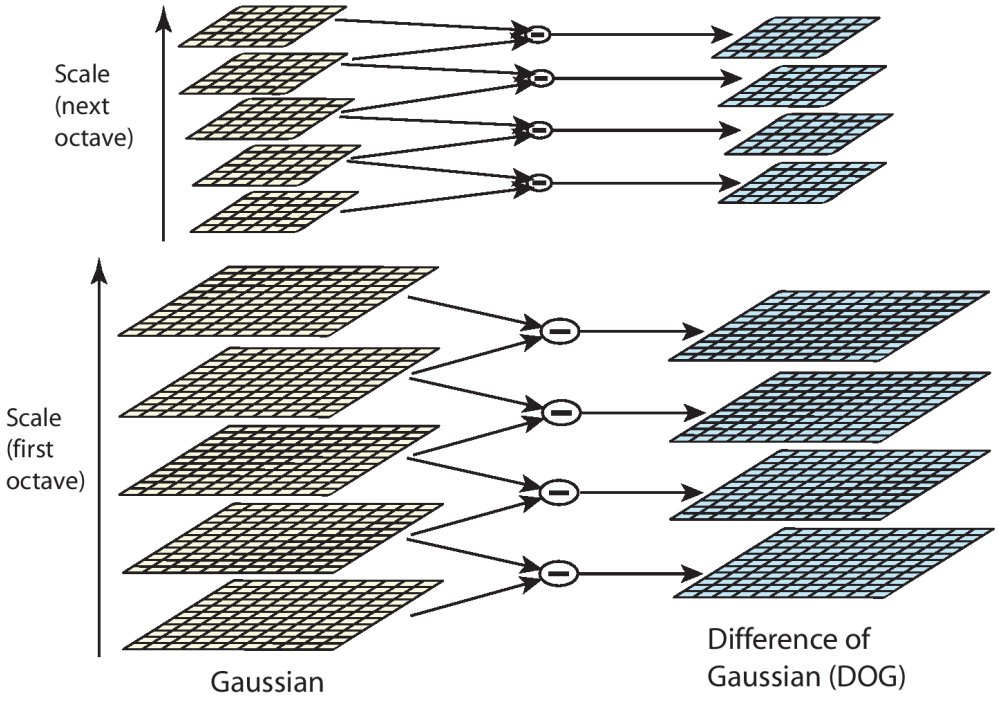


Figure 11.0.1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated. Image from [Low04].

be computed efficiently by simple image subtraction between smoothed images obtained from the scale space image function L defined in (11.0.1). Each octave of scale space (i.e., doubling of σ) is divided into an integer number, s, of intervals, so $k = 2^{1/s}$.

In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below (see Figure 11.0.2). It is selected only if it is larger or smaller than all of these neighbors. The cost of this check is reasonably low due to the fact that most sample points will be eliminated following the first few checks.

Keypoint localization and filtering

The local maxima and minima are key-points candidates but some of them may have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. The filtering of these poorly defnined key-points candidates is done by performing a detailed fit to the nearby data for location, scale, and ratio of principal curvatures.

The computation of the detailed fit for location and scale uses the method developed by Brown in [BL02] for fitting a 3D quadratic function to the local sample points to determine the interpolated location of the maximum, and his experiments showed that

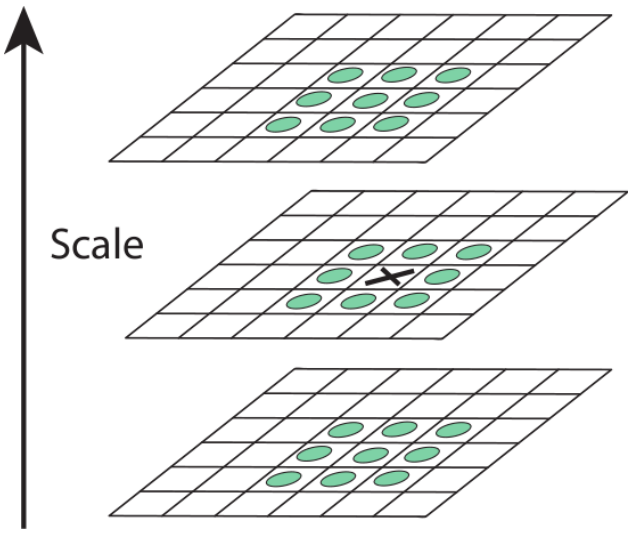


Figure 11.0.2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles). Image from [Low04].

this provides a substantial improvement to matching and stability. His approach uses the Taylor expansion (up to the quadratic terms) of the scale-space function, $D(x, y, \sigma)$, shifted so that the origin is at the sample point:

$$D(\mathtt{x}) = D + \frac{\partial D}{\partial \mathtt{x}}^T \mathtt{x} + \frac{1}{2}\mathtt{x}^T \frac{\partial^2 D}{\partial \mathtt{x}^2}\mathtt{x} \tag{11.0.4}$$

where D and its derivatives are evaluated at the sample point and $\mathtt{x} = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum, $\hat{\mathtt{x}}$, is determined by taking the derivative of this function with respect to $\mathtt{x}$ and setting it to zero, giving

$$\hat{\mathtt{x}} = -\frac{\partial^2 D}{\partial \mathtt{x}^2}^{-1} \frac{\partial D}{\partial \mathtt{x}} \tag{11.0.5}$$

The offset $\hat{\mathtt{x}}$ is added to the location its sample point to get the interpolated estimate for the location of the extremum. A filtering of the keypoints obtained after extrema localization is performed in order to keep the most meaningful keypoints. The function value at the extremum, $D(\hat{\mathtt{x}})$, is useful for rejecting unstable extrema with low contrast. Substituting equation (11.0.5) into equation (11.0.4), we have:

$$D(\hat{\mathtt{x}}) = D + \frac{1}{2}\frac{\partial D}{\partial \mathtt{x}}^T \hat{\mathtt{x}} \tag{11.0.6}$$

Low contrast extrema, i.e. those with a value of $|D(\hat{\mathtt{x}})| < K$ with K a threshold value are filtered out.

Finally, points located on the edges, which yields a strong response of the Difference-of-Gaussian function might be poorly located along the edges. A poorly defined peak in the Difference-of-Gaussian function will have a large principal curvature across the edge but a small one in the perpendicular direction. Keypoints that have a ratio between their principal curvatures higher than a threshold are discarded. To evaluate this ratio a 2x2 Hessian matrix **H** is computed at the location and scale of the keypoint:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$$

The derivatives D_{xx}, D_{yy} and D_{xy} are estimated by taking differences of neighboring sample points. The eigenvalues of **H** are proportional to the principal curvatures of D. Since only the ratio r between the larger magnitude eigenvalue and the smaller one is needed, the explicit computation of the eigenvalues can be avoided. Let α be the eigenvalue with the largest magnitude and β the smaller one, so $\alpha = r\beta$. The sum of the eigenvalues can be computed from the trace of **H** and the product from the determinant:

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - D_{xy}^2 = \alpha\beta$$

Therefore to discard a keypoint which have large principal curvature across the edge but a small one in the perpendicular direction, one only need to check the constraint of (11.0.7) for a chosen value of r.

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r} \tag{11.0.7}$$

Orientation assignment Invariance to image rotation is a desirable property for the keypoint descriptors. By assigning a consistent orientation to each keypoint based on local image properties, invariance to rotation is achieved. The method proposed by Lowe for orientation assignment is the following: the scale of the keypoint is used to select the Gaussian smoothed image L with the smallest scale so that all computations are performed in a scale invariant manner. For each image sample, $L(x, y)$ at this scale, the gradient magnitude (11.0.8) and orientation (11.0.9) is precomputed using pixel differences:

$$m(x, y) = \sqrt{(L(x-1, y) - L(x+1, y))^2 + (L(x, y-1) - L(x, y+1))^2} \tag{11.0.8}$$

$$\theta(x, y) = arctan \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \tag{11.0.9}$$

The gradient orientations of sample points within a region around the keypoint are collected into an histogram of 36 bins covering the full 360 degrees range of orientations. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint. The orientation assigned to the keypoint corresponds to the orientation of the highest peak in the histogram. If any other peak in the histogram is within 80% of the highest peak, a new keypoint is created with the same location and scale but with this different orientation. Lowe experimented that even if multiple orientation assignment is quite rare (about 15% of the keypoints) it contributes significantly to the stability of matching. The orientation is accurately computed by fitting a parabola to the 3 histogram values closest to the peak.

Keypoint description The previous steps have led to the computation of repeatable stable keypoints which are assigned a location, scale and orientation. The method of computation ensures robustness to affine transformations and noise in the image. The next step is to compute a descriptor for the local image region that is highly distinctive yet is as invariant as possible to remaining variations, such as change in illumination or 3D viewpoint. The descriptor computation was inspired by the work of Edelman, Intrator, and Poggio [EIP97], which have shown that in a model of biological vision, the perception of 3D objects is driven by the orientation and spatial frequency of gradients but the location of the gradient is allowed to be shifted.

The computation of the keypoint descriptors is illustrated in Figure 4.3.2. First the image gradient magnitudes and orientations are sampled around the keypoint location, using the scale of the keypoint to select the level of Gaussian blur for the image. In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation. These are illustrated with small arrows at each sample location on the left side of Figure 4.3.2. A Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point as illustrated by the circular window on the left of Figure 4.3.2. The Gaussian weighting avoids sudden changes in the descriptor with small changes of the position and decreases the influence of gradient samples that are far from the center of the window.

The keypoint descriptor is shown on the right side of 4.3.2. It allows for significant shift in gradient positions by creating orientation histograms over 4x4 sample regions. The figure shows eight directions for each orientation histogram, with the length of each arrow corresponding to the magnitude of that histogram entry. A gradient sample on the left can shift up to 4 sample positions while still contributing to the same histogram on the right, thereby achieving the objective of allowing for larger local positional shifts. The descriptor is formed from a vector containing the values of all the orientation histogram entries, corresponding to the lengths of the arrows on the right side of Figure 4.3.2. The figure shows a 2x2 array of orientation histograms, whereas the original sampling for SIFT descriptors is achieved with a 4x4 array of histograms with 8 orientation bins in each. Therefore, the each keypoint is described using a 4x4x8 = 128 dimensional feature

vector. The vector is normalized to unit length, being therefore invariant to contrast and linear illumination changes. To cope with non-linear illumination changes which would affect mostly large gradient magnitudes, Lowe reduces the influence of large gradient magnitudes by thresholding the values in the unit feature vector to each be no larger than 0.2, and then renormalizing the descriptor vector to unit length.

Details on SURF

We will here first detail the idea and use of integral images and then review the whole keypoint detection and description process of SURF.

Integral images Every entry of an integral image $I_\Sigma(\mathbf{x})$ is the sum of all pixels value contained in the rectangle between the origin (top-left corner) to the current position $\mathbf{x} = (x, y)^T$, equation (11.0.10). The integral image can be computed in linear time using an incremental algorithm. With an integral image, the sum of intensities in any rectangular region of the initial image can be computed in only three additions and four memory accesses, see Figure 11.0.3. Hence, the calculation time is independent of the rectangle size.

$$I_\Sigma(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,\, j) \tag{11.0.10}$$

Detection of keypoint SURF points detection relies on a Hessian-matrix approximation. Blob-like structures are detected at locations where the determinant of the Hessian-matrix is maximum. Given a point $\mathbf{x} = (x,\, y)$ in an image I, the Hessian matrix $\mathcal{H}(x,\, \sigma)$ in $\mathbf{x}$ at scale is defined as follows:

$$\mathcal{H}(x,\, \sigma) = \begin{bmatrix} L_{xx}(x,\, \sigma) & L_{xy}(x,\, \sigma) \\ L_{xy}(x,\, \sigma) & L_{yy}(x,\, \sigma) \end{bmatrix} \tag{11.0.11}$$

where $L_{xx}(x,\, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image I in point $\mathbf{x}$, and similarly for $L_{xy}(x,\, \sigma)$ and $L_{yy}(x,\, \sigma)$. The computation of Gaussian convolutions of parts of discrete images necessarily results in an approximation. Bay proposed to use box-filters, which is a stronger approximation, but has a computational cost independent of size filters using integral images. This box-filter approach shows performances which are comparable or better in terms of repeatability with regard to rotation than with the discretised and cropped Gaussians. The 9x9 box filters in Figure 11.0.4 are approximations of a Gaussian with $\sigma = 1.2$ and represent the lowest scale (i.e. highest spatial resolution) for computing the blob response maps. The approximate Gaussian second order derivative computed with box filters are denoted D_{xx}, D_{yy} and D_{xy}. The determinant of the Hessian-matrix is expressed in (11.0.12) where w is a relative weight of filter responses used to balance the expression for the Hessian's determinant. This is needed for the energy conservation between the Gaussian kernels and the

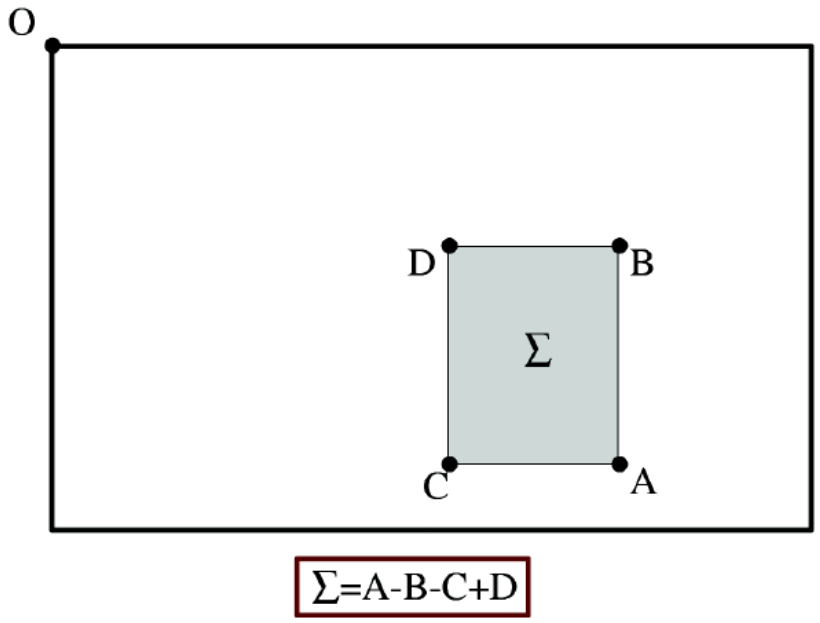


Figure 11.0.3: Using integral images, it takes only three additions and four memory accesses to calculate the sum of intensities inside a rectangular region of any size. Image from [BETVG08].

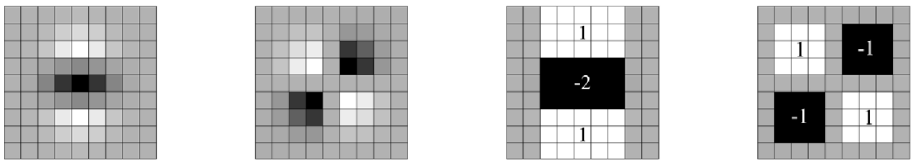


Figure 11.0.4: Left to right: the (discretised and cropped) Gaussian second order partial derivative in y- (L_{yy}) and xy-direction (L_{xy}), respectively; Bay's approximation for the second order Gaussian partial derivative in y- (D_{yy}) and xy-direction (D_{xy}). The grey regions are equal to zero. Image from [BETVG08].

approximated Gaussian kernels. The filter responses are normalized with respect to their size. The approximated determinant of the Hessian represents the blob response in the image at location `x`. These responses are stored in a blob response map over different scales.

$$det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{11.0.12}$$

The method for SIFT key-points presented by Lowe needs the computation of convolutions at different scales. More precisely, in the work of Lowe the scale space is represented as a pyramid of images resulting from convolutions of the original image with Gaussian filters of growing σ and then sub-sampled. The layers of the pyramid are subtracted in order to get the DoG (Difference of Gaussians) images where edges and blobs can be found.

The scale space is divided into octaves. An octave represents a series of filter response maps obtained by convolving the same input image with a filter of increasing size. Each octave is subdivided into a constant number of scale levels. In total, an octave encompasses a scaling factor of 2. Finally, in order to localize interest points in the image and over scales, a non-maximum suppression in a 3x3x3 neighborhood is applied.

Orientation assignment

SURF keypoint are assigned an orientation to ensure rotation invariance. The Haar Wavelet response to x and y directions are computed in a circular windows of size $6s$ around the interest point, with s the scale of the keypoint. Using integral images, Haar Wavelet are computed efficiently as only six operations are needed to compute the response in x or y direction at any scale. The wavelet responses are weighted with a Gaussian of $\sigma = 2s$ centered at the interest point. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window covering an angle of $\pi/3$.

Keypoint description

The extraction of the descriptor is done considering an oriented square window centered at the interest point of size $20s$, Figure 4.3.4. This region is split up into 4x4 sub-regions. The Haar wavelet response in the horizontal and vertical direction (with respect to the keypoint orientation) are computed within each sub-region. The responses are weighted with a Gaussian $\sigma = 3.3s$ centered at the interest point to increase robustness. Then, each sub-region yields a feature vector of size 4 consisting of the sum of the wavelet responses in the x and y directions, and the sum of the absolute value of the wavelet responses in the x and y directions. If we note dx, dy the wavelet responses, the feature vector for a sub-region is $\mathtt{x} = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$, Figure 11.0.5. The total SURF descriptor hence is a 4x4x4 = 64-dimensional feature vector. The descriptor is normalized to the unit vector to ensure invariance to contrast. An extended version of the SURF descriptor can be computed by summing the positive and negative wavelet responses separately. The extended feature vector is hence of size 128.

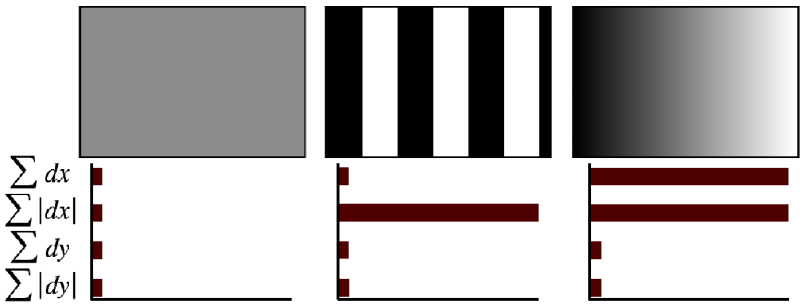


Figure 11.0.5: The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |dx|$ is high, but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum dx$ and $\sum |dx|$ are high.

Bibliography

[AB95] A. Adjoudani and C. Benoit. Audio-visual speech recognition compared across two architectures. In *Fourth European Conference on Speech Communication and Technology*, 1995.

[AMC10] R. Albatal, P. Mulhem, and Y. Chiaramella. Visual phrases for automatic images annotation. In *Content-Based Multimedia Indexing (CBMI), 2010 International Workshop on*, pages 1–6. IEEE, 2010.

[AO88] R. André-Obrecht. A new statistical approach for the automatic segmentation of continuous speech signals. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36(1):29–40, 1988.

[AOJS96] R. André-Obrecht, B. Jacob, and C. Senac. How merging acoustic and articulatory informations to automatic speech recognition. In *EUSIPCO'96*, Trieste (Italie), 10-13 Septembre 1996.

[ASP99] H. Aoki, B. Schiele, and A. Pentland. Realtime personal positioning system for wearable computers. In *iswc*, page 37. Published by the IEEE Computer Society, 1999.

[AV07] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[BB95] S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433–466, 1995.

[BBDBS09] L. Ballan, M. Bertini, A. Del Bimbo, and G. Serra. Video event classification using bag of words and string kernels. *Image Analysis and Processing–ICIAP 2009*, pages 170–178, 2009.

[BBLP10] Y.L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. 2010.

[BD01] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.

[BDLR+06] Y. Bengio, O. Delalleau, N. Le Roux, J.F. Paiement, P. Vincent, and M. Ouimet. Spectral dimensionality reduction. *Feature Extraction*, pages 519–550, 2006.

[BDMGO92] F. Brugnara, R. De Mori, D. Giuliani, and M. Omologo. A family of parallel hidden markov models. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 377–380. IEEE, 1992.

[BE67] L.E. Baum and J.A. Egon. An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. In *Bull. Amer. Meteorology Soc.*, volume 73, pages 360–363. IEEE, 1967.

[BETVG08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[BGCG+92] P. Barberger-Gateau, D. Commenges, M. Gagnon, L. Letenneur, et al. Instrumental activities of daily living as a screening tool for cognitive impairment and dementia in elderly community dwellers. *Journal of the American Geriatrics Society*, 1992.

[BGS+05] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. 2005.

[BKW+07] E. Berry, N. Kapur, L. Williams, S. Hodges, P. Watson, G. Smyth, J. Srinivasan, R. Smith, B. Wilson, and K. Wood. The use of a wearable camera, sensecam, as a pictorial diary to improve autobiographical memory in a patient with limbic encephalitis: A preliminary report. *Neuropsychological Rehabilitation, 17*, 4(5):582–601, 2007.

[BL02] M. Brown and D.G. Lowe. Invariant features from interest point groups. In *British Machine Vision Conference, Cardiff, Wales*, pages 656–665. Citeseer, 2002.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 509–522, 2002.

[BMW+11] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. *International Workshop on Human Behavior Understanding*, pages 29–39, 2011.

[BN78] H. Blum and R.N. Nagel. Shape description using weighted symmetric axis features. *Pattern recognition*, 10(3):167–180, 1978.

[BOP97] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *cvpr*, page 994. Published by the IEEE Computer Society, 1997.

[BPSW70] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

[BS68] L.E. Baum and G.R. Sell. Growth functions for transformations on manifolds. In *Pacific J. Math.*, volume 27, pages 211–227, 1968.

[BSI08] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[Bur98] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[BW98] J.S. Boreczky and L.D. Wilcox. A hidden markov model framework for video segmentation using audio and image features. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 6, pages 3741–3744. IEEE, 1998.

[CDF+04] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22. Citeseer, 2004.

[CHD97] K. Cameron, K. Hughes, and K. Doughty. Reducing fall incidence in community elders by telecare using predictive systems. In *Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*, volume 3, pages 1036–1039. IEEE, 1997.

[CKV08] S.P. Chatzis, D.I. Kosmopoulos, and T.A. Varvarigou. Robust sequential data modeling using an outlier tolerant hidden markov model. *IEEE transactions on pattern analysis and machine intelligence*, pages 1657–1669, 2008.

[CMO04] O. Chum, J. Matas, and S. Obdrzalek. Enhancing ransac by generalized model optimization. In *Proc. of the ACCV*, volume 2, pages 812–817, 2004.

[CMP00] B. Clarkson, K. Mase, and A. Pentland. Recognizing user context via wearable sensors. In *Wearable Computers, 2000. The Fourth International Symposium on*, pages 69–75. IEEE, 2000.

[CS95] C. Cedras and M. Shah. Motion-based recognition a survey. *Image and Vision Computing*, 13(2):129–155, 1995.

[DA94] R. David and H. Alla. Petri nets for modeling of dynamic systems:: A survey. *Automatica*, 30(2):175–202, 1994.

[DBP01] M. Durik and J. Benois-Pineau. Robust motion characterisation for video indexing based on mpeg2 optical flow. *Proc. of Content Based Multimedia Indexing, CBMI'01*, pages pp. 57–64, 2001.

[DF08] Y. Ding and G. Fan. Multi-channel segmental hidden markov models for sports video mining. In *Proceeding of the 16th ACM international conference on Multimedia*, pages 697–700. ACM, 2008.

[DGG08] M. Delakis, G. Gravier, and P. Gros. Audiovisual integration with segment models for tennis video parsing. *Computer vision and image understanding*, 111(2):142–154, 2008.

[DHMV09] Fernando De la Torre, Jessica K. Hodgins, Javier Montano, and Sergio Valcarcel. Detailed human data acquisition of kitchen activities: the cmu-multimodal activity database (cmu-mmac). In *Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, in conjuction with CHI 2009*, 2009.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[DMB11] Vladislavs Dovgalecs, Rémi Mégret, and Yannick Berthoumieu. Time-aware Co-Training for Indoors Localization in Visual Lifelogs. In *ACM Multimedia 2011*, Scottsdale, United States, November 2011.

[DRCB05] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005.

[DSGP03] R. DeVaul, M. Sung, J. Gips, and A.S. Pentland. Mithril 2003: Applications and architecture. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, page 4. IEEE Computer Society, 2003.

[DT05] 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*, volume 1, pages 886–893. Ieee, 2005.

[DWV99] H. Drucker, D. Wu, and V.N. Vapnik. Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5):1048–1054, 1999.

[EBMM03] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. 2003.

[EIP97] S. Edelman, N. Intrator, and T. Poggio. Complex cells and object recognition. *Unpublished manuscript: http://kybele.psych.cornell.edu/ edelman/archive. html*, 1997.

[FD96] A. Foucault and P. Deleglise. Système acoustico-labial de reconnaissance automatique de la parole. In *Journées d'Etude de la Parole*, 1996.

[FFFP06] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 594–611, 2006.

[FJ73] G.D. Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3): 268–278, 1973.

[Fre61] H. Freeman. On the encoding of arbitrary geometric configurations. *Electronic Computers, IRE Transactions on*, (2):260–268, 1961.

[FST98] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

[GCPF08] P.H. Gosselin, M. Cord, and S. Philipp-Foliguet. Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval. *Computer Vision and Image Understanding*, 110(3):403–417, 2008.

[GD05] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. October 2005.

[GK95] G.H. Granlund and H. Knutsson. *Signal processing for computer vision*. Kluwer Academic Pub, 1995.

[GSS03] F.A. Gers, N.N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *The Journal of Machine Learning Research*, 3:115–143, 2003.

[GVSG10] JV Gemert, C. Veenman, A. Smeulders, and J. Geusebroek. Visual word ambiguity. *IEEE PAMI*, 2010.

[GY93] M.J.F. Gales and S.J. Young. *The theory of segmental hidden Markov models*. University of Cambridge, Department of Engineering, 1993.

[Hac94] V. Hachinski. Vascular dementia: A radical redefinition. *Dementia*, 1994.

[HBS07] T. Huynh, U. Blanke, and B. Schiele. Scalable recognition of daily activities with wearable sensors. *Location-and context-awareness*, pages 50–67, 2007.

[HLK09] N. Harte, D. Lennon, and A. Kokaram. On parsing visual sequences with the hidden markov model. *EURASIP Journal on Image and Video Processing*, pages 1–13, 2009.

[HPL+06] C. Helmer, K. Peres, L. Letenneur, LM. Guttierez-Robledo, H. Ramaroson, P. Barberger-Gateau, C. Fabrigoule, JM. Orgogoz, and JF. Dartigues. Dementia in subjects aged 75 years or over within the paquid cohort: prevalence and burden by severity. 1(22):87–94, 2006.

[HS04] A. Hakeem and M. Shah. Ontology and taxonomy collaborated framework for meeting classification. *Pattern Recognition*, 4:219–222, 2004.

[Hu62] M.K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[Huy08] D.T.G. Huynh. Human activity recognition with wearable sensors. 2008.

[HWB+06] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood. Sensecam: A retrospective memory aid. *UbiComp 2006: Ubiquitous Computing*, pages 177–193, 2006.

[IB00] Y.A. Ivanov and A.F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):852–872, 2000.

[JLZZ02] F. Jing, M. Li, H.J. Zhang, and B. Zhang. An effective region-based image retrieval framework. In *Proceedings of the Tenth ACM international conference on Multimedia*, pages 456–465. ACM, 2002.

[Jou95] P. Jourlin. Automatic bimodal speech recognition. In *Proceedings of the International Congress of Phonetic Sciences, ICPhS'95*, 1995.

[JT05] Frederic Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision*, 2005.

[JV87] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[KBP+05] P. Krämer, J. Benois-Pineau, et al. Camera motion detection in the rough indexing paradigm. In *TREC Video Retrieval Evaluation Online Proceedings, TRECVID*, volume 5. Citeseer, 2005.

[KGG+03] E. Kijak, G. Gravier, P. Gros, L. Oisel, and F. Bimbot. Hmm based structuring of tennis videos using visual and audio cues. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 3, pages III–309. IEEE, 2003.

[KSH05] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. 2005.

[KSS03] N. Kern, B. Schiele, and A. Schmidt. Multi-sensor activity context detection for wearable computing. *Ambient Intelligence*, pages 220–232, 2003.

[Lap05] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[LB69] M.P. Lawton and E.M. Brody. Assessment of older people: self-maintaining and instrumental activities of daily living. *The gerontologist*, 9(3 Part 1): 179, 1969.

[Lew98] D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *ECML-98: 10th European Conference on Machine Learning*, pages 4–15. Springer, April 1998.

[Lin94] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *J. of Applied Statistics*, 21(2):224–270, 1994.

[LKF10] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.

[LKHH10] H. Li, E. Kim, X. Huang, and L. He. Object matching with a locally affine-invariant constraint. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1641–1648. IEEE, 2010.

[Llo57] SP Lloyd. Least square quantization in pcm. bell telephone laboratories paper. published in journal much later: Lloyd., sp (1982). least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2): 129–137, 1957.

[LM01] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[LS06] Z. Liu and S. Sarkar. Improved gait recognition by gait dynamics normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 863–876, 2006.

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. Ieee, 2006.

[LZF03] F. Long, H. Zhang, and D.D. Feng. Fundamentals of content-based image retrieval. *Multimedia Information Retrieval and Management*, 17:1–26, 2003.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[MBPB01] A. Mahboubi, J. Benois-Pineau, and D. Barba. Joint tracking of polygonal and triangulated meshes of objects in moving sequences with time varying content. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 403–406. IEEE, 2001.

[MCUP04] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22 (10):761–767, 2004.

[MDF+84] G. McKhann, D. Drachman, M. Folstein, R. Katzman, D. Price, and E.M. Stadlan. Clinical diagnosis of alzheimer's disease. *Neurology*, 34(7):939, 1984.

[ME02] D. Moore and I. Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *Proceedings of the National Conference on Artificial Intelligence*, pages 770–776. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.

[MM05] WW Mayol and DW Murray. Wearable hand activity recognition for event summarization. In *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, pages 122–129. IEEE, 2005.

[MMMP02] H. Müller, S. Marchand-Maillet, and T. Pun. The truth about corel-evaluation in image retrieval. *Image and Video Retrieval*, pages 38–49, 2002.

[MN88] A.A. Markov and N.M. Nagornyi. *The theory of algorithms*. Kluwer Academic, 1988.

[MOVY01] B.S. Manjunath, J.R. Ohm, V.V. Vasudevan, and A. Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703–715, 2001.

[MS83] M.J. McGill and G. Salton. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[MS98] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1376–1381, 1998.

[MS02] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. *Computer Vision - ECCV 2002*, pages 128–142, 2002.

[MSBP+08] R. Megret, D. Szolgay, J. Benois-Pineau, P. Joly, J. Pinquier, J.F. Dartigues, and C. Helmer. Wearable video monitoring of people with age dementia: Video indexing at the service of helthcare. In *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pages 101–108. IEEE, 2008.

[MSS02] BS Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7: multimedia content description interface*, volume 1. John Wiley & Sons Inc, 2002.

[NPVB05] N.T. Nguyen, D.Q. Phung, S. Venkatesh, and H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. 2005.

[NS06] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2161–2168. IEEE, 2006.

[NW70] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[ODK96] M. Ostendorf, V.V. Digalakis, and O.A. Kimball. From hmm's to segment models: A unified view of stochastic modeling for speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 4(5):360–378, 1996.

[OF97] B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[OR89] M. Ostendorf and S. Roukos. A stochastic segment model for phoneme-based continuous speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(12):1857–1869, 1989.

[ORP98] N. Oliver, B. Rosario, and A. Pentland. Statistical modeling of human interactions. In *CVPR Workshop on Interpretation of Visual Motion*, pages 39–46. Citeseer, 1998.

[PAO06] J. Pinquier and R. André-Obrecht. Audio indexing: primary components retrieval. *Multimedia tools and applications*, 30(3):313–330, 2006.

[PCI+07] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[PFKP05] D.J. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, pages 44–51. IEEE, 2005.

[PFP+04] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, pages 50–57, 2004.

[PHA+08] K. Peres, C. Helmer, H. Amieva, J.-M. Orgogozo, I. Rouch, J.-F. Dartigues, and P. Barberger-Gateau. Natural history of decline in instrumental activities of daily living performance over the 10 years preceding the clinical diagnosis of dementia: A prospective population-based study. *Journal of the American Geriatrics Society*, 56(1):37–44, 2008.

[PJZ01] M. Petkovic, W. Jonker, and Z. Zivkovic. Recognizing strokes in tennis videos using hidden markov models. In *Proceedings of Intl. Conf. on Visualization, Imaging and Image Processing, Marbella, Spain*. Citeseer, 2001.

[PNB+07] L. Piccardi, B. Noris, O. Barbey, A. Billard, G. Schiavone, F. Keller, and C. von Hofsten. Wearcam: A head mounted wireless camera for monitoring gaze attention and for the diagnosis of developmental disorders in young children. In *International Symposium on Robot & Human Interactive Communication*, 2007.

[PP91] KM Ponting and SM Peeling. The use of variable frame rate analysis in speech recognition. *Computer Speech & Language*, 5(2):169–179, 1991.

[PS98] C.H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publications, 1998.

[QBPM+08] G. Quenot, J. Benois-Pineau, B. Mansencal, E. Rossi, M. Cord, D. Gorisse, F. Precioso, P. Lambert, B. Augereau, L. Granjon, D. Pellerin, M. Rombaut, and S. Ayache. Rushes summarization by irim consortium: redundancy removal and multi-feature fusion. In *ACM International Conference on Multimedia*, pages 1–20, Vancouver, BC, Canada, oct 2008. ACM.

[RA06] M.S. Ryoo and J.K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. 2006.

[Rab89] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[RGZ+05] Rouhollah Rahmani, Sally A. Goldman, Hui Zhang, John Krettek, and Jason E. Fritts. Localized content based image retrieval. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, MIR '05, pages 227–236, New York, NY, USA, 2005. ACM. ISBN 1-59593-244-5.

[RLYB10] J. Revaud, G. Lavoué, A. Yasuo, and A. Baskurt. Scale-invariant proximity graph for fast probabilistic object recognition. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 414–421. ACM, 2010.

[SAK10] H. Sahbi, J.Y. Audibert, and R. Keriven. Context-dependent kernels for object classification. *IEEE transactions on pattern analysis and machine intelligence*, pages 699–708, 2010.

[SARK08] H. Sahbi, J.Y. Audibert, J. Rabarisoa, and R. Keriven. Robust matching and recognition using context-dependent kernels. In *Proceedings of the 25th international conference on Machine learning*, pages 856–863. ACM, 2008.

[SC09] S. Sundaram and WWM Cuevas. High level activity recognition using low resolution wearable vision (pdf). 2009.

[SCB+06] C.N. Scanaill, S. Carew, P. Barralon, N. Noury, D. Lyons, and G.M. Lyons. A review of approaches to mobility telemonitoring of the elderly in their living environment. *Annals of Biomedical Engineering*, 34(4):547–563, 2006.

[Ser96] Arturo Serna. Implementation of hierarchical clustering methods. *Journal of computational physics*, 129, 1996.

[SG86] B. W. Silverman and P. J. Green. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

[She96] J. Shewchuk. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. *Applied Computational Geometry Towards Geometric Engineering*, pages 203–222, 1996.

[SHVLS08] M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele. Adl recognition based on the combination of rfid and accelerometer sensing. In *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, pages 258–263. IEEE, 2008.

[SPL+07] D. Surie, T. Pederson, F. Lagriffoul, L.E. Janlert, and D. Sjölie. Activity recognition using an egocentric perspective of everyday objects. *Ubiquitous Intelligence and Computing*, pages 246–257, 2007.

[SSM98] B. Schölkopf, A. Smola, and K.R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[SZ03] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, October 2003.

[TCSU08] P. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

[TD08] S. Tran and L. Davis. Event modeling and recognition using markov logic networks. *Computer Vision–ECCV 2008*, pages 610–623, 2008.

[Ved09] A. Vedaldi. *Invariant representations and learning for computer vision*. PhD thesis, University of California, Los Angeles, 2009.

[Vit67] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.

[VR79] CJ Van Rijsbergen. Information retrieval. *Journal of the American Society for Information Science*, 30(6):374–375, 1979.

[WCM05] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. 2005.

[YEK+97] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK book*, volume 2. Citeseer, 1997.

[YOI92] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992.

[YY94] S.J. Young and S. Young. The htk hidden markov model toolkit: Design and philosophy. In *Entropic Cambridge Research Laboratory, Ltd.* Citeseer, 1994.

[YYGH09] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[ZBT+09] N. Zouba, F. Brémond, M. Thonnat, A. Anfosso, E. Pascual, P. Mallea, V. Mailland, and O. Guerin. A computer system to monitor older adults at home: preliminary results. 2009.

[ZBTV07] N. Zouba, F. Bremond, M. Thonnat, and V.T. Vu. Multi-sensors analysis for everyday activity monitoring. *Proc. of SETIT*, pages 25–29, 2007.

[ZMI01] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–123. IEEE, 2001.

[ZYCA06] Q. Zhu, M.C. Yeh, K.T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.

[ZZN+08] Y.T. Zheng, M. Zhao, S.Y. Neo, T.S. Chua, and Q. Tian. Visual synset: towards a higher-level visual representation. pages 1–8, 2008.