ALMA MATER STUDIORUM
UNIVERSITÁ DI BOLOGNA

ARCES – ADVANCED RESEARCH CENTER ON ELECTRONIC SYSTEMS
FOR INFORMATION AND COMMUNICATION TECHNOLOGIES E. DE CASTRO

DETECTING CHANGES IN VIDEO SEQUENCES

Alessandro Lanza

TUTORS

Professor
Luigi Di Stefano

COORDINATOR

Professor
Riccardo Rovatti

PHD. THESIS
January, 2004 – December, 2006

PHD PROGRAM IN INFORMATION TECHNOLOGY
CYCLE XIX – ING-INF/05

Contents

Chapter 1

Introduction

1.1 The context

Computer vision can be defined as the deduction of information about the world by automatic analysis of images taken from either a single or multiple viewpoints. Computer vision is a fascinating and challenging research field, with many established (e.g., automated visual inspection, robot guidance, optical character recognition, medical imaging, remote sensing) as well as emerging (e.g., video surveillance, traffic monitoring, human-computer communication) application domains.

In the last decade, a wide range of research areas concerned with real-time applications have received a growth in attention , due to a considerable performance boost of off-the-shelf computing platforms. Among these, it is worth pointing out those paving the way for emerging applications in unconstrained environments wherein, unlike established industrial application, a complex and changing world must be accurately and reliably modeled.

One of these research fields is undoubtedly change detection. Change detection deals with the automatic detection of changes occurring in a scene by the elaboration of single or multiple video sequences of the scene captured from single or multiple view-points by fixed or moving imaging devices. Change detection is the first crucial processing step in many Computer vision applications, such as video-surveillance, traffic monitoring and remote sensing. In fact, upon a reliable preliminary change detection step higher level capabilities can be built, such as those concerned with objects tracking, classification and behavior analysis.

1.2 The problem

The input of a typical change detection algorithm at time $t = \bar{t}$ is a set $\mathcal{I}_{\bar{t}}$ of synchronized video sequences $\boldsymbol{S}^v_{\bar{t}}$ of the same scene captured by different imaging devices from different view-points:

$$\mathcal{I}_{\bar{t}} = \left(\boldsymbol{S}^1_{\bar{t}}, \boldsymbol{S}^2_{\bar{t}}, \ldots, \boldsymbol{S}^V_{\bar{t}} \right) \tag{1.1}$$

Depending on the number V of different view-points (i.e. of input video sequences), change detection is denoted as *multi-view* ($V > 1$) or *single-view* ($V = 1$). Each input video sequence $\boldsymbol{S}^v_{\bar{t}}$ consists of a finite number of digital (discrete domain, discrete range) images $\boldsymbol{I}^v_t$ of the scene captured by the same imaging device at different discrete times $t \leq \bar{t}$:

$$\boldsymbol{S}^v_{\bar{t}} = \left(\boldsymbol{I}^v_{\bar{t}-T}, \ldots, \boldsymbol{I}^v_{\bar{t}-1}, \boldsymbol{I}^v_{\bar{t}} \right)^\top \tag{1.2}$$

It is worth pointing out that in case of multi-view change detection the input video sequences are assumed to be synchronized, so that they contain the same number $(T+1)$ of images, captured at common times $(\bar{t}, \bar{t}-1, \ldots, \bar{t}-T)$. As a consequence, the input information $\mathcal{S}_{\bar{t}}$ of a typical change detection algorithm can be written as a matrix of $(T+1) \cdot V$ digital images, where $(T+1)$ is the number of images in each video sequence and V is the number of different views-sequences:

$$\mathcal{I}_{\bar{t}} = \begin{pmatrix} \boldsymbol{I}^1_{\bar{t}-T} & \boldsymbol{I}^2_{\bar{t}-T} & \ldots & \boldsymbol{I}^V_{\bar{t}-T} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{I}^1_{\bar{t}-1} & \boldsymbol{I}^2_{\bar{t}-1} & \ldots & \boldsymbol{I}^V_{\bar{t}-1} \\ \boldsymbol{I}^1_{\bar{t}} & \boldsymbol{I}^2_{\bar{t}} & \ldots & \boldsymbol{I}^V_{\bar{t}} \end{pmatrix} \tag{1.3}$$

It is clear that each column of $\mathcal{I}_{\bar{t}}$ represents a different input video sequence $\boldsymbol{S}^v_{\bar{t}}$, but it is worth noticing that each row (denoted as $\boldsymbol{R}_t$) contains a set of simultaneous images of the monitored scene taken from different view-points. Each input digital image $\boldsymbol{I}^v_t$ can be regarded as a function mapping a pixel coordinates l-dimensional integer vector $\boldsymbol{p}$ (hereinafter, pixel) to a pixel intensities m-dimensional integer vector $\boldsymbol{c} = \boldsymbol{I}^v_t(\boldsymbol{p})$ (pixel color):

$$\boldsymbol{I}^v_t : \quad \mathbb{Z}^l \ni \boldsymbol{p} \mapsto \boldsymbol{c} = \boldsymbol{I}^v_t(\boldsymbol{p}) \in \mathbb{Z}^m \tag{1.4}$$

Typically, $m = 1$ (e.g., grey level images) or $m = 3$ (e.g., RGB color images), but other values are possible. For instance, multi-spectral images have values of m in the tens, while hyper-spectral images have values in the hundreds. Typically, $l = 2$ (e.g., satellite or surveillance images) or $l = 3$ (e.g., volumetric medical or biological microscopy data). In the rest of the thesis, we will take into consideration just change detection for planar grey level images ($l = 2$, $m = 1$), mapping a 2-dimensional pixel $\boldsymbol{p} = (i, j)$ to a scalar grey level $g = I^v_t(i, j)$:

$$I^v_t : \quad \mathbb{Z}^2 \ni \boldsymbol{p} = (i, j) \mapsto g = I^v_t(i, j) \in \mathbb{Z} \tag{1.5}$$

Figure 1.1(a) shows the input of a typical multi-view change detection algorithm.

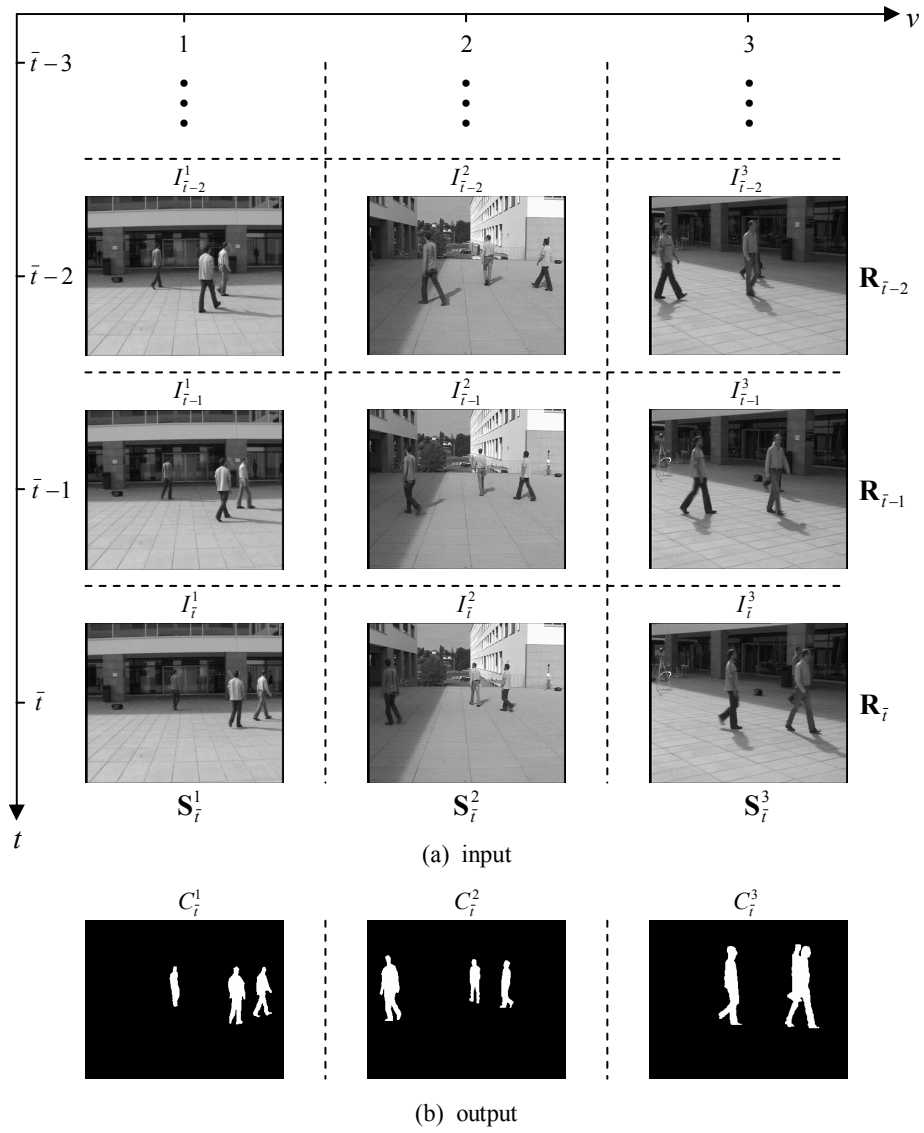


Figure 1.1: Input (a) and output (b) of a typical multi-view change detection algorithm ($V = 3$).

The output of a typical multi-view change detection algorithm at time $t = \bar{t}$ is a set $O_{\bar{t}}$ of V images $\boldsymbol{C}^v_{\bar{t}}$, one for each view-point, called *change masks*:

$$O_{\bar{t}} = \left(\boldsymbol{C}^1_{\bar{t}}, \boldsymbol{C}^2_{\bar{t}}, \ldots, \boldsymbol{C}^V_{\bar{t}} \right) \tag{1.6}$$

so that a multi-view change detection algorithm can be regarded as a function CD_{MV} mapping, at each time t, the input $\mathcal{I}_t$ to the output O_t:

$$O_t = CD_{MV}(\mathcal{I}_t) \tag{1.7}$$

Each change mask $\boldsymbol{C}^v_{\tilde{t}}$ is a binary image having the same domain as the images contained in the corresponding (i.e. relative to the same view-point) input sequence $\boldsymbol{S}^v_{\tilde{t}}$ and defined as follows:

$$C^v_{\tilde{t}}(\boldsymbol{p}) = CD^v_{MV}(\mathcal{I}_{\tilde{t}}) = \begin{cases} 1 & \text{if a significant change occurred at pixel } \boldsymbol{p} \text{ of } I^v_{\tilde{t}} \\ 0 & \text{otherwise} \end{cases} \tag{1.8}$$

The output of a typical multi-view change detection algorithm is shown in figure 1.1(b). To make the change mask definition in equation 1.8 clear, it is necessary to give an answer to the following two questions:

a) What does "significant" change mean?

b) What should a change be detected with respect to?

1.2.1 What does significant change mean?

When a scene is imaged by a capturing device, it can be regarded as 3-dimensional (not necessarily planar) surface, that is the portion of the physical surface of the objects in the scene that is visible through the perspective projection characterizing the imaging device, immersed in a 3-dimensional Euclidean space, that is the physical space. A digital image of the scene is a geometrical appearance model of the scene. On one hand, it is a geometrical model since it is a perspective projection of the 3-dimensional scene surface to the 2-dimensional image plane of the capturing device. On the other hand, it is an appearance model since it is a measure of the radiance (i.e. the electromagnetic radiation in the visible spectrum), emitted by the scene surface. Hence, the intensity of a pixel in a scene image is a measure of the radiance emitted by the patch of scene surface connected to the pixel itself by the central projection (through the optical center) of the capturing device (figure 1.2).

We call a scene (or semantic) information an information on the 3-dimensional geometry of the scene surface. On the other hand, we call an image (an appearance) information an information contained in the scene image (i.e. the measured pixel intensities). In general, given an image information it is not straightforward to infer scene information about the connected patch of scene surface. In fact, any given 3-dimensional position in the imaged scene is univocally connected with a pixel in the scene image. On the contrary, given a pixel in the scene image all the 3-dimensional positions in the imaged scene lying on the line (optical ray) passing through the pixel and the optical center are possible (figure 1.2). Shape from shading, photometric stereo and multi-view stereo are disciplines in the computer vision field aiming at inferring scene information from image information. In particular, shape from shading aims at computing the 3-dimensional shape of a scene surface from a single image of the

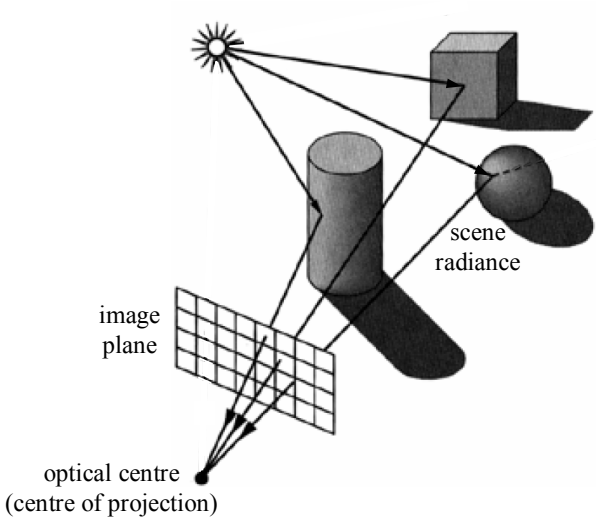


Figure 1.2: Image formation process: a scene image is a geometric appearance model of the imaged scene.

scene. The shape is inferred by assuming a reflectance model (i.e. a model of the light reflection physical phenomenon) for the physical scene surface and then computing the 3-dimensional shape which maximizes the likelihood of the scene image given the assumed reflectance model. It is well-known that the shape from shading problem is in general (i.e. for a generic shape of the scene surface) ill-posed, even in the case of simple reflectance models (e.g. Lambertian reflectance model). Also in the simple cases where a solution exists (thanks to the assumption of scene surface smoothness constraints), more than just local image information has to be processed. Instead of using a single image, photometric stereo tries to solve the recovering problem by processing the information contained in two or more different scene images. The images are taken by the same imaging device and from the same view-point, but the light source position in the scene is different. The problem is mathematically well-posed, that is the orientation of the normal to the scene surface can be, in theory, determined for each surface patch connected to a scene image pixel. However, it is clear how photometric stereo can not be used in unconstrained environments, where the light source (or, better, sources) position can not be controlled. Multi-view stereo recovers information on the scene surface geometry by using two scene images taken from different view-points and, in general, by different imaging devices. By detecting couples of points (i.e. pixels) in the two images corresponding to the same scene surface patch (matching procedure) and by exploiting geometrical properties related to the central projection characterizing the capturing device (disparity computation), the *depth* of the pixels is computed, that is the distance between the scene surface patch imaged by the two pixels and the

imaging devices principal plane. Multi-view stereo provides good results also in case of unconstrained environments, however:

a) the imaging devices must be carefully calibrated and temporally synchronized;

b) the matching procedure is the most important and also the most critical part of a multi-view stereo algorithm. In fact, not all the couples of points which actually correspond to the same scene surface patch can be detected.

All this discussion was aimed at pointing out how the inference of scene information from image information is a complex and hard-to-solve problem. By aiming "lower", that is by passing from the absolute continuous formulation of the problem (i.e. to compute the scene surface 3-dimensional geometry) to the differential dichotomic one (i.e. to detect if a change of the 3-dimensional scene surface geometry has occurred), a more tackleble problem is attained. This is the change detection problem. Finally, we can answer the question giving the title to this paragraph by saying that change detection aims at detecting scene (or semantic) changes, that is changes of the scene surface geometry. Hence, "significant" in equation 1.8 can be replaced by "scene" so that the output of a change detection algorithm is a change mask defined as follows:

$$C_{\bar{t}}^{v}(\boldsymbol{p}) = CD_{MV}^{v}(\mathcal{I}_{\bar{t}}) = \begin{cases} 1 & \text{if a scene change occurred at pixel } \boldsymbol{p} \text{ of } I_{\bar{t}}^{v} \\ 0 & \text{otherwise} \end{cases} \tag{1.9}$$

It is worth spending right now some words about the two main problems arising in change detection, that is *camouflage* and *disturbance factors*. Both the problems can be the cause of detection errors, but in general of opposite "sign". In fact, camouflage always gives rise to missed detections (i.e. false negatives) while disturbance factors almost always yields false detections (false positives). As regards camouflage, it is due to the fact that a change of the scene surface 3-dimensional geometry does not necessarily cause a change of the emitted radiance and, as a consequence, of the measured pixel intensities. As an example, we can think of a moving object having a very similar color (i.e. radiance) to that of the covered scene surface. As one can easily understand, this problem is inherently not solvable on a local basis, that is by just considering the measured intensities in a small neighborhood of pixels. In fact, images are nothing else than a measure of the radiance emitted from the scene surface. If the radiance does not change, nothing can be said about possible scene changes. The problem can be afforded just at a higher level, that is at the objects-level. In practice, more or less explicit assumptions about the foreground objects shape have to be made. These assumptions can be directly included within the change mask computation procedure or implemented as a change mask post-processing (e.g. mathematical or statistical morphology). Apart form chapter 4, where we present an ad-hoc filling algorithm aimed

at removing possible missed detections (which, however, are not due to camouflage), in this thesis we focus our attention on low-level change detection algorithms without any morphological assumption. Hence, we renounce in advance to detect "almost perfectly" camouflaged scene changes. In this framework, we can say that all the scene changes we aim to detect by our algorithms give rise to a measurable image change, that is:

$$\textit{scene change} \quad \Longrightarrow \quad \textit{image change} \tag{1.10}$$

The opposite problem to the one just discussed, that is the detection of false scene changes, can arise in change detection as well. In fact, not all the measurable image changes are in general due to scene changes. We call disturbance factors all the possible causes of measurable image changes not related to changes of the scene surface geometry. The most important disturbance factors are the following:

a) scene illumination changes: changes of the amount of light emitted by the sources present in the scene.

b) imaging system noise: statistical error affecting the measured pixel intensities due to a variety of phenomena occurring along the imaging process.

c) dynamic adjustments of the imaging system parameters: changes of the parameters which characterize the transfer function mapping the sensed scene radiances to the measured pixel intensities.

The first disturbance factor is the only one actually affecting the radiance emitted from the scene surface. The other two, in fact, arise in the scene radiance measurement step inside the imaging device. However, the effect of all disturbance factors is an image change, so that we can write:

$$\textit{disturbance factors} \quad \Longrightarrow \quad \textit{image change} \tag{1.11}$$

By remembering that a change detection algorithm aims at detecting scene changes (i.e. the cause) from images (i.e. the effect) and by looking at rules 1.10 and 1.11, change detection can be regarded as the logical abduction problem shown in figure 1.3 : In practice, if no image change is measured (i.e. pixel intensities are unchanged) no scene change is detected. On the contrary, if an image change is measured the abduction of the right (i.e. the true) cause has to be carried out between scene changes and disturbance factors. Finally, a good change detector should be able to discriminate between (i.e. to classify) the effects of scene changes and of disturbance factors on the measured image intensities.

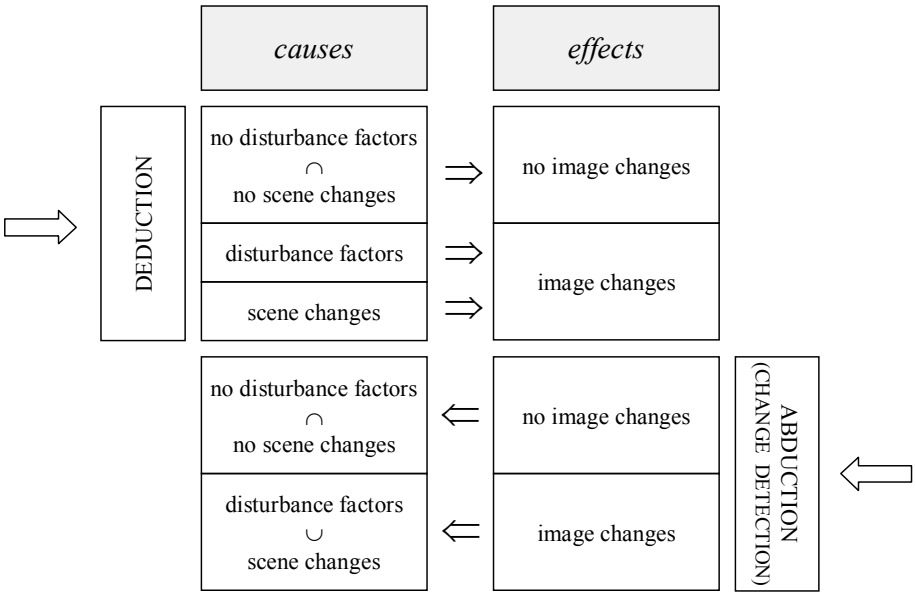


Figure 1.3: Change detection as a logical abduction problem.

1.2.2 What should a change be detected with respect to?

The discussion carried out in the previous paragraph is valid both for multi-view and for single-view change detection. On the contrary, a distinction has to be made to say what a change should be detected with respect to. In fact, since a multi-view change detector elaborates scene images taken at different times and from different view-points, changes can be detected in the temporal (given a view-point, along frames captured at different times) as well in the spatial (given a capturing time, along frames taken from different view-points) domain. Moreover, a variety of hybrid solutions are possible as well. In this thesis, we propose a hybrid solution in which the detection of changes in the temporal domain represents the main part. In practice, single-view change detection is carried out independently for each view-point. Just as a final processing step, the information contained in all the attained single-view change masks is fused by the multi-view spatial constraint. The aim of this final processing step is to filter-out a particular type of false changes (i.e. very local false changes, such as those due to shadows cast by foreground objects), which can be hardly dealt with by a single-view approach. Indeed, most of this thesis is devoted to single-view change detection. The multi-view change masks fusion approach is presented in the last chapter. For this reason, we postpone to that point the discussion of how time, space or a combination of both can be the basis on which changes are detected in a multi-view change detection approach.

As regards single-view change detection, changes are necessarily detected on a

temporal basis. However, as far as the answer given to the question put in this paragraph is concerned, two main classes of single-view change detection algorithms can be identified:

a) algorithms based on temporal frame-difference: at time $t = \bar{t}$, changes occurring in a pixel $\boldsymbol{p}$ of the current frame $I_{\bar{t}}$ (to simplify notations, hereinafter we drop the superscript v when dealing with single-view change detection) are detected with respect to one (two-frame difference) or two (three-frame difference) previous frames. The change mask computation rule of equation 1.9 can be expressed as follows:

$$C_{\bar{t}}(\boldsymbol{p}) = CD_{SV}(\mathcal{I}_{\bar{t}} = \boldsymbol{S}_{\bar{t}}) = \begin{cases} 1 & \text{if} \quad d\left(I_{\bar{t}}(\boldsymbol{p}), I_{\bar{t}-1}(\boldsymbol{p}), \ldots\right) > T \\ 0 & \text{otherwise} \end{cases} \tag{1.12}$$

where CD_{SV} denotes the overall single-view change detection algorithm, mapping the input information $\mathcal{I}_{\bar{t}}$, corresponding to the single-view input sequence $\boldsymbol{S}_{\bar{t}}$, to the single-view output change mask $C_{\bar{t}}$. d is a function giving a measure of dissimilarity between the current and the previous frame intensities.

b) algorithms based on background subtraction: changes in the current frame are detected with respect to the "stationary" part of the scene surface, commonly called scene *background*. An almost philosophical discussion may be carried out about the meaning of background, that is of the adjective "stationary". In the most common acceptation, background is the portion of imaged scene surface having a constant 3-dimensional geometry since a sufficiently long time. Hence, background subtraction consists in the comparison between the current frame $I_{\bar{t}}$ and an image (or, more frequently, an appearance model) of the scene background $\hat{B}_{\bar{t}}$.

$$C_{\bar{t}}(\boldsymbol{p}) = CD_{SV}(\mathcal{I}_{\bar{t}} = \boldsymbol{S}_{\bar{t}}) = \begin{cases} 1 & \text{if} \quad d\left(I_{\bar{t}}(\boldsymbol{p}), \hat{B}_{\bar{t}}(\boldsymbol{p})\right) > T \\ 0 & \text{otherwise} \end{cases} \tag{1.13}$$

In general, algorithms based on temporal frame difference are computationally very efficient since, at each capturing time, they just have to compute the dissimilarity between the current and the previous frame intensities. Moreover, disturbance factors are a problem just when they cause very fast image changes (e.g. a light turned on/off in the imaged scene) and just during the change transitory (i.e. after the light has been turned on/off, no more false changes are detected). In fact, capturing frame rate of common imaging systems is in the order of tens, so that the inter-frame acquisition time is in the order of hundredths of seconds. Even a light switch produces an image change which is spread over some frames. Slower changes, such as those related to the time of the day, does not cause inter-frame image changes which can be confused with the ones

produced by scene surface changes. Besides, the change mask computation process by temporal frame difference is a ”process without memory”, in the sense that the binary decision for a pixel to be changed or unchanged is taken without considering the past decisions. By remembering that each decision for the current frame is taken just on the basis of a comparison with the previous frame, it is easy to understand how, even for sudden image changes due to disturbance factors, a problem can arise just during the change transitory. However, temporal frame difference suffers from two inherent problems, called *ghosting* and *foreground aperture*, respectively. Figure 1.4, on the left, shows the effects of these two problems for a couple of successive frames of a sample sequence. The ghosting problem consists of false detections occurring in the pixels not

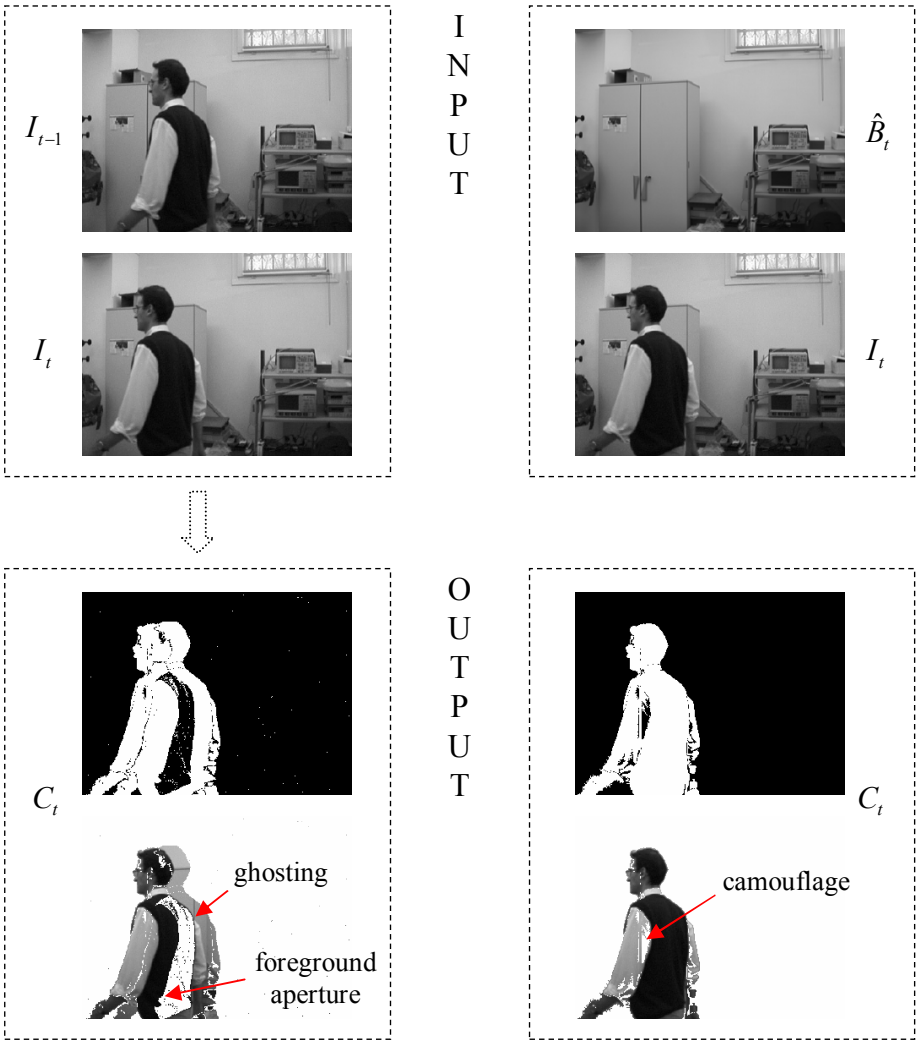


Figure 1.4: An example of change masks computed by temporal frame difference (on the left) and background subtraction (on the right) change detection algorithms.

covered by foreground objects in the current frame but covered in the previous one. Foreground aperture can be regarded as a particular type of camouflage, occurring between different surface patches of the same foreground object. While ghosting can be dealt with quite effectively (e.g. by using more than just two successive frames), the foreground aperture problem is actually inherent to the temporal difference method. To understand this, it is sufficient to think of a perfectly evenly "colored" foreground object moving in the scene or to whatever foreground object staying still in the scene. In these cases, temporal frame difference is trying to detect scene changes by comparing image intensities which actually correspond to the same emitted radiance. The result is a set of unavoidable missed detections.

As regards background subtraction, it suffers neither from ghosting nor from the foreground aperture problem. In fact, given that the background model is a "perfect" appearance model of the stationary part of the imaged scene (i.e. an exact measure of the radiance emitted by the scene background surface), the detection rule of equation 1.13 allows to detect exactly all the pixels which are sensing an incoming radiance different from the radiance emitted by the scene background surface. In other words, apart from possible missed detections due to camouflage, a perfect change mask should be attained. Also if we admit the presence of imaging system noise, an accurate change mask can be computed by choosing a proper value for the threshold T in the detection rule. In figure 1.4, on the right, we show the change mask computed by the detection rule of equation 1.13 for the same sample frame considered in the temporal frame difference case. In particular, the background model is simply an image of the scene captured before the person enters the scene and a value $T = 9$ has been chosen. The change mask is clearly much better than the one attained by the temporal frame difference approach. Some missed detections are present, due to a partial camouflage effect occurring in some pixels. Due to the absence of inherent problems, background subtraction is the most studied and the most applied change detection approach. Thought not inherent, the big problem arising in background subtraction is the maintenance of the background model. In fact, assumed that a good initial model has been generated, disturbance factors can subsequently change the radiance emitted by the scene background surface (illumination changes) or, however, the measured image intensities (adjustments of the imaging system parameters). Two main classes of background subtraction algorithms can be identified, based on a different way of facing the disturbance factors problem:

a) algorithms based on a temporally adaptive statistical model of the scene background appearance: the background model is a statistical model of the appearance (i.e. the sensed radiance) of the scene background surface. The detection rule is just a comparison between the appearance of the scene in the current frame

and the appearance of the scene background expressed by the background model. In particular, for each pixel the probability of observing (i.e. the likelihood of) the currently measured intensity given that the pixel is imaging the background surface is computed and then thresholded. Clearly, for the algorithm to be robust to the effects of disturbance factors the background model must be updated. Actually, the background model updating procedure is the most important and, at the same time, the most critical part of the background subtraction algorithms belonging to this class. In fact, if it is true that the imaging system noise effects are dealt with effectively by the statistical nature of the background model, illumination changes and dynamic adjustments of the imaging system parameters can be faced just by updating the background model.

b) disturbance factors invariant algorithms: the background model is usually a much simpler statistical model of the scene background appearance. Moreover, the model does not need to be updated. All the efforts to filter-out the effects of disturbance factors are concentrated in the background comparison step, that is in the dissimilarity computation procedure. To this purpose, an accurate modeling of the disturbance factors effects on the image intensities must be carried out. Besides, differently from the algorithms of class a), the decision for a pixel to be changed or unchanged can not be taken on a totally local basis, that is by just comparing the pixel currently measured intensity with the pixel intensity in the background model. In fact, for the disturbance factors effects to be distinguishable from actual scene changes effects at least a small patch of neighboring pixels has to be considered.

In this thesis, both the classes a) and b) of background subtraction algorithms will be dealt with. In particular, two very different algorithms we devised will be presented, each one belonging to a different class. For simplicity, hereinafter we call *temporally adaptive* and *disturbance factors invariant* the change detection algorithms of class a) and b), respectively.

We conclude this discussion on the change detection problem by pointing out two requirements that every *good* change detector should fulfill:

r.1) detection accuracy, that is the ability to compute accurate change masks;

r.2) computational efficiency, that is the ability to process a high number of frames per second.

Every change detection algorithm is a trade-off between *r.1* and *r.2*. Obviously, the goal of a researcher in this field should be to achieve the best trade-off, given a target detection accuracy.

1.3 The Solution and the Structure of the Thesis

The research work carried out during my PhD was almost entirely focused on the change detection problem. Actually, not just ”pure” but also applied research was conducted. In other words, not all the PhD was dedicated to the aim of devising new algorithms. In fact, thanks to the opportunity of the research results to be applied and commercialized within a spin-off company of which I am a current partner, part of the efforts were spent for the accurate implementation of the devised algorithms.

The first two years of the PhD were devoted to the single-view change detection problem. A deep investigation of the existing literature allowed me to get a clear idea of the state of the art in the field. In particular, the two classes of change detection algorithms mentioned in the previous section arose as the most studied as well as the ones providing the best results.

In the very first part of the PhD, an algorithm belonging to the first class was devised. The algorithm is presented in chapter 2. It is a background subtraction algorithm based on a statistical, temporally adaptive, non-parametric model of the scene background appearance. In particular, the statistical background model consists of a temporally adaptive couple of percentiles (i.e. a lower and an upper percentile) of the background process ensemble pdf. At each processing step, for each pixel the change mask is computed by checking whether the currently measured intensity falls inside the interval between the two percentiles. The novelty of the algorithm consists mainly in the procedure which provides the two percentiles. A statistical non-parametric model of the imaging system noise is inferred once and for all by an initial training sequence of frames. At each subsequent processing step, the model allows to have reliable percentiles at disposal for the change mask computation.

An algorithm belonging to the second class was devised as well. It is presented in chapter 3. A very simple background model is generated by processing a training sequence of frames. Differently from the algorithm presented in chapter 2, the background need not to be updated. Detection of changes in each pixel is computed by comparing the intensities (in the current frame and in the background model) not just of the pixel itself but also of a small neighborhood of pixels. In particular, based on the assumption that disturbance factors produce image changes identifiable with local monotonic non decreasing intensity mapping functions, a maximum likelihood isotonic regression procedure is used to discriminate between the disturbance factors and the scene changes effects.

Chapter 4 presents an hybrid or, better, a comprehensive solution. In particular, a coarse-to-fine approach to the single-view change detection problem is proposed. The basic idea consists in assigning to a preliminary coarse-level detection the task to filter-out most of the possible effects of disturbance factors. In particular, the disturbance

factors invariant algorithm presented in chapter 3 is used. As a consequence, reliable coarse-grain change masks are attained, which are a superset of the semantically changed pixels. The coarse-grain masks can be used as a work-area by the subsequent fine-level detection algorithm.

In chapter 5, a multi-view change detection approach is presented. It relies on single-view change detection, in the sense that the multi-view constraint is applied just as a final processing step. In practice, single-view change detection is carried out independently in each view. The attained change masks are then fused to filter-out very local false detections, such as those due to specularities and to shadows.

Chapter 2

Temporally Adaptive Change Detection

In this Chapter we present a change detection algorithm for grey level sequences aimed at achieving a good trade-off between time performance and detection quality. The algorithm relies on background subtraction and on the extraction of a statistical model of the imaging system noise. In particular, in Section 2.1 the noise model extraction algorithm is presented. Section 2.2 outlines the procedure used to initialize the background model and Section 2.3 describes the background subtraction and updating algorithms.

2.1 Imaging System Noise Modeling

Apart from change detection, many other computer vision algorithms (e.g. shape from shading, photometric stereo) require precise measures of scene radiance. The more accurately the measured image brightness represents the scene radiance, the higher the performance of the algorithms is. Unfortunately, real imaging devices deviate from an ideal behavior, mainly for two reasons. Firstly, the camera response function (the function which relates scene radiance to image brightness) is generally non-linear. Secondly, the imaging process is inherently affected by various sources of noise, ranging from the shot photon noise which depends on radiation physics to the technological read-out noise. An accurate photometric calibration should allow to recover not only the camera response function but also the imaging system noise (hereinafter, camera noise, CN) characteristics.

However, most of the works in literature dealing with photometric calibration focus on recovering the camera response function. The classical and most popular approach consists in imaging a uniformly illuminated chart with patches of known reflectance,

such as the Macbeth chart, as done in [10]. Recently, a number of algorithms have been proposed ("chartless" or "self-calibration" methods) which estimate the camera response function from multiple images of an arbitrary scene taken with different exposures ([13, 31, 29, 17]). Only a few works exist that try to extract a model of the CN. In [9] the authors analyze the noise of the cameras based on ionization sensors, such as the vidicon and the CCD cameras. In particular, they single out three different sources of noise. The electronic noise (leakage currents and Johnson noise) is modeled as Gaussian and spatially stationary (i.e., independent of the pixel position), with correlations expected only in the read-out scan direction. The photon noise, due to the quantum nature of light, is considered spatially stationary as well, but it is statistically characterized by a Poisson distribution, thus a variance depending on the signal level is expected. At last, the fixed pattern noise for the CCD cameras is considered. By experimentally measuring the pixel intensity variations for uniformly dark and uniformly bright scenes, the authors validate the proposed models for the electronic and the photon noises. In [11] the statistics of the granular camera noise of high-quality pick-up tube cameras are investigated. First of all, the authors highlight the relative unimportance of chrominance noise with respect to luminance noise. Then, the granular camera noise for each pixel is shown to be a stationary random process, not to be Gaussian, not to be zero-mean and to have a variance that depends on the pixel luminance and chrominance level. Finally, by a spatio-temporal extension of the Kolmogorov-Smirnov test, the authors demonstrate that the noise is white in the temporal domain but mildly colored in the spatial one. In [30] the noise of the CCD sensors is analyzed. The authors recognize three different noise regimes, each one corresponding to a different range of the signal level and to the predominance of a particular noise component: the low regime is dominated by the CCD on-chip amplifier noise (read-out noise), the intermediate regime by the photon shot noise and the high regime by the fixed pattern noise. Although these works discuss the statistical characteristics of the CN, they do not propose methods to extract these statistics. In [20] the CCD cameras imaging process is accurately modeled by explicitly accounting for both the two classes of spatially stationary and non-stationary noise sources corrupting the digital pixel values. By making a priori assumptions on the statistics of the different noise components, the spatially uniform noise is shown to be a zero-mean random variable and to have a variance linearly depending on the signal level. Both the classes of noise are estimated by using flat field images. Also in [38] the authors model accurately the various steps of the CCD imaging process and the different noise sources. Besides, they account for some of the artificial transformations possibly occurring in the current cameras, such as the white balancing, the gamma correction and the auto gain control. They propose a self-calibration procedure that utilizes a set of images of an arbitrary

static scene taken under different exposure settings. The series of values for each pixel are separately considered, thus decoupling the temporal random noise from the spatially non-stationary noise. In this way, by a non-parametric iterative algorithm the authors infer the camera response function. Finally, the variances of the shot photon noise, of the thermal noise and of the read-out noise are separately estimated. In [40] the photometric calibration is performed by using the method presented in the previous work. Differently from all the other approaches, in this work the authors show how the noise level, given a camera response function, can be seen as a function of the measured pixel brightness instead of the incoming radiance. The noise is modeled as a zero mean Gaussian random variable.

Hence, only a few works exist dealing with the self-calibration of the CN characteristics ([38],[40]). All of them rely on a priori assumptions regarding both the different types of noise sources they account for and the parametric form of the statistical models employed. This yields methods that depend on the actual structure of the imaging system device. Besides, these approaches extract the CN characteristics by processing images taken at different exposures.

We present a simple self-calibration algorithm aimed at inferring a reliable statistical model of the CN. In particular, the proposed approach models the imaging system as a "black-box" and uses a non-parametric statistical model for the CN. This yields a method totally independent of the actual structure of the imaging device. Besides, the model is extracted directly from the pixel intensity variations measured along a short *training* sequence of an arbitrary scene. The only a priori assumption, widely accepted in literature and confirmed by experiments, is that the noise level for a pixel only depends on its brightness value.

2.1.1 Probabilistic Framework and Theoretical Assumptions

Let us consider the scalar integer brightness values $p_i(t)$ that a pixel $i \in [1; n]$ (where n is the total number of pixels) assumes in the 8-bit grey level frames of a time (frame) interval I. Then we define a *time series* P_i^I as follows:

$$P_i^I = \{p_i(t) : t \in I\} \tag{2.1}$$

Now let us define the *relative temporal histogram* $h_i^I(v)$ as the relative frequency of the values $v \in [0; 255]$ the pixel i assumes along I. μ_i^I and med_i^I represent the *temporal mean* and the *temporal median*, respectively. Using the terms of Mathematical Statistics, the values that a pixel i may assume over time can be considered as a one-sided discrete time scalar stochastic process called *pixel stochastic process* ($P_i(t)$). Therefore, a time series P_i^I represents a realization of the underlying random process $P_i(t)$. A pixel stochastic process is characterized by ensemble statistics, such as the *ensem-*

ble probability density function $pdf_i(t, v)$, the *ensemble mean* $\mu_i(t)$ and the *ensemble median* $med_i(t)$.

The proposed algorithm infers the CN model from a scene not necessarily free of moving objects but where the background must be stationary. Hence, let us consider a pixel i belonging to a stationary background, where lighting changes and background motion (e.g., camera vibrations or swaying trees) are negligible (that is, the pixel measures a constant radiance). The stochastic process $P_i(t)$ of pixel i can be modeled as the sum of two distinct processes:

$$P_i(t) = B_i(t) + N_i(t) = B_i + N_i(t) \tag{2.2}$$

where $B_i(t)$ is a deterministic constant process B_i, giving the value of the background pixel as if it was measured by an ideal noiseless camera, and $N_i(t)$ is a stochastic process representing the CN affecting the pixel. Besides, as for $N_i(t)$ in case of a pixel measuring a constant radiance we assert the following three claims:

- C. 1: $N_i(t)$ is modeled as a scalar stochastic process, that is any spatial statistical dependence is neglected;
- C. 2: $N_i(t)$ is a stationary and ergodic stochastic process (briefly, a *SESP*);
- C. 3: $N_i(t)$ statistical properties only depends on B_i, that is on the pixel i deterministic ideal noiseless value.

Based on the claims, for 8-bit grey level sequences the CN can be modeled by means of 256 scalar SESP , $N_w(t)$, one for each possible integer brightness value $w \in [0; 255]$. Hence expression 2.2 becomes:

$$P_i(t) = B_i + N_{w=B_i}(t) \tag{2.3}$$

Since a SESP is completely defined by its ensemble probability density function, the statistical CN model (hereinafter, sCNM) we are going to infer consists of 256 ensemble probability density functions $pdf_w(v)$. As for C. 2, stationary (*stat*) means that ensemble statistics are constant over time, while ergodic (*erg*) means that the temporal statistics computed for a single realization are a good estimation of the underlying SESP ensemble statistics if the cardinality of the sample set is greater than a certain value c_{erg}. Hence, for $N_w(t)$ and for a SESP in general:

$$\begin{cases} pdf_i(t, v) \overset{stat}{=} pdf_i(v) \overset{erg}{\simeq} h_i^I(v) \\ \mu_i(t) \overset{stat}{=} \mu_i \overset{erg}{\simeq} \mu_i^I \\ med_i(t) \overset{stat}{=} med_i \overset{erg}{\simeq} med_i^I \end{cases} \tag{2.4}$$

Let us now look at equation 2.3: since both $N_w(t)$ (C. 2) and $B_i(t)$ (it is a deterministic and constant process) are SESP, we can state that the stochastic process $P_i(t)$ for a stationary background pixel is a SESP as well, thus satisfying expression 2.4.

2.1.2 The Camera Noise Model Extraction Algorithm

The non-parametric sCNM is generated by processing a training sequence (corresponding to a frame interval I) of few seconds acquired by a static camera and free of moving objects. As well as other initialization methods ([18], [2]), our algorithm relies on a background that must be stationary along the training sequence. Since in practice the lower the elapsed time the higher the probability of the stationarity assumption to be fulfilled, we aim to keep the training sequence as short as possible. Hence, we process a training sequence such that $card(I) = c_{erg}$. To extract the sCNM, first we compute statistics of the stationary background process for each pixel, then we use these statistics for the non-parametric inference.

In particular, for each pixel i we build the temporal relative histogram $h_i^I(v)$. Since the background is assumed to be stationary along the training sequence, equation 2.3 and expression 2.4 (Section 2.1.1) hold. Thus, the attained temporal relative histogram $h_i^I(v)$ represents the ensemble probability density function $pdf_i(v)$ of the pixel i stationary background process. Hence, we vote the temporal mean μ_i^I (which is equivalent to the ensemble mean μ_i) as the ideal noiseless background value for each pixel:

$$B_i = \mu_i^I \tag{2.5}$$

By using the computed statistics and by exploiting the claims asserted in Section 2.1.1, we can extract the non-parametric sCNM. From expression 2.2 and equation 2.5:

$$N_i(t) = P_i(t) - B_i = P_i(t) - \mu_i^I \tag{2.6}$$

Hence, the ensemble probability density function $pdf_i^N(v)$ of the CN stochastic process $N_i(t)$ for each pixel i can be deducted by simply translating by an horizontal offset $O_i = -B_i = -\mu_i^I$ the previously computed ensemble probability density function $pdf_i(v)$ of the pixel i background process. Following from the assumption that the statistics of the CN affecting a stationary pixel only depends on its ideal noiseless intensity (C. 3):

$$N_{w=B_i}(t) = N_i(t) = P_i(t) - B_i \tag{2.7}$$

Practically speaking, the time series of the CN values for a pixel i can be considered not just a realization of the random process $N_i(t)$ representing the CN for that pixel, but also a realization of the more general stochastic process $N_{w=B_i}$ representing the CN for the grey value $w = B_i$. Therefore, the time series of the CN values for all the pixels which have had the same background value according to equation 2.5 represent different realizations of the same random process. Hence, for each grey level w we sum the attained $pdf_i^N(v)$ of all the pixels i to which we assigned w as the background value. Then we normalize the outcome, thus attaining a unique non-parametric ensemble probability density function (that is a relative histogram) $pdf_w(v)$ for each grey level

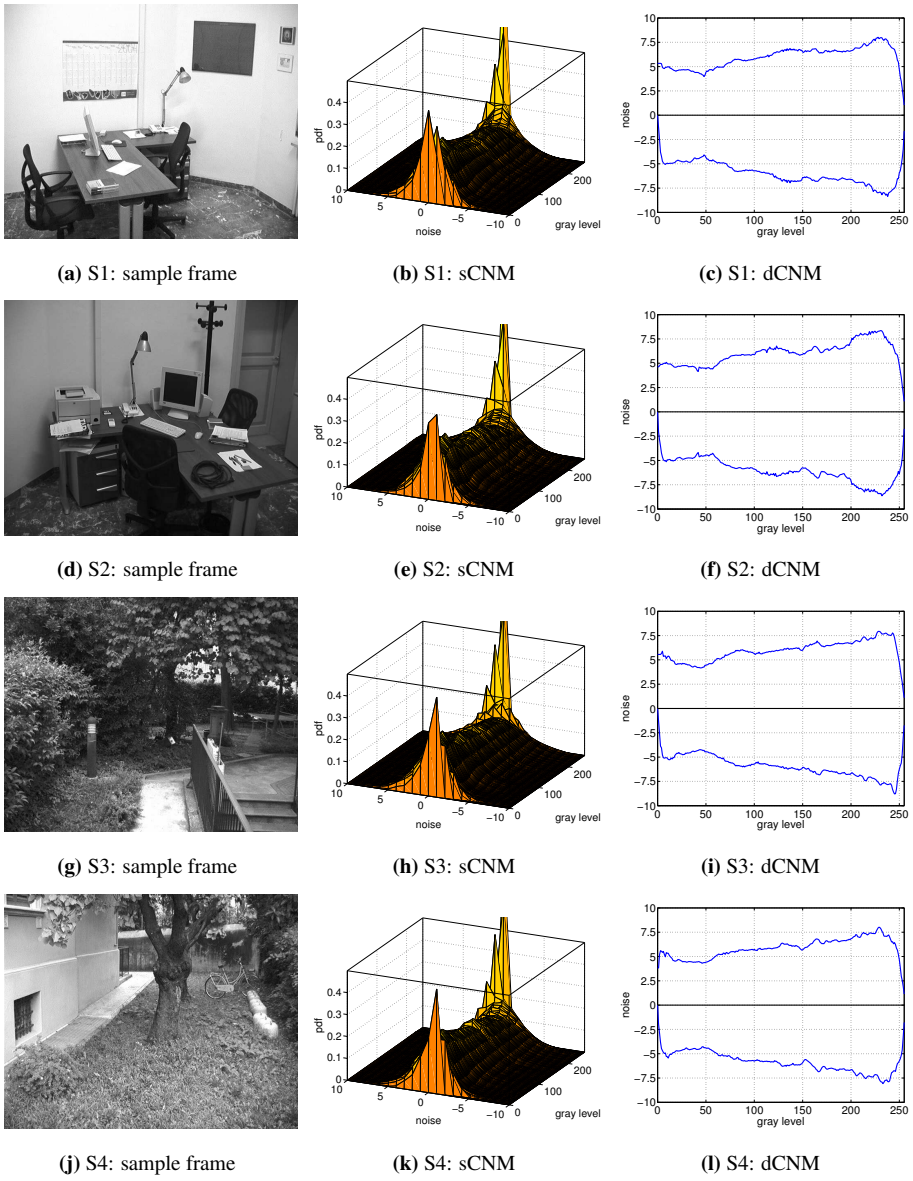
w, that is the sCNM. Finally, by extracting a lower and an upper percentile ($perc_w^{low}$ and $perc_w^{up}$, respectively) with fixed ranks from each $pdf_w(v)$, we also build a deterministic CN model (dCNM).

2.1.3 Experimental Results

In order to validate the model and the method we conceived, we show how for a given imaging device the extracted CN models are strongly scene-independent. In fact, if the noise for a pixel depends significantly on other factors apart from the brightness level of that pixel (e.g., on the pixel position in the image or on the brightness level of the pixel's neighbors), inferring the CN models from training sequences of different stationary scenes would give rise to remarkable dissimilarities among the models themselves. In particular, in figure 2.1 we show the results for four test sequences (S1, S2, S3 and S4) acquired with a Sony DCR-TRV900E and sampled in progressive scan mode at 12,5 Hz at a resolution of 720x576. Figures 2.1(a,d,g,j) show a sample frame for each test sequence. We have chosen two indoor (S1 and S2) and two outdoor (S3 and S4) sequences in order to represent very different lighting conditions. The CN models extracted for the four sequences are shown as well. In particular, figures 2.1(b,e,h,k) depict the statistical CN models, that is the 256 probability density functions representing the CN distribution for each grey level. Figures 2.1(c,f,i,l) show the deterministic CN models, that is the 256 couples of lower and upper percentiles. The strong similarity of the inferred deterministic CN models allow to assess the validity of the proposed CN model extraction approach. Finally, we can say that a "black-box" modeling of the imaging system together with a non-parametric form of the noise model and a fully automatic procedure to extract the model itself give rise to a really "general-purpose" approach.

2.2 Background Initialization

The background subtraction technique relies on the feasibility to have a reliable background model at disposal along the processing stage. Hence, the background model has to be initialized and then updated. As far as the background model initialization is concerned, some algorithms ([39],[33]) infer the model by assuming to have a *bootstrap* sequence free of moving objects at disposal. These methods fail when the area being monitored can not be easily controlled, so that a sequence of background frames can not be acquired. As for the methods dealing with the presence of moving objects, they can be divided into two main classes: the "blind" and the "selective" methods. The formers generate a background model for each pixel by means of temporal statistics computed using the whole time series of the pixel intensities. These background sta-



(a) S1: sample frame **(b)** S1: sCNM **(c)** S1: dCNM

(d) S2: sample frame **(e)** S2: sCNM **(f)** S2: dCNM

(g) S3: sample frame **(h)** S3: sCNM **(i)** S3: dCNM

(j) S4: sample frame **(k)** S4: sCNM **(l)** S4: dCNM

Figure 2.1: Camera noise model extraction results for four test sequences.

tistics may be "dirty", by retaining information not just about the background process but also about some possible foreground processes due to the moving objects covering the pixel along the bootstrap sequence. On the contrary, the selective approaches try to isolate the background process for each pixel, thus computing "clean" background statistics. Moreover, as for the selective methods a further distinction can be done between the temporal and the spatio-temporal methods. The formers extract the background model for each pixel by using just the intensities assumed by the pixel itself.

The latters exploits also spatial information, that is the values of other pixels (only the neighbors or even all the pixels) in the image.

As for the blind methods, in [36] the authors vote the temporal mode as the background value for each pixel, implicitly assuming that the background value will be more frequent than any other possible foreground value. The temporal median is used in [16], based on the assumption that the background at every pixel will be visible more than fifty percent of the frames during the bootstrap sequence. Although the blind methods are very efficient, in case of sequences containing many moving objects they need a great number of bootstrap frames to extract a reliable background model.

As regards the selective temporal methods, in [19] a two-stage algorithm is used to generate the background model. The first stage extracts a temporary background by means of a median filter applied to a bootstrap sequence of several seconds. The second stage uses that background for detecting reliable background regions where to extract the clean statistics to be used for generating the final background model. This method is similar to our approach, but it requires a much longer bootstrap sequence (more than 10 seconds). The authors in [35] propose a single-stage algorithm, based on a simple *background detection* consisting of a temporal frame difference followed by a morphological opening. As soon as a pixel is detected as belonging to the background, its value is voted as the final background value. The method is efficient and needs a low number of bootstrap frames, but it easily includes in the background model the pixel intensities due to the foreground objects. In [28] the "adaptive smoothness method" is presented. It finds intervals of stable intensity for each pixel, then uses a heuristic which chooses the longest and most stable interval as the one most likely representing the background process. This approach is effective, but it requires a quite long batch processing of the bootstrap sequence. In [26] a running mean and variance for each pixel are incrementally computed over the bootstrap frames. When the variance drops below a predefined threshold, the pixel is considered stable and the mean is voted as the background value. The method is quite efficient, but stationary foreground objects can be easily included in the background model. In [12] the temporal evolution of each pixel intensity is modeled by means of a HMM. The parameters of each HMM are inferred by using a standard Baum-Welch procedure, then are used to build the background model for each pixel. The method is effective, but the computational burden of the training procedure leads to a long batch processing phase.

As for the selective spatio-temporal methods, in [2] the author presents a single-stage approach based on the Bayes theory. It performs a background detection by using a simple temporal frame difference. It exploits spatial information, in fact the information about the whole reliable background regions are used to update the likelihood-based background model of each pixel. In case of slow moving objects this method

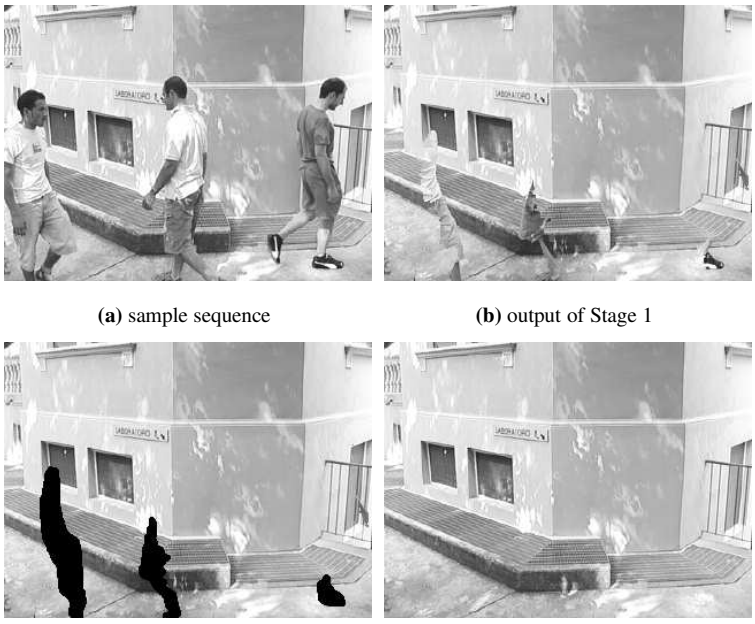
can include in the background model the foreground pixel intensities, thus requiring a long bootstrap sequence. In [18] the authors isolate the background process by means of a two-stage algorithm performing a batch processing of the bootstrap sequence. The first stage works in the time domain, locating for each pixel all the time intervals of stable intensity. The second stage exploits spatio-temporal information for choosing the time interval most likely representing the background process. In particular, the optical flow in the neighborhood of the pixel is computed for each bootstrap frame. Then, from the chosen time interval the background model is extracted . This method is effective, but the optical flow computation make it less efficient than all the previous approaches. In [27] a two-stage algorithm is presented, called "ComMode" (Competitive Mode Estimation) by the authors. As well as in [18], the first stage uses temporal information and detects the time intervals of stable intensity for each pixel. To this purpose, a region growing algorithm in the time domain is used. The second stage exploits spatial information to choose the *best* time interval. In particular, a competitive spatial propagation of the clusters (called "modes") detected in stage 1 is performed until stability is reached (5-10 iterations). Finally, for each pixel the temporal mean of the chosen cluster is voted as the background value. The approach is effective, but as well as the ones in [18] and [28] it requires a long batch processing of the bootstrap sequence.

We propose a novel selective spatio-temporal approach which allows to generate a reliable deterministic model of a stationary background by using a bootstrap sequence of few seconds where moving objects can also be present. By performing a sequential processing of the frames of the sequence, it aims to be efficient and effective. The algorithm works with pixel-wise temporal statistics and consists of three subsequent stages.

2.2.1 The Multi-stage Background Initialization Algorithm

The deterministic background model is generated by means of a three-stage algorithm. The first two stages isolate the background process, thus voting the temporal median as the good background value. In the third stage a model of the imaging system noise is inferred by applying the algorithm presented in Sec. **??** to the background process statistics computed in the second stage. Then, the noise model is used to complete the background model generation. We divide the bootstrap sequence into three consecutive time (frame) intervals $I^1 = [t_0; t_1]$, $I^2 = (t_1; t_2]$ and $I^3 = (t_2; t_3]$, each one corresponding to a different stage. While t_1 and t_2 are fixed (we use $t_1 = 10$, $t_2 = 50$), t_3 varies depending on the number of frames necessary to complete the background model initialization. The algorithm relies on a background that must be stationary along the bootstrap sequence. In figure 2.2(a) we show a frame of a sample 8-bit grey level

bootstrap sequence, while figures 2.2(b,c,d) depict the backgrounds output of the three stages.



(a) sample sequence **(b)** output of Stage 1

(c) output of Stage 2 **(d)** output of Stage 3

Figure 2.2: Subsequent steps of the background initialization algorithm.

Stage 1: rough background

We extract a temporary *rough* background to be used in the further stage. To this purpose, we try to isolate the stationary background process for each pixel by performing a rough background detection for each frame in I^1. In particular, the background regions are detected by means of a simple temporal two-frame difference with an a priori fixed threshold T_1, spatially (over the different pixels) and temporally (over the different frames) constant. To improve the reliability of the detected background regions, we aim at minimizing the number of false negatives among the changed pixels by using a *low* value for T_1 (we use $T_1 = 10$, which is a low value even for low-noise cameras) and by performing a series of morphological operations on the computed binary image. In particular, we use an initial size-filtering operator (area-opening) followed by a morphological closing with a kernel of size 3x3 and by a filling. Hence, for each pixel i we compute the selective absolute temporal histogram $H_i^{I_i^1}$ by using just the sample values assumed in the set of frames $I_i^1 \subseteq I^1$ in which i has been detected as a background pixel. Finally, we vote the selective temporal median $med_i^{I_i^1}$ as the background value

for each pixel i:

$$\hat{B}_i^r = med_i^{I_i^1} \tag{2.8}$$

This could be a rough background because of the rough background detection employed (the temporal two-frame difference suffers of well known limits).

Stage 2: good background

To compute more reliable background process statistics, for each frame in I^2 we perform a background subtraction with the background just extracted by using a threshold $T_2 = T_1/\sqrt{2}$. Then, we apply the same morphological operations used in stage 1, thus identifying more reliable background regions where to infer the new background statistics. In particular, as well as in the previous stage we compute for each pixel i the selective absolute temporal histogram $H_i^{I_i^2}$ by using the sample values assumed in the set of frames $I_i^2 \subseteq I^2$ in which i has been detected as a background pixel. If $card(I_i^2) \leq C_{erg}$ (we use $C_{erg} = 20$), i is marked as an "unreliable" pixel and a background value will be inferred in stage 3. On the contrary, the computed statistics are regarded as reliable and the selective temporal median $med_i^{I_i^2}$ is voted as the *good* background value:

$$\hat{B}_i^g = med_i^{I_i^2} \tag{2.9}$$

Stage 3: background completion

To extract a background value for the "unreliable" pixels, the model of imaging system noise is inferred by applying the algorithm presented in Sec. **??** to the selective background process statistics computed in Stage 2. Then, this model is used to complete the background model generation as follows. The CN allows to identify time (frame) intervals of stationary intensities (i.e. grey level variation can be explained well by the noise model). In fact, a necessary condition for a pixel intensity to be stationary is that it is affected by the variations due just to the CN. Therefore, if the measured distribution of these variations computed around a *central* real value V *matches* with the distribution of the inferred dCNM corresponding to the integer grey level $[V]$, the pixel can be considered stationary. Hence, for each "unreliable" pixel i, we search incrementally for the first time (frame) interval $I_i^{stat} \subseteq I^3$ of stationary intensities having a sufficient length in terms of frames ($card(I_i^{stat}) > C_{stat}$, we use $C_{stat} = 10$). To this purpose, for each pixel we use a FIFO queue to store the last C_{stat} sample intensities and at each new frame $t \in I^3$ we compute the distribution of the variations of the intensities in the queue using the computed median $med_i^{I_i^{stat}}$ as the central value. To perform a simpler matching operation, we extract a lower and un upper percentile from the distribution, thus comparing them with the ones in the dCNM corresponding to the integer grey level $[med_i^{I_i^{stat}}]$. If they match, the computed median is voted as the background value

and the pixel i is removed from the "unreliable" pixels set. The algorithm stops when the percentage of the number of "unreliable" pixels in respect of the total number of pixels either decreases below a predefined threshold or becomes stable.

2.2.2 Experimental Results

To test the proposed approach, we have compared its performance with the ones of two different selective spatio-temporal background generation algorithms. In particular, we have chosen the methods proposed in [2] and in [18]. We have run the algorithms on several sequences, having different amounts of motion. Figure 2.2.2 shows a sample frame for each of the two test sequences (*S1* and *S2*) chosen to outline the results. They have been taken by a static CCD camera, sampled in progressive scan mode at

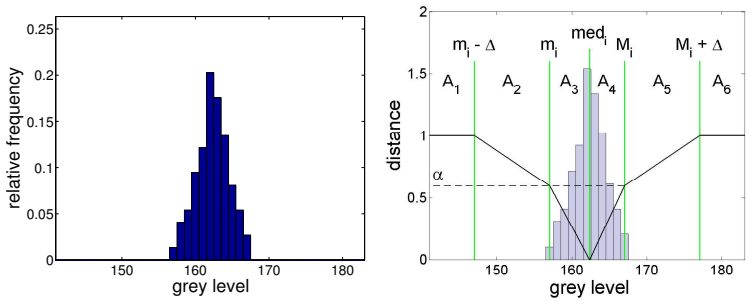
(a) S1: a sample frame **(b)** S2: a sample frame

Figure 2.3: A sample frame for each of the two test sequences.

12,5 Hz at a resolution of 320x240. The sequences represent approximately the same background scene but are characterized by an increasing amount of motion, given by the number of persons walking in the scene (two and three, respectively). Since we want to compare the algorithms on the basis of the quality of the estimated background, we need a ground truth and a function to compute the *distance* from that ground truth.

Ground truth and distance function

To generate a reliable ground truth for the generated backgrounds, we have taken the test sequences so that they contain two different subsequences. The former, that we call the *truth* subsequence, consists in an interval of frames I^T representing the background scene free of moving objects. The latter, namely the *estimation* subsequence, is the actual test sequence on which the algorithms will be run and consists of a different interval of frames I^E imaging the same background scene but in which moving objects can be present. From the truth subsequence, for each pixel i we compute the relative temporal histogram $h_i^{I^T}$. The set of all these histograms represents the ground truth. Figure 2.4(a) shows the ground truth for a sample pixel.



(a) ground truth **(b)** distance function

Figure 2.4: Ground truth and distance function for a pixel.

In order to define the distance function, from the truth subsequence we extract also the temporal median $med_i^{I^T}$, the temporal minimum $m_i^{I^T}$ and the temporal maximum $M_i^{I^T}$ for each pixel i (in figure 2.4(b) and in expression 2.10 we drop the superscript I^T). Hence, we define a *local* distance d_i which represents a measure of the distance between the ground truth of the pixel i, that is the histogram $h_i^{I^t}$, and the background value $\hat{B}_i$ estimated for the same pixel by the background generation algorithm. It is a piece-wise linear continuous function, represented in figure 2.4(b) for the same sample pixel of figure 2.4(a) and mathematically defined as follows:

$$d_i(\hat{B}_i) = \begin{cases} 1 & \text{if } \hat{B}_i \in A_1 \\ 1 - \frac{(1-\alpha)(\hat{B}_i - m_i + \Delta)}{\Delta} & \text{if } \hat{B}_i \in A_2 \\ \alpha - \frac{\alpha(\hat{B}_i - m_i)}{med_i - m_i} & \text{if } \hat{B}_i \in A_3 \\ 0 & \text{if } \hat{B}_i = med_i \\ \alpha - \frac{\alpha(\hat{B}_i - med_i)}{M_i - med_i} & \text{if } \hat{B}_i \in A_4 \\ 1 - \frac{(1-\alpha)(M_i + \Delta - \hat{B}_i)}{\Delta} & \text{if } \hat{B}_i \in A_5 \\ 1 & \text{if } \hat{B}_i \in A_6 \end{cases} \tag{2.10}$$

where α and Δ are a priori fixed parameters having the same value for all the pixels (we use $\alpha = 0.6$ and $\Delta = 10$) and A_i, $i = 1, \cdots, 6$ are the six *pieces* (intervals) the function domain is divided into (figure 2.4(b)). Then, we define the *global* distance D of the estimated background from the ground truth by the following simple expression:

$$D = \frac{\sum_{i=1}^{n} d_i}{n} \tag{2.11}$$

where n is the total number of pixels. While the meaning of the global distance is clear, representing a simple averaging of all the local distances, it is worth spending some words about the choice of the local distance function. The quality criteria that drove this choice is based on the idea that the background will be used to detect foreground points by thresholding the absolute difference between the current pixel intensity and

the estimated background value. Since the median of a random variable X is the statistical estimator $\hat{X}$ that minimizes the expected value of the absolute error $|X - \hat{X}|$, the median of the ground truth histogram for a pixel i is the *best* value a background generation algorithm can estimate for that pixel ($d_i(\hat{B}_i) = 0$). In fact, it allows the use of the lowest threshold in the background differencing stage, thus minimizing the false negatives due to the possible camouflaging between the foreground objects and the background. As for the rest of the function, we assign the same maximum distance ($d_i(.) = 1$) to all the values differing more than a fixed parameter Δ from the ground truth minimum m_i (A_1) or maximum M_i (A_6). In other words, we consider all these values equally *wrong* with reference to the background differencing stage. If we consider the ground truth as a *perfect* estimate of the background model to be generated, a value $\Delta = 0$ could be used. Nevertheless, this is not the case, mainly for two reasons: firstly, the truth and the test subsequences are temporally deferred, secondly, the ground truth is inferred from a finite number of sample background values. Hence, the parameter Δ defines a sort of *tolerance area* around the ground truth ($A_2 \cup A_5$). Finally, the parameter α manages the slope of the distance function in the ground truth ($A_3 \cup A_4$) and tolerance ($A_2 \cup A_5$) intervals.

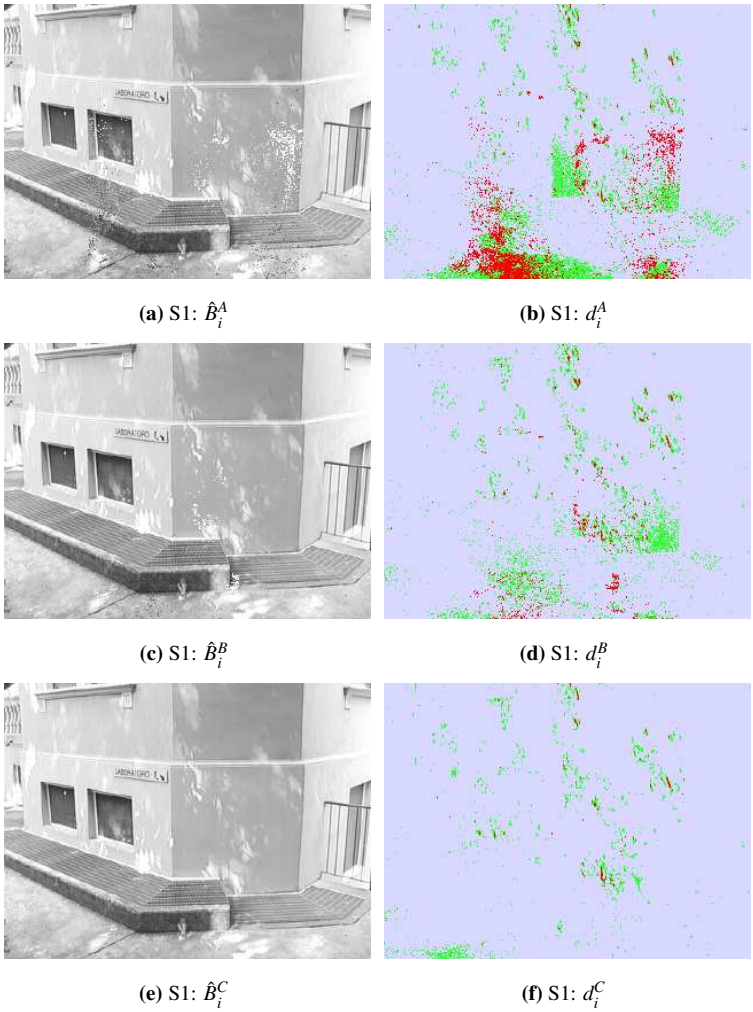
Algorithms comparison

For each test sequence, we have generated the ground truth from the truth subsequence, then we have run the three compared algorithms, hereinafter denoted with A ([2]), B ([18]) and C (the proposed approach), on the estimation subsequence, thus attaining the three different background models $\hat{B}_i^A$, $\hat{B}_i^B$ and $\hat{B}_i^C$, respectively. Hence, we have computed all the local distances by expression 2.10, thus attaining the three *local distance maps* d_i^A, d_i^B and d_i^C. Finally, by equation 2.11 we have computed the global distances D^A, D^B and D^C. Figure **??** depicts the background models and the local distance maps generated by the compared algorithms. As for the maps, to visualization purposes we have divided the range of the possible distance values into three classes, related to the *pieces* of the distance function domain: $C_1 = A_3 \cup med_i \cup A_4$ (*low* distances, azure-light grey in the figure), $C_2 = A_2 \cup A_5$ (*medium* distances, green-grey) and $C_3 = A_1 \cup A_6$ (*high* distances, red-black). Table 2.1 shows the global distances. From these results

	A	B	C
S1	0.32	0.24	0.13
S2	0.31	0.26	0.14

Table 2.1: Global distances.

we can state that the proposed approach generates a background model of higher *global*



(a) S1: $\hat{B}_i^A$ **(b)** S1: d_i^A

(c) S1: $\hat{B}_i^B$ **(d)** S1: d_i^B

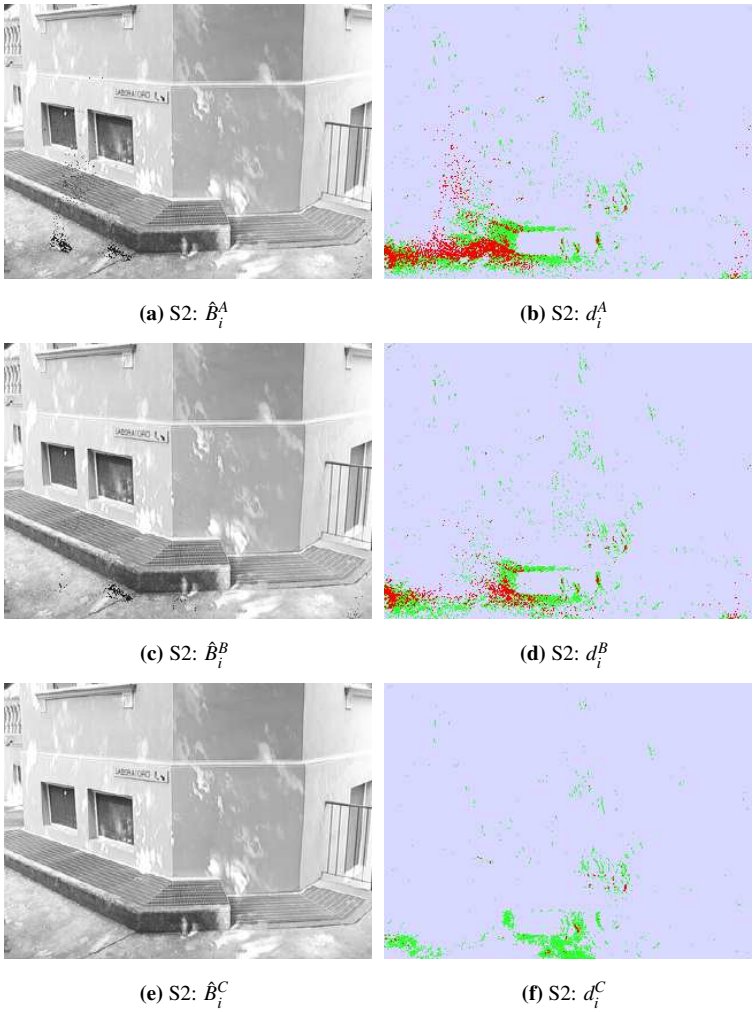
(e) S1: $\hat{B}_i^C$ **(f)** S1: d_i^C

Figure 2.5: Generated backgrounds and distance maps for S1.

quality in respect with the other compared methods.

2.3 Background Subtraction and Updating

Many temporally adaptive change detection algorithms based on background subtraction have been proposed in the past. In [39] the authors model the background process for each pixel as a unique spatially independent stochastic gaussian process. The parameters of the gaussian distribution representing the ensemble pdf for each pixel are initialized through a bootstrap sequence free of moving objects. While the mean is recursively updated using a simple adaptive filter, the covariance matrix is extracted once and for all, thus yielding a threshold that does not adapt to scene changes. Authors in



(a) S2: $\hat{B}_i^A$ **(b)** S2: d_i^A

(c) S2: $\hat{B}_i^B$ **(d)** S2: d_i^B

(e) S2: $\hat{B}_i^C$ **(f)** S2: d_i^C

Figure 2.6: Generated backgrounds and distance maps for S2.

[15] model the pixel process instead of the background process only: a spatially independent random process is used for each pixel, representing both the background and the foreground processes due to moving objects and to cast shadows possibly covering the pixel. A weighted sum of three gaussian distributions (background, moving objects and shadow distributions) is used to model the ensemble pdf for each pixel. Nevertheless, the background is still represented by a unique gaussian random process for each pixel. Background subtraction consists in choosing for each pixel which of the three classes has the highest a posteriori probability. An incremental EM algorithm is used to both learn and update the distribution parameters. A generalization of the previous approach is presented in [37]. Each pixel is still modeled as a spatially independent stochastic process having a mixture of K (a small number from 3 to 5) gaussian dis-

tributions as ensemble pdf. At each time step and for each pixel, the distributions are ordered according to the value of a ratio attained dividing the evidence of the distribution by its variance. The first B distributions are selected to represent the background process and if the pixel value is not represented by any of these distributions it is classified as moving. The parameters of the mixture are updated by means of a simple adaptive filter. In [22] authors improves the method outlined in [37]. In particular, they present a different approach for initialising and for updating the parameters of the mixture model, based on an incremental EM algorithm. A further generalization of the previous approaches is outlined in [14]. The ensemble pdf of the spatially independent stochastic process of each pixel is modeled in a non-parametric manner. At each time step and for each pixel the ensemble pdf is non-parametrically estimated by means of a gaussian kernel estimator function applied to a window of recent sample intensities for that pixel. The model update consists in simply shifting the samples window. Even though the methods described in [15]-[14] model the background more and more accurately, their complexity make them not suitable to be used efficiently in many real-time applications.

In this Section we show how the imaging system noise model extracted by the procedure illustrated in Section 2.1 can be used to attain a background subtraction approach which achieves a good trade-off between time performance and quality of the detection. In fact, by scaling all the percentiles of the inferred dCNM by a unique factor greater than one, we attain 256 couples of thresholds ($ts_{inf}(v)$ and $ts_{sup}(v)$), one for each grey level $v \in [0; 255]$, to be used in the background subtraction. In this way we retain both the advantages arising from the simplicity of setting up a unique threshold and the effectiveness of 256 different couples of thresholds. This results in an effective yet efficient thresholding operation. In particular, for each pixel the algebraic difference between the current frame $F_{i,j}$ and the background $B_{i,j}$ is computed. The outcome is then compared with the couple of thresholds $ts_{inf}(v)$ and $ts_{sup}(v)$, depending on the current background value $B_{i,j}$, thus attaining a binary image $M_{i,j}$ representing the moving pixels:

$$M_{i,j}(t) = \begin{cases} 1 & \text{if} \quad F_{i,j}(t) - B_{i,j}(t) \notin A_{i,j}(t) \\ 0 & \text{if} \quad F_{i,j}(t) - B_{i,j}(t) \in A_{i,j}(t) \end{cases} \tag{2.12}$$

where $A_{i,j} = [ts_{inf}(B_{i,j}(t)); ts_{sup}(B_{i,j}(t))]$.

The deterministic background is updated by a simple and efficient adaptive recursive filter:

$$B_{i,j}(t+1) = (1-\alpha)B_{i,j}(t) + \alpha F_{i,j}(t) \tag{2.13}$$

where $\alpha \in [0; 1]$ represents the adaptation rate.

2.3.1 Experimental Results

Tests were performed on several 8-bit grey level sequences representing typical surveillance scenes, taken by a single stationary CCD camera and sampled at 25 Hz at a resolution of 320x240. The target PC is an AMD Athlon MP 1800+, 1 GB RAM. The experimental results we accomplished assess both the efficiency and the effectiveness of our algorithm. As for the effectiveness, we stated that our method acts as we apply 256 couples of different thresholds, one for each grey level. As one could infer, it is impracticable such an experiment, therefore we will assess the effectiveness by showing the capability of our algorithm to detect moving pixels in situation of camouflage between the background and the moving objects. The results of our algorithm run on four test sequences are shown in figure 2.7. The attained change masks are very accurate, thus validating the proposed approach. As regards time performance, our method reveals to be very efficient, working off-line at 40 fps.

2.4 Considerations

The presented background subtraction algorithm, as well as all the algorithms based on a continuously updated background model, has two problems:

a- a blind or a selective background updating procedure has to be chosen. If a blind procedure is chosen, slowly moving objects may be included in the background model. On the contrary, false changes due to disturbance factors may be continuously detected, since they can not be absorbed in the background model.

b- the background model adaptation rate must be chosen accurately. However, it always represents a trade-off between the ability of the updating procedure to adsorb false changes and the risk to include foreground objects in the background model

However, sudden changes of pixel intensities due to disturbance factors can not be dealt with successfully by this class of background subtraction algorithms.

(a) background **(b)** background

(c) change mask **(d)** change mask

(e) change mask boundaries **(f)** change mask boundaries

Figure 2.7: The change detection results.

Chapter 3

Disturbance Factors Invariant Change Detection

In this chapter we present a change detection algorithm which is very different from the one presented in chapter 1. In fact, detection of changes in a pixel is not performed by computing a distance between the measured current intensity and a temporally adaptive statistical model of the scene background appearance at the same pixel. Instead, classification of a pixel as changed or unchanged is carried out by comparing the measured intensities of a patch of pixels (around the considered one) in the current frame with the intensities of the same patch in the background image. In particular, based on an accurate investigation of the possible disturbance factors effects on the image intensities, a Maximum-Likelihood isotonic regression procedure is proposed to detect changes.

The chapter is organized as follows. In section 3.1 the problem of change detection for disturbance factors invariant approaches is defined and formalized. An accurate investigation of the possible effects of disturbance factors on the measured intensities is carried out in section **??**. The proposed algorithm is presented in section 3.3. Experimental results are discussed in section **??** and conclusions are drawn in section 3.5.

3.1 Problem Definition and Formalization

Let us consider two grey level images captured at different times by the same stationary camera. To the purpose of detecting changes, we can identify the images as the background B and the currently processed frame F (figures 3.1(a,b)):

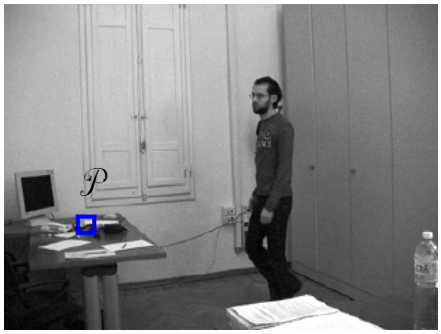
$$B, F: \quad D \ni \boldsymbol{p} = (i, j) \;\mapsto\; g = B, F(\boldsymbol{p}) \in R \tag{3.1}$$

It is worth pointing out that the images have a common domain $D \subset \mathbb{Z}^2$ and a common range $R \subset \mathbb{Z}$ since they are acquired by the same imaging device. Moreover, since the device is stationary a common pixel in the two images measures the radiance of the same portion of the scene. Let $\boldsymbol{p} = (i, j)$ be a pixel and $\mathcal{P}(\boldsymbol{p})$ a connected domain patch (i.e a connected set of pixels) containing $\boldsymbol{p}$:

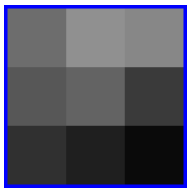
$$\boldsymbol{p} \in \mathcal{P}(\boldsymbol{p}) \subset D \tag{3.2}$$

Although what we are going to say is valid for a generic patch, for the sake of simplic-

(a) sample domain patch $\mathcal{P}$ in the background B

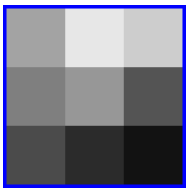


(b) sample domain patch $\mathcal{P}$ in the current frame F



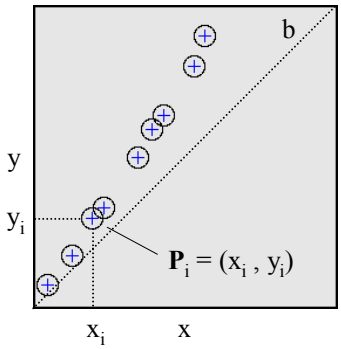
x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

(c) background range patch $B(\mathcal{P}) = \boldsymbol{X}$



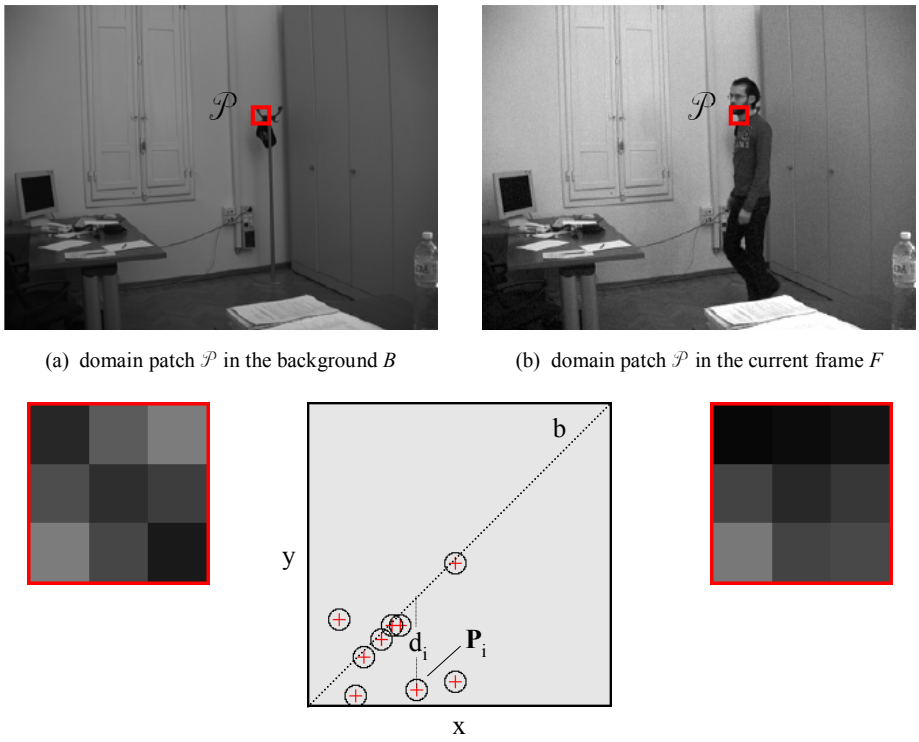
y_1	y_2	y_3
y_4	y_5	y_6
y_7	y_8	y_9

(d) current frame range patch $F(\mathcal{P}) = \boldsymbol{Y}$



(e) 2-dimensional representation of the features vector $\boldsymbol{f} = (\boldsymbol{X}, \boldsymbol{Y})$

Figure 3.1: Domain patch (a,b), range patches (c,d) and 2-dimensional representation of the features vector $\boldsymbol{f} = (\boldsymbol{X}, \boldsymbol{Y})$ (e) for a sample semantically unchanged pixel in two images of the same scene taken at different times.



(a) domain patch $\mathcal{P}$ in the background B

(b) domain patch $\mathcal{P}$ in the current frame F

(c) range patches and 2-dimensional representation of the features vector $\boldsymbol{f} = (\boldsymbol{X}, \boldsymbol{Y})$

Figure 3.2: Domain patch (a,b), range patches and 2-dimensional representation of the features vector $\boldsymbol{f} = (\boldsymbol{X}, \boldsymbol{Y})$ (c) for a sample semantically changed pixel.

ity let us consider a symmetric $\boldsymbol{p}$-centered square patch of odd side s pixels:

$$\mathcal{P}(\boldsymbol{p} = (i, j)) = (\boldsymbol{p}_z = (k, l) : \ i - \Delta \leq k \leq i + \Delta, \ j - \Delta \leq l \leq j + \Delta, \ z = 1, \ldots, N) \quad (3.3)$$

where $\Delta = \frac{s-1}{2}$ and $N = s^2$ is the number of pixels contained in the patch. In figures 3.1(a,b) a 3×3 square patch $\mathcal{P}$ for a sample pixel is pointed out in the background and the current frame images, respectively. Let $B(\mathcal{P})$ and $F(\mathcal{P})$ be the range patches induced by $\mathcal{P}$ on B and F, that is the set of intensities assumed by the images in the pixels of the patch (figures 3.1(c,d), on the left):

$$B, F(\mathcal{P}) = (g_z = B, F(\boldsymbol{p}_z), \ \boldsymbol{p}_z \in \mathcal{P}, \ z = 1, \ldots, N) \quad (3.4)$$

To simplify notations, hereinafter we denote the range patches as follows:

$$B(\mathcal{P}) = \boldsymbol{X} = (x_1, x_2, \ldots, x_N) \qquad F(\mathcal{P}) = \boldsymbol{Y} = (y_1, y_2, \ldots, y_N) \quad (3.5)$$

where the pixels are taken in lexicographical order, as shown in figures 3.1(c,d), on the right.

To detect scene changes occurring in a pixel $\boldsymbol{p}$ of the current frame F, a typical disturbance factors invariant change detector exploits just the information contained in $\boldsymbol{X}$ and $\boldsymbol{Y}$. In other words, information about the temporal dynamics of pixel intensities is neglected. Typically, the binary change mask C is computed by thresholding a particular function measuring the dissimilarity between $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$C(\boldsymbol{p}) \;=\; t\,(d(\boldsymbol{X},\boldsymbol{Y})) \;=\; \begin{cases} 1 & \text{if } \; d(\boldsymbol{X},\boldsymbol{Y}) > T \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

The problem can be formalized into a binary classification framework. The N-dimensional vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be merged into a $2N$-dimensional features vector $\boldsymbol{f}$:

$$\mathbb{Z}^{2N} \supset \mathcal{F} \ni \boldsymbol{f} = (\boldsymbol{X},\boldsymbol{Y}) = (x_1, y_1, \ldots, x_N, y_N) = (\boldsymbol{P}_1, \ldots, \boldsymbol{P}_N) \tag{3.7}$$

where $\mathcal{F}$ is the features space and $\boldsymbol{P}_i = (x_i, y_i)$ denotes a point in a 2-dimensional representation of the features space, as shown in figure 3.1(e). On the basis of $\boldsymbol{f}$ a pixel has to be classified into one of the two following classes:

C: a local scene change has occurred. As an effect, a local image change has occurred as well ($\exists i : x_i \neq y_i$);

U: no local scene change has occurred. As a consequence, no local image change has occurred ($x_i = y_i \;\forall i$) or a change has occurred due to disturbance factors ($\exists i : x_i \neq y_i$).

It is worth pointing out that a scene change always yields an image change but the observation of an image change does not allow to abduct a scene change as the certain cause. In fact, disturbance factors (e.g. scene illumination changes, imaging system noise, dynamic adjustments of the imaging system parameters) can yield changes of pixel intensities even stronger than those produced by scene changes. In figure 3.1(e) a 2-dimensional representation of the features vector $\boldsymbol{f}$ for the sample patch $\mathcal{P}$ is given. A quite strong image change occurs (some of the points $\boldsymbol{P}_i$ lie quite far off the bisector b of the quadrant), but the imaged scene portion is clearly unchanged. In fact, the image change is due to a variation of the scene illumination. Figure 3.2(c) shows the features vector for a different patch of pixels (figures 3.2(a,b)). A remarkable image change occurs as well, but in this case it is due to both the illumination change and the presence of the person.

Therefore, the dissimilarity function of expression 3.6 can not be too "simple" since it must be able to discriminate between intensities variations due to disturbance factors and intensities variations due to scene changes. For example, by using the well known *SSD* (Sum-of-Square-Differences) function:

$$d(\boldsymbol{X},\boldsymbol{Y}) \;=\; SSD(\boldsymbol{X},\boldsymbol{Y}) \;=\; \sum_{i=1}^{N} (x_i - y_i)^2 \tag{3.8}$$

a bad discrimination is obtained. In the 2-dimensional representation of $\boldsymbol{f}$ (figures 3.1(e) and 3.2(c)), the SSD function is equivalent to the sum of the square vertical distances d_i of the points $\boldsymbol{P}_i$ from the bisector b of the quadrant (figure 3.2(c)). It is quite clear by looking at figures 3.1(e) and 3.2(c) that a good discrimination in the features space $\mathcal{F}$ between the features vectors of semantically changed and semantically unchanged patches is not possible by means of the SSD dissimilarity function.

A clear statistical formalization of the problem is worth to be carried out. To the purpose, let us formalize the binary classification problem into a Bayesian framework. The Bayes *MAP* (Maximum A Posteriori) decision rule is the following:

$$\left|\begin{array}{lll} p(U|\boldsymbol{f}) < p(C|\boldsymbol{f}) & \Rightarrow & C \\ \text{otherwise} & \Rightarrow & U \end{array}\right. \Longrightarrow \left|\begin{array}{lll} p(\boldsymbol{f}|U)p(U) < p(\boldsymbol{f}|C)p(C) & \Rightarrow & C \\ \text{otherwise} & \Rightarrow & U \end{array}\right. \tag{3.9}$$

where $p(C\,|\,\boldsymbol{f})$ and $p(U\,|\,\boldsymbol{f})$ are the posterior class probabilities, $p(C)$ and $p(U)$ the prior class probabilities, $p(\boldsymbol{f}\,|\,C)$ and $p(\boldsymbol{f}\,|\,U)$ the features vector likelihoods. Dividing both sides of the expression by $p(\boldsymbol{f}|C)p(U)$ (which is a positive number) and by noticing that $p(U) = 1 - p(C)$, we attain the likelihood ratio formulation:

$$\left|\begin{array}{lll} \dfrac{p(\boldsymbol{f}|U)}{p(\boldsymbol{f}|C)} < \dfrac{p(C)}{1-p(C)} & \Rightarrow & C \\ \text{otherwise} & \Rightarrow & U \end{array}\right. \tag{3.10}$$

The right-hand side of expression 3.10 allows to set a spatially (across different pixels in a given frame) and temporally (across different frames in a given pixel) adaptive threshold for the decision rule, based on the prior probability of the considered pixel (patch) in the current frame to be the image of a semantically changed scene portion, $p(C)$. This prior could be set by exploiting temporal (e.g. prediction of objects position by tracking) and/or spatial (e.g. statistical morphology) information. However, here we are interested in change detectors which exploits just the information contained in the features vector $\boldsymbol{f}$, so we assign equal prior probabilities to the classes ($p(U) = p(C) = 0.5$), thus attaining a *ML*(Maximum Likelihood) classification rule:

$$\left|\begin{array}{lll} \dfrac{p(\boldsymbol{f}|U)}{p(\boldsymbol{f}|C)} < 1 & \Rightarrow & C \\ \text{otherwise} & \Rightarrow & U \end{array}\right. \tag{3.11}$$

Unfortunately, a statistical characterization of the likelihood $p(\boldsymbol{f}\,|\,C)$ is in general unfeasible. In fact, $p(\boldsymbol{f}\,|\,C)$ represents the probability of observing the measured features vector $\boldsymbol{f}$ given that a foreground object is covering the patch. But, unless a priori assumptions are made about the appearance (e.g., colour and orientation) of objects entering the scene, a local scene change does not yield a statistically predictable pattern in the features space. On the contrary, the likelihood $p(\boldsymbol{f}\,|\,U)$ (i.e. the probability of observing the features vector $\boldsymbol{f}$ given that only disturbance factors are acting in the

considered patch) can be characterized once the classes of disturbance factors to be considered are chosen and their effects in the features space are made clear. Therefore, without the above-mentioned prior assumptions about the foreground objects appearance, from a statistical point of view the change detection problem consists in testing the hypothesis that just disturbance factors are acting in the pixels of the considered patch. In particular, a test statistics $\mathcal{S}$ depending on the likelihood $p(\boldsymbol{f} \mid U)$ has to be chosen, so that change detection is carried out by a thresholding of the statistics. Within this statistical framework, the change detection rule of expression 3.6 becomes:

$$C(\boldsymbol{p}) = \begin{cases} 1 & \text{if } \mathcal{S}(p(\boldsymbol{f} \mid U)) > T \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

In the next Section we show how disturbance factors yield a recognizable pattern in the chosen features space.

3.2 Disturbance Factors

Figure 3.3 shows a model of the imaging process for two images of the same scene captured at different times ($t_1 \rightarrow I_1$; $t_2 \rightarrow I_2$) in which just disturbance factors are acting. We give here the definition of two important radiometric quantities:

- radiance: emitted energy (from a source or a surface). In particular, it is the power emitted from a unit area of the surface in a specified direction per unit solid angle (measured in $Wm^{-2}sr^{-1}$);
- irradiance: incident energy (upon a surface). In particular, it is the power falling upon a unit area of a surface (measured in Wm^{-2}).

When dealing with electromagnetic radiations in the visible spectrum (visible light), radiance and irradiance are also called luminance and illuminance, respectively. Scene illuminance $Q_t(\boldsymbol{p})$ is the power of light incident at time t upon the patch of the scene surface $S(\boldsymbol{p})$ imaged by the pixel $\boldsymbol{p}$. It is worth pointing out that we are assuming that no semantic change occurs in the scene (i.e. just disturbance factors act), hence all the quantities related to scene surface physical properties can be assumed to be constant in time, thus dropping the subscript t. Light incident on the surface patch $S(\boldsymbol{p})$ is reflected, so that the incoming scene illuminance $Q_t(\boldsymbol{p})$ is transformed into the outgoing scene radiance $L_t(\boldsymbol{p})$, as shown in figure 3.3(a). This process follows a generic reflectance model r which can be expressed as follows:

$$L_t(\boldsymbol{p}) = r\,(Q_t(\boldsymbol{p}), \boldsymbol{m}(\boldsymbol{p}), \boldsymbol{g}(\boldsymbol{p})) \tag{3.13}$$

where $\boldsymbol{m}(\boldsymbol{p})$ denotes a set of local physical properties of the surface patch (e.g. material, roughness) and $\boldsymbol{g}(\boldsymbol{p})$ a set of geometrical properties related to the reciprocal position of

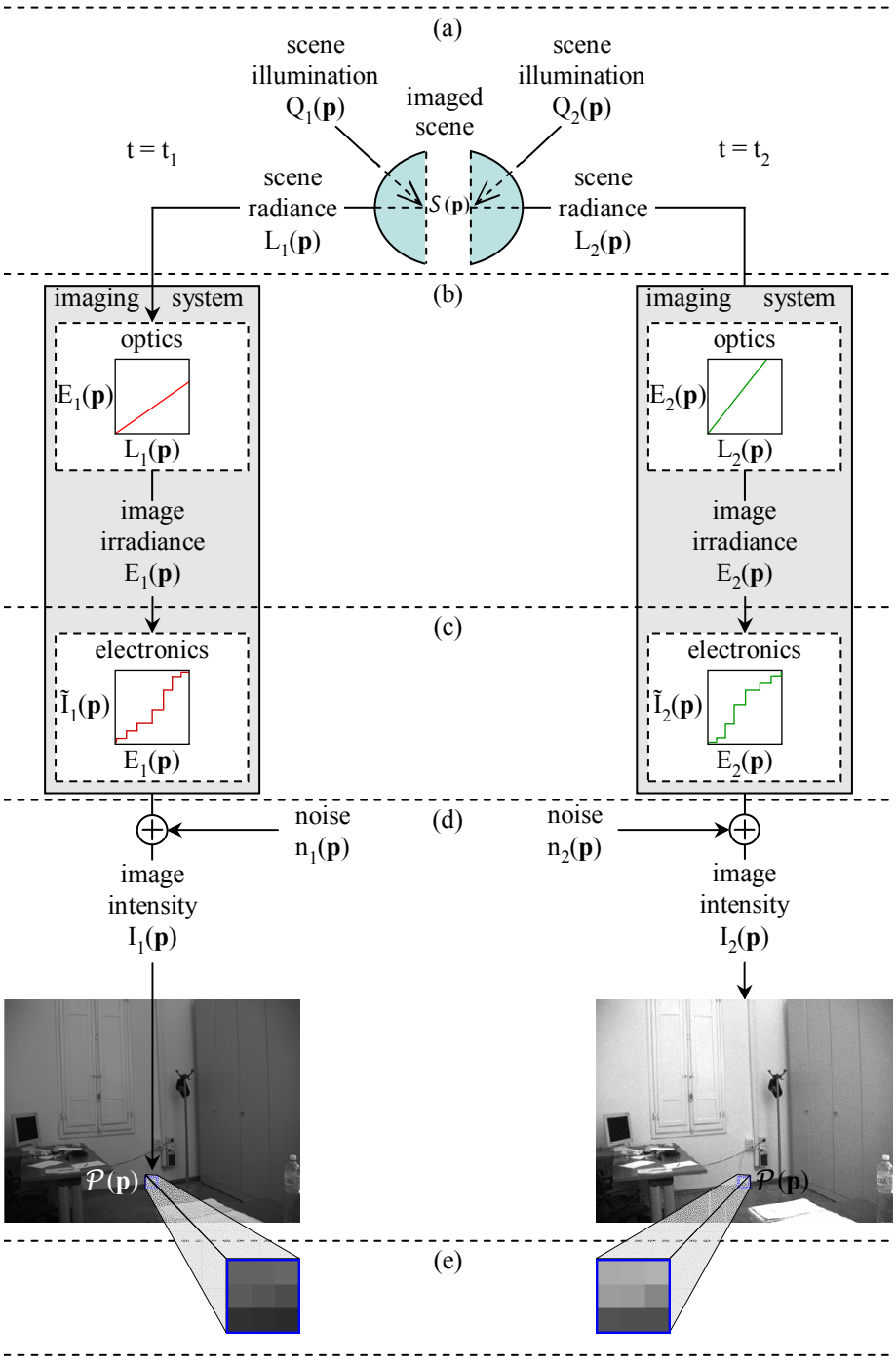


Figure 3.3: Model of the imaging process for two images of the same scene captured at different times.

light source, surface patch and observer (i.e. the considered pixel). It is worth noticing that, for a given pixel (surface patch), a variation of scene radiance can occur only as a consequence of a variation of scene illuminance. In fact, all the other quantities in equation 3.13 are related to surface patch physical and geometrical properties, which are assumed to be constant in time. Various reflectance models have been proposed in the field of computer vision. In general, they can be divided into two main classes: the *physical* models and the *geometrical* models. The physical models use electromagnetic wave theory to analyze the light reflection phenomenon. It is a very general approach, since it can describe reflection from almost every type of material and surface. However, physical models are often inappropriate for use in machine vision as they have functional forms which are very difficult to manipulate. On the other hand, geometrical models are derived by just analyzing the surface and illumination geometry and have simpler functional forms. One of the most commonly used reflectance models is the Phong model ([34]). This model takes into consideration both the diffuse (Lambertian) and the specular reflection. Moreover, ambient light is accounted for. According to this model, we can say that scene irradiance (illuminance) is mapped into scene radiance by a locally order-preserving transformation.

By passing through the imaging system optics, the scene radiance $L_t(\boldsymbol{p})$ is transformed into the image irradiance $E_t(\boldsymbol{p})$, that is the amount of light incident at time t on the pixel $\boldsymbol{p}$ of the capturing system image plane (figure 3.3(b)). Simple geometrical considerations allows to formalize the transformation as follows:

$$E_t(\boldsymbol{p}) = \left(\frac{\cos^4\alpha(\boldsymbol{p})}{f^2}\right) \cdot \left(\frac{\pi d_t^2}{4}\Delta_t\right) \cdot L_t(\boldsymbol{p}) = k(\boldsymbol{p}) \cdot e_t \cdot L_t(\boldsymbol{p}) \tag{3.14}$$

where $\alpha(\boldsymbol{p})$ is the angle between the direction of the principal ray incident on $\boldsymbol{p}$ and the optical axis of the imaging system lens (it just depends on the pixel position), f is the focal length (we assume fixed focal length), d_t is the lens aperture diameter (it may vary along time) and Δ_t is the exposure time, that is the time per frame the image detector is exposed to the incoming light (it may vary along time as well). Hence, the quantity $k(\boldsymbol{p})$ depends on the pixel position but it is constant in time. On the contrary, the quantity e_t, which is called the exposure of the imaging device, is global to all the pixels in the frame, but it may vary along time due to Auto-Exposure (AE) mechanisms of the device.

As shown in figure 3.3(c), image irradiance $E_t(\boldsymbol{p})$ is processed by the imaging system electronics and transformed into the ideal noiseless (we will consider noise as a separate effect) discrete image intensity $\tilde{I}_t(\boldsymbol{p})$ (that is, apart from noise, the pixel grey level we take as input in our algorithms) by a transfer function h_t, commonly called camera transfer function:

$$\tilde{I}_t(\boldsymbol{p}) = h_t\left(E_t(\boldsymbol{p})\right) \tag{3.15}$$

The camera transfer function is a characteristic of each particular imaging system device. In general, it can be assumed spatially invariant. In other words, at a given time t equal values of image irradiance incident on different pixels are mapped by h_t to the same image intensity. On the contrary, the camera transfer function is in general time-variant. In fact, mechanisms of dynamic adjustment of the transfer function parameters (e.g. auto-gain control, *AGC*) are often present in modern cameras. Although the transfer function is in general non linear, at a given time it is always monotonic non-decreasing.

Finally, the measured image intensity $I_t(\boldsymbol{p})$ is affected by noise. In fact, the imaging process is inherently affected by various sources of noise (e.g., shot photon noise, thermal noise, read-out noise and quantization noise). However, to change detection purposes this noise can be modeled as an additive statistical disturb $n_t(\boldsymbol{p})$ affecting the output ideal noiseless image intensity $\tilde{I}_t(\boldsymbol{p})$, as shown in figure 3.3(d):

$$I_t(\boldsymbol{p}) = \tilde{I}_t(\boldsymbol{p}) + n_t(\boldsymbol{p}) \tag{3.16}$$

The overall imaging process can thus be formalized as follows (equations 3.13,3.14, 3.15, 3.16):

$$I_t(\boldsymbol{p}) = \tilde{I}_t(\boldsymbol{p}) + n_t(\boldsymbol{p}) = h_t\Big(k(\boldsymbol{p}) \cdot e_t \cdot r\Big(Q_t(\boldsymbol{p}), \boldsymbol{m}(\boldsymbol{p}), \boldsymbol{g}(\boldsymbol{p})\Big)\Big) + n_t(\boldsymbol{p}) \tag{3.17}$$

In case that no scene change occurs, considering a pixel $\boldsymbol{p}$ at two different times t_1 and t_2 an image (intensity) change occurs if:

$$I_1(\boldsymbol{p}) \neq I_2(\boldsymbol{p}) \tag{3.18}$$

The causes of this change are the disturbance factors, which can be derived from equation 3.17:

$Q_1(\boldsymbol{p}) \neq Q_2(\boldsymbol{p})$: change of the scene illuminance (illumination);

$e_1 \neq e_1$: change of the imaging system exposure;

$h_1 \neq h_1$: change of the imaging system transfer function;

$n_1(\boldsymbol{p}) \neq n_2(\boldsymbol{p})$: statistical fluctuation of intensity due to noise.

Actually, we are not just interested in the effects of disturbance factors on the intensity measured in a pixel. In fact, we aim at investigating if and how the effects of disturbance factors can be expressed by a relation $r_{1\to 2}$ between the intensities of the

pixels in a common domain patch $\mathcal{P}$ of the two images I_1 and I_2, captured "before" and "after" the action of disturbance factors (figure 3.3(e)). In other words, we look for a subset $\mathcal{D}$ of the features space $\mathcal{F}$ which is able to delimitate all the disturbance factors effects. Apart from noise, which will be dealt with in the next section, we can say that disturbance factors yield an order-preserving (i.e. monotonic non decreasing) relation between the ideal noiseless intensities of the pixels in a common domain patch. In fact, the overall imaging process transformation (equation 3.17) from the incoming scene illuminance to the ideal noiseless image intensity is order-preserving, since it is the composition of an order-preserving scene illuminance to scene radiance transformation (equation 3.13, under the Phong reflectance model), a linear (order-preserving) scene radiance to image irradiance transformation (equation 3.14, imaging system optics) and an order-preserving image irradiance to image (noiseless) intensity transformation (equation 3.15, imaging system transfer function). By considering two pixels $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ at times t_1 and t_2, we can write:

$$\begin{cases} \left(\tilde{I}_1(\boldsymbol{p}_1) - \tilde{I}_2(\boldsymbol{p}_1)\right) \cdot (Q_1(\boldsymbol{p}_1) - Q_2(\boldsymbol{p}_1)) \geq 0 \\ \left(\tilde{I}_1(\boldsymbol{p}_2) - \tilde{I}_2(\boldsymbol{p}_2)\right) \cdot (Q_1(\boldsymbol{p}_2) - Q_2(\boldsymbol{p}_2)) \geq 0 \end{cases} \tag{3.19}$$

By assuming a smooth variation of the scene illumination, we can write:

$$(Q_1(\boldsymbol{p}_1) - Q_2(\boldsymbol{p}_1)) \cdot (Q_1(\boldsymbol{p}_2) - Q_2(\boldsymbol{p}_2)) \geq 0 \tag{3.20}$$

Finally, from expressions 3.19 and 3.20:

$$\left(\tilde{I}_1(\boldsymbol{p}_1) - \tilde{I}_2(\boldsymbol{p}_1)\right) \cdot \left(\tilde{I}_1(\boldsymbol{p}_2) - \tilde{I}_2(\boldsymbol{p}_2)\right) \geq 0 \tag{3.21}$$

That is, the relation between intensities of corresponding pixels of two images taken at different times is order-preserving.

3.2.1 Imaging Process Noise

Differently from the other disturbance factors, noise inherently affects the imaging process. Even when all is stationary, the measured image intensities are affected by a statistical error due to noise. For this reason, in every change detection algorithm noise must be accounted for and more or less accurately modeled. As stated before, imaging process noise can be modeled as an additive statistical disturb affecting the output image intensity (equation 3.16). As far as the noise probability distribution is concerned, different choices are possible depending also on how the change detection algorithm will use the distribution. For example, the non-parametric modeling we presented in Chapter 2 is useful for a change detector that exploits the distribution just to extract the threshold for the change mask computation. On the contrary, a change detector that performs a Maximum Likelihood regression to test the hypothesis that a

pixel is changed or unchanged needs a parametric distribution. In practice, for disturbance factors invariant change detectors, the most common assumption is that noise affecting a pixel $\boldsymbol{p}$ at time t is zero-mean gaussian:

$$n_t(\boldsymbol{p}) \sim N\Big(0, \sigma_t^2(\boldsymbol{p})\Big) \tag{3.22}$$

As a consequence, the measured image intensity $I_t(\boldsymbol{p})$ is a gaussian random variable as well, with the same variance of the noise and with mean equal to the ideal noiseless image intensity $\tilde{I}_t(\boldsymbol{p})$:

$$I_t(\boldsymbol{p}) \sim N\Big(\tilde{I}_t(\boldsymbol{p}), \sigma_t^2(\boldsymbol{p})\Big) \tag{3.23}$$

Let us now consider the features vector $\boldsymbol{f}$, which is the input to our change detection problem. We are interested in the noise affecting all the pixel intensities both in the

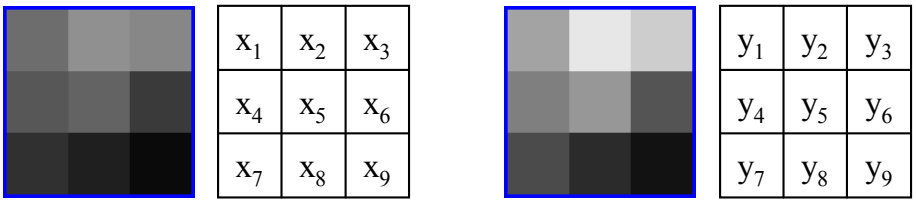


Figure 3.4: Features vector $\boldsymbol{f} = (\boldsymbol{X}, \boldsymbol{Y})$ for a sample pixel.

background B (figure 3.4, on the left) and in the current frame F (figure 3.4, on the right). To this purpose, let $\tilde{\boldsymbol{f}} = (\tilde{x}_1, \ldots, \tilde{x}_N, \tilde{y}_1, \ldots, \tilde{y}_N)$ be the ideal noiseless features vector. According to the assumptions made so far, the distributions are:

$$\begin{aligned} x_i &\sim N\Big(\tilde{x}_i, \sigma_B^2(i)\Big) \\ y_i &\sim N\Big(\tilde{y}_i, \sigma_F^2(i)\Big) \end{aligned} \tag{3.24}$$

where the subscripts B and F indicates a time (i.e. the times at which the background and the current frame images were captured) and the index i denotes a pixel position in the patch. It is reasonable to assume that noise affecting image intensities is white in the spatial as well as in the temporal domain. Namely, noise affecting a pixel at a given time is independent from noise affecting the same pixel at a different time and from noise affecting different pixels at the same time. Hence, the probability distribution of the entire $2N$-dimensional features vector $\boldsymbol{f}$ is multi-variate gaussian:

$$\boldsymbol{f} \sim \frac{1}{(2\pi)^{N/2}\Sigma^{1/2}} \exp\Big(-\frac{1}{2}\Big(\boldsymbol{f} - \tilde{\boldsymbol{f}}\Big)^{\top} \Sigma^{-1} \Big(\boldsymbol{f} - \tilde{\boldsymbol{f}}\Big)\Big) \tag{3.25}$$

with diagonal covariance matrix Σ:

$$\Sigma = \begin{pmatrix} \sigma_x^2(1) & 0 & \cdots & & 0 \\ 0 & \sigma_y^2(1) & & & \\ \vdots & & \ddots & & \vdots \\ & & & \sigma_y^2(N) & 0 \\ 0 & & \cdots & 0 & \sigma_y^2(N) \end{pmatrix} \tag{3.26}$$

In this formulation, the variance of the noise depends both on time and on the pixel position. This is the most complete and effective formulation. in fact, actually the noise varies from pixel to pixel in a given frame and from frame to frame in a given pixel. However, to use this formulation is unfeasible in practice, since an estimation of the noise variance should be performed at each capturing time and for each pixel. The opposite solution, that is the simpler and less effective, consists in assuming a variance which is constant in time and space. In other words, the variance is the same for all the pixels in all the frames. This assumption is quite far from reality. However, it has the advantage that the value of the variance could be estimated once and for all at the beginning of the elaboration. We chose an intermediate solution, based on the assumption that noise affecting a pixel $\boldsymbol{p}$ at time t just depends on the pixel ideal noiseless image intensity $\tilde{I}_t(\boldsymbol{p})$:

$$\sigma_t^2(\boldsymbol{p}) = \sigma^2\Big(\tilde{I}_t(\boldsymbol{p})\Big) \tag{3.27}$$

This assumption is the same we used in Chapter 2 to infer a non-parametric model of the noise. After the estimation of a probability distribution for noise affecting each possible ideal noiseless intensity, a couple of percentiles was extracted from each distribution. Here, the same algorithm can be used to estimate the parametric model of equation 3.27. In fact, we just have to compute the variance of each distribution instead of extracting the percentiles. It is worth pointing out that this model yields a variance which varies with time and space (through the variation of pixel intensities), but the estimation is computed once and for all by processing a short bootstrap sequence.

Based on equations 3.25, 3.26, 3.27, the probability distribution of the features vector $\boldsymbol{f}$ can be written as:

$$\boldsymbol{f} \sim \frac{1}{(2\pi)^{N/2}\ \Pi_{i=1}^{N}\Big(\sigma(\tilde{x}_i)\sigma(\tilde{y}_i)\Big)}\ e^{-\frac{1}{2}\sum_{i=1}^{N}\frac{(x_i-\tilde{x}_i)^2}{\sigma^2(\tilde{x}_i)}+\frac{(y_i-\tilde{y}_i)^2}{\sigma^2(\tilde{y}_i)}} \tag{3.28}$$

If the background image is just a frame captured when the scene was free of foreground objects, the above equation holds. In fact, it relies on the assumption that pixels in the current frame and pixels in the background image are affected by the same ”amount” of noise, given by the noise model of equation 3.27. In particular, the algorithm used to

infer the model yields a model for the noise affecting a raw frame. However, sometimes the background image is estimated through a statistical elaboration of a sequence of frames (e.g., by a temporal averaging of pixel intensities). In general, the higher is the number of frames used to infer the background image, the lower is the ratio between the noise variances of the background and the current frame, respectively. If the sequence of frames used to generate the background is long (tens or even hundreds of frames), the noise variance of the background can be reasonably set to zero, thus attaining the following simplified features vector distribution:

$$\boldsymbol{f} \sim \frac{1}{(2\pi)^{N/2} \prod_{i=1}^{N} \left(\sigma(\tilde{y}_i)\right)} \; e^{-\frac{1}{2}\sum_{i=1}^{N} \frac{(y_i - \tilde{y}_i)^2}{\sigma^2(\tilde{y}_i)}} \tag{3.29}$$

In other words, the imaging process noise affects just the measured intensities of the current frame (i.e. the y_is), while the measured (i.e. estimated from a set of measured) intensities of the background (the x_is) are assumed to be deterministic and equal to the ideal noiseless intensities:

$$x_i = \tilde{x}_i \qquad \forall\, i = 1, \ldots, N \tag{3.30}$$

3.3 The Proposed Algorithm

To apply the change detection rule of expression 3.12, at each time and for each pixel we have to compute the likelihood $p(\boldsymbol{f} \mid U)$, that is the probability of observing the measured (noisy) features vector $\boldsymbol{f} = (x_1, \ldots, x_N, y_1, \ldots, y_N)$ given that no local semantic change is occurring, that is given that just disturbance factors are acting. But we know that disturbance factors yield order-preserving relations between the ideal noiseless intensities of pixels in a common domain patch. In other words, disturbance factors yield an ideal (noiseless) features vector $\tilde{\boldsymbol{f}} = (\tilde{x}_1, \ldots, \tilde{x}_N, \tilde{y}_1, \ldots, \tilde{y}_N)$ belonging to the subspace $\mathcal{D}$ of the features space $\mathcal{F}$ containing all the features vectors representing order-preserving relations. Hence, the likelihood $p(\boldsymbol{f} \mid U)$ can be regarded as the probability of observing the noisy feature vector $\boldsymbol{f}$ given that the noiseless feature vector $\tilde{\boldsymbol{f}}$ belongs to the subspace $\mathcal{D}$:

$$p(\boldsymbol{f} \mid U) = p(\boldsymbol{f} \mid \tilde{\boldsymbol{f}} \in \mathcal{D}) \tag{3.31}$$

Practically speaking, the likelihood $p(\boldsymbol{f} \mid U)$ can be seen as a statistical distance between the measured features vector $\boldsymbol{f}$ and the subspace $\mathcal{D}$ characterizing the disturbance factors effects. To compute the projection of $\boldsymbol{f}$ onto $\mathcal{D}$, $\tilde{\boldsymbol{f}}_{ML}$ we can perform a (non-parametric) Maximum-Likelihood isotonic regression ([1]). The inference prob-

lem can be formalized as follows:

$$\tilde{\boldsymbol{f}}_{ML} = \underset{\tilde{\boldsymbol{f}} \in \mathcal{D}}{argmax}\ p(\boldsymbol{f} \,|\, \tilde{\boldsymbol{f}}) \tag{3.32}$$

Once the projection $\tilde{\boldsymbol{f}}_{ML}$ has been inferred, the likelihood $p(\boldsymbol{f} \,|\, U)$ can be obtained by computing the statistical distance between $\boldsymbol{f}$ and $\tilde{\boldsymbol{f}}_{ML}$:

$$p(\boldsymbol{f} \,|\, U) = p(\boldsymbol{f} \,|\, \tilde{\boldsymbol{f}}_{ML}) \tag{3.33}$$

By making $p(\boldsymbol{f} \,|\, \tilde{\boldsymbol{f}})$ as well as $\mathcal{D}$ explicit and by transforming likelihood maximization into log-likelihood minimization, the inference problem of equation 3.32 can be written as follows:

$$\begin{aligned} \tilde{\boldsymbol{f}}_{ML} = \underset{\tilde{\boldsymbol{f}}}{argmin} & \sum_{i=1}^{N} \frac{(x_i - \tilde{x}_i)^2}{\sigma^2(\tilde{x}_i)} + \frac{(y_i - \tilde{y}_i)^2}{\sigma^2(\tilde{y}_i)} \qquad i, j \in [1, N] \\ & (\tilde{x}_i - \tilde{x}_j)(\tilde{y}_i - \tilde{y}_j) \geq 0 \end{aligned} \tag{3.34}$$

or as follows:

$$\begin{aligned} \tilde{\boldsymbol{f}}_{ML} = \underset{\tilde{\boldsymbol{f}}}{argmin} & \sum_{i=1}^{N} \frac{(y_i - \tilde{y}_i)^2}{\sigma^2(\tilde{y}_i)} \qquad i, j \in [1, N] \\ & (x_i - x_j)(\tilde{y}_i - \tilde{y}_j) \geq 0 \end{aligned} \tag{3.35}$$

depending on the procedure used to generate the background model. If the background is just an image of the scene free of foreground objects, the noise model of equation 3.28 can be used, thus attaining the inference problem of equation 3.34. If the background model is extracted by a statistical estimation procedure, the problem of equation 3.35 is attained by exploiting the noise model of equation 3.29. Both 3.34 and 3.35 are convex programming problems, since the cost function is quadratic and the constraints are convex. In particular, 3.34 is characterized by $2N$ unknowns (i.e. the entire ideal noiseless features vector $\tilde{\boldsymbol{f}} = (\tilde{x}_1, \ldots, \tilde{x}_N, \tilde{y}_1, \ldots, \tilde{y}_N)$) and $\binom{N}{2}$ constraints. On the other hand, 3.35 is characterized by N unknowns (i.e. just the half of the ideal noiseless features vector $\tilde{\boldsymbol{f}}$ corresponding to the pixel intensities in the current frame $\tilde{\boldsymbol{Y}} = (\tilde{y}_1, \ldots, \tilde{y}_N)$) and $(N - 1)$ constraints.

Hereinafter, we assume that the background model is generated by an estimation procedure and take into consideration the problem of equation 3.35. It is a classical isotonic non-parametric regression problem, that can be solved by an $O(N)$ iterative algorithm, called *Pool Adjacent Violators Algorithm* (PAVA) ([1]). To illustrate the algorithm, let us consider a sample 8-dimensional measured (noisy) features vector $\boldsymbol{f} = (\boldsymbol{X}, \boldsymbol{Y}) = (x_1, \ldots, x_4, y_1, \ldots, y_4)$ corresponding to a 2×2 domain patch, as shown in figure 3.5. We denote as $\boldsymbol{f}^o = (\boldsymbol{X}^o, \boldsymbol{Y}^o) = ({x_1}^o, \ldots, {x_4}^o, {y_1}^o, \ldots, {y_4}^o)$ the features vector attained by ordering the $\boldsymbol{X}$ vector components and shuffling the $\boldsymbol{Y}$ components accordingly, as shown in figure 3.5. The problem of equation 3.35 can be written as

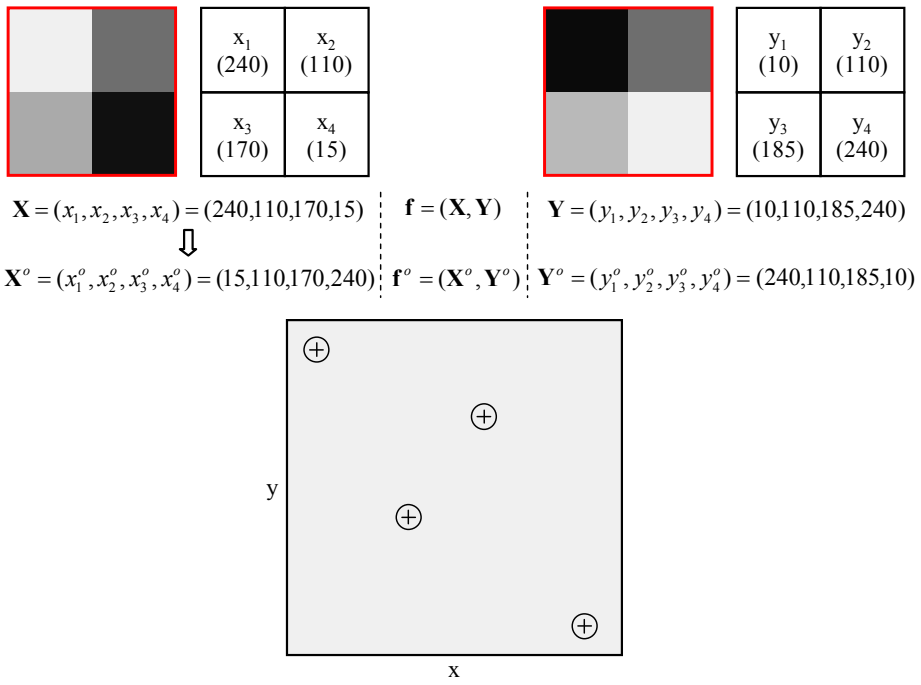


Figure 3.5: Range patches and features vector 2-dimensional representation for a sample 2×2 domain patch.

follows:

$$\tilde{\boldsymbol{f}}_{ML} = \underset{\tilde{f}}{argmin} \sum_{i=1}^{N} \frac{\left(y_i^o - \tilde{y}_i\right)^2}{\sigma^2(\tilde{y}_i)} \quad i \in [1,4] \qquad \tilde{y}_i \leq \tilde{y}_{i+1} \quad i \in [1,3] \tag{3.36}$$

that is:

$$\tilde{\boldsymbol{f}}_{ML} = \underset{\tilde{f}}{argmin} \left[\frac{(240-\tilde{y}_1)^2}{\sigma^2(240)} + \frac{(110-\tilde{y}_2)^2}{\sigma^2(110)} + \frac{(185-\tilde{y}_3)^2}{\sigma^2(185)} + \frac{(10-\tilde{y}_4)^2}{\sigma^2(10)} \right] \qquad (\tilde{y}_1 \leq \tilde{y}_2) \ \wedge \ (\tilde{y}_2 \leq \tilde{y}_3) \ \wedge \ (\tilde{y}_3 \leq \tilde{y}_4) \tag{3.37}$$

In practice, the points in the features vector 2-dimensional representation of figure 3.5 has to be ”moved” toward the ”nearest” isotonic configuration (i.e. a configuration satisfying the constraints). In particular, since the measured pixel intensities in the background are assumed to be deterministic the points can be moved just along the y axis. Figure 3.6 shows the processing steps of the PAVA algorithm applied to the sample problem of equation 3.37. Once computed the projection $\tilde{\boldsymbol{f}}_{ML}$, we use the cost function in equation 3.37 as the statistics to be thresholded in the change detection rule of expression 3.12. It is worth pointing out that this cost function is nothing else than the Mahalanobis distance between the measured features vector $\boldsymbol{f}$ and the subspace $\mathcal{D}$ characterizing the disturbance factors effects.

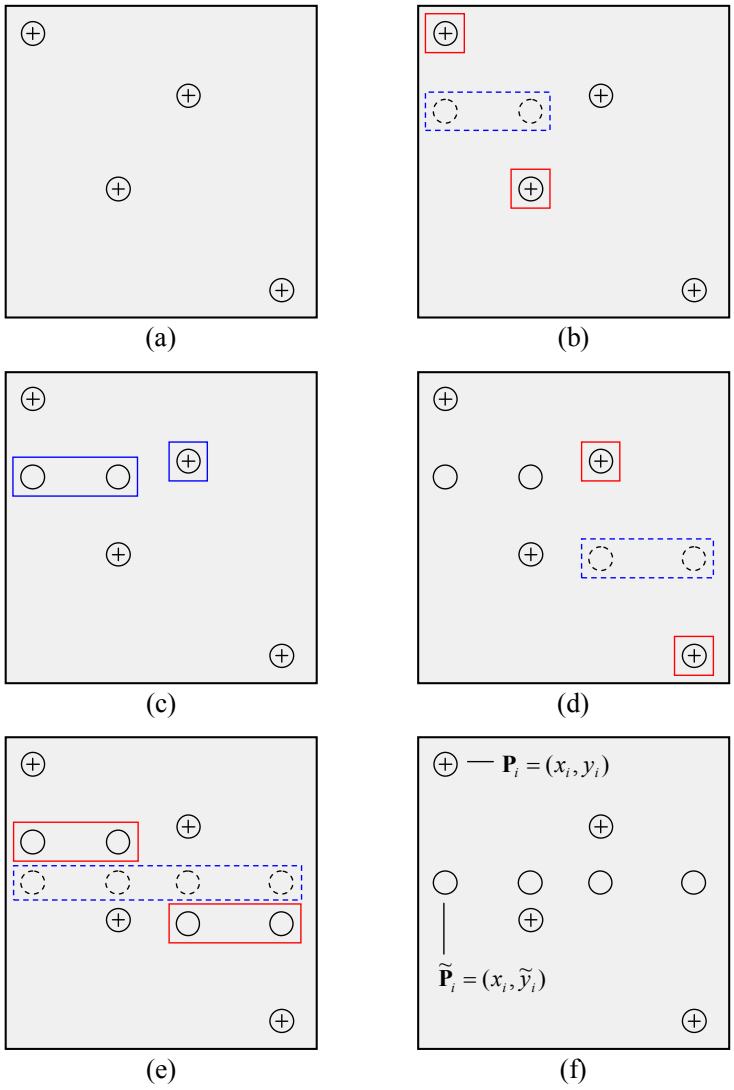


Figure 3.6: Steps of the PAVA algorithm for a sample features vector.

3.4 Experimental Results

Experiments have been carried out by comparing the detection results provided by the proposed approach with the results attained by three different state-of-the art disturbance factors invariant algorithms ([41], [40], [32]). For simplicity, hereinafter we denote by *C*, *B* and *O* the algorithms proposed in [41], [40] and [32], respectively. Moreover, we denote our approach by *P*. In figures 3.7-3.18 the detection results are shown. Each figure corresponds to a different sample frame. The first six frames belongs to an indoor video sequence, in which real and sudden scene illumination changes

occur. The other frames belongs to an outdoor sequence, in which synthetic changes have been created by applying non-linear intensity mapping functions. In each figure, the left column shows the results attained by using a 3×3 image patch as the support for the algorithms decision rule. The results attained by a 7×7 support are depicted in the right column. In each of the change masks brighter points represents higher probabilities of change. In the second row of each figure we depict the comparagram $J_{\hat{B}} \rightarrow F$. The comparagram is nothing else than a generalization of the features vector 2-dimensional representation used so far. Namely, the comparagram is the 2-dimensional joint histogram of the intensities of corresponding pixels in the two considered images (here, the background model $\hat{B}$ and the current frame$F$). The comparagram provides an indication of the intensity mapping functions between $\hat{B}$ and F.

By looking at the change masks, it is straightforward pointing out how the proposed approach outperforms all the other algorithms in the outdoor sequence, where synthetic non-linear intensity mapping functions have been applied. In the indoor sequence, just O provides comparable results.

3.5 Conclusions

In this chapter we have presented a disturbance factors invariant single-view change detection algorithm aimed at filtering-out most of the possible disturbance factors effects. Apart from the imaging system noise which can be modeled as an additive gaussian disturb, the global effect of disturbance factors on the measured intensities in a small patch of pixels can be reasonably assumed to be a monotonic non-decreasing intensity mapping function. Hence, a maximum-likelihood isotonic regression procedure can be used to recognize and discriminate false appearance changes caused by disturbance factors. We have carried out experiments by comparing the detection results provided by the proposed algorithm with the ones attained by three state-of-the-art disturbance factors invariant approaches. Apart from the quite rare case in which disturbance factors yield a linear local intensity mapping function, the proposed algorithm gives the best results. As well as all the disturbance factors invariant change detection algorithms, the proposed approach suffers of an inherent problem of missed detections in correspondence of poorly structured patch of pixels. In the next chapter we present a coarse-to-fine change detection approach, which solves this problem by using the algorithm presented in this chapter at a reduced resolution level and a temporally adaptive algorithm at the full resolution level.

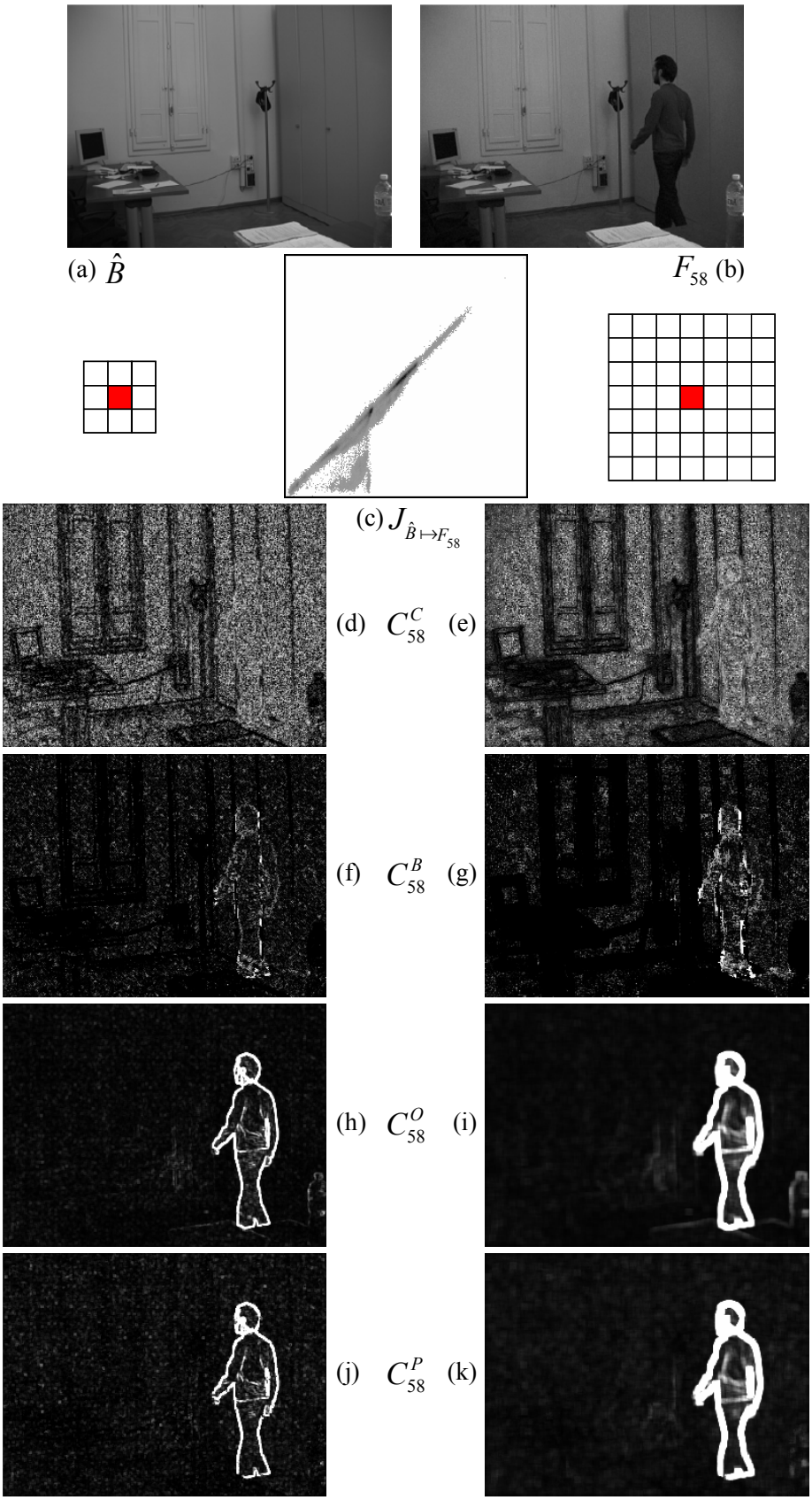


Figure 3.7: Comparative detection results.

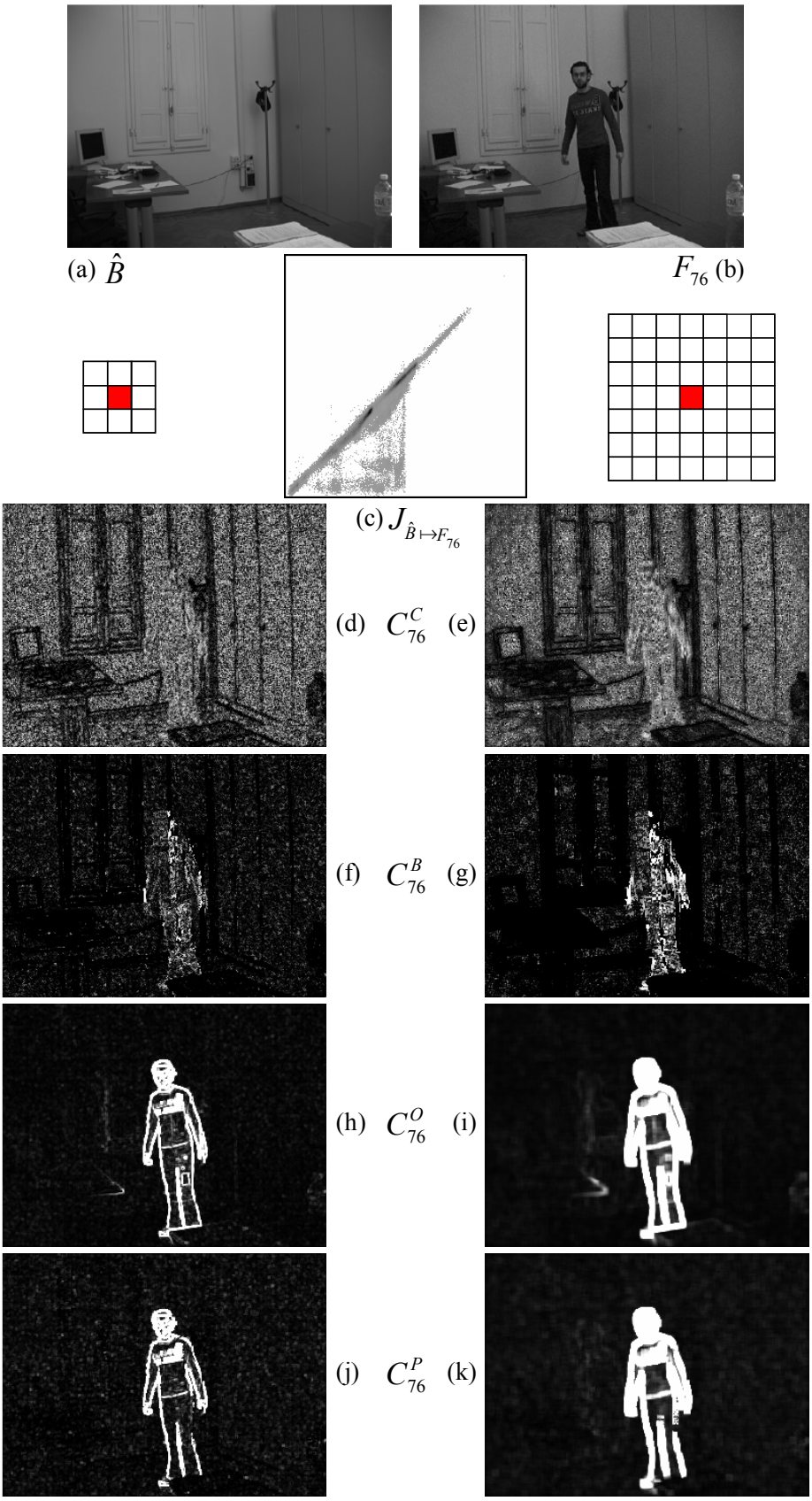


Figure 3.8: Comparative detection results.

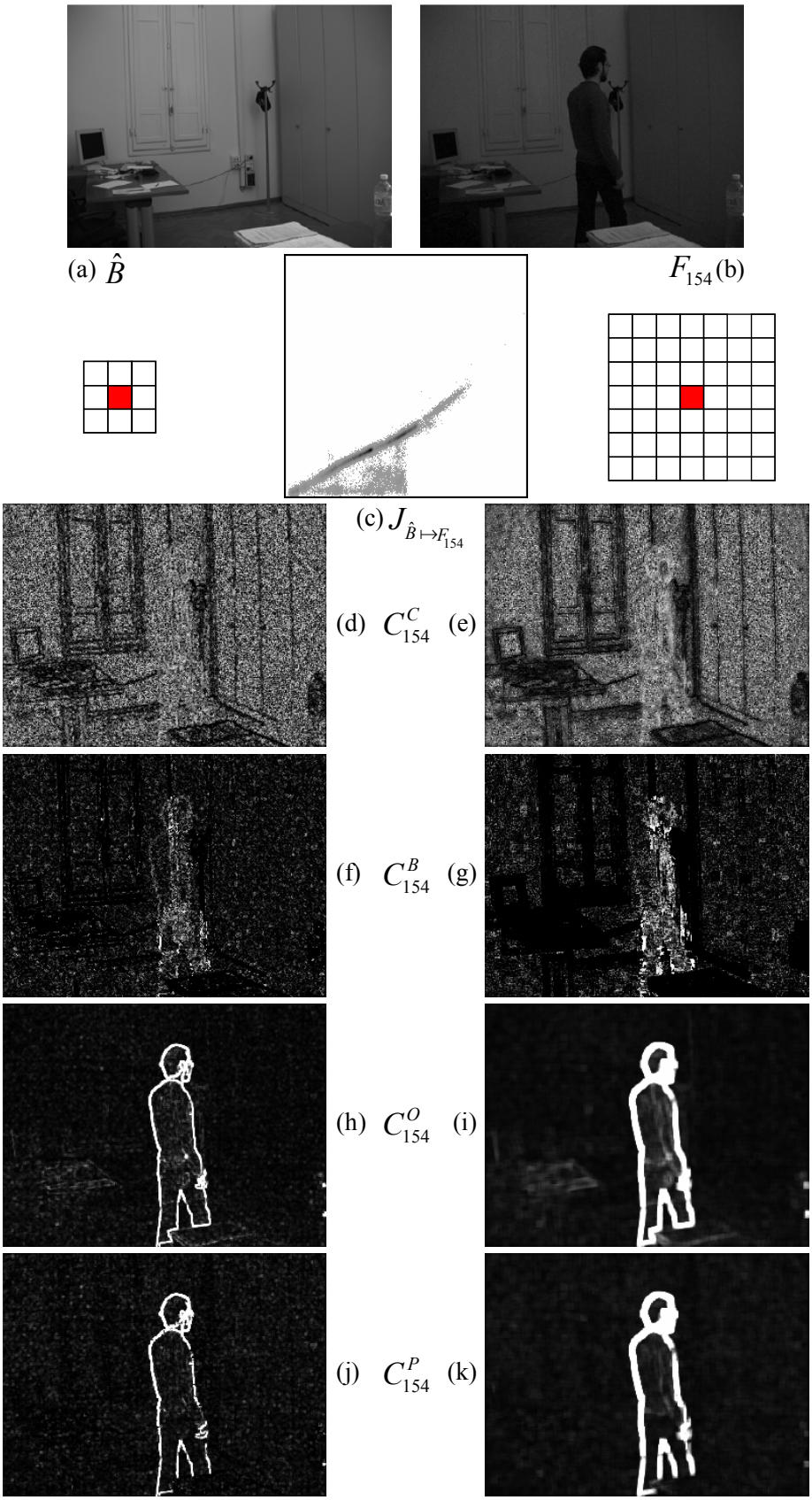


Figure 3.9: Comparative detection results.

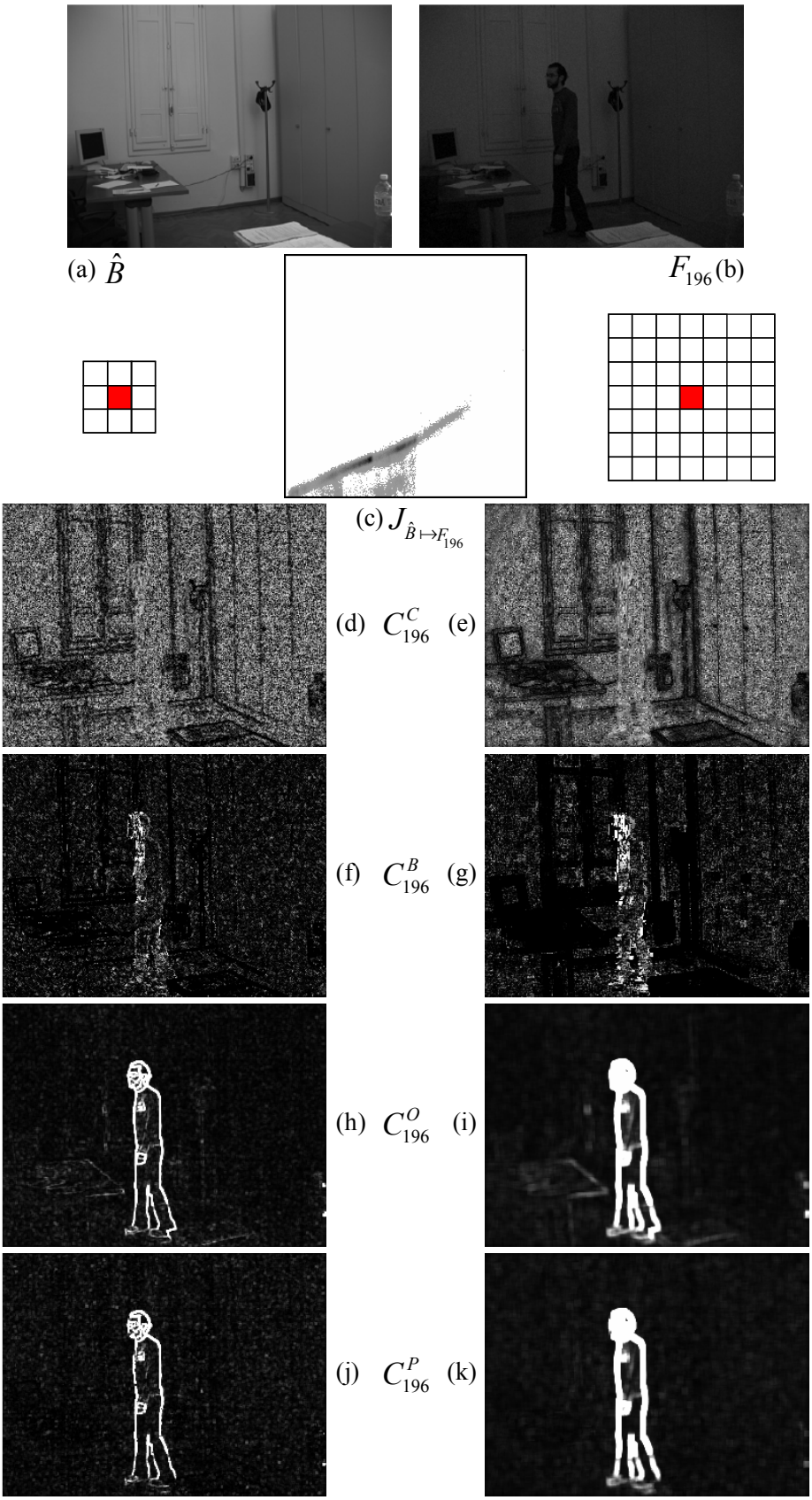


Figure 3.10: Comparative detection results.

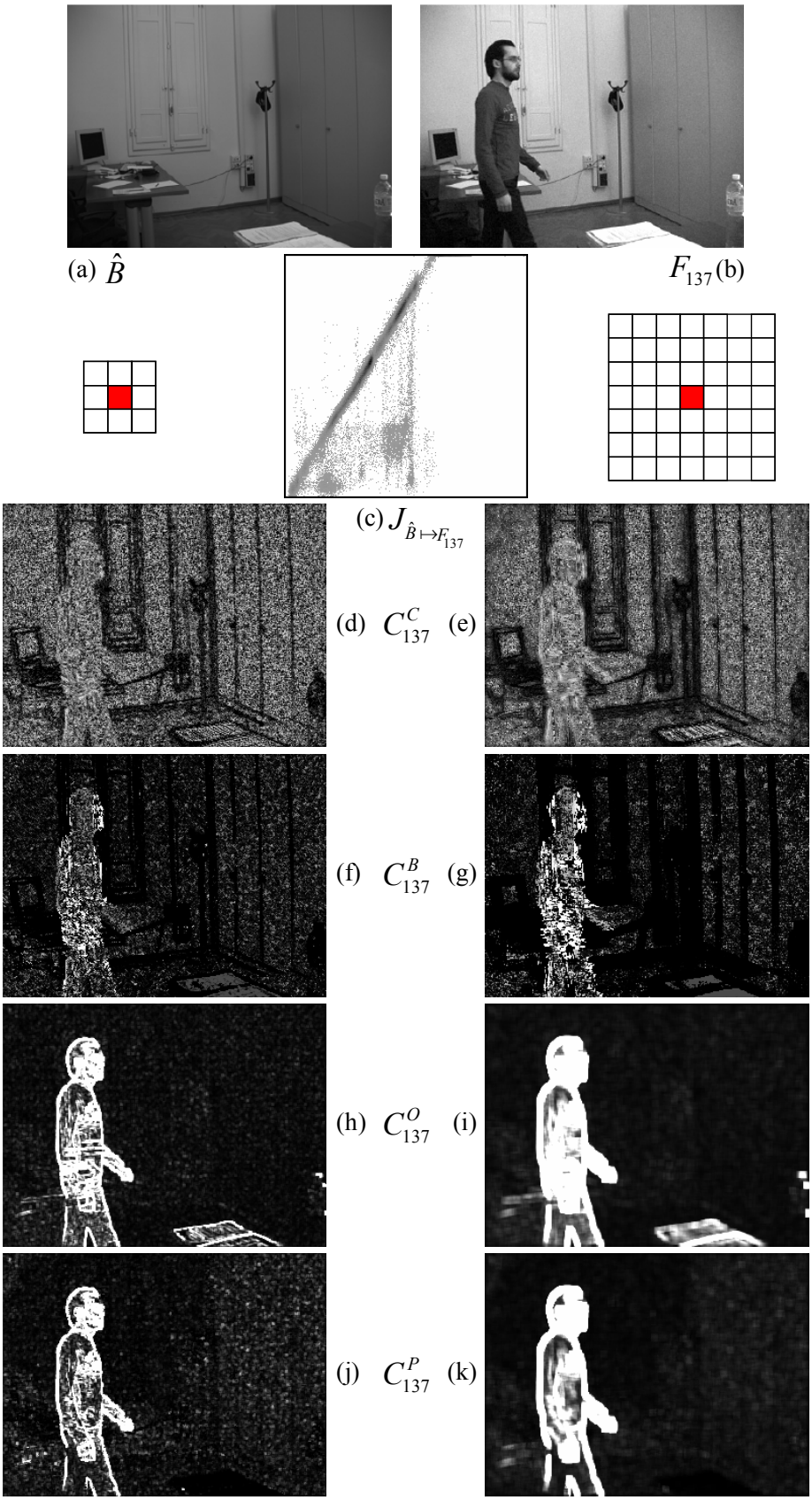


Figure 3.11: Comparative detection results.

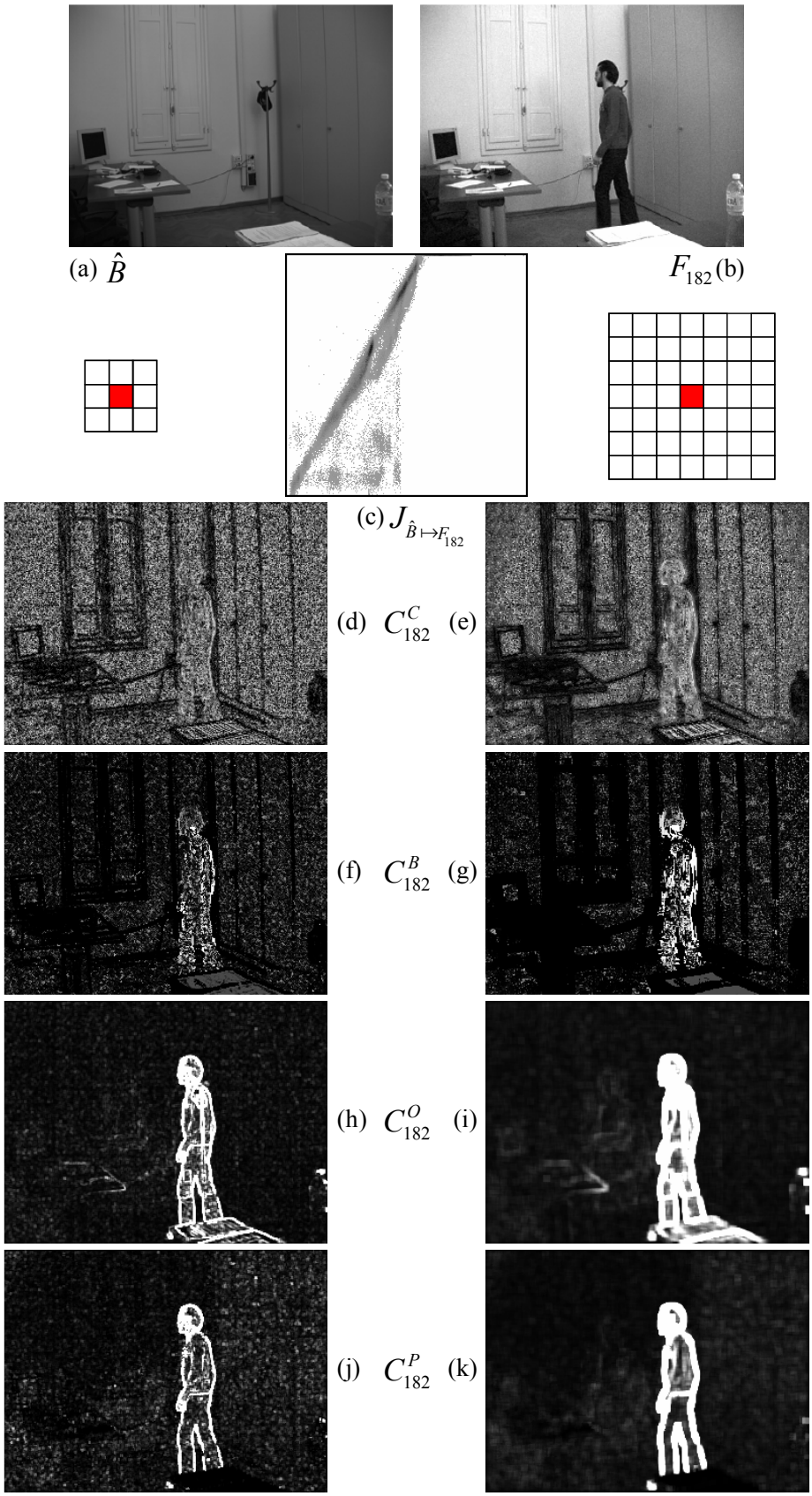


Figure 3.12: Comparative detection results.

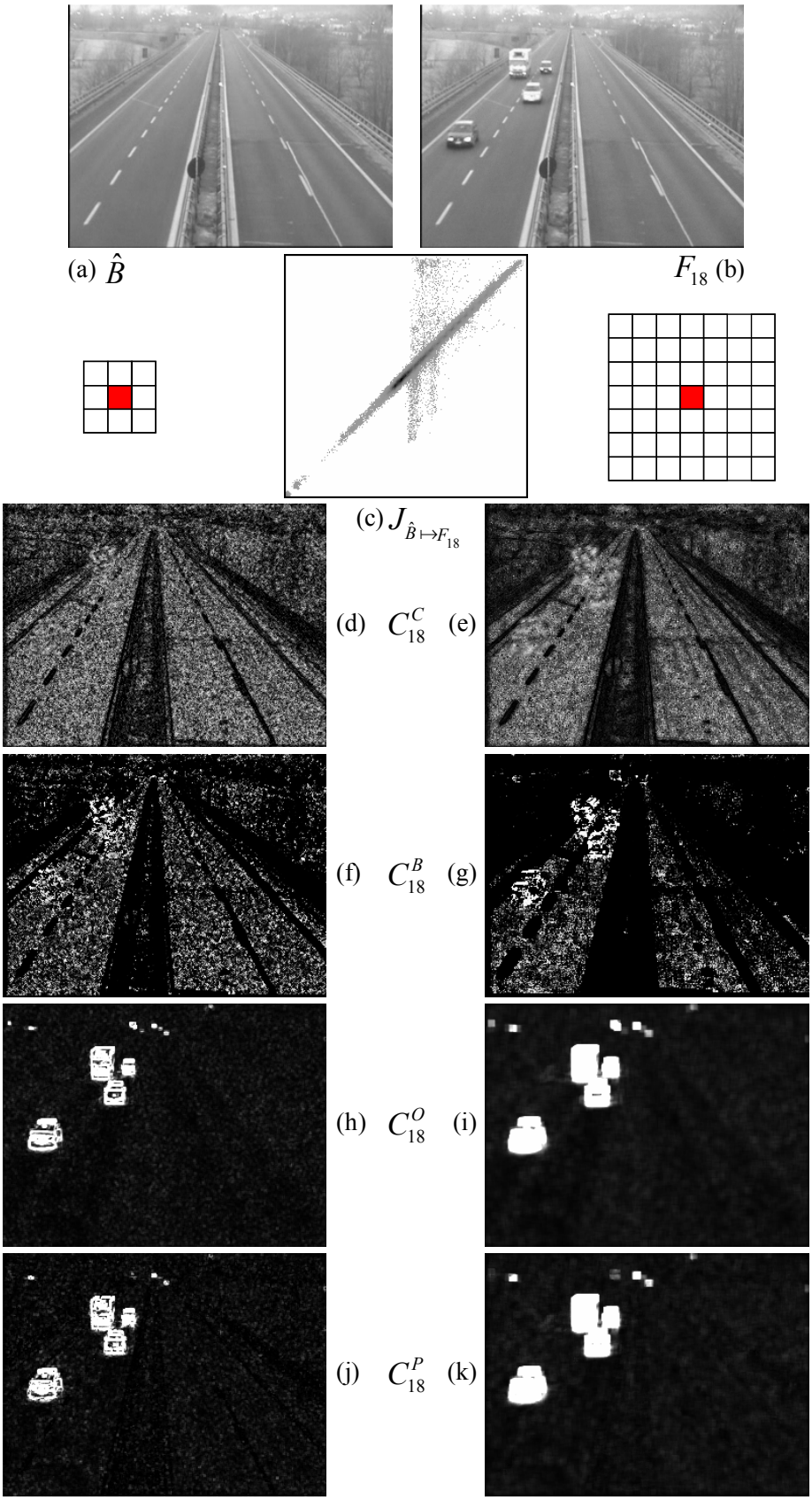


Figure 3.13: Comparative detection results.

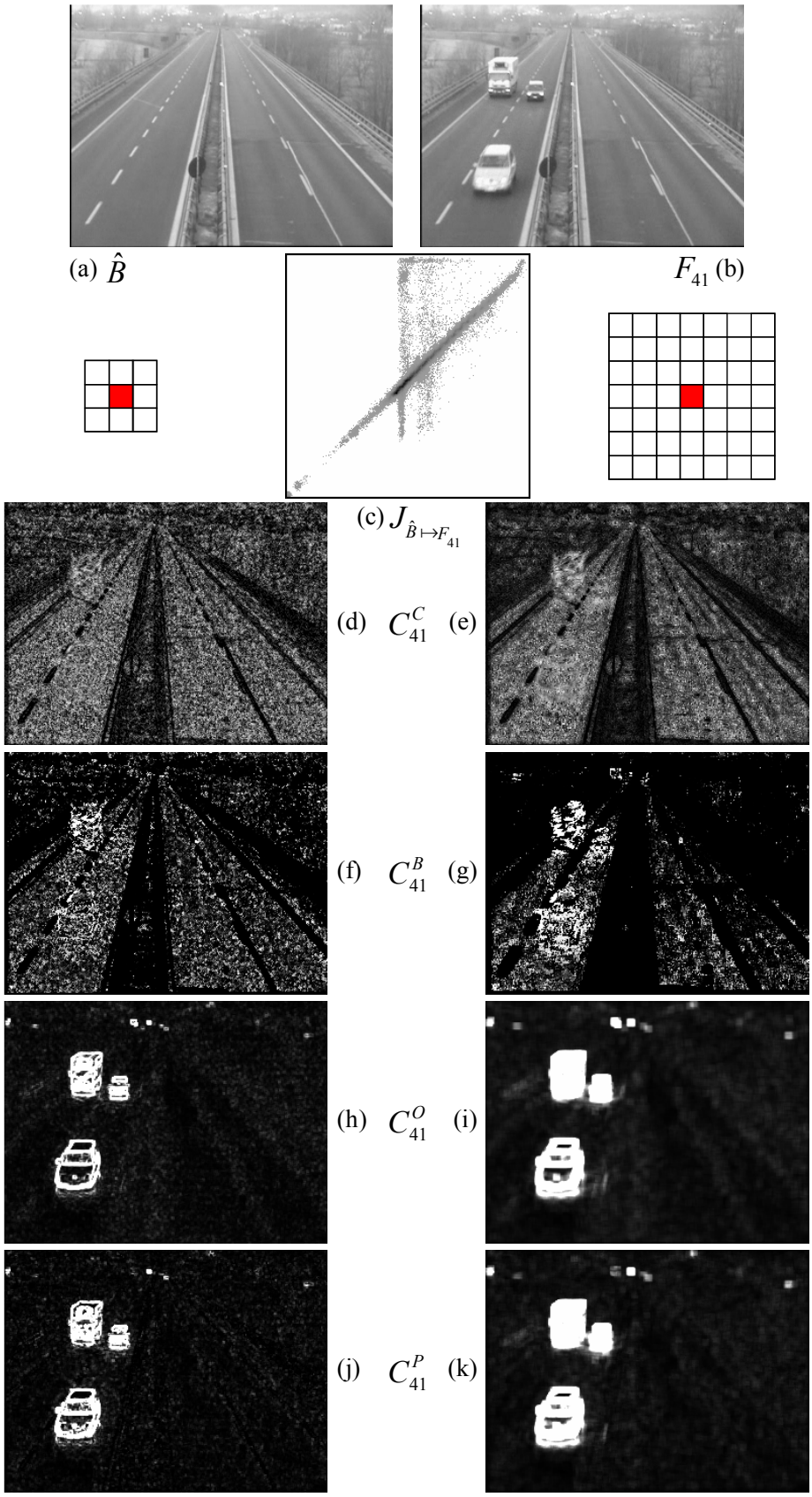


Figure 3.14: Comparative detection results.

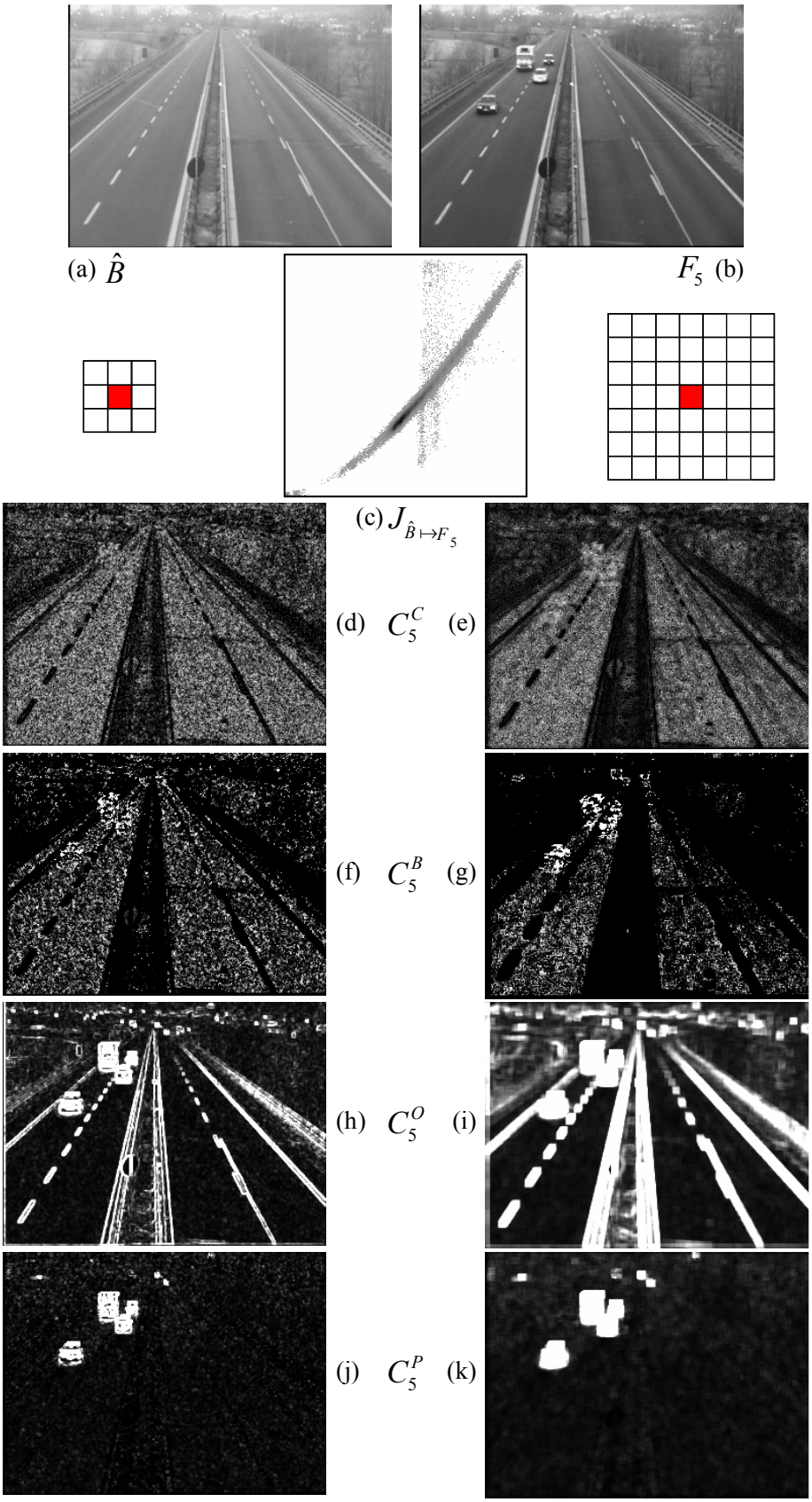


Figure 3.15: Comparative detection results.

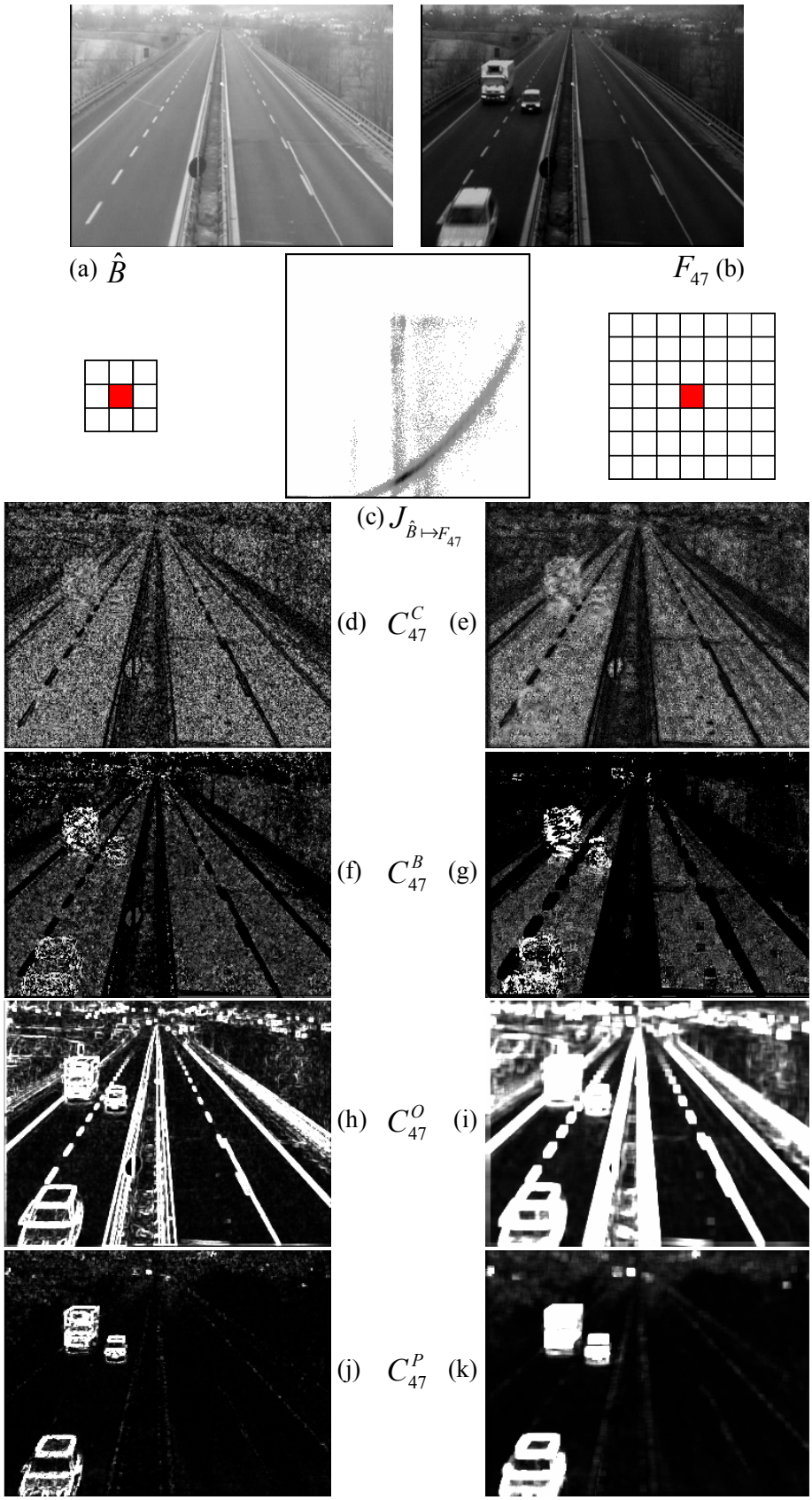


Figure 3.16: Comparative detection results.

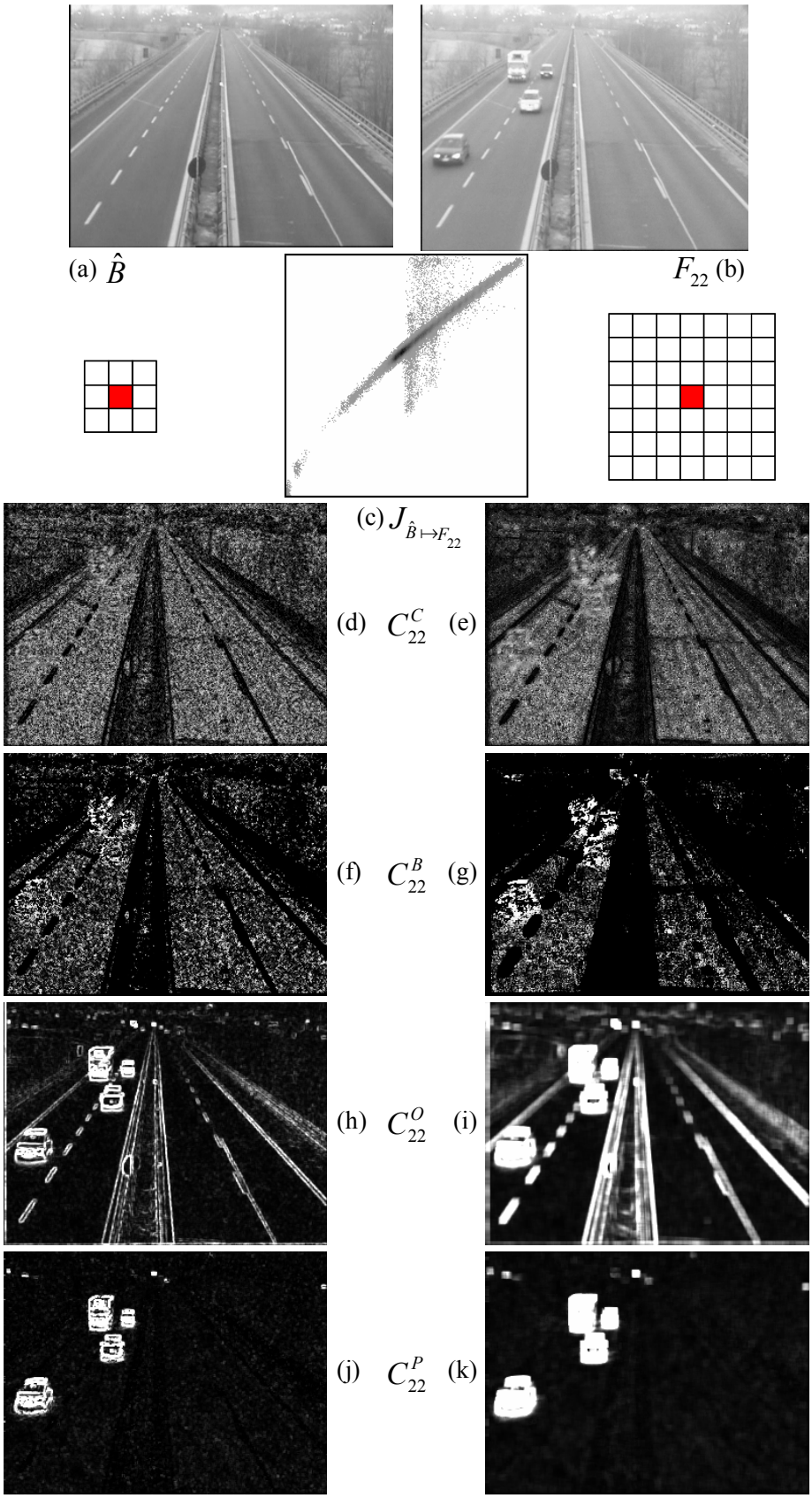


Figure 3.17: Comparative detection results.

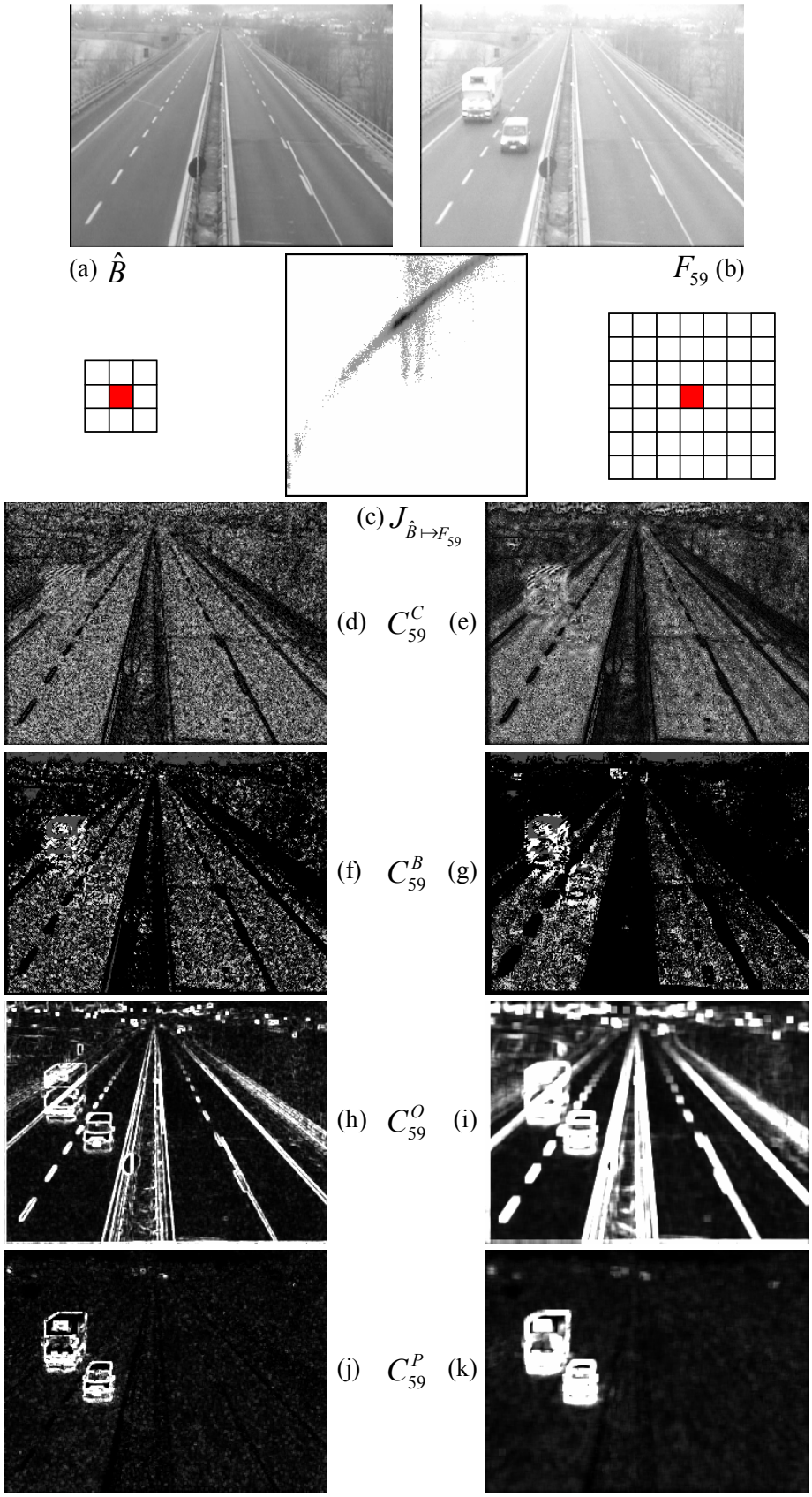


Figure 3.18: Comparative detection results.

Chapter 4

Coarse-to-Fine Approach

In chapters 2 and 3 we have presented two very different single-view change detection approaches. Each of the approaches belongs to one of the two different classes of change detection algorithms pointed out in chapter 1. In particular, a temporally adaptive approach is presented in chapter 2 and a disturbance factors invariant algorithm is proposed in chapter 3. Though the presented approaches provide good detection results, they suffer of different problems, which unfortunately are inherent to every devisable algorithm in their class. In particular, temporally adaptive approaches can not deal effectively with sudden appearance changes of the scene background surface (e.g. sudden scene illumination changes, dynamic adjustments of the imaging system parameters). On the other hand, disturbance factors invariant approaches are very robust to sudden appearance changes, but they can detect semantic changes just in correspondence of quite structured patch of pixels.

In this chapter we show how the two approaches can be used together in a coarse-to-fine framework to attain better results. The basic idea consists in assigning to an efficient preliminary coarse-level (reduced resolution level) the task to filter-out effectively most of the possible false appearance changes, thus providing the subsequent fine-level (full resolution level) with a coarse-grain reliable and tight super-mask of the semantically changed pixels. In particular, we apply the disturbance factors invariant change detection algorithm proposed in chapter 3 at a reduced resolution. In other words, reduced resolution versions of the background model as well as of the currently processed frame are computed, so that the background subtraction algorithm of chapter 3 can be applied to these ”smaller” images. The attained coarse-grain super-mask can be used at the fine-level for a threefold purpose. Firstly, it can act as a reliable work-area for the fine-level detection, that has just to switch-off the pixels it detects as unchanged. Secondly, the complement of these super-mask (that is a tight sub-mask of the semantically unchanged pixels) can represent a just as reliable work-area for

a robust selective background updating procedure at the fine-level (the fine-level is a temporally adaptive background subtraction algorithm, hence a continuous updating of the background appearance model has to be carried out). Finally, the complement can be used also to infer information on global false appearance changes possibly occurring in the scene, such as those due to global scene illumination changes and to dynamic adjustments of the imaging system parameters, so that a tonal registration of the fine-level current background can be carried out. In this way, the temporally adaptive change detection algorithm used at fine-level can face sudden appearance changes as well.

The chapter is organized as follows. In section 4.1 the proposed coarse-to-fine approach is presented. Experimental results are discussed in section 4.2 and conclusions are drawn in section 4.3.

4.1 The Proposed Approach

From the full resolution background B and the full resolution current frame F (figures 4.1(a,b)), the ρ-reduced resolution versions ${}_{\rho}B$ and ${}_{\rho}F$ (figures 4.1(c,d)) are extracted by a *regular* resolution reduction. Namely, both the full resolution images are divided into equal non-overlapping square blocks of side length ρ pixels (in figure 4.1 a value $\rho = 8$ is used). To each block is assigned a unique grey level by computing the median of the intensities of all the pixels contained in the block. Resolution reduction by median intensity commutes with images transformations by order-preserving (i.e. monotonic non-decreasing) intensity mapping functions. Hence, disturbance factors yield local order-preserving intensity transformations at reduced resolution as well, so that the assumption standing at the basis of the algorithm proposed in chapter 3 is still valid. This is not rue, for example, by computing a resolution reduction by mean intensity. Actually, the reduced resolution background is computed once and for all at the beginning of the elaboration. In fact, differently from the temporal adaptive approaches, disturbance factors invariant algorithms need not to update the appearance background model. In particular, the reduced resolution background model is extracted from the full resolution background model generated by the fine-level background initialization procedure. On the contrary, the reduced resolution current frame has to be extracted at each processing step from the new incoming frame.

Hence, at each processing time we apply the algorithm presented in chapter 3 by using the reduced resolution background and current frame, thus attaining a reduced resolution change probability map (figure 4.1(e)). The map gives for each patch of pixels in the reduced resolution domain, that is for each patch of blocks of pixels in the full resolution domain, the statistical distance between the measured features vector

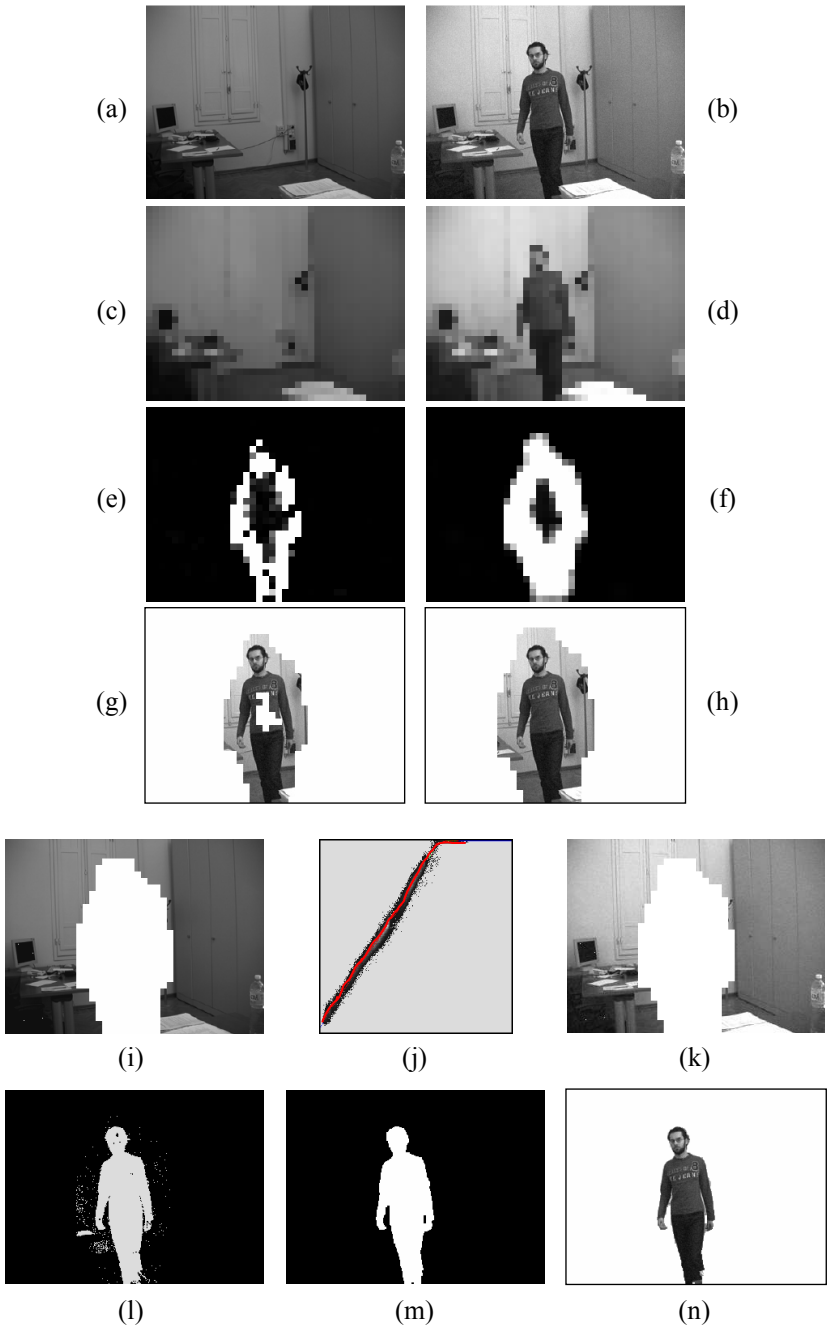


Figure 4.1: Main processing steps of the proposed coarse-to-fine approach.

and the sub-space of all the isotonic features vectors. In other words, the map gives for each block a measure of the probability to be the image of a semantically changed scene background surface patch. The map is convolved by a 3×3 gaussian kernel, thus

attaining a ”smoother” change probability map (figure 4.1(f)). The smooth map is then thresholded (figure 4.1(g)), filled and dilated by a 3×3 binary kernel (figure 4.1(h)). As pointed out in chapter 3, we are using a disturbance factors invariant change detection algorithm which inherently suffers of the missed detections problem in correspondence of poorly structured patch of pixels (here, of blocks of pixels). On the other hand, the algorithm has very good detection capabilities in correspondence of structured patches. In particular, foreground objects boundaries are detected in a quite continuous manner. Here, we are applying the algorithm at reduced resolution. The effect is that the detected coarse-grain objects boundaries are almost always continuous. In practice, at reduced resolution the application of simple binary morphological elaborations allows to solve the missed detections problem, so that reliable super-masks of the semantically changed pixels are very likely to be attained. Conversely, the complements of this change masks are very likely to contain just semantically unchanged pixels. Therefore, both in the full resolution background and in the full resolution current frame the pixels belonging to the complement of the currently computed coarse-grain change mask are very likely imaging the scene background surface (figures 4.1(i,k)). In particular, a common pixel in the background and in the current frame is imaging the same scene background surface patch. Since the coarse-level detection we are using is very robust to even very fast appearance changes of the background surface, as it is the case in figure 4.1, the complement of the computed coarse-grain change mask can be exploited to ”understand” the effects of the occurring appearance changes before the fine-level detection is performed. In practice, the measured intensities in the background and in the current frame for all the pixels belonging to the complement can be ”compared” to infer the intensity mapping function that best explains the appearance changes effects. In figure 4.1(j) we show the comparagram computed by using the background and current frame intensities of all the pixels in the complement. The intensity mapping function may be computed by a simple comparagram regression, after having assumed a parametric functional form. To avoid arbitrary assumptions, we use an alternative method, called histogram specification. In practice, the two cumulative histograms of the intensities of the complement pixels in the background and in the current frame are computed. The intensity mapping function is inferred by looking for the function which best transforms the background histogram into the current frame histogram. In figure 4.1(j) we show the intensity mapping function inferred by the histogram specification procedure by drawing it on the comparagram.

Once the intensity mapping function has been computed, it can be applied to the current full-resolution background model. In this way we perform a tonal registration of the background ”toward” the current frame. What we are doing is filtering-out the effects of the scene surface appearance changes. As a consequence, the subsequent

fine-level will always have at disposal a tonally registered background model, even in case of sudden changes. Moreover, since the coarse-grain change masks are very likely a superset of the semantically changed pixels, just the foreground pixels in the coarse-grain change masks have to be considered. In practice, for each of these pixels the fine-level computes the background subtraction by comparing the intensities in the current frame and in the registered background. In figure 4.1(l) we show the fine-level change mask attained by applying the background subtraction procedure proposed in chapter 2. Figures 4.1(m,n) depict the final change mask, attained after a simple morphological elaboration.

4.2 Experimental Results

We present the detection results of the proposed coarse-to-fine approach for two different sample frames, each one belonging to a different video sequence. In figure 4.2 a sample frame of an indoor video sequence is taken into consideration. In particular, a real scene illumination change (a scene darkening) is occurring, yielding a quite linear intensity mapping function. In figure 4.3 a frame of an outdoor sequence is shown. Here, a synthetic non linear intensity mapping function has been applied to the frame before the elaboration. It is straightforward noticing how in both the cases the proposed approach allows to attain quite accurate change masks. These results can be attained neither by the temporally adaptive approach presented in chapter 2 nor by the disturbance factors invariant algorithm proposed in chapter 3.

4.3 Conclusions

In this chapter we have presented a single-view change detection approach based on a coarse-to-fine strategy. In particular, we have shown how a disturbance factors invariant approach and a temporally adaptive approach can be used together in a coarse-to-fine framework to attain very good detection results. This approach allows to deal effectively with all the disturbance factors yielding effects corresponding to global (i.e. spatially invariant in the entire frame) intensity mapping functions. Actually, very local false changes, such as those due to specularities and shadows cast by foreground objects, can not be filter-out by the algorithm. In the next chapter, we show how a multi-view change detection approach allows to face this challenging problem as well.

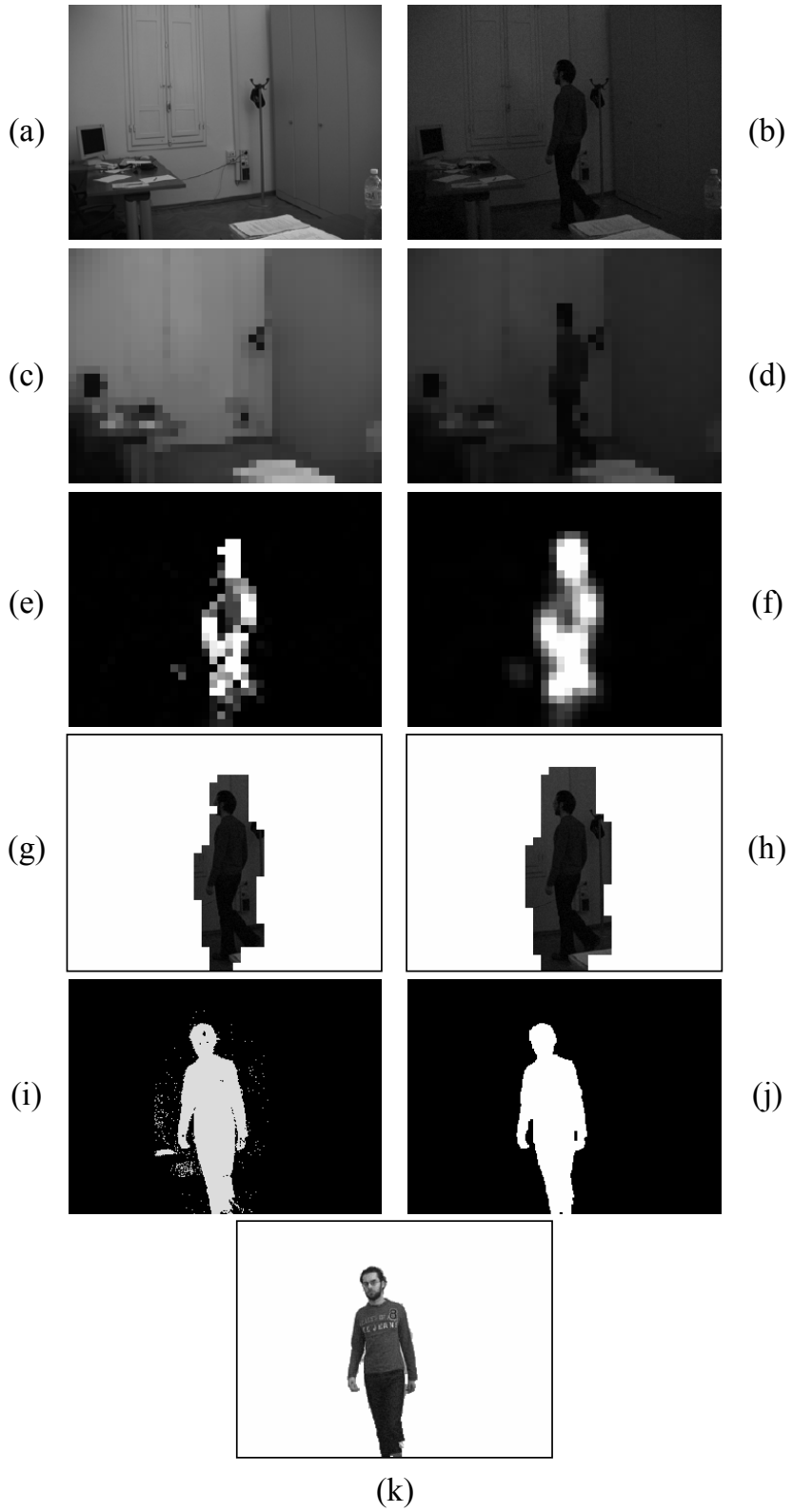


Figure 4.2: Main processing steps of the proposed coarse-to-fine approach for a sample frame of an indoor sequence in which a real scene illumination change is occurring.

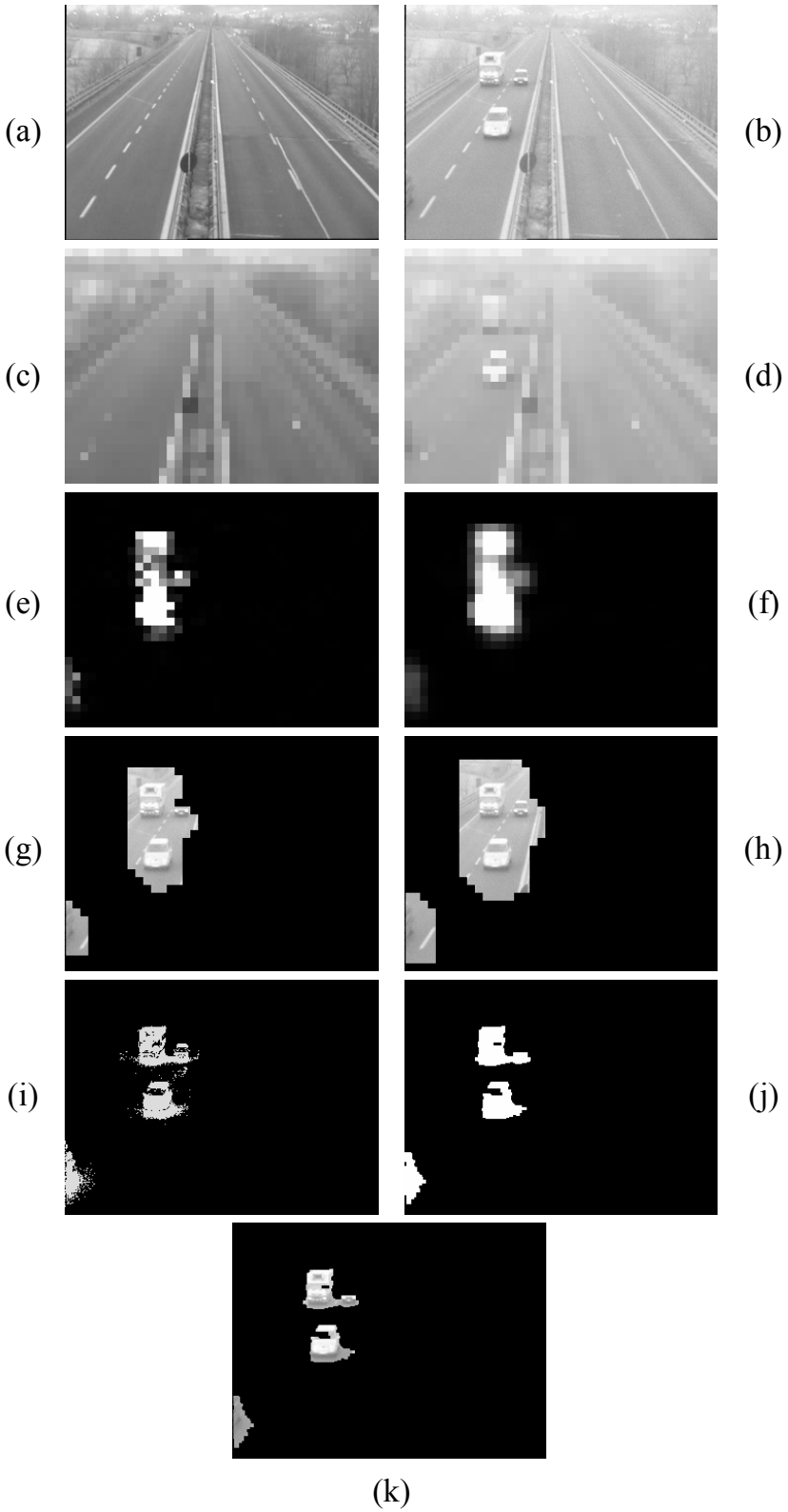


Figure 4.3: Main processing steps of the proposed coarse-to-fine approach for a sample frame of an outdoor sequence in which a synthetic non-linear intensity mapping function has been applied.

Chapter 5

Multi-view Change Detection

By means of a coarse-to-fine strategy, the single-view change detection approach presented in the previous chapter allows to attain accurate change masks also in case that most of the possible disturbance factors are acting. In particular, quite "global" scene illumination changes as well as all the possible disturbance factors due to the capturing device (e.g. noise, AE, AGC, γ-correction) can be dealt with effectively by the proposed algorithm. On the contrary, the effects of very local scene illumination changes, such as those due to specularities and to shadows cast by foreground objects, can be filtered-out neither by the disturbance factors invariant coarse-level nor by the tonally-registered temporally adaptive fine-level. Indeed, specularities are inherently a very hard-to-solve problem in change detection, since their effect is often very local and strongly dependent on the scene surface physical properties. In other words, specular reflection is a complex phenomenon which yields hardly predictable effects on the measured image intensities. This is even more the case for change detection from grey level images, where no color information can be exploited to recognize specularities effects. As regards shadows, many algorithms have been proposed aimed at detecting and removing image changes due to shadows from single-view video sequences. Most of them rely on photometric assumptions concerning the local effects of shadows on radiance emitted by the scene surface, that is on the measured intensities. In this chapter we present a multi-view change detection approach aimed at filtering-out all the local false image changes by exploiting just a geometrical constraint.

The chapter is organized as follows. In section 5.1 the multi-view change detection problem is defined and formalized. In section 5.2 the state-of-the-art in multi-view change detection is outlined. The proposed algorithm is presented in section 5.3. Experimental results are discussed in section 5.4 and conclusions are drawn in section 5.5.

5.1 Problem Definition and Formalization

At time $t = \bar{t}$, the input $\mathcal{I}_{\bar{t}}$ of a typical multi-view change detection algorithm is a matrix of $(T+1) \cdot V$ different scene images, as illustrated in figure 5.1(a):

$$\mathcal{I}_{\bar{t}} = \begin{pmatrix} I^1_{\bar{t}-T} & I^2_{\bar{t}-T} & \dots & I^V_{\bar{t}-T} \\ \vdots & \vdots & & \vdots \\ I^1_{\bar{t}-1} & I^2_{\bar{t}-1} & \dots & I^V_{\bar{t}-1} \\ I^1_{\bar{t}} & I^2_{\bar{t}} & \dots & I^V_{\bar{t}} \end{pmatrix} \tag{5.1}$$

where V is the number of different view-points and $(T+1)$ is the number of different images taken as input for each view-point. In particular, each column $\boldsymbol{S}^v_{\bar{t}} = \left(I^v_{\bar{t}-T}, \dots, I^v_{\bar{t}-1}, I^v_{\bar{t}} \right)^\top$ represents a different input video sequence, that is it contains a set of $(T+1)$ scene images captured at different times from the same view-point. On the other hand, each row $\boldsymbol{R}_t = \left(I^1_t, I^2_t, \dots, I^v_t \right)$ contains a set of V scene images taken at the same time from different view-points.

At time $t = \bar{t}$, the output $\mathcal{O}_{\bar{t}}$ of a typical multi-view change detection algorithm is in general a set of change masks, that is of binary images. In particular, three different output types are possible:

o.1) the output consists of V different change masks, that is a change mask is computed for each one of the original views (figure 5.1(b)):

$$\mathcal{O}_t = \left(\boldsymbol{C}^1_{\bar{t}}, \boldsymbol{C}^2_{\bar{t}}, \dots, \boldsymbol{C}^V_{\bar{t}} \right) \tag{5.2}$$

o.2) the output consists of a single change mask, computed for one of the original views, called the "reference" or "primary" view (figure 5.1(c)):

$$\mathcal{O}_t = \left(\boldsymbol{C}^r_{\bar{t}} \right) \qquad r \in [1, V] \tag{5.3}$$

o.3) the output consists of a single change mask, computed for a virtual view (figure 5.1(d)):

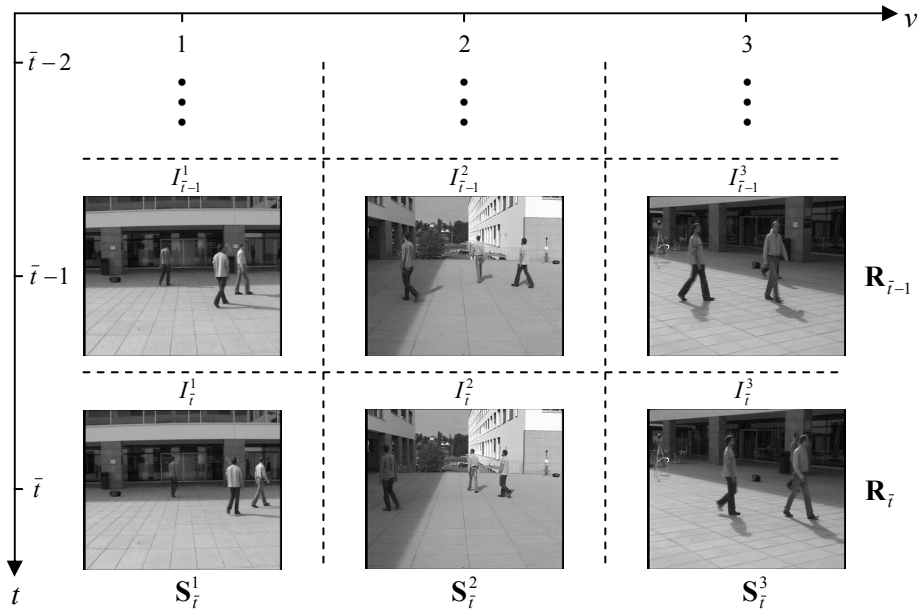
$$\mathcal{O}_t = \left(\boldsymbol{C}^{\mathcal{V}}_{\bar{t}} \right) \tag{5.4}$$

In general, a multi-view change detection algorithm is a procedure aimed at computing the output $\mathcal{O}_{\bar{t}}$ given the input $\mathcal{I}_{\bar{t}}$:

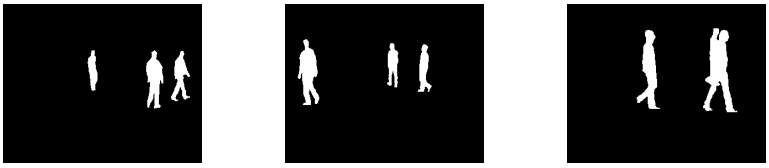
$$\mathcal{O}_t = CD_{MV}(\mathcal{I}_t) \tag{5.5}$$

As regards the different types of possible procedures, we define:

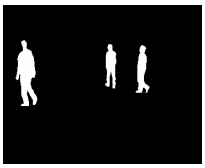
c.1) *temporal consistency* constraint: the frames contained in a column of the input matrix $\mathcal{I}_{\bar{t}}$ of equation 5.1 are images of the same scene taken at different times from the same view-point;



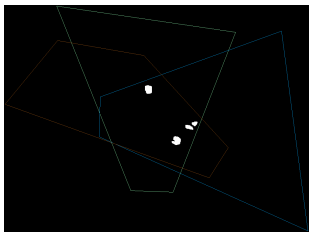
(a) input



(b) output type 1: a change mask in each original view



(c) output type 2: a change mask in a reference view



(d) output type 3: a change mask in a virtual view

Figure 5.1: Typical input and outputs of a multi-view change detection algorithm ($V = 3$).

c.1) *spatial coherence* constraint: the frames contained in a row of the input matrix $\mathcal{I}_{\bar{t}}$ of equation 5.1 are images of the same scene taken at the same time from

different view-points;

By applying just the temporal consistency constraint, we perform V independent single-view change detections, one for each different view-point. On the other hand, by applying just the spatial coherence constraint the simplest multi-view change detection approach is attained. In practice, at each time t the output O_t is attained by processing the row $\boldsymbol{R}_t$, that is by comparing all the current simultaneous scene images captured from different view-points. Finally, by applying both the temporal consistency and the spatial coherence constraints, all the available information is exploited. Hence, this is in theory the most effective approach. We present a multi-view change detection algorithm of this type. In particular, we apply the temporal consistency constraint as a first processing step by performing a single-view change detection in each original view. Then, the spatial coherence constraint is applied by computing a ”fusion” of the attained change masks in a virtual top-view.

5.2 Related Work

In [21] a ”lighting independent” multi-view change detection algorithm is presented. Stationarity of the capturing devices as well as of the scene background surface geometry is assumed, so that the geometric transformations warping one of the views (called ”primary” view) into all the other views (called ”auxiliary” views) can be computed off-line. On-line, just the change mask in the primary view is computed. Moreover, just the spatial coherence constraint is applied. In practice, at each time the color of every pixel in the primary view is compared with the color of corresponding pixels (through the geometric transformations) in the auxiliary views. If color is similar (according to a simple metric consisting in the absolute value of the Euclidean distance), the pixel in the primary view is marked as background; otherwise, it is marked as foreground. This approach inherently suffers from both false and missed detections. False detections (called ”occlusion shadows”) occur when a background pixel in the primary view is occluded by a foreground object in the auxiliary view. Missed detections occur when an evenly colored foreground object occludes a pair of corresponding pixels, for color being very similar. The authors propose to filter-out false detections by using more than two views (at least two auxiliary views) and ANDing the binary masks attained by comparing the primary view to each of the auxiliary views. However, they do not discuss how to deal with missed detections.

The work in [25] is aimed at improving the approach proposed in [21]. As in [21], the change mask in the primary view is computed by just applying the spatial coherence constraint. However, the following improvements are proposed:

a) a slightly more complex and effective metric (i.e. a normalized color difference averaged on a $n \times n$ neighborhood of pixels) is used to measure color similarity between corresponding pixels in different views;

b) the occlusion shadows problem is addressed by a sensor planning perspective. In particular, it is shown how false detections can be removed by using just two views, provided that a suitable configuration of the capturing devices is adopted;

c) the missed detections problem is tackled as well. The particular sensors configuration adopted to filter-out occlusion shadows yields missed detections localized only at the lower portion of each detected foreground blob. However, a complex and quite fragile algorithm is proposed to fill-in the possible missed detections. In fact, for each foreground blob in the primary view detected by the spatial coherence constraint application, all the "top-most" pixels along each epipolar line passing through the blob bounding-box are identified. For each of these pixels, the corresponding "base-point" pixel is computed, that is the pixel lying below on the ground plane. The computation is performed by an iterative search along the epipolar line through the top-most pixel;

d) specularities tend to be removed as a side effect of the missed detections reduction algorithm explained in c).

Both the algorithms in [21] and [25] rely on the assumption that a patch of the scene background surface yields a very similar color into simultaneous images taken from different view-points (figure 5.2). If this is true, a total invariance to temporal

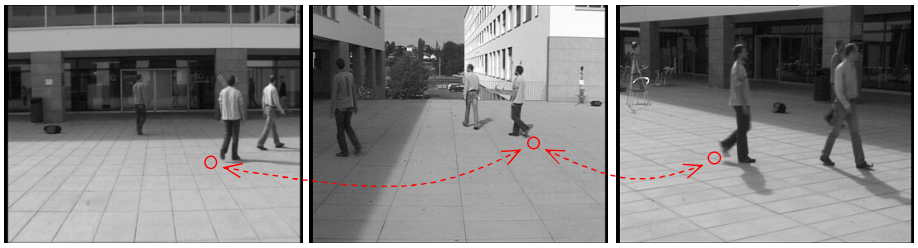


Figure 5.2: Sample patch of a scene background surface imaged simultaneously by different view-points.

changes of the radiance emitted by the scene background surface (e.g. scene illumination changes, shadows) is achieved, since such changes will affect simultaneous views in an identical manner. However, in practice this assumption may not be satisfied, mainly for two reasons. Firstly, in case of non-lambertian surfaces the reflected light intensity depends on the viewing angle (specular reflection). Secondly, dynamic adjustments of the capturing devices parameters (e.g. auto-gain and auto-exposure) may

occur for different views at different times and by a different intensity mapping function. As for specularities, [25] can deal with this problem as a side effect of the method used to reduce missed detections. Conversely, dynamic adjustments of imaging systems parameters cannot be handled inherently by neither [21] nor [25]. In turn, [21] recommends explicitly to disable the auto-gain mechanism of the capturing devices. However, to disable these dynamic adjustment mechanisms is a strong limitation in many practical applications, especially as regards outdoor installations.

The most related work to the approach we are going to propose in this chapter is that presented in [23]. It is focused on tracking but relies on multi-view change detection as the first processing step. People moving on a ground plane (i.e. a planar background surface) are tracked by their ground locations, that is the feet. At each processing time, feet are detected by a multi-view change detection approach, that we call here "Change Maps Fusion" (*CMF*): the ground plane homographies warping a reference view into each of the other views are inferred off-line. On-line, single-view change detection is performed independently in each view to attain a change probability map. To this purpose, a well-known background subtraction algorithm ([37]) based on a statistical temporally adaptive background modeling by mixture of gaussians is deployed. Hence, the computed change probability maps are warped in the reference view by using the inferred homographies and then multiplied together, thus attaining a "synergy map". It is easy to understand how this map gives for each pixel in the reference view the probability to be the image of a scene background surface patch (i.e. of the ground plane) for which the emitted radiance is changed (with respect to the current appearance background model and according to the chosen single-view change detection algorithm). Finally, the synergy map is thresholded. By this procedure, the authors assume to detect just the ground plane locations of people, that is the feet. Hence, feet are tracked in the reference view by a spatio-temporal clustering approach (graph cuts). However, the proposed use of the CMF approach will inherently detect as foreground not just the feet but also possible appearance changes of the scene background surface due to specularities and shadows, unless such changes are filtered-out by the single-view change detection processes (indeed, this would not happen with the approach used in [37]). Hence, detection of shadows or specularities in addition to feet is likely to induce failures in the tracking algorithm proposed in [23].

5.3 The proposed algorithm

We assume a stationary scene background surface and stationary capturing devices. Moreover, we consider a planar background surface (hereinafter, ground plane). Off-line, for each original view v we infer the homography H^v warping each pixel (x^v, y^v)

imaging a ground plane patch in the original view into the pixel (x^t, y^t) imaging the same ground plane patch in a common virtual top-view T:

$$H^v : \mathbb{R}^2 \ni (x^v, y^v) \mapsto (x^T, y^T) \in \mathbb{R}^2 \qquad H^v : \begin{cases} x^T = \dfrac{h_1^v \cdot x^v + h_2^v \cdot y^v + h_3^v}{h_4^v \cdot x^v + h_5^v \cdot y^v + 1} \\ y^T = \dfrac{h_6^v \cdot x^v + h_7^v \cdot y^v + h_8^v}{h_4^v \cdot x^v + h_5^v \cdot y^v + 1} \end{cases} \tag{5.6}$$

where $\boldsymbol{h}^v = (h_1^v, h_2^v, \ldots, h_8^v)$ is the homography parameters array. The inference is computed by considering a set of $N > 4$ chosen original view $\leftrightarrow$ top-view points correspondences:

$$\begin{array}{c} (x_1^v, y_1^v) \mapsto (x_1^T, y_1^T) \\ (x_2^v, y_2^v) \mapsto (x_2^T, y_2^T) \\ \vdots \\ (x_N^v, y_N^v) \mapsto (x_N^T, y_N^T) \end{array} \tag{5.7}$$

and by solving the following over-determined system of linear equations:

$$\underbrace{\begin{pmatrix} x_1^v & y_1^v & 1 & 0 & 0 & 0 & -x_1^v \cdot x_1^T & -y_1^v \cdot x_1^T \\ 0 & 0 & 0 & x_1^v & y_1^v & 1 & -x_1^v \cdot y_1^T & -y_1^v \cdot y_1^T \\ \multicolumn{8}{c}{\vdots} \\ x_N^v & y_N^v & 1 & 0 & 0 & 0 & -x_N^v \cdot x_N^T & -y_N^v \cdot x_N^T \\ 0 & 0 & 0 & x_N^v & y_N^v & 1 & -x_N^v \cdot y_N^T & -y_N^v \cdot y_N^T \end{pmatrix}}_{\boldsymbol{A}^v} \underbrace{\begin{pmatrix} h_1^v \\ h_2^v \\ \vdots \\ h_8^v \end{pmatrix}}_{\boldsymbol{h}^v} = \underbrace{\begin{pmatrix} x_1^T \\ y_1^T \\ \vdots \\ x_N^T \\ y_N^T \end{pmatrix}}_{\boldsymbol{b}^v} \tag{5.8}$$

In particular, a simple least squares solution is computed as follows:

$$\boldsymbol{h}^v = \left((\boldsymbol{A}^v)^\top \cdot \boldsymbol{A}^v \right)^{-1} \cdot (\boldsymbol{A}^v)^\top \cdot \boldsymbol{b}^v \tag{5.9}$$

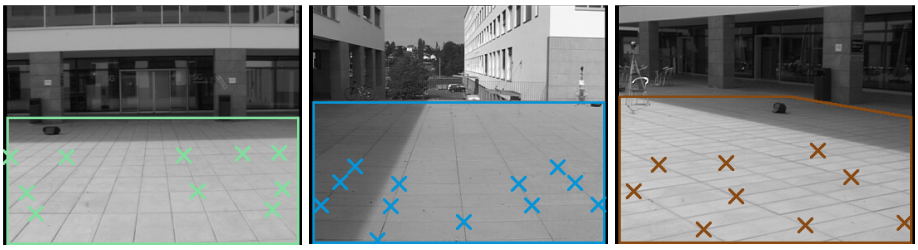
An input and output example for the homographies inference procedure is illustrated in figure 5.3.

As far as the on-line elaboration is concerned (figure 5.4), at each processing time t the temporal consistency constraint is firstly applied by performing a single-view change detection independently in each original view. A set of V binary change masks C_t^v, one for each original view v, is attained (figures 5.4(d-f)):

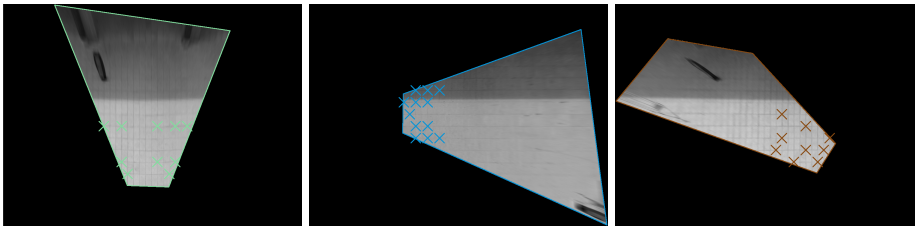
$$C_t^v = CD_{SV}(\boldsymbol{S}_t^v) \qquad v = 1, 2, \ldots, V \tag{5.10}$$

The spatial coherence constraint is then applied by projecting all the change masks (actually, just the change masks portion inside the ground plane limits are projected) into the virtual top-view, thus attaining a set of V top-view change masks $C_t^{v,T}$ (figures 5.4(g-i)):

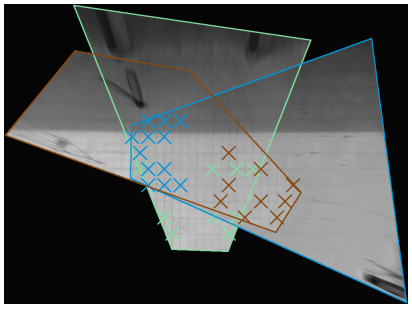
$$C_t^{v,T} = H^v(C_t^v) \tag{5.11}$$



(a) scene background surface (ground plane) imaged by different view-points; ground plane limits and control points in the original views used to infer the original views ↔ top-view homographies



(b) control points in the virtual top-view used to infer the homographies; ground plane limits and ground plane images projected in the virtual top-view by the inferred homographies



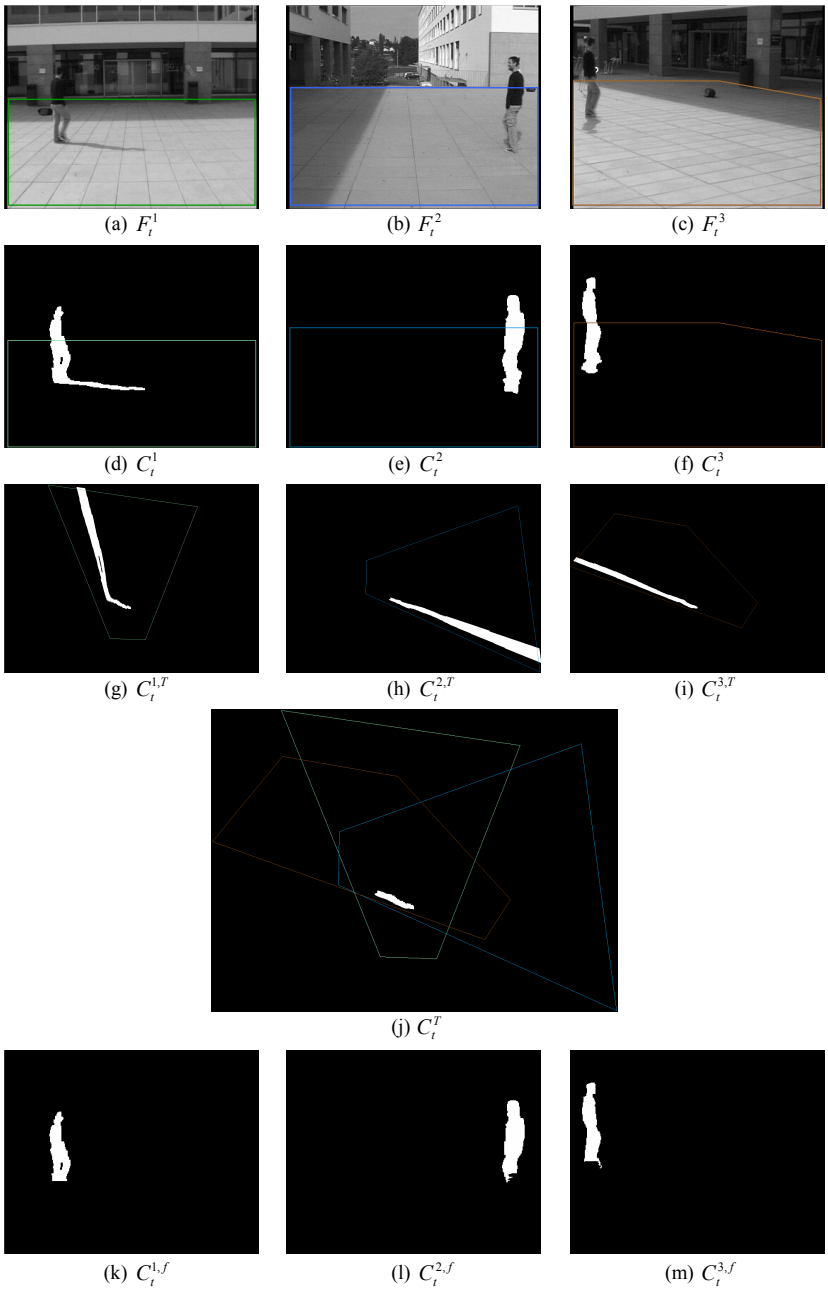
(c) mosaic of control points, ground plane limits and ground plane images projected in the virtual top-view by the inferred homographies

Figure 5.3: Off-line inference of the original views ↔ top-view homographies

Hence, a common top-view change mask C_t^T is attained by computing the intersection of all the top-view change masks (figure 5.4(j)):

$$C_t^T(\boldsymbol{p}) = \prod_{v=1}^{V} C_t^{v,T}(\boldsymbol{p}) \tag{5.12}$$

The procedure outlined so far is very similar to the change maps fusion approach presented in [23]. The only difference is that change maps binarization is performed directly as a final processing step of the temporal consistency constraint application. On the other hand, in [23] binarization is carried out in the virtual top-view after the spatial coherence constraint has been applied as well. We can call *change masks fusion* this slightly different approach and *synergy mask* the binary mask of equation 5.12. The synergy mask is nothing else than a binary (i.e. thresholded) version of the synergy



(a) F_t^1 (b) F_t^2 (c) F_t^3

(d) C_t^1 (e) C_t^2 (f) C_t^3

(g) $C_t^{1,T}$ (h) $C_t^{2,T}$ (i) $C_t^{3,T}$

(j) C_t^T

(k) $C_t^{1,f}$ (l) $C_t^{2,f}$ (m) $C_t^{3,f}$

Figure 5.4: On-line main processing steps of the proposed multi-view change detection approach

map computed in [23]. In practice, the synergy mask contains the pixels characterized by a high probability to be the image of a ground plane patch for which the emitted radiance is changed (i.e. people feet as well as shadows cast by people on the ground

plane). Now, we utilize the synergy mask in a "dual" manner with respect to [23]. Instead of looking at the synergy mask as to a detection of just the foreground objects ground locations (people feet), we consider it as a detection of just the false changes due to variations of the ground plane emitted radiance (shadows). Hence, instead of considering the synergy mask as the final output of the multi-view change detection, we back-project the synergy mask into all the original views, thus attaining a set of V original views synergy masks $C_t^{T,v}$:

$$C_t^{T,v} = (H^v)^{-1}\left(C_t^T\right) \tag{5.13}$$

Finally, for each view v we filter-out the foreground pixels of the just computed original view synergy mask $C_t^{T,v}$ from the previously computed original view change mask C_t^v, thus attaining a set of V final change masks $C_t^{v,f}$ (figures 5.4(k-m)):

$$C_t^{v,f}(\boldsymbol{p}) = \begin{cases} 0 & \text{if} \quad C_t^{T,v}(\boldsymbol{p}) = 1 \\ C_t^v(\boldsymbol{p}) & \text{otherwise} \end{cases} \tag{5.14}$$

Differently from [23], where the output is a single change mask in the virtual top-view, we attain a set of V change masks, one for each different view.

The proposed approach is "general-purpose", in the sense that all the scene appearance changes detected by the employed single-view change detection algorithm which satisfy the spatial coherence constraint (i.e. which arise "near" the ground plane in a 3-dimensional sense) are filtered-out. In fact, no selectivity criterion is utilized in the removing rule of expression 5.14. In practice, just a geometrical constraint is applied, without considering any photometric information. On one hand this approach is general-purpose, but on the other hand a missed detections problem may arise due to the following two causes:

a) part of the foreground objects ground locations (people feet) may be removed together with the actual false changes (shadows) from the final change masks (figure 5.4(k)). This is an inherent and easy to understand problem of the proposed approach, since ground locations of foreground objects yield appearance changes lying "near" the ground plane (i.e. they satisfy the spatial coherence constraint);

b) some "off-ground" portions of the foreground objects may be removed as well. This may occur for the original views in which the ground plane appearance changes are covered by foreground objects (figure 5.4(l)). This is an inherent problem as well. In general, the higher is the number of foreground objects present in the scene, the higher is the probability of this problem to occur.

To face these two inherent problems, we propose a less "general-purpose" removing rule than the one in expression 5.14. We call this rule a "shadows-focused" removing

rule. In fact, we try to compute a selective removal of just the ground plane appearance changes due to shadows. To this purpose, we exploit simple, well-known and commonly used photometric properties characterizing scene surfaces covered by shadows. The basic idea is that the measured intensity of a pixel imaging a scene background surface patch decreases according to a limited darkening factor d when covered by a cast shadow, independently from the considered view-point. Hence, the selective "shadows-focused" removing rule is the following:

$$C_t^{v,f}(\boldsymbol{p}) = \begin{cases} 0 & \text{if} \quad \left(C_t^{T,v}(\boldsymbol{p}) = 1\right) \wedge \left(d_{low} < \dfrac{F_t^v(\boldsymbol{p})}{\hat{B}_t^v(\boldsymbol{p})} < 1\right) \\ C_t^v(\boldsymbol{p}) & \text{otherwise} \end{cases} \tag{5.15}$$

where d_{low} is the lower darkening factor assumed for shadows effect and F_t^v, $\hat{B}_t^v$ are, respectively, the current frame and the current background model used by the single-view change detection algorithm in the original view v. In practice, for each view v the final change mask $C_t^{v,f}$ is not attained by filtering-out blindly all the foreground pixels of the original view synergy mask $C_t^{T,v}$ from the original view change mask C_t^v. Instead, just the foreground pixels satisfying the shadows photometric constraint are removed.

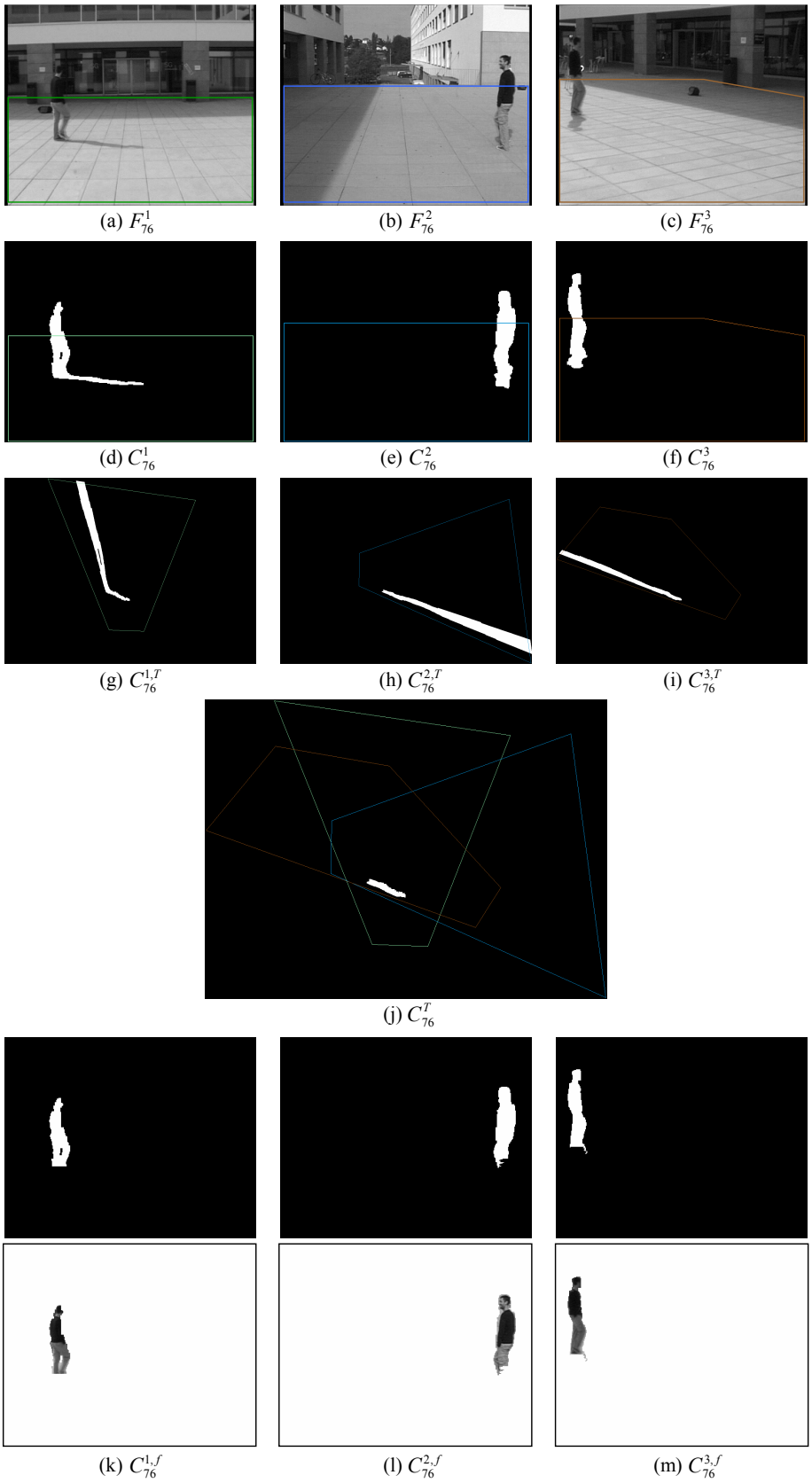
5.4 Experimental Results

Experiments have been carried out by elaborating several multi-view input video sequences, all taken by the same outdoor multi-view installation. The installation consists of three synchronized capturing devices imaging a common scene from very different view-points. Here we present the detection results for four different capturing times (i.e. for four different triples of simultaneous frames) of one of these sequences. In figures 5.5-5.8 the on-line main processing steps of the proposed algorithm are shown for each one of the four chosen capturing times. In particular, the results attained by the general-purpose approach (blind removing rule in 5.14) are depicted. To point-out how the shadows-focused approach (selective removing rule in 5.15) can improve the detection results, in figures 5.9-5.12 we directly compare the change masks attained by the general-purpose and the shadows-focused versions of the proposed multi-view change detection algorithm. In particular, a value $d_{low} = 0.5$ is used in the shadows-focused removing rule.

5.5 Conclusions

In this chapter we have presented a multi-view change detection approach aimed at being very robust to most of the possible disturbance factors. To this purpose, the task

to filter-out the effects of "global" false changes, such as those due to global scene illumination changes or to dynamic adjustments of the capturing devices parameters (e.g. AE, AGC and γ-correction) is assigned to a single-view change detection performed independently in each original view (for example, by the algorithm presented in chapter 4). A very hard-to-solve problem in single-view change detection is the effect of local changes of the radiance emitted by the scene background surface (e.g. changes due to specularities and shadows cast by foreground objects). By applying the spatial coherence constraint as a final processing step, the proposed multi-view approach filters-out effectively all these local false changes, as pointed out by experiments. However, a missed detections problem may arise, due to causes which are inherent to the proposed algorithm. For this reason, a less general-purpose version of the algorithm has been proposed, aimed at removing just local false changes due to shadows. Since the available sample multi-view sequences are all characterized by the presence of local false changes only due to shadows, the shadows-focused approach yields better results than the general-purpose approach.



(a) F_{76}^{1} (b) F_{76}^{2} (c) F_{76}^{3}

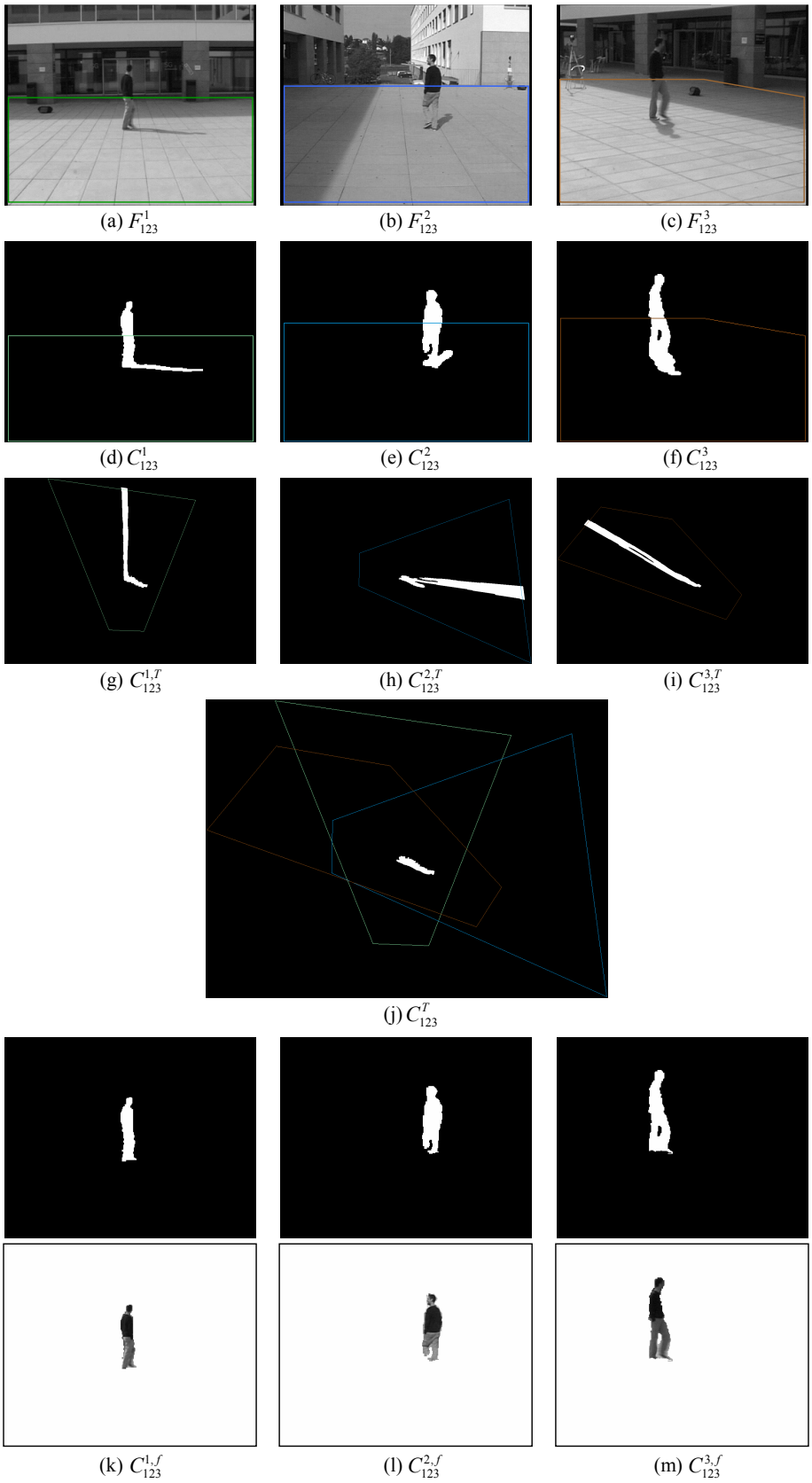
(d) C_{76}^{1} (e) C_{76}^{2} (f) C_{76}^{3}

(g) $C_{76}^{1,T}$ (h) $C_{76}^{2,T}$ (i) $C_{76}^{3,T}$

(j) C_{76}^{T}

(k) $C_{76}^{1,f}$ (l) $C_{76}^{2,f}$ (m) $C_{76}^{3,f}$

Figure 5.5: On-line main processing steps of the proposed multi-view change detection approach (general-purpose version) for frame 76.



(a) F_{123}^{1} (b) F_{123}^{2} (c) F_{123}^{3}

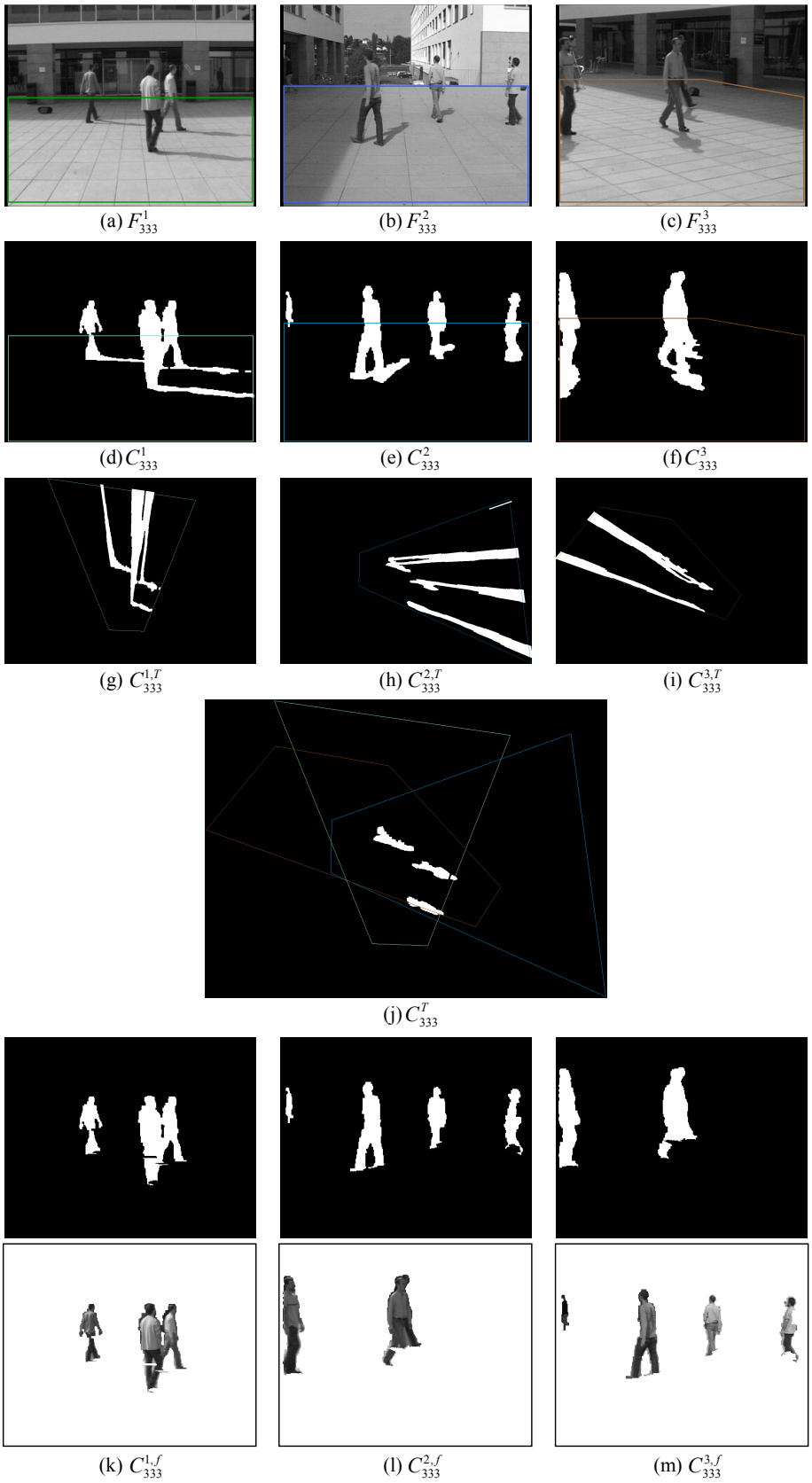
(d) C_{123}^{1} (e) C_{123}^{2} (f) C_{123}^{3}

(g) $C_{123}^{1,T}$ (h) $C_{123}^{2,T}$ (i) $C_{123}^{3,T}$

(j) C_{123}^{T}

(k) $C_{123}^{1,f}$ (l) $C_{123}^{2,f}$ (m) $C_{123}^{3,f}$

Figure 5.6: On-line main processing steps of the proposed multi-view change detection approach (general-purpose version) for frame 123.



(a) F^1_{333} (b) F^2_{333} (c) F^3_{333}

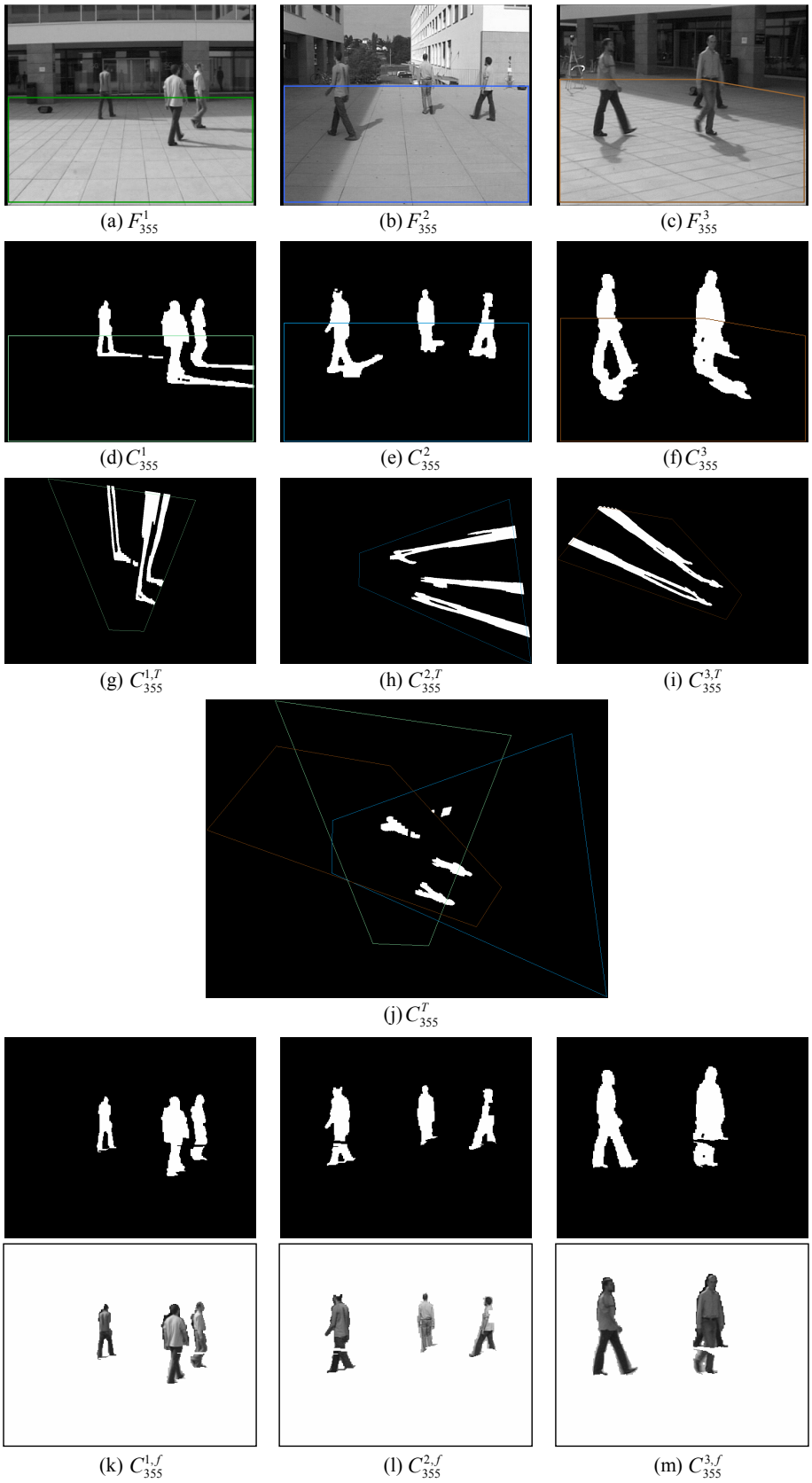
(d) C^1_{333} (e) C^2_{333} (f) C^3_{333}

(g) $C^{1,T}_{333}$ (h) $C^{2,T}_{333}$ (i) $C^{3,T}_{333}$

(j) C^T_{333}

(k) $C^{1,f}_{333}$ (l) $C^{2,f}_{333}$ (m) $C^{3,f}_{333}$

Figure 5.7: On-line main processing steps of the proposed multi-view change detection approach (general-purpose version) for frame 333.



(a) F_{355}^{1} (b) F_{355}^{2} (c) F_{355}^{3}

(d) C_{355}^{1} (e) C_{355}^{2} (f) C_{355}^{3}

(g) $C_{355}^{1,T}$ (h) $C_{355}^{2,T}$ (i) $C_{355}^{3,T}$

(j) C_{355}^{T}

(k) $C_{355}^{1,f}$ (l) $C_{355}^{2,f}$ (m) $C_{355}^{3,f}$

Figure 5.8: On-line main processing steps of the proposed multi-view change detection approach (general-purpose version) for frame 355.

Figure 5.9: Comparison between the change masks attained by the general-purpose (center row) and the shadows-focused (bottom row) versions of the proposed multi-view change detection approach for frame 76 (top row).

Figure 5.10: Comparison between the change masks attained by the general-purpose (center row) and the shadows-focused (bottom row) versions of the proposed multi-view change detection approach for frame 133 (top row).

Figure 5.11: Comparison between the change masks attained by the general-purpose (center row) and the shadows-focused (bottom row) versions of the proposed multi-view change detection approach for frame 333 (top row).

Figure 5.12: Comparison between the change masks attained by the general-purpose (center row) and the shadows-focused (bottom row) versions of the proposed multi-view change detection approach for frame 355 (top row).

Chapter 6

Conclusions

This thesis deals with the development and evaluation of change detection algorithms aimed at being robust to most of the possible disturbance factors arising in the typical unconstrained environments concerned by the most common change detection applications (e.g. video-surveillance, traffic monitoring).

Two very different change detection approaches have been presented in chapters 2 and 3. The algorithm proposed in chapter 2 is a temporally adaptive method, that is it relies on a statistical appearance model of the scene background surface which has to be updated along time. As well as all the algorithms of this type, the proposed approach yields accurate detection results in ”common” cases, that is when no sudden appearance changes of the scene background surface occur. On the contrary, if these changes occur a remarkable false detections problem unavoidably arises. The approach presented in chapter 3 is a disturbance factors invariant algorithm. The classification of each pixel of the current frame as changed or unchanged is carried out by exploiting the information contained in a neighborhood of pixels. In particular, the hypothesis that just disturbance factors are acting in the neighborhood is tested by a maximum-likelihood isotonic regression approach. The background model has not to be updated and disturbance factors are dealt with effectively. However, an inherent missed detections problem arises in correspondence of ”poorly structured” scene regions.

In chapter 4 we show how the two algorithms can be used together to attain a very effective change detection approach. In particular, by using the algorithms within a coarse-to-fine framework all the ”global” effects of disturbance factors are filtered-out effectively. However, very local effects can not be dealt with by this approach.

In chapter 5 we propose a multi-view change detection, aimed at improving the results attained by the single-view coarse-to-fine approach, that is at filtering-out possible local effects of disturbance factors, such as shadows and specularities. We show how the application of the multi-view spatial coherence constraint as the final processing

step (i.e. after the temporal consistency constraint) allows to remove quite effectively this local false changes as well.

Depending on the single-view or multi-view available installation, the algorithms presented in chapters 4 and 5 allows to attain accurate change masks. Hence, they can be used as e reliable first processing step upon which higher level capabilities (e.g. objects tracking, classification and behavior analysis) can be built.

Parts of the research results presented in this thesis have been published in [4], [3], [8], [7], [6], [5], [24]. The other results will be the subject of future papers.

Bibliography

[1] R. E. Barlow, D. J. Bartholomew, J. M. Bremmer, and H. D. Brunk. *Statistical Inference under Order Restrictions*. Wiley, New York, 1972.

[2] A. Bevilacqua. A novel background initialization method in visual surveillance. In *IAPR Workshop on Machine Vision Applications (MVA02)*, pages 614–617, December 2002.

[3] A. Bevilacqua, L. Di Stefano, and A. Lanza. An efficient change detection algorithm based on a statistical non-parametric camera noise model. In *Proceedings of 2004 IEEE International Conference on Image Processing (ICIP'04)*, 2004.

[4] A. Bevilacqua, L. Di Stefano, and A. Lanza. High-quality speed dilemma: a comparison between segmentation methods for traffic monitoring applications. In *Proceedings of 2004 International Conference on Image Analysis and Recognition (ICIAR'04)*, 2004.

[5] A. Bevilacqua, L. Di Stefano, and A. Lanza. Coarse-to-fine strategy for robust and efficient change detectors. In *Proceedings of 2005 IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS'05)*, 2005.

[6] A. Bevilacqua, L. Di Stefano, and A. Lanza. An effective multi-stage background generation algorithm. In *Proceedings of 2005 IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS'05)*, 2005.

[7] A. Bevilacqua, L. Di Stefano, and A. Lanza. A novel approach to change detection based on a coarse-to-fine strategy. In *Proceedings of 2005 IEEE International Conference on Image Processing (ICIP'05)*, 2005.

[8] A. Bevilacqua, L. Di Stefano, and A. Lanza. A simple self-calibration method to infer a non-parametric model of the imaging system noise. In *2005 IEEE Workshop on Motion and Video Computing (MOTION'05)*, 2005.

[9] R.A. Boie and I.J. Cox. An analysis of camera noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6):671–674, June 1992.

[10] Y.C. Chang and J.F. Reid. RGB calibration for color image analysis in machine vision. *IEEE Transactions on Image Processing*, 5(10):1414–1422, 1996.

[11] G. Cortellazzo, G.A. Mian, and R. Parolari. Statistical characteristics of granular camera noise. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(6):536–543, December 1994.

[12] M. Cristani, M. Bicego, and V. Murino. Multi-level background initialization using hidden markov models. In *Proceedings of the* 1^{st} *ACM SIGMM Workshop on Video Surveillance (IWVS03)*, pages 11–20, November 2003.

[13] P.E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the* 24^{th} *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH97)*, pages 369–378, August 1997.

[14] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proceedings of 1999 IEEE International Conference on Computer Vision (ICCV99) FRAME-RATE workshop*, September 1999.

[15] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Annual Conference on Uncertainty in Artificial Intelligence*, pages 175–181, 1997.

[16] B. Gloyer, H.K. Aghajan, K.Y. Siu, and T. Kailath. Video-based freeway monitoring system using recursive vehicle tracking. In *Proceedings of IS T-SPIE Symposium on Electronic Imaging: Image and Video Processing*, volume 2421, pages 173–180, 1995.

[17] M.D. Grossberg and S.K. Nayar. Determining the camera response from images: What is knowable. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1455–1467, November 2003.

[18] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A.K. Jain. A background model initialization algorithm for video surveillance. In *Proceedings of the* 8^{th} *International Conference on Computer Vision (ICCV01)*, volume 1, pages 733–740, July 2001.

[19] I. Haritaoglu, D. Harwood, and L.S. Davis. A fast background scene modeling and maintenance for outdoor surveillance. In *Proceedings of the* 15^{th} *International Conference on Pattern Recognition (ICPR00)*, volume 4, pages 179–183, September 2000.

[20] G. E. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, March 1994.

[21] Y. A. Ivanov, A. F. Bobik, and J. Liu. Fast lighting independent background subtraction. *International Journal of Computer Vision*, 37(2):199–207, 2000.

[22] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proceedings of* 2^{nd} *European Workshop on Adavanced Video Based Surveillance Systems (AVBS01)*, September 2001.

[23] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proceedings of 2006 European Conference on Computer Vision (ECCV'06)*, 2006.

[24] A. Lanza and L. Di Stefano. Detecting changes in grey level sequences by ml isotonic regression. In *Proceedings of 2006 IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS'06)*, 2005.

[25] S. N. Lim, A. Mittal, L. S. Davis, and N. Paragios. Fast illumination-invariant background subtraction using two-views: Error analysis, sensor placement and applications. In *Proceedings of 2005 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.

[26] A.J. Lipton, M. Allmen, N. Haering, W. Severson, and T. Strat. Video segmentation using statistical pixel modeling. In *US Patent 20020159634*, October 2002.

[27] A.J. Lipton and N. Haering. Commode: An algorithm for video background modeling and object segmentation. In *Proceedings of the* 7^{th} *International Conference on Control, Automation, Robotics and Vision (ICARCV02)*, volume 3, pages 1603–1608, December 2002.

[28] W. Long and Y.H. Yang. Stationary background generation: An alternative to the difference of two images. *Pattern Recognition*, 23(12):1351–1359, 1990.

[29] S. Mann and R. Mann. Quantigraphic imaging: Estimating the camera response and exposures from differently exposed images. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR01)*, volume 1, pages 842–849, December 2001.

[30] D.R. Mendoza. An analysis of CCD camera noise and its effect on pressure sensitive paint instrumentation system signal-to-noise ratio. In *Proceedings of the*

1997 International Congress on Instrumentation in Aerospace Simulation Facilities (ICIASF97), pages 22–29, October 1997.

[31] T. Mitsunaga and S.K. Nayar. Radiometric self calibration. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR99)*, volume 1, pages 374–380, June 1999.

[32] N. Ohta. A statistical approach to background subtraction for surveillance systems. In *Proceedings of the 2001 IEEE International Conference on Computer Vision (ICCV'01)*, volume 2, pages 481–486, July 2001.

[33] J. Pan, C.W. Lin, C. Gu, and M.T. Sun. A robust video object segmentation scheme with prestored background information. In *Proceedings of the 2002 IEEE International Symposium on Circuits and Systems (ISCAS02)*, pages 803–806, May 2002.

[34] B. Phong. Illumination for computer-generated images. *Comm. ACM*, 18(6):311–317, 1975.

[35] E. Rivlin, M. Rudzsky, R. Goldenberg, U. Bogomolov, and S. Lapchev. A real-time system for classification of moving objects. In *Proceedings of the* 16^{th} *International Conference on Pattern Recognition (ICPR02)*, volume 3, pages 688–691, August 2002.

[36] A. Shio and J. Sklansky. Segmentation of people in motion. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 325–332, October 1991.

[37] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of 1999 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR99)*, volume 2, pages 246–252, June 1999.

[38] Y. Tsin, V. Ramesh, and T. Kanade. Statistical calibration of CCD imaging process. In *Proceedings of the* 8^{th} *International Conference on Computer Vision (ICCV01)*, volume 1, pages 480–487, July 2001.

[39] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: Real-time tracking of the human body. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 780–785, July 1997.

[40] B. Xie, V. Ramesh, and T. Boult. Sudden illumination change detection using order consistency. *Image and Vision Computing*, 22(2):117–125, February 2004.

[41] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the 3rd European Conference on Computer Vision (ECCV'94)*, 1994.