DISI - Dipartimento di Informatica e Scienze dell'Informazione
Università degli studi di Genova

PhD Thesis Proposal - XXII Ciclo

Learning to classify visual dynamic cues

Nicoletta Noceti

Thesis Supervisor
Dott. Francesca Odone

Abstract

Classification based on dynamic information is a challenging research domain that finds application in a number of fields, including video-surveillance and video retrieval.
Focusing on the video-surveillance framework, traditional approaches based on motion analysis address many interesting applications, such as access control, anomaly detection, congestion analysis and multicamera event description: in all these cases it is common practice to devise a measurement phase that extracts low level information from videos. To this purpose a wide variety of methods have been presented in the computer vision literature, leading to solutions that effectively describe the video content in moderately difficult conditions. A well known limit of these methods is that while they provide effective tools to model the dynamics of a single video, they do not suffice when the problem of interest requires a higher generalization level.
In the case of behaviour modelling or motion classification, it is advisable to increase the abstraction of the data, designing higher-level descriptions able to model broader construct: in recent years a few intersting works employing statistical methods showed how these techniques may improve performance in terms of accuracy and efficiency.
The learning from examples supplies statistical methods to study the connections between the measurements and provides the systems with the ability of being adaptive, acquiring behaviour models by long time observations.
This thesis focuses on:

- to study and develop robust methods to retrieve space-time information from a video;
- to study and develop higher-level descriptions, to plug the video processing phase in the learning phase;
- to devise dimensionality reduction and unsupervised learning strategies to model common events and anomalies from examples.

These objectives will be addressed both from the theoretical and the applicational standpoint and will be integrated in a prototype architecture that combines vision methods for scene perception and analysis, feature selection and learning techniques for high level description and decision making.

Contents

1 Introduction

Classification based on dynamic information is a challenging problem in computer vision with central importance in a number of present-day applications, including video-surveillance and video retrieval. In the last decades we have assisted to the growing interest on the use of video, rather than still images: advances on hardware components designed for digital acquisition made storage easier and processing faster, promoting their diffusion on a large scale. For these reasons, they provide nowadays an effective source of data and an appealing tool for many applications that cannot do without temporal information. Video-surveillance and monitoring [35], automatic sign reading [48] or expression recognition [51], just to name a few, involve the study of events characterized by strong variations of the data in both the spatial and the temporal dimensions.
The main objective of this project is designing methods for space-time descriptions of dynamic information in unsupervised and adaptive settings: the video-surveillance framework, today very popular also for recent history events, will be the inspiration for the study and the design of methods for gathering low-level measurements from videos and abstracting to obtain general model for dynamic descriptors.
Video processing is the focus of the first part of my work: we will study methods to address the different stages included in typical processing framework of visual surveillance, taking into account the computational requirements dictated from a practical use. The structure of such framework can be summarized as follow:

- **environment modeling** that is studying the scene context in order to
  - exploit information about acquisition setting, classes of possible moving structures and types of motion;
  - model the background;
- **detection and description of motion in the scene** that is segmenting moving structures of interest from the static elements of the scene and to find representations appropriate for further steps of analysis;
- **targets tracking** i.e. following the motion of each foreground object, obtaining a trajectory of measurements over the sequence.

Although the computer vision literature provides a wide variety of methods to these purposes, the set of problems we mentioned above is not completely solved, except for moderately simple scenarios: our working setting must face a number of additional problems, mainly caused by unsupervision assumptions and requirement for robustness over time.
This is true in particular in the case of human tracking, where the non-rigid structure of the body and the typical interactions between people make detection and tracking harder from the standpoint of efficiency and quality of results.
A well known limit of vision approaches is that they are very effective when the focus is on single events but fail when the problem of interest requires a higher generalization.
The second part of my work, thus, aims at designing and evaluating techniques of data abstraction: starting from motion features measured in a video, the goal is moving to more general descriptions of motion patterns. It is argued nowadays that the combination of statistical learning from examples and vision processing is very effective in many cases in which a great amount of data is required to be classified or analized to detect internal structures (*video mining*) and general models (*predictive systems*).A few recent works [4, 34] show how this approach allows to model sufficiently broad dynamic

constructs, providing descriptions suitable for a learning framework.
Focusing on the video-surveillance framework, in particular, there is a growing need for adaptive systems, able to learn behaviour models by long time observations and exploiting the knowledge coming from *previously seen scenarios*. The goal of the last part of this project is exploiting learning from examples to tell common events apart from anomalies: the almost complete absence of labeled data requires to plug the higher-level descriptions coming from the abstraction step into an unsupervised learning framework. We need also to consider that the notions of "anomaly" as well as "common event" strongly depend on the context.
The remainder of this research proposal is organized as follow: Section 1 details the most common vision methods able to process a video signal to extract low-level measurements; tracking approaches and temporal series modeling are the focus of Section 3. Sections 4 and 5 are devoted to present the learning from examples theory and to introduce a few works proposed in computer vision literature which try to combine vision and statistical methods for designing vision task solutions. The results we achieved during this year will be discussed in Section 6. In the remainder of the proposal, I will detail our project in Section 7, concluding, in Section 8, with an outline of objectives.

2 Low level video analysis

The apparent motion of objects in the image plane is a strong visual cue to understand the semantics of 3D motion. Assuming that the illumination does not change, image variations are due to the relative motion between camera and scene. On this respect, it is worth considering the following acquisition settings:

SCENE / CAMERA	Static	Dynamic
Static	absence of motion	only a subset of moving pixels
Moving	Ego motion	2 types of motions

The most general setting is the one with moving camera and dynamic scene and a reliable way to address motion analysis in this case would be able to cope also with the others. However, such methods are usually too noisy and computationally intractable, so for practical reasons it is advisable to adopt relative easier conditions when it is the case.
The basic operation to study the motion of a scene is doing pixel-based analysis to determine the displacements of each pixel between two consecutive frames: this operation can be performed globally (optical flow, Section 2.2) or using a sparse approach (local features, Section 2.3). In both cases, we address the problem of estimating the motion vector in a point.
An alternative level of analysis is centered on the *motion segmentation* problem. When the camera is still this can be solved as a pixel based classification (moving or still?). A connected component within moving areas can be seen as a visual representation of a moving structure in the scene: it is usually called *blob*.
In the case of moving camera, segmentation is interwined with motion estimation.

2.1 Motion-based image segmentation

Motion-based image segmentation classifies a pixel as belonging to the *background* of the scene or to a moving structure of *foreground*. When the camera is still (the background is also still), the typical approach for discriminating moving objects of a certain frame I_t is called *change detection* [39]. It

consists in comparing the current frame against a background model [5, 18, 19, 26] with the so-called *background subtraction* method: the idea is to subtract the current image from a reference one which is assumed to model the static scene, enphasizing non-stationary or new objects

$$\Delta_t(i,j) = \mid I_t(i,j) - B(i,j) \mid$$

The pixel classification takes place by thresholding the map of changes (see Figure 1):

$$M_t(i,j) = \begin{cases} 1 & \text{if } \mid \Delta_t(i,j) \mid > \tau \\ 0 & \text{otherwise} \end{cases}$$

The role of the background is fundamental since it guarantees robustness against the typical problems

(a) (b) (c)

Figure 1: A reference frame (a) represents the background in the change detection method: the current scene (b) is subtracted from it to detect the moving foreground (c).

that a video processing application faces:

- local and global illumination changes (shadows and highlights);
- static background variations occuring after the modeling phase (object added or removed) or multiple backgrounds (due, for example, to waving trees).

2.1.1 Background modeling

The change detection methods for motion detection differ from each other mostly in the way the background model is built. The simplest way is by averaging a sequence of background images, i.e. frames without foreground moving objects [49, 22]. In many pactical cases, however, such a sequence is not available and so a method able to dismiss moving structures is needed. Moreover it is required to be adaptive, coping with the static variations occuring to the background as time passes. Considering complex approaches, many works in literature are based on parametrizing grey level changes over a time space[18, 19, 26]. These methods have been widely incorporated in algorithm with Bayesian framework [29], mean-shift analysis [38] and region-based information [8]. These kind of techniques, together with the ones based on probability estimation (see [14, 23] as examples), in spite of their accuracy in terms of resulting segmentation, are not suitable for an applied videosurveillance framework, where the computational efficiency plays a fundamental role.
Most of the methods applied in practice are based on simple incremental strategy combined with the output of the change detection process, so that pixels laying in moving areas are discarded: starting from an average of the first N frames or, alternatively, an initial empty background (in both cases we call it B_0), the estimate at time $t \geq 1$ is;

$$B_t(i,j) = \begin{cases} \alpha \dot{B}_{t-1}(i,j) + (1-\alpha)\dot{I}_t & \text{if (i,j) is classified as static pixel} \\ B_{t-1}(i,j) & \text{otherwise} \end{cases}$$

where $0 \leq \alpha \leq 1$ controls how fast new structures are included in the background and the pixel classification is performed using the change detection discussed in the previous section. The procedure finishes when every pixel is assigned with a value but it is advisable to periodically update the reference frame following the same procedure. A qualitative evaluation of this method will be reported in Section 6.
Even if very effective in many practical cases thanks to the computational efficiency which makes them suitable for real-time applications, the methods of background modelling analyzed so far do not suffice in presence of more difficult conditions. The intersting work [27] proposes a real-time algorithm for foreground-background segmentation based on the use of codebooks [9] which captures structural background variations due to periodic-like motion over a long period of time under limited memory. In Section 6 the method will be described in depth and we will discuss the benefit coming from its employment, enphasizing the capability to deal with both dynamic and multiple backgrounds and illumination changes preserving the efficiency from the computational standpoint.

2.1.2 Motion segmentation with moving camera

The change detection approach fails in presence of a moving camera, since in that case every pixel in the image appears to move. The situation is even more complicated if both the camera and the scene are dynamical: in this case two different kind of motion should be detected in the sequence, the first coming from the 3D camera movements, often called *ego motion*, the second caused by the scene changes.
The typical approach against this setting is to compute the ego motion, using for instance the optical flow method (see Section 2.2), to identify the corresponding object which is regarded as the dominant one, and then to exclude it from the region of analysis to repeat again the process on the remaining area.
Among the other it is worth mentioning the work proposed in [24] which purpose is to detect and track occluding and transparent moving objects using temporal integration and without assuming motion constancy. The more complicated case of moving camera (as in the case of vibrations which cause significant motion changes between frames) is taken into account. The hierarchical estimation framework proposed in [2] allows to compute different representation of motion information, obtaining a global model that constraints the overall atructure of the motion estimated.

2.2 Dense motion models

If no a priori information about the acquisition setting is available, or we are interested in a more global study of the motion, the concept of optical flow allows us to obtain dense motion field, being computed for each pixel of images sequence.
Optical flow is the apparent motion of the image brightness pattern, given the assumption that the image brightness is continuous and differentiable as many times as needed in both the spatial and temporal domain. It approximates the physical motion of objects in the 3D world in the sense that it is not always reflected in gray value or color changes in the corresponding images and the gray value changes are not always due to motion.
Starting from the hypothesis of image brightness constancy

$$\frac{d}{dt}(x(t), y(t), t) = 0 \text{ with } I = (x(t), y(t), t)$$

we obtain

$$I_x(u) + I_y(v) + I_t = 0$$

where (u, v) is the motion field, the 2D vector of velocities of image points which can be interpreted as the projection of the 3D velocity filed on the image plane.
It follows that the optical flow results in a vector field subject to the constraint

$$(\nabla I)^t \cdot \vec{v} + I_t = 0$$

This allows to measure just the component of optical flow in the direction of the intensity gradient: this ambiguity is known as *aperture problem*. The typical approach to obtain the remaining information is adding *further constraints* or *further assumptions*. The proposed methods for determining optical flow include phase correlation, block-based methods and differential methods among which it is worth mentioning the very popular by **Lucas** and **Kanade** [32] and **Horn** and **Schunck** [21].
Figure 2 shows possible examples of optical flow estimation. On the first one the camera was moving, thus optical flow captures mostly its motion. An analysis of input frames reveals that the botton-right image part is covered by a saturation area which afflicts the results. On the second case instead, the camera was almost still while a person was walking, thus in the result one can appreciate how, in such area, the estimation is more dense than elsewhere.

2.3 Sparse motion models: local features

Althought very effective in many practical applications, dense estimations of motion field are often too costly from a computational standpoint: in this situation a sparse approximation of motion leads to compact estimates with a limited loss of information.
Given the assumption of small spatial and temporal differences between consecutive frames, structure and motion recovery can proceed by first extracting features and then using them to compute image matching relations, which are the part of the motion that can be computed directly from image correspondences. The detection step allows us to reduce the amount of information and the workload, and also avoids the aperture problem, but, on the other side, this stage requires an accurate and reliable location which is proved to be a non-trivial task.
This choice is crucial to avoid the aperture problem, looking for sparse solutions only in points where local analysis is not ambiguous. Tipically they consist of two main steps:

1. feature **extraction** (e.g. corner [20], DoG features [31])
2. feature **matching** over consecutive frames (tracking)

Thus, a typical algorithm for sparse motion estimation results in

1. to extract corners from frame I_0
2. for each subsequent frame I_i
    - to compute the displacements between corners from the current frame to the successive one
    - to check the needed for new corner extraction (goto 1)

2.4 Summary of low-level measurements

The vision methods presented so far can produce different kind of descriptions which can be summarized as follows:

- pixel-based descriptors

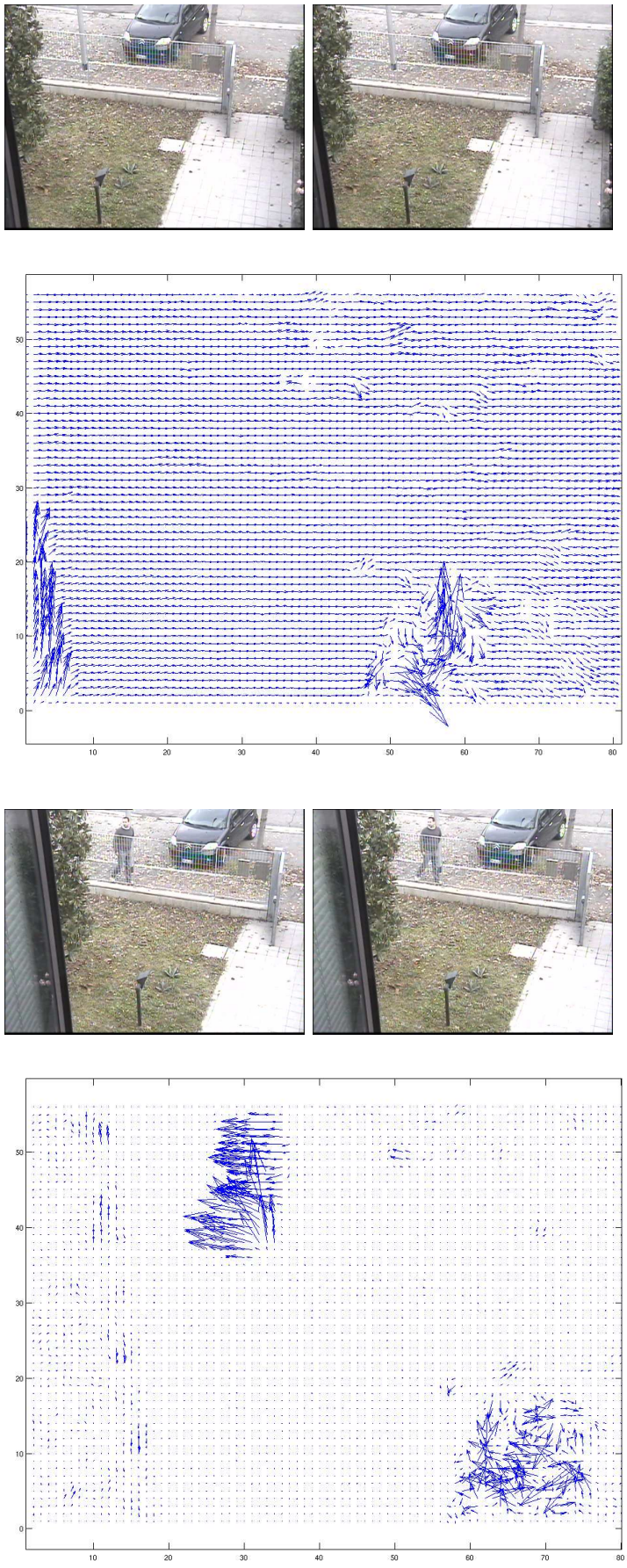


Figure 2: Examples of optical flow results.

  - global approach: dense estimation of motion field → optical flow
  - local approach: sparse estimation of motion field → features extraction
    * motion information (sparse approximation of motion fiels)
    * feature description
  - information about groups of pixels moving coherently
- blob-based
  - global features
    * geometrical information: position, area, perimeter, ...
    * color
    * texture
    * ...
  - local features to be computed within the blob area

3 Intermediate descriptions for dynamic cues

Low-level measurements coming from a pure video analysis phase are commonly used as a first step in many complex vision systems. Depending on the context, however, further analysis may require the compression and the arrangement of the data, moving to higher-level descriptions more connected to the requirments of each specific application. A fundamental tool to recover basic space-time description of a motion event is *tracking* elements (pixels, features or blob). It consists of two subproblems:

- **trajectory initialization** tipically rely on background subtraction but several recent approaches have started to explore the possibilities of combining tracking with detection [30];
- **target following** is usually addressed by classical tracking approaches strenghtened by the employment of prediction filters.

The Kalman filter [47] is an efficient recursive filter which estimates the state of a dynamic system from a series of incomplete and noisy measurements but it is limited to a linear assumption. However, most non-trivial systems are non linear: in such cases the Extended Kalman filter [16], the Unscented Kalman filter and the Particle filter [25] may help to address the non-linearity problem. Particle filters, also known as Sequential Monte Carlo methods (SMC), are sophisticated model estimation techniques based on simulation. They are usually used to estimate Bayesian models and the advantage with respect to the Kalman tools is that, with sufficient samples, they approach the Bayesian optimal estimate, achieving an higher level of accuracy. The approaches can also be combined by using a version of the Kalman filter as a proposal distribution for the particle filter.
Mean-Shift tracking [7] belongs to the family of kernel-based approaches to the problem: it is a nonparametric estimator of density gradient employed in the joint, spatial-range domain of gray level and color images for discontinuity preserving filtering and image segmentation.
In the remainder of this section we discuss possible approaches to blob and feature tracking.

3.1 Blob tracking

The main goal of object tracking is to determine the correspondences between foreground moving structures segmented from two successive frames of a video. Once the change detection is done, a

Figure 3: Examples of blob tracking: on the left multiple targets, on the right a static blob parts from a moving one.

labeling step is performed (see Figure 3) to assign an identity to each object, which is tracked over the video sequence to collect information about its displacements: as intermediate description, this operation provides a trajectory of observations which can be thougth of as the path follown by an object on the image plane.
It is clear that the only binary map of variations does not suffice in general to correctly associate the data, but a description of each connected component arised after the change detection (blob), is required to provide accuracy and robustness. In the particular case of human motion analysis, the non-rigid structure of people and their interactions makes tracking more difficult.
The main difficulties that a blob tracking system must address are:

- failures of the previous change detection step;
- occlusions, which happen when an object temporaly or permanently hides another;
- splitting of a blob in two or more and union of two or more blobs in just one.

3.1.1 Blob description

Data association is a difficult problem in multi-objects tracking scenarios, due to the presence of many similar and mutually occluding targets: designing a reliable blob description is the first important step to aim at a robust tracking system. The approach based on employing geometrical information, as blob centroid, area and perimeter, elongation, density and so on, in addition to dynamic hints (as velocity), fails in presence of complex conditions or crowded scenes.
Many tracking methods are based either on geometric shape information, such as edges [42], or photometric information, such as color [7]: it has been demonstrated that algorithms incorporating both geometric and photometric information are less susceptible to noise, and improve their performance in cluttered environments. The value of using shape priors has been shown in a variety of tracking contexts. Model-based methods have been used in computer vision for a long time, especially for rigid objects. Deformable templates and dynamical models are effective and powerful methods to model prior shapes and allow for many deformations modes of shapes. Geometric PDEs and variational methods are increasingly used in image segmentation and object tracking: the level set method is an effective framework for implementing these PDEs due to its numerical stability and its ability to cope with topology changes. Deformable models can be applied with success to a number of difficult tracking problems, included tracking in medical images. They often rely on an accurate initialization

phase which is not always available in video-surveillance applications.
Color and texture information are realible tools when dealing with video acquired from a medium distance: in this case, the visual appearance of blobs covers a proper area, so that the amount of measurements extracted suffices to correctly identify the object in the two features spaces. This is especially helpful in cases of intersections or temporal occlusions, when it is needed to re-associate the correct label to each blob involved in the event. Tracking methods exploiting color or texture usually rely on segmentation [50] but an alternative is to embed these information in the description of each blob.

3.1.2 Modeling time series

At the most basic level, time series modelling consists of building a probabilistic model of the present observation given all past observations

$$p(y_t|y_{t1}, y_{t2}, ...)$$

Because the history of observations grows arbitrarily large it is necessary to limit the complexity of such a model. There are essentially two ways of doing this. The first approach is to limit the window of past observations. Thus one can simply model $p(y_t|y_{t1})$ and assume that this relation holds for all t. This is known as a first-order Markov model [40]. A second-order Markov model would be $p(y_t|y_{t1}, y_{t2})$, and so on. Such Markov models have two limitations: first, the influence of past observations on present observations vanishes outside this window, which can be unrealistic. Second, it may be unnatural and unwieldy to model directly the relationship between raw observations at one time step and raw observations at a subsequent time step. For example, if the observations are noisy images, it would make more sense to de-noise them, extract some description of the objects, motions, illuminations, and then try to predict from that. The second approach is to make use of latent or hidden variables. Instead of modelling directly the effect of y_{t1} on y_t, we assume that the observations were generated from some underlying hidden variable x_t which captures the dynamics of the system. We usually call this hidden variable x the state variable since it is meant to capture all the aspects of the system relevant to predicting the future dynamical behaviour of the system. In order to understand more complex time series models, it is essential that one be familiar with statespace models (SSMs) and hidden Markov models (HMMs). These two classes of models have played a historically important role in control engineering, visual tracking, speech recognition, protein sequence modelling, and error decoding. They form the simplest building blocks from which other richer time-series models can be developed.

3.2 Feature tracking

The typical choice is to track corner points [41], which do not suffer from the aperture problem [43]: since they are defined as points in which the signal changes in almost two directions, their motion can be completely reconstructed. The simpler way to perform feature tracking is computing correspondences, either by comparing an image patch centered around the feature or by other descriptors (such as SIFT), between successive frames I_{t-1} and I_t: for each corner extracted at time $t-1$ we check whether it is detected also at time t.
Often prediction filters, as Kalman and Particle filters, are used to make tracking more robust, similarly to what described before in relation to blob tracking.

3.2.1 Feature descriptors

Feature tracking may take advantage from appropriate descriptions: local photometric descriptors computed at interesting points of the image have proved to be very successful in applications such as matching and recognition. A method for feature description tipically represents an interesting point detected in an image in a compact way, summarizing important properties. The main goal is to obtain robust descriptors able to identify such point in further steps on analysis. A simple descriptor is a vector of pixel values which, despite its simplicity, is a high dimensional solution. In the last years a number of methods have been proposed and improved, making them suitable for applications with accuracy and compactness requirements. In [33] feature descriptors are classified in:

- *Distribution based descriptors* is histogram of pixel intensities computed over an image patch;
- *Non-parametric transformations* rely on local transform based on non-parametric statistics exploiting information about ordering and reciprocal relations between data;
- *Spatial-frequency techniques* describe the frequency content of an image using Fouries analysis, Gabor filters and wavelets, among the others;
- *Differential descriptors* involve the computation, using gaussian functions, of a set of image derivatives computed up to a given order to approximate a point neighborhood;
- *Sift vectors* [31] deserve to be mentioned apart since they have been used in a number of applicative fields, showing great performances both from the computational standpoint and the results.

3.2.2 Space-time features models

Similarly to the case of images, recently some local descriptors for space-time modeling have been proposed that could be seen as abstractions of the information content derived by the process of tracking local patterns.
Local space-time features capture local events in video and can be adapted to the size, the frequency and the velocity of moving patterns. It has been demonstrated how such features can be used for recognizing complex motion patterns, building a video representations and integrating it with some classification schemes for recognition. The representation could be interesting points extracted in the 3D space (space and time) [28] or models derived from features trajectories along the image sequence [12, 10, 11, 17].
The number of features used to describe the events and their size could make less effective the representation from a computational point of view. Dimensionality reduction (see Section 4) can help to compress the data.

4 Learning methods for data classification and clustering

4.1 Supervised learning

Many computer vision applications, as automatic image and video annotation and categorization, require to address a classification problem which can be better dealt following a learning from example approach [44]. The standard formulation of a learning framework is stated as follow: given a set of labeled examples

$$(x_i, y_i) \text{ with } i = 1, ..., N$$

where x_i are the input data and y_i the associated output values (the labels in a classification setting), we look for a function f^* such that

$$f^*(x_{new}) \sim y_{new}$$

i.e. given a new example, f^* is able to correctly approximate its output. The problem is faced with a statistical approach, assuming that the given examples are generated by an unknown probability distribution $P(x, y)$. Using the regularization technique [37], the function we are interested to can be written as

$$f^* = argmin_{f \in H_K} \frac{1}{N} \sum_{i=1}^{N} L(x_i, y_i, f) + \lambda_A \|f\|_K^2$$

where H_h is an appropriate hypotesis space, L is a loss function that evaluates the cost of approximating y_i with $f(x_i)$ and λ a regularization parameter that tunes the trade off between the empirical term and the penalty.
Ideally, in order to obtain a good generalizing solution, we should gather a high number of data w.r.t. the input space size. This is particularly crucial in the case of images or videos to which we usually associate high dimensional feature vectors.
However, in practice, labeling examples is not always simple: their acquisition often requires a skilled human agent to manually classify training examples. The cost associated with the labeling process thus may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised and unsupervised learning can be of great practical value: focusing on our system, the lack of examples labeled as anomalies leads naturally to an unsupervised framework, which will be briefly discussed in the remainder of this section.

4.2 Unsupervised learning

In an unsupervised setting, a data set of input $\{x_1, x_2, ..., x_t\}$ is gathered but there is no a priori information about their outputs. The goal is to build representations of the input that can be used for decision making and predicting future inputs: in a sense, unsupervised learning can be thought of as finding patterns in the data. Two very simple classic examples of unsupervised learning are clustering and dimensionality reduction.
Almost all work in unsupervised learning can be viewed in terms of learning a probabilistic model of the data, i.e. to estimate a model that represents the probability distribution for a new input $\{x_t\}$ given previous inputs $\{x_1, x_2, ..., x_t\}$

$$P(x_t|x_1, ..., x_{t1})$$

In simpler cases, where the order in which the inputs arrive is irrelevant or unknown, the machine can build a model of the data which assumes that the data points are independently and identically drawn from some distribution $P(x)$. Such a model can be used for outlier detection or monitoring and for classification.

4.2.1 Clustering and dimensionality reduction

The framework described above can be applied to a wide range of models: no single model is appropriate for all data sets, it is advisable, instead, to develop models which are appropriate for the data set being analysed, and which have certain desired properties. For example, for high dimensional data sets it might be necessary to use models that perform dimensionality reduction. Of course, ultimately, the machine should be able to decide on the appropriate model without any human intervention, but to achieve this in full generality requires significant advances in artificial intelligence. A tentative

summary of probabilistic models that are defined in terms of some latent or hidden variables must include **Factor analysis**, **Principal component analysis**, **Indipendent component analysis**, **Mixture of gaussians** and **K_Means**. These models can be used to do dimensionality reduction and clustering, the two cornerstones of unsupervised learning.

4.3 Manifold learning

A large number of data such as images and characters under varying intrinsic principal features are thought of as constituting highly nonlinear manifolds in the high-dimensional observation space. Visualization and exploration of high-dimensional vector data are therefore the focus of much current machine learning research. However, most recognition systems using linear method are bound to ignore subtleties of manifolds such as concavities and protrusions, and this is a bottleneck for achieving highly accurate recognition. This problem has to be solved before we can make a high performance recognition system. Recent years have seen progress in modeling nonlinear manifolds. Rich literature exists on manifold learning. On the basis of different representations of manifold learning, this can be roughly divided into four major classes:

- projection methods
- generative methods
- embedding methods
- mutual information methods

5 Coupling vision task with learning approach

The very interesting work by S. Avidan [1] integrates the classification abilities of an SVM into an optical flow tracker. Instead of minimizing an intensity difference between successive frames, the author maximizes the SVM classification score. To account for large motions between successive frames he builds pyramids from the support vectors and uses a coarse-to-fine approach in the classification stage. A few years earlier a work integrating optical flow to eigenimages was presented in [3]: the paper describes an approach for tracking rigid and articulated objects using a view-based representation. Viola *et al.* [45] combine their fast AdaBoost algorithm with brightness and motion patterns to detect pedestrians: they use a detection style algorithm that scans a detector over two consecutive frames of a video sequence. The detector is trained to take advantage of both motion and appearance information to detect a walking person.

In the contexts of close-range event detection most works are based on fairly complex representation and recognition schemes: Bregler [6] uses many levels of representation based on mixture models, EM, and recursive Kalman and Markov estimation to learn human dynamics. The Pfinder system [49] adopts a maximum a posteriori probability approach to detect and track humans using simple 2D representations of head and hands using a multi-class statistical model of color and shape, in a wide range of viewing conditions. W^4S [19] is a real-time visual surveillance system for detecting and tracking people while monitoring their activities in an outdoor environment. It operates on monocular grey-scale video imagery or on video imagery from an infrared camera. The method employs a combination of shape analysis and tracking to locate people and their parts and to create models of people's appearance, so that they can be tracked through interactions such as occlusions. The work presented by Pittore *et al* [36] is one of the first attempts of coupling dynamic information with statistical learning: SVMs are used to classify events out of a small range of possible actions

using temporal descriptions based on blob trajectory [46]. They discuss the benefits of using SVM for performing effective classification of events and for building noise tolerant representations.

6 Current state of work

I spent the first year of my PhD acquiring the theoretical background knowledge for appropriate handling video processing problems in the framework of interest. I went in particular into the theory of motion segmentation and blob-based descriptions (see Section 1), since I was already familiar with feature-based representations thanks to past works [12, 10, 11].
From the application standpoints, my activities can be summarized as follows:

- analysis of background modeling methods;
- design and analysis of robust blob description to address the tracking problem in presence of occlusions;
- an application to face recognition as a test of adaptability of a view-based architecture of video content modeling.

6.1 Background segmentation for change detection

The incremental background modeling discussed in Section 2.1.1 has been proved to be very effective in many practical situations, when fast processing is required: it is, among the other methods, one of the best compromise between computational efficiency and quality of results, as one can appreciate in Figure 1.

Figure 4: The comparison between foreground segmentation obtained by incremental and codebook methods show how the second better deal with multiple backgrounds.

The incremental background modeling fails in presence of multiple or dynamic backgrounds: to be able to handle also with outdoor environments, we considered the codebook approach proposed in [27], in which the background is built on a pixel-wise base. A comparison between the two methods is shown is Figure 4: on the left, the segmentation obtained by the incremental method cannot deal with multiple backgrounds (in the scene waving trees were present), while on the right the foreground object is correctly detected.
The information collected to describe the dynamic of each pixel allow also to distinguish between

different levels of background: traditional approaches consider static variations of the scene (objects added or removed) as variations with respect to the reference image, in other words as motion. But if we consider a structure added to the scene and stable over a proper interval of time, it should be interpreted as a further layer of the static background. A visual representation of this situation is visible in Figure 5.

Figure 5: Different layers of static background: an object (magenta) is added to the scene, becoming a stable variation.

6.2 Descriptions design for robust tracking

We decided to adopt the incremental strategy for background modeling in the context of a tracking framework: the focus was on finding robust blob description, able to solve data association after occlusion events. The motion segmentation in Figure 1(c) is the result of a few post-processing steps after the change detection:

- shadows are detected, so that they are not included in the motion ares. We used a RGB color-based method, which classifies a pixel as shadow if the variations of the three channels are similar;
- a set of morphological operation, *erosion*, *dilation*, *closing* and *aperture*, are applied in order to better approximate shape and area of a blob.

For the tracking part, consisting of a correspondenced-based step and the use of a predictive filter, we focused mainly on indoor environment: a number of blob information has been taken into account, to determine the most reliable and stable in time. The results can be summarized as follows:

- althougth it is a basic descriptor, the blob position is the most reliable and efficient when the situation is easy;
- geometrical information, as height and width, area and perimeter, are not stable but they can be used to reject noisy or less meaningful blobs;
- color information is useful when dealing with mid-distance sequences. Histograms intersection and correlation resulted to be appropriate similarity measures;

- results on the use of texture seems to be not primising.

We tested the performances of this system considering its capability to recover after occlusion events: exploiting low-level measurements allow to address the problem when two blobs merge, while the case of intersection between more than two blobs need a further improvement of the system.

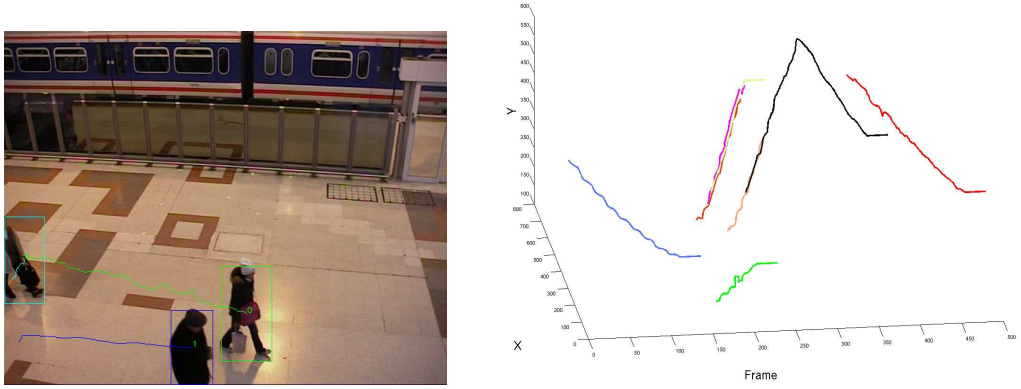


Figure 6: Blob tracking provides a trajectory of a person. On the left, a 3D visualization of such trajectories.

We then tried to cope with the problem of detecting abandoned object, connecting it, when possible, with the blob (person) that abandoned it: this problem was tackled by an analysis relating to the "stability" of a blob for a certain temporal interval, while spatial proximity considerations were used to detect the blob from which the abandoned package separated.

6.3 A view-based approach to face validation and recognition

Closely related to the analysis of the dynamic of a scene is the understanding of its content from a semantic viewpoint. In the video-surveillance framework the main focus is on people, therefore face validation and recognition are important topics. In the case of face recognition from video it is common practice to obviate to the lack of signal quality due to the use of video-surveillance devices to the redundancy of video information. A possible approach is to integrate the object recognition architecture proposed in [12, 10, 11] to a face detection method [13].

The method works on a sequence of face images, produced by the face detection phase: the resulting model is a compact description of that face itself and can be used in further steps of recognition. The recognition is performed comparing the model of a subject against a test video, described in the same way, following the matching strategies proposed in [12].
To check the effectivness of the approach, we collected a dataset of 18 people, acquiring 4 videos for each: the videos were acquired in different conditions to model the typical variations of people movement and appearance. The training sequences (one per person) are acquired on a plain background, while the subject was asked to rotate the head so that different views of the face are gathered.
A validation step based on ROC curve method (Figure 8) has been performed on a second set of videos to determine an appropriate choice for the parameter involved in the matching phase (a threshold on the number of robust matches). Also in this case a video per person has been acquired: the background

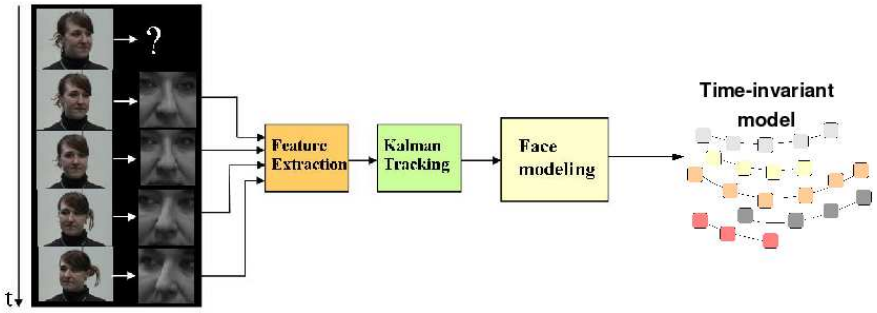


Figure 7: An outline of our system for view-based face recognition.

is still plain but we asked the subject to change face expression and let free movement.
We obtained very promising results testing the system on 36 test videos, taken with the same conditions above mentioned but changing elements of visual aspect (as, for example, wearing glasses or hats). Over a total of 365 experiments we got

$$TruePositiveRate = \frac{TruePositive}{FalseNegative + TruePositive} = 0.81$$

$$FalsePositiveRate = \frac{FalsePositive}{FalsePositive + TrueNegative} = 0.15$$

We noticed how the system requires a reliable face tracking: if one or more faces are missed, the tracking suffers and the resulting trajectories may be unstable or less accurate.

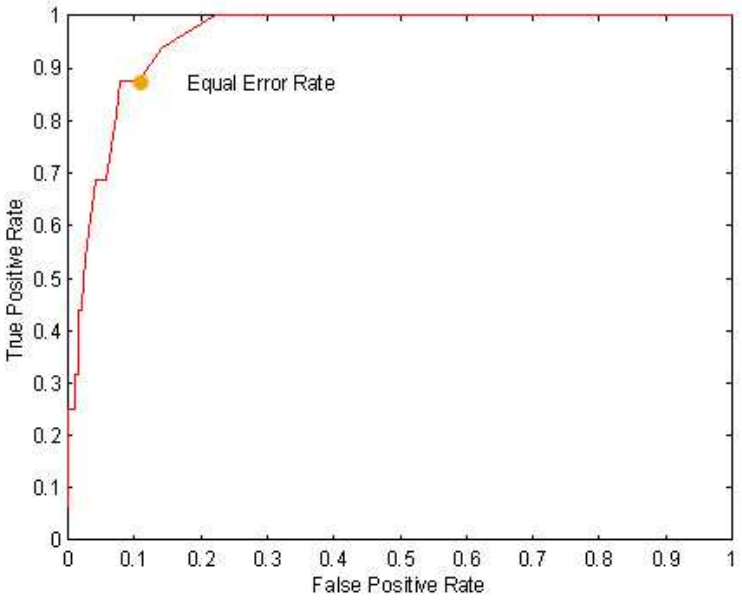


Figure 8: the ROC curve obtained varying the values of the matching parameter.

Since they are at the basis of the final modeling phase, it is clear that the face detection is crucial. We think that our results could be improved solving this problem and the variations to the architecture are currently on-going: a Kalman filtering-based tracking will be soon integrated in the face detection to ensure the system against failures.

7 Framework of the project

The final goal of our project is to develop a "decision making" architecture able to combine

- video processing methods for scene perception and analysis, both in space and time;
- statistical learning techniques to classify dynamic events

in order to perform event modeling and anomaly detection. The purpose is to design systems able to process a video signal without temporal contraints and making decisions possibly with a limited human supervision. This assumption leads to a list of desired characteristics for our methods:

- the absence of temporal limit binds the system to have an appropriate spatial complexity;
- robustness over time of each processing component is desirable;
- adaptability allows to fit the system to different settings, on which the notions of "common events" and "anomaly" depend.

7.1 Our approach

A general architecture for our video analysis process is shown in Figure 9: it is composed by main components which will be better defined depending on the specific application.

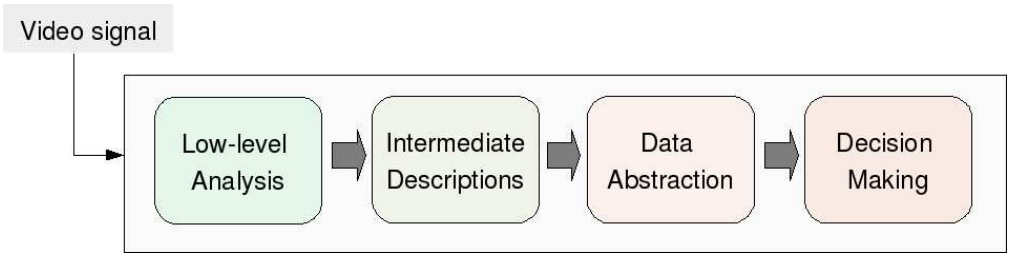


Figure 9: A general architecture for video analysis system.

The objectives of the thesis cover a wide and rather ambitious spectrum. Following Figure 9, they range from robust data extraction (both in space and time), require meaningful intermediate descriptions of objects, individual actions, and global dynamics, aim at understanding the data structure from long time observations, also they aim at becoming automatic tools for decision making.
In this context many different smaller problems could be casted, and a possible hierarchical structure that hosts problems at various abstractions levels could be envisaged. At this stage of the thesis it is still difficult to formalise such a complex stream of information, but it is clear that by changing the focus of attention of the general problem a number of better defined, simpler problems take shape. In the following we detail some examples of those, with the ambition of being able in the near future to cast them in a more general formulation.

1. **frame-based scene labeling**, whose aim is to estimate the global state of a scene on the basis of low-level measurements coming from a relative small temporal interval (a few frames). Possible labels can concern the level of dynamic clutte in terms of people present (empty, not empty, crowded, ...);
2. **video-based scene labeling** based on segmenting foreground from background areas. The study is still scene-based but the focus is on two main questions

   a. how many (not better defined) structures are moving in the scene that I am observing?

   b. how can I approximate and describe their overall dynamics or appearance?

   This also concerns the study of social behaviours between groups of people [15].
3. **appearance-based blob descriptions** obtained starting from features extracted with image and video processing methods. The description may help on one hand to classify the blob (human, car, luggage, a specific person), on the other to identify it in a recognition phase (to decide whether a blob has been already observed in the same scene in previous times);
4. **dynamics-based blob description** based on describing a blob as a moving entity, using trajectory, velocity, and so on. This requires to employ methods for intermediate descriptions.

The four possible views can be seen as parts of a hierarchical architecture, according to the dependences graph in Figure 10 and, depending on the case, one may be seen as a pre-processing step of one of the others.

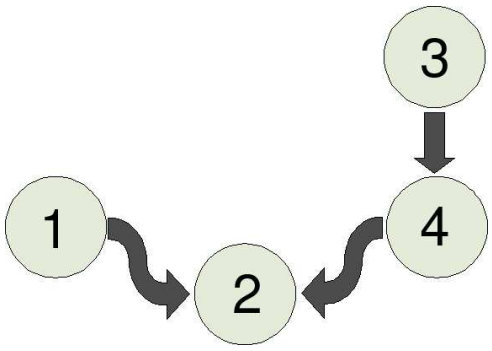
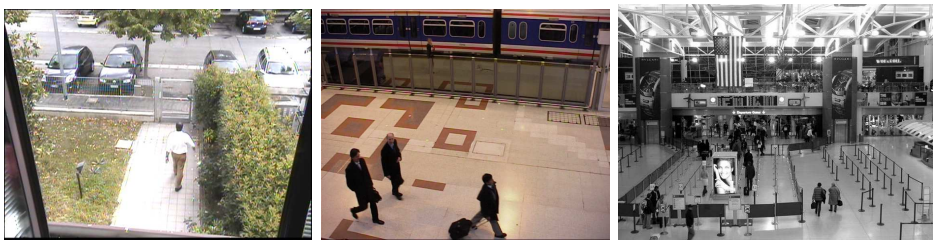


Figure 10: Dependencies graph of the hierarchical architecture for video analysis.

7.2 Scenarios of interest

Concerning the possible scenarios of interest, on a first period we have concentrated mainly on medium distance videos of indoor and outdoor environments acquired with a static camera(Figure 11 (a) and (b)), where the focus is on detecting moving objects under different scene conditions. Onward, we also want to consider sequences of indoor crowded scenes (the camera is still static) taken from an higher distance (Figure 11 (c)): in this case the attention moves to a more global analysis of the motion.
In order to cope with automatic modeling of dynamic behaviours a high quantity of video is needed, acquired in a variety of conditions over a long time period of time (days, weeks). To this purpose we are collaborating with Imavis s.r.l. [1] which is supplying us videos acquired from their prototype surveillance systems.

[1] www.imavis.com



(a) (b) (c)

Figure 11: Possible scenarios of interest: (a) an outdoor environment observed from a medium distance with presence of moving structures of different type (humans and cars); (b) indoor acquisitions where data association and tracking must cope with the typical interactions between human ; (c) a crowded and complex scene in which the focus is studyng the motion from a global view-point.

8 Objectives

In my research plan theoretical and application aspects are tightly coupled: my purpose is to achieve a deeper understanding of vision and statistical techniques involved in my work, proceeding at the same time with the implementative and experimental parts. The development of some tasks may require to deepen specific theorethical aspects or building other applications. For these reasons, I plan my future work as shown in Figure 12: the top diagram is an overall of the next two years, while on the bottom there is a more detailed description of the first year.
In the remainder of this section I detail short term plan and long term work.

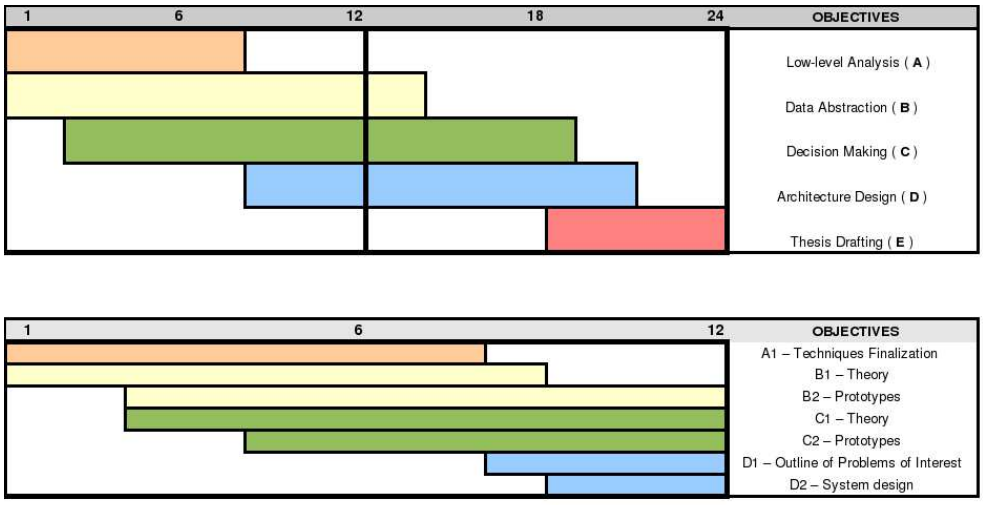


Figure 12: An overview of objectives and details about short term work plan.

8.1 Short term objectives

On the next months I will study more in depth statistical learning theory, scheduling my work as follows:

- first, I will analyze techniques to model temporal series and I will develop appropriate task to experience their use on our datasets;
- then I will move to unsupervised and manifold learning theory, studying and testing their applications to problems of interest.

After this initial studying phase, I plan to achieve a proper understanding of learning aspects so that I can outline specific tasks and systems able to solve them. At the same time, this will require to finalize the works on video processing we have done so far, to improve robustness and efficiency where it is needed.

8.2 Long term objectives

Long term objectives refer to the remainder of my PhD program. Althougth theoretical aspects could require further analysis which will be thus protracted, my purpose is to focus in designing solutions for specific target: from the architectural standpoint, systems should respect the general outline discussed in Section 7 but each element will be appropriately developed according to the requirements coming from the specific task. Our ambition is to have at the end of my PhD a complex architecture which satisfies adaptability properties and able to cope with high-level video modeling and decision making.

8.3 Possible structure of my thesis

A possible structure of my thesis is the following:

1. *Introduction*
2. *Low-level vision methods*
3. *Higher-level space-time description*
4. *Statistical learning from examples*
5. *Designing problems of interest*
6. *Designing the architecture*
7. *Conclusion*

References

[1] S. Avidan. Support vector tracking. In *Proc. IEEE Conf. CVPR*, 2001.

[2] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, pages 237–252, 1992.

[3] M. Black and A. D. Jepson. Eigentracking: Robust matching and tracking for articulated objects using a view based representation. In *ECCV*, 1996.

[4] A. Bobick and J. Davis. The recognition of human movement using temporal templates. In *IEEE Trans. Pattern Recognition Machine Intelligence*, volume 23, pages 257–267, 2001.

[5] T. Boult and et. al. Frame-rate multibody tracking for surveillance. In *IUW*, pages 305–308, 1998.

[6] Bregler. Learning and recognizing human dynamics in video sequences. In *Proc IEEE Conf. CVPR*, 1997.

[7] D. Comaniciu, V. Ramesh, , and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. CVPR*, volume 2, pages 142–149, 2000.

[8] M. Cristiani, M. Bicego, and V. Murino. Integrated region- and pixel-based approach to background modeling. In *Proceedings of IEEE Workshop on Motion and Video Computing*, 2002.

[9] G. Csurka and et. al. Visual categorization with bags of keypoints. In *ECCV*, 2004.

[10] E. Delponte, N. Noceti, F. Odone, and A. Verri. Appearance-based 3d object recognition with time-invariant features. In *ICIAP*, 2007.

[11] E. Delponte, N. Noceti, F. Odone, and A. Verri. The importance of continuous views for real-time 3d objects recognition. In *ICCV*, 2007.

[12] E. Delponte, N. Noceti, F. Odone, and A. Verri. Spatio-temporal constraints for matching view-based descriptions of 3d objects. In *WIAMIS*, 2007.

[13] A. Destrero, C. De Mol, F. Odone, and A. Verri. A regularized approach to feature selection for face detection. In *ACCV*, 2007.

[14] A. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *ECCV*, 2000.

[15] A. French, A. Naeem, I. Dryden, and T. Pridmore. Using social effects to guide tracking in complex scenes. In *AVSS*, 2007.

[16] A. Gelb. Applied optimal estimation. In *MIT Press*, 1996.

[17] M. Grabner and H. Bischof. Object recognition based on local feature trajectories. In *I cognitive vision works.*, 2005.

[18] W. E. L. Grimson and et. al. Using adaptive tracking to classify and monitor activities in a site. In *Conference on Computer Vision and Pattern Recognition*, pages 22–29, 1998.

[19] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Who? when? where? what? a real-time system for detecting and tracking people. In *the third IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society Press*, pages 222–227, 1998.

[20] C. Harris and M.Stephens. A combined corner and edge detector. In *Plessey Reasearch Roke Manor, UK, The Plessey Company plc.*, 1988.

[21] B.K.P. Horn and B.G.Schunk. Determining optical flow. In *Artificial Intelligence*, volume 17, pages 185–204, 1981.

[22] T. Horprasert, D. Harwood, and L. S. Davis. A statisctical approach for real-time robust background subtraction and shadow detection. In *IEEE Frame-Rate Applications Workshop*, 1999.

[23] Y. Hsu, H. Nagel, and G. Rekers. New likelihood test methods for change detection in image sequences. In *Comput. Vis. Graph. Image Process.*, volume 26, pages 73–106, 1984.

[24] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. In *IJCV*, volume 12(1), pages 5–16, 1994.

[25] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. In *IJCV*, 1998.

[26] T. Kanade and at. al. Advances in cooperative multi-sensor video surveillance. In *IUW*, pages 3–24, 1998.

[27] K. Kim, T. C. Chalidabhongse, D. Harwood, and L. S. Davis. Real-time foreground-background segmentation using codebook model. In *www.sciencedirect.com*, 2005.

[28] I. Laptev and Tony Lindeberg. Space-time interest points. In *Proc of the IEEE ICCV*, 2003.

[29] D. S. Lee, J. J. Hull, and B. Erol. A bayesian framework for gaussian mixture background modeling. In *IEEE International Conference on Image Processing*, 2003.

[30] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

[31] D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[32] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging understanding workshop*, pages 121–130, 1981.

[33] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 27(10), 2005.

[34] N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 22, pages 831–843, 2000.

[35] S. Ong and S. Ranganath. Automatic sign language analysis: a survey and the future beyond lexical meaning. In *IEEE Trans. on PAMI*, volume 27(6), 2005.

[36] M. Pittore, M. Campani, and A. Verri. Learning to recognize visual dynamic events from examples. *IJCV*, 2000.

[37] T. Poggio, T. Evgeniou, and M. Pontil. Regularization networks and support vector machines. In *Advances in Computational Mathematics*, 1999.

[38] F. Porikli and O. Tuzel. Human body tracking by adaptive background models and mean-shift analysis. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2003.

[39] et al. R. J. Radke. Image change detection algorithms: A systematic survey. In *IEEE Trans. Image Processing*, volume 14(3), pages 294–307, 2005.

[40] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[41] J. Shi and C. Tomasi. Good features to track. In *IEEE CVPR*, pages 593–600, 1994.

[42] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *ICCV*, 2001.

[43] E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision*. 1998.

[44] V.N. Vapnik. Statistical learning theory. In *Wiley-Interscience*, 1998.

[45] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc of the IEEE ICCV*, 2003.

[46] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *ICCV*, page 257ff, 2003.

[47] G. Welch and G. Bishop. An introduction to the kalman filter. 2006.

[48] W.Hu and T.Tan. A survey on visual surveillance of object motion and behaviors. In *IEEE Trans. on Systems, Man, and Cypernetics*, volume 34(3), 2004.

[49] C. R. Wren, A. Azarbayejiani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *IEEE Trans. on Pattern Analysis and Machine Intelligence, IEEE Computer Society Press*, volume 19(7), 1997.

[50] Y. Wu and T. S. Huang. Color tracking by transductive learning. In *Proc. of IEEE Conf. on CVPR*, volume I, 2000.

[51] Y. Zhang and Q. Li. Active and dynamic imformation fusion for facial expression understanding from image sequences. In *IEEE Trans. on PAMI*, volume 27(5), 2005.