A COMPUTER VISION SYSTEM FOR IN-HOUSE VIDEO SURVEILLANCE

R. Cucchiara, C. Grana, A. Prati, R. Vezzani

Dipartimento di Ingegneria dell'Informazione
Università di Modena e Reggio Emilia, Italy
Via Vignolese 905, Modena Italy
{cucchiara.rita,grana.costantino,prati.andrea,vezzani.roberto}@unimore.it

Abstract

In this paper, we focus on the framework of *In-House Video Surveillance* to control the safety of people living in domestic environments. In this context, common problems and general purpose computer vision techniques are discussed and implemented in an integrated solution composed of a robust moving object detection module, which is able to disregard shadows, a tracking module designed to handle large occlusions and a posture detector. Shadows, large occlusions and deformable models of people are the key problems that must be taken into account for in-house surveillance. Moreover, the requirements of high speed reactions to dangerous situations and the need to implement a reliable and low cost tele-viewing system, led to the introduction of a new multimedia model of semantic transcoding, capable of supporting the different user's requests and meeting the constraints of their devices (PDA, UMTS cellular phones, …).

Our application context is the emerging area of *Domotics* (from the Latin word *domus* that means "home" and informatics) and, in particular, indoor video surveillance, where people with some difficulties (elderly and disabled people) can now live with a sufficient degree of autonomy, thanks to the strong interaction with the new technology that can be distributed in the house, with affordable costs and high reliability.

1 Introduction

Tele-presence and tele-viewing are essential for the future systems of people's health care and in-house video surveillance. People will interact in an ever-increasing manner with sensors distributed throughout the house, expressing their requests and desires in explicit and implicit ways. Thus, new paradigms of communication and human-machine interaction will be developed using domotic technology. On one hand, a large research activity is devoted to new explicit human-to-home communications, in particular for disabled or elderly people and children. On the other, implicit communication will be provided between the home and humans, with networks of sensors capable of detecting dangerous situations and reacting to alert human care operators, relatives and others, only when required (and thus with total respect of privacy issues). Among the sensors, in terms of the amount and value of semantics that can be provided,

the superiority of visual sensors is well known and endorsed by many new prototypes.

The proposal described here was devised in this framework of implicit interaction between humans and houses by means of video processing: people's safety at home can be monitored by computer vision systems that, using a single static camera for each room, detect human presence, track people's motion, recognize behaviour (e.g., using posture), assess dangerous situations completely automatically and allow efficient on-demand remote connection.

Although most of the problems of visual surveillance systems are common to all the contexts, outdoor and indoor surveillance have some different requirements, and, among indoor scenarios, in-house surveillance has some non-negligible peculiarities. Indoor surveillance can, in a sense, be considered as less complex than outdoor: the scene is known a-priori, cameras are normally fixed and are not subject to vibration, weather conditions do not affect the scene, and the moving targets are normally limited to people. Despite these simplified conditions, in-house scenes are characterized by other peculiarities that increase the difficulties of surveillance systems (see Table 1 at the end of the paper):

a) *Multi-source artificial illumination in the rooms* affects moving object detection, due to the generation of *large shadows* connected to the objects. Shadows affect shape-based posture classification and often lead to object under-segmentation. Thus, reliable shadow detection is necessary.

b) Houses are full of *different objects that are likely to be moved*. Object segmentation based on background suppression (usual for fixed camera systems) must react as quickly as possible to these changes in the background. This calls for knowledge-based background modelling.

c) The main targets of tracking algorithms are people with *sudden or quick appearance and shape changes*, *self-occlusions* and *track-based occlusions* (i.e., occlusion between two or more moving people); moreover, tracking must deal with large *object-based occlusions* due to the presence of objects in the scene.

d) Due to the above-mentioned presence of numerous objects in the scene, the *posture classification algorithm* must cope with *partial* (or total, but short-term) *body occlusions*.

e) To ensure the health monitoring of the disabled and elderly, all the time and from all positions, remote access to the cameras must be granted from any type of device. Thus, a *content-based video adaptation* or *transcoding* module must be implemented to also allow access to devices with limited capabilities (such as PDAs or UMTS cellular phones).

Points a) and b) have been thoroughly addressed in the past, proposing statistical and adaptive background models [3, 5, 11, 12] and colour-based shadow-removing algorithms [3, 5, 6, 10]. The topic of human motion capture has also been investigated in the literature [1, 7], often by means of probabilistic and appearance-based tracking techniques, to cope with non-rigid body motion, frequent shape changes and self-occlusions [6, 9]. When several people interact,

overlapping each other, most of the known techniques tend to lose the covered tracks, claiming the presence of a group of people, and possibly restoring the situation after the group has split up [6]. Then, problems arise for keeping the track history consistent before and after the occlusion occurs. For in-house surveillance systems, low-cost solutions are preferable, thus stereovision or 3D multi-camera systems must be discarded. Consequently, algorithms of people's posture classification should be designed to work with simple visual features from a single camera, such as silhouettes, edges and so on [4,7]. Finally, most of the systems for security and safety purposes require remote access for external people (such as relatives or public officers for law enforcement). Thus, on-the-fly semantic video adaptation techniques have been designed in order to allow the user remote access to visual data in an efficient and multimodal way [2, 13].
The contribution of this work is not merely the description of the integrated system, but mainly an in-depth discussion of the trade-off between employing general purpose video surveillance techniques and addressing specific problems of house environments by using tailored approaches

2 The overall system

Our system is structured like client-server architecture, as in Fig. 1. The server side contains several pipelined modules: in in-house video surveillance, motion is a key aspect and, thus, object detection and motion analysis are embodied in the first module. The set of salient visual objects (*VOs*, hereinafter), along with their features (shape, area, colour distribution, average motion vector and so on), is the output of this module. The VOs classified as potential people are tracked along time; the output of tracking is the set of *tracks* representing people and the associated computed features (expressed with probabilistic and appearance-based maps); each track is evaluated to provide posture classification and analyze people's behaviour, in order to identify dangerous events. Thus, a dangerous event, such as a person lying down for long time, is detected by the transition from the state of "standing" to that of "lying", and by permanency in the latter state for a long period of time. The alarms generated can be sent to a control centre or can trigger remote connections with a set of authorized users, which exploit the transcoding server to receive visual information of the objects (people) involved in the dangerous situations. According to the user's requests and the device's capabilities, the transcoding server adapts the video content, modifying the compression rate of the regions of interest, thus sending only what is required and only in the case of a significant event occurring.

2.1. Visual Object Segmentation

As a basic step for further processing, object segmentation must be performed. A peculiarity of in-house surveillance is that segmentation must not be limited to generic moving objects, but also extended to still objects classified as people. In other words, salient objects in our application are both generic moving objects and still people, in order to correctly

analyze people's behaviour and avoid missing dangerous situations, such as falls. Therefore, the aim of this module is twofold: firstly, the detection of salient objects with high accuracy, limiting false negatives (pixels of objects that are not detected) as much as possible; secondly, the extraction of objects by adapting to background changes with high responsiveness, avoiding the detection of transient spurious objects, such as cast shadows, static non relevant objects or noise. To accomplish this, the system segments the current frame into visual objects, their shadows and possible "ghosts", i.e. apparent moving objects typically associated with the "aura" that an object that starts to move leaves behind it [3]. To distinguish between apparent and actual moving objects, a post processing validation step is performed by evaluating the number of pixels changed in the last two frames.

Visual object detection is performed by means of background suppression with pixel-wise temporal median statistics [3]. However, the key novelty of this approach is the exploitation of the type of object (VO, shadow or ghost) associated with each pixel to selectively update the reference background model. Thus, moving shadows and objects are not included in the background, while ghosts are pushed into the background to improve responsiveness.

Moreover, in the case of surveillance of people, the background updating process must take into account whether or not the detected object is a person: if it is classified as a person, it is not used to update the background, even if it is stopped, to avoid the person being included in the background and its track being lost.

The classification of objects into people is based on geometrical property checking (ratio between width and height of the bounding box) and the presence of elliptical, almost pink, connected areas (assumed to be the person's face).

Shadows are detected by assuming that they darken the underlying background, but do not significantly change its colour. Thus, a model of shadow detection in the HSV colour space is proposed as in the following equation:

$$SP_x = \begin{cases} 1 & if \quad \alpha \le \dfrac{I_x.V}{B_x.V} \le \beta \wedge |\, I_x.S - B_x.S \,| \le \tau_S \wedge D_x \le \tau_H \\ 0 & otherwise \end{cases} \tag{1}$$

where I is the current frame and B is the current background model, and where D_x is computed as:

$$D_x = min(|\, I_x.H - B_x.H \,|, 360 - |\, I_x.H - B_x.H \,|) \tag{2}$$

Further details can be found in [3].

The presence of shadows is a critical problem in all video surveillance systems and a large effort has been devoted to shadow detection [8]. This is even more crucial in indoor, in-house surveillance where both artificial and natural light sources interact.

Thus, in indoor environments many shadows are visible and no straightforward assumptions on the direction of the lights can be made. Moreover, most of the shadow detection algorithms exploit a certain number of parameters to obtain the best trade-off between shadow detection and foreground preservation, and to adapt to illumination changes. Ideally,

a good shadow detector should avoid using parameters or, at least, use a fixed and stable set of them. Using the algorithm here described, results are well acceptable outdoors or when the camera acquisition system is good enough (as in the case of Fig. 2 (a) and Fig. 2(b), where white pixels indicate shadow points). Instead, when a low-quality camera is used (as is usual in the case of in-house installation), image quality can prevent us from detecting a good shadow (Fig. 2(c)). In this case, the chosen shadow parameters can lead to over-segmentation (Fig. 2 (d)) or under-segmentation (Fig. 2(e)) and correct segmentation can be hard to achieve.

Our experiments demonstrate that under-segmentation is not acceptable since the shape distorted by cast shadows (as in Fig. 2 (f)) cannot be easily classified as a person, and people walking in a room are easy merged in a single visual object. Conversely, we decided to fix the parameters of shadow detection accepting a possible over-segmentation (as in Fig. 2(d), where the sweater of the person on the left is detected as shadow), devolving the task of managing temporary split components or holes to the tracking module.

2.2 Probabilistic Tracking

During the tracking phase, each VO has to be matched with one of the tracks, which represent the known objects present in the scene. These are described by an *appearance image AI(x)* (the estimated appearance of the object in the RGB space), a *probability mask* $P_M(x)$ (each of its values defines the probability that the point belongs to the track), a motion vector estimated for the next frame, and a depth order. Initially, a Boolean correspondence matrix *C* between VOs and tracks is computed using the Bounding Box Distance, as proposed in [9]. Many types of correspondence can arise: to cope with this, we define the concept of the Macro-Object (*MO*) as the union of VOs associated with the same track T_k. In this way, we search the correspondences between a subset of Visual Objects (that potentially could be part of the same track) and a single track or a subset of tracks (that potentially are overlapping from the given point of view as in Fig. 3).

The estimated position of each track is refined in depth order, starting from the front-most track, maximizing a fitting function P_{FIT} with a gradient descent approach.

$$P_{FIT}(T_k, \vec{\delta}_{BF}) = Likelihood \cdot Confidence = \frac{\sum_{x \in MO} P_{APP}(I(x - \vec{\delta}_{BF}), AI_k(x)) \cdot PM_k(x)}{\sum_{x \in MO} PM_k(x)} \cdot \frac{\sum_{x \in MO} PM_k(x)}{\sum_{x \in T_k} PM_k(x)} \tag{3}$$

with

$$P_{APP}(RGB_i, RGB_j) = (2\pi\sigma^2)^{-3/2} e^{-\frac{\|RGB_i - RGB_j\|^2}{2\sigma^2}} \tag{4}$$

The fit measure is composed of two terms: the *Likelihood* is a measure of how similar the corresponding pixels of the

MO and the track are, after translation with a displacement vector $\vec{\delta}_{BF}$; the *Confidence* term is the percentage of track points, weighted with their probability, that are visible on the current frame and belonging to the *MO*.

Each pixel is assigned to the track that maximizes the conditional probability of finding the track in that position, given by the product of PM_k and P_{APP}. P_{APP} is measured as a distance between the Gaussian model *AI(x)* and the actual image *I(x)*. If the best-fit value goes too low, the track is considered *occluded*. The process is refined with the use of occlusion classification: a set of non-visible regions of the track is divided into apparent occlusions (shape changes), track-based occlusions (visual overlap, as in Fig. 3) and still object-based occlusions (furniture, corners). In Fig. 4 an example of object-based occlusion is shown, together with the Confidence measure: the tracks between frames 227 and 241 are considered occluded. Track-based occlusions are recognized by the presence of contended points between tracks. To recognize object-based occlusions, the edges of the non-visible regions are extracted along with those of the background. The edge pixels that touch the visible pixels of the track are classified as *bounding pixels*. If the majority of the bounding pixels match the background edges, we can assume that an object hides a part of the track, and the region is labelled as occluded by an object, otherwise as *changing*. The estimated position of the track, its appearance image and probability mask are not modified if the track is occluded by other tracks.

In the other cases, in order to have a more reactive system, the visible parts of the tracks are updated. An adaptive function is employed for the probability mask and the appearance image:

$$PM^t(x) = \begin{cases} \lambda PM^{t-1}(x) + (1-\lambda) & x \text{ is not occluded} \\ \lambda PM^{t-1}(x) & x \text{ is changing} \\ PM^{t-1}(x) & \text{otherwise} \end{cases}$$
$$AI^t(x) = \begin{cases} \lambda AI^{t-1}(x) + (1-\lambda) I^t(x) & x \text{ is not occluded} \\ AI^{t-1}(x) & \text{otherwise} \end{cases} \quad (5)$$

The depth order is evaluated by sorting the tracks along their probability of not being occluded. If a cyclic occlusion occurs, all the involved tracks are assigned to the same depth order. Finally, the motion vector is estimated according to constant speed assumption, but enforced by a segmented trajectory scheme: starting from an initial reference position, we collect a certain number of successive motion vectors and when we obtain a sufficient number of samples, they are linearly interpolated by finding the least squares solution. The solution vector is the estimation for the motion in future frames. This solution is checked to see if the interpolation correctly describes the last vector by verifying the ratio between the two eigenvalues of the principal direction computation and also if the angle or modulus has changed a lot from the first value. If the solution fails these checks, a new reference position is created and a new direction can be searched. In this way, an adaptive finite window is used to infer the future motion of the object.

Track merging is managed by means of the trajectory comparison: tracks that have merged often and kept moving in the same direction are merged. The split section exploits a vertical projection of the probability mask, combined with a separate trajectory analysis of the different components that are supposed to be separated. Only if the hypothesis is confirmed by the separate trajectory analysis, is the split operation performed.

2.3 People's Posture Classification

Once people's tracks are created, they are analyzed to detect posture. The posture classification described here is able to distinguish among four main postures: *Standing*, *Crouching*, *Sitting* and *Lying*. Each of them is further distinguished with left-, right- or front-headed. This classifier exploits the intrinsic characteristics of the silhouette to recall the person's posture and it is based on *projection histograms* $(\vartheta(x);\pi(y))$ that describe the way in which the silhouette's shape is projected on the *x* and *y* axes. An example of a projection histogram is depicted in Fig. 5. Though projection histograms are very simple features, they have proven to be sufficiently detailed to discriminate between the postures we are interested in. However, these descriptors have two limitations: 1) they depend on the silhouette's size (due to the perspective, the size of the detected VO can be different) and 2) they are too sensitive to the unavoidable non-rigid movements of the human body. To mitigate the point 1), we exploit camera calibration to compute the distance between the object and the camera and then we scale the silhouettes of the person proportionally with it.

For the second problem, we developed a suitable model capable of generalizing the peculiarities of a training set of postures. In particular, we compute a pair of Projection Probabilistic Maps (PPMs, $\Theta_i(x,y)$ and $\Pi_i(x,y)$) for each posture with a supervised machine-learning phase. The probabilistic approach included in the PPMs allows us to filter the useless moving parts of the body (such as the arms and the legs) that can generate misclassifications of the posture:

$$\Theta_i(x,y)=\frac{1}{T_i}\cdot\sum_{t=1}^{T_i} g\left(\theta_i^t(x),y\right);\Pi_i(x,y)=\frac{1}{T_i}\cdot\sum_{t=1}^{T_i} g\left(x,\pi_i^t(y)\right) \tag{6}$$

with

$$g(s,t)=\frac{1}{|s-t|+1} \tag{7}$$

At the classification stage, the projection histograms obtained by each blob are compared with the PPM of each class by computing a similarity measure.

This classifier is precise enough if the lower level segmentation module produces correct silhouettes. However, by exploiting knowledge embedded in the tracking phase, many possible classification errors due to the imprecision of the blob extraction can also be corrected. Fig. 6 shows some examples of the tracking module output for different postures.

We have defined a *state-transition graph*, where we use two states for each posture, one stable and the other unstable. The transitions between states are guided by two inputs: the posture detected frame by frame as previously explained

and the confidence value of the track. If the confidence decreases, the classification result is no longer reliable, but we keep the posture estimation unchanged by moving into the corresponding unstable state. Transitions between two stable states must pass through an unstable state, for at least one frame and require a high confidence value. Obviously, this introduces a short delay in posture change detection but solves spurious classification errors.

2.4 Universal Multimedia Access

As shown in Fig. 1, an unusual characteristic of our system is that of providing access to the camera content from everywhere and also by using a device with limited capabilities. This possibility is often called *UMA* (Universal Multimedia Access) and requires the adaptation of the video content, in order to meet the device constraints and the user's requests. This process is referred to as *content-based* or *semantic video adaptation*.

With these premises, the information provided by the previous modules is exploited to extract the semantics from the video. These semantics drive the content-based adaptation in accordance with the user's requests. In particular, the ontology of the current application is organized into *classes of relevance* [2] C_k defined as pairs $C_k =< o_i, e_j >$, where o_i represents an object class and e_j is an event class (selected among the set of objects and event classes that are detectable by the system). The user can associate a weight w_k with each class: the higher the weight, the more relevant the class must be considered, and less aggressive transcoding policies must be applied.

In in-house surveillance systems, the classes with higher relevance are those including people as objects (segmented and classified by the segmentation module, and tracked by the tracking one), and falls and dangerous situations as relevant events (extracted by the behaviour analysis derived by the posture classification module). An example of content-based video adaptation is provided in Fig. 7.

The adapted video is coded by the server as a stream, in order to provide live and timely content to the client device. To implement a coding driven by the semantics, we modified the MPEG-2 standard to change the compression factor of the single macro-blocks of each frame, in accordance with the relevance of the majority of its pixels. We have called this method *SAQ-MPEG*, i.e. Semantic Adaptive Quantization-MPEG [2].

Eventually, we developed a client decoder for PDA devices, which decodes the stream coming from the server and visualizes on-the-fly the final adapted video on the PDA screen.

3 Discussion and conclusions

In the introduction we listed some peculiarities of in-house video surveillance that must be taken into account to develop a reliable and efficient system. We can summarize the distinctive problems to be addressed in in-house video

surveillance as in Table 1.

The proposed solution for shadow detection proves to be reliable for detecting shadow points and discriminating them with reference to foreground points [3] and its benefits for posture classification are evident in Fig. 5. It can be noticed that the cast shadows underneath and on the left of the person heavily modify the shape and, consequently, the projection histograms. The probabilistic tracking described in Section 2.2 solves many problems due to both occlusions and non-rigid body models. Fig. 3 shows an example of track-based occlusion in which the tracks associated with the two people (indicated by capital letters) are never lost, even if there is total occlusion. Fig. 4 reports a case of object-based occlusion and shows how the confidence decreases when the person goes behind the object and the model is not updated. How the non-rigid body model is treated with the appearance model and probability mask for different postures is shown in Fig. 6. The probabilistic tracking, together with camera calibration and consequent model scaling, provides a scale-invariant, occlusion-free, and well-segmented shape of the person, as input to the posture classification module.

Only by taking into account all these aspects is there very high accuracy performance of the system (in terms of correctly classified postures). We tested the system over more than 2 hours of video (provided with ground-truths), achieving an average accuracy of 97.23 %. The misclassified postures were mainly due to confusion between sitting and crouching postures.

References

[1] J. K. Aggarwal, Q. Cai. “Human Motion Analysis: A Review.” In *Computer Vision and Image Understanding* 73(**3**): pp 428-440, (1999).

[2] M. Bertini, R. Cucchiara, A. Del Bimbo, A. Prati, "An Integrated Framework for Semantic Annotation and Transcoding" *in press on Multimedia Tools and Applications* - Kluwer Academic Publishers, (2003).

[3] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, "Detecting Moving Objects, Ghosts and Shadows in Video Streams" in *IEEE Transactions on PAMI*, **25**, n. 10, pp. 1337-1342, (2003).

[4] I. Haritaoglu, D. Harwood, L.S. Davis: “Ghost: A Human Body Part Labeling System Using Silhouettes”, *In Proceedings of Fourteenth International Conference on Pattern Recognition*, Brisbane, (1998).

[5] I. Haritaoglu, D. Harwood, and L.S. Davis, “W4: real-time surveillance of people and their activities,” *IEEE Transactions on PAMI*, **22**, n. 8, pp. 809–830, (2000).

[6] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H.Wechsler, “Tracking groups of people,” *Computer Vision and Image Understanding*, **80**, no. 1, pp. 42–56, (2000).

[7] T.B. Moeslund, E. Granum: “A Survey of Computer Vision-Based Human Motion Capture”, *Computer Vision and Image Understanding*, **81**, Elsevier Science Pubs., North Holland, pp 231-268, (2001).

[8] A. Prati, I. Mikic, M.M. Trivedi, R. Cucchiara, “Detecting Moving Shadows: Algorithms and Evaluation” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**, n. 7, pp. 918-923, (2003).

[9] A. Senior, A. Hampapur, Y.L. Tian, L. Brown, S. Pankanti, R. Bolle, “Tracking people with probabilistic appearance models”, *Proc. of International Workshop on Performance Evaluation of Tracking and Surveillance Systems*, (2002).

[10] J. Stauder, R. Mech, and J. Ostermann, “Detection of moving cast shadows for object segmentation,” *IEEE Transactions on Multimedia*, **1**, no. 1, pp. 65–76, (1999).

[11] C. Stauffer and W. Grimson. “Adaptive background mixture models for real-time tracking”. *In Int. Conf. Computer Vision and Pattern Recognition*, **2**, (1999).

[12] 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).

[13] A. Vetro, T. Haga, K. Sumi, and S. H. “Object-based coding for long-term archive of surveillance video”, *in IEEE Conference on Multimedia & Expo*, **2**, pp. 417–420, (2003).

PROBLEM	CAUSE	EFFECT	SOLUTION
large shadows	diffuse and close sources of illumination	shape distortion and object merging	shadow detection in the HSV colour space (**Sec. 2.1**)
deformable object model	non-rigid human body	shape-based algorithms are misled	probabilistic tracking based on appearance models (**Sec. 2.2**)
track-based occlusions	position of the camera; interaction between humans	tracking problems; shape-based algorithms are misled	probabilistic tracking based on appearance models (**Sec 2.2**)
object-based occlusion	presence of many objects		
scale-dependent shapes	position and distance of the camera	size of people depends on their distance from the camera	camera calibration and body model scaling (**Sec. 2.3**)
object displacement	non static scene	reference background changes	statistical and knowledge-based background model (**Sec. 2.1**)
accessibility	need for every time/everywhere access and large bandwidth	inaccessibility for devices of limited capability	content-based video adaptation (**Sec. 2.4**)

Table 1. Summary of in-house surveillance problems, with their cause and effect, and the solutions we propose.

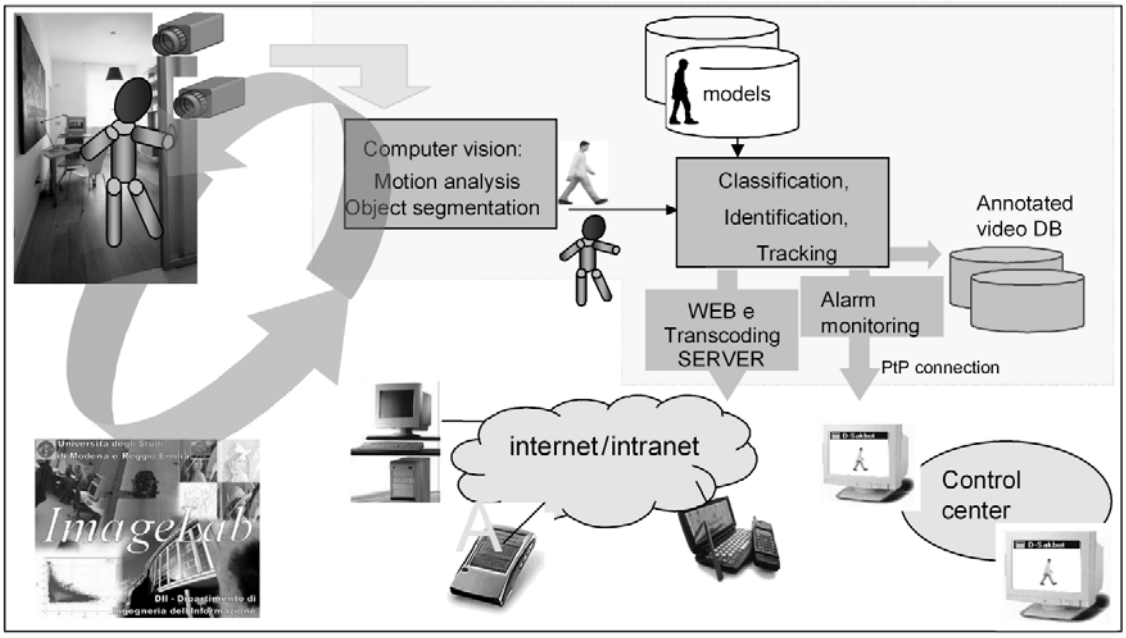
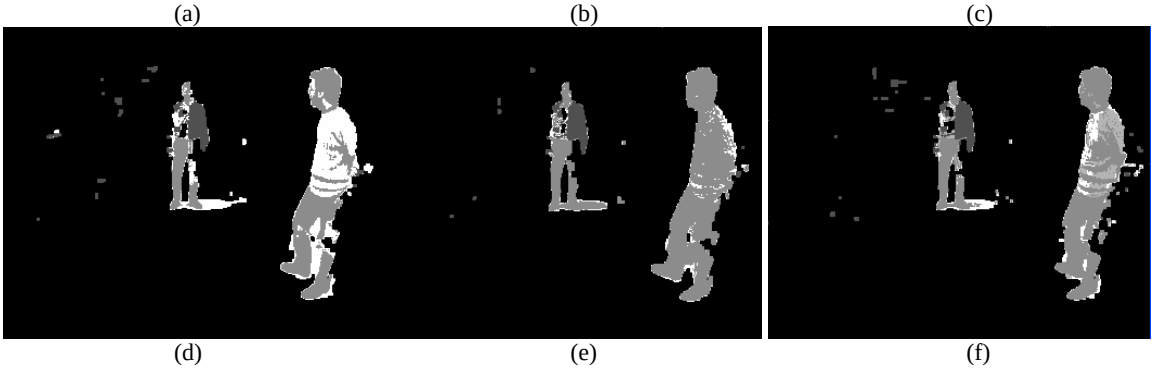


Fig. 1. Overall scheme of the system

(a) (b) (c)



(d) (e) (f)

Fig. 2. Examples of shadow detection under different conditions.

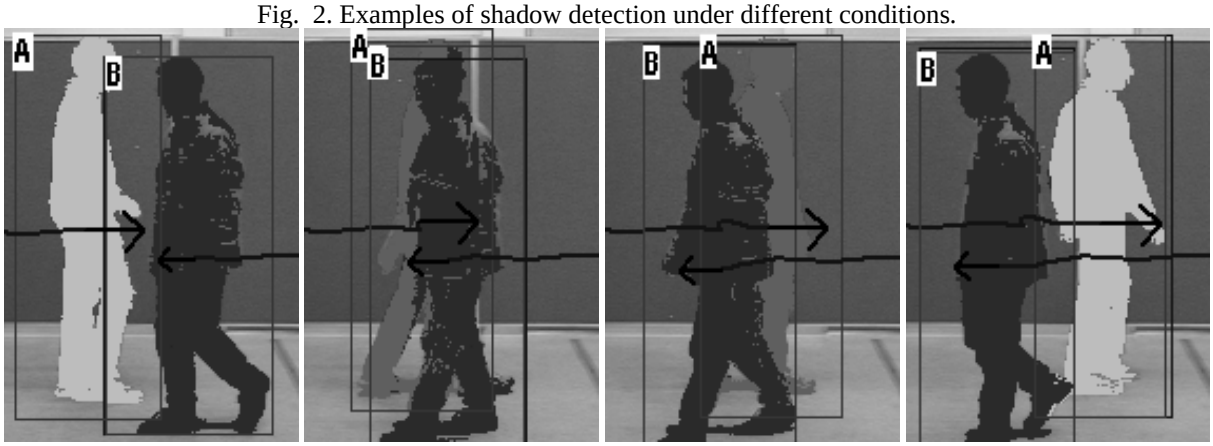
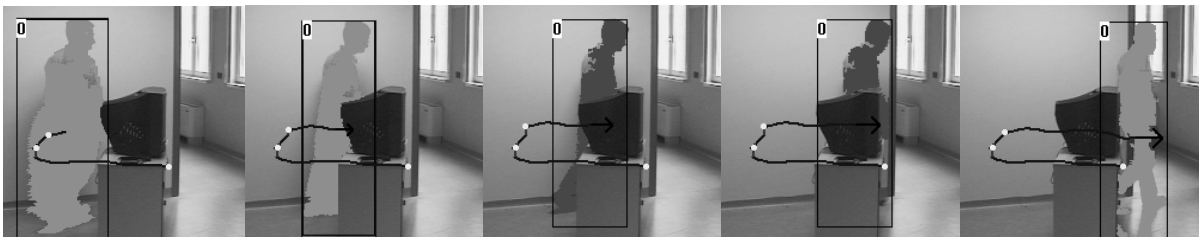


Fig. 3: Correct track-based occlusion solving.



211 **222** **227** **234** **244**

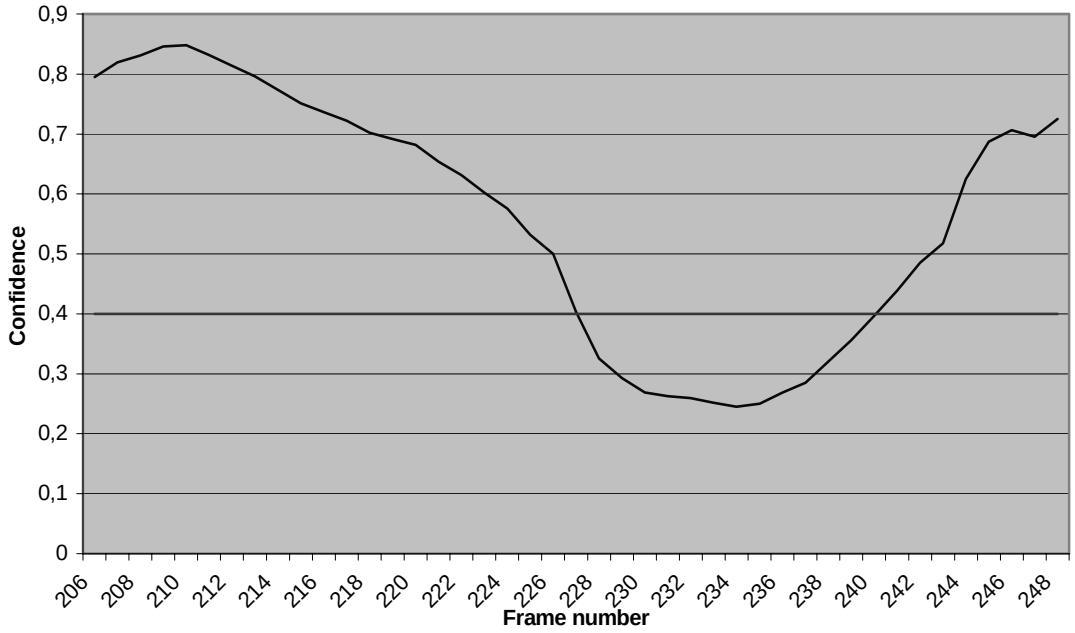


Fig. 4. Example of object-based occlusion and the corresponding confidence value.

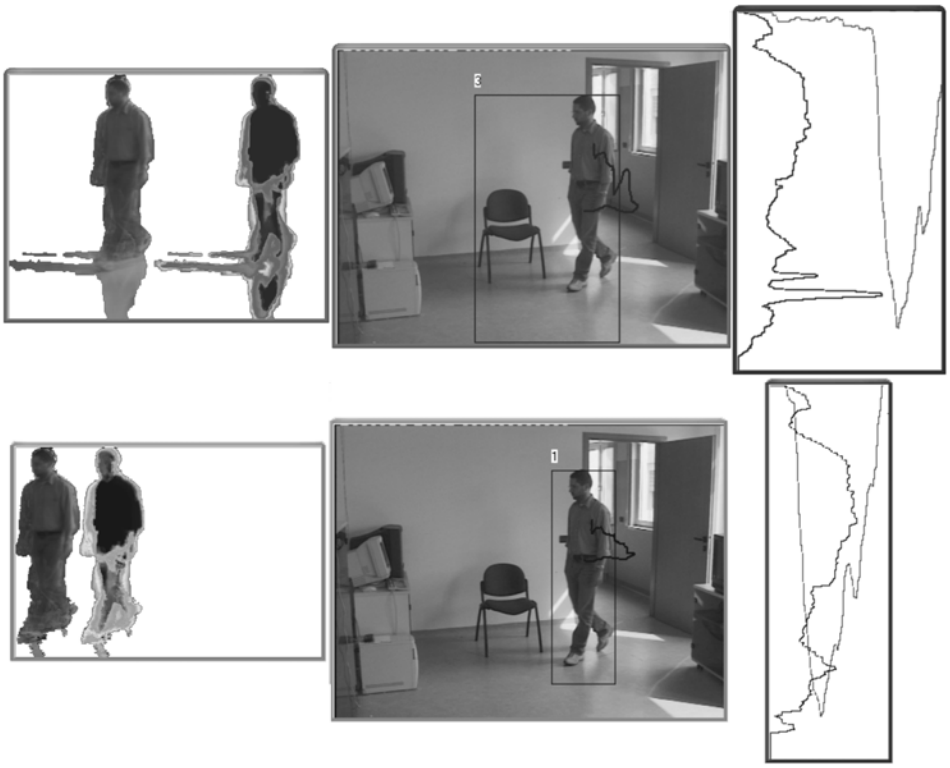


Fig. 5: Tracks, VO extent and projection histograms computed without (top) or with shadow suppression (bottom)

Fig. 6: Appearance images (left) and probability masks (right) of a person's track in different postures

Fig. 7: Example of content-based video adaptation.