Marco Cristani

Statistical generative modelling of audio-video sequences for scene analysis

Ph.D. Thesis

11th April 2006

Università degli Studi di Verona
Dipartimento di Informatica

Advisor:
Prof. Vittorio Murino

Series N°: **TD-03-06**

Università di Verona
Dipartimento di Informatica
Strada le Grazie 15, 37134 Verona
Italy

Contents

1

Introduction

Video analysis and understanding is undoubtedly an important research area, whose interest has grown in the last decade, promoting a set of interesting applications, each one characterized by different goals.
In general, when describing a signal, ideally the model should mimic the underlying process that is thought to generate it. The advantage is that the inferences over such model can easily be interpretable, applicable and generalizable to other signals.
This is mainly true when the signal is a video or an audio sequence representing human activities: in this case, the aim of a good model is to encode as parameters meaningful gestures, words, actions and the like, performing classification and recognition tasks in an intuitive way.
Recently, video processing systems have become expressive benchmarks of how automatic systems could model general complex dynamic visual events. An optimal system should perform analysis at the same level of complexity and semantics that a human would employ while analyzing the content. Moreover, the currently available hardware resources may allow an ideal video processing system to easily outperform human abilities, providing superior grade analysis. Anyway, such commercially successful systems do not exist yet, because humans assimilate visual information at semantic level using a mapping that is highly subjective, and is not formally completely known, kinds of features which are difficult to take properly into account in an automatic digital system.
In this context, two main research fields can be revealed in the scientific community, which specify different approaches to video processing.
The first field is the one that minimizes any automatic semantic analysis, by mainly operating matching between some models and the video entities of interest, demanding all the possible interpretation tasks to the human beings. In this fashion, the outperforming capabilities of the automatic machinery can be easily exploited. This area, commonly referred to as *video analysis*, is more involved with pure algebraic modelling, where the contextual knowledge is present as an a-priori element, and the uncertainty minimized.
In the last years, such kind of research is rapidly increased, due to the availability of more and more powerful hardware, and to the development of effective algorithmic techniques [6]. This promoted a set of interesting applications, each one charac-

terized by different specific goals, such as video segmentation [30, 47, 47, 52, 110], video super resolution [20, 133], video impainting [152], and tracking [33, 79, 155].

The second significant research area holds when the visual data inherently become more complex, depicting activities whose semantics is not predetermined a priori as contextual knowledge. An immediate example would be whatever sequence in which humans are captured, in which the meaning of their actions and eventual interactions turns into our central interest. As written above, such task is nearer to the human ability of content management, far away from being a bijective function.
The sub-areas that try to manage such kind of video sequences are grouped together under the name of *video or scene understanding techniques* [5, 27, 132].

In the field of the video understanding, a hierarchical structure appears advantageous [159]: at the bottom level, useless information is disregarded and raw data are processed in order to extract low-level information, while in the upper level abstract reasoning is performed.
We call this structure *hierarchy of understanding*.
The proposed hierarchy of understanding is not a novel idea of course. Since the very beginning of the Computer Vision, scientists proposed hierarchical systems to deal with recognition and understanding problems [26, 121]. The main feature here regards the fact that the proposed hierarchy focuses on the *visual* aspect of information. Actually, each level of the hierarchy is designed to allow a visual interpretation of the produced output, so that possible errors or misunderstanding could immediately be evidenced before provoking errors at higher processing levels. In other words, each processing layer in the hierarchy, starting from the lower ones, is drawn trying to exploit as much as possible the low level data (e.g., numerical data in input) to gain high level information, without resorting to complex reasoning modules too early in the hierarchy. In fact, it is opinion of the author that numerical data embeds many types of high level information which can be extracted without applying complex reasoning methods; further, reasoning methods are more efficient and effective when the related input data are less noisy, discriminant and representative of the target to be achieved.

Video understanding has increased its importance only in the last years and is still growing. The visual activities to be investigated may vary from small-scale actions such as facial expressions, hand gestures, and human poses, [55, 108, 153] to large-scale activities that may involve physical interactions among locomotory objects moving around in the scene for a long period of time [67, 76].

Automatic video surveillance is one of the current video understanding subfields more studied and developed, that perfectly fits as benchmark for the novel techniques drawn [32, 67, 71, 76, 142]. For this area, the hierarchy of understanding is usually exploited as workhorse. In addition, video surveillance presents some facilitative starting conditions: the range of the available situations is constrained, semantic details are usually neglected, the typical ideal goal is to perform matching between visual data and a set of semantic primitives, and the final target is to label the activities as *normal* or *abnormal*, whose last ones should be detected and reported as soon as possible [142]. The matching between visual data and semantic

primitives takes the name of *event recognition*, and represents a promising, still open, and fertile research field [67, 76].

In the field of the automatic video surveillance, statistical reasoning represents a convenient way to deal with situations that include a high degree of uncertainty, as shown by the growing group of proposed statistical-based video surveillance frameworks [32, 65, 71, 102].
The key concept under all these video surveillance approaches is the "learning", i.e., the capability of gaining knowledge by training a particular model on the basis of the information extracted from a video sequence. In this way, the trained model can be subsequently used to generalize to other situations.

Graphical generative models [61, 72, 84] are discovered to be optimal frameworks in this sense: they offer an elegant mathematical framework to combine the observations of the activities to be modeled (bottom-up) with complex behavioral priors (top-down), in order to provide expectations about the processes and dealing properly with the uncertainty.
More specifically, the generative graphical modelling aims at formally developing models that can explain visual input as generated from a combination of hidden and observed variables, that are eventually coupled by conditional interdependencies. The possibility to make such interdependencies arbitrarily complex in dependence on the nature of the problem, gives to such framework the power, flexibility, and capability to manage, properly and efficiently, situations of various kinds.

1.1 Motivations and contributions

In this thesis, the study has been focused on various aspects of the video understanding field.

A video-analysis hierarchical framework is proposed, that exploits the most relevant characteristic of a video sequence, i.e. its representative visual information (Sec. 1.1.1). Roughly speaking, each module of such a hierarchy deals with a basic visual entity of a video sequence (the pixel, regions of pixels), producing descriptions at different levels of detail.
These descriptions can be taken as outcomes of individual analysis processes, giving insight of the sequence under a particular point of view, or used as input for higher level processing modules.
In this thesis, we propose two examples of such a hierarchy, each one providing individual contributions to the related state of the art, in both theoretical and applicative senses.

The second aspect of the video understanding field investigated in this thesis regards the possibility of adding audio analysis, as a way to improve the robustness of the overall process of understanding . This add-on can be embedded in the visual hierarchy of understanding as higher level module in a natural way.

1.1.1 A visual hierarchy of understanding

The typical video surveillance setting is a clear example of hierarchy of understanding aimed to extract knowledge from a sequence. This hierarchy is composed

by several modules, each one analyzing the input and providing as output the input for a higher level processing unit, or the result of the whole analysis.

Naturally, using a hierarchy as a logical framework causes the typical problems of bottom-up (BU) approaches, for instance, the BU propagation of errors in the hierarchy that often makes the upper levels less robust than the lower ones.
This problem can be related to the fact that, in a hierarchy of understanding, the final output is usually given after the top layer processing, while in the intermediate stages of the hierarchy the correspondent results are often not intuitively understandable, and therefore merged eventually with "invisible" and not recoverable errors. We call this problem the "masked output" issue.
An example of such "black box" characterization can be proposed by a classical face recognizer. Here, the hierarchy of understanding is usually formed by a bottom layer that extracts features, and a top layer that performs classification, using the processed data coming from the feature extraction module. Just as an example, a feature extraction module may be a Principal Component Analyzer [99], that captures the picture of a face as a multidimensional feature. The output of such a module is a set a novel features with lower dimensionality. Looking at such features (also known as *eigenfaces*) , we usually lose the capability of recognize a "good" processing, and only the top processing module (the classifier) can define whether the bottom processing unit did a good job or not.

Another problem related with the structure of the hierarchy relates its fragmentation in different modules. An over-fragmented hierarchy produces a succession of modules, prone to the "masked output" problem described above. Conversely, a too tight structure creates macro modules where the quality of the results is likely managed with difficulty.
An example of the latter type of problem comes from a video surveillance context, that we call the *Track + Cluster* system, where the aim is to track moving objects producing trajectories that can be grouped together by clustering processes. Such tracking task can be exploited by using a model-based tracking module [79], where the aim is to fit a parametric model with a moving object, disregarding the static scene. In this case, different factors should be considered, like the robustness of the model with respect to the noise (background misunderstood as foreground and viceversa, sensor noise, etc.), the capability of the model to follow the dynamics of the object to track, and others. The interdependencies among these factors are often hidden and merged in a complex structure, so that it's often difficult to figure out where and why a particular objective is failed.

These problems indicate the need of an *ideal* hierarchy of understanding, with a right number of modules, each one performing basic operations: each intermediate output has to be easily analyzed and used as feedback to improve the module which produced it.

A simple observation can ease the task of improving the quality of the hierarchy of understanding. In a video understanding framework, the object to process is a *visual* object: it means that the produced reasoning is dependent on the visual entities identified in the video sequence, and on their spatio-temporal inter-relations.

In this thesis, a hierarchical framework of understanding is proposed, in which each module outputs a type of visual information related with the kind of the basic

visual descriptor considered.
At the bottom level, we have a *low-level video understanding* module, that takes as basic entity the pixel signal, considered as an independent process (for each pixel); at the higher level, we have a *high-level video understanding* module, that considers group of pixel signals sharing the same properties as atomic entity .

Unlike other hierarchical structures, each module of the hierarchy produces a local result that can be considered as an independent top-end analysis. In other words, at each layer we gain a description of the video sequence at a different level of detail, hence, needing a different level of semantic interpretation from a human (see Fig. 1.1). Actually, at pixel level, the analysis performed is obviously

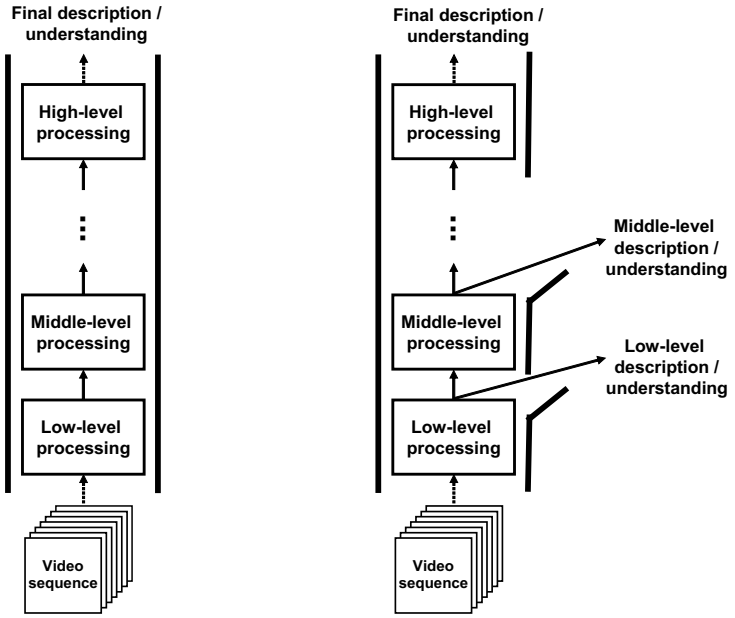


Fig. 1.1. Proposed hierarchy: on the left, a classical bottom-up hierarchy, where the output is gained after the last high-level processing; on the right, the proposed framework, in which the outputs of each layer are considered as descriptions of the sequence, at different level of detail

quite general, subject to several valid interpretations that the user may give; consequently, the user can decide how to design possible higher-level modules.
At the higher level, the possible interpretations diminish, due to the computation performed at the lower levels that pruned away useless data, and guided the user to the choice of the strategy of interpretation.
In this way, we gather a structure where the quality of the data outcoming from each processing module can be *directly* evaluated, and possible improvements can be required by only inspecting the local output of the module, without referring

exclusively to the final output.

Such idea of visual hierarchy of understanding is exploited in some parts by several video surveillance systems, in an implicit way.
In fact, the above-mentioned *Track + Cluster* video surveillance system can be reformulated in a visual hierarchy framework, by fragmenting the tracking issue in two separate, hierarchically related, modules. The bottom module performs the so-called background modelling [138, 144]. By analyzing each frame considering the pixels signals as independent processes, it aims at discriminating the expected information, namely, the background (BG), from that describing the moving entities, i.e., the foreground (FG). This kind of analysis can be considered in our visual hierarchy as a low-level on-line analysis, because it processes numerical information (each pixel signal is considered as an independent process) at each time the frame arrives.
The top module consists in a tracking method, that takes into account for the analysis produced by the bottom layer.

In this case, we have a structure of modules that produces local outputs, functional to gather a better sensibility to the quality of the obtained results. For instance, if we see that the background module is not able to perform FG detection of moving foliage, we can adopt a BG subtraction method that better focuses on this problem.

The first contribution of the thesis is to define and realize such logical organization, by developing several computational modules aimed at extracting visual knowledge from the video data, putting them together, and showing the effectiveness of the consequent results.
In particular, we developed different local modules aimed to describe video sequences at various levels of detail, namely *low-level* and *high-level*, where in the former the atomic visual entity considered is the pixel signal, and, in the latter, is the region of pixels.

In the low-level context, the aim is to extract interesting properties of each pixel signal, such as the more stable gray level values, the sequentiality with which such levels do evolve, etc.. Each pixel signal is modeled using two generative statistical models, the Time Adaptive Mixture of Gaussians (TAMG), and the (Gaussian) Hidden Markov Model (HMM). In the latter case, the contribution given to the state of the art is twofold, both theoretical and applicative.
More specifically, we devise a novel measure of *activity*, that gives an idea of how much of the signal observed represents dynamic objects.

This quantity has been used to describe the concept of *per pixel activity* occurred in a pixel location.

In the region-level context, the basic entity is the patch of pixels showing a similar behavior. The main contribution here is given by a novel similarity measure among Hidden Markov Models that permits to group models which exhibit similar "stable" temporal evolution.

As an additional contribute, a probabilistic region description coupled with a statistical module that deals with audio data is also proposed. The coupling of the audio and video analysis gives a novel hybrid region-level description, useful to

performs classification and clustering tasks of video sequences (see in the following for further details).

1.1.2 Multimodal modelling

In general, almost all of human activity recognition systems work mainly at visual level only, but other information modalities can easily be available (e.g., audio), and can be used as complementary information to discover and explain interesting "activity patterns" in a scene. Computer Vision researchers devoted their efforts towards audio-video (AV) data fusion only in the last few years, and the AV data fusion with aim of surveillance can be thought as novel sub-field of research, that increased its importance due to the presence of cheap audio and visual sensors.

In this thesis, the AV research trend has been explored; our intent was initially to import in the auditory field the low-level operation of background subtraction, which in this case is translated into an operation aimed to detect unexpected audio signal values, in an on-line fashion. The analysis was performed considering environmental sounds, without aiming at speech processing purposes.
We performed such investigation by taking into account for the poorer hardware setting in an automated surveillance system, i.e. using only one camera, equipped with a monaural microphone. This choice was due in order to detect what expressivity were reachable in such a configuration, without using microphone array, thus avoiding spatialization.

The final aim was to bring the resulting analysis module back into the visual hierarchy of understanding, in order to perform a sort of audio-video background subtraction.

The results of the research are appealing, producing the following contributes:

- A formal definition of the concept of *audio foreground*, together with a technique able to individuate audio foreground patterns, based on generative modelling.
- A formal definition of *audio-video foreground*, together with an algorithm able to couple audio foreground patterns with video data, based on the intuitive concept of *synchrony*.
- The introduction of a multi-modal feature, based on the concept of audio-video foreground, useful to perform tasks of classification and clustering of human activities in a surveillance setting.

1.2 Organization of the thesis

The thesis is organized as follows. In Part I, the needed mathematical notions will be reviewed, regarding the probabilistic generative modelling. In particular, the Hidden Markov Models will be presented as a well-known example of generative graphical model.

In Part II, the contributes given in terms of low-level description of a video sequence will be given in terms of methodological issues and practical relapses.

Part III will be focused on the region-level description of a video sequence, explaining the novel similarity measure that operates on Hidden Markov Models, and the novel formal description that considers a video sequence as an ensemble of regions of pixels related by visual and audio properties.
In the last Part IV, conclusive remarks will be reported and future perspectives envisaged, and lastly the publications derived from the thesis will be presented.

Part I

Mathemathical foundations

2

Overview of the part

In this part, we propose a personal overview of the theoretical framework analyzed in the thesis, i.e. the (graphical) statistical generative modelling.
This part is so organized: in chap. 3 we will give an introductory short overview of the statistical pattern recognition: this includes the formal definition of statistical model and the parameter estimation issue. This last point is analyzed considering the principal statistical parameter estimation paradigms: the Maximum Likelihood and the Maximum a Posteriori, and the Bayesian paradigms (Chap. 4). After that, an important ML estimation technique is introduced, the Expectation Maximization algorithm, together with an applicative example of such technique (Chap. 5). This algorithm introduces the concept of hidden information, that leads to the main idea of generative (graphical) modelling, explored in Chap. 6. Finally, a widespread generative graphical model is introduced, the Hidden Markov Model, that will be widely used and developed in the thesis (Chap. 7). Further theoretical issues, that are tightly coupled with the innovative contributes given in the thesis, will be given in the next parts.

3

Statistical pattern recognition: brief technical overview

3.1 Preliminaries

The pattern recognition research area is highly pervasive in several scientific subfields: fundamental tasks of this discipline are the classification and the automatic recognition of heterogeneous data. Different pattern recognition frameworks help in performing such tasks, as the structural and the neural paradigms [131], but the statistical paradigm appears to be the most used and investigated; actually, the *statistical pattern recognition* has grown hugely, forming *per se* an independent research field.

The statistical approach is founded on the systematic probabilistic modelling of any problem, in which the information on the present data, the dependencies among the different factors involved and the nature itself of the problem are merged in an substrate of uncertainty.

The advantages that derive from the statistical modelling approach are:

- the capability to use the statistical properties of the problem, as well as its purely mathematical and physical aspects. This leads to model events with random variables, adopting the concept of conditional independency, together with the fundamental Bayes law, and others, formally including the involved uncertainty;
- the creation of models able *to generalize* to other data

The main aim of the statistical pattern recognition is to analyze processes or sets of data using *statistical models*, that assign the data or the processes to classes of membership ω_j with $j = 1, 2, ..., J$.
Statistical models are built by using *training observations*, i.e. a set of delegates that represent the data or the process being modelled.

3.2 General form of a statistical model

A statistical model of a random process can be represented by a random variable X_i defined in a probability space, which characteristics are[1]: the cumulative dis-

[1] we consider continue random variables

tribution function (CDF) $F(X_i)$, that defines the "structure" of the model, and the parameters $\boldsymbol{\theta}_i$, that codify its density function $p_{X_i}(x_i)$, where x_i is a possible value assumed by the random variable. These parameters are useful to adapt the structure to the data modelled.

We define the generic form of a statistical model as

$$M \; = < \text{structure} \; , \; \text{parameters} > = \; < F(X_i), \boldsymbol{\theta}_i > \tag{3.1}$$

In general, the observed quantities $x_1, ..., x_n$ are modelled as values assumed by a family $\mathbf{X} = (X_1, ..., X_n)$ of random variables defined over a probability space $\mathcal{SP}$

$$\mathcal{SP} \; = \; (\Omega \, , \, \mathcal{A} \, , \, (P^{\boldsymbol{\theta}})_{\boldsymbol{\theta} \in \mathcal{S}}) \tag{3.2}$$

where:

- Ω is the set of possible events;
- $\mathcal{A}$ is a Borel σ-algebra of Ω, that codifies the operators that manipulate statistical quantities;
- $P^{\boldsymbol{\theta}}$ is a probability distribution which related random vector $\mathbf{X} = (X_1, ..., X_n)$ has continue density $p_X(\mathbf{x}|\boldsymbol{\theta})$ with $\mathbf{x} \in \Re^n$, i.e.:
  $P^{\boldsymbol{\theta}}(X \in A) = \int_A p_X(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x}$, with $A \in \Omega$
- $\mathcal{S}$ is the parameter space.

Using the notations above, we can rewrite a statistical model as

$$M =< F(\mathbf{X}), \boldsymbol{\theta} > \tag{3.3}$$

In the *generative models*, other than the observed quantities, we model the generative process that produced the observations, which is not directly visible from the data, it is *hidden*, eventually encoded in an a-priori knowledge.

3.3 Learning step as parameter estimation

The step with which a statistical model is built is called *learning*.
The learning step is defined *supervised* when the membership class label of each sample is known a priori. Such a priori knowledge can be less informative: when all the following hypotheses do hold, i.e.

- the training set is a collection of data "not labelled", i.e. when we do not know the membership classes of the samples;
- we have no information about the classes (their number is unknow, as well as their possible parametric form)

then we call the learning step as *completely unsupervised*. In this case, the learning step permits also to discover automatically the *natural* clusters of the observations. If the membership classes can be described by parameters, we call the learning step as *parameter estimation step*.

In the following sections, after a formal introduction of the statistical learning issue, we present the fundamental approaches of learning: the Maximum Likelihood (ML), the Maximum A Posteriori (MAP) and the Bayesian approaches.

4

Statistical parameter estimation: line guides

In this section, the line guides underlying the Maximum Likelihood, the Maximum A Posteriori and the Bayesian parameters estimation approaches are introduced.

Initial Hypotheses

For simplicity, we restrict to a supervised problem. Suppose that we have J classes, $w_1, ..., w_J$; our task is to separate a collection of samples, depending on their class membership, in order to obtain J data sets $\mathcal{X}_1, ..., \mathcal{X}_J$, where the samples in $\mathcal{X}_j = \{\mathbf{x}_1^{(j)}, \mathbf{x}_2^{(j)}, ..., \mathbf{x}_N^{(j)}\}$ are i.i.d following $p(\mathbf{x}^{(j)}|\omega_j, \boldsymbol{\theta}_j)$, that is a density with known parametric form, where: $\boldsymbol{\theta}_j$ is the unknown parameter vector. The value of $\boldsymbol{\theta}_j$ is unknown. In order to ease the notation, we assume that the samples in $\mathcal{X}_i$ do not provide any information on $\boldsymbol{\theta}_j$ if $i \neq j$: therefore, the parameters vectors belonging to different classes are independent. As consequence, we work on each class separately, avoiding to refer to all the individual categories and highlighting the dependency on the parameters, i.e.:

$$p(\mathbf{x}^{(j)}|\omega_j, \boldsymbol{\theta}_j) = p(\mathbf{x}^{(j)}|\boldsymbol{\theta}_j) \tag{4.1}$$

At this point, we introduce the function

$$p(\mathcal{X}_j|\boldsymbol{\theta}_j) = p(\mathbf{x}_1^{(j)}, ..., \mathbf{x}_N^{(j)}|\boldsymbol{\theta}_j) \tag{4.2}$$

that is the *likelihood* of $\boldsymbol{\theta}_j$ with respect to $\mathcal{X}_j$, that, for independency hypotheses becomes

$$p(\mathcal{X}_j|\boldsymbol{\theta}_j) = \prod_{i=1}^{N} p(\mathbf{x}_i^{(j)}|\boldsymbol{\theta}_j) \tag{4.3}$$

4.1 Maximum-Likelihood and Maximum a Posteriori estimation

Starting from the definitions of Sec.4, as in [53], we suppose to have a set of i.i.d observations, $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, whose parametric structure is known, i.e. they are generated by $p(\mathbf{x}|\boldsymbol{\theta})$, that is known only in parametric form and dependent on the

unknown parameter vector $\boldsymbol{\theta}$. The task is to use the data in $\mathcal{X}$ in order to estimate $\boldsymbol{\theta}$. We can write:

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}|\mathcal{X}) \tag{4.4}$$

$\mathcal{L}(\boldsymbol{\theta}|\mathcal{X})$ is the *likelihood* function of $\boldsymbol{\theta}$ *with respect to* the data $\mathcal{X}$. The ML estimation of $\boldsymbol{\theta}$ is the value $\hat{\boldsymbol{\theta}}$ which maximizes $\mathcal{L}(\boldsymbol{\theta}|\mathcal{X})$, i.e. the one that better models the data. In formulae

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\theta}|\mathcal{X}) \tag{4.5}$$

To ease the computation, it is simpler to work with the logarithm, obtaining the *log-likelihood*, rather than the likelihood itself: being the logarithm a growing monotonic function, maximizing the log-likelihood with respect to $\boldsymbol{\theta}$ is equivalent to maximize $\mathcal{L}(\boldsymbol{\theta}|\mathcal{X})$. If the likelihood is differentiable with respect to $\boldsymbol{\theta}$, then the value of $\hat{\boldsymbol{\theta}}$ can be estimated using standard differential calculus.

Whenever novel training observations appear, the whole process of estimation has to be repeated, because the classification is performed on the actual value of the parameters. This is the main difference with respect to the Bayesian approach, as we see in the following.

The ML estimation spans the entire domain $\mathcal{S}$ related to $\boldsymbol{\theta}$. The drawback is that each point of $\mathcal{S}$ is equally valid, with respect to the data likelihood. A standard way to weight the points of $\mathcal{S}$ in dependence with an a priori knowledge on $\boldsymbol{\theta}$ holds by using the Maximum A Posteriori (MAP) paradigm: basically, we look for the maximum of $l(\boldsymbol{\theta})p(\boldsymbol{\theta})$, where $p(\boldsymbol{\theta})$ represents the a priori probability density on the values of $\boldsymbol{\theta}$.

If the likelihood is not directly differentiable, we can apply other forms of optimization. In the section 5 we see the *Expectation-Maximization* algorithm, that operates on a simplified version of the likelihood.

4.2 Bayesian estimation

The fundamental aspect of this approach is that the parameter $\boldsymbol{\theta}$ is considered as a random variable governed by a certain distribution, i.e. an a priori density $p(\boldsymbol{\theta})$. During the classification of a novel sample $\hat{\mathbf{x}}$, this density is translated in an a posteriori density, by observing the training set and applying the Bayes formula.

Let $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of i.i.d. observations, generated by $p(\mathbf{x}|\boldsymbol{\theta})$, and $p(\boldsymbol{\theta})$ the a priori density. Let $\hat{\mathbf{x}}$ be a novel sample, and let us define the density for $\hat{\mathbf{x}}$ as the integral over the whole parameter space, i.e.

$$p\left(\hat{\mathbf{x}}|\mathcal{X}\right) = \int_{\boldsymbol{\theta}} p\left(\hat{\mathbf{x}}\,,\,\boldsymbol{\theta}\,|\mathcal{X}\right) d\boldsymbol{\theta} \tag{4.6}$$

By using the definition of conditional probability, we obtain the following expression for the joint density:

$$p\left(\hat{\mathbf{x}}\,,\,\boldsymbol{\theta}\,|\mathcal{X}\right) = p\left(\hat{\mathbf{x}}|\,\boldsymbol{\theta}\,,\,\mathcal{X}\right) p\left(\,\boldsymbol{\theta}\,|\mathcal{X}\right) \tag{4.7}$$

The first factor of the right term is independent on $\mathcal{X}$ because, for hypothesis, such a function is known in parametric form, and once known the value of $\boldsymbol{\theta}$ it becomes completely specified. Therefore, we can rewrite Eq. 4.6, considering Eq. 4.7, as

$$p\left(\hat{\mathbf{x}}|\mathcal{X}\right) = \int_{\boldsymbol{\theta}} p\left(\hat{\mathbf{x}}|\,\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\,|\mathcal{X}\right) d\boldsymbol{\theta} \tag{4.8}$$

Now, rather than searching for a point estimation of $\boldsymbol{\theta}$, the Bayesian approach computes $p\left(\,\boldsymbol{\theta}\,|\mathcal{X}\right)$, i.e. the posterior density of $\boldsymbol{\theta}$, by using the Bayes law

$$p\left(\,\boldsymbol{\theta}\,|\mathcal{X}\right) = \frac{p\left(\mathcal{X}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right)}{p\left(\mathcal{X}\right)} \tag{4.9}$$

that expresses a goodness degree over the value of $\boldsymbol{\theta}$ given the observations $\mathcal{X}$. The function $p\left(\mathcal{X}|\boldsymbol{\theta}\right)$ is the likelihood and $p\left(\mathcal{X}\right)$ the evidence, i.e. the normalization factor that assures that $\int p\left(\,\boldsymbol{\theta}\,|\mathcal{X}\right) d\boldsymbol{\theta} = 1$.
Assuming that the data are i.i.d., we can write the likelihood as

$$p\left(\mathcal{X}|\boldsymbol{\theta}\right) = \prod_{i=1}^{N} p\left(\mathbf{x}_i|\,\boldsymbol{\theta}\right) \tag{4.10}$$

and the evidence as

$$p\left(\mathcal{X}\right) = \int_{\boldsymbol{\theta}'} p\left(\boldsymbol{\theta}'\right) \prod_{i=1}^{N} p\left(\mathbf{x}_i|\,\boldsymbol{\theta}'\right) d\boldsymbol{\theta}' \tag{4.11}$$

The posterior for $\boldsymbol{\theta}$ becomes

$$p\left(\,\boldsymbol{\theta}\,|\mathcal{X}\right) = \frac{p\left(\,\boldsymbol{\theta}\,\right) \prod_{i=1}^{N} p\left(\mathbf{x}_i|\,\boldsymbol{\theta}\right)}{\int_{\boldsymbol{\theta}'} p\left(\boldsymbol{\theta}'\right) \prod_{i=1}^{N} p\left(\mathbf{x}_i|\,\boldsymbol{\theta}'\right) d\boldsymbol{\theta}'} \tag{4.12}$$

that works as "weight" of the value of $\boldsymbol{\theta}$ used in the conditional density over the new data, as expressed in Eq. 4.8. Being present the integral, the probability to have the new sample given $\mathcal{X}$ does not takes into account for a particular value of $\boldsymbol{\theta}$, but for a "mean" of values instead, weighted by $p\left(\,\boldsymbol{\theta}\,|\mathcal{X}\right)$.
Often the calculus of the integrals in Eqs. 4.8,4.11 is complex, and, in general, it is easy to compute when the likelihood (4.10) and $p(\boldsymbol{\theta})$ are function of the same nature; in this case, we talk about *conjugate* a priori density over $p(\boldsymbol{\theta})$.

4.3 Remarks

In the following table, we compare the three basis techniques of parameter estimation, comparing the predictive density resulting from each one of them. Further details can be readed in [101] and [14].

method	**estimated quantity**	**prediction**
Maximum-Likelihood	$p(\mathcal{X} \mid \theta)$	$p(\hat{x} \mid \theta_{ML})$
Maximum A Posteriori	$p(\theta \mid \mathcal{X})$	$p(\hat{x} \mid \theta_{MAP})$
Bayesian Learning	$p(\theta \mid \mathcal{X})$	$p(\hat{x} \mid \mathcal{X})$

Table 4.1. Different estimation techniques

5

Expectation-Maximization algorithm

Different limitations can be present in the real observation (or measure) processes: for instance, loss of data or scarce observations amount. Moreover, in the estimations problems, often the likelihood function is difficult to differentiate in an analytic form, especially when we are dealing with mixture models.

The EM algorithm, introduced in [50, 157], further proposed in [19, 106, 107], and in general heavily used in the Computer Vision community, is a method to obtain ML parameter estimates that deals with the above situations. Actually:

- it permits to calculate ML estimations starting from an *incomplete* set of samples;
- it considers the presence of *hidden variables* strongly connected with the observations, that make the estimation problem tractable. The existence of these hidden variables (not the values assumed by) derives from an a priori knowledge of the data observed.

5.1 Standard EM

We assume to have a set of i.i.d. data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, modelled by the likelihood

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}|\mathcal{X})$$

where $\boldsymbol{\theta}$ is the unknown parameter vector.

The ingredients of the EM algorithm are:

- an *incomplete* (and observable) data set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$;
- a set of *hidden variables* $\mathcal{Y}$, whose values can be estimated by $\mathcal{X}$ and $\boldsymbol{\theta}$.

The incomplete data set and the hidden variables form the *complete data set* $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$, that leads to define the complete joint density

$$p(\mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta}) \tag{5.1}$$

in a way that quantifies the relations among observable data ($\mathcal{X}$), hidden data ($\mathcal{Y}$) and unknown parameters ($\boldsymbol{\theta}$) in probabilistic terms.

With this novel density, we can define a new likelihood function

$$\mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{Z}) = \mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y}) = p(\mathcal{X}, \mathcal{Y} \,|\, \boldsymbol{\theta}) \tag{5.2}$$

which name is *complete data likelihood*. The quantity $\mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{X})$ becomes the *incomplete data likelihood*.

The set $\mathcal{Y}$ represents a hidden information, consequently unknown; therefore, it is modelled as a random quantity, that can be estimated from $\mathcal{X}$ and $\boldsymbol{\theta}$, and described by the density $p(\mathbf{y} \,|\, \mathbf{x}, \boldsymbol{\theta})$ in (5.1).

If we choose a particular value for $\boldsymbol{\theta}$, the likelihood (5.2) becomes a function dependent only on $\mathcal{Y}$, i.e.

$$\mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y}) = \mathcal{F}_{\mathcal{X},\boldsymbol{\theta}}(\mathcal{Y}) \tag{5.3}$$

being $\mathcal{X}$ a known set.

At this point, the EM algorithm proceeds iteratively with two steps: expectation (E) step and maximization (M) step.

5.1.1 E-step

The likelihood $\mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y}) = p(\mathcal{X}, \mathcal{Y} \,|\, \boldsymbol{\theta})$ tell us that the information needed to know $\boldsymbol{\theta}$ is enclosed in $\mathcal{X}$ and $\mathcal{Y}$. The E step is useful to eliminate the dependency of $\boldsymbol{\theta}$ on $\mathcal{Y}$ and therefore to use only $\mathcal{X}$ (that we know) in order to estimate $\boldsymbol{\theta}$.

In this fashion, the uncertainty over $\mathcal{Y}$ does not transmit over $\boldsymbol{\theta}$, and, at the same time, $\mathcal{Y}$ can be used to simplify the estimation computation.

Formally, the elimination of the dependency on $\mathcal{Y}$ is performed by computing the expected value of the log-likelihood $\ln p(\mathcal{X}, \mathcal{Y} \,|\, \boldsymbol{\theta})$ with respect to $\mathcal{Y}$, given $\mathcal{X}$ and $\boldsymbol{\theta}^{(i-1)}$, where this last quantity is the actual parameter estimate (i is the current iteration index).

The result is the marginalization over $\mathcal{Y}$. The expected value defines the function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$, i.e.

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) = E\left[\ln p(\mathcal{X}, \mathcal{Y} \,|\, \boldsymbol{\theta}) | \, \mathcal{X}, \boldsymbol{\theta}^{(i-1)}\right] \tag{5.4}$$

where $\boldsymbol{\theta}^{(i-1)}$ is a constant (being the current estimation of the parameter), while $\boldsymbol{\theta}$ represent the new parameter to estimate in order to increase Q.
Starting from the definition of conditional expected value

$$E[f(Y)|X = x] = \int_y f(y) p_{Y|X}(y|x) dy \tag{5.5}$$

the right term of Eq. 5.4 becomes

$$E\left[\ln p(\mathcal{X}, \mathcal{Y} \,|\, \boldsymbol{\theta}) | \, \mathcal{X}, \boldsymbol{\theta}^{(i-1)}\right] = \int_{\mathbf{y} \in \Upsilon} \ln p(\mathcal{X}, \mathbf{y} \,|\, \boldsymbol{\theta}) p(\mathbf{y} \,|\, \mathcal{X}, \boldsymbol{\theta}^{(i-1)}) \, d\mathbf{y} \tag{5.6}$$

where the density $p(\mathbf{y} \,|\, \mathcal{X}, \boldsymbol{\theta}^{(i-1)})$ is the distribution of $\mathcal{Y}$, which is dependent on the observed data $\mathcal{X}$ and on the current estimation of the parameters; Υ is the space where $\mathcal{Y}$ lives.

The function to maximize becomes then

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) = \int_{\mathbf{y}\in\Upsilon} \ln p(\mathcal{X}, \mathbf{y} \mid \boldsymbol{\theta}) p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}^{(i-1)})\, d\mathbf{y} \tag{5.7}$$

this is a deterministic function with respect to $\boldsymbol{\theta}$. Using the density of Eq. 5.1, the function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$ can be factorized in a summation, simpler to deal with, i.e.

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) &= \int_{\mathbf{y}\in\Upsilon} \ln [p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\mathcal{X} \mid \boldsymbol{\theta})]\, p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}^{(i-1)})\, d\mathbf{y} \\ &= \int_{\mathbf{y}\in\Upsilon} \ln p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}^{(i-1)}) d\mathbf{y} \\ &+ \int_{\mathbf{y}\in\Upsilon} \ln p(\mathcal{X} \mid \boldsymbol{\theta}) p(\mathbf{y} \mid \mathcal{X}, \boldsymbol{\theta}^{(i-1)}) d\mathbf{y}. \end{aligned} \tag{5.8}$$

5.1.2 M-step

The M step produces a ML estimation of $\boldsymbol{\theta}$, obtained through maximization of the function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$, i.e.

$$\boldsymbol{\theta}^{(i)} = \underset{\boldsymbol{\theta}}{\text{argmax}}\ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) \tag{5.9}$$

At each iteration, the increasing of the complete data log-likelihood is guaranteed as well as the convergency of the algorithm to a *local* maxima of the likelihood function [50], [156], [106].

A modified M-step occurs when, instead to maximize directly $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$, we are looking for the value of $\boldsymbol{\theta}^{(i)}$ such that

$$Q(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i-1)}) > Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$$

In this case the algorithm is called *Generalized EM* (GEM), which convergence is also guaranteed [19].

In the following we show a mixture of Gaussians example.

5.2 Example: mixture of Gaussians

Given a set of i.i.d. samples $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, with $\mathbf{x}_i \in \Re^d$, we assume the following hypotheses:

- the samples belong exactly to C classes ω_j, $j = 1, 2, ..., C$;
- the a priori $p(\omega_j)$, where $\sum_{j=1}^{C} p(\omega_j) = 1$, are not given;
- the parametric form of the densities $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$ (with parameters $\boldsymbol{\theta}_j$) is known, that is, a multivariate d-dimensional Gaussian with mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$, leading to: $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j) \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$;
- the parameter vector is $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ for $j = 1, 2, ..., C$, and its values are unknown;

- each sample $\mathbf{x}$ is obtained by selecting a class ω_j with probability density $p(\omega_j)$, considering then the density $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$. Therefore, the likelihood $p(\mathbf{x}|\boldsymbol{\Theta})$ is modelled by a mixture of Gaussians

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{j=1}^{C} p(\omega_j) p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j) \tag{5.10}$$

  where $\boldsymbol{\Theta} = (p(\omega_1), p(\omega_2), ..., p(\omega_C), \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_C)$ is the vector of the parameters that needs to be estimated.

To ease the notation:

- $p(\omega_j) = \alpha_j$ per $j = 1, 2, ..., C$;
- $\boldsymbol{\Theta} = (\alpha_1, \alpha_2, ..., \alpha_C, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_C)$.
- $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j) = p(\mathbf{x}|\boldsymbol{\theta}_j)$
- the mixture of Gaussians becomes

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{j=1}^{C} \alpha_j p_j(\mathbf{x}|\boldsymbol{\theta}_j) \tag{5.11}$$

  The incomplete log-likelihood is

$$\mathcal{L}(\boldsymbol{\Theta} \,|\, \mathcal{X}) = \ln \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\Theta}) = \sum_{i=1}^{N} \ln \left(\sum_{j=1}^{C} \alpha_j p_j(\mathbf{x}_i|\boldsymbol{\theta}_j) \right) \tag{5.12}$$

that is difficult to optimize with respect to $\boldsymbol{\Theta}$ because we have a logarithm of a summation.

At this point, we assume that $\mathcal{X}$ is an *incomplete* set of samples, and $\mathcal{Y}$ a set of hidden data

$$\mathcal{Y} = \{y_i\}_{i=1}^{N} \qquad \text{where} \qquad \forall i \; y_i \in \{1, 2, ..., C\} \tag{5.13}$$

such that

$$y_i = k \quad \Longrightarrow \quad \mathbf{x}_i \text{ is generated by } \omega_k \tag{5.14}$$

The value of y_i tells us which component of the mixture of Gaussians generated the sample x_i. The knowledge of such values permit us to simplify the likelihood expression.

If we knew the values of $\mathcal{Y}$, the complete data likelihood becomes

$$\mathcal{L}(\boldsymbol{\Theta} \,|\, \mathcal{X}, \mathcal{Y}) = \ln p(\mathcal{X}, \mathcal{Y} \,|\, \boldsymbol{\Theta}) = \sum_{i=1}^{N} \ln(p(\mathbf{x}_i|y_i) p(y_i)) \tag{5.15}$$

By indexing the components of the mixture with the y_i and setting

$$p(y_i) = \alpha_{y_i} \tag{5.16}$$

we obtain

$$\mathcal{L}(\boldsymbol{\Theta} \mid \mathcal{X}, \mathcal{Y}) = \sum_{i=1}^{N} \ln(\alpha_{y_i} p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i})) \tag{5.17}$$

that can be optimized with respect to $\boldsymbol{\Theta}$ by using simple techniques [53].

The problem is that we do not know the values of $\mathcal{Y}$: the solution is to assume that $\mathcal{Y}$ is a random vector, described by a density function.

E-step: computation of the conditional expectation Q

As first step, we derive an expression for the distribution of $\mathcal{Y}$: to do this, we fix an initial value of $\boldsymbol{\Theta}$, i.e.

$$\boldsymbol{\Theta}^{(0)} = (\alpha_1^0, \alpha_2^0, ..., \alpha_C^0, \boldsymbol{\theta}_1^0, \boldsymbol{\theta}_2^0, ..., \boldsymbol{\theta}_C^0) \tag{5.18}$$

We can calculate the value of the density $p_j(\mathbf{x}_i|\boldsymbol{\theta}_j^0)\ \forall i, j$. Due to the Eq. 5.16 and applying the Bayes law, we obtain

$$p(y_i|\mathbf{x}_i, \boldsymbol{\Theta}^{(0)}) = \frac{\alpha_{y_i}^0 p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i}^0)}{p(\mathbf{x}_i|\boldsymbol{\Theta}^{(0)})} = \frac{\alpha_{y_i}^0 p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i}^0)}{\sum_{k=1}^{C} \alpha_k^0 p_k(\mathbf{x}_i|\boldsymbol{\theta}_k^0)} \tag{5.19}$$

The density function of $\mathcal{Y}$ results to be

$$p(\mathbf{y}|\mathcal{X}, \boldsymbol{\Theta}^{(0)}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \boldsymbol{\Theta}^{(0)}) \tag{5.20}$$

where $\mathbf{y} = (y_1, y_2, ..., y_N)$ is an instance of the hidden variable.

Now we are able to compute the conditional expectation $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(0)})$:

$$\begin{aligned}
Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(0)}) &= \sum_{\mathbf{y} \in \Upsilon} \ln(\mathcal{L}(\boldsymbol{\Theta}|\mathcal{X}, \mathbf{y})) p(\mathbf{y}|\mathcal{X}, \boldsymbol{\Theta}^{(0)}) \\
&= \sum_{\mathbf{y} \in \Upsilon} \sum_{i=1}^{N} \ln(\alpha_{y_i} p_{y_i}(\mathbf{x}_i|\theta_{y_i})) \prod_{j=1}^{N} p(y_j|\mathbf{x}_j, \boldsymbol{\Theta}^{(0)}) \\
&= \sum_{y_1=1}^{C} \sum_{y_2=1}^{C} \cdots \sum_{y_N=1}^{C} \sum_{i=1}^{N} \ln(\alpha_{y_i} p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i})) \prod_{j=1}^{N} p(y_j|\mathbf{x}_j, \boldsymbol{\Theta}^{(0)}) \\
&= \sum_{y_1=1}^{C} \sum_{y_2=1}^{C} \cdots \sum_{y_N=1}^{C} \sum_{i=1}^{N} \sum_{\ell=1}^{C} \delta_{\ell,y_i} \ln(\alpha_\ell p_\ell(\mathbf{x}_i|\boldsymbol{\theta}_\ell)) \prod_{j=1}^{N} p(y_j|\mathbf{x}_j, \boldsymbol{\Theta}^{(0)}) \\
&= \sum_{\ell=1}^{C} \sum_{i=1}^{N} \ln(\alpha_\ell p_\ell(\mathbf{x}_i|\boldsymbol{\theta}_\ell)) \sum_{y_1=1}^{C} \sum_{y_2=1}^{C} \cdots \sum_{y_N=1}^{C} \delta_{\ell,y_i} \prod_{j=1}^{N} p(y_j|\mathbf{x}_j, \boldsymbol{\Theta}^{(0)}) \\
&= \sum_{\ell=1}^{C} \sum_{i=1}^{N} \ln(\alpha_\ell) p(\ell|\mathbf{x}_i, \boldsymbol{\Theta}^{(0)}) + \sum_{\ell=1}^{C} \sum_{i=1}^{N} \ln(p_\ell(\mathbf{x}_i|\boldsymbol{\theta}_\ell)) p(\ell|\mathbf{x}_i, \boldsymbol{\Theta}^{(0)})
\end{aligned} \tag{5.21}$$

where δ_{ℓ,y_i} is Dirac delta function, i.e. $\delta_{\ell,y_i} = 1$ if $\ell = y_i$, 0 otherwise. As we can note, Eq. 5.21 is formed by two elements: the former is dependent from the

α_ℓs', the latter from all the $\boldsymbol{\theta}_\ell$s'. The using of the index ℓ instead of j is due to some algebraic passages explained in [19].
We therefore obtain a function that is easy to optimize with respect to $\boldsymbol{\Theta}$: actually we can optimize, in independent fashion, with respect to α_ℓ, by considering only the first term, and then with respect to $\boldsymbol{\theta}_\ell$.

M-step: α_ℓ estimation

In order to find the estimation of α_ℓ we introduce the Lagrange factor [64] λ with $\sum_\ell \alpha_\ell = 1$ and we solve the equation

$$\frac{\partial}{\partial \alpha_\ell}\left[\sum_{\ell=1}^{C}\sum_{i=1}^{N}\ln(\alpha_\ell)p(\ell|\mathbf{x}_i,\boldsymbol{\Theta}^{(0)}) + \lambda\left(\sum_\ell \alpha_\ell - 1\right)\right] = 0 \tag{5.22}$$

which becomes, by partially differentiating

$$\sum_{i=1}^{N}\frac{1}{\alpha_\ell}p(\ell|\mathbf{x}_i,\boldsymbol{\Theta}^{(0)}) + \lambda = 0 \tag{5.23}$$

Considering all the feasible α_ℓ we obtain the value of λ, $\lambda = -N$, with which we can obtain the expression for $\ell = 1, 2, ..., C$

$$\alpha_\ell = \frac{1}{N}\sum_{i=1}^{N}p(\ell|\mathbf{x}_i,\boldsymbol{\Theta}^{(0)}) \tag{5.24}$$

M-step: $\boldsymbol{\theta}_\ell$ estimation

We have

$$\boldsymbol{\theta}_\ell = (\mu_\ell, \boldsymbol{\Sigma}_\ell)$$

and

$$p(\mathbf{x}|\boldsymbol{\theta}_\ell) \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)$$

that in explicit form is

$$p(\mathbf{x}|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_\ell)^T\boldsymbol{\Sigma}_\ell^{-1}(\mathbf{x}-\boldsymbol{\mu}_\ell)\right] \tag{5.25}$$

The estimations of $\boldsymbol{\mu}_\ell$ and $\boldsymbol{\Sigma}_\ell$ are found by starting from the second factor of the Eq. 5.21, differentiating with respect to $\boldsymbol{\mu}_\ell$ and $\boldsymbol{\Sigma}_\ell$. After some algebraic manipulations [19] we obtain, for $\ell = 1, 2, ..., C$

$$\boldsymbol{\mu}_\ell = \frac{\sum_{i=1}^{N}\mathbf{x}_i p(\ell|\mathbf{x}_i,\boldsymbol{\Theta}^{(0)})}{\sum_{i=1}^{N}p(\ell|\mathbf{x}_i,\boldsymbol{\Theta}^{(0)})} \tag{5.26}$$

and

$$\boldsymbol{\Sigma}_\ell = \frac{\sum_{i=1}^{N}p(\ell|\mathbf{x}_i,\boldsymbol{\Theta}^{(0)})(\mathbf{x}_i-\boldsymbol{\mu}_\ell)(\mathbf{x}_i-\boldsymbol{\mu}_\ell)^T}{\sum_{i=1}^{N}p(\ell|\mathbf{x}_i,\boldsymbol{\Theta}^{(0)})} \tag{5.27}$$

The estimations for the parameters α_ℓ, $\boldsymbol{\mu}_\ell$ e $\boldsymbol{\Sigma}_\ell$ can be used in the next iteration of the EM, forming a vector $\boldsymbol{\Theta}^{(1)}$ which will substitute $\boldsymbol{\Theta}^{(0)}$, and so on until a convergence criteria is met; for the convergence issue, see [157].

6

Generative models

In the EM estimation performed in the mixture of Gaussians example, we discover the main aim of the generative modelling, that is the explanation of an observation set as *result of an invisible process* that generated it. The nature of such process, or cause, usually is expressed using hidden variables, which represent different factors concurring in the generation process; the relationship among these factors is mostly described as a set of conditional dependencies, described by conditional densities.
In the previous section the hidden variables explained the process of generation of the observations, or the *generative process* of the observations, describing a hidden process that selects one of the mixture's component, which generates the particular observed data.

Under this point of view, the mixture of Gaussians can be considered as an example of generative statistical model, or simply a generative model.
The chapter is organized as follows: in Sec. 6.1 we indicate a pseudo-algorithm to build a generative model; Sec. 6.2 analyzes the most significant step involved in the building of a generative model, i.e. the learning step and, finally, in Sec. 6.3 the graphical methodology to express a generative model is detailed.

6.1 Building a generative model

Given a set of observations considered as a set of instances of i.i.d. random variables V, the steps that lead to the related generative model are:

1. *Intuitive definition*: definition of the *intuitive hidden causes* that generated the *observed variables*. The hidden causes, eventually, can be related by conditional dependency.
2. *Statistical definition*: this step is needed to manage the uncertainty, both in the definition of the "right" hidden variables and in the observations themselves, eventually corrupted by noise. It consists in coding each hidden cause as hidden random variable and describing the whole generation process with a joint distribution. This distribution should be factorized, taking into account for the relations among the causes defined in the previous step. Conditional dependencies lead to conditional densities, independencies bring to independent

densities and so on. In this step, the definition of conditional independency needs to be included, in order to simplify the factorization of the joint distribution, by canceling some useless conditioning (see later)
3. *Parametrization*: this step consists in the parametrization of all the densities involved in the factorization. Even the parameters should be thought as hidden quantities and, together with the values assumed by the hidden variables, forming the set H. Therefore, the joint distribution can be indicated as $P(H, V)$.
4. *Learning*: at this point, the most important step consists in the learning of the hidden quantities using the observations, i.e. choosing a possible instance of values for H that maximize the posteriori distribution $P(H|V)$.

The step 1) of the building process of a generative model needs to be highlighted: in facts, the a priori knowledge that tell us the causes that generated the observations, as long as their parametric form and dependency, is not always present in a modelling process. This problem leads to a difficult and open research field, the model selection, aimed at selecting the best model given a set of observations.

The step 4) is the core of the generative statistical models, deeply investigated in the literature. An exhaustive analysis of all the existent learning strategies goes beyond the scope of this thesis; for details, see [100].
Instead, in the following we explain the learning step as a problem of minimizing distances among functions, recently presented in [61].

6.2 The learning step

The intuitive way to estimate $P(H|V)$ starts supposing the joint distribution of the generative process as known, i.e. $P(H, V)$, from which we can estimate

$$p(H|V) = \frac{p(V, H)}{\int_H p(V, H)} \tag{6.1}$$

where $\int_H p(V, H) = p(V)$ is a normalization form.

When $p(H|V)$ is a tractable and computable (in a efficient way) function, then an immediate differentiation can be applied. In such case the inference is said *exact*.

Anyway, in most of the cases, it is impossible to find the analytical maximization, as we saw for the mixture of Gaussians example. This occurs often when the number of hidden variables is high, and each one of them can assumes a wide range of values [61]. In such cases, it is needed to assume an *approximation* of $p(H|V)$ in the process of estimation.

6.2.1 Approximation $p(H|V)$

Assuming $p(H|V)$ as intractable, the idea is to search for a simpler but similar function $Q(H)$, subjected to

$$\int_H Q(H) = 1 \tag{6.2}$$

The similarity is expressed as a distance in the space of the functions to minimize. Instead of a distance, in literature is mostly used the divergence of *Kullback-Leibler* $D(Q,P)$ [35]

$$D(Q,P) = \int_H Q(H) \ln \frac{Q(H)}{p(H|V)} \tag{6.3}$$

where the following properties are satisfied:

$$D(Q,P) \geq 0 \tag{6.4}$$

$$D(Q,P) = 0 \qquad \Longleftrightarrow \qquad Q(H) = p(H|V) \tag{6.5}$$

Evidently, the function $D(Q,P)$ can not be minimized, being present $p(H|V)$. Instead of $D(Q,P)$ the solution consists in evaluating another divergence, known as *Helmholtz Free Energy* $F(Q,P)$, without constraining the search space of Q.

The divergence $F(Q,P)$ is a functional that can be derived from $D(Q,P)$ by subtracting it the term

$$\log p(V) = \int_H Q(H) \ln p(V) \tag{6.6}$$

that is independent from H, thus leaving unchanged the search space of $Q(H)$. Explicitly, $F(Q,P)$ is

$$\begin{aligned} F(Q,P) &= D(Q,P) - \ln p(V) \qquad (6.7)\\ &= \int_H Q(H) \ln \frac{Q(H)}{p(H|V)} - \int_H Q(H) \ln p(V) \\ &= \int_H Q(H) \ln \frac{Q(H)}{p(H|V)p(V)} \\ &= \int_H Q(H) \ln \frac{Q(H)}{p(V,H)} \qquad (6.8) \end{aligned}$$

that can be intended as the KL divergence between $Q(H)$ and $p(V,H)$, that replaces the untractable $p(H|V)$. The add-on of this form is that the joint $p(H,V)$ is known, therefore $F(Q,P)$ is computable.

Another way to derive $F(Q,P)$ consists in using the Jensen's inequality; Jensen's inequality states that a concave function of a convex combination of points in a vector space is greater than or equal to the convex combination of the concave function applied to the points. To bound the log-probability of the visible variables $\ln P(V) = \ln(\int_H P(H,V))$, we use the arbitrary distribution $Q(H)$ (a set of convex weights, summing them to 1) to obtain a convex combination inside the concave logarithm function:

$$\begin{aligned} \ln P(V) &= \ln \left(\int_H P(H,V) \right) = \ln \left(\int_H Q(H) \frac{P(H,V)}{Q(H)} \right) \\ &\geq \int_H Q(H) \ln \left(\frac{P(H,V)}{Q(H)} \right) = -F(Q,P) \qquad (6.9) \end{aligned}$$

At this point, the trick of all the learning algorithms stand on how to define $Q(H)$ in order to make the inference plausible. The loosest constraint for $Q(H)$,

($\int_H Q(H) = 1$), means to apply simple functional differentiating rules in the functional of Eq. 6.8, obtaining $Q(H) = P(H|V)$.

The task is to find simplificative constraints for $Q(H)$.

6.2.2 An alternative way to consider the EM algorithm

The EM algorithm can be explained in this framework by defining

$$Q(H) = \delta(H^{(\theta)} - \hat{H}^{(\theta)}) \; Q_h(H^{(1)}, H^{(2)}, ..., H^{(T)}) \tag{6.10}$$

where T is the number of the visible variables $V^{(1)}, ..., V^{(T)}$, with $\{H^{(t)}\}$, $t = 1, \ldots, T$ the hidden variables, one for each visible variable, and $H^{(\theta)}$ the parameters, that we assume shared by all the variables.

By assuming the visible variables $\{H^{(t)}\}$ as i.i.d., and that $H^{(t)}$ is conditionally related to the respective $V^{(t)}$, we can rewrite the right term of Eq. 6.10 as

$$Q(H) = \delta(H^{(\theta)} - \hat{H}^{(\theta)}) \; \prod_{t=1}^{T} Q_h(H^{(t)}) \tag{6.11}$$

with $\hat{H}^{(\theta)}$ the set of "best" parameters, in a likelihood sense. Substituting in Eq. 6.8 the following joint distribution

$$p(V, H) = p(H^{(\theta)}) \prod_{t=1}^{T} p(H^{(t)}, V^{(t)}|H^{(\theta)})$$

and considering $Q(H)$ as defined in Eq. 6.11, the Helmholtz free energy becomes

$$F(Q, P) = -\log p(\hat{H}^{(\theta)}) + \sum_{t=1}^{T} \int_{H^{(t)}} Q_h(H^{(t)}) \log \frac{Q_h(H^{(t)})}{p(H^{(t)}, V^{(t)}|\hat{H}^{(\theta)})} \tag{6.12}$$

By interpreting this function as complete data log-likelihood, the EM algorithm can be rewritten as:

- **Initialization**: we assume

  $$Q(H) = \delta(H^{(\theta)} - \hat{H}^{(\theta)}) \; \prod_{t=1}^{T} Q_h(H^{(t)})$$

  and we guess an initial "best" parametrization, $\hat{H}^{(\theta)}$, assuming further the parameters as independent from each other.
- **E-step**: minimize $F(Q, P)$ with respect to $Q_h(H^{(t)})$, with only constraints that $\int_{H^{(t)}} Q_h(H^{(t)}) = 1$, using the guessed $\hat{H}^{(\theta)}$. The minimum is achievable (by functional derivative calculus) for

  $$Q_h(H^{(t)}) = p(H^{(t)}|V^{(t)}, \hat{H}^{(\theta)}) \tag{6.13}$$

  for $t = 1, 2, ..., T$.

- **M-step** : minimize $F(Q,P)$ with respect to $\hat{H}^{(\theta)}$, by using the form of $Q(H^{(t)})$ found in the E-step, and computing, for each $\hat{h}^{(\theta)} \in \hat{H}^{(\theta)}$, a **Maximum-Likelihood estimation**, by analytically solving $\partial F(Q,P)/\partial \hat{h}^{(\theta)} = 0$, i.e.

$$-\frac{\partial}{\partial \hat{h}^{(\theta)}} \log p(\hat{h}^{(\theta)}) - \sum_{t=1}^{T} \left(\int_{H^{(t)}} Q_h(H^{(t)}) \frac{\partial}{\partial \hat{h}^{(\theta)}} \log p(H^{(t)}, V^{(t)} | \hat{H}^{(\theta)}) \right) = 0 \tag{6.14}$$

  ($Q_h(H^{(t)})$ does not depend from any hidden parameters).
- repeat E-step and M-step until convergence, using the maximization found at the n-th step as guess in the $n+1$-th step.

6.3 Generative graphical model

Usually, the generative models are better explained and managed using graphs, where the nodes represent the variables and the edges the conditional dependencies.

Graphical models [36, 63, 84, 100, 107, 113, 114, 127] use graphs to represent and manipulate joint probability distributions, and are useful instruments aimed at the construction of efficient generative models.

A graphical model has both a structural component, encoded by the pattern of edges in the graph, and a parametric component, encoded by a numerical "potential" associated with sets of edges in the graph. The relationship between these components underlies the computational machinery associated with graphical models. In particular, general inference methods allow statistical quantities (such as likelihoods or conditional probabilities) and information theoretic quantities (such as mutual information and conditional entropies) to be computed efficiently.
Learning algorithms are derived from these inference algorithms and allow parameters and structures to be estimated from data.

In the following, we provide the definition of Bayes Network, or simply Bayes Net.

Definition 1 *A Bayes Network for the set of random variables $S = \{s_1, ..., s_T\}$ is an acyclic directed graph* [1]*, where each node is associated with a random variable and the presence of edges between nodes represents conditional relationship ("a causes b"). At each node s_i, whose parent nodes are $P_i = \{s_k, ..., s_l\}$, is associated a conditional density $p(s_i|P_i)$. The joint distribution over all the variables, $p(S)$, is equal to the product of all the conditional densities [61]*

$$p(S) = \prod_{i=1}^{T} p(s_i|P_i) \tag{6.15}$$

[1] An acyclic directed graph $G = (N, E)$ consists in a set of nodes N connected by a set of edges $E = \{(a,b)|a,b \in N\}$. If an edge $(a,b) \in E$ exists in the graph, then a is a parent node for b, and b is a child nod for a; [89].

A well known example of Bayes Netwok is given in Fig.6.1, called "The sprinkler network". In this toy problem, the task is to analyze the process that leads the grass to being wet.

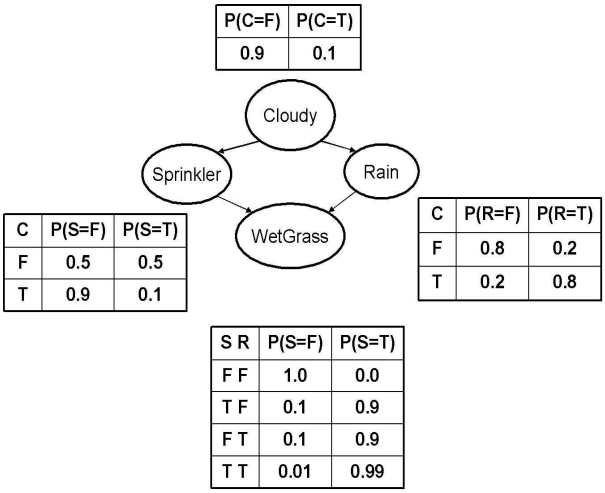


Fig. 6.1. Bayes Net structure

As we see in the figure, the generative model that describes the causes related with the wet grass is fully specified: the pseudo algorithm described in Sec.6.1 has been completely exploited.

In facts:

- all the hidden causes supposed to be related with the visible event "the grass is wet", i.e., the sprinkler, the rain, the presence of clouds, are present in the graph and related with edges; this corresponds to step 1 ("Intuitive definition");
- the uncertainty in the process is added in the analysis, by defining the joint probability of all the nodes in the graph, i.e. $P(C, S, R, W)$ (step 2 of the building process, "Statistical Definition"). The graphical framework, that intuitively codifies our prior knowledge over the hidden process, permits to factorize the joint distribution in a compact way. A "blind" factorization would bring to
  $P(C, S, R, W) = P(C) \cdot P(S|C) \cdot P(R|C, S) \cdot P(W|C, S, R)$.
  By including the graphical definition of conditional independency, this factorization can be simplified. The simplest conditional independence relationship encoded in a Bayesian network can be stated as follows: a node is independent of its ancestors given its parents, where the ancestor/parent relationship is with respect to some fixed topological ordering of the nodes. By using conditional independence relationships, we can rewrite this as
  $P(C, S, R, W) = P(C) \cdot P(S|C) \cdot P(R|C) \cdot P(W|S, R)$
  where we were allowed to simplify the third term because R is independent of S given its parent C, and the last term because W is independent of C given its parents S and R. We can see that the conditional independence relationships

allow us to represent the joint more compactly. Here the savings are minimal, but in general, if we had n binary nodes, the full joint would require $O(2^n)$ space to represent, but the factored form would require $O(n2^k)$ space to represent, where k is the maximum fan-in of a node. And fewer parameters makes learning easier;
- the step 3 ("Parametrization") in this case is encoded supposing each variable as discrete variable, thus requiring for each node a Conditional Probability Distribution (CPD). In this case, all nodes are binary, i.e., have two possible values, denoted by T (true) and F.
- the step 4 ("learning step") has been also performed, being all the CPD fully specified.

6.3.1 Graphical representation of the mixture of Gaussians

The graphical representation of the mixture of Gaussians is reported in Fig. 6.2.

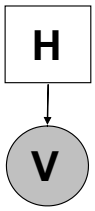


Fig. 6.2. Graphical representation of a mixture of Gaussians generative model: usually in the graphical representation only the variables are depicted, not the parameters.

The gray-shaded circle represent the visible variable O, which parametrization is given by two parameters (mean and variance). This variable is dependent on the hidden variable H, depicted as the white box, that indicates which of the components has been chosen to generate that data. As usual in literature, circles mean continue variables, squared boxes point out discrete variables, white nodes are hidden variables, gray shaded nodes indicates visible variables.

6.3.2 Conditional independence in Bayes Nets

The relation of conditional independence is fundamental to syntactically encode a graphical "intuition" of the process; in general, the conditional independence relationships encoded by a Bayes Net are best explained by means of the "Bayes Ball" algorithm (due to Ross Shachter [149]), which is as follows: two (sets of) nodes $\mathcal{A}$ and $\mathcal{B}$ (Fig. 6.3) are conditionally independent (d-separated) given a set $\mathcal{C}$ if and only if there is no way for a ball to get from $\mathcal{A}$ to $\mathcal{B}$ in the graph, where

the allowable movements of the ball are shown below. The dotted arcs indicate direction of flow of the ball.

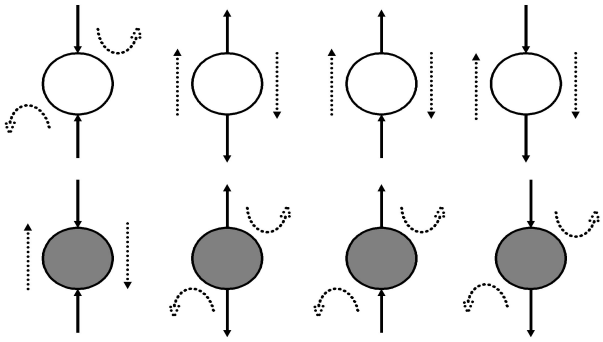


Fig. 6.3. Bayes Ball: two sets $\mathcal{C}$ of nodes, that separate the sets $\mathcal{A}$ and $\mathcal{B}$, not visible in the scheme

In the first column, we have two arrows converging on a node C (so C is a "leaf" with two parents). If C is hidden, its parents are marginally independent, and hence the ball does not pass through (the ball being "turned around" is indicated by the curved arrows); but if C is observed, the parents become dependent, and the ball does pass through.

In the second column in which we have two diverging arrows from C (so C is a "root"). If C is hidden, the children are dependent, because they have a hidden common cause, so the ball passes through. If C is observed, its children are rendered conditionally independent, so the ball does not pass through. Finally, consider the case in which we have one incoming and outgoing arrow to C. It is intuitive that the nodes upstream and downstream of C are dependent iff C is hidden, because conditioning on a node breaks the graph at that point.

7

Hidden Markov Models

This chapter introduces the Hidden Markov Model methodology: the literature is briefly reviewed, applications are described and some technical details are presented.

The literature concerning Hidden Markov Models is very vast. At the risk of unintentional unfairness, the first cited paper is nevertheless the survey that most strongly influenced the author: the Rabiner paper [123], which clearly introduces Hidden Markov Models and their application to speech recognition. Other useful review papers were proposed by Bengio [13], where recent learning algorithms and extensions of the basic model are reviewed, and by Ghahramani [62], in which HMMs are introduced in the context of recent literature on Bayesian Networks [74]. Finally, a very comprehensive list of references on Hidden Markov Models can be found on [29], updated to March 2001.

The chapter is organized as follows: in Section 7.1, a non comprehensive list of applications where HMM were successfully employed is presented. Subsequently, in Section 7.2, HMMs are formally introduced, and the related three common problems are described.

7.1 Applications of Hidden Markov Models

Speech recognition is surely, in order of time, the first and most important application of HMMs. Hundreds of papers appeared on this argument, but, for not unfairly leaving out important works, only few historical papers are reported: [95], [123], and [122], which not only contains information about HMM, but also a very comprehensive review of the speech recognition problem.

In the last decade HMMs were successfully applied to a impressively large number of problems: in the following, a list of applications using HMMs is reported, with some references as example:

- handwriting character recognition: on-line [77, 94, 145] and off-line [112] recognition;
- computer vision: image classification [98], gesture recognition [55, 153], action classification [81], face classification [56, 90, 130], 2D shape classification [73], texture classification [120] and 3D range object recognition [68];

- signal processing [44];
- finance [126, 129];
- meteorology [126];
- geomagnetism [126];
- neurons signal analysis [28, 124];
- acoustics [125];
- bioinformatics: DNA sequence and protein analysis [54] and identification of ion channel currents [146–148];
- EEG modelling [116, 117];
- robotics [69];
- communications [91];

This is far from being an exhaustive list, but its aim is simply giving an idea of the several possible applications of this methodology: for a wider list of references, see the very comprehensive [29]. Other examples of applications will be presented in the following chapters of this thesis.

7.2 Fundamentals

A discrete-time first-order HMM [123] is a probabilistic generative model that describes a stochastic sequence $\mathbf{O} = O_1, O_2, ..., O_T$ as being an indirect observation of an underlying (hidden) random sequence $\mathbf{Q} = Q_1, Q_2, ..., Q_T$, where this hidden process is Markovian, even the observed process may not be so. Let us briefly recall the concept of Markovian process: a process is said to be Markovian of order p if

$$P(Q_t|Q_{t-1}, Q_{t-2}, ..., Q_1) = P(Q_t|Q_{t-1}, Q_{t-2}, ..., Q_{t-p}) \tag{7.1}$$

The process associated with the HMM is Markovian of order one, i.e.

$$P(Q_t|Q_{t-1}, Q_{t-2}, ..., Q_1) = P(Q_t|Q_{t-1}) \tag{7.2}$$

A discrete first-order HMM is formally defined by the following elements:

- A set $S = \{S_1, S_2, \cdots, S_k\}$ of (hidden) states. Although these states are hidden, there are some practical applications (especially in speech and handwriting recognition tasks) where some physical significances could be attached to the said states. We denote the state at time t as Q_t.
- A transition matrix $\mathbf{A} = \{a_{ij}\}$, of dimension $k \times k$, where element $a_{ij} \geq 0$ represents the probability of going from state S_i to state S_j:

$$a_{ij} = A(S_i \rightarrow S_j) = P[Q_{t+1} = S_j|Q_t = S_i] \qquad 1 \leq i, j \leq k \tag{7.3}$$

  Of course

$$a_{ij} \geq 0 \qquad \text{and} \qquad \sum_{j=1}^{k} a_{ij} = 1$$

  Such a matrix is called a *stochastic matrix*;
- A set $V = \{v_1, v_2, \cdots, v_m\}$ of observation symbols: this is the alphabet, and it corresponds to the physical outputs of the process being modelled.

- A $(k \times m)$ emission matrix $\mathbf{B} = \{b(j|S_i)\}$, indicating the probability of emission of symbol v_j from state S_i:

$$b(j|S_i) = P[O_t = v_j | \, Q_t = S_i] \qquad 1 \leq i \leq k, \;\; 1 \leq j \leq m \tag{7.4}$$

with

$$b(j|S_i) \geq 0 \qquad \text{and} \qquad \sum_{j=1}^{m} b(j|S_i) = 1$$

- An initial state probability distribution $\boldsymbol{\pi} = \{\pi_i\}$,

$$\pi_i = \pi(S_i) = P[q_1 = S_i] \qquad 1 \leq i \leq k, \tag{7.5}$$

with,

$$\pi_i \geq 0 \qquad \text{and} \qquad \sum_{i=1}^{k} \pi_i = 1$$

An HMM is completely specified by a 5-tuple $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and defines a joint probability distribution on the space of hidden and observed sequences, *i.e.*, $P(\mathbf{O} = \mathbf{o}, \mathbf{Q} = \mathbf{q}|\boldsymbol{\lambda})$.

The HMM could also be used as generator model, in order to give an observation sequence $\mathbf{O} = O_1, O_2, ..., O_T$. This is carried out by following the algorithm [123]:

1. choose an initial state $Q_1 = S_i$ according to the initial state distribution $\boldsymbol{\pi}$;
2. set $t = 1$;
3. chose $O_t = v_j$ according to the symbol probability distribution in state S_i, that is $b(j|S_i)$;
4. transit to a new state $Q_{t+1} = S_j$ according to the state transition probability distribution for state S_i, i.e. a_{ij};
5. set $t = t+1$; if $t < T$ return to step 3, otherwise terminate the procedure.

7.2.1 Types of HMM

The previous definition specifies *discrete*, *ergodic* and *stationary* Hidden Markov Models. These three characteristics are related to two parameters: the emission matrix (discrete) and the transition matrix (ergodic and stationary). There are different alternatives to them, implying different types of Hidden Markov Models.

With regards to the transition matrix, the ergodic model is the most common type: the HMM has a full state transition matrix, and every state could be reached from every other state of the model (see Fig. 7.1(a)). For some applications, as speech recognition, other topologies have been found to account for the specific problem better than the standard ergodic topology. One example of such a model is the *left-right* HMM [4], presented in Fig. 7.1(b). In this case, the HMM has only a partial state transition matrix: as time increases, the state index increases (or remains the same), i.e. the states proceed from left to right. Formally, this implies

$$a_{ij} = 0, \quad \forall j < i$$

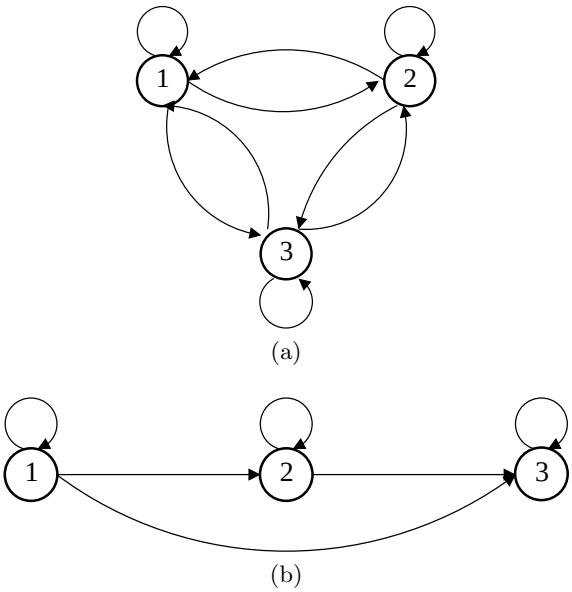


Fig. 7.1. Different topologies: (a) three state ergodic HMM; (b) three state left-right HMM.

Another aspect concerning the transition matrix is its stationarity: if

$$P[Q_{t+1} = S_j | Q_t = S_i] = P[Q_{t+r+1} = S_j | Q_{t+r} = S_i] = a_{ij} \quad \forall r$$

then the Hidden Markov Model is called *stationary* (a_{ij} does not vary over time). Otherwise, HMM is called *non-stationary*.

With regards to the emission probability, it is worth noting that in many interesting applications V is a continuous set, e.g. $V = I\!R$, and that it is advantageous to use HMMs with continuous observation densities. In this case, instead of a matrix of symbol probabilities $\mathbf{B}$, for each state S_i we have an emission probability density function (pdf) $b(o|S_i)$, for $o \in V$, and of course with $\int_V b(o|S_i)\, do = 1$. The most general representation of the pdf is a finite mixture of the form

$$b(o|S_i) = \sum_{m=1}^{M_i} c_{im} \mathcal{F}(o, \boldsymbol{\mu}_{im}, \mathbf{U}_{im}) \quad 1 \leq i \leq k \tag{7.6}$$

where o is the vector being modelled, c_{im} is the mixture coefficient for the mth mixture in state S_i and $\mathcal{F}$ is any log-concave or elliptically symmetric density, with mean vector $\boldsymbol{\mu}_{im}$ and covariance matrix $\mathbf{U}_{im}$. For real (scalar or vectorial) observations, a very common approach is to model $b(o|S_i)$ as a mixture of Gaussians,

$$b(o|S_i) = \sum_{j=1}^{M_i} c_{ij} \mathcal{N}\left(o|\boldsymbol{\mu}_{ij}, \boldsymbol{\Sigma}_{ij}\right). \tag{7.7}$$

In the equation above, $\mathcal{N}(o|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a Gaussian density of mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, evaluated at o. The observations from state S_i are therefore modelled

as samples from a Gaussian mixture with M_i components. In this mixture-based case, for which the adaptation of the Baum-Welch procedure is straightforward [86], $\mathbf{B}$ contains all the mixtures parameters (all the M_i's, all the $\boldsymbol{\mu}_{ij}$'s, etc.); the HMM, in this case, is completely defined by $\boldsymbol{\lambda} = (S, \mathbf{A}, \boldsymbol{\pi}, \mathbf{B})$.

Although the general formulation of continuous density HMMs could be applied to a wide range of problems, there is another very interesting class of HMMs that seems to be particularly suitable in certain cases, as for example speech recognition or EEG modelling: the autoregressive HMMs [87]. In this case, the observation vectors are drawn from an autoregressive process, and the emission probability $b(O_t|S_i)$ is defined as

$$P(O_t|S_i) = \mathcal{N}(O_t - \mathbf{F}_t\hat{\mathbf{r}}_i, \sigma_i^2) \tag{7.8}$$

where $\mathbf{F}_t = -[O_{t-1}, O_{t-2}, \cdots, O_{t-p}]$, $\hat{\mathbf{r}}_i$ is the (column) vector of AR coefficients for the ith state and σ_i^2 is the estimated observation noise for the i-th state. The prediction for the ith state is

$$\hat{O}_t^i = \mathbf{F}_t\hat{\mathbf{r}}_i$$

The order of the AR model is p.

7.3 The three basic problems of HMM

There are three main problems related to the HMM use:

1. Given the HMM $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and the observation sequence $\mathbf{O} = O_1, O_2, \cdots, O_T$ (with $O_t \in V$, for $t = 1, 2, ..., T$), compute the marginal probability $P(\mathbf{O}|\boldsymbol{\lambda})$, i.e. the likelihood function, representing the probability that $\mathbf{O}$ was generated by the model $\boldsymbol{\lambda}$. This is usually defined as the *evaluation problem*.
2. Given $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and an observed sequence $\mathbf{O} = O_1, O_2, ..., O_T$, determine the state sequence $\hat{\mathbf{Q}} = \hat{Q}_1, \hat{Q}_2, \cdots, \hat{Q}_T$ (with $\hat{Q}_t \in \{S_1 ... S_k\}$) that best "explain" the observations, i.e. that most probably generated the observation. This is usually called the *decoding problem*.
3. Given a set of L observed strings $\mathcal{O} = \{\mathbf{O}^{(l)}\}$, where $1 \leq l \leq L$, and $\mathbf{O}^{(l)} = O_1, O_2, ..., O_{T_l}$, adjust the parameters of a HMM $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ in order to maximize $P(\mathcal{O}|\boldsymbol{\lambda})$. This is usually referred to as the *training problem*.

7.3.1 Solution to Problem 1

Given the HMM $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and an observations sequence $\mathbf{O} = O_1, O_2, \cdots, O_T$, the goal is the computation of the marginal probability $P(\mathbf{O}|\boldsymbol{\lambda})$. By using marginalization and the Bayes theorem, $P(\mathbf{O}|\boldsymbol{\lambda})$ could be computed as

$$\begin{aligned} P(\mathbf{O}|\boldsymbol{\lambda}) &= \sum_{\text{all } \mathbf{Q}} P(\mathbf{O}, \mathbf{Q}|\boldsymbol{\lambda}) \\ &= \sum_{\text{all } \mathbf{Q}} P(\mathbf{O}|\mathbf{Q}, \boldsymbol{\lambda}) P(\mathbf{Q}|\boldsymbol{\lambda}) \end{aligned}$$

Now, assuming statistical independence for observations, we have that

$$\begin{aligned} P(\mathbf{O}|\mathbf{Q},\boldsymbol{\lambda}) &= \prod_{t=1}^{T} P(O_t|Q_t,\boldsymbol{\lambda}) \\ &= b(O_1|Q_1)b(O_2|Q_2)...b(O_T|Q_T) \end{aligned} \tag{7.9}$$

The other factor $P(\mathbf{Q}|\boldsymbol{\lambda}))$ could be computed as

$$P(\mathbf{Q}|\boldsymbol{\lambda})) = \pi_{Q_1} a_{Q_1 Q_2} a_{Q_2 Q_3} ... a_{Q_{T-1} Q_T}$$

Putting all together, we obtain

$$\begin{aligned} P(\mathbf{O}|\boldsymbol{\lambda}) &= \sum_{\text{all } \mathbf{Q}} P(\mathbf{O}|\mathbf{Q},\boldsymbol{\lambda}) P(\mathbf{Q}|\boldsymbol{\lambda})) \\ &= \sum_{\text{all } Q_1,Q_2,...,Q_T} \pi_{Q_1} b(O_1|Q_1) a_{Q_1 Q_2} b(O_2|Q_2) ... a_{Q_{T-1} Q_T} b(O_T|Q_T) \end{aligned} \tag{7.10}$$

A little thought should convince the reader that the calculation of $P(\mathbf{O}|\boldsymbol{\lambda})$ with (7.10) involves the order of $2T \cdot k^T$ calculation: this computation is unfeasible, and a more efficient procedure is required. Such a method exists, the so-called *forward-backward procedure* [8, 11]. This technique is based on two variables, the *forward* variable $\alpha_t(i)$ and the *backward* variable $\beta_t(i)$. The former ($\alpha_t(i)$), is defined as

$$\alpha_t(i) = P(O_1 ... O_t, Q_t = S_i|\boldsymbol{\lambda}) \tag{7.11}$$

and represents the probability to have observed the sequence $O_1 ... O_t$ up to time t, and being in state S_i. It is recursively computed by the following formulas

$$\begin{aligned} \alpha_1(i) &= \pi_i b(O_1|S_i) && 1 \le i \le k \\ \alpha_{t+1}(i) &= \left[\sum_{j=1}^{k} \alpha_t(j) a_{ji} \right] b(O_{t+1}|S_i) && 1 \le t \le T-1,\ 1 \le i \le k \end{aligned}$$

The *backward* variable is defined as

$$\beta_t(i) = P(O_{t+1} ... O_T|Q_t = S_i, \boldsymbol{\lambda}) \tag{7.12}$$

and represents the probability to observe the symbols $O_{t+1} ... O_T$, being in the state S_i at time t. This variable is recursively computed:

$$\begin{aligned} \beta_T(i) &= 1 && 1 \le i \le k \\ \beta_t(i) &= \sum_{j=1}^{k} a_{ij} b(O_{t+1}|S_j) \beta_{t+1}(j) && t = T-1, ..., 1,\ 1 \le i \le k \end{aligned}$$

$P(\mathbf{O}|\boldsymbol{\lambda})$ is then computed as,

$$P(\mathbf{O}|\boldsymbol{\lambda}) = \sum_{i=1}^{k} \alpha_t(i)\beta_t(i) \qquad \forall t \tag{7.13}$$

By fixing $t = T$ we obtain

$$P(\mathbf{O}|\boldsymbol{\lambda}) = \sum_{i=1}^{k} \alpha_T(i) \tag{7.14}$$

7.3.2 Solution to Problem 2

Given $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and an observed sequence $\mathbf{O} = O_1, O_2, ..., O_T$, the aim is to determine the "optimal" state sequence that generated $\mathbf{O}$. Several criteria could be adopted to define the concept of "optimality". Finding the single best path that maximizes the probability to generate the sequence is the usual one. The goal is therefore to find the state sequence $\hat{\mathbf{Q}} = \hat{Q}_1, \hat{Q}_2, \cdots, \hat{Q}_T$, such that

$$\hat{\mathbf{Q}} = \arg\max_{\mathbf{Q}} P(\mathbf{O}, \mathbf{Q}|\boldsymbol{\lambda}).$$

This problem is solved by the *Viterbi algorithm* [59,150]. This procedure starts by defining the quantity

$$\delta_t(i) = \max_{Q_1 \cdots Q_{t-1}} P(Q_1, Q_2, ..., Q_t = S_i, O_1, O_2, ..., O_t|\boldsymbol{\lambda}) \tag{7.15}$$

representing the best score (i.e. the highest probability) along a single path, at time t, which accounts for the first t observations and ends at state S_i. To retrieve the state sequence, the argument of the δ_i has to be stored for each t and each i: this is obtained using the vector $\psi_t(i)$.

The Viterbi algorithm is then defined by the following recursive steps:

1. Initialization:
$$\begin{aligned} \delta_1(i) &= \pi_i b(O_1|S_i) \quad && 1 \le i \le k \\ \psi_1(i) &= \emptyset && 1 \le i \le k \end{aligned}$$
2. Recursion:
$$\begin{aligned} \delta_t(i) &= \max_{1 \le i \le k} [\delta_{t-1}(i) a_{ij}] b(O_t|S_j) && 1 \le j \le k \\ & && 2 \le t \le T \end{aligned}$$
$$\begin{aligned} \psi_t(i) &= \arg\max_{1 \le i \le k} [\delta_{t-1}(i) a_{ij}] && 1 \le j \le k \\ & && 2 \le t \le T \end{aligned}$$
3. Termination:
$$\begin{aligned} \hat{P} &= \max_{1 \le i \le k} [\delta_T(i)] \\ \hat{Q}_T &= \arg\max_{1 \le i \le k} [\delta_T(i)] \end{aligned}$$
4. State sequence backtracking:
$$\hat{Q}_t = \psi_{t+1}(\hat{Q}_{t+1}) \qquad t = T-1, T-2, ..., 1$$

7.3.3 Solution to Problem 3

Given a set of L observed strings $\mathcal{O} = \{\mathbf{O}^{(l)}\}$, where $1 \leq l \leq L$, and $\mathbf{O}^{(l)} = O_1, O_2, ..., O_{T_l}$, assumed to be independent samples taken from a common HMM $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, the aim is to determine $\boldsymbol{\lambda}$. This is the most difficult problem, which is usually solved by adopting the Maximum Likelihood criterion, that is,

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} P(\mathcal{O}|\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{\lambda}} \prod_{l=1}^{L} P(\mathbf{O}^{(l)}|\boldsymbol{\lambda});$$

The best-known algorithm to implement this ML criterion is the so-called *Baum-Welch re-estimation* technique [7–11]. This is a particularization of the Expectation-Maximization algorithm (see Chap.5; here the missing data is the hidden sequence $\mathbf{Q}$).

In order to describe the procedure for the re-estimation of the HMM parameters, two (hidden) variables have to be introduced:

- $\xi_t(i, j)$: it represents the probability of passing from state S_i at time t to state S_j at time $t + 1$, given the observations and the model, i.e.

$$\xi_t(i, j) = P(Q_t = S_i, Q_{t+1} = S_j|\mathbf{O}, \boldsymbol{\lambda}) \tag{7.16}$$

  This variable is computed using forward and backward variables as

$$\begin{aligned}
\xi_t(i, j) &= P(Q_t = S_i, Q_{t+1} = S_j|\mathbf{O}, \boldsymbol{\lambda}) \\
&= \frac{P(Q_t = S_i, Q_{t+1} = S_j, \mathbf{O}|\boldsymbol{\lambda})}{P(\mathbf{O}|\boldsymbol{\lambda})} \\
&= \frac{\alpha_t(i) a_{ij} b(O_{t+1}|S_j) \beta_{t+1}(j)}{P(\mathbf{O}|\boldsymbol{\lambda})} \\
&= \frac{\alpha_t(i) a_{ij} b(O_{t+1}|S_j) \beta_{t+1}(j)}{\sum_i \sum_j \alpha_t(i) a_{ij} b(O_{t+1}|S_j) \beta_{t+1}(j)}
\end{aligned} \tag{7.17}$$

  Note that the sum of $\xi_t(i, j)$ over time t could be interpreted as the expected number of transitions from state S_i to state S_j.
- $\gamma_t(i)$: it represents the probability of being in state S_i at time t, given the observation and the model, i.e.

$$\gamma_i = P(Q_t = S_i|\mathbf{O}, \boldsymbol{\lambda}) \tag{7.18}$$

  This variable is computed as

$$\begin{aligned}
\gamma_t(i) &= P(Q_t = S_i|\mathbf{O}, \boldsymbol{\lambda}) \\
&= \frac{P(Q_t = S_i, \mathbf{O}|\boldsymbol{\lambda})}{P(\mathbf{O}|\boldsymbol{\lambda})} \\
&= \frac{\alpha_t(i) \beta_t(i)}{\sum_i \alpha_t(i) \beta_t(i)}
\end{aligned} \tag{7.19}$$

The variable $\gamma_i(t)$ could be also expressed in terms of the variable $\xi_t(i,j)$, giving

$$\gamma_t(i) = \sum_{j=1}^{k} \xi_t(i,j) \tag{7.20}$$

In this case too, the sum of $\gamma_t(i)$ over the time t can be interpreted as the expected number of transitions from S_i.

Given those two variables, the re-estimation procedure determines the values of the parameters $\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}$ with the following procedure:

$$\begin{aligned} \bar{\pi}_i &= \text{expected frequency in state } S_i \text{ at time } t = 1 \\ &= \gamma_1(i) \end{aligned} \tag{7.21}$$

$$\begin{aligned} \bar{a}_{ij} &= \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \tag{7.22}$$

$$\begin{aligned} \bar{b}(v_j|S_i) &= \frac{\text{expected number of times in state } S_i \text{ and observing symbol } v_j}{\text{expected number of times in state } S_i} \\ &= \frac{\sum_{\substack{t=1 \\ \text{s.t. } O_t = v_j}}^{T-1} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \tag{7.23}$$

If we define the current model as $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and use it to compute the right hand side of (7.21), (7.22) and (7.23), we define the re-estimated model as $\bar{\boldsymbol{\lambda}} = (S, V, \bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\boldsymbol{\pi}})$. It has been shown by Baum and his colleagues that

$$P(\mathbf{O}|\bar{\boldsymbol{\lambda}}) > P(\mathbf{O}|\boldsymbol{\lambda})$$

which means that, at each iteration, the likelihood has increased, until a maximum is reached. The final result of this procedure is the trained HMM, called the *Maximum Likelihood estimate* of the HMM, because, as we see in Sec. 4.1, the parameters of the HMM are considered as unknown but fixed quantities.

Moreover, it has been proved [123] that this formulation could be perfectly casted into the Expectation-Maximization framework.

Finally, as this problem is an optimization problem, some gradient descent techniques have been employed [95], yielding nevertheless solutions comparable to those of standard re-estimation procedure above presented.

This technique is quite effective and really fast: typically the algorithm converges after about ten iterations. The main drawback is its local behavior, implying the convergence to the nearest local maximum. Since the likelihood is highly multimodal, the initial conditions could crucially affect the effectiveness of the learning. Another severe limitation is the fact that re-estimation formulas are based on computation of frequencies, or expected times, of events. This implies that a big training set is indispensable in order to obtain reliable estimate of the HMM parameters.

Part II

Low level description of a video sequence

8

Overview of the part

This part presents a new approach to video understanding, which aims at extracting structured information from a video surveillance sequence using directly pixel-level data. The method models the sequence using a forest of Hidden Markov Models, which are able to extract a sort of *dynamic* information, that helps to detect those areas more affected by foreground activity.

This information is estimated using an entropy-like measure defined on the stationary probability of the Markov chain associated to the HMMs, producing a visual per-pixel analysis. This approach constitutes an example of per-pixel description of a video sequence, that is unsupervised in nature.

9

Introduction to a pixel-level description

In a pixel level description the video sequence is intended as a bunch of independent signals, one for each pixel location. In this chapter we present a novel approach, where generative modelling is applied on each pixel signal, in order to discover hidden per pixel properties. The exhibition of such properties after the analysis will give a per pixel insight over the sequence.

Actually, many approaches in the literature [1, 71, 151] base their analysis on typical and well known per-region descriptions, i.e., segmentation and tracking. Typically, these procedures are adequate when a priori knowledge is available (for example, the shape of an object to be identified, the number of objects to be tracked, the location of appearing/disappearing objects), but they are weak when this information is not provided. This problem occurs, for instance, when a camera is monitoring a crowded scene, in which multiple occlusions and clutter are present.

A per-pixel description of a video sequence circumvents this problem by performing an analysis at the lowest level, *i.e.*, by considering directly and only the temporal pixel-level behavior.

A somewhat similar idea is at the basis of the methods proposed in [66, 67, 108, 134, 158], in which the extraction of semantic information is carried out without segmentation or trajectory extraction, but performing low- (pixel) and mid-level (blob) analysis, after a background analysis modelling. Nevertheless, our approach is only similar in spirit to the above quoted work since, unlike those approaches, only low-level analysis at pixel level is performed and a different probabilistic method is used.

Another important characteristic of the description regards the modelling tool used to analyze the video sequence. The sequence is modelled using a forest of Hidden Markov Models (HMMs), each one modelling the temporal evolution of a pixel.

The HMMs appear a suitable choice for our tasks: they represent a good combination between expressivity power and low computational complexity.

The rest of the chapter is organized as follows. In Section 9.1, some basic principles related to the Hidden Markov Models, not previously explained, are presented. The proposed strategy is then detailed in Section 9.2, and extensive experimental results and a comparative analysis are presented in Section 9.3. Finally, in Section

9.3.1, conclusions are drawn and remarks about the per pixel modeling of video sequences are presented.

9.1 Methodological issues

In this section, the theoretical concept of stationary probability of a HMM is presented, representing a key entity in the video description proposed.

9.1.1 The stationary probability distribution

Given a HMM $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, consider the Markov chain $\mathbf{Q} = Q_1, Q_2, Q_3...$ with state set $S = \{S_1, ..., S_k\}$, stochastic transition matrix $\mathbf{A}$, and initial state probability $\boldsymbol{\pi}$. We can define the vector of state probabilities at time t as

$$\mathbf{p}_t = [\mathbf{p}_t(1), ..., \mathbf{p}_t(j), ..., \mathbf{p}_t(k)] = [P(Q_t = S_1), ..., P(Q_t = S_j), ..., P(Q_t = S_k)].$$

where $\mathbf{p}_t(i)$ represents the probability of being in state S_i at time t. Obviously, $\mathbf{p}_t$ can be computed recursively from $\mathbf{p}_1 = \boldsymbol{\pi}\mathbf{A}$, $\mathbf{p}_2 = \mathbf{p}_1\mathbf{A} = \boldsymbol{\pi}\mathbf{A}\mathbf{A}$, and so on. In short, $\mathbf{p}_t = \boldsymbol{\pi}\mathbf{A}^t$.

We are interested in the *stationary probability distribution* $\mathbf{p}_\infty$, which characterizes the equilibrium behavior of the Markov chain, *i.e.* when we let it evolve indefinitely. This vector represents the probability that the system is in a particular state after an infinite number of iterations. Since it is a stationary distribution, $\mathbf{p}_\infty$ has to be a solution of

$$\mathbf{p}_\infty = \mathbf{p}_\infty\mathbf{A}$$

or, in other words, it has to be a left eigenvector of $\mathbf{A}$ associated with the unit eigenvalue. Under some conditions (see [24] for details), the Perron-Frobenius theorem states that matrix $\mathbf{A}$ has a unit (left) eigenvalue and the corresponding left eigenvector is $\mathbf{p}_\infty$. All other eigenvalues of $\mathbf{A}$ are strictly less than 1, in absolute value. Finding $\mathbf{p}_\infty$ for a given $\mathbf{A}$ then amounts to solving the corresponding eigenvalue/eigenvector problem.

9.2 The proposed approach

In this section the proposed approach is presented. In 9.2.1 the probabilistic modelling of the sequence is introduced, while the following two sections describe how this representation is used to infer information (Sect. 9.2.2) about the scene.

9.2.1 The probabilistic modelling of video sequences

The proposed approach models the whole sequence as a set of independent per pixel processes (x,y,t), each one describing the temporal gray-level evolution of the location (x,y) of a scene (since the camera is fixed). Given this set of sequences, we want to model them in order to capture their most important characteristics. In

particular, we need a model able to determine: 1) the most stable gray-level components measured in the whole sequence; 2) the temporal chromatic variation of these components; 3) the sequentiality in which the components vary. An adequate computational framework showing these features is constituted by the Hidden Markov Model, presented in Chapter 7. Using this model, all the above requirements can be accomplished. In particular, using HMMs with continuous Gaussian emission probability, the most important gray level components are modelled by the means $\boldsymbol{\mu}_i$ of the Gaussian functions associated to the states, the variability of those components are encoded in the covariance matrices $\boldsymbol{\Sigma}_i$, and the sequentiality is encoded in the transition matrix $\mathbf{A}$. The HMM methodology has been preferred to other similar modelling techniques, such as mixture of Gaussians models (GMM), due to its important characteristic of being able to deal with the temporal sequentiality of the data, which is crucial when analyzing video sequences. GMMs are indeed not able to capture the temporal variability, *i.e.*, the model does not change if the frames of the video-sequence are randomly shuffled, as temporal information is not considered.

Summarizing, the sequence is modelled using a forest of HMMs, one for each pixel.

For what concerns the model selection, the different approaches for determining the number of states of a HMM directly from data (e.g., [16, 17, 22, 141]) are typically computationally demanding. Since the proposed approach trains one HMM per pixel, we have chosen to fix a priori the number of states in order to maintain the computational effort at a reasonable level. This choice is not critical and can be guided from opportune considerations about the complexity of the scene, especially in relation to the complexity of the background. Actually, three states are considered a reasonable choice, taking into account the possibility of a bimodal BG, and one component for the foreground activity [138].

Once fixed the number of states, the HMM training has been carried out using the standard Baum-Welch procedure, paying particular attention to the initialization. Since the Baum-Welch procedure, starting from some initial estimates, converges to the nearest local maximum of the likelihood function, which is typically highly multi-modal, the initialization issue is particularly crucial for the effectiveness of the training. In our approach, a GMM clustering is used to initialize the emission matrix of the HMM before training. In particular, the initialization phase first considers the sequence of pixel gray levels as a set of scalar values (no matter in which order the coefficients appear); second, these values are grouped into three clusters by following a GMM clustering approach, *i.e.*, fitting the data by using three Gaussian distributions, in which the Gaussian parameters are estimated by an EM-like [51, 157] method. Finally, the mean and variance of each cluster are used to initialize the Gaussian of each state, with a direct correspondence between clusters and states.

The computational complexity of the training phase is $O(nI_{max}N^2T)$, where n is the number of the pixels, $I_{max}N^2T$ is due to the standard complexity of the Baum-Welch training phase for each pixel; I_{max} is the maximum number of iterations permitted during the learning step, N^2T is due to the forward and backward variables calculation, where N is the number of the states and T is the length of the sequence.

9.2.2 Dynamic information: the activity maps

The proposed method is able to infer the degree of foreground activity in the scene; we characterize such information as *dynamic*, highlighting the main aspect of the foreground, i.e. of being (spatially) dynamic in the scene. Strictly speaking, we define a measure which is able to quantify for each pixel the related level of activity. By visualizing all these measures, we estimate the "activity map" of the scene, in which the areas more affected by foreground motion are recognizable.

A similar goal was achieved in [49], where the activity zones were found by clustering the object trajectories derived from the tracking. Nevertheless, in our case the analysis is performed without resorting to trajectories, but by the direct use of the pixel signals. Other approaches similar in spirit to our objectives are presented in [21], in which a Motion Energy Image (MEI) is used to represent and index human gestures, in [48], where an enhancement of the MEI is proposed, namely the Motion History Image (MHI), and in [82], where Spatio-Temporal Entropy Image (STEI) were used to detect foreground activity. Some of these approaches are summarized in the experimental section, where they have been experimentally compared with our approach.

In our framework, we define a measure which is able to quantify for each pixel the level of activity related to that pixel, and this is carried out by analyzing the parameters of the associated HMM. The key idea is that the temporal evolution of the pixel grey-level could be considered as composed by different components, each one assigned to a particular state during the HMM training. Each component is then characterized by a degree of importance: some are more important, *i.e.*, "explain" more data, others are less important since they result from disturbing sources (*e.g.*, noise). Therefore, if we are able to measure the "importance" of a state, we could infer the importance of the components of the signal which represents the information that we will use to determine the activity zones. As explained later, the "state importance" can be measured using the stationary probability distribution of the Markov Chain associated with the HMM.

Given a HMM $\boldsymbol{\lambda}_{xy}$ with N states, trained on the sequence of the gray-level values assumed by the pixel (x,y), all the information we need is in the vector $\mathbf{p}_{\infty}^{xy}$. The activity measure $AL(x, y)$ should show some precise characteristics, *i.e.*, it should discard the components relative to the background (unimodal or multimodal), and should clearly detect those relative to the foreground, giving a response proportional to the amount of foreground passed over the pixel. These requirements are accomplished by the following measure:

$$AL(x, y) = \sum_{i=1}^{N} \omega_i^{xy} \mathbf{p}_{\infty}^{xy}(i) \log_2 \frac{1}{\mathbf{p}_{\infty}^{xy}(i)} \tag{9.1}$$

with

$$w_i^{xy} = \log(1 + \sigma_i^{xy}) \tag{9.2}$$

where σ_i^{xy} is the variance of the Gaussian associated to the state S_i of the HMM $\boldsymbol{\lambda}_{xy}$. The term 1 added to the variance ensures that the weights are all positive.

We use the logarithm in order to ensure a smoother increasing behavior of the weights. This formula is a sort of "weighted entropy", and is the result of two

ideas: the use of the entropy, and the weighting of the components in the entropy computation. The measure of entropy has been chosen since it is able to quantify the uncertainty linked to the model of the pixel gray level evolution. The idea of weighting has been introduced in order to deal with multi-modal background (for example in the case of moving foliage), which produces an erroneous high entropy: the idea is to assign lower weight to the terms of the computation that are most related to the background. In this case, we are in fact more interested in the entropy of the states that most probably do not correspond to background, since they represent the activity. The weight is linked to the variance of the Gaussian of the state, so that the lower the variance, the higher the probability that the state corresponds to a background component. By computing this quantity for all the frame pixels, we could finally obtain an activity map of the observed scene. The computational effort required to calculate the activity map is $O(Nn)$.

An immediate consideration that could arise is why we do not directly use the entropy of the gray level evolution of the pixel, rather that the pseudo entropy of the *model* of the gray level evolution. The reasons are essentially two: first, the use of a HMM permits to recover from noise that is present in the video sequence, which cannot be accomplished by the raw entropy computation. Second, and more important, the use of HMMs permits to deal also with multimodal background: the entropy of the raw signal results large in case of multimodal background, whereas with our approach this does not occur since the background states are in some way disregarded from the measure computation. This behavior is confirmed by results presented in the experimental section.

9.3 Experimental trials and comparative analysis

In this section, some comparative experimental evaluations of the proposed approaches are presented, where the strength and the limitations of the proposed description are discussed. Finally, Section 9.3.1 contains some suggestions about the possible use of the information extracted from the video sequence.

As comparative techniques, we considered methods present in the literature (see [21], [82]) and simple modified versions of them.

Summarizing, all the approaches employed in this section are named as follows:

- Motion Energy Image (MEI) [21]: the MEI is the sum of the squared differences between each frame and one chosen as reference (the first of the sequence); in particular, to each difference image is applied a threshold T_{MEI} in order to disregard little values due to noise. The best results of this approach have been obtained using $T_{\mathrm{MEI}} = 4$.
- Modified Motion Energy Image (MMEI): the same approach as above, but the differences are calculated between consecutive frames. This measure will weight much more sudden foreground activities;
- Median over reference squared difference (MedReF): the median operator is applied over the volume of the squared differences with respect to the first frame.
- Median over consecutive squared difference (MedCDif): the median operator is applied over the volume of the consecutive squared differences.

- Simple entropy: for each pixel, we calculate the associated signal entropy in a range of 255 gray-level values. This measure is quite similar to that proposed in [82]: in such approach the entropy is calculated over a time interval of 5 frames, and over a square spatial window of 3×3 pixels.
- The proposed approach.

The first test sequence is composed of 390 frames, acquired at a rate of 15 frames per second. The sequence regards an indoor scene, where a man is entering from the left, walking to a desk, and making a phone call. After the phone call, he leaves the scene going out to the right. Some frames of the sequence are shown in Fig. 9.1.

Fig. 9.1. Some frames of the original sequence.

The activity zones resulting from the application of the proposed approach and the comparative methods are displayed in Fig. 9.2, in which higher gray-level values correspond to larger activity. All the output values of the different methods are scaled in the pictures in the interval [0,255]. The results show that the methods based on the differences with respect to an initial frame (Fig. 9.2a and c) are "complementary" with respect to the ones based on consecutive differences (Fig. 9.2b and d). In the former case, the person near the phone represents the biggest amount of activity, while in the latter the slow motion of the person makes the vibrating phone wire as the strongest activity. The simple entropy method (Fig. 9.2e) includes both the person and the wire as energetic objects in the scene, and it is also visible in the center of the scene a mild energy zone, due to the approaching phase of the person to the phone. Another drawback of this method is that also the background signals (due to reflecting effects in the scene and in the decoding of the movie) are taken into account in the calculus of the activity map. Therefore, high energy patterns are detected in correspondence of the bookshelf, over the chair and under the phone; moreover, a general energy amount is detected over all the scene, due to the compression coding of the sequence. Our method (Fig. 9.2f) avoids all the noise due to the background, highlighting a more precise description of the activity present in the scene. The resulting image is very informative: one

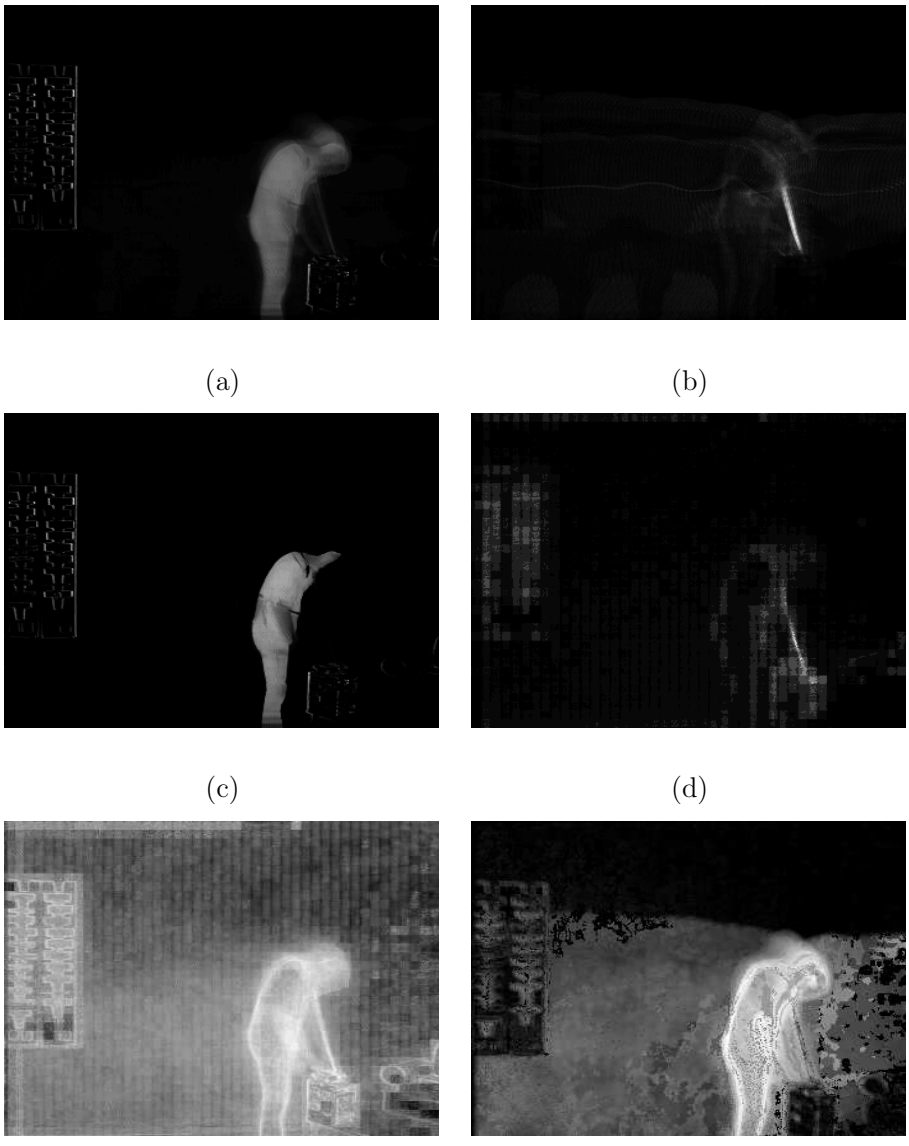


Fig. 9.2. Activity zones resulting from: (a) MEI over reference frame; (b) MEI over consecutive difference (MMEI); (c) Median over reference squared difference (MedReF); (d) Median over consecutive squared difference (MedCDif); (e) Simple entropy; (f) The proposed approach. The whiter the pixels the higher the activity.

could see that the walking zone (*i.e.*, the zone to the left of the desk) is quite active, while the zone near the phone is very active. The zone in the top of the image, where no foreground objects pass, is darker, *i.e.* no activity is present, and only some noisy behavior is visible.

Another interesting example is proposed in Fig. 9.3, where some frames of the video sequence are shown. The camera is monitoring an outdoor scene, where there

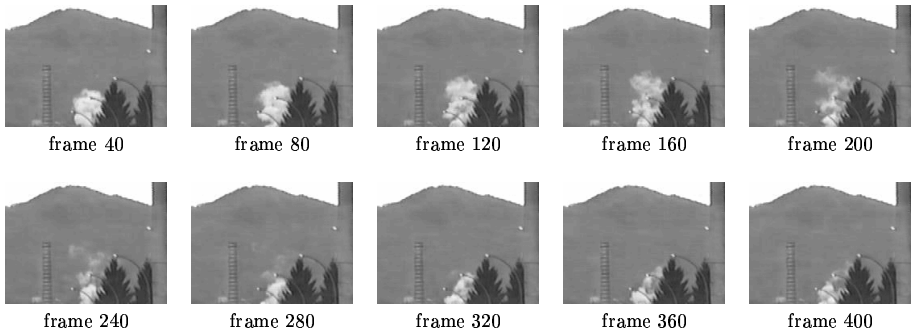


Fig. 9.3. Some frames of the sequence of the fire.

is a starting fire (please note the smoke in the middle). This is a clear example in which object-centered trajectories cannot be extracted, since the moving object has neither a clear shape nor a well-defined contour. The sequence is 450 frames long, which represents 30 seconds of observation. The activity zones, extracted from this video sequence using all the methods, are shown in Fig. 9.4.

In this case, all the methods based on consecutive differences (Fig. 9.4b and d) fail due to the slow motion of the smoke. The methods based on the difference with respect to a reference frame (Fig. 9.4a and c) perform better, even if a clear pattern is not identifiable. In general, all these methods are able to absorb the background noise. The entropy of the sequence, shown in Fig. 9.4e, highlights also the background noise, resulting in a overall high energy scene. Using our approach, depicted in Fig. 9.4f, it is possible to clearly identify the smoke zone, indicating that there is a certain activity. Further, it is important to note that the fire has been detected analyzing only 30 seconds of the scene. The holes present in the image can derive from the lamp-posts which are located ahead of the smoke, as can be noticed by looking at Fig. 9.3. Comparing the two last images we can also notice that they carry similar information: in both cases the smoke area is clearly identified. This is obvious, since the same guiding principle is used: in our case, it is the entropy of the *model* of the signal, whereas in the second it is the entropy of the signal itself. However, the image resulting from our approach clearly separates activity from inactivity (all the remaining part of the scene is dark), while using the "simple entropy" approach the activity in the mountains zone is larger than that of the sky, and this represents an erroneous interpretation.

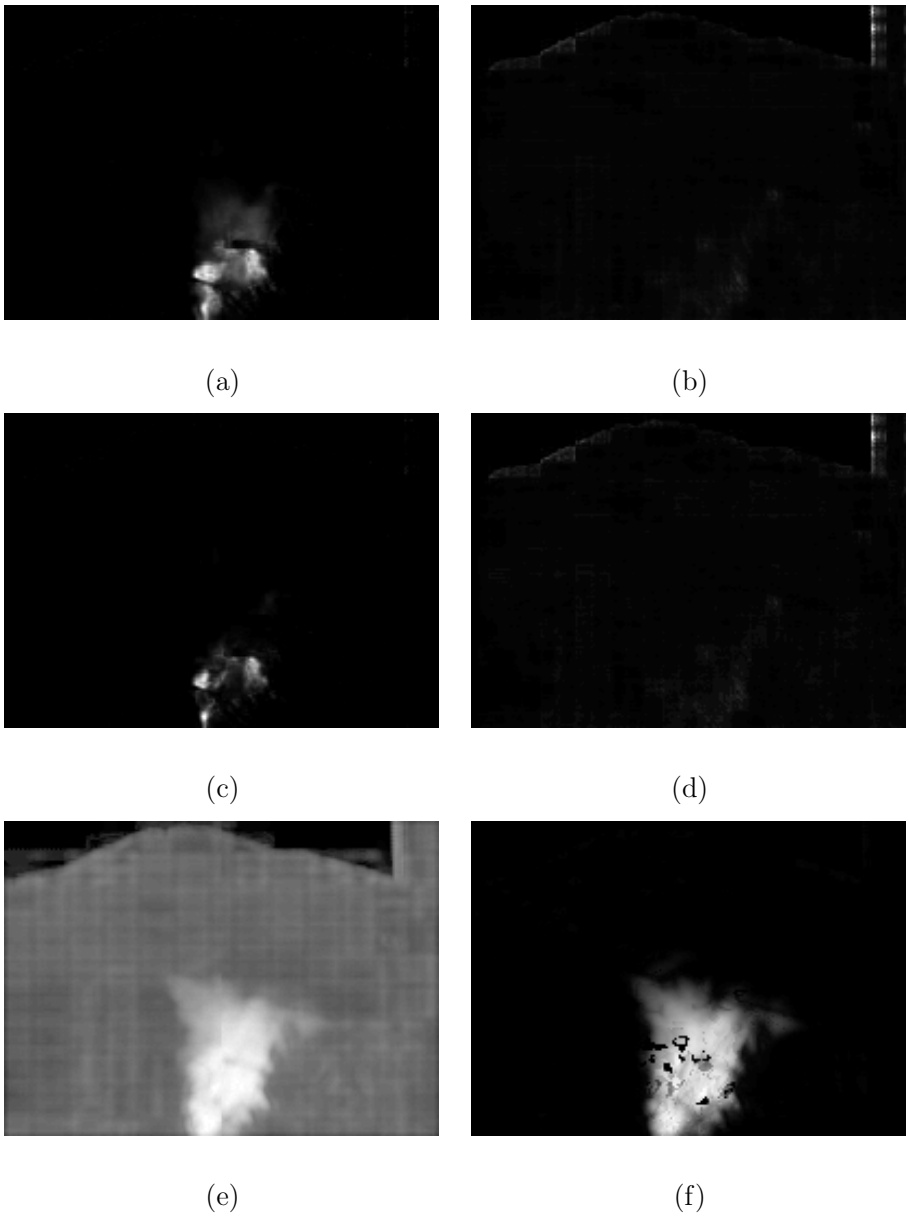


Fig. 9.4. Activity zones resulting from: (a) MEI over reference frame; (b) MEI over consecutive difference; (c) Median over reference squared difference (MedReF); (d) Median over consecutive squared difference (MedCDif); (e) Simple entropy; (f) The proposed approach. The whiter the pixels the higher the activity.

Another interesting aspect has to be pointed out. In the MEI and MedReF approaches, the reference frame has to be carefully chosen: essentially, being the reference frame fixed over time, the evolution of the light and the weather (the background) is not modelled. Our method, as shown in the following example, is able to deal with such kinds of situations.

In another experiment, we have a sequence that shows a person walking in an indoor environment, in which a sudden change of illumination occurs (Fig .9.5). The sequence is formed by 135 frames, (320×240 pixels) acquired at 20 frame/sec..

Fig. 9.5. Frames of the second indoor sequence.

After applying the different approaches, we obtain the following results, shown in Fig. 9.6. The change of illumination in the sequence produces erroneous activity maps in the methods based on differences over a reference frame (Fig. 9.6a and c). As stated before, all these methods work well in the situations in which the background is highly static, as in the case of well constrained indoor environments, or environments considered over short periods of time. In situations in which the chromatic aspect of the background is changing over time, all these methods are not applicable. In the methods based on consecutive differences (Fig. 9.6b and d), the change of illumination is better absorbed: it is 5 frames long, therefore, each consecutive difference image has smaller pixel (absolute) values than the one built between the current frame and the reference one. Nevertheless, that quantity is bigger with respect to the values of the consecutive differences due to the moving person in the scene: the overall result is that the change of illumination visually predominates on the moving object. Looking at the Fig. 9.6e, we can notice that the "simple entropy" method completely fails in that the illumination change occurring in the middle of the sequence makes not possible to recover any meaningful information. Actually, one can notice that the resulting image does not provide any expressive interpretation being quite uniform. On the other side, our method is able to recover useful information about the movement of the person in the hallway. In particular, looking at the Fig. 9.8(f), we could infer three correct information: 1) the top part of the scene is not active, which is correct; 2) there

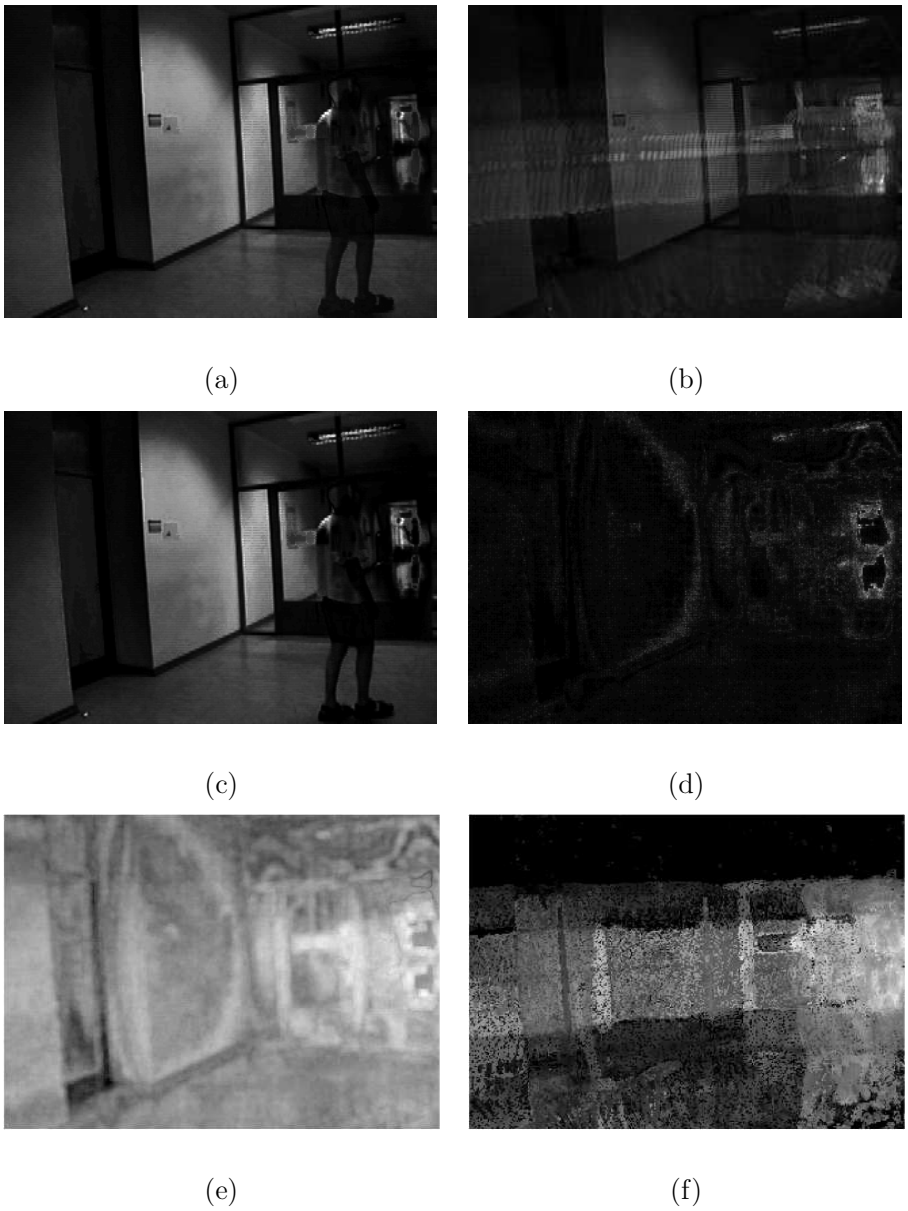


Fig. 9.6. Activity zones resulting from: (a) MEI over reference frame; (b) MEI over consecutive difference (MMEI); (c) Median over reference squared difference (MedReF); (d) Median over consecutive squared difference (MedCDif); (e) Simple entropy; (f) The proposed approach. The whiter the pixels the higher the activity.

is something moving in the bottom, going through all the scene, and this is also correct; 3) the right part of the scene is more active than the left part: this is still correct, since the man starts walking (Fig. 9.5) in the middle part of the scene and come back in from the right.

Another testing sequence regards an outdoor environment where two persons are closing and come back. A few frames of the sequence are presented in Fig. 9.7.

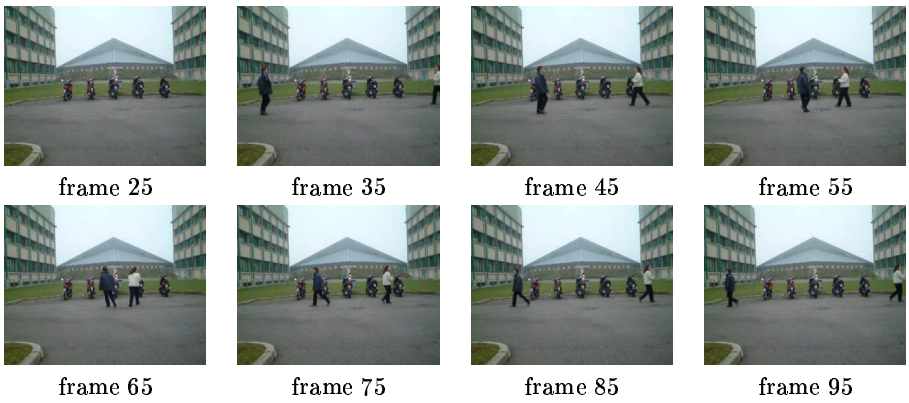


Fig. 9.7. Some frames from the outdoor sequence.

The comparative results express the same considerations made for the previous examples, highlighting a general robustness of the proposed method (see Fig. 9.8). In general, all the noisy background situations have great impact over the final energy image: the more noisy the background, the less important the role of the foreground on the final map. In general, the entropy-based method over relatively short sequences (100 - 500 frames) is highly prone to over estimated energy errors. Moreover, when the foreground appears briefly in the scene, the median-based methods (Fig. 9.8c and d) tend to prune away the correspondent activity, and the simple entropy method (Fig. 9.8e) strongly highlights the light noise activity, in this case due to the video compression. Looking at the proposed approach (Fig. 9.8f), one could notice that the image is quite informative: the part of the scene where people are walking is clearly expressed, as well as the non active part. Moreover it is interesting to notice that it is possible to precisely infer also some further details, as the positions where the legs of the people are standing more time, which represents a larger level of detail. This detail is also represented by the MEI and MMEI approaches (Fig. 9.8a and b, respectively), although with less strength.

The second and the third sequence of this section should be considered the most hard onesThese limits will draw the directions of our research. The former sequence represents an outdoor environment[1], in which a traffic situation over a

[1] downloaded from `ftp://ftp.ira.uka.de/pub/vid-text/image_sequences/kwbB/sequence.mpg`.

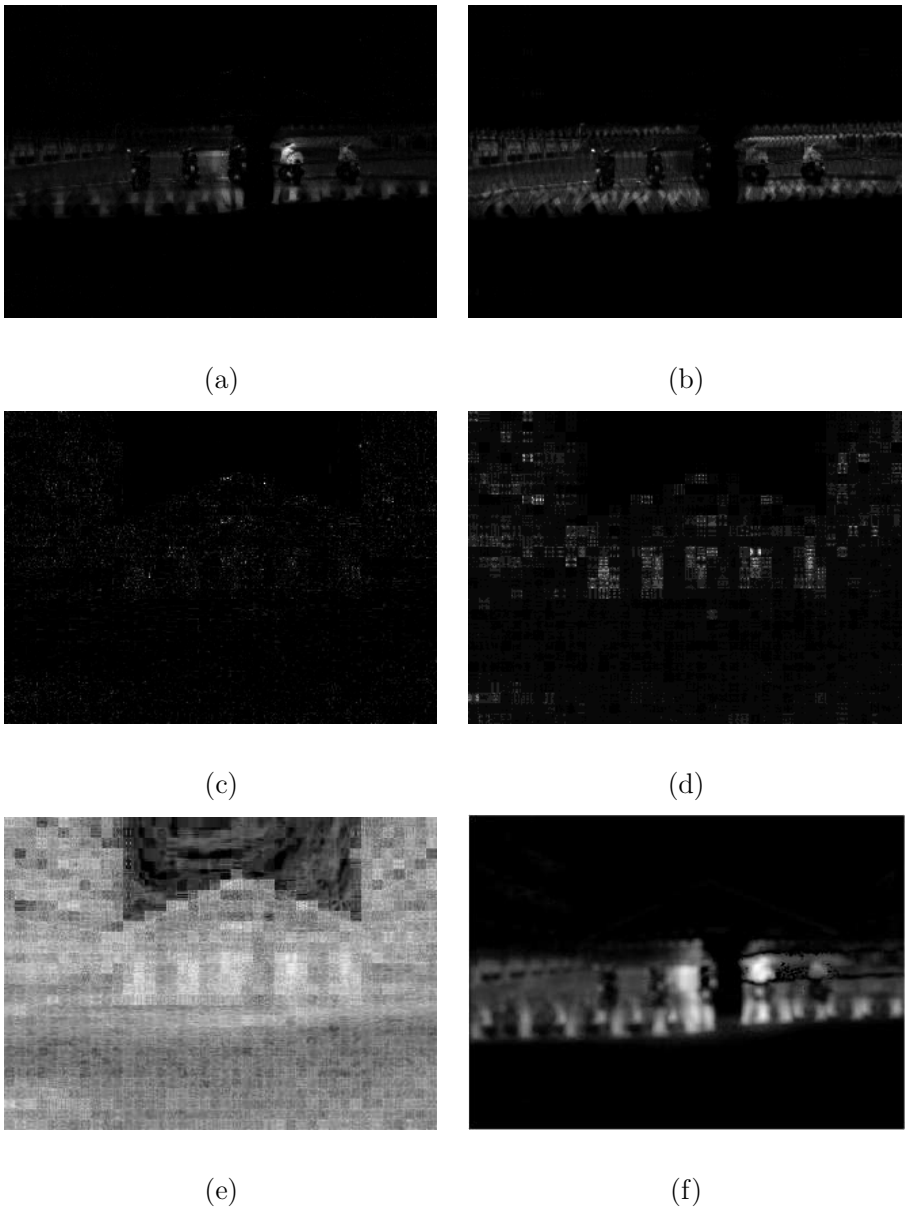


Fig. 9.8. Activity zones resulting from: (a) MEI over reference frame; (b) MEI over consecutive difference (MMEI); (c) Median over reference squared difference (MedReF); (d) Median over consecutive squared difference (MedCDif); (e) Simple entropy; (f) The proposed approach. The whiter the pixels the higher the activity.

square is monitored via a fixed camera (Fig. 9.9). The sequence is 1710 frames

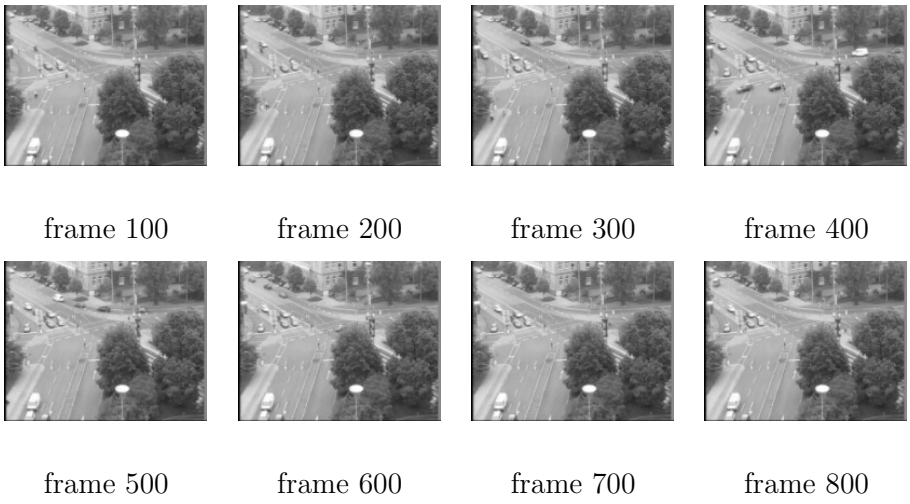


Fig. 9.9. Some frames from the traffic sequence.

long, acquired at 30 fps.

The results are shown below (Fig. 9.10).

As one can notice, the only useful results are the ones relatives to the entropy-based approaches (e) and (f) (MMEI is also good, but not so informative; actually, the activity due to the people in the upper right part of the scene and in the middle of the crossing street is not shown). It is interesting to note the cylindric high energy zone detected in the middle-right part of the scene. That part represents a rotating billboard, detected as high energetic foreground pattern. This represents a wrong estimation, due to the low number of states with which the HMMs have been trained. Actually, the behavior of the area is 5-modal, with the modes fixed over time. Intuitively, this area should be interpreted as background, with low energy in the activity map, and this situation could be recovered using HMM with 5 number of states (Fig.9.11).

The last sequence, 8 minutes long acquired at 25 fps, represents an indoor environment of a mall[2] (some frames are depicted in Fig. 9.12). One of the purpose of this experiment is to assess the performance of our description over different sequence lengths. The test is divided in two stages: the first in which an initial short part of the sequence is evaluated (48 seconds long, 1/10 of the original one); in the second stage the whole sequence is analyzed.

This is the most difficult indoor sequence analyzed, due to strong noisy effects degrading the data quality, like reflections over the floor and over the lateral windows.

[2] Downloaded from `http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1`

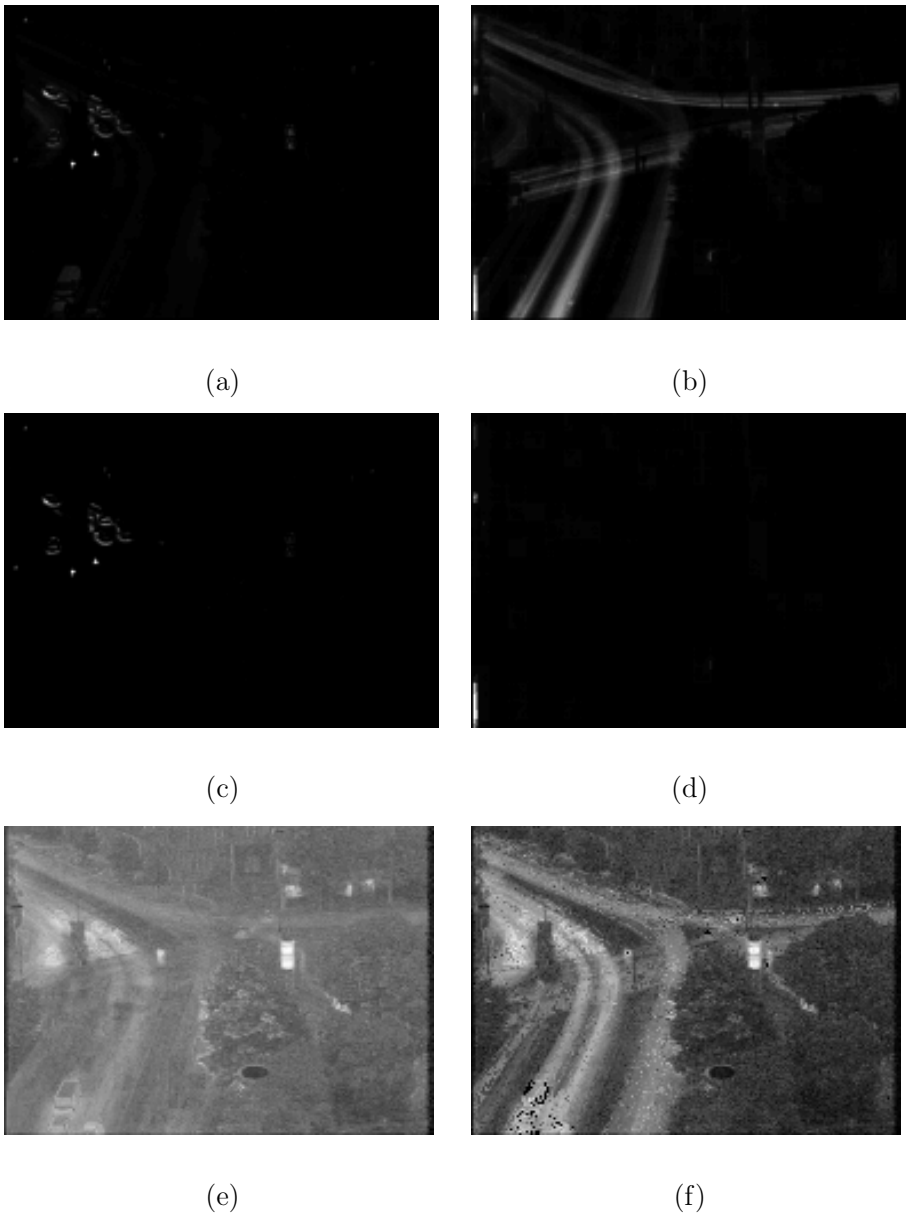
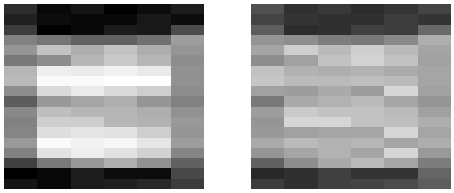


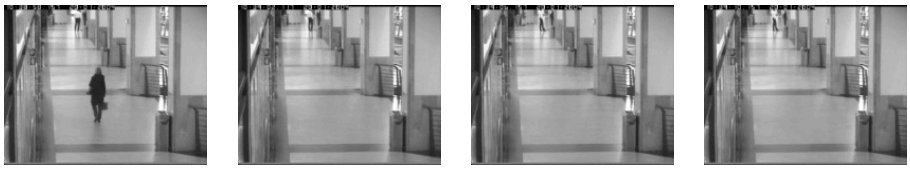
Fig. 9.10. Activity zones resulting from: (a) MEI over reference frame; (b) MEI over consecutive difference (MMEI); (c) Median over reference squared difference (MedReF); (d) Median over consecutive squared difference (MedCDif); (e) Simple entropy; (f) The proposed approach. The whiter the pixels the higher the activity.



(a) (b)

Fig. 9.11. Traffic sequence: the detail of the billboard in Fig. 9.10, whose pixels are trained with HMMs having 3 (a) or 5 states (b). It is possible to evaluate the decreasing of activity detected in (b) due to the correct estimation of the 5 modalities which characterize the billboard behavior.

frame 100 frame 200 frame 300 frame 400



frame 500 frame 600 frame 700 frame 800

Fig. 9.12. Some frames from the traffic sequence.

In this case, the techniques based on differences between frames give poor results in both tests, hence only the method based on the entropy and the proposed approach are presented.

As shown in the previous results, the proposed approach works better compared to the entropy method, individuating the zones in which the foreground activity is located, disregarding the noise. In particular, in the Fig.9.13f1 is possible to clearly detect the left drift of energy, that model the fact that the people enters frequently in the mall. In Fig.9.13e1 this aspect is not so clearly highlighted. Moreover, augmenting the sequence length, the effect of the entropy takes strongly

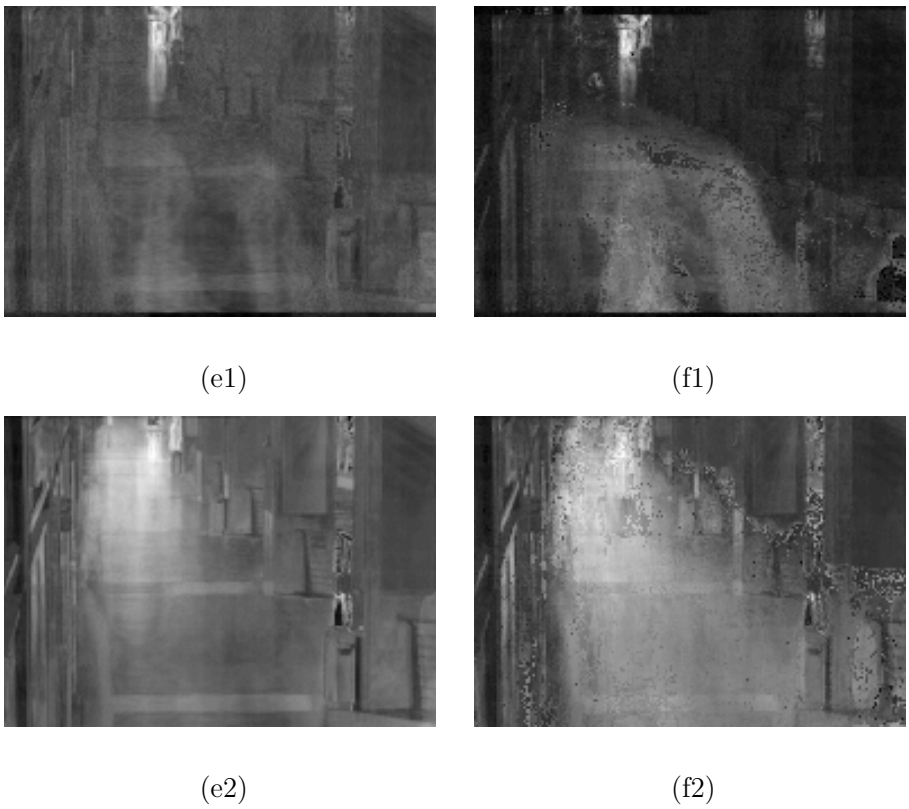


Fig. 9.13. Activity zones resulting from: (e1) Simple entropy and (f1) the proposed approach calculated over the 48 seconds sequence; (e2) simple entropy and (f2) the proposed approach calculated over the entire 8 minutes sequence; the whiter the pixels the higher the activity.

into account the noise due to the reflection on the floor: this results in an activity map that "forgets" the amount of activity present in the bottom part of the hallway. Conversely, our approach is able to represent all the foreground activity as shown in Fig.9.13f2.

Summarizing, the per/pixel descritpion outperforms in general all the tested comparative methods, showing a certain degree of robustness over all the input (long/short sequences with low/high noise levels).

9.3.1 Remarks and possible applications

In this part, we propose a novel inference on the Hidden Markov Model that permits to investigate the behavior of the underlying Markov chain. In details, this analysis gives insight on the presumed long-term behavior of the signal observed by the HMM. We use this analysis in order to differentiate those components of

the signal more related with a stable level of gray, characterizing the background of the sequence, from those related to unstable components, witnesses of foreground activity.

This analysis is performed per-pixel, and the output is visual, providing a computational module of video processing that can be embedded as the bottom layer of the proposed hierarchy of understanding or used as per se framework of low level video understanding.
In the following part, we will see as the present module can be used to perform region level analysis, providing an example of structured hierarchy of understanding.

The fact that the output is visual means that the quality of the results can be directly assessed by the user. Therefore, the user can use the visual feedback to interpret easily the relationship among computational machinery and results obtained. This helps to improve in an intuitive way further analysis;

There are two other ways of employing the information extracted from the proposed approach, which are currently in progress. The first is to use the activity map in order to decide the level of detail of a variable resolution background modelling scheme: the idea is that in those zones where no activities typically occur, a very accurate background analysis is not necessary, and a coarse analysis could be sufficient. The second application can be to use the activity maps to infer the zones of appearance of the foreground with high probability, the so-called source detection problem [137]. The idea is that it is not useful to accurately monitor the zones of the scene where typically no foreground objects are likely to occur.

One constraint of the described approach regards the requirement of a fixed camera. In principle, this condition can be relaxed by performing a pre-registration of the image pixels using an estimate of dominant motion of the scene so that temporal gray-level profiles can be reliably evaluated. Further, such registration could not be critical if small local areas are considered instead of single pixels like in one of the experiments above.

Part III

High level description of an audio video sequence

10

Overview of the part

In this part, the idea of *visual* high level description of a video sequence is given by providing two different methodologies of modelling. These methodologies use the generative paradigm in order to explain automatically a sequence as an ensemble of *regions*. The *region* entity is identified as a set of pixels which share some properties. This kind of representation is highly exploited in the video processing community: most of the tracking algorithms work on compact regions of pixels, using model or aspect based algorithms. In the following, we present two region based descriptions of video sequences based on two different generative model, the HMM and the Time adaptive mixture of Gaussian.
The part is so organized: the first approach (Chap.11) is directly related to the per-pixel framework presented in the previous part. This is an example of how the high level module description is built on top of a per-pixel description, exploiting the hierarchy of understanding discussed in the introduction. Basically, the method is able to provide a visual description of the background of the video sequence, considered as a static object with a chromatic evolution.

The second region description (Chap.12) brings two contributes in the state of the art of the video analysis, because

- is able to detect in an on/line fashion regions of moving pixels that share same visual properties *and* particular audio properties.
- it exploits the use of the synchrony in order to couple audio and video together, translating psychological theories in a computational framework.

11

Region level description of the background

This chapter presents a video modelling technique that produces region representations starting from per-pixel descriptors. In specific, this approach takes as input the output of the low level description module presented in the previous chapter, i.e. a battery of Hidden Markov Models, one for each pixel.

Even in this case, the method is principally aimed to model video sequences coming from a typical video surveillance setting, in which the camera is in a fixed location.

Basically, the method drafts a description about how the background evolves (e.g., periodic chromatic fluctuations possibly due to local/global illumination changes): this is accomplished via a segmented image in which each region corresponds to a compact patch of *background* pixels with similar gray level *and* with similar time-chromatic behavior[1];

The chromo-spatio-temporal segmentation is obtained by clustering the pixel-wise HMMs. To this end, a new similarity measure between HMMs is proposed, able to remove non-stationary components of the sequence. Using this measure and a simple region-growing procedure, a segmentation of the scene is obtained in which the regions show a homogeneous gray-level *and* a similar temporal evolution. In this case, the resulting segmentation is a spatial segmentation of the scene, obtained by using all available information: chromatic (different regions have different gray level values), spatial (each region is connected in the image space), and temporal (each region varies its color homogeneously along time). It may be argued that a similar result could be obtained by a standard spatial segmentation on the first frame or on the average frame, but this method results too limited.

Considering this method jointly with the per pixel description proposed in the previous chapter, it is possible to devise an instance of visual hierarchy of understanding, which scheme is proposed in Fig.11.1 Actually, our approach has two main advantages: first, the spatial knowledge, typically used to obtain standard segmentation, is augmented with temporal information. This is useful to discover, in a region with homogeneous color, additional subregions subjected to periodic

[1] In this chapter, the use of the words *chromo* or *chromatic* concerns the aspects related to the *gray level* values of the image pixels as all the processed sequences are converted in gray-level values. Nevertheless, the extension to the RGB scale is straightforward, and does not raise particular issues to the proposed method.

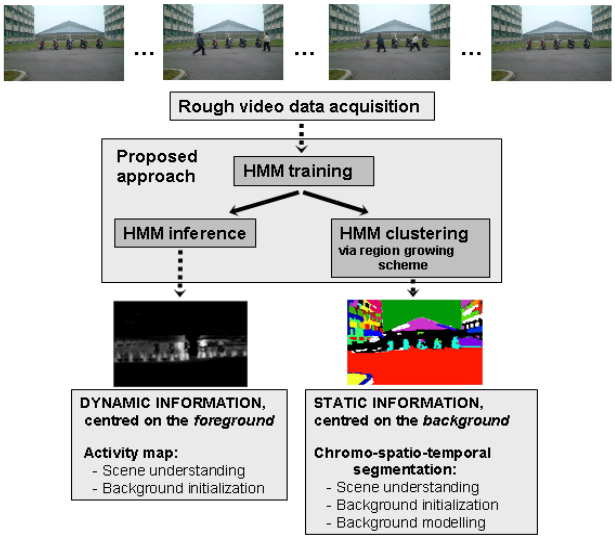


Fig. 11.1. Input and output of the video understanding system, resulting by putting together the method proposed in this section and the one, pixel based, proposed in Part II: starting from the rough pixel data (top), the analysis performed can be represented using two images (bottom). On the left, the pixel level reasoning, represented by the activity map, that indicates the total amount of foreground activity in the scene (dynamic information). On the right, the region level understanding, exploited through a chromo-spatio temporal segmentation, that describes how the chromatic aspect of the background evolve (static information). In the bottom of both the boxes, the related applications to which our analysis can be devoted, described at the end of the previous Part, in Sec. 9.3.1, for what concerns the per-pixel part and in Sec. 11.3.2 for the per-region part.

chromatic fluctuations (caused for example by changes of illumination). Therefore, this segmentation could generally appear as over-segmented, but each region is however meaningful for the addressed task. The second advantage is that moving objects have not to be removed from the sequence (as in the single image segmentation), since this operation is automatically accomplished by the envisioned similarity measure.

The rest of the chapter is organized as follows. In Section 11.1, basic notions regarding the clustering with the Hidden Markov Models are presented. The proposed strategy is then detailed in Section 11.2, and extensive experimental results and a comparative analysis are presented in Section 11.3. Finally, in Section 11.3.2, conclusions are drawn and remarks about the possible applications of the proposed description are presented.

11.1 Methodological issues

In this section, the description of the HMM-based clustering approach is presented. HMMs have not been extensively employed for clustering sequences, literature. Even if some alternative approaches to HMM-based clustering have been proposed (e.g., [18,93]), the typically employed method is the so-called proximity-based approach [85], which uses the HMM modelling to compute distances between sequences; another standard approach is based on pairwise distance matrices (as hierarchical agglomerative) to obtain clustering [96,97,111,136]. The distance between sequences is typically based on the likelihood of the HMM, and could be obtained using several methods (for example, see [1,3,111]).

In more detail, given a set of R sequences $\{\mathbf{O}_1 ... \mathbf{O}_R\}$, the standard approach to clustering trains one HMM $\boldsymbol{\lambda}_i$ for each sequence $\mathbf{O}_i$. Subsequently, a pairwise distance between sequences is defined using these models, in which the key entity is the likelihood L_{ij}, defined as $L_{ij} = P(\mathbf{O}_j|\boldsymbol{\lambda}_i)$. This probability is used to devise a distance (or a similarity) measure between sequences. The simplest example has been proposed in [85], and is defined as

$$D(i,j) = \frac{1}{2}(L_{ij} + L_{ji}) \tag{11.1}$$

A more complex one, inspired from the Kullback-Leibler measure [92] and proposed in [111], is defined as

$$D(i,j) = \frac{1}{2}\left\{\frac{L_{ij} - L_{jj}}{L_{jj}} + \frac{L_{ji} - L_{ii}}{L_{ii}}\right\} \tag{11.2}$$

Once given these distances, any standard pairwise distance-based clustering algorithm could be used, such as those belonging to the hierarchical agglomerative family.

In Section 11.2, we will see how this standard method could be extended in order to deal with spatial segmentation, which represents a particular kind of clustering.

11.2 The proposed approach

Given a battery of previously learned HMM as described in Part II, the kind of information extracted with the proposed approach is a *static* information, that provides knowledge about the structure of the scene. The probabilistic representation of the video sequence is used to obtain a "chromo-spatio-temporal segmentation" of the background. In other words, we want to segment the background of the video sequence in regions showing a homogeneous color and a similar temporal evolution, considering pixel-wise information. In this case, the result is a spatial segmentation, obtained by using all available information: chromatic (different regions have different gray level values), spatial (each region is connected in the image space), and temporal (each region varies its color similarly along time). In this way, spatial knowledge, typically used to obtain spatial segmentation, is augmented with temporal information, allowing a more detailed and informative partitioning. In the

literature, spatio-temporal segmentation typically assume slightly different meanings, depending on the considered context. For instance, in videosurveillance it represents the background/foreground discrimination; in video indexing and retrieval, it provides a compact visual representation, eliminating the redundancy in contiguous frames (in this context it is also called video-segmentation). Nevertheless, our proposed definition of spatio-temporal segmentation is rather intuitive in that it consists in the detection of regions that are characterized by a similar spatio-temporal behavior.

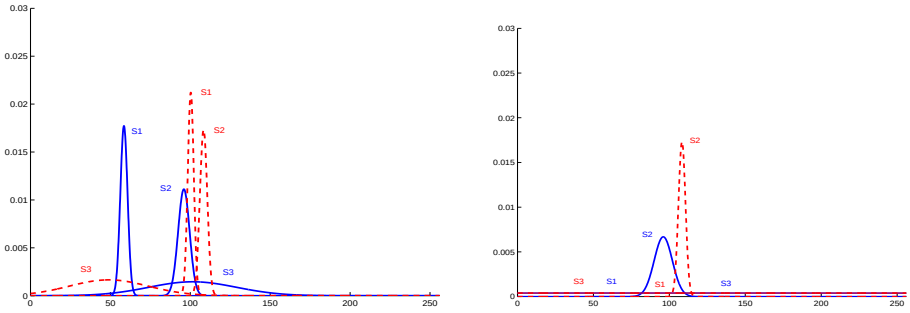
A similar definition was proposed also in [52], where a spatial segmentation was obtained at each time step, using spatial and temporal information, but, in that case, a segmentation was first obtained from the initial frame, and was iteratively used to obtain subsequent partitioning. The proposed HMM representation implies to define a similarity measure, to decide when a group (at least, a couple) of neighboring pixels must be labelled as belonging to the same region. The basic idea is to define a distance between locations on the basis of the distance between the trained Hidden Markov Models: in this way the segmentation process is obtained using a spatial clustering of the HMMs. The similarity measure should exhibit some precise characteristics: two sequences have to be considered similar if they share a comparable main chromatic and temporal behavior, independently from the values assumed by the less important components. By using the measure proposed in eqs. (11.1) or (11.2), we have that the Gaussian of each state contributes in the same way at the computation of the probability, because of the forward-backward procedure. For our goal, however, we need that the Gaussian of each state contributes differently to the probability computation, depending on the "importance" of the corresponding state.

To this end, we have regularized the HMMs' states S_i, for every HMMs, with respect to the related stationary probability $\mathbf{p}_\infty(i)$, which is a quantitative index of the state importance, introduced in Sec. 9.1.1. Actually, $\mathbf{p}_\infty$ indicates the "average" occupation of each state, after the Markov chain has achieved the stationary state [17], hence, it represents the degree of saliency associated to the states. This operation allows to normalize the behavior of the several HMMs so as to allow an effective and reliable comparison between them.

The normalization operation is carried out by operating on the Gaussian parameters of each state, in particular, each original model $\boldsymbol{\lambda}$ is transformed into a new model $\boldsymbol{\lambda}'$, where all components remain unchanged, except variances σ_i of state S_i, for each state $i = 1, .., N$, for all HMMs, *i.e.*:

$$\sigma_i' = \frac{\sigma_i}{\mathbf{p}_\infty(i)} \tag{11.3}$$

The new distance, called $D_{\mathrm{ES}}(k, j)$ (*Enhanced Stationary*), is then computed using equation (11.2) on the modified HMMs $\boldsymbol{\lambda}_k'$. The normalization of the state variances σ_i with respect to the related $\mathbf{p}_\infty(i)$, corresponds to associate the correct significance to the Gaussian $\mathcal{N}(\mu_i, \sigma_i^2)$, and has two beneficial effects: 1) Gaussians of unimportant states are under-graded, reducing their contribution to the probability computation, which results in eliminating moving objects from the video sequence, as they are considered as non-stationary components of the background model; 2) the possibility of match between Gaussians of important states of different models is increased. These concepts are exemplified in Fig. 11.2. Assuming



(a) (b)

Fig. 11.2. Normalized variances: in (a) the original representation of the pdf of each HMM state is shown, for 2 HMM models, λ_1 (in red, dashed) and λ_2 (in blue solid). The variances of each state are $\sigma_1 = 2.2500$ $\sigma_2 = 3.5865$ $\sigma_3 = 27.4072$ for λ_1 and $\sigma_1 = 1.8804$ $\sigma_2 = 2.3042$ $\sigma_3 = 24.6667$ for λ_2. The associated $\mathbf{p}_\infty$ is $< 0.0228, 0.7752, 0.2020 >$ for S_1, S_2, S_3 of λ_1 and $< 0.0001, 0.9998, 0.0001 >$ for S_1, S_2, S_3 of λ_2, respectively. The result of the variance normalization is shown in (b): one could notice the high importance of states S_2 for both λ_1 and λ_2, and the low importance of the other state variances: in the similarity measure computation their contribution results therefore low.

this kind of similarity measure between sequences, the segmentation process can be developed as an ordinary segmentation process of static images. We adopt a simple region growing algorithm: starting the process from some seed-points, we use a threshold θ to estimate when two adjacent sequences are similar using the distance $D_{\mathrm{ES}}(k, j)$.

In our case, the threshold has been heuristically fixed after few experimental trials, and is not a particular critical parameter to set up. The complexity of the segmentation process is $O(nN^2T)$, where N^2T is due to the calculation of the distance among models, and n is the total number of the pixels.

We will see in the experimental section that the modification of the metric in eq. (11.2), with the integration of the chromatic-temporal information of the video-sequence, allows us to obtain a meaningful segmentation.

11.3 Experimental trials and comparative analysis

In this section, some comparative experimental evaluations of the proposed approaches are presented where the strength and the limitations of the proposed description are discussed. Finally, Section 11.3.2 contains some suggestions about the possible use of the information extracted from the video sequence.

11.3.1 Spatio-temporal segmentation

The approach proposed in section 11.2 is tested using two real sequences: the first one regards a person walking in a corridor in which several doors are present. Some frames of the sequence are presented in Fig. 11.3. Looking at the figure

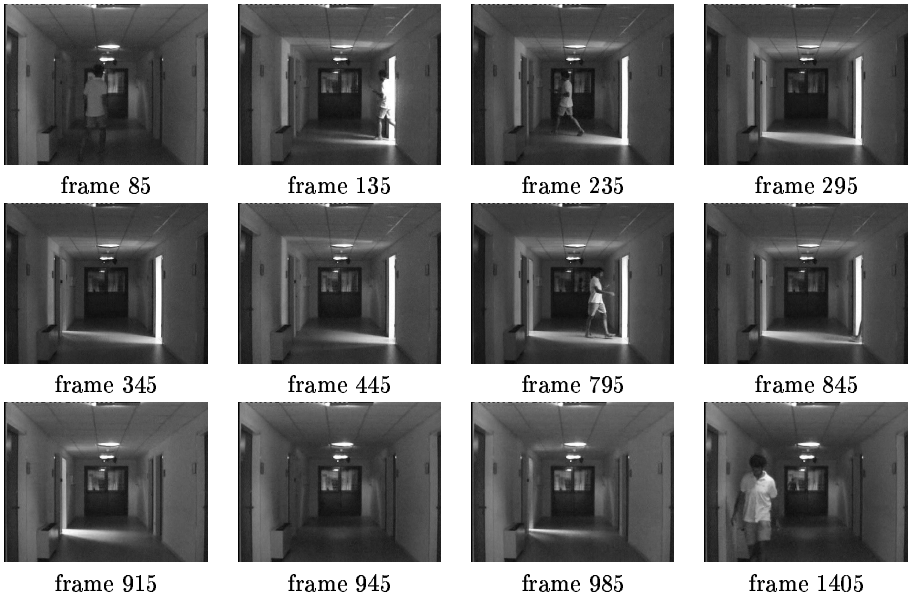


Fig. 11.3. Frames of the first indoor sequence.

(video sequence), you can notice that some doors are opened and closed several times, each one with a random different frequency. The action of opening/closing a door determines a local variation of the illumination, i.e., there are two particular regions of the corridor in which the illumination changes with different frequencies, that it would be reasonable to separate. These different spatial chromatic zones are highlighted in Fig. 11.4: one is on the left part of the corridor, and the other on the right part. This example shows all the potentialities of the proposed approach: the sequential information employed by our approach is essential in order to recover all the different semantic regions of the scene. As an example, let us consider only the median (or the mean) of the sequence, i.e., the image formed by the median (mean) values of each pixel signal, displayed in Fig. 11.5. From these images it is not possible to detect the two semantically different zones of the background. Actually, any spatial segmentation technique applied to these images would segment the zone between the two doors as belonging to the same region. In Fig. 11.6, the segmentation resulting from our approach is displayed. One can easily notice that our approach clearly separates the two zones, labelled as different regions of the scene. In order to assess the gain obtained with the Enhanced Stationary similarity

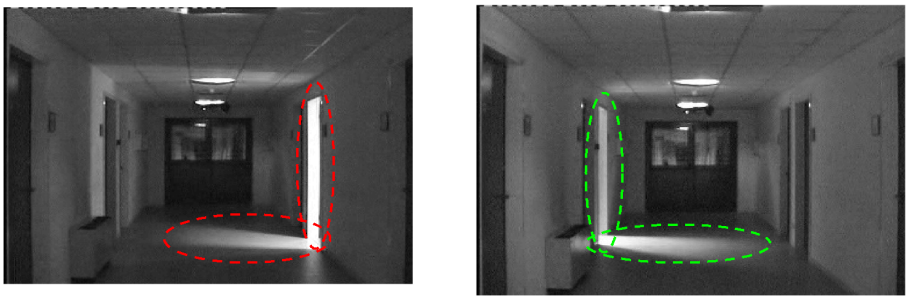


Fig. 11.4. Different spatial chromatic zones.

(a) (b)

Fig. 11.5. (a) Average frame (b) Median frame.

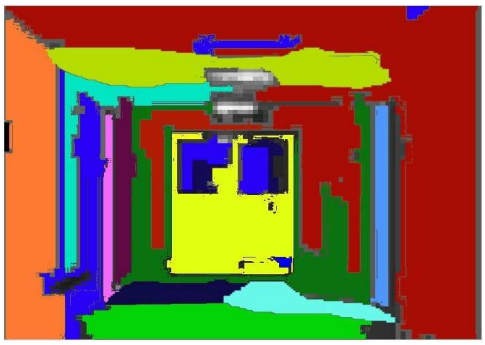


Fig. 11.6. Static information: Spatio-temporal segmentation of the first indoor sequence.

measure D_{ES}, the segmentation of the corridor sequence based on the measure of the equation (11.2) is depicted in Fig. 11.7. It is evident that the noise affecting

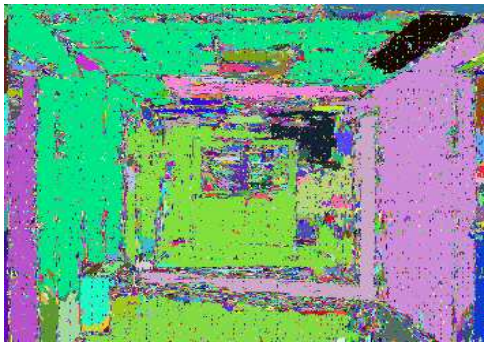


Fig. 11.7. Segmentation of the first indoor sequence using the similarity measure without the HMM states' normalization.

the video sequence and the presence of foreground produce a very noisy and heavy over-segmentation, whereas our approach is able to manage foreground objects and noise.

The second sequence used for testing is obtained from [140] and regards the monitoring of an indoor environment with one moving object. Some of the frames of the sequence are presented in Part II, Sec. 9.3, in Fig. 9.5, showing a sudden not uniformly distributed change of the illumination. Such non uniform luminosity change could drastically affect the comprehension of the sequence, and only a method that uses spatio-temporal information can be able to correctly identify the semantically separated regions. To slow down the computational effort, we partitioned the field of view in a grid with circular Gaussian filters of 5×5 pixels, and at each time step each filter provides one single weighted value (this improvement has drastically reduced the computation time). The result of the segmentation, after the HMM training is reported in Fig. 11.8: the segmentation is highly informative in that the foreground does not appear in the resulting segmentation, and the change of illumination does not influence the spatial chromatic structure of the scene. Actually, areas of different chromaticity (the floor, portions of the wall) remain separated despite the light reduction narrows down the chromatic difference among them.

Another testing sequence regards an outdoor environment where two persons are closing and come back. A few frames of the sequence are presented in Part II, Sec. 9.3, Fig. 9.7.

Looking at the result (Fig. 11.9), one could notice that the segmentation is clear, expressive, and quite accurate: zones with similar gray level and similar chromatic behavior (the road, the sky, and the motorbikes) are represented as single regions. Other zones characterized by a different chromatic behavior (the two buildings and part of the pyramid) are over-segmented. It is worthwhile to notice

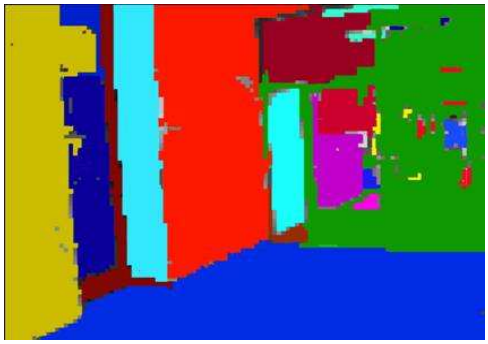


Fig. 11.8. Static information: spatio-temporal segmentation of the second indoor sequence.

Fig. 11.9. Information extracted from the outdoor sequence using the proposed approach: Static information (Spatio-temporal segmentation).

that this segmentation is obtained by processing the whole sequence, without any need to remove the moving objects, in that they are naturally removed by the procedure used to compute the Enhanced Stationary distance D_{ES}.

The second and the third sequence of this section are the most complex one, the same considered in the previous part (Part II, Sec. 9.3, Fig. 9.9 and Fig. 9.12), in which our current region description shows its limits. These limits will draw the directions of our research. In the first sequence, a traffic situation over a square is monitored via a fixed camera. The chromo-spatio-temporal segmentation, in this case, is highly over-segmented. This is due to the intrinsic irregularity with which the static zone evolve, and to the difficulty to clearly distinguish what is the background and what the foreground (some blocked cars could be detected as background). One of the possible solutions is to restrict the static analysis to the zones where the activity map gives low values.

The last sequence, 8 minutes long acquired at 25 fps, represents an indoor environment of a mall.

Even in this case, as performed in the previous part, the test is divided in two stages: the first in which an initial short part of the sequence is evaluated (48 seconds long, 1/10 of the original one); in the second stage the whole sequence is analyzed.
In both processing cases, the outcome of the static analysis results over-segmented, as expected. Since the results on the short sequence are not significative, only the results related to the longer sequence are shown in Fig. 11.10.

Fig. 11.10. Information extracted from the whole outdoor sequence using the proposed approach: static information.

Looking at this figure, it is possible to reason about what are the biggest areas whose chromatic behavior is similar in time, so as to detect the most "stable" scene areas. In particular, it is possible to detect stable zones in proximity of the lateral columns, and in some parts of the floor, while the area corresponding to the left wall, with several glass windows, is in general over-segmented.

11.3.2 Remarks and possible applications

The region level description of the sequence through the chromo-spatio-temporal segmentation is useful to detect zones in which the chromo-temporal trend is similar; this means that in high variable environments the segmentations are over-fragmented. Considering the feasible applications of the chromo spatio temporal segmentation, this cannot be stated as a drawback, as shown in [38, 39].
In [39], this segmentation has been used in order to initialize an integrated pixel- and region-based approach to background modelling, further used in a tracking application [38]. In such case, indoor tracking is performed, where the background

appearance is subjected to abrupt change in the illumination level . This background model uses information derived from a spatial segmentation of the scene in order to modulate the response of a standard pixel-level background modelling scheme [138], increasing the robustness against local non uniform illumination changes.

As future perspectives, the segmentation scheme could be highly improved, introducing the concept of "main" HMM able to model the overall chromatic behavior of the region considered. In that sense, better result are expected; actually, the incremental scheme of segmentation, or the one centroid-driven, appear rough even if effective.

The computational complexity of the approach could also be improved, by applying approximate inferences strategy and on-line learning techniques.

12

Region level description of audio-video data

In general, almost all of human activity recognition systems work mainly at visual level only, but other information modalities can easily be available (e.g., audio), and used as complementary information to discover and explain interesting "activity patterns" in a scene. Computer Vision researchers devoted their efforts towards AV data fusion only in the last few years (see Sec. 12.1 for a critical review of the related literature).

This chapter explores this research trend, providing a novel strategy for activity analysis, able to integrate audio and video information at the feature level. By adopting this strategy, it results possible to product region level descriptions that characterize and distinguish human activities.

The video information is provided by a classical per pixel BG modelling module, based on a time-adaptive per-pixel mixture of Gaussians process [138], able to model the background of a static scene while highlighting the foreground.

This module is enhanced with a novelty detection module aimed at detecting new objects appearing in the scene, thus allowing to discriminate different FG entities. The monaural audio information is acquired by introducing the idea of FG audio events, i.e. unexpected audio patterns, that are detected automatically by modelling in an adaptive way the audio background. The adaptive video and audio modules work on-line and in parallel, so that, at each time step, they can detect separate audio and visual FG patterns in the scene.

On top of the unimodal processing stages, there is the core module, aimed at establishing a binding of audio and visual modalities, so that correlated audio and video cues can be aggregated leading to the detection of *audio-visual* (AV) events. This binding process is based on the notion of *synchrony* between the unimodal FG events occurring in the scene. This choice is motivated from the fact that the simultaneity is one of the most powerful cues available for determining whether two events define a single or multiple objects, as stated in early studies about audio-visual synchrony coming from the cognitive science [46]. Moreover, psychophysical studies have shown that the human attention focuses preferably on sensory information perceived coupled in time, suppressing those cues that are not [109]; particular importance was devoted to the study of situations in which the inputs arrive through two different sensory modalities (such as sight and sound) [139].

In our approach, the binding process is realized by building and on-line updating the so-called *Audio-Video Concurrence* (AVC) matrix. Such matrix permits to detect significant non-overlapped joint AV events and represents a clear and meaningful description of them. Such representation, built on line and without the need of training sequences, is so effective to allow to accurately discriminate between different AV events using simple classification or clustering techniques, like K-Nearest Neighbors (KNN) [53].

In summary, the approach introduces several concepts related to the multimodal scene analysis, discussing the involved problems, showing potentialities and possible future directions of the research. The major contributions of this work can be summarized in the following. We introduce: 1) an audio BG/FG modelling system coupled with a related video BG/FG module, working on line in an adaptive way and able to detect separate audio and visual foreground at each time instant; 2) a method for integrating audio and video information in order to discover multisensory FG patterns, potentially increasing the capabilities of surveillance systems; 3) a multimodal and multidimensional feature (i.e., the AVC matrix), based on the concurrency between audio and video patterns, which is proved to be expressive of the events occurring in an observed scene; 4) an audio-visual fusion criteria embedded in a probabilistic framework working on-line, without the necessity of training sequences.

The rest of the chapter is organized as follows. Section 12.1 reviews the AV fusion literature, clearly indicating the main differences between the proposed approach and the state of the art. In Section 12.2, the whole strategy is detailed, and experimental results are reported in Section 12.3. Finally, in Section 12.4, conclusions are drawn and future perspectives are envisaged.

12.1 State of the art of the audio-visual analysis

In the context of audio-visual data fusion it is possible to individuate two principal research fields: the audio-visual association, in which audio data are spatialized using a microphone array (mainly devoted to tracking tasks), and the more general audio-visual analysis, in which the audio signal is acquired using only one microphone.

In the former, the typical scenario is a known environment (mostly indoor), augmented with fixed cameras and acoustic sensors. Here, a multimodal system locates moving sound sources (persons, robots) by utilizing the audio signal time delays among the microphones and the spatial trajectories performed by the objects [31, 162]. In [31], the tackled situation regards a conference room equipped with 32 omnidirectional microphones and two stereo cameras, in which a multiobject 3D tracking is performed. With the same environmental configuration, in [154], an audio source separation application is proposed. In another application [164], the audio information (constituted by footstep sounds) is used to distinguish a walking person among other moving objects by using a framework based on dynamic Bayes nets.

Other approaches based on the learning and inference of a graphical model can be found in [12], in which person tracking in an indoor environment is performed using video and audio cues derived from a camera and two microphones,

respectively. In [105], a 2-layer HMM framework is used to model predetermined individual and group multimodal meeting actions.

The second class of approaches employs only one microphone. In this case, audio spatialization is no more explicitly recoverable, so the audio-visual binding must rely on other techniques. A well-known technique is the canonical correlation analysis (CCA) [70], a statistical way of measuring linear relationships between two multidimensional random variables. In the audio-visual context, the random variables are represented by the audio and video signals, i.e., spectral bands for the audio space and the image pixels for the video one. CCA extracts a linear combination of a subset of pixels and a subset of bands that are maximally correlated. The fundamental problem of the CCA-based approaches is the need of a large amount of data, that consequently leads to off-line applications in which the visual regions that emit sounds are constrained to being well localized in the scene. Therefore, this method well behaves in the case of strongly supervised applications. A CCA-based approach is represented by FaceSync [135], an algorithm that measures the degree of synchronization between the video image of a face and the associated audio signal. A solution to the demand of huge amount of data is proposed in [163], in which a presumed sparsity of the audio-visual events is exploited.

Another class of inter-modal relationship detection is based on the maximization of the mutual information (MMI) between two sets of multivariate random variables. Audio-visual systems based on mutual information maximization is proposed in [58] and [78]. In [45], an information theoretic approach to modelling the audio and video signals using Markov chains is proposed, in which the audio and video joint densities are estimated using a set of training sequences. The methods based on the MMI inherit the potentialities and the drawbacks of the CCA approaches: in [88], it has been shown the equivalence between CCA and MMI under certain hypotheses on the underlying distributions.

The explicit detection of synchrony between audio and video represents another way to detect cross-modality relations, even if not so deeply investigated from the computer vision community for what concerns localization aims. For example, in [160], audio and visual patterns are used to train an incrementally structured Hidden Markov Model in order to detect unusual AV events. Here, the audio pattern are formed by Mel-Frequency Cepstral coefficients from the raw audio signal, and the video patterns are composed by motion and color features from moving blocks of each frame. The joining of the audio and visual features is performed by simply concatenating both patterns.

Another research field in which the audio-video analysis is largely exploited is the video retrieval by content, in which the objects to be analyzed are typically entertainment sequences (movies, commercials, news, etc.). The ultimate goal is to enable users to retrieve the desired video clip among massive amounts of heterogeneous visual data in a semantically meaningful and efficient manner. In this field, high-level concepts as video object and events (spatio-temporal relations among objects) are exploited. The heterogeneity of the sequences considered urges the use of general high-level approaches, further heavily relying on automatic video annotation techniques [118, 119].

The proposed approach is different with respect to those of the state of the art presented above from both what concerns the complexity of the considered data, and the basic idea underlying the analysis performed. In our setting, audio-video sequences come from a video surveillance context, in which the camera is still, apart small movements, and the audio comes directly from the scene being monitored, without any kind of control. Then, regarding to the nature of the proposed approach, we studied an intuitive and accurate audio-visual fusion criterion that do not require the formulation of any complex statistical model describing the relationships between audio and video information, working on line and without the need of training sequences (like in [45], for example). In particular, the proposed method is heavily based on the concept of synchrony, a well motivated basic principle derived from psycho-physiological research, also able to handle the localization issue.

12.2 The proposed method

12.2.1 Overview

The system is composed by several stages, starting with two separate audio and visual background modelling and foreground detection modules, as shown in Fig. 12.1.

For the visual channel, the model operates at two levels. The first is a typical time-adaptive per-pixel mixture of Gaussians generative model [138], able to identify the visual FG present in a scene. The second model is a region level description that works on the FG color histogram, and is able to detect different novel FG events. Despite the simple representation, this mixture model is able to characterize the appearance of FG data, and to discriminate different FG objects.

Concerning the audio processing scheme, the novel concept of *audio* BG modelling[1] is introduced, capable to detect unexpected audio activities.

A multi-band frequency analysis is first carried out to characterize the monaural audio signal by extracting characteristic features from a parametric estimation of the power spectral density. The audio BG is then obtained by modelling such features related to each frequency band using an adaptive mixture of Gaussians, so allowing to detect, at each time step, a novel audio signal (e.g., a door that is closed, a ringing phone bell, etc., see Fig.12.1). These modules work on-line, in parallel, and the output are the separate audio and video FG occurring in a scene at each time step.

Audio-visual association is subsequently developed by constructing the so-called *Audio-Video Concurrence (AVC)* matrix, which encodes the degree of simultaneity of the audio and the video FG patterns.

As assessed by psychophysical studies (see in the opening of this chapter), we assume that visual and audio FG that occur "simultaneously" are likely causally correlated. In particular, the FG contributes of each modality are collected at each time step, and then combined in the AVC matrix, whose i, j entry represents the importance of the audio FG energy localized in the i-th audio sub-band *and* the FG

[1] A first-stage version appeared in [40].

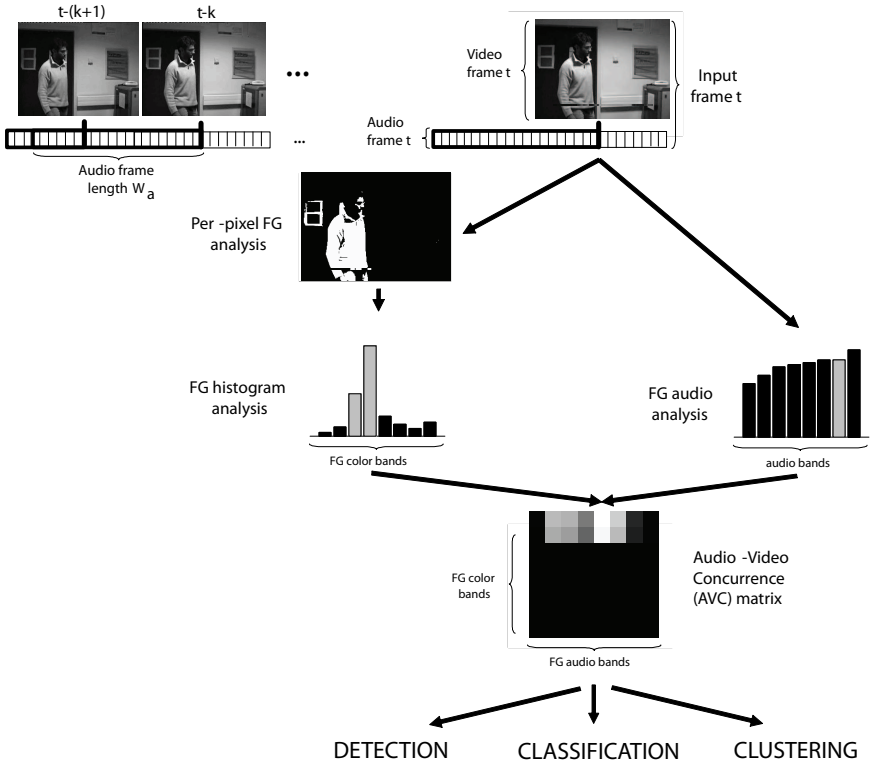


Fig. 12.1. Outline of the proposed system.

novel appearance of a particular color range belonging to the j-th FG histogram bin. This association is able to assess how much the inter-modal concurrency holds along time, permitting to individuate the most salient and permanent audio-visual bindings. The resulting AVC matrix is therefore a multidimensional feature that, at each time step, summarizes and describes the audio-visual activity being occurring in the scene (see Fig.12.1).

The high expressivity of such feature allows to effectively characterize and discriminate between such events, outperforming clustering and classification performances obtained using the individual modalities, as we will see in the following.

The reminder of the section will give all the details of the proposed approach, starting from the basic time-adaptive mixture of Gaussians (Sect. 12.2.2), and subsequently explaining how the video and the audio channels are modelled (Sect. 12.2.3 and 12.2.4). Then, Sect. 12.2.5 provides details about the audio-video fusion, and Sect. 12.2.6 and 12.2.7 contain the description of how to perform audio-visual event detection and discrimination, respectively.

12.2.2 The time-adaptive mixture of Gaussians method

The Time-Adaptive mixture of Gaussians method is a probabilistic tool able to discover the deviance of a signal from the expected behavior in an on-line fashion.

The signal is modelled using a dynamic mixture of Gaussians, whose weights are updated iteratively. Looking at the weights it is possible to classify new observations as "expected" or "unexpected". A typical video application is the well-known BG modelling scheme proposed in [138]. Since this tool is present in different parts of our system, it is briefly summarized in this section.

In the general method [138], the temporal signal is modelled with a time-adaptive mixture of Gaussians with R components. The probability to observe the value $z^{(t)}$, at time t, is given by:

$$P(z^{(t)}) = \sum_{r=1}^{R} w_r^{(t)} \mathcal{N}\left(z^{(t)} | \mu_r^{(t)}, \sigma_r^{(t)}\right) \tag{12.1}$$

where $w_r^{(t)}$, $\mu_r^{(t)}$ and $\sigma_r^{(t)}$ are the mixing coefficients, the mean, and the standard deviation, respectively, of the r-th Gaussian of the mixture associated to the signal at time t. The Gaussians are ranked in descending order using the w/σ value: the most ranked components represent the "expected" signal, or the background.

At each time instant, the Gaussians are evaluated in descending order to find the first matching with the observation acquired (a match occurs if the value falls within 2.5σ of the mean of the component). If no match occurs, the last ranked component (the least important) is discarded and replaced with a new Gaussian with mean equal to the current value, high variance, and low mixing coefficient. If r_{hit} is the matched Gaussian component, the value $z^{(t)}$ is labelled as FG if

$$\sum_{r=1}^{r_{hit}} w_r^{(t)} \; > \; T \tag{12.2}$$

where T is a threshold representing the minimum portion of the data that supports the "expected behavior". We call this test as *FG test*, that is positive if the value is labelled as FG ($z^{(t)} \in FG$), negative viceversa.

The equations that drive the evolution of the mixture parameters are the following :

$$w_r^{(t)} = (1-\alpha) w_r^{(t-1)} + \alpha M^{(t)}, 1 \leq r \leq R, \tag{12.3}$$

where $M^{(t)}$ is 1 for the matched Gaussian (indexed by r_{hit}), and 0 for the others; the weights are re-normalized at each iteration. Typically, the adaptive rate coefficient α remains fixed along time. The μ and σ of the matched Gaussian component are updated:

$$\mu_{r_{hit}}^{(t)} = (1-\rho)\mu_{r_{hit}}^{(t-1)} + \rho z^{(t)} \tag{12.4}$$

$$\sigma_{r_{hit}}^{2\,(t)} = (1-\rho)\sigma_{r_{hit}}^{2\,(t-1)} + \rho \left(z^{(t)} - \mu_{r_{hit}}^{(t)}\right)^T \left(z^{(t)} - \mu_{r_{hit}}^{(t)}\right) \tag{12.5}$$

where $\rho = \alpha \mathcal{N}\left(z^{(t)} | \mu_{r_{hit}}^{(t)}, \sigma_{r_{hit}}^{(t)}\right)$. The other parameters remain unchanged. It is worth to notice that the higher the adaptive rate α, the faster the model is "adapted" to signal changes.

This approach can be considered as an approximated version of the Expectation Maximization algorithm, in which the learning of a mixture of Gaussians occurs in an on-line fashion. We remind that the hidden variable in that case is the

class membership variable, indicating which Gaussian component generates the observed data.

In this case, the approximation stands in the fact here we do not marginalize over the hidden variable; instead, we take only into account for the most probable component, in a likelihood sense.

In this sense, this approach can be considered more precisely as an on line version of the k-means, where the parameters of the Gaussians are updated at each observation.

12.2.3 Visual analysis

This section describes the visual module of the proposed system, which is able to detect atypical *visual* activity patterns. The designed method is composed by two parts: a standard per-pixel FG detection module, and a histogram-based novelty detection module (see Fig. 12.1).

The former is a standard realization of the model explained in (Sec. 12.2.2), where each pixel signal $z_n^{(t)}$ is independently described by a TAPPMOG model: an unexpected valued pixel represents the visual per-pixel FG, $z_n^{(t)} \in FG$. Note that all mixtures' parameters are updated with a common fixed learning coefficient $\tilde{\alpha}$, and using a fixed value T as FG detection threshold, which are the same for audio and video channels.

The second module is a novelty detection system, able to detect when new objects appear in the scene. This part is fundamental in the proposed method, since the audio and visual pairing can be assessed if and only if a visual object occurs in the scene (and remains FG) together with an audio FG signal: it is therefore fundamental to detect new objects appearing in the scene.

To this end, the idea is to compute at each time step the gray level histogram of the sole FG pixels, which we called *Video Foreground Histogram* (VFGH). Each bin of the histogram, at time t, is denoted by $v_j^{(t)}$, where j varies from 1 to J, the number of bins. In practice $v_j^{(t)}$ represents the quantity of pixels of the FG, present in a scene at time t, with intensity values falling in the gray level range j. Obviously, the accuracy of the description depends on the total number of bins J.

Then, we associate a TAPPMOG to each bin of the VFGH, looking for variations of the bins' value. When the number of foreground pixels significantly change, also changes obviously the related FG histogram, and an occurring novel visual event can be inferred.

The probability to observe the value $v_j^{(t)}$, at time t, is modelled using a TAPPMOG:

$$P(v_j^{(t)}) = \sum_{r=1}^{R} w_{(V,r,j)}^{(t)} \mathcal{N}\left(v_j^{(t)} | \mu_{r,j}^{(t)}, \sigma_{r,j}^{(t)}\right) \tag{12.6}$$

Defining u the matched Gaussian component, we can label the j-th bin of the VFGH at time step t as *visual FG value*, if

$$\sum_{r=1}^{u} w_{(V,r,j)}^{(t)} > T \tag{12.7}$$

This scheme permits to detect both appearing and disappearing objects (an object is appearing in the scene when bins suddenly increase their values, disappearing when bins values decrease). Actually, we are interested only in appearing objects, since this represents the sole case in which audio-visual synchrony is significant (a disappearing object, like a person that exits from the scene, should not be considered as it does not belong to the scene anymore). To this end, we disregard visual FG values deriving from negative variations of the foreground histogram bins, considering only the positive variations.

We are aware that the characterization based on the histogram leaves some ambiguities (e.g., two equally colored objects are not distinguishable, even if the impact of this problem may be weakened by refining the number of bins), but this representation has the appealing characteristic of being invariant to spatial localization of the FG (as in other audio-video analysis approaches). This characteristic is not recoverable by monitoring only the FG pixels directly[2].

12.2.4 Audio analysis

The audio BG modelling module aims at extracting information from an audio signal acquired by a *single* microphone. In the literature, several taxonomies can be drawn, in order to categorize the huge amount of approaches present. The "computational auditory scene analysis" methods (CASA) [23] translate psychoacoustics theories to automatically separate and classify sounds present in a specific environment using signal processing techniques. The "computational auditory scene recognition" (CASR) approaches [37,115] are aimed at environment interpretation instead of analyzing the different sound sources. More related to the statistical pattern recognition literature, a third class of approaches tried to fuse "blind" statistical knowledge with biologically-driven representations derived from the two previous fields, performing audio classification and segmentation tasks [161], and blind source separation [75, 128].

The approach presented in this chapter can be cast in this last category. Roughly speaking, a multi-band spectral analysis of the audio signal at video frame rate is performed, extracting energy features from I frequency sub-bands, $a_1, a_2, \ldots, a_I$. More specifically, we subdivide the audio signal in overlapped temporal windows of fixed length W_a, in which each temporal window ends at the instant corresponding to the t-th video frame[3] (see Fig.12.1).

For each window, a parametric estimation of the power spectral density with the Yule-Walker Auto Regressive method [103] is performed: this method has been used in several time series modelling approaches [2,25], showing good performances whatever audio window length is used. From this process, the energy samples (measured in decibel, dB) $\{X^{(t)}(f_w)\},\ w = 1, \ldots, W$ are obtained, where f_w is the frequency expressed in Hz, and the maximal frequency is $f_W = F_s/2$, with F_s the sampling rate.

[2] Actually, this is a simple way of detecting novel FG without resorting to more sophisticated tracking approaches based on histograms [104], that will be subject of future work.

[3] In the following, we use a temporal indexing leaded by the *video* frame rate; therefore, the $t-$th time step of the analysis is relative to the $t-$th video frame.

Subsequently, we introduce the *Subband Energy Amount* (SEA), representing the histogram of the spectral energy, where each bin of the histogram, at time t, is denoted with $a_i^{(t)}, \quad 1 \leq i \leq I$. The SEA features have been chosen for their capability to discriminate between different sound events [40, 115], and because they can be easily computed at an elevate temporal rate, permitting to discover unexpected audio behaviors for each channel at each time step.

Regarding the modelling of the time evolution of the SEA features, we assume that the energy during time at different frequency bands can transport independent information, as stated in [128]. Therefore, we instantiate one independent time-adaptive mixture of Gaussians (Sec. 12.2.2) for each SEA channel. That is, the probability to observe the value $a_i^{(t)}$, at time t, is modelled using a TAPPMOG:

$$P(a_i^{(t)}) = \sum_{r=1}^{R} w_{(A,r,i)}^{(t)} \mathcal{N}\left(a_i^{(t)} | \mu_{r,i}^{(t)}, \sigma_{r,i}^{(t)}\right) \tag{12.8}$$

Let q be the Gaussian component matched when new observation arrives, we can identify the SEA band value a_i as *audio FG value*, if

$$\sum_{r=1}^{q} w_{(A,r,i)}^{(t)} \; > \; T, \tag{12.9}$$

where the threshold T and the audio learning rate $\tilde{\alpha}$ are fixed and common parameters, equal to those used for the video channel.

12.2.5 The Audio-Visual fusion

The audio and visual channels are now partitioned in different independent subspaces, the audio sub-bands $a_1, a_2, \ldots, a_I$, and the video FG histogram bins $v_1, v_2, \ldots, v_J$, respectively, in which independent unimodal FG values may occur. The leading idea is to find causal relations among each possible couple of audio and video bins at each time step t, with the condition that both considered subspaces bring FG information.

Without loss of generality, let's consider the $i-$th audio subspace and $j-$th video subspace; more specifically, let $a_i^{(t)}$ be the energy of the audio signal relative to the $i-$th subband at time step t and $v_j^{(t)}$ the amount of FG pixels at time step t in the scene (that correspond to the $j-$th FG histogram bin).

Technically, we define a general *audio FG pattern* $A_i^{(t_{init}^A, t_{end}^A)}$ related to band i as the time interval when band a_i is foreground:

$$A_i^{(t_{init}^A, t_{end}^A)} = \left[a_i^{(t_{init}^A)}, a_i^{(t_{init}^A+1)}, \ldots, a_i^{(t)}, \ldots, a_i^{(t_{end}^A)}\right] \tag{12.10}$$

where the interval $t_{init}^A, \ldots, t, \ldots, t_{end}^A$ is such that $a_i^{(t)} \in \text{FG}, \forall t \in [t_{init}^A, t_{end}^A]$

In a very similar way we can define the *video FG event* $V_j^{(t_{init}^V, t_{end}^V)}$, representing the interval time when the video foreground histogram band v_j is labelled as FG.

Given two FG patterns, we introduce the *Potential Relation Interval* as the time interval containing the possible overlapping of the audio and video patterns $A_i^{(t_{init}^A, t_{end}^A)}$ and $V_j^{(t_{init}^V, t_{end}^V)}$. Defining

$$t^{AV}_{init} = \max(t^{A}_{init}, t^{V}_{init}) \qquad t^{AV}_{end} = \min(t^{A}_{end}, t^{V}_{end})$$

then the Potential Relation Interval could be described as

$$PRI^{(t^{AV}_{init}, t^{AV}_{end})}_{i,j} = [t^{AV}_{init}, t^{AV}_{end}] \tag{12.11}$$

where $t^{AV}_{end} > t^{AV}_{init}$. The $PRI^{(t^{AV}_{init}, t^{AV}_{end})}_{i,j}$ represents the time interval in which there is a concurrence between audio and video patterns, i.e., when the audio and video bands are synchronously FG.

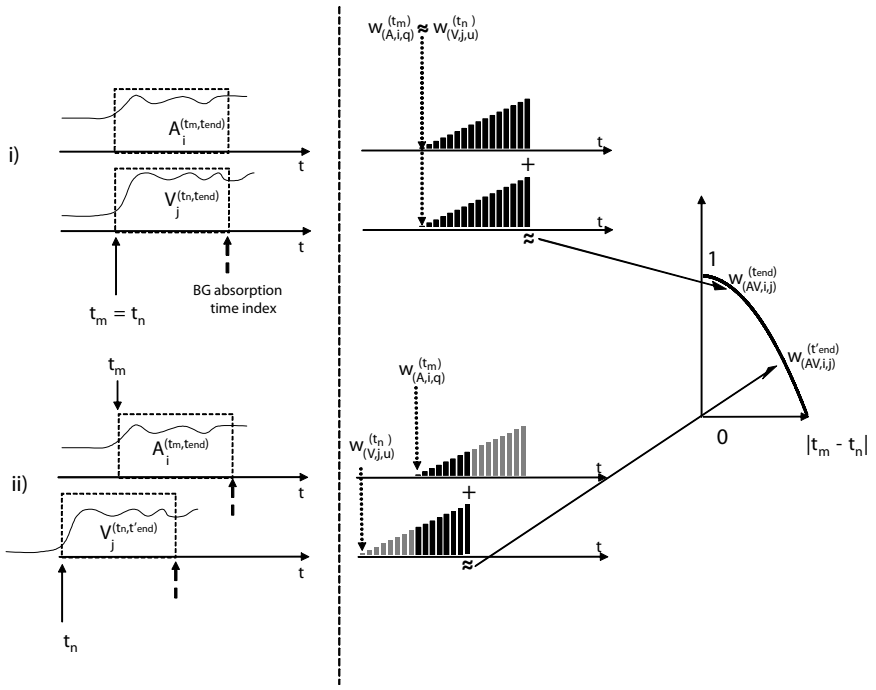


Fig. 12.2. Graphical definition of the AV coupling weight: this value is able to distinguish different degrees of multimodal synchrony; on the left column, two cases of audio and video foreground patterns (i) strongly synchronous (ii) loosely synchronous. On the right column (on the left), the FG mixing coefficients of the Gaussian components that model the FG patterns. On the right, the behavior of the AV coupling weight: maximum when a complete overlapping of the FG patterns is present, decreasing when the synchrony degree of the FG patterns diminishes.

Now we could define AV coupling weight $w^{(t)}_{AV}(i,j)$ as

$$w^{(t)}_{AV}(i,j) = \begin{cases} \dfrac{w^{(t)}_{(A,i,q)} + w^{(t)}_{(V,j,u)}}{2} & \text{if } t \in PRI^{(t^{AV}_{init}, t^{AV}_{end})}_{i,j} \\ 0 & \text{otherwise} \end{cases} \tag{12.12}$$

where $w^{(t)}_{(A,i,q)}$ ($w^{(t)}_{(V,j,u)}$) is the weight of the Gaussian matched by the audio (video) band $a_i(t)$ ($v_j(t)$) in the audio (video) time adaptive mixture of Gaussians model.

If the information carried out by the audio channel i is synchronous with respect to the information carried out by the video channel j, then the patterns are correlated, and the AV coupling weight permits to measure the strength of the AV association. A synthetical example is shown in Fig. 12.2: on the left column, we see two cases of audio and video foreground patterns that (i) are strongly synchronous and (ii) are loosely synchronous. The area placed in the dashed-line box indicates a FG pattern, whose initial instant is indicated with a solid arrow, and the relative BG absorption is pointed out with a dashed arrow. On the right column (on the left), the mixing coefficients of the Gaussian components that model the FG patterns are indicated. One can notice that these coefficients (the first of them indicated by a dotted line) are proportional with the time spent by the related Gaussian components to model the FG values: the FG mixing coefficients increase (if the pattern is always modelled by the same component) until the FG test (explained in Sec. 12.2.2) is negative, then, such value is labelled as background. The AV coupling weights are built using the unimodal FG mixing coefficients in the related Potential Relation Interval (indicated as black bars in the picture). As one can observe (on the right), this value is maximum when a complete overlapping of the FG patterns is present, while a strong decreasing appears when the synchrony degree of the FG patterns diminishes.

Now, we are ready to introduce the main feature, namely the *Audio-Video Concurrence (AVC)* matrix. This matrix, of size $I \times J$, is able to accurately describe the audio-visual history until time t: the i, j entry, at time t, is defined as:

$$AVC^{(t)}(i,j) = \sum_{t'=0}^{t} w_{AV}^{(t')}(i,j) \tag{12.13}$$

At time $t = 0$, this matrix is empty. The AVC feature is computed on line, describes the audio-video synchrony from time 0 to t, and represents the core of the proposed approach. We will see in the next sections that AV event detection is directly derived from this feature, as well as a discriminative description of the AV events. Moreover, this AVC matrix, used in a surveillance context, permits spatial localization of the audio foreground [41].

12.2.6 Audio-visual event detection

The segmentation of the whole video sequence in audio-visual events can be straightforwardly performed starting from the AVC matrix. Before describing how to segment it, let us define an *Audio Video Event (AVE)*: it occurs when a FG audio and a FG video are synchronously present in the scene. This can be detected by looking at the AVC matrix: if a there is synchrony in the scene events, for some audio band a_i and video band v_j, the AV coupling weight is non zero. Therefore, an AVE is detected in the time interval $[t_{init}^{AV},\ t_{end}^{AV}]$ if the following conditions hold contemporaneously [4]:

1. $AVC^{(t_{init}^{AV}-2)} - AVC^{(t_{init}^{AV}-1)} = \mathbf{0}$ (no synchrony before t_{init}^{AV})

[4] Note that the following operations are computed among matrices: in particular, the relation of $\neq$ is valid if it holds for at least one matrix element i, j.

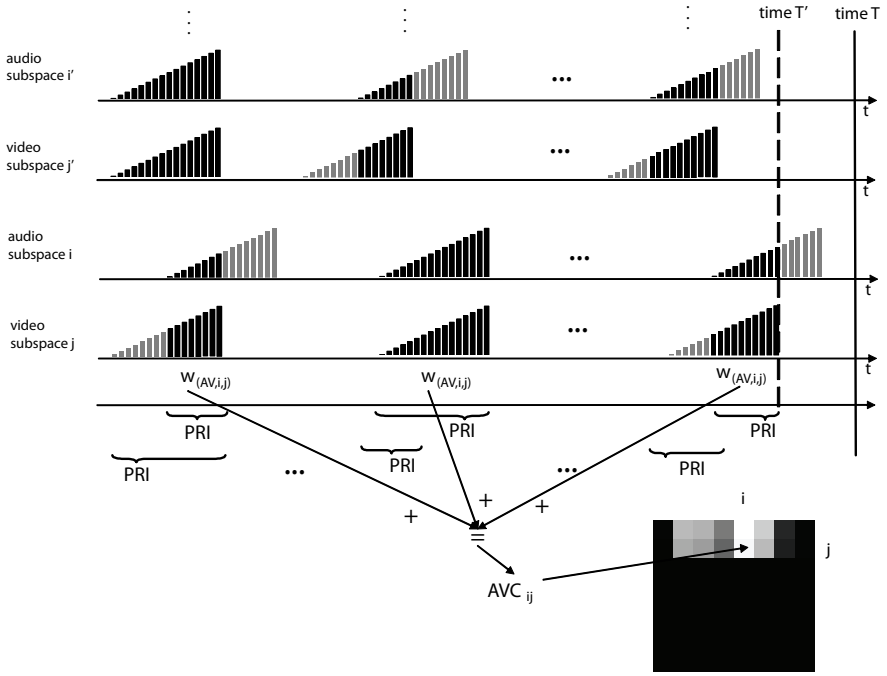


Fig. 12.3. Building process of the AVC matrix and AVE detection (the time indexing is removed for clarity): considering the audio and video subspaces i and j, we sum in the position i, j of the AVC matrix all the AV coupling weights calculated until the time step T. The dashed line corresponds to the final step T' of an AVE. After that time step, no AV association holds among any subspaces i' and j', and the AVC matrix remains unchanged.

2. $\forall t \in [t^{AV}_{init},\ t^{AV}_{end}]$, $AVC^{(t+1)} - AVC^{(t)} \neq \mathbf{0}$ (synchrony during the event)
3. $AVC^{(t^{AV}_{end}+1)} - AVC^{(t^{AV}_{end})} = \mathbf{0}$ (no synchrony after t^{AV}_{end})

In other words, an audio-video event starts when the AVC matrix changes, and terminates when the AVC matrix does not change anymore. Using this simple rule, we can segment on line the whole sequence in different K AV events AVE_k, $k = 1 \cdots K$, where each event is defined as

$$AVE_k = [t^{AV}_{init}(k),\ t^{AV}_{end}(k)] \tag{12.14}$$

and $t^{AV}_{init}(k)$ ($t^{AV}_{end}(k)$) indicates the initial (final) time step of the k-th audio video event (see Fig.12.3).

12.2.7 Audio-visual event discrimination

We have seen in the previous section that the AVC matrix can be used to segment different AV events in the sequence. Nevertheless, this matrix could produce another useful information, since it contains also a rich description of the nature of the AV event, which can be used for classifying it. In detail, we propose to extract

from the AVC matrix a feature, named *AVD* (*Audio Video Description*), defined as

$$AVD(AVE_k) = AVC^{(t^{AV}_{end}(k))} - AVC^{(t^{AV}_{init}(k)-1)} \quad (12.15)$$

In simple words, this represents the AV information accumulated only during the event k. This matrix is then vectorized and directly used as a fingerprint vector for characterizing the AV event.

In the experimental part, we performed classification and clustering trials on different audio video examples. In order to focus on the expressivity of such features, we perform clustering and classification using simple and well-known methods as K-Nearest Neighborhood for the classification, and hierarchical clustering.

12.3 Experimental Results

In this section, we will show various results obtained by applying our AV analysis to real video sequences. The aims of this section are multiple: 1) finding if the characterization of the AV events is meaningful (no over-segmentation or under-segmentation respect to a segmentation performed by a human operator); 2) testing if the features that describe the AV events are discriminant with respect to classification and clustering tasks. In Sec. 12.3.1, we will present the data set used, and we briefly discuss the role of the parameters and their selection. In Sec. 12.3.2, we will show an example of the computation of the AVC matrix, highlighting the key phases of the analysis. The remaining sections are devoted to show the method performances in the (i) detection (Sect. 12.3.3), (ii) classification (Sect. 12.3.4), and (iii) clustering (Sect. 12.3.5).

12.3.1 Data set and parameter setting

In order to test the proposed framework we concentrate on different individual activities performed in an indoor environment, captured using a standing camera and only a single microphone. The activities (some shots are depicted in Fig. 12.4) are composed by basic actions like entering the office, exiting the office, answering a phone call, talking, switching on/off the lights, and so on. Moreover, they are not overlapped, in the sense that the person appears in the scene, performs a set of basic actions and disappears, reappearing later (with time gap varying from 0.5 sec to 10 sec) to perform another sequence. The data gathering process was repeated in two separate sessions with three weeks of distance between them. In each session, a further level of variability was due to the frequent change of clothes of the person in the video. The result was 2 long video sequences (more than 2 hours overall). The sequences have been captured using a 320×240 CCD camera, 20 frames per second. The audio signal is captured at 22050 Hz, and the samples are subdivided using temporal windows with length $W_a = 1$ s, and all the windows are overlapped of 70%.

For what concerns the number I of audio spectral subbands over which calculating the SEA features and the number J of the FG histogram bins, we find that using $I = J = 8$ we have a good compromise between accuracy and low computational requirements. Nevertheless, other experiments have been performed using

Fig. 12.4. Some pictures of the two activities sequences.

different I and J, realizing this parameter is not crucial. We avoid to use $I, J > 32$ due to the curse of dimensionality problem otherwise involved.

In specific, we have considered the SEA of $I = 8$ equally subdivided subbands, in the range of $[0, 22050/4]$Hz. A 3-components mixture of Gaussians have been instantiated for each subband. This choice is not critical and can be guided from opportune considerations about the complexity of the scene, especially in relation to the complexity of the background. Actually, three components are considered a reasonable choice, taking into account the possibility of a bimodal BG, and one component for the foreground activity [138]. The FG threshold T has been set to 0.8, and the learning rate is set to $\alpha = 0.001$. For what concerns the video channel, we spatially sub-sample the video sequence by a factor of 4, in order to speed up the computation of the per-pixel FG, and we use a 3-components mixture of Gaussians for each subsampled location. Then, we build the video FG histogram using $J = 8$ bins, obtained by equally partitioning the level of FG gray interval [0,255] in 8 intervals. Each of the corresponding FG histogram signals is modelled using again 3-components mixture of Gaussians. Note that we use the same FG threshold T and the same learning rate for both the per-pixel FG detection and for the video novelty detection. The only difference among the various mixture sets consists in the initial standard deviation σ_{init} with which the mixture components are initialized when a new Gaussian is added to the model (no match), due to the different range of variability of the values of the various subspaces. We noticed that these thresholds are important: a too low initial standard deviation means that too different Gaussian components are introduced in the mixture, and consequently the resulting FG patterns are too numerous. On the other side, a too large initial standard deviation means that we model with a unique Gaussian component a signal produced by different processes. In our experiments, we noticed that the range of variability of the SEA signal is about $[0, 150]$, for the pixel signal is $[0, 255]$ and for the VFGH signal is $[0, 320 * 240/\gamma^2 = 4800]$. After analyzing several

different configurations (whose results are not shown here), we have found that the best sensitivity parameters, for the TAPPMOGs are $\sigma^{\mathrm{A}}_{init} = 10$ for the audio subspace and $\sigma^{\mathrm{P}}_{init} = 30$ $\sigma^{\mathrm{V}}_{init} = 50$ for the pixel and FG histogram subspaces, respectively.

12.3.2 An illustrative example

In order to understand the meaning of the AVC matrix, in this part we show its computation step by step, by using a real sequence (namely, the "Phone" sequence) manually extracted from the experimental dataset. The sequence, 834 frames long, depicts a static scene in which a phone is located on the top of a mobile; at some point the phone rings, and a person comes and answers the call, concluding then the conversation and going out of the scene at the end.

A graphical representation of the computation process is shown in Fig. 12.5. The salient points of the computation are:

- frame 180: no FG patterns occur, neither audio nor video: the AVC matrix is empty;
- frame 303: the phone is ringing, as we see in the audio scheme depicting the SEA values, but no video FG is present, therefore the AVC matrix remains empty;
- frame 319: the person comes and answer the call, then, the Video Foreground Histogram relative to the interval detects a FG video pattern that is concurrent with the audio FG pattern. Therefore, the starting point of an AV event is detected, and the AVC matrix shows some non null entries in the related audio-visual coordinates;
- frame 426: the conversation continues, and consequently the AVC values grow up;
- frame 660: the conversation ends, hence the FG patterns are over. The detection module communicates the end of the audio-video event, the corresponding AVD feature can be computed.

The AVD represents the feature that will be used in the following classification and clustering tasks.

12.3.3 Detection results

The sequence has been segmented automatically in audio-video events using the definition presented in Sec. 12.2.5. As ground truth, we asked a human operator to perform a segmentation of the two long sequences, highlighting human activities. Once the segmentation was performed, the 66 obtained segments were manually classified in 6 classes (situations) as follows:

1. *Make a call*: a person goes to the lab phone, dials a number, and makes a call.
2. *Receive a call*: the lab phone is ringing, a person goes to the phone and makes a conversation.
3. *First at work*: a person enters into the lab, switch on the light, and walks in the room, without talking.

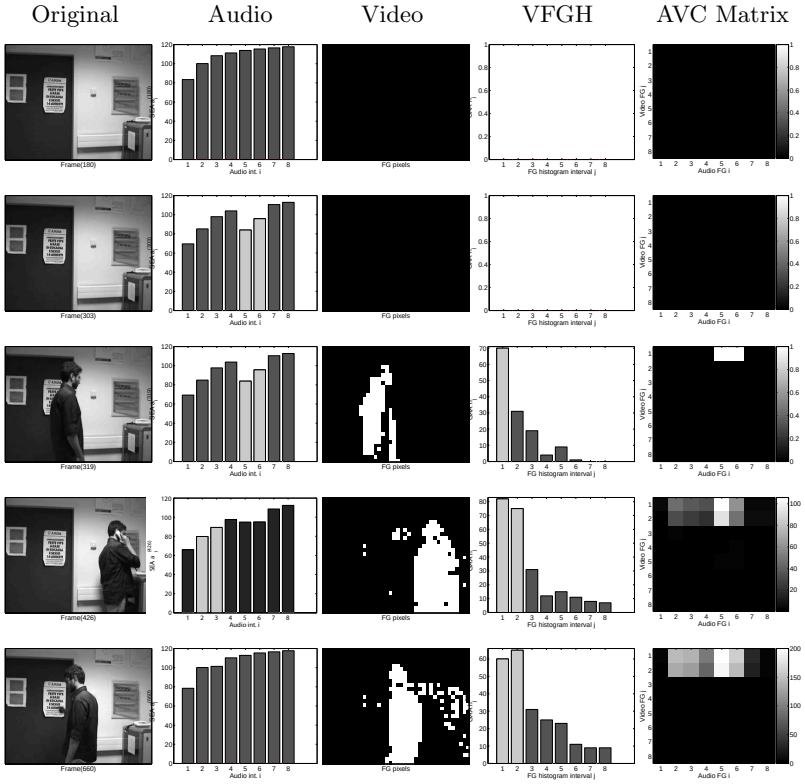


Fig. 12.5. AVC matrix computation steps for the "Phone" sequence.

4. *Not first at work*: a person enters into the lab with the light already switched on, walks in the room, and talks.
5. *Last at work*: a person exits from the lab switching off the light without talking.
6. *Not last at work*: a person exits from the lab leaving the light on, and talking.

Therefore, the original sequences were used as input to our system. The result of the automatic segmentation was optimal in the sense that all the 66 events were identified as different. Moreover, our method was good in the sense that no over-segmentation or under-segmentation was obtained, primal element to be investigated. Further testing in which the AV events will occur overlapped is actually under exploration.

12.3.4 Classification results

We tested the classification accuracy with the 66 labelled audio video events derived from the previous part, in four different scenarios, listed below:

Scenario A - Situation 1 vs situation 2: making or receiving a phone call.

Scenario B - Situation 3 vs situation 4: entering in an empty or non empty lab.

Scenario C - Situation 5 vs situation 6: exiting from an empty or a non empty lab.

Scenario D - Total problem: discrimination between all the six situations.

The classification accuracy was computed using the Nearest Neighbor classifier with Euclidean distance [53], which represents the simplest classifier permitting to understand the discriminative power of the proposed features. Classification accuracies have been estimated using the Leave One Out (LOO) scheme [53]. In order to have a better insight into the proposed method, we compare the proposed approach with the individual separated audio and video processing; in particular, the 66 audio and the video FG patterns (see Sec. 12.2.4 and Sec. 12.2.3), extracted in the same time intervals of the AVC features (i.e. during the Potential Relation Interval - PRI), have been directly used as features to characterize the events. The classification accuracies for the three methods are presented in Table 12.1. Just at a first look of the table, one can notice a general benefit in integrating

Scenario	Audio	Video	Audio-video
A	100.00%	86.35%	100.00%
B	60.87%	95.65%	95.65%
C	95.24%	85.71%	95.24%
D	62.12%	66.67%	89.39%

Table 12.1. LOO classification accuracies for the four different problems.

audio and visual information: audio-visual accuracies are the best results in all the experiments. Looking better at such figures, one can better figure out the outcome of the method, underlining some issues as follows.

- The scenario A is devoted to discriminate between making or receiving a phone call: clearly, most of the information is embedded into the audio part (when receiving a call there is a ringing phone), whereas the visual part is really similar (going to the phone, hanging up and talk). Actually, the audio signal itself is able to completely discriminate between these two events, whereas the video gets worst results. It is important to notice that the audio-visual integration does not inhibit the information brought in the audio part.
- The scenarios B and C are characterized by two similar audio-visual situations. Regarding the audio part, there is a difference between talking in the lab or not talking, whereas regarding the video part there is the difference between switching on or off the lights. Actually, both single audio and video features get good results.
- The scenario D is the most complex and interesting. In this case, which involves 6 different classes, the integrated use of audio and video information permits to drastically improve the classification accuracy of about 25%. The tasks are complicated, and only a proper integration of audio and visual information could lead to a definite satisfactory classification results.

12.3.5 Clustering results

This last section reports results about clustering, in order to really discover patterns and natural groups of audio-video events. Given the automatically segmented

dataset, we perform hierarchical clustering [80] using the Ward scheme, and we consider the Euclidean distance as distance between elements. As in the classification task, we use a simple rule for performing clustering, in order to discover the expressivity of the AVD feature. We only set the number of clusters to 6, and let the algorithm to make the natural clusters. The resulting dendrogram is shown in Fig. 12.6, where in abscissas there are the situation labels. Observing the den-

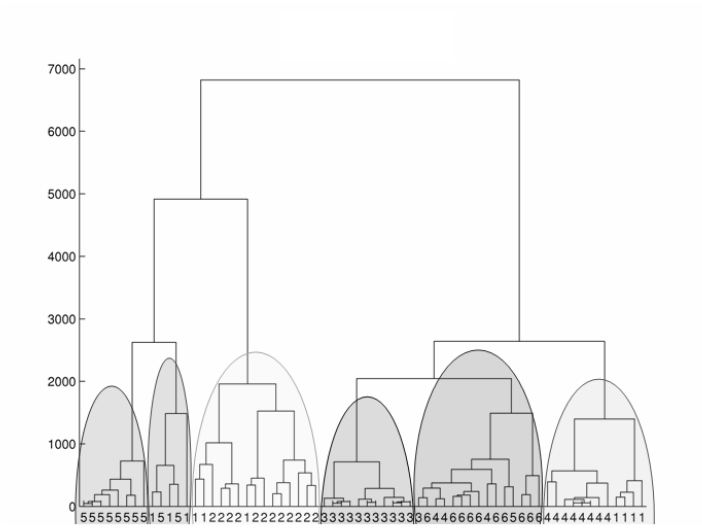


Fig. 12.6. Hierarchical clustering dendrogram.

drogram, we can see that the underlying structure of the dataset is satisfactorily represented, but, obviously, there are some errors as the task is not easy. The most separated and well identified clusters are the situations 2 and 3: they are characterized by a clearly identifiable pattern (the ringing phone and the light on). Also cluster 5 is quite well identified; little confusion is present between patterns in cluster 4 and in cluster 6. This is a difficult case: the discrimination between them lies only in the fact that in the situation 4 the AV coupling happens among the opening door and the person entering in the scene (visual FG), together with the impulsive noise of the closing door and the speech signal (audio FG). Instead, in situation 6, the coupling between the closing door and the moving person is absent because the person exited from the scene producing no visual FG.

The accuracy of clustering can be quantitatively assessed by computing the number of errors: a clustering error occurs if a pattern is assigned to a cluster in which the majority of the patterns belongs to another class. In this case, we obtain a clustering accuracy of 75.76%, which is a really satisfactory result, also compared with single modalities results: clustering of audio patterns obtained an

accuracy of 53.03%, whereas the video obtained 60.61%. Also in this case, it is evident the gain obtained by the fusion of audio-video information.

12.4 Conclusions

In this approach, a new method for characterizing video sequences using audio visual events has been presented. Separate audio and video signals have been processed using two different adaptive modules, aimed at distilling audio and video information in a unique fashion, using only one camera and one microphone. It is worth to note that the video module is a region level description, producing regions that share the property of being formed by FG pixel values, belonging to the same gray level range. Then, the audio and video patterns have been integrated, exploiting the concept of synchrony, in order to recognize audio-video events. The association is realized by the means of the Audio-Video Concurrence matrix, a feature that permits to detect and segment AV events, as well as discriminating among them. Experimental results on real sequences have shown promising results, both in terms of classification and clustering.

Part IV

Conclusions

13

Final notes

13.1 Remarks

In this thesis, the statistical generative modelling has been explored and proposed as a means to describe and analyze a video sequence. More specifically, a typical video surveillance environment has been considered, in which a static camera captures a scene situation where possible moving objects are present.

The probabilistic generative modelling provides statistical tools (actually, models) that describe a set of observations, by giving particular insight on the generative process that generated them.
The generative process is described as a random quantity, characterized by a joint distribution. The ingredients of such distribution are the visible variables, i.e. the observations, and a set of hidden random variables, formally explaining the entities that *caused* the observations. The hidden variables are possibly inter-related by conditional dependency relations, forming a structure that mirrors the dynamic of the generative process.
The parametrization of the random variables, and the realization of the hidden ones, are estimated from the observations by using statistical learning strategies. In the thesis, a particular learning strategy has been considered: the Expectation Maximization algorithm. This estimation strategy takes into account for all the possible realizations of the hidden variables, in order to derive analytically the best values of the parameters.

Concerning purely theoretical issues, the thesis has brought contributions in the inference expressivity of the Hidden Markov models. In particular, two measures able to characterize the states of the underlying Markov model have been proposed. The first measure provided an entropy-like estimation of the degree of instability of the signal modelled, which has been derived by using the concept of stationary probability distribution of the Markov chain.
The second measure served as similarity measure, that explained how much two HMMs are similar, by taking into account for their most stable components, disregarding the transient signal components observed.

The generative modelling has been adopted as a versatile framework of knowledge extraction from a video sequence. An abstract concept of *visual* hierarchy

of understanding, i.e., a hierarchical structure aimed to analyze a video sequence under different point of views, has been developed.
The hierarchical logical framework is a well-known strategy that organizes the understanding process in a bottom-up fashion, starting from raw data analysis and ending with high level reasoning.
At each layer of the hierarchy, the input data is *distilled*, meaning that the portion of data considered useless for the higher level is disregarded. In other words, all the data that is not functional to be subject to further reasoning is likely to be eliminated. Therefore, at each step of the hierarchical structure, we perform a sort of *local* reasoning.

In this thesis, the concept of "visual" hierarchy has been introduced as a necessary aspect characterizing the hierarchical structure.
The process of understanding a video sequence is based on the interpretation that raises the visual appearance to an abstract level of meaning. These meanings are highly subjective, and far for being completely captured with a bijective interpretation function *visual appearance* $\leftrightarrow$ meaning.
We did not propose any novel interpretation function. Instead, we tried to build a hierarchical structure, in which the result of each reasoning module provides a different level of understanding of the video sequence.
Each level gives a description of the sequence by using a particular visual descriptor, that is simplest at the bottom level, i.e. the pixel signal, growing up in complexity at the higher levels.

In this way, the local reasoning performed becomes immediately usable and interpretable by the user, that can improve or eventually drift adequately the higher-level analysis, with the design of ad-hoc modules.

More specifically, we proposed a visual hierarchy of understanding where the video sequence is analyzed using a per-pixel analysis, which represents the bottom layer of the hierarchy, and where each pixel evolution is modelled with a Hidden Markov Model. After this step, the visual result of the analysis is the degree of transient activity occurred in the scene. Finally, the high-level interpretation of these results is demanded to the user, who can also consider them as having *per se* a semantic meaning, derived from a low-level scene understanding process.
Alternatively, the output of this step can be used as input of an higher-level analysis process, as described in the thesis. In this case, the region entity consisted of a group of adjacent pixel locations whose HMMs represent similar chromatic evolution of the background.
The convincing results, together with the numerous applicative relapses (see Sec. 13.3), showed the effectiveness of the proposed hierarchy.

The second example of visual hierarchy of understanding started with a per-pixel standard analysis, namely the background subtraction. The higher-level online reasoning module considered a region as formed by those FG pixels (the output of the per-pixel analysis) with similar color *and* synchronized with a novel audio signal.

The concept of novel audio signal, here defined as FG audio signal, represents another main contribution of the thesis, together with the mechanisms of audio-video coupling, based on the notion of synchrony.

The coupling of audio-video FG patterns defines a multimodal region description of an event in a video sequence.
In order to test the effectiveness of such an event representation, we used it as discriminative feature, useful to perform clustering and classification tasks. As a result, we observed that the multimodal representation is more expressive with respect to the single modality representations, leading to better clustering and classification performances.

13.2 Future perspectives

The statistical generative modelling appears to be a powerful way to tackle with the uncertainty in general; for this reason, it seems a reasonable paradigm to deal with the video understanding issue.
Actually, the incertitude is a widespread but not so formally defined aspect, that assumes different names, like sensor noise, model uncertainty, etc.. Therefore, it seems advantageous to perform a hierarchical partition of the video sequence analysis, in order to separately deal with the various kind of uncertainties, in a bottom-up fashion.

In this thesis, we provided a hierarchy where the structure at the various levels is dependent on the complexity of the visual descriptor adopted to specify the sequence. Essentially, we perform analysis on the pixel level and on the region level; in the pixel level the research performed leaved us satisfied, being us able to characterize a video sequence in on-line fashion (by using the standard background subtraction technique presented in Part III) and off-line fashion (by using the proposed HMM based method). In the region level, we perform on-line analysis, by providing a multimodal description of the scene (Part III). This description has been used later as feature for clustering and classification tasks. A generative, purely off-line description of the sequence has left out of the drafting of the thesis, due to its embryonal development status, and is actually under study. The direction that we are following is the one proposed in [83], in which the sequence is characterized by a set of moving flexible sprites.

The generative modelling, as pure theoretical framework, is characterized by open questions, far for being considered as solved. At first, the model selection issue, which is tightly related with the video sequence modelling, at each descriptive level.
Especially at the region-based level, for instance, where each region could be modelled separately, a model selection technique can individuate the number of entities present in a video sequence.
Actually, the usual way to solve this issue is represented by a try & score process, where a set of models are created and evaluated using Bayesian Estimation Criteria (BIC), Minimum Description Length (MDL) tests, and a few others [57, 67].
Other ways to discover the "right" topology come from the Artificial Intelligence community, where graph-based methods are exploited, based on the automatic discovering of statistical relations such as likelihood, conditioning, relevance, causation [113, 114]. At the best of our knowledge, this latter group of methods has

never been used by the Computer Vision (CV) community for video sequence modelling; therefore, our attention will be focused on the first group of techniques.

Concerning the multimodal modelling, we are aware that the introduction of audio analysis in the CV community can be useful not only for tracking tasks, but also for off-line applications such as classification and clustering. In this sense, a good coupling of audio and video data can be considered as a useful feature extraction step, that define audio-visual entities in an intuitive and effective way.

13.3 Publications and other contributions

Some parts of this study have been published in conference proceedings or international journals. More specifically, Part II and Chapter 11 of Part III have been published in [15, 38, 39]. In [15] the complete hierarchy of understanding based on the HMMs is presented. In [38] an application of the region representation to tracking issues is proposed, and in [39] the region based representation is used as a pixel-region background initialization algorithm.

The remaining of Part III has been published in [40, 41]. In [40] the definition of the audio foreground is given, as well as an algorithm to identifying audio foreground in an audio stream. A coupling algorithm between audio and video data, based on the concept of synchrony exploited in the thesis, is presented in [41]; in this paper, the audio video region description is aimed to solve a problem typical of the video surveillance literature, i.e. the sleeping foreground problem. Finally, the whole system of multimodal region description has been submitted to the IEEE Transactions on Multimedia.

The study of these three years concerned also further Computer Vision applications; actually, our intent was primarily to evaluate the effective expressivity of the generative statistical modelling.

In this thesis, we prefer not to present all the studies carried out, for a twofold reason: principally, in order to privilege the theoretical results covered by a deep and adequate experimental section; secondly, because some studies are still under development.

Anyway, in order to give a global vision of the research tackled, in the following we cite the other studies, published or submitted.

In [43], we faced the process of super resolution in a generative way [143], in which each frame is considered as observation of a (hidden) higher resolved exemplar. In specific, we propose a method to extract from a video sequence a set of higher resolved images, following the idea of video clustering proposed in [60]. In the visual hierarchy of understanding this approach can be considered as an high level module, considering as basic visual entities the frames of the sequence.

In [34], we proposed a novel method for BG initialization and recovery, that merges interesting ideas coming from the video inpainting and the generative modelling subfields. The method takes as input a video sequence, in which several objects move in front of a stationary BG. The method is based on the following hypotheses: (i) a portion of the BG, called *sure BG*, can be identified with high certainty by using only per-pixel reasoning and (ii) the remaining scene BG can

be generated utilizing exemplars of the sure BG. In the visual hierarchy of understanding this approach can be considered as a couple of modules pixel level and region level, that cooperate in an iterative way.

In [42], currently under review, we propose a probabilistic model aimed at modelling interactions among human subjects. The rationale under the proposed model lies in the possibility to deal with interactive processes whose characterizing states exhibit different temporal durations and whose dynamics can be modeled as Markovian. The study presents preliminary results, and is still under development.

References

1. J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic. Discovering clusters in motion time-series data. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 375–381, 2003.
2. F. R. Bach and M. I. Jordan. Learning graphical models for stationary time series. *IEEE Trans. Signal Process.*, 52(8):2189–2199, 2004.
3. C. Bahlmann and H. Burkhardt. Measuring hmm similarity with the bayes probability of error and its application to online handwriting recognition. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 406–411, 2001.
4. R. Bakis. Continuous speech word recognition via centisecond acoustic states. In *Proc. ASA Meeting*, Washington, DC, 1976.
5. E.M. Bakker, T.S. Huang, M.S. Lew, N. Sebe, and X.S. Zhou, editors. *Image and Video Retrieval, Second International Conference, CIVR 2003, Urbana-Champaign, IL, USA, July 24-25, 2003, Proceedings*, volume 2728 of *Lecture Notes in Computer Science*. Springer, 2003.
6. Dana H. Ballard. Animate vision. *Artif. Intell.*, 48(1):57–86, 1991.
7. L.E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequality*, 3:1–8, 1970.
8. 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. *Bull. Amer. Meteorology Soc.*, 73:360–363, 1967.
9. L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Math. Statistics*, 37:1,554–1,563, 1966.
10. L.E. Baum, T.E. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Math. Statistics*, 41(1):164–171, 1970.
11. L.E. Baum and G.R. Sell. Growth functions for transformations on manifolds. *Pacific J. Math.*, 27(2):211–227, 1968.
12. M. Beal, N. Jojic, and H. Attias. A graphical model for audiovisual object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(7):828–836, 2003.
13. Y. Bengio. Markovian models for sequential data. *Neural Computing Surveys*, 2:129–162, 1999. available at `http://www.icsi.berkeley.edu/~jagota/NCS`.
14. James O. Berger, Victor De Oliveira, and Bruno Sans. Objective bayesian analysis of spatially correlated data. Technical report, Institute of Statistics and Decision Sciences Duke University Durham (USA), Departemento de Cmputo Cientfico y Estadstica Universidad Simn Bolvar Caracas (Venezuela), April 2000.

15. M. Bicego, M. Cristani, and V. Murino. Unsupervised scene analysis: A hidden markov model approach. *Computer Vision and Image Understanding*, not known yet. in print.
16. M. Bicego, A. Dovier, and V. Murino. Designing the minimal structure of Hidden Markov Models by bisimulation. In M.A.T. Figueiredo, J. Zerubia, and A.K. Jain, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, LNCS 2134, pages 75–90. Springer, 2001.
17. M. Bicego, V. Murino, and M.A.T. Figueiredo. A sequential pruning strategy for the selection of the number of states in Hidden Markov Models. *Pattern Recognition Letters*, 24(9–10):1395–1407, 2003.
18. M. Bicego, V. Murino, and M.A.T. Figueiredo. Similarity-based clustering of sequences using hidden Markov models. In P. Perner and A. Rosenfeld, editors, *Machine Learning and Data Mining in Pattern Recognition*, volume LNAI 2734. Springer, 2003.
19. Jeff Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report ICSI-TR-97-021, ICSI, 1997.
20. C.M. Bishop, A. Blake, and B. Marthi. Super-resolution enhancement of video. In C.M. Bishop and B. Frey, editors, *Proceedings Artificial Intelligence and Statistics*, 2003.
21. A. Bobick and J. Davis. An appearance-based representation of action. In *ICPR '96: Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I*, pages 307–310, 1996.
22. M. Brand. An entropic estimator for structure discovery. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1999.
23. A.S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, London, 1990.
24. P. Brémaud. *Markov Chains*. Springer-Verlag, 1999.
25. P.M.T. Broersen. Automatic spectral analysis with time series models. *IEEE Transactions on Instrumentation and Measurement*, 51(2):211–216, 2002.
26. R.A. Brooks, R. Greiner, and T.O. Binford. The acronym model-based vision system. In *IJCAI79*, pages 105–113, 1979.
27. H. Buxton and S. Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78(1-2):431–459, 1995.
28. A. Camproux, F. Saunier, and G. Thomas. A hidden Markov model approach to neuron firing patterns. *Biophysical journal*, 71(5):2404–2412, 1996.
29. Olivier Cappé. Ten years of HMMs, 2001. Available at `http://www.tsi.enst.fr/ cappe/docs/hmmbib.html`.
30. K.R. Castleman. *Digital Imgage Processing*. Prentice Hall, 1996.
31. N. Checka and K. Wilson. Person tracking using audio-video sensor fusion. Technical report, MIT Artificial Intelligence Laboratory, 2002.
32. R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, 2000.
33. R.T. Collins. Mean-shift blob tracking through scale space. In *CVPR (2)*, pages 234–240, 2003.
34. A. Colombari, M. Cristani, V. Murino, and A. Fusiello. Exemplar-based background model initialization. In *Proceedings of ACM VSSN 2005 Workshop on Video Surveillance*, pages 29–36, 2005.

35. T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley and Sons, 1991.
36. R.G. Cowell, S.L. Lauritzen, A.P. David, D.J. Spiegelhalter, V. Nair, J. Lawless, M. Jordan, and David J. Spiegelhater. *Probabilistic Networks and Expert Systems.* Springer-Verlag New York, Inc., 1999.
37. M. Cowling and R.Sitte. Comparison of techniques for environmental sound recognition. *Pattern Recognition Letters*, 24:2895–2907, 2003.
38. M. Cristani, M. Bicego, and V. Murino. Integrated region- and pixel-based approach to background modelling. In *Proc. of IEEE Workshop on Motion and Video Computing*, pages 3–8, 2002.
39. M. Cristani, M. Bicego, and V. Murino. Multi-level background initialization using hidden markov models. In *Proc. of ACM SIGMM Workshop on Video Surveillance*, pages 11–19, 2003.
40. M. Cristani, M. Bicego, and V. Murino. Audio background modelling. In *Proceedings of International Conference on Pattern Recognition (ICPR 2004)*, pages 30–38, 2004.
41. M. Cristani, M. Bicego, and V. Murino. Audio-video background modelling. In *Proceedings of European Conference on Computer Vision (ECCV 2004)*, 2004.
42. M. Cristani, D.S. Cheng, V. Murino, and R. Nevatia. Not given. In *NOT GIVEN*, 2006. under review.
43. M. Cristani, D.S. Cheng, V. Murino, and D. Pannullo. Distilling information with super-resolution for video surveillance. In *Proceedings of ACM VSSN 2004 Workshop on Video Surveillance*, pages 5–13, 2004.
44. M.S. Crouse, R.D. Nowak, and R.G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Trans. on Signal Processing*, 46(4):886–902, 1998.
45. M. Bach Cuadra, L. Cammoun, T. Butz, and J. Cuisenaire Oand Thiran. From error probability to information theoretic (multi-modal) signal processing. *Signal Processing*, 85(5):875–902, 2005.
46. T. Darrell, J. Fisher, and K. Wilson. Geometric and statistical approaches to audio-visual segmentation for unthetered interaction. Technical report, CLASS Project, 2002.
47. E.R. Davies. *Machine Vision.* Academic Press, second edition, 1997.
48. James W. Davis and Aaron F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 928–934, 1997.
49. D. Demirdjian, K. Tollmar, K. Koile, N. Checka, and T. Darrell. Activity maps for location-aware computing. In *Proc. of IEEE Workshop on Applications of Computer Vision*, pages 70–75, 2002.
50. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. In *Proceedings of the Royal Statistical Society*, volume 39, pages 1–38, 1977.
51. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, 39:1–38, 1977.
52. Y. Deng and B.S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.
53. R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* John Wiley and Sons, second edition, 2001.
54. R. Durbin, S. Eddy, A. Krogh, and G.J. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, 1998.

55. S. Eickeler, A. Kosmala, and G. Rigoll. Hidden Markov Model based continuous online gesture recognition. In *IEEE Proc. Int. Conf. Pattern Recognition*, volume 2, pages 1206–1208, 1998.
56. S. Eickeler, S. Mller, and G. Rigoll. Recognition of jpeg compressed face images based on statistical methods. *Image and Vision Computing*, 18:279–287, March 2000.
57. M.A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.
58. J.W. Fisher III, T. Darrell, W.T. Freeman, and P.A. Viola. Learning joint statistical models for audio-visual fusion and segregation. In *Advances in Neural Information Processing Systems*, pages 772–778. MIT Press, 2000.
59. G.D. Forney. The Viterbi algorithm. *Proc. of IEEE*, 61:268–278, 1973.
60. B.J. Frey and N. Jojic. Transformation-invariant clustering using the EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):1 – 17, January 2003.
61. B.J. Frey and N. Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1392–1413, 2005.
62. Z. Ghahramani. An introduction to Hidden Markov Models and Bayesian Networks. *Int. Journal of Pattern Recognition and Artificial intelligence*, 15(1):9–42, 2001. Special Issue in Hidden Markov Models in Vision.
63. Z. Ghahramani and M.J. Beal. Graphical models and variational methods. In *Advanced mean field methods: theory and practice*. MIT Press, 2000.
64. Enrico Giusti. *Analisi Matematica 2*, chapter 7, pages 318–334. Bollati Boringhieri, second edition, 1989.
65. B. Gloyer, H. K. Aghajan, K. Y. Siu, and T. Kailath. Video-based freeway monitoring system using recursive vehicle tracking. In *IS&T-SPIE Symposium on Electronic Imaging: Image and Video Processing*, 1995.
66. S. Gong, J. Ng, and J. Sherrah. On the semantics of visual behaviour, structured events and trajectories of human action. *Image and Vision Computing*, 20(12):873–888, 2002.
67. S. Gong and T. Xiang. Recognition of group activities using a dynamic probabilistic network. In *Proc. of Int. Conf. on Computer Vision*, pages 742–749, 2003.
68. Y.K. Ham and R.-H. Park. 3D object recognition in range images using hidden Markov models and neural networks. *Pattern Recognition*, 32(5):729–742, 1999.
69. B. Hannaford and P. Lee. Hidden Markov model analysis of force/torque information in telemanipulation. *International Journal of Robotics Research*, 10(5):528–539, 1991.
70. D.R. Hardoon, S. Szedmak, and J.Shawe-Taylor. Canonical correlation analysis an overview with application to learning methods. Technical Report CSD-TR-03-02, Royal Holloway University of London, 2003.
71. I. Haritaoglu, D. Harwood, and L.S. Davis. W^4: real-time surveillance of people and their activities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.
72. H.Buxton. Learning and understanding dynamic scene activity: a review. *Image and Vision Computing*, 21:125–136, 2003.
73. Y. He and A. Kundu. 2-D shape classification using Hidden Markov Model. *IEEE Trans. Pattern Analysis Machine Intelligence*, 13(11):1172–1184, 1991.
74. D. Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, 1995. Revised November, 1996 - dowloadable from `ftp://ftp.research.microsoft.com/pub/tr/tr-95-06.pdf`.

75. K.E. Hild II, D. Erdogmus, and J.C. Principe. On-line minimum mutual information method for time-varying blind source separation. In *Intl. Workshop on Independent Component Analysis and Signal Separation (ICA '01)*, pages 126–131, 2001.
76. S. Hongeng and R. Nevatia. Large-scale event detection using semi-hidden markov models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, Nice, France, October 2003.
77. J. Hu, M.K. Brown, and W. Turin. HMM based online handwriting recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, 1996.
78. J.W. Fisher III and T. Darrell. Speaker association with signal-level audiovisual fusion. *IEEE Trans. on Multimedia*, 6(3):406–413, 2004.
79. M. Isard and A. Blake. CONDENSATION: Conditional density propagation for visual tracking. *Int. J. of Computer Vision*, 29(1):5–28, 1998.
80. A.K. Jain and R.C. Dubes. *Algorithms for clustering data.* Prentice Hall, 1988.
81. T. Jebara and A. Pentland. Action reaction learning: Automatic visual analysis and synthesis of interactive behavior. In *Proc. Int Conf. Computer Vision Systems*, 1999.
82. G. Jing, C.E. Siong, and D. Rajan. Foreground motion detection by difference-based spatial temporal entropy image. In *IEEE Tencon 2004*, 2004.
83. N. Jojic and B.J. Frey. Learning flexible sprites in video layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 199–206, 2001.
84. Michael I. Jordan. *Learning in graphical models.* MIT Press, 1999.
85. B. Juang and L. Rabiner. A probabilistic distance measure for hidden markov models. *AT&T Tech. Journal*, 64(2):391–408, 1985.
86. B.H. Juang, S.E. Levinson, and M.M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of Markov Chain. *IEEE Trans. Information Theory*, 32(2):307–309, 1986.
87. B.H. Juang and L.R. Rabiner. Mixture autoregressive hidden Markov models for speech signals. *IEEE Trans. Acoustic Speech and Signal Processing*, 33(6):1404–1413, 1985.
88. J. Kay. Feature discovery under contextual supervision using mutual information. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4, pages 79–84. IEEE Computer Society, 1992.
89. Jeffrey H. Kingston. *Algorithms and Data Structures*, chapter Graphs, pages 252–253. Addison-Wesley, 1998.
90. V. V. Kohir and U. B. Desai. Face recognition using DCT-HMM approach. In *Workshop on Advances in Facial Image Analysis and Recognition Technology (AFIART)*, Freiburg, Germany, June 1998.
91. V. Krishnamurthy, S. Dey, and J. LeBlanc. Blind equalization of iir channels using hidden Markov models and extended least squares. *IEEE Trans on Signal Processing*, 43(12):2994–3006, 1995.
92. S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.
93. M.H. Law and J.T. Kwok. Rival penalized competitive learning for model-based sequence. In *Proc. Int. Conf. Pattern Recognition*, volume 2, pages 195–198, 2000.
94. J.J. Lee, J. Kim, and J.H. Kim. Data-driven design of HMM topology for online handwriting recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15(1):107–121, 2001.
95. S.E. Levinson, L.R. Rabiner, and M.M Sondhi. An introduction to the application of the theory of probabilistic function of a Markov process to automatic speech recognition. *Bell Syst. Tech. J.*, 62(4):1035–1074, 1983.

96. C. Li and G. Biswas. A bayesian approach to temporal data clustering using hidden Markov models. In *Proc. Int. Conf. on Machine Learning*, pages 543–550, 2000.
97. C. Li and G. Biswas. Applying the Hidden Markov Model methodology for unsupervised learning of temporal data. *Int. Journal of Knowledge-based Intelligent Engineering Systems*, 6(3):152–160, 2002.
98. J. Li, A. Najmi, and R.M. Gray. Image classification by a two-dimensional hidden Markov model. *IEEE Trans on Signal Processing*, 48(2):517–533, 2000.
99. A.P. Pentland M.A. Turk. Face recognition using eigenfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586 –591, June 1991.
100. D. Mackay. Introduction to Monte Carlo methods. In M. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1999. `available as ftp://wol.ra.phy.cam.ac.uk/pub/mackay/erice.ps.gz`.
101. David J. C. Mackay. *Information theory, inference, and learning algorithms*, chapter Exact Marginalization. Cambridge University Press, 2003.
102. D. Magee. Tracking multiple vehicles using foreground, background and motion models. In *Int. Workshop Statistical Methods in Video Processing*, pages 7–12, 2002.
103. S.L. Marple. *Digital Spectral Analysis*. Prentice-Hall, second edition, 1987.
104. M. Mason and Z. Duric. Using histograms to detect and track objects in color video. In *The 30th IEEE Applied Imagery Pattern Recognition Workshop (AIPR'01)*, pages 154–159, Washington, D.C., USA, October 2001.
105. I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang. Automatic analysis of multimodal group actions in meetings. *IEEE Transactions on Pattern Analysis and Machine Intelligence (to appear)*, 2004. To appear.
106. G. J. McLachan and T.Krishnan. *The EM algorithm and extensions*. Wiley Interscience, first edition, 1997.
107. R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.
108. J. Ng, S. Kwong, and S. Gong. Learning pixel-wise signal energy for understanding semantics. *Image Vision Comput.*, 21(13-14):1183–1189, 2003.
109. E. Niebur, S.S. Hsiao, and K.O. Johnson. Synchrony: a neuronal mechanism for attentional selection? *Current Opinion in Neurobiology*, 12:190–194, 2002.
110. S.E. Palmer. *Vision Science*. MIT press, 1998.
111. A. Panuccio, M. Bicego, and V. Murino. A Hidden Markov Model-based approach to sequential data clustering. In T. Caelli, A. Amin, R.P.W. Duin, M. Kamel, and D. de Ridder, editors, *Structural, Syntactic and Statistical Pattern Recognition*, LNCS 2396, pages 734–742. Springer, 2002.
112. H.-S. Park and S.-W. Lee. A truly 2D hidden Markov model for off-line handwritten character recognition. *Pattern Recognition*, 31(12):1849–1864, 1998.
113. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.
114. J. Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, 2000.
115. V. Peltonen. Computational auditory scene recognition. Master's thesis, Tampere University of Tech., Finland, 2001.
116. W.D. Penny and S.J. Roberts. Dynamic models for nonstationary signal segmentation. *Computers and Biomedical Research*, 32(6):483–502, 1998.
117. W.D. Penny, S.J. Roberts, E. Curran, and M. Stokes. EEG-based communication: a pattern recognition approach. *IEEE Trans. Rehabilitation Engineering*, 8(2):214–215, 2000.

118. M. Petkovic and W. Jonker. *Content-Based Video Retrieval : A Database Perspective (Multimedia Systems and Applications)*. Springer, 2003.
119. S. Pfeiffer, R. Lienhart, and W. Effelsberg. Scene determination based on video and audio features. *Multimedia Tools Appl.*, 15(1):59–81, 2001.
120. B. Povlow and S. Dunn. Texture classification using noncausal hidden Markov models. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 17(10):1010–1014, 1995.
121. J. Prager, P. Nagin, R. Kohler, A. Hanson, and E.M. Riseman. Segmentation processes in the visions system. In *Proc. of the 5th IJCAI*, pages 642–643, 1977.
122. L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, 1993.
123. L.R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. of IEEE*, 77(2):257–286, 1989.
124. G. Radons, J.D. Becker, B. Dülfer, and J. Krüger. Analysis, classification, and coding of multielectrode spike trains with hidden Markov models. *Biol. Cybern*, 71:359–373, 1994.
125. C. Raphael. Automatic segmentation of acoustic musical signals using hidden Markov models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):360–370, 1999.
126. C.P. Robert, T. Rydén, and D.M. Titterington. Bayesian inference in hidden Markov models through jump Markov chain Monte Carlo. *J. Royal Statistic Society B*, 62(1):57–75, 2000.
127. S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. Technical report, 6 King's College Road, Toronto M5S 3H5, Canada, 1997.
128. S.T. Roweis. One microphone source separation. In *Advances in Neural Information Processing Systems*, pages 793–799, 2000.
129. T. Rydén, T. Teräsvirta, and S. Åsbrink. Stylized facts of daily return series and the hidden Markov model of absolute returns. *Journal of Applied Econometrics*, 13:217–244, 1998.
130. F. Samaria. *Face recognition using Hidden Markov Models*. PhD thesis, Engineering Department, Cambridge University, October 1994.
131. R. Schalkhoff. *Pattern Recognition, statistical, structural and neural approaches*. John Wiley and Sonsg, 1992.
132. M. Shah. Understanding human behavior from motion imagery. *Mach. Vision Appl.*, 14(4):210–214, 2003.
133. E. Shechtman, Yaron Caspi, and Michal Irani. Increasing space-time resolution in video. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 753–768. Springer-Verlag, 2002.
134. J. Sherrah and S. Gong. Continuous global evidence-based bayesian modality fusion for simultaneous tracking of multiple objects. In *Proc. of Int. Conf. on Computer Vision*, pages 42–29, 2001.
135. M. Slaney and M. Covell. Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. In *Proc. Neural Information Processing (NIPS 2000)*, 2000.
136. P. Smyth. Clustering sequences with hidden Markov models. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing*, volume 9. MIT Press, 1997.
137. C. Stauffer. Estimating tracking sources and sinks. In *IEEE Workshop on Event Mining*, pages 34–39, 2003.
138. C. Stauffer and W.E.L Grimson. Adaptive background mixture models for real-time tracking. In *Int. Conf. Computer Vision and Pattern Recognition*, volume 2, 1999.

139. B.E. Stein and M.A. Meredith. *The Merging of the Senses.* MIT Press, Cambridge, 1993.
140. B. Stenger, V. Ramesh nad N. Paragios, F.Coetzee, and J. M. Buhmann. Topology free hidden Markov models: Application to background modeling. In *Int. Conf. Computer Vision*, volume 1, pages 294–301, 2001.
141. A. Stolcke and S. Omohundro. Hidden Markov Model induction by Bayesian model merging. In S.J. Hanson, J.D. Cowan, and C.L Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 11–18. Morgan Kaufmann, San Mateo, CA, 1993.
142. R. Collins T. Kanade and A. Lipton. Special issue on video surveillance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8), 2000.
143. M.E. Tipping and C.M. Bishop. Bayesian image super-resolution. In *Neural Information Processing Systems - NIPS'2002*, Vancouver, 2002.
144. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Int. Conf. Computer Vision*, pages 255–261, 1999.
145. S.R. Veltman and R. Prasad. Hidden Markov models applied to on-line handwritten isolated character recognition. *IEEE Trans. on Image Processing*, 3(3):314–318, 1994.
146. L. Venkataramanan, R. Kuc, and F.J. Sigworth. Identification of hidden Markov models for ion channel currents. Part II. State-dependent excess noise. *IEEE Trans. on Signal Processing*, 46(7):1916–1929, 1998.
147. L. Venkataramanan, R. Kuc, and F.J. Sigworth. Identification of hidden Markov models for ion channel currents. Part III: Bandlimited sampled data. *IEEE Trans. on Signal Processing*, 48(2):376–385, 2000.
148. L. Venkataramanan, J.L. Walsh, R. Kuc, and F.J. Sigworth. Identification of hidden Markov models for ion channel currents. Part I. Colored background noise. *IEEE Trans. on Signal Processing*, 46(7):1901–1915, 1998.
149. T. Verma and J. Pearl. Causal networks: semantics and expressiveness. In R. D. Shachter, T. S. Levitt, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 4*, pages 69–76, Amsterdam, 1990. North-Holland.
150. A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. on Information Theory*, IT-13:260–269, 1967.
151. T. Wada and T. Matsuyama. Multiobject behavior recognition by event driven selective method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):873–887, 2000.
152. Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *CVPR (1)*, pages 120–127, 2004.
153. A.D. Wilson and A.F. Bobick. Parametric hidden Markov models for gesture recognition. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 21(9):884–900, 1999.
154. K. Wilson, N. Checka, D. Demirdjian, and T. Darrell. Audio-video array source separation for perceptual user interfaces. In *Proceedings of Workshop on Perceptive User Interfaces*, 2001.
155. C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.
156. C. F. J. Wu. On the convergence properties of the em algorithm. In *The Annals of Statistics*, volume 11(1), pages 95–103, 1983.
157. C.F.J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

158. T. Xiang, S. Gong, and D. Parkinson. Autonomous visual event detection and classification without explicit object-centred segmentation and tracking. In *Proc. of the British Machine Vision Conference*, pages 233–242, 2002.
159. A. Bobick Y. Ivanov, C. Stauffer and W. E. L. Grimson. Video surveillance of interactions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages ??–??, 1999.
160. D. Zhang, D. Gatica-Perez, and S. Bengio. Semi-supervised adapted HMMs for unusual event detection. In *CVPR '05: Proceedings of the 2005 Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pages 0–0, 2005.
161. T. Zhang and C. Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on Speech and Audio Processing*, 9(4):441–457, 2001.
162. D.N. Zotkin, R. Duraiswami, and L.S. Davis. Joint audio-visual tracking using particle filters. *EURASIP Journal of Applied Signal Processing*, 11:1154–1164, November 2002.
163. X. Zou and B. Bhanu. Pixels that sound. In *CVPR '05: Proceedings of the 2005 Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pages 88–95, 2005.
164. X. Zou and B. Bhanu. Tracking humans using multi-modal fusion. In *CVPR '05: Proceedings of the 2005 Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pages 0–0, 2005.