Université de Liège
Faculté des Sciences Appliquées
Département d'Électricité, Électronique et Informatique
Institut Montefiore

Motion detection and human recognition in video sequences

Olivier Barnich

Thèse de doctorat présentée en vue de l'obtention du grade de
Docteur en Sciences de l'Ingénieur (électricité et électronique)

Année Académique 2009-2010

Marc	Van Droogenbroeck	*Promoteur*
Jean-François	Delaigle	*Examinateur*
Antoine	Manzanera	*Examinateur*
Wilfried	Philips	*Examinateur*
Justus	Piater	*Examinateur*
Louis	Wehenkel	*Examinateur*
Jacques	Destiné	*Président*

Abstract

This thesis is concerned with the design of a complete framework that allows the real-time recognition of humans in a video stream acquired by a static camera. For each stage of the processing chain, which takes as input the raw images of the stream and eventually outputs the identity of the persons, we propose an original algorithm. The first algorithm is a background subtraction technique named ViBe. The purpose of ViBe is to detect the parts of the images that contain moving objects. The second algorithm determines which moving objects correspond to individuals. The third algorithm allows the recognition of the detected individuals from their gait.

Our background subtraction algorithm, ViBe, uses a collection of samples to model the history of each pixel. The current value of a pixel is classified by comparison with the closest samples that belong to the collection. We propose an innovative model update technique which allows to obtain an appropriate modeling of the history of the pixel with a reduced number of samples. Furthermore, our model update policy ensures the spatial coherence of the background model and enables the use of a background model composed exclusively of background samples. We show that ViBe outperforms other state-of-the-art techniques while being faster than most of them. Our algorithm is actually fast enough to run in real-time on the low speed processor of a Canon camera.

We then introduce an algorithm that processes the silhouettes of the moving objects detected by ViBe. The purpose of this algorithm is to detect and locate humans. The silhouettes of the moving objects are classified as being either *human* or *non-human*. This classification is performed on the basis of the *cover by rectangles* of the silhouettes, which is a new morphological operator introduced in this thesis. Rectangles from the cover by rectangles of a silhouette are classified individually. The silhouette is then classified according to a majority vote policy among its rectangles. We show that the detection of the persons is robust and can be computed in real-time.

The last stage of the processing chain recovers the identity of the detected persons from their gait. Our gait recognition technique is a logical follow-up to our person detection algorithm. This algorithm is based on the classification of a gait signature computed from the covers by rectangles of the silhouettes of a walker. The purpose of a gait signature is to capture the dynamics of the time series of the silhouettes of a walking person. Experiments on a public database show that the results of our gait recognition technique are on par with those of other techniques described in the literature. In the last part of this thesis, we apply our gait recognition algorithm to an original application: the intelligent control of access to a secure area.

Résumé

Cette thèse porte sur la conception d'un système complet permettant la reconnaissance en temps réel de personnes dans des séquences vidéo acquises par une caméra fixe. Pour chaque étape de la chaîne de traitement allant des images fournies par la caméra jusqu'à l'identité de la personne filmée, nous proposons un algorithme original. Nous introduisons ainsi trois algorithmes. Le premier est une technique de soustraction de l'arrière-plan dont l'objectif est d'identifier les zones de l'image qui correspondent à des objets en mouvement. Le second permet de déterminer parmi les objets détectés par le premier, les zones qui correspondent à des être humains. Le troisième et dernier algorithme a pour but l'identification des personnes ainsi détectées sur base de leur démarche.

Notre algorithme de soustraction de l'arrière-plan, ViBe, est basé sur la modélisation de chaque pixel de l'image par une collection d'échantillons. La valeur de chaque pixel est classifiée en la comparant aux échantillons qui en sont les plus proches parmis ceux contenus dans le modèle du pixel. Nous innovons en proposant une technique originale de mise à jour des modèles de pixel qui permet (1) de limiter le nombre d'échantillons nécessaires pour modéliser fidèlement l'historique d'un pixel, (2) d'assurer une cohérence spatiale à l'ensemble du modèle d'arrière-plan et (3) de rendre viable l'emploi d'un modèle constitué exclusivement d'échantillons ayant été classifiés comme faisant partie de l'arrière-plan. Comparé à l'état de l'art, ViBe produit des résultats plus précis tout en étant plus rapide que la grande majorité des autres techniques. ViBe est même suffisamment rapide pour être exécuté en temps réel dans un appareil photo compact.

Nous présentons ensuite un algorithme capable de détecter et de localiser les personnes. Cet algorithme détermine, parmi les silhouettes d'objets en mouvement détectées par ViBe, quelles sont les silhouettes qui correspondent à des personnes. Pour cela, l'algorithme classifie les silhouettes sur base d'une série de rectangles extraits aléatoirement de leur *recouvrement par des rectangles*, un nouvel opérateur morphologique présenté dans cette thèse. Cette classification est effectuée rectangle par rectangle et la silhouette se voit attribuer la classe ayant collecté la majorité des votes parmi les rectangles. Nous montrons que la détection des personnes est robuste et suffisamment rapide pour être effectuée en temps réel.

La dernière étape de la chaîne de traitement consiste à identifier les personnes détectées sur base de leur démarche. Pour cela, nous proposons un troisième algorithme qui constitue une suite logique à notre algorithme de détection de personnes. Cet algorithme repose sur la classification de signatures de démarche. Ces signatures sont basées sur le recouvrement par des rectangles des silhouettes binaires de l'individu. Elles synthétisent l'information présente dans la suite temporelle des silhouettes d'un individu qui marche. Testé sur une base de données publique, notre algorithme de reconnaissance de personnes obtient des résultats comparables à ceux des techniques de l'état de l'art. Dans la dernière partie de cette thèse, nous effectuons une application originale de cet algorithme dans le cadre de la conception d'un système intelligent de contrôle d'accès à une zone sécurisée.

Acknowledgments

Merci à Marc Van Droogenbroeck pour m'avoir donné l'occasion de réaliser cette thèse et pour le soutien et la confiance qu'il m'a accordés durant toutes ces années.

Merci à l'ensemble des membres du jury de cette thèse pour avoir accepté de l'évaluer.

Merci au FNRS/FRS, à la Région Wallonne, au Patrimoine de l'Université et à BEA pour le soutien financier qu'il m'ont accordé.

Merci à Sébastien Jodogne qui m'a gracieusement laissé employer son implémentation des extra-trees.

Merci à Nicole Antheunis pour son aide précieuse tout au long du processus de dépôt du brevet concernant ViBe.

Merci à Marie-Thérèse et Maxime pour avoir accepté de relire des parties de ce document.

Merci à Thomas pour ses suggestions.

Merci aux membres anciens ou actuels de l'Institut Montefiore et de l'Université pour les moments passés ensemble. Je pense notamment, mais pas seulement, à Caroline, Charline, David, Florence, Jean-François, Julien, Lionel, Livia, Minh, Philippe, Raphaël, Renaud, Sébastien, Sylvain et Vincent.

Merci à mes parents, ma famille et mes amis pour leur présence et leurs encouragements.

Enfin, et surtout, merci à Anne-Cécile.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, our objective was to design a complete framework able to automatically detect and recognize humans in video sequences (see Figure 1.1).

The first problem we tackled was to get an answer to the following questions: "Is someone present in the field of view of the camera? And where?". At the time, the most famous techniques we found in the literature answered this question by processing the texture patterns of each frame of the video stream, taken individually [26, 92, 119]. But we wanted to design a real-time method able to deal with any surrounding environment, disregarding the individuals' appearance (clothes, ...). To do so, we decided to restrict ourselves to static cameras and to focus on an analysis of the binary silhouettes produced by a background subtraction algorithm. The method we designed relied on a surfacic morphological analysis of the silhouettes. The purpose was to extract a set of discriminant features fed into a machine learning algorithm which classified them into one of these two categories: "human" and "non-human".

We were then able to tell if someone was present in the scene. The next objective was to recognize the individual. At the time, we had just installed surveillance network cameras in the hallways of the Montefiore Institute. Obviously, it was appealing to try to recognize people using the video streams provided by these cameras. As it turned out, the image resolution of these cameras was not high enough to use a frontal face recognition technique as in most of the systems described in the literature. Instead, we turned towards human recognition from gait. On top of complete non-intrusiveness, the principal advantage of gait recognition over frontal face recognition is its improved robustness against poor imaging conditions. Moreover, due to the arrangement of our cameras, we had to design a technique able to recognize individuals from their gait using a front view camera, which is rather uncommon in the literature. The technique we designed was a logical follow-up to our person detection algorithm. From the time series of binary silhouettes of the walkers, we computed an inter-frame gait signature that is an aggregation of intra-frame statistics of morphological measures. The gait signature was then classified using the machine learning algorithm that we used previously to detect persons. The output class labels of the learning algorithm identified the walker. We tested the method on a frontal database of 21 persons and were able to recognize walkers in up to 97% of the cases. We also tested our algorithm on a public multi-view database of 25 persons. Using the front view only, we got ratios of correct classifications ranging from 96 to 100%. These results are on par with those of the techniques described in the recent literature that were tested on the same database.

All our experiments related to silhouette interpretation led to the conclusion that the performance of our classification tools was very dependent on the quality of the binary silhouettes fed into the silhouette analysis algorithms. Until then, we were using one of the best, and by far most popular, background subtraction algorithms described in the literature: the gaussian mixture model (GMM). But we weren't satisfied with the behavior of the GMM algorithm, especially on the noisy video streams provided by real surveillance cameras such as the one installed in the hallways of our institute. This convinced us that despite the abundant literature on the subject, there was still room for improvement in the field of background subtraction. We then decided to

design a new algorithm from the ground up. While most of the existing techniques modeled the history of each pixel with a parametric statistical model, our intuition was that the modeling of the history of a pixel should be non-parametric and based on samples instead. Such techniques had already been described in the literature. However, all of them used a first-in first-out policy to update their collection of samples. But why discard a relevant sample on the sole basis of it's "age"? Instead, we designed a simple policy to ensure smooth decaying lifespans for the samples. Our background subtraction algorithm, called ViBe, combines this policy with a method that brings a global spatial consistency to its model and enables the use of strict conservative update policy. The comparison of the results of ViBe with those of existing algorithms established the fact that, to our knowledge, ViBe outperforms all the existing techniques while being faster than the vast majority of them. As a matter of fact, ViBe is fast enough for embedded applications: our implementation processes six frames per second on the low speed processor of a Canon digital camera. The accuracy and the speed of ViBe pushed us to apply, with success, for a patent.

After the publication of our work on gait recognition, we initiated a collaboration with BEA, a Belgian company active in the design and manufacturing of devices for people and vehicle detection. The purpose of our collaboration was to incorporate our gait recognition technique in an intelligent access control system embedded in a revolving door. It turned out to be an ambitious project as we had to give up on the use of regular cameras to recover the time series of the silhouettes of the walkers. Instead, we used the distance measures captured by rotating laser scanners manufactured by BEA to reconstruct the silhouette of the persons walking through the door. Because of inevitable occlusions and shadowing phenomena, these reconstructed silhouettes came with many artifacts and turned out to be a true challenge for our recognition algorithms. Nevertheless, we fulfilled the objective and two real-time demonstrators were successfully built. The first was installed in our lab, and the second at BEA.

Generally speaking, our algorithms carry out simple and intuitive strategies for complex problems that involve a large part of uncertainty at decision time. We believe that one of the keys to get simple, yet effective, solutions to such problems is to isolate the uncertainty part to determined points where the best decision to take is a random decision. Using these strategies we showed, among other things, that the problem of background subtraction can be efficiently handled with a simple algorithm that reduces the amount of required memory and computations to the absolute minimum.

In this manuscript, we present our algorithms in the order of the processing chain, which is shown in Figure 1.1. Chapter 2 is devoted to our background subtraction algorithm. Our work on silhouette classification for the detection of humans is described in Chapter 3. In Chapter 4, we detail our gait recognition algorithm and its application to an intelligent security system. Our conclusions are given in Chapter 5.

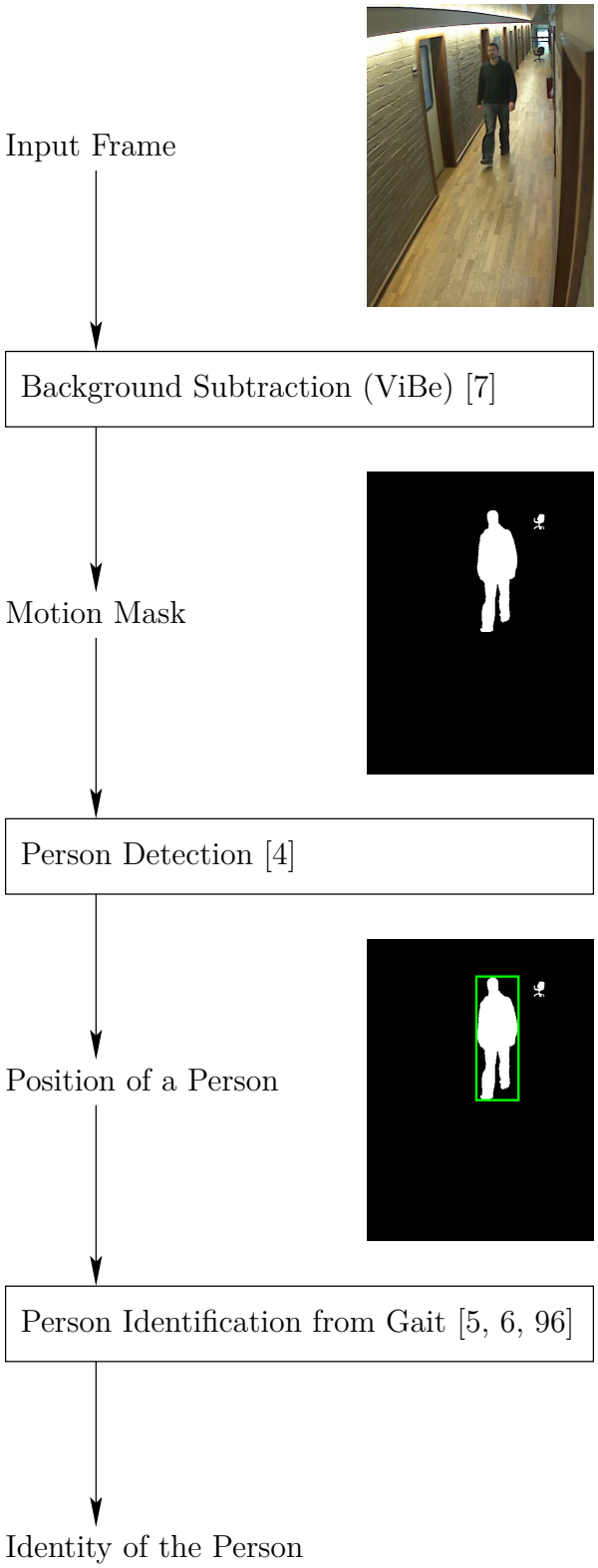


Figure 1.1: Framework of the thesis.

Chapter 2

ViBe: A universal background subtraction algorithm for video sequences

In this chapter, we describe a new algorithm for background subtraction. This algorithm detects motion in video streams captured by a static camera. In our experiments, we compare our technique with other state-of-the-art methods. Therefore, we are grateful to Antoine Manzanera, who provided us with the source code of his algorithms, and to Zoran Zivkovic for publishing his code on the Internet.

Abstract

This chapter presents a technique for motion detection that incorporates several innovative mechanisms. For example, our proposed technique stores, for each pixel, a set of values taken in the past at the same location or in the neighborhood. It then compares this set to the current pixel value in order to determine whether that pixel belongs to the background, and adapts the model by choosing randomly which values to substitute from the background model. This approach differs from those based on the classical belief that the oldest values should be replaced first. Finally, when the pixel is found to be part of the background, its value is propagated into the background model of a neighboring pixel.

We describe our method in full details (including the parameter values used) and compare it to other background subtraction techniques. Efficiency figures show that our method outperforms recent and proven state-of-the-art methods in terms of both computation speed and detection rate. We also analyze the performance of a downscaled version of our algorithm to the absolute minimum of one comparison and one byte of memory per pixel. It appears that even such a simplified version of our algorithm performs better than mainstream techniques.

2.1 Introduction

The number of cameras available worldwide has increased dramatically over the last decade. But this growth has resulted in a huge augmentation of data, meaning that the data are impossible either to store or to handle manually. In order to detect, segment, and track objects automatically in videos, several approaches are possible. Simple motion detection algorithms compare a static background frame with the current frame of a video scene, pixel by pixel. This is the basic principle of background subtraction, which can be formulated as a technique that builds a model of a background and compares this model with the current frame in order to detect zones where a significant difference occurs. The purpose of a background subtraction algorithm is therefore to distinguish moving objects (hereafter referred to as the *foreground*) from static, or slow moving, parts of the scene (called *background*). Note that when a static object starts moving, a background subtraction algorithm detects the object in motion as well as a hole left behind in the background (referred to as a *ghost*). Clearly a ghost is irrelevant for motion interpretation and has to be discarded. An alternative definition for the background is that it corresponds to a reference frame with values visible most of the time, that is with the highest appearance probability, but this kind of framework is not straightforward to use in practice.

While a static background model might be appropriate for analyzing short video sequences in a constrained indoor environment, the model is ineffective for most practical situations; a more sophisticated model is therefore required. Moreover, the detection of motion is often only a first step in the process of understanding the scene. For example, zones where motion is detected might be filtered and characterized for the detection of unattended bags, gait recognition, face detection, people counting, traffic surveillance, etc. The diversity of scene backgrounds and applications explains why countless papers discuss issues related to background subtraction.

In this chapter, we present a universal method for background subtraction. This method has been briefly described in [7] and in a patent [116]. In Section 2.2, we extensively review the literature of background subtraction algorithms. This review presents the major frameworks developed for background subtraction and highlights their respective advantages. We have implemented some of these algorithms in order to compare them with our method. Section 2.3 describes our technique and details our major innovations: the background model, the initialization process, and the update mechanism. Section 2.4 discusses experimental results including comparisons with other state-of-the-art algorithms and computational performance. We also present a simplified version of our algorithm which requires only one comparison and one byte of memory per pixel; this is the absolute minimum in terms of comparisons and memory for any background subtraction technique. We show that, even in its simplified form, our algorithm performs better than more sophisticated techniques. Section 2.5 concludes the chapter. A C-like pseudo-code of ViBe is given in Section 2.6.

2.2 Review of background subtraction algorithms

The problem tackled by background subtraction techniques involves the comparison of an observed image with an estimated image that does not contain any object of interest; this is referred to as the background model (or background image) [80]. This comparison process, called *foreground detection*, divides the observed image into two complementary sets of pixels that cover the entire image: (1) the foreground that contains the objects of interest, and (2) the background, its complementary set. As stated in [98], it is difficult to specify a gold-standard definition of what a background subtraction technique should detect as a foreground region, as the definition of foreground objects relates to the application level.

Many background subtraction techniques have been proposed with as many models and segmentation strategies, and several surveys are devoted to this topic (see for example [9, 13, 33, 80, 94, 95, 98]). Some algorithms focus on specific requirements that an ideal background subtraction technique could or should fulfill. According to [95], a background subtraction technique must adapt to gradual or fast illumination changes (changing time of day, clouds, etc), motion

changes (camera oscillations), high frequency background objects (e.g. tree leave or branches), and changes in the background geometry (e.g. parked cars). Some applications require background subtraction algorithms to be embedded in the camera, so that the computational load becomes the major concern. For the surveillance of outdoor scenes, robustness against noise and adaptivity to illumination changes are also essential.

Most techniques described in the literature operate on each pixel independently. These techniques relegate entirely to post-processing algorithms the task of adding some form of spatial consistency to their results. Since perturbations often affect individual pixels, this results in local misclassifications. By contrast, the method described by Seiki *et al.* in [101] is based on the assumption that neighboring blocks of background pixels should follow similar variations over time. While this assumption holds most of the time, especially for pixels belonging to the same background object, it becomes problematic for neighboring pixels located at the border of multiple background objects. Despite this inconvenience, pixels are aggregated into blocks and each $N \times N$ block is processed as an N^2-component vector. A few samples are then collected over time and used to train a Principal Component Analysis (PCA) model for each block. A block of a new video frame is classified as background if its observed image pattern is close to its reconstructions using PCA projection coefficients of 8-neighboring blocks. Such a technique is also described in [97], but it lacks an update mechanism to adapt the block models over time. In [90], the authors focus on the PCA reconstruction error. While the PCA model is also trained with time samples, the resulting model accounts for the whole image. Individual pixels are classified as background or foreground using simple image difference thresholding between the current image and the back-projection in the image space of its PCA coefficients. As for other PCA-based methods, the initialization process and the update mechanism are not described.

A similar approach, the Independent Component Analysis (ICA) of serialized images from a training sequence, is described in [114] in the training of an ICA model. The resulting de-mixing vector is then computed and compared to that of a new image in order to separate the foreground from a reference background image. The method is said to be highly robust to indoor illumination changes.

A two-level mechanism based on a classifier is introduced in [65]. A classifier first determines whether an image block belongs to the background. Appropriate blockwise updates of the background image are then carried out in the second stage, depending on the results of the classification. Classification algorithms are also the basis of other algorithms, as in the one provided in [71], where the background model learns its motion patterns by self organization through artificial neural networks.

Algorithms based on the framework of compressive sensing perform background subtraction by learning and adapting a low dimensional compressed representation of the background [17]. The major advantage of this approach lies in the fact that compressive sensing estimates object silhouettes without any auxiliary image reconstruction. On the other hand, objects in the foreground need to occupy only a small portion of the camera view in order to be detected correctly.

Background subtraction is considered to be a sparse error recovery problem in [29]. These authors assumed that each color channel in the video can be independently modeled as the linear combination of the same color channel from other video frames. Consequently, the method they proposed is able to accurately compensate for global changes in the illumination sources without altering the general structure of the frame composition by finding appropriate scalings for each color channel separately.

Background estimation is formulated in [22] as an optimal labeling problem in which each pixel of the background image is labeled with a frame number, indicating which color from the past must be copied. The author's proposed algorithm produces a background image, which is constructed by copying areas from the input frames. Impressive results are shown for static backgrounds but the method is not designed to cope with objects moving slowly in the background, as its outcome is a single static background frame.

The authors of [72] were inspired by the biological mechanism of motion-based perceptual grouping. They propose a spatio-temporal saliency algorithm applicable to scenes with highly dynamic backgrounds, which can be used to perform background subtraction. Comparisons of their

algorithm with other state-of-the-art techniques show that their algorithm reduces the average error rate, but at a cost of a prohibitive processing time (several seconds per frame), which makes it unsuitable for real-time applications.

Pixel-based background subtraction techniques compensate for the lack of spatial consistency by a constant updating of their model parameters. The simplest techniques in this category are the use of a static background frame (which has recently been used in [106]), the (weighted) running average [16], first-order low-pass filtering [30], temporal median filtering [1, 105], and the modeling of each pixel with a gaussian [18, 27, 127].

Probabilistic methods predict the short-term evolution of a background frame with a Wiener [113] or a Kalman [57] filter. In [113], a frame-level component is added to the pixel-level operations. Its purpose is to detect sudden and global changes in the image and to adapt the background frame accordingly. Median and gaussian models can be combined to allow inliers (with respect to the median) to have more weight than outliers during the gaussian modeling, as in [28] or [51]. A method for properly initializing a gaussian background model from a video sequence in which moving objects are present is proposed in [39].

The W^4 model presented in [42] is a rather simple but nevertheless effective method. It uses three values to represent each pixel in the background image: the minimum and maximum intensity values, and the maximum intensity difference between consecutive images of the training sequence. The authors of [48] bring a small improvement to the W^4 model together with the incorporation of a technique for shadow detection and removal.

Methods based on $\Sigma - \Delta$ (sigma-delta) motion detection filters [73, 74, 75] are popular for embedded processing [58, 59]. As in the case of analog-to-digital converters, a $\Sigma - \Delta$ motion detection filter consists of a simple non-linear recursive approximation of the background image, which is based on comparison and on an elementary increment/decrement (usually -1, 0, and 1 are the only possible updating values). The $\Sigma - \Delta$ motion detection filter is therefore well suited to many embedded systems that lack a floating point unit.

All these unimodal techniques can lead to satisfactory results in controlled environments while remaining fast, easy to implement, and simple. However, more sophisticated methods are necessary when dealing with videos captured in complex environments where moving background, camera egomotion, and high sensor noise are encountered [9].

Over the years, increasingly complex pixel-level algorithms have been proposed. Among these, by far the most popular is the Gaussian Mixture Model (GMM) [109, 110]. First presented in [109], this model consists of modeling the distribution of the values observed over time at each pixel by a weighted mixture of gaussians. This background pixel model is able to cope with the multimodal nature of many practical situations and leads to good results when repetitive background motions, such as tree leaves or branches, are encountered. Since its introduction, the model has gained vastly in popularity among the computer vision community [63, 95, 97, 98, 124, 125], and it is still raising a lot of interest as authors continue to revisit the method and propose enhanced algorithms [4, 43, 52, 61, 132, 136]. In [126], a particle swarm optimization method is proposed to automatically determine the parameters of the GMM algorithm. The authors of [117] combine a GMM model with a region-based algorithm based on color histograms and texture information. In their experiments, the authors' method outperform the original GMM algorithm. However, the authors' technique has a considerable computational cost as they only manage to process seven frames of 640×480 pixels per second with an Intel Xeon 5150 processor.

The downside of the GMM algorithm resides in its strong assumptions that the background is more frequently visible than the foreground and that its variance is significantly lower. None of this is valid for every time window. Furthermore, if high- and low-frequency changes are present in the background, its sensitivity cannot be accurately tuned and the model may adapt to the targets themselves or miss the detection of some high speed targets, as detailed in [32]. Also, the estimation of the parameters of the model (especially the variance) can become problematic in real-world noisy environments. This often leaves one with no other choice than to use a fixed variance in a hardware implementation. Finally, it should be noted that the statistical relevance of a gaussian model is debatable as some authors claim that natural images exhibit non-gaussian statistics [108].

To avoid the difficult question of finding an appropriate shape for the probability density function, some authors have turned their attention to non-parametric methods to model background distributions. One of the strengths of non-parametric kernel density estimation methods [31, 32, 82, 104, 112, 137] is their ability to circumvent a part of the delicate parameter estimation step due to the fact that they rely on pixel values observed in the past. For each pixel, these methods build a histogram of background values by accumulating a set of real values sampled from the pixel's recent history. These methods then estimate the probability density function with this histogram to determine whether or not a pixel value of the current frame belongs to the background. Non-parametric kernel density estimation methods can provide fast responses to high-frequency events in the background by directly including newly observed values in the pixel model. However, the ability of these methods to successfully handle concomitant events evolving at various speeds is questionable since they update their pixel models in a first-in first-out manner. This has led some authors to represent background values with two series of values or models: a short term model and a long term model [32, 83]. While this can be a convenient solution for some situations, it leaves open the question of how to determine the proper time interval. In practical terms, handling two models increases the difficulty of fine-tuning the values of the underlying parameters. Our method incorporates a smoother lifespan policy for the sampled values, and as explained in Section 2.4, it improves the overall detection quality significantly.

In the codebook algorithm [55, 56], each pixel is represented by a codebook, which is a compressed form of background model for a long image sequence. Each codebook is composed of codewords comprising colors transformed by an innovative color distortion metric. An improved codebook incorporating the spatial and temporal context of each pixel has been proposed in [128]. Codebooks are believed to be able to capture background motion over a long period of time with a limited amount of memory. Therefore, codebooks are learned from a typically long training sequence and a codebook update mechanism is described in [56] allowing the algorithm to evolve with the lighting conditions once the training phase is over. However, one should note that the proposed codebook update mechanism does not allow the creation of new codewords, and this can be problematic if permanent structural changes occur in the background (for example, in the case of newly freed parking spots in urban outdoor scenes).

Instead of choosing a particular form of background density model, the authors of [120, 121] use the notion of "consensus". They keep a cache of a given number of last observed background values for each pixel and classify a new value as background if it matches most of the values stored in the pixel's model. One might expect that such an approach would avoid the issues related to deviations from an arbitrarily assumed density model, but since values of pixel models are replaced according to a first-in first-out update policy, they are also prone to the problems discussed previously, for example, the problem of slow and fast motions in the background, unless a large number of pixel samples are stored. The authors state that a cache of 20 samples is the minimum required for the method to be useful, but they also noticed no significant further improvement for caches with more than 60 samples. Consequently, the training period for their algorithm must comprise at least 20 frames. Finally, to cope with lighting changes and objects appearing or fading in the background, two additional mechanisms (one at the pixel level, a second at the blob level) are added to the consensus algorithm to handle entire objects.

The method proposed in this chapter operates differently in handling new or fading objects in the background, without the need to take account of them explicitly. In addition to being faster, our method exhibits an interesting asymmetry in that a ghost (a region of the background discovered once a static object starts moving) is added to the background model more quickly than an object that stops moving. Another major contribution of this chapter resides in the proposed update policy. The underlying idea is to gather samples from the past and to update the sample values by ignoring when they were added to the models. This policy ensures a smooth exponential decaying lifespan for the sample values of the pixel models and allows our technique to deal with concomitant events evolving at various speeds with a unique model of a reasonable size for each pixel.

2.3 Description of a universal background subtraction technique: ViBe

Background subtraction techniques have to deal with at least three considerations in order to be successful in real applications: (1) what is the model and how does it behave?, (2) how is the model initialized?, and (3) how is the model updated over time? Answers to these questions are given in the three subsections of this section. Most papers describe the intrinsic model and the updating mechanism. Only a minority of papers discuss initialization, which is critical when a fast response is expected, as in the case inside a digital camera. In addition, there is often a lack of coherence between the model and the update mechanism. For example, some techniques compare the current value of a pixel p to that of a model b with a given tolerance T. They consider that there is a good match if the absolute difference between p and b is lower than T. To be adaptive over time, T is adjusted with respect to the statistical variance of p. But the statistical variance is estimated by a temporal average. Therefore, the adjustment speed is dependent on the acquisition framerate and on the number of background pixels. This is inappropriate in some cases, as in the case of remote IP cameras whose framerate is determined by the available bandwidth.

We detail below a background subtraction technique, called "ViBe" (for "VIsual Background Extractor"). For convenience, we present a complete version of our algorithm in a C-like code in Section 2.6.

2.3.1 Pixel model and classification process

To some extent, there is no way around the determination, for a given color space, of a probability density function (pdf) for every background pixel or at least the determination of statistical parameters, such as the mean or the variance. Note that with a gaussian model, there is no distinction to be made as the knowledge of the mean and variance is sufficient to determine the pdf. While the classical approaches to background subtraction and most mainstream techniques rely on pdfs or statistical parameters, the question of their statistical significance is rarely discussed, if not simply ignored. In fact, there is no imperative to compute the pdf as long as the goal of reaching a relevant background segmentation is achieved. An alternative is to consider that one should enhance statistical significance over time, and one way to proceed is to build a model with real observed pixel values. The underlying assumption is that this makes more sense from a stochastic point of view, as already observed values should have a higher probability of being observed again than would values not yet encountered.

Like the authors of [121], we do not opt for a particular form for the pdf, as deviations from the assumed pdf model are ubiquitous. Furthermore, the evaluation of the pdf is a global process and the shape of a pdf is sensitive to outliers. In addition, the estimation of the pdf raises the non-obvious question regarding the number of samples to be considered; the problem of selecting a representative number of samples is intrinsic to all the estimation processes.

If we see the problem of background subtraction as a classification problem, we want to classify a new pixel value with respect to its immediate neighborhood in the chosen color space, so as to avoid the effect of any outliers. This motivates us to model each background pixel with a set of samples instead of with an explicit pixel model. Consequently no estimation of the pdf of the background pixel is performed, and so the current value of the pixel is compared to its closest samples within the collection of samples. This is an important difference in comparison with existing algorithms, in particular with those of consensus-based techniques. A new value is compared to background samples and should be close to some of the sample values instead of the majority of all values. The underlying idea is that it is more reliable to estimate the statistical distribution of a background pixel with a small number of close values than with a large number of samples. This is somewhat similar to ignoring the extremities of the pdf, or to considering only the central part of the underlying pdf by thresholding it. On the other hand, if one trusts the values of the model, it is crucial to select background pixel samples carefully. The classification of pixels in the background therefore needs to be conservative, in the sense that only background

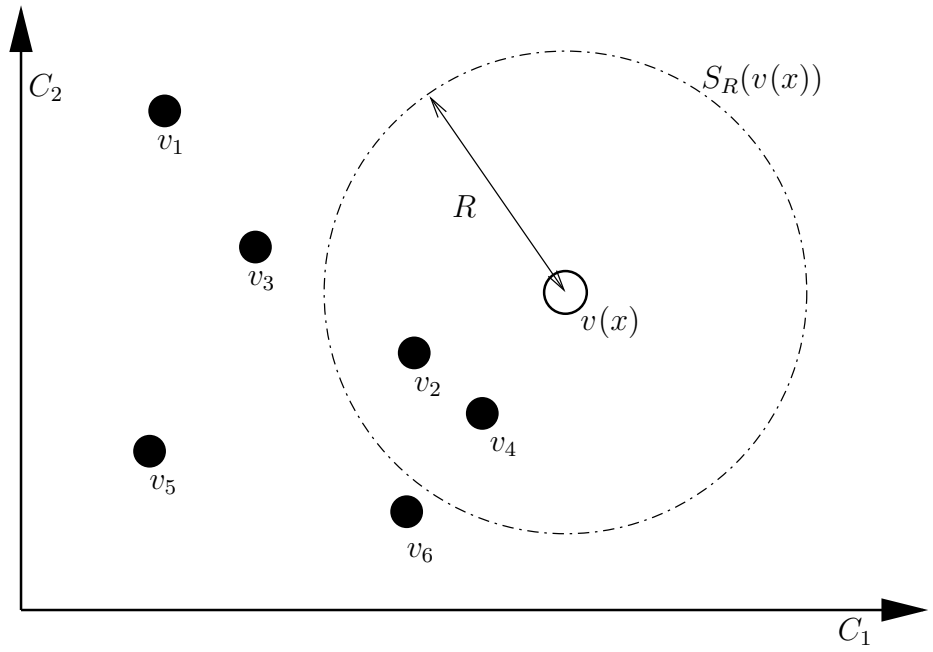


Figure 2.1: Comparison of a pixel value with a set of samples in a two dimensional Euclidean color space (C_1, C_2). To classify $v(x)$, we count the number of samples of $\mathcal{M}(x)$ intersecting the sphere of radius R centered on $v(x)$.

pixels should populate the background models.

Formally, let us denote by $v(x)$ the value in a given Euclidean color space taken by the pixel located at x in the image, and by v_i a background sample value with an index i. Each background pixel x is modeled by a collection of N background sample values

$$\mathcal{M}(x) = \{v_1, v_2, \ldots, v_N\} \tag{2.1}$$

taken in previous frames. For now, we ignore the notion of time; this is discussed later.

To classify a pixel value $v(x)$ according to its corresponding model $\mathcal{M}(x)$, we compare it to the *closest* values within the set of samples by defining a sphere $S_R(v(x))$ of radius R centered on $v(x)$. The pixel value $v(x)$ is then classified as background if the cardinality, denoted $\sharp$, of the set intersection of this sphere and the collection of model samples $\mathcal{M}(x)$ is larger than or equal to a given threshold $\sharp_{\min}$. More formally, we compare $\sharp_{\min}$ to

$$\sharp\{S_R(v(x)) \cap \{v_1, v_2, \ldots, v_N\}\}. \tag{2.2}$$

According to equation 2.2, the classification of a pixel value $v(x)$ involves the computation of N distances between $v(x)$ and model samples, and of N comparison with a thresholded Euclidean distance R. This process is illustrated in Figure 2.1. Note that, as we are only interested in searching for a few matches, the segmentation process of a pixel can be stopped once $\sharp_{\min}$ matches have been found.

As can easily be seen, the accuracy of our model is determined by two parameters only: the radius R of the sphere and the minimal cardinality $\sharp_{\min}$. Experiments have shown that a unique radius R of 20 (for monochromatic images) and a cardinality of 2 are appropriate (see Section 2.4.1 for a thorough discussion on parameter values). There is no need to adapt these parameters during the background subtraction nor do we need to change them for different pixel locations within the image. Note that since the number of samples N and $\sharp_{\min}$ are chosen to be fixed and since they impact on the same decision, the sensitivity of the model can be adjusted using the following ratio

$$\frac{\sharp_{\min}}{N}, \tag{2.3}$$

but in all our comparative tests we kept these values unchanged.

So far we have detailed the nature of the model. In the coming sections, we explain how to initialize the model from a single frame and how to update it over time.

2.3.2 Background model initialization from a single frame

Many popular techniques described in the literature, such as [32], [56], and [121], need a sequence of several dozens of frames to initialize their models. Such an approach makes sense from a statistical point of view as it seems imperative to gather a significant amount of data in order to estimate the temporal distribution of the background pixels. But one may wish to segment the foreground of a sequence that is even shorter than the typical initialization sequence required by some background subtraction algorithms. Furthermore, many applications require the ability to provide an *uninterrupted* foreground detection, even in the presence of sudden light changes, which cannot be handled properly by the regular update mechanism of the algorithm. A possible solution to both these issues is to provide a specific model update process that tunes the pixel models to new lighting conditions. But the use of such a dedicated update process is at best delicate, since a sudden illumination may completely alter the chromatic properties of the background.

A more convenient solution is to provide a technique that will initialize the background model from a single frame. Given such a technique, the response to sudden illumination changes is straightforward: the existing background model is discarded and a new model is initialized instantaneously. Furthermore, being able to provide a reliable foreground segmentation as early on as the second frame of a sequence has obvious benefits for short sequences in video-surveillance or for devices that embed a motion detection algorithm.

Since there is no temporal information in a single frame, we use the same assumption as the authors of [50], which is that neighboring pixels share a similar temporal distribution. This justifies the fact that we populate the pixel models with values found in the spatial neighborhood of each pixel. More precisely, we fill them with values randomly taken in their neighborhood in the first frame. The size of the neighborhood needs to be chosen so that it is large enough to comprise a sufficient number of different samples, while keeping in mind that the statistical correlation between values at different locations decreases as the size of the neighborhood increases. From our experiments, selecting samples randomly in the 8-connected neighborhood of each pixel has proved to be satisfactory for images of 640×480 pixels.

Formally, we assume that $t = 0$ indexes the first frame and that $N_G(x)$ is a spatial neighborhood of a pixel location x, therefore

$$\mathcal{M}^0(x) = \{v^0(y \,|\, y \in N_G(x))\} \tag{2.4}$$

where locations y are chosen randomly according to a uniform law. Note that it is possible for a given $v^0(y)$ to be selected several times (for example if the size of the neighborhood is smaller than the cardinality of $\mathcal{M}^0(x)$) or to not be selected at all. However, this is not an issue if one acknowledges that values in the neighborhood are excellent sample candidates.

This strategy has proved to be successful. The only drawback is that the presence of a moving object in the first frame will introduce an artifact commonly called a *ghost*. According to [105], a ghost is "a set of connected points, detected as in motion but not corresponding to any real moving object". In this particular case, the ghost is caused by the unfortunate initialization of pixel models with samples coming from the moving object. In subsequent frames, the object moves and uncovers the real background, which will be learned progressively through the regular model update process, making the ghost fade over time. Fortunately, as shown in Section 2.4.3, our update process ensures both a fast model recovery in the presence of a ghost and a slow incorporation of real moving objects into the background model.

2.3.3 Updating the background model over time

In this Section, we describe how to continuously update the background model with each new frame. This a crucial step if we want to achieve accurate results over time: the update process

must be able to adapt to lighting changes and to handle new objects that appear in a scene.

2.3.3.1 General discussions on an update mechanism

The classification step of our algorithm compares the current pixel value $v^t(x)$ directly to the samples contained in the background model of the previous frame, $\mathcal{M}^{t-1}(x)$ at time $t-1$. Consequently, the question regarding *which* samples have to be memorized by the model and for *how long* is essential. One can see the model as a background memory or background history, as it is often referred to in the literature. The classical approach to the updating of the background history is to discard and replace old values after a number of frames or after a given period of time (typically about a few seconds); the oldest values are substituted by the new ones. Despite the rationale behind it, this substitution principle is not so obvious, as there is no reason to remove a valid value if it corresponds to a background value.

The question of including or not foreground pixel values in the model is one that is always raised for a background subtraction method based on samples; otherwise the model will not adapt to changing conditions. It boils down to a choice between a conservative and a blind update scheme. Note that kernel-based pdf estimation techniques have a softer approach to updating. They are able to smooth the appearance of a new value by giving it a weight prior to inclusion.

A conservative update policy never includes a sample belonging to a foreground region in the background model. In practice, a pixel sample can be included in the background model only if it has been classified as a background sample. Such a policy seems, at first sight, to be the obvious choice. It actually guarantees a sharp detection of the moving objects, given that they do not share similar colors with the background. Unfortunately, it also leads to deadlock situations and everlasting ghosts: a background sample incorrectly classified as foreground prevents its background pixel model from being updated. This can keep indefinitely the background pixel model from being updated and could cause a permanent misclassification. Unfortunately, many practical scenarios lead to such situations. For example, the location freed by a previously parked car cannot be included in the background model with a purely conservative update scheme, unless a dedicated update mechanism handles such situations.

Blind update is not sensitive to deadlocks: samples are added to the background model whether they have been classified as background or not. The principal drawback of this method is a poor detection of slow moving targets, which are progressively included in the background model. A possible solution consists of using pixel models of a large size, which cover long time windows. But this comes at the price of both an increased memory usage and a higher computational cost. Furthermore, with a first-in first-out model update policy such as those employed in [32] or [121], 300 samples cover a time window of only 10 seconds (at 30 frames per second). A pixel covered by a slowly moving object for more than 10 seconds would still be included in the background model.

Strictly speaking, temporal information is not available when the background is masked. But background subtraction is a spatio-temporal process. In order to improve the technique, we could assume, as we proposed in Section 2.3.2, that neighboring pixels are expected to have a similar temporal distribution. According to this hypothesis, the best strategy is therefore to adopt a conservative update scheme and to exploit spatial information in order to inject information regarding the background evolution into the background pixel models masked locally by the foreground. This process is common in inpainting, where objects are removed and values are taken in the neighborhood to fill holes [23]. In Section 2.3.3.4, we provide a simple but effective method for exploiting spatial information, which enables us to counter most of the drawbacks of a purely conservative update scheme.

Our update method incorporates three important components: (1) a memoryless update policy, which ensures a smooth decaying lifespan for the samples stored in the background pixel models, (2) a random time subsampling to extend the time windows covered by the background pixel models, and (3) a mechanism that propagates background pixel samples spatially to ensure spatial consistency and to allow the adaptation of the background pixel models that are masked by the foreground. These components are described, together with our reasons for using them, in the following three subsections.

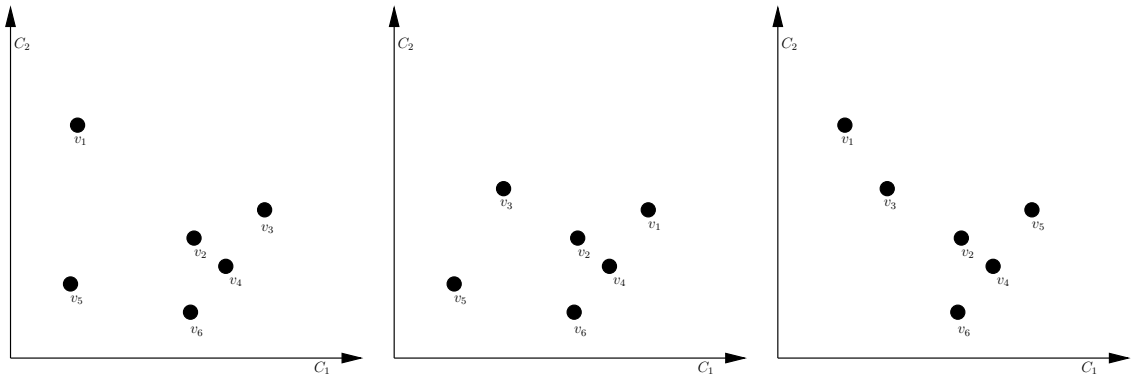


Figure 2.2: 3 of the 6 possible outcomes of the updating of a pixel model of size $N = 6$. We assume that values occupy the same color space as in Figure 2.1 and that we have decided to update the model. This figure shows 3 possible models after the update. The decision process for selecting one particular model is random (with equal probabilities).

2.3.3.2 A memoryless update policy

Many sample-based methods use first-in first-out policies to update their models. In order to deal properly with wide ranges of events in the scene background, Wang *et al.* [121] propose the inclusion of large numbers of samples in pixel models. But as stated earlier, this may still not be sufficient for high framerates. Other authors [32, 137] incorporate two temporal sub-models to handle both fast and slow modifications. This approach proved to be effective. However, it increases the parametrization problem, in that it makes necessary to determine a greater number of parameter values in order to achieve a practical implementation.

From a theoretical point of view, we believe that it is more appropriate to ensure a monotonic decay of the probability of a sample value to remain inside the set of samples. A pixel model should contain samples from the recent past of the pixel but older samples should not necessarily be discarded.

We propose a method that offers an exponential monotonic decay for the remaining lifespan of the samples. The method improves the time relevance of the estimation by allowing a few old samples to remain in the pixel model. Remember that this approach is combined with a conservative update policy, so that foreground values should never be included in the models.

The technique, illustrated in Figure 2.2, is simple but effective: instead of systematically removing the oldest sample from the pixel model, we choose the sample to be discarded randomly according to a uniform probability density function. The new value then replaces the selected sample. This random strategy contradicts the idea that older values should be replaced first, which is not true for a conservative update policy. A conservative update policy is also necessary for the stability of the process. Indeed, the random update policy produces a non-deterministic background subtraction algorithm (to our knowledge, this is the first background subtraction algorithm to have that property). Only a conservative update policy ensures that the models do not diverge over time. Despite this, there may be slight differences, imperceptible in our experience, between the results of the same sequence processed by our background subtraction algorithm at different times.

Mathematically, the probability of a sample present in the model at time t being preserved after the update of the pixel model is given by $\frac{N-1}{N}$. Assuming time continuity and the absence of memory in the selection procedure, we can derive a similar probability, denoted $P(t,\, t+dt)$ hereafter, for any further time $t+dt$. This probability is equal to

$$P(t,\, t+dt) = \left(\frac{N-1}{N}\right)^{(t+dt)-t} \tag{2.5}$$

which can be rewritten as

$$P(t,\, t+dt) = e^{-\ln\left(\frac{N}{N-1}\right)dt}. \tag{2.6}$$

This expression shows that the expected remaining lifespan of any sample value of the model decays exponentially. It appears that the probability of a sample being preserved for the interval $(t,\, t+dt)$, assuming that it was included in the model prior to time t, is independent of t. In other words, the past has no effect on the future. This property, called the *memoryless* property, is known to be applicable to an exponential density (see [93]). This is a remarkable and, to our knowledge, unique property in the field of background subtraction. It completely frees us to define a time period for keeping a sample in the history of a pixel and, to some extent, allows the update mechanism to adapt to an arbitrary framerate.

2.3.3.3 Time subsampling

We have shown how the use of a random replacement policy allow our pixel model to cover a large (theoretically infinite) time window with a limited number of samples. In order to further extend the size of the time window covered by a pixel model of a fixed size, we resort to random time subsampling. The idea is that in many practical situations, it is not necessary to update each background pixel model for each new frame. By making the background update less frequent, we artificially extend the expected lifespan of the background samples. But in the presence of periodic or pseudo-periodic background motions, the use of fixed subsampling intervals might prevent the background model from properly adapting to these motions. This motivates us to use a *random* subsampling policy. In practice, when a pixel value has been classified as belonging to the background, a random process determines whether this value is used to update the corresponding pixel model.

In all our tests, we adopted a time subsampling factor, denoted ϕ, of 16: a background pixel value has 1 chance in 16 of being selected to update its pixel model. But one may wish to tune this parameter to adjust the length of the time window covered by the pixel model.

2.3.3.4 Spatial consistency through background samples propagation

Since we use a conservative update scheme, we have to provide a way of updating the background pixel models that are hidden by the foreground. A popular way of doing this is to use what the authors of the W^4 algorithm [42] call a "*detection support map*" which counts the number of consecutive times that a pixel has been classified as foreground. If this number reaches a given threshold for a particular pixel location, the current pixel value at that location is inserted into the background model. A variant consists of including, in the background, groups of connected foreground pixels that have been found static for a long time, as in [24]. Some authors, like those of the W^4 algorithm and those of the SACON model [120, 121], use a combination of a pixel-level and an object-level background update.

The strength of using a conservative update comes from the fact that pixel values classified as foreground are never included in any background pixel model. While convenient, the support map related methods only delay the inclusion of foreground pixels. Furthermore, since these methods rely on a binary decision, it takes time to recover from a improper inclusion of a genuine foreground object in the background model. A progressive inclusion of foreground samples in the background pixel models is more appropriate.

As stated earlier, we have a different approach. We consider that neighboring background pixels share a similar temporal distribution and that a new background sample of a pixel should also update the models of neighboring pixels. According to this policy, background models hidden by the foreground will be updated with background samples *from neighboring pixel locations* from time to time. This allows a spatial diffusion of information regarding the background evolution that relies on samples classified *exclusively* as background. Our background model is thus able to adapt to a changing illumination and to structural evolutions (added or removed background objects) while relying on a *strict* conservative update scheme.

More precisely, let us consider the 4- or 8-connected spatial neighborhood of a pixel x, that is $N_G(x)$, and assume that it has been decided to update the set of samples $\mathcal{M}(x)$ by inserting $v(x)$. We then also use this value $v(x)$ to update the set of samples $\mathcal{M}(y \in N_G(x))$ from one of the pixels in the neighborhood, chosen at random according to a uniform law.

Since pixel models contain many samples, irrelevant information that could accidentally be inserted into the neighborhood model does not affect the accuracy of the detection. Furthermore, the erroneous diffusion of irrelevant information is blocked by the need to match an observed value before it can propagate further. This natural limitation inhibits the diffusion of error.

Note that neither the selection policy nor the spatial propagation method is deterministic. As stated earlier, if the algorithm is run over the same video sequence again, the results will always differ slightly (see Figure 2.2). Although unusual, the strategy of allowing a random process to determine which samples are to be discarded proves to be very powerful. This is different from known strategies that introduce a fading factor or that use a long term and a short term history of values.

This concludes the description of our algorithm. ViBe makes no assumption regarding the video stream framerate or color space, nor regarding the scene content, the background itself, or its variability over time. Therefore, we refer to it as a universal method.

2.4 Experimental results

In this section, we determine optimal values for the parameters of ViBe, and compare its results with those of six other algorithms: one simple method and five state-of-the art techniques. We also describe some advantageous and intrinsic properties of ViBe, and finally, we illustrate the suitability of ViBe for embedded systems.

For the sake of comparison, we have produced manually ground-truth segmentation maps for subsets of frames taken from two test sequences. The first sequence (called "house") was captured outdoor on a windy day. The second sequence ("pets") was extracted from the PETS2001 public data-set (data-set 3, camera 2, testing). Both sequences are challenging as they feature background motion, moving trees and bushes, and changing illumination conditions. The "pets" sequence is first used below to determine objective values for some of the parameters of ViBe. We then compare ViBe with 6 existing algorithms on both sequences.

Many metrics can be used to assess the output of a background subtraction algorithm given a series of ground-truth segmentation maps. These metrics usually involve the following quantities: the number of true positives (TP), which counts the number of correctly detected foreground pixels; the number of false positives (FP), which counts the number of background pixels incorrectly classified as foreground; the number of true negatives (TN), which counts the number of correctly classified background pixels; and the number of false negatives (FN), which accounts for the number of foreground pixels incorrectly classified as background.

The difficulty of assessing background subtraction algorithms originates from the lack of a standardized evaluation framework; some frameworks have been proposed by various authors but mainly with the aim of pointing out the advantages of their own method. According to [33], the metric most widely used in computer vision to assess the performance of a binary classifier is the Percentage of Correct Classification (PCC), which combines all four values:

$$PCC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.7}$$

This metric was adopted for our comparative tests. Note that the PCC percentage needs to be as high as possible, in order to minimize errors.

2.4.1 Determination of our own parameters

From previous discussions, it appears that ViBe has the following parameters:

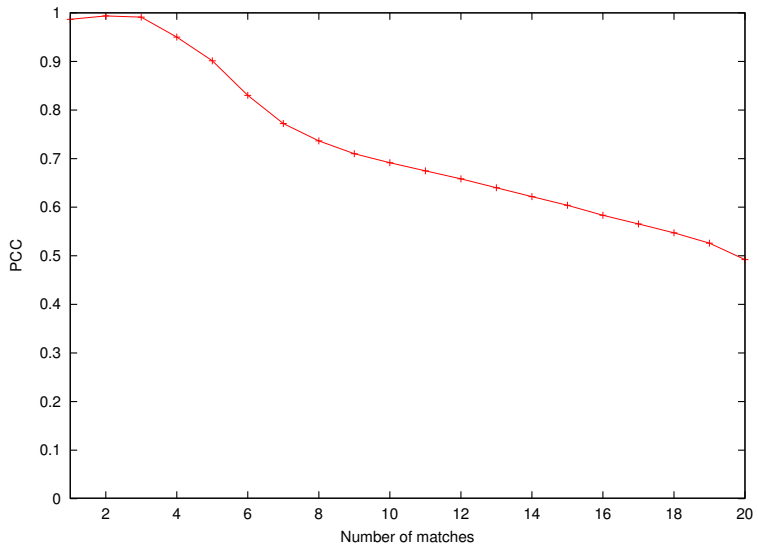


Figure 2.3: Percentages of Correct Classification (PCCs) for $\sharp_{\min}$ ranging from 1 to 20. The other parameters of ViBe were set to $N = 20$, $R = 20$, and $\phi = 16$.

- the radius R of the sphere used to compare a new pixel value to pixel samples (see equation 2.2),
- the time subsampling factor ϕ,
- the number N of samples stored in each pixel model,
- and the number $\sharp_{\min}$ of close pixel samples needed to classify a new pixel value as background (see equation 2.2).

In our experience, the use of a radius $R = 20$[1] and a time subsampling factor $\phi = 16$ leads to excellent results in every situation. Note that the use of $R = 20$ is an educated choice, which corresponds to a perceptible difference in color.

To determine an optimal value for $\sharp_{\min}$, we compute the evolution of the PCC of ViBe on the "pets" sequence for $\sharp_{\min}$ ranging from 1 to 20. The other parameters were fixed to $N = 20$, $R = 20$, and $\phi = 16$. Figure 2.3 shows that the best PCCs are obtained for $\sharp_{\min} = 2$ and $\sharp_{\min} = 3$.

Since a rise in $\sharp_{\min}$ is likely to increase the computational cost of ViBe, we set the optimal value of $\sharp_{\min}$ to $\sharp_{\min} = 2$. Note that in our experience, the use of $\sharp_{\min} = 1$ can lead to excellent results in scenes with a stable background.

Once the value of 2 has been selected for $\sharp_{\min}$, we study the influence of the parameter N on the performance of ViBe. Figure 2.4 shows percentages obtained on the "pets" sequence for N ranging from 2 to 50 (R and ϕ were set to 20 and 16). We observe that higher values of N provide a better performance. However, they tend to saturate for values higher than 20. Since as for $\sharp_{\min}$, large N values induce a greater computational cost, we select N at the beginning of the plateau, that is $N = 20$.

2.4.2 Comparison with other techniques

We now compare the results of ViBe with those of five state-of-the-art background subtraction algorithms and a basic method: (1) the gaussian mixture model proposed in [52] (hereafter referred to as GMM); (2) the gaussian mixture model of [136] (referred to as EGMM); (3) the Bayesian

[1] All our experiments were conducted using either grayscale images that have a pixel color depth of 8 bits or color images that have a pixel color depth of 24 bits.

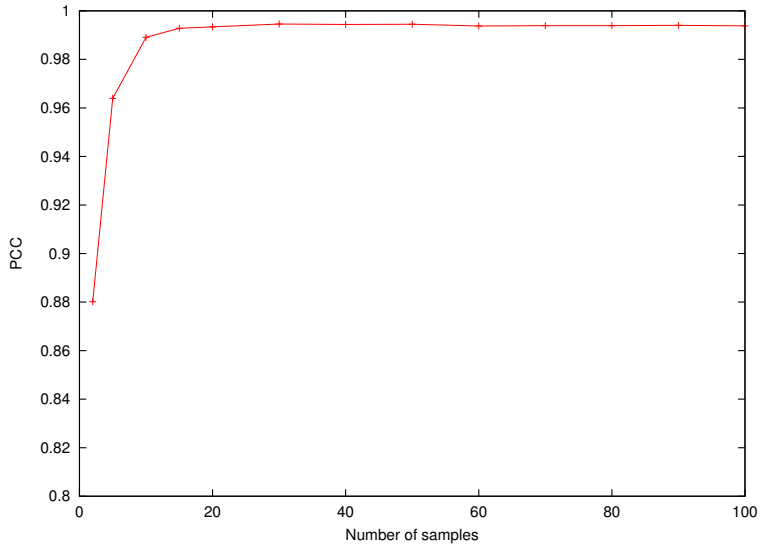


Figure 2.4: Percentages of Correct Classification (PCCs) given the number of samples collected in a background model.

algorithm based on histograms introduced in [64]; (4) the codebook algorithm [56]; (5) the zipfian $\Sigma - \Delta$ estimator of [73]; and (6) the first-order low-pass filter (that is $B_t = \alpha I_t + (1 - \alpha B_{t-1})$, where I_t and B_t are respectively the input and background images at time t), which is used as a baseline.

The first-order low-pass filter was tested using a fading factor α of 0.05 and a detection threshold T of 20. The GMM of [64] and the Bayesian algorithm of [52] were tested using their implementations available in Intel's IPP image processing library. For the EGMM algorithm of [136], we used the implementation available on the author's website[2]. The authors of the zipfian $\Sigma - \Delta$ filter were kind enough to provide us with their code to test their method. We implemented the codebook algorithm ourselves and used the following parameters: 50 training frames, $\lambda = 34$, $\epsilon_1 = 0.2$, $\epsilon_2 = 50$, $\alpha = 0.4$, and $\beta = 1.2$. ViBe was tested with the default values proposed in this chapter: $N = 20$, $R = 20$, $\sharp_{\min} = 2$, and $\phi = 16$. Most of the algorithms were tested using the RGB color space; the codebook uses its own color space, and the $\Sigma - \Delta$ filter implementation works on grayscale image. In addition, we implemented a grayscale version of ViBe.

Figures 2.5 and 2.6 show examples of foreground detection for one typical frame of each sequence. Foreground and background pixels are shown in white and black respectively.

Visually, the results of ViBe look better and are the closest to ground-truth references. This is confirmed by the PCC scores; the PCC scores of the eight comparison algorithms for both sequences are shown in Figure 2.7.

We also compared the computation times of these eight algorithms with a profiling tool, and expressed the computation times in terms of achievable framerates. Figure 2.8 shows their average processing speed on our platform (2.67GHz Core i7 CPU, 6GB of RAM, C implementation). We did not optimize the code of the algorithms explicitely, except in the case of the $\Sigma - \Delta$ algorithm, which was optimized by its authors, the algorithms of the IPP library (GMM and Bayesian histogram), optimized for Intel processors, and ViBe to some extent. To speed up operations involving random numbers in ViBe, we used a buffer pre-filled with random numbers.

We see that ViBe clearly outperforms the six other techniques: its PCCs are the highest for both sequences and its processing speed is as high as a framerate of 200 frames per second, that is 5 times more than algorithms optimized by Intel. Compare these figures to those obtained by the algorithm proposed by Chiu *et al. [21]* recently; they claim to segment 320×240 images at

[2]http://staff.science.uva.nl/~zivkovic/DOWNLOAD.html

(a) Input image (b) Ground-truth (c) ViBe (RGB) (d) ViBe (gray) (e) Bayesian hist.

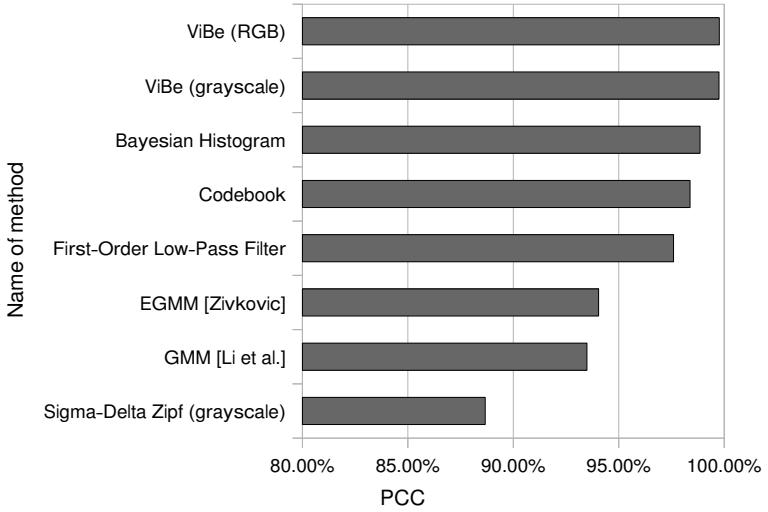
(f) Codebook (g) EGMM [Zivkovic] (h) GMM [Li et al.] (i) 1^{st} order filter (j) Sigma-Delta Z.

Figure 2.5: Comparative results for one frame taken from the "house" sequence.

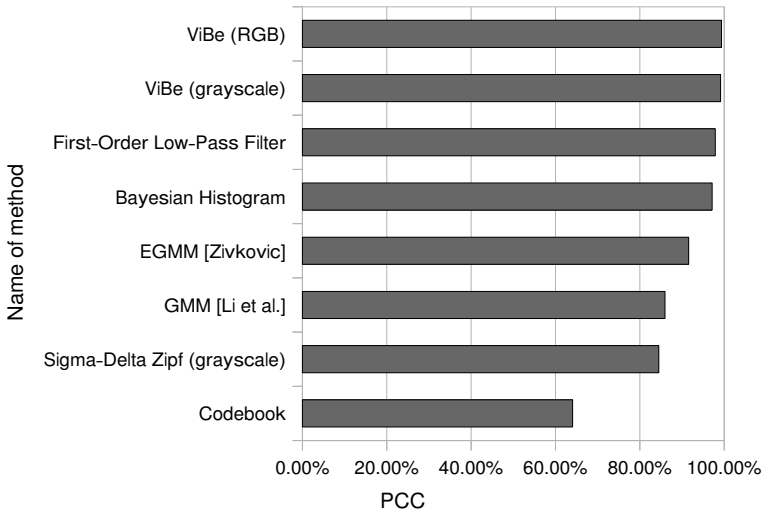
(a) Input image (b) Ground-truth (c) ViBe (RGB) (d) ViBe (gray) (e) Bayesian hist.

(f) Codebook (g) EGMM [Zivkovic] (h) GMM [Li et al.] (i) 1^{st} order filter (j) Sigma-Delta Z.

Figure 2.6: Comparative results for one frame taken from the "pets" sequence.



(a) Results for the first sequence (“house”).



(b) Results for the second sequence (“pets”).

Figure 2.7: Comparative results.

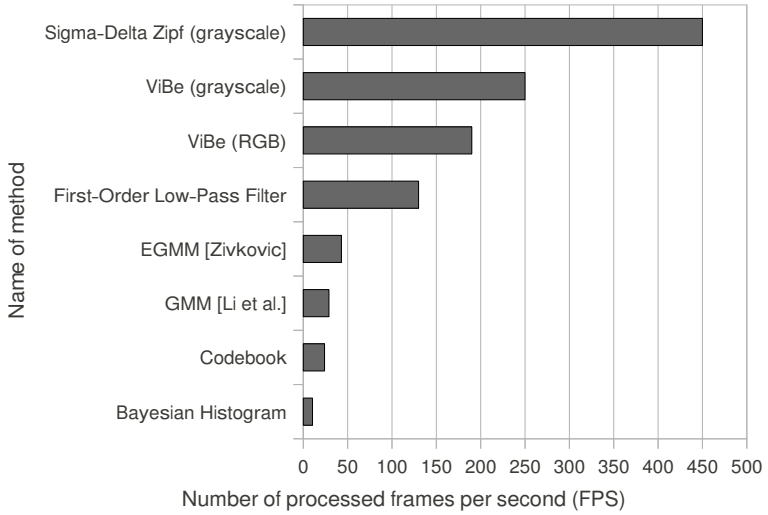


Figure 2.8: Processing speed of the tested methods for images of 640×480 pixels.

a framerate of around 40 frames per second. A simple rescaling to the size of our images lowers this value to 10 frames per second.

The only method faster than ViBe is the zipfian $\Sigma - \Delta$ estimator, whose PCC is 12 to 15% smaller than that of ViBe. The authors of the zipfian sigma-delta algorithm provided us with post-processed segmentation maps of the "house" sequence which exhibit an improved PCC but a the cost of a lower processing speed. In terms of PCC scores, only the Bayesian algorithm of [64] based on histograms competes with ViBe. However, it is more than 20 times slower than ViBe. As shown in Figures 2.5 and 2.6, the grayscale and the color versions of ViBe manage to combine both a very small rate of FP and a sharp detection of the foreground pixels. The low FP rate of ViBe eliminates the need for any post-processing, which further alleviates the total computational cost of the foreground detection.

Next, we concentrate on the specific strengths of ViBe: fast ghost suppression, intrinsic resilience to camera shake, noise resilience, and suitability for embedding.

2.4.3 Faster ghost suppression

A background model has to adapt to modifications of the background caused by changing lighting conditions but also to those caused by the addition, removal, or displacement of some of its parts. These events are the principal cause of the appearance of ghosts: regions of connected foreground points that do not correspond to any real object.

When using a detection support map or a related technique to detect and suppress ghosts, it is very hard, if not impossible, to distinguish ghosts from foreground objects that are currently static. As a result, real foreground objects are included in the background model if they remain static for too long. This is a correct behavior since a static foreground object must eventually become part of the background after a given time. It would be better if ghosts were included in the background model more rapidly than real objects, but this is impossible since they cannot be distinguished using a detection support map.

Our spatial update mechanism speeds up the inclusion of ghosts in the background model so that the process is faster than the inclusion of real static foreground objects. This can be achieved because the borders of the foreground objects often exhibit colors that differ noticeably from those of the samples stored in the surrounding background pixel models. When a foreground object stops moving, the information propagation technique described in Section 2.3.3.4 updates the pixel models located at its borders with samples coming from surrounding background pixels. But these samples are irrelevant: their colors do not match at all those of the borders of the

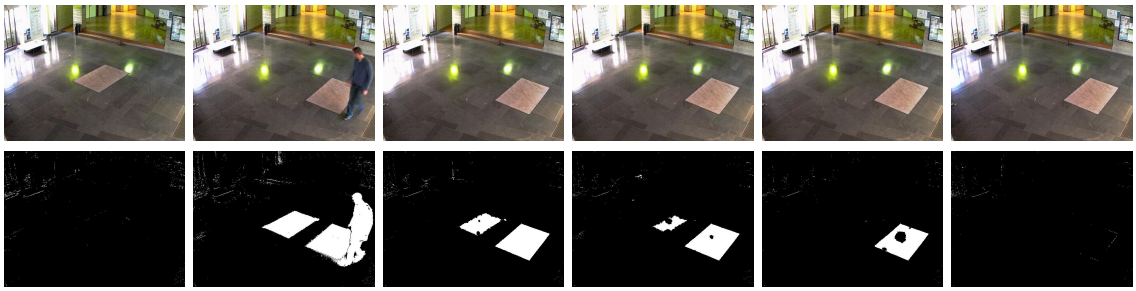


Figure 2.9: Fast suppression of a ghost. In this scene, an object (a carpet) is moved, leaving a ghost behind it in the background, and is detected as being part of the foreground. It can be seen that the ghost is absorbed into the background model much faster than the foreground region corresponding to the real physical object.

object. In subsequent frames, the object remains in the foreground, since background samples cannot diffuse inside the foreground object via its borders.

By contrast, a ghost area often shares similar colors with the surrounding background. When background samples from the area surrounding the ghost try to diffuse inside the ghost, they are likely to match the actual color of the image at the locations where they are diffused. As a result, the ghost is progressively eroded until it disappears entirely. Figure 2.9 illustrates this discussion.

The speed of this process depends on the texture of the background: the faster ghost suppressions are obtained with backgrounds void of edges. Furthermore, if the color of the removed object is close to that of the uncovered background area, the absorption of the ghost is faster. When needed, the speed of the ghost suppression process can be tuned by adapting the time subsampling factor ϕ. For example, in the sequence displayed in Figure 2.9, if we assume a framerate of 30 frames per second, the ghost fades out after 2 seconds for a time subsampling factor ϕ equal to 1. However, if we set ϕ to 64, it takes 2 minutes for ViBe to suppress the ghost completely. For the sake of comparison, the Bayesian histogram algorithm suppresses the same ghost area in 5 seconds.

One may ask how static foreground objects will ultimately be included in the background model. The responsibility for the absorption of foreground pixels into the background lies with the noise inevitably present in the video sequence. Due to the noise, some pixels of the foreground object end up in the background, and then serve as background seeds. Consequently, their models are corrupted with foreground samples. These samples later diffuse into their neighboring models, as a result of the spatial propagation mechanism of the background samples, and allow a slow inclusion of foreground objects in the background.

2.4.4 Resistance to camera displacements

In many situations, small displacements of the camera are encountered. These small displacements are typically due to vibrations or wind and, with many other techniques, they cause significant numbers of false foreground detections.

Another obvious benefit of the spatial consistency of our background model is an increased robustness against such small camera movements (see Figure 2.10). Since samples are shared between neighboring pixel models, small displacements of the camera introduce very few erroneous foreground detections.

ViBe also has the capability of dealing with large displacements of the camera, at the price of a modification of the base algorithm. Since our model is purely pixel-based, we can make it able to handle moving cameras by allowing pixel models to follow the corresponding physical pixels according to the movements of the camera. The movements of the camera can be estimated either using embedded motion sensors or directly from the video stream using an algorithmic technique. This concept is illustrated in Figures 2.11 and 2.12. The first series shows images taken from an

Figure 2.10: Background subtraction for a slightly moving camera. If spatial propagation is deactivated, the camera motions produce false positives in high-frequency areas (image in the center), while the activation of spatial propagation avoids a significant proportion of false positives (right-hand image).

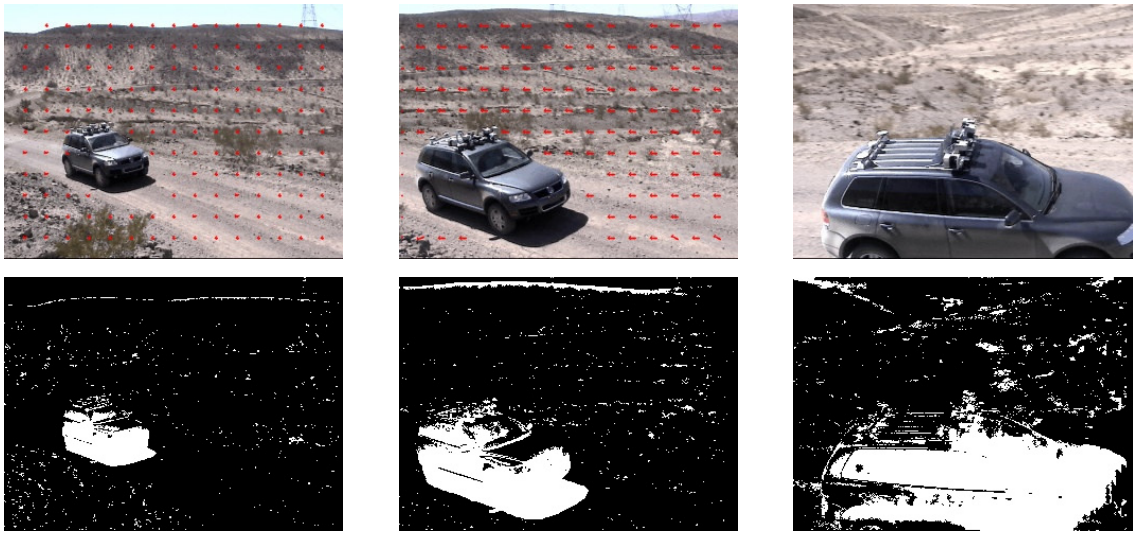


Figure 2.11: Segmentation maps for a sequence taken with a moving camera (from the DARPA challenge).

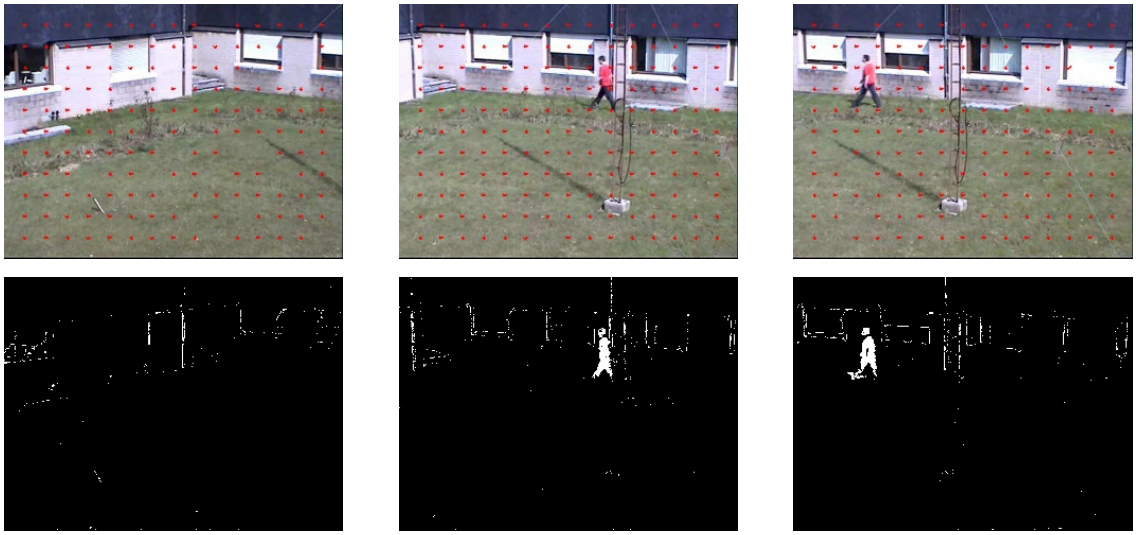
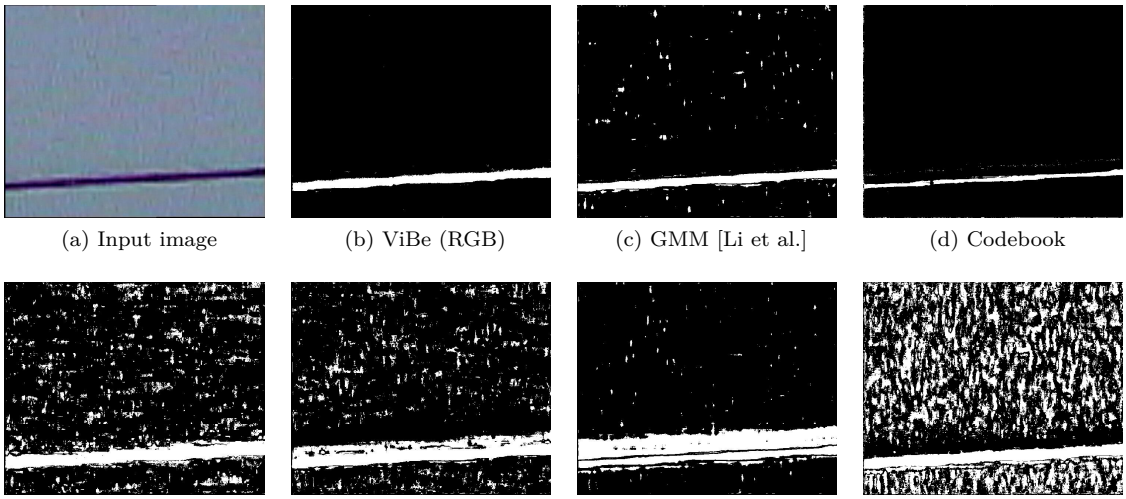


Figure 2.12: Segmentation maps for a sequence taken with a moving camera (surveillance camera).



(a) Input image (b) ViBe (RGB) (c) GMM [Li et al.] (d) Codebook

(e) Bayesian histogram (f) EGMM [Zivkovic] (g) 1^{st} order filter (h) Sigma-Delta Zipf

Figure 2.13: Comparative results for one frame taken from the noisy "cable" sequence.

old DARPA challenge. The camera pans the scene from left to right and the objective is to follow the car. Figure 2.12 shows a similar scenario acquired with a Pan-Tilt Zoom video-surveillance camera; the aim here is to track the person.

To produce the images of Figures 2.11 and 2.12, the displacement vector between two consecutive frames is estimated for a subset of background points located on a regularly spaced grid using Lucas and Kanade's optical flow estimator [70]. The global displacement vector of the camera is computed by averaging these pixel-wise displacement vectors. The pixel models are then relocated according to the displacement of the camera inside a larger mosaic reference image. The background model of pixels that correspond to areas seen for the first time is initialized instantaneously using the technique described in Section 2.3.2. It can be seen that, even with such a simple technique, the results displayed in Figures 2.11 and 2.12 are promising.

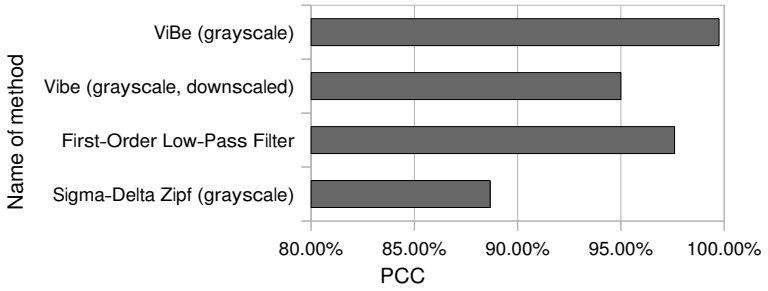
2.4.5 Resilience to noise

To demonstrate the resilience of ViBe to noise, we compared it to 6 other techniques on a difficult noisy sequence (called "cable"). This sequence shows an oscillating electrical cable filmed at a large distance with a 40× optical zoom. As can be seen in Figure 2.13a, the difficult acquisition conditions result in an significant level of noise in the pixel values. Background/foreground segmentation maps displayed in Figure 2.13 demonstrate that ViBe is the only technique that manages to combine a low rate of FP with both a precise and accurate detection of the foreground pixels.

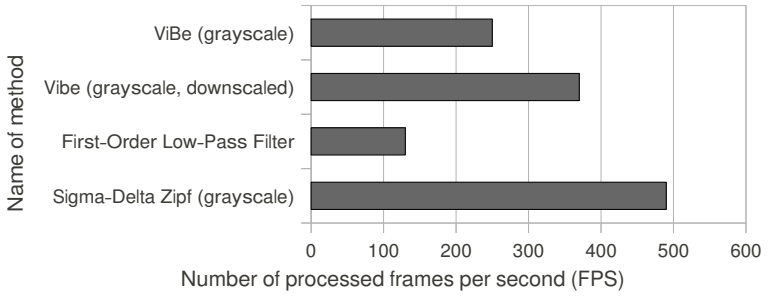
Two factors must be credited for ViBe's high resilience to noise. The first originates from our design, allowing the pixel models of ViBe to comprise *exclusively* observed pixel values. The pixel models of ViBe adapt to noise automatically, as they are constructed from noisy pixel values. The second factor is the pure conservative update scheme used by ViBe (see Section 2.3.3). By relying on pixel values classified exclusively as background, the model update policy of ViBe prevents the inclusion of any outlier in the pixel models. As a result, these two factors ensure a continuous adaptation to the noise present in the video sequence while maintaining coherent pixel models.

2.4.6 Downscaled version and embedded implementation

Since ViBe has a low computational cost (see Figure 2.8) and relies exclusively on integer computations, it is particularly well suited to an embedded implementation. Furthermore, the compu-



(a) PCCs



(b) FPS for images of 640×480 pixels.

Figure 2.14: PCCs and processing speeds of fast techniques, including a downscaled version of ViBe which requires only one comparison and one byte of memory per pixel.

tational cost of ViBe can be further reduced by using low values for N and $\sharp_{\min}$. In Figure 2.14, we give the PCC scores and framerates for a downscaled version of ViBe, which uses the absolute minimum of one comparison and one byte of memory per pixel. We also give the PCC scores and framerates for the full version of ViBe and for the two faster techniques from our tests in Section 2.4.2. One can see, on the left hand side of the graph in Figure 2.14, that the downscaled version of ViBe maintains a high PCC. Note that its PCC is higher than that of the two GMM-based techniques tested in Section 2.4.2 (see Figure 2.7a). In terms of processing speed or framerate, the zipfian $\Sigma - \Delta$ filter method of [73] is the only one to be faster than the downscaled version of ViBe. However, a post-processing step of the segmentation map is necessary to increase the low PCC score of the zipfian $\Sigma - \Delta$ method, and the computational cost induced by this post-processing process reduces the framerate significantly.

To illustrate the low computational cost of ViBe and its simplicity, we embedded our algorithm in a digital camera. The porting work of ViBe on a *Canon PowerShot SD870 IS* was performed with a modified version of the open source alternative firmware CHDK[3]. Parameters of ViBe were set to $N = 5$ and $\sharp_{\min} = 1$. Despite the camera's low speed ARM processor, we managed to process 6 frames of 320×240 pixels wide images per second on average. The result is shown in Figure 2.15.

2.5 Conclusions

In this chapter, we introduced a universal sample-based background subtraction algorithm, called ViBe, which combines three innovative techniques.

Firstly, we proposed a classification model that is based on a small number of correspondences between a candidate value and the corresponding background pixel model. Secondly, we explained how ViBe can be initialized with a single frame. This frees us from the need to wait for several

[3]http://chdk.wikia.com

Figure 2.15: Embedded implementation of ViBe in a Canon camera.

seconds to initialize the background model, an advantage for image processing solutions embedded in digital cameras and for short sequences. Finally, we presented our last innovation: an original update mechanism. Instead of keeping samples in the pixel models for a fixed amount of time, we ignore the insertion time of a pixel in the model and select a value to be replaced randomly. This results in a smooth decaying lifespan for the pixel samples, and enables an appropriate behavior of the technique for wider ranges of background evolution rates while reducing the required number of samples needing to be stored for each pixel model. Furthermore, we also ensure the spatial consistency of the background model by allowing samples to diffuse between neighboring pixel models. We observe that the spatial process is responsible for a better resilience to camera motions, but that it also frees us from the need to post-process segmentation maps in order to obtain spatially coherent results. To be effective, the spatial propagation technique and update mechanism are combined with a strictly conservative update scheme: no foreground pixel value should ever be included in any background model.

After a description of our algorithm, we determined optimal values for all the parameters of the method. Using this set of parameter values, we then compared the classification scores and processing speeds of ViBe with those of six other background subtraction algorithms on two sequences. ViBe is shown to outperform all of these algorithms while being faster than five of them. Finally, we discussed the performance of a downscaled version of ViBe, which can process more than 350 frames per second on our platform. This downscaled version was embedded in a digital camera to prove its suitability for platforms with low computational power. Interestingly, we found that a version of ViBe downscaled to the absolute minimum amount of resources for any background subtraction algorithm (that is one byte of memory and one comparison with a memorized value per pixel) performed better than the state-of-the-art algorithms in terms of the Percentage of Correct Classification criterion. ViBe might well be a new milestone for the large family of background subtraction algorithms.

2.6 C-like source code of ViBe

Hereafter, we give a C-like pseudo-code of ViBe for grayscale images, comprising default values for all the parameters of the method.

```
1  int width;                                  // width of the image
2  int height;                                 // height of the image
3  byte image[width][height];                  // current image
4  byte segmentationMap[width][height];        // foreground detection map
5
6  int nbSamples = 20;                         // number of samples per pixel
7  int reqMatches = 2;                         // #_min
8  int radius = 20;                            // R
9  int subsamplingFactor = 16;                 // amount of random subsampling
10
11 byte samples[width][height][nbSamples];     // background model
12
13 for (int x = 0; x < width; x++) {
14   for (int y = 0; y < height; y++){
15     // comparison with the model
16     int count = 0, i = 0;
17     while ((count < reqMatches) && (index < nbSamples)){
18       int distance = getEuclideanDist(image[x][y], samples[x][y][i]);
19       if (distance < radius)
20         count++;
21       i++;
22     }   // pixel classification according to reqMatches
23     if (count >= reqMatches){ // the pixel belongs to the background
24       // stores the result in the segmentation map
25       setPixelBackground(segmentationMap[x][y]);
26       // gets a random number between 0 and subsamplingFactor-1
27       int randomNumber = getRandomNumber(0, subsamplingFactor-1);
28       // updates of the current pixel model
29       if (randomNumber == 0){ // random subsampling
30         // other random values are ignored
31         randomNumber = getRandomNumber(0, nbSamples-1);
32         samples[x][y][randomNumber] = image[x][y];
33       }
34       // updates of a neighboring pixel model
35       randomNumber = getRandomNumber(0, subsamplingFactor-1);
36       if (randomNumber == 0){ // random subsampling
37         // chooses a neighboring pixel randomly
38         (int neighborX, neighborY) = chooseRandomNeighbor(x, y);
39         // chooses the value to be replaced randomly
40         randomNumber = getRandomNumber(0, nbSamples-1);
41         samples[neighborX][neighborY][randomNumber] = image[x][y];
42       }
43     }
44     else // pixel belongs to the foreground
45       // stores the result in the segmentation map
46       setPixelForeground(segmentationMap[x][y]);
47   }
48 }
```

Chapter 3

Person detection: Robust analysis of silhouettes by morphological size distributions

In this chapter, we describe a method able to detect and locate persons in a video stream captured with a static camera. This chapter in an extended version of an article that was published in the proceedings of the 2006 Advanced Concepts for Intelligent Vision Systems conference [4]. This work was made in close collaboration with Sébastien Jodogne who took care of most of the aspects related to the classification algorithms we used.

Abstract

We address the topic of real-time analysis and recognition of silhouettes. The method that we propose first produces object features obtained by a new type of morphological operators, which can be seen as an extension of existing granulometric filters, and then insert them into a tailored classification scheme.

Intuitively, given a binary segmented image, our operator produces the set of all the maximal rectangles that can be wedged inside any connected component of the image. The latter are obtained by a standard background subtraction technique and morphological filtering. To classify connected components into one of the known object categories, the rectangles of a connected component are submitted to a machine learning algorithm called EXtremely RAndomized trees (Extra-trees). The machine learning algorithm is fed with a static database of silhouettes, which contains both positive and negative instances. The whole process, including image processing and rectangle classification, is carried out in real-time.

Finally we evaluate our approach on one of today's hot topics: the detection of human silhouettes. We discuss experimental results and show that our method is stable and computationally effective. Therefore, we assess that algorithms like ours introduce new ways for the detection of humans in video sequences.

3.1 Introduction

During the recent years, the rise of cheap sensors has made of video surveillance a topic of very active research and wide economical interest. In this field, one of the expected major breakthrough would be to design automatic image processing systems able to detect, to track, and to analyze human activities. Unfortunately the amount of data generated by cameras is prohibitively huge, although the informative part of such signals is very tight with respect to their raw content.

Several algorithms in computer vision have been developed to summarize such informative patterns as a set of *visual feature*s. These algorithms generally rely on the detection of discontinuities in the signal selected by *interest point detectors* [100]. A local description of the neighborhood of these interest points is then computed [81] and this description serves to track a feature in the successive frames of a video sequence. Methods like this, referred to as *local-appearance methods*, have been used with some success in computer vision applications, such as image matching, image retrieval, and object recognition (see [69, 99]).

From the current literature, it is still unclear whether such local-appearance descriptors are appropriate for tracking human silhouettes, or more specifically for gait analysis. Indeed, they are rather computationally expensive, and as they are inherently local, it is impossible for them to represent the overall geometry of a silhouette. There are two potential solutions to this problem: (1) introduce higher-level descriptors able to represent the relative spatial arrangements between visual features [79], or (2) take global appearance (such as contours) into consideration instead of local appearance.

Gait analysis techniques based on the global geometry of the objects have been discussed by Boulgouris *et al.* [11]. According to them, techniques that employ binary images are believed to be particularly suited for most practical applications since color or texture information might not be available or appropriate. The contour of a silhouette is probably the most sensible visual feature in this class. A direct use of the contour is possible. Another solution is to transform the contour into a series of Fourier descriptors, which are common in shape description. Other candidates have been proposed, such as the width of the silhouette, its horizontal and vertical projections, or its angular representation.

In this chapter, we propose a novel approach that is at the crossroad between local- and global-appearance techniques. Our approach innovates in that we propose a new family of visual features that rely on a surfacic description of a silhouette. Intuitively, we cover the silhouette by the set of all the maximal rectangles that can be wedged inside it. More precisely, each (local) position in the silhouette is linked to the subset of the maximal rectangles that cover the position and that are entirely included in the (global) silhouette.

Surfacic descriptors, such as the morphological skeleton [102], have already been studied in the scope of shape compression, whose goal is to reduce the amount of redundant information. Many of them require large computation times. This makes them less suitable for real-time applications. This contrasts with our features, as it is possible to compute them in real-time.

This chapter describes an attempt to take advantage of such novel features. To illustrate our approach, we focus on the detection of human bodies in a video stream, as in [91, 127]. Basically, we apply *machine learning* algorithms on the rectangles of a silhouette to decide, in real-time, whether this silhouette corresponds to that of a learned instance of a human silhouette. This decision is a compulsory step for any gait recognition task, and improvements in this area will impact on the overall performances of algorithms that deal with the automatic analysis of human behavior. Our results show how promising an approach like ours can be.

The chapter is organized as follows. We start by describing the architecture of our silhouettes detection and analysis technique in Section 3.2, which mainly consists in three steps (silhouettes extraction, description, and classification) respectively detailed in Sections 3.3, 3.4, and 3.5. Experimental results, which consist in the application of our method for the detection of human people in video sequences, are discussed in Section 3.6.

3.2 Overall architecture

The overall architecture of our silhouettes detection, analysis and classification system is depicted in Figure 3.1.

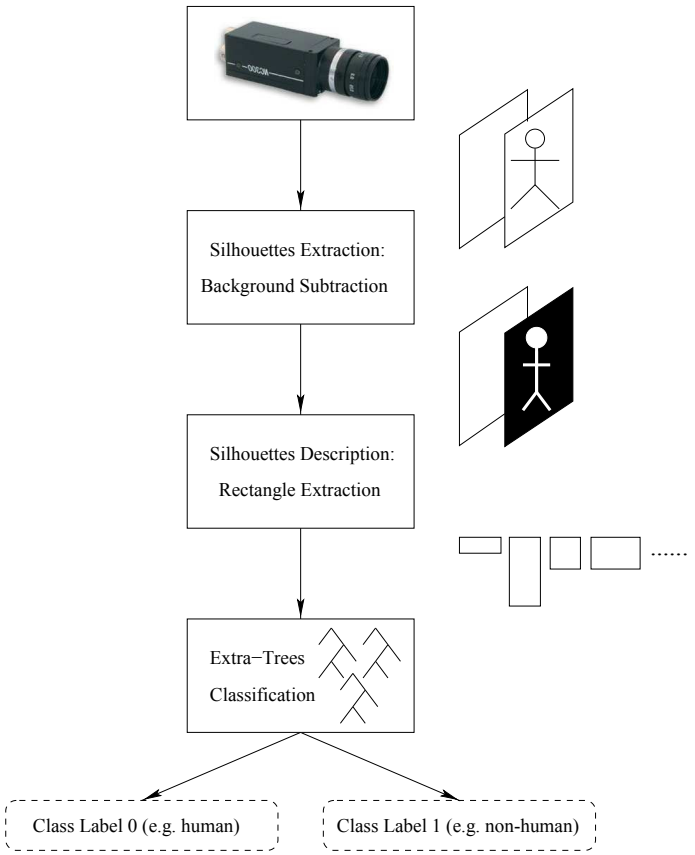


Figure 3.1: Overall architecture.

It comprises three modules.

1. The first module consists in extracting the candidate silhouettes from the video stream. It is described in Section 3.3.
2. One of the major difficulties in classification lies in finding appropriate feature measures. In the second module, we use our new granulometric operator to produce a set of features (wedged maximal rectangles) describing the extracted silhouettes.
3. The task of the third module is to classify rectangle features to decide whether or not the silhouettes belong to the class of interest. The classification is achieved by the means of an extra-tree learning algorithm. The third module is detailed in Section 3.5.

3.3 Extraction of silhouettes[1]

The first step of our system consists in the segmentation of the input video stream in order to produce binary silhouettes, which will be fed into the silhouettes description module. We achieve this by a motion segmentation based on an adaptive background subtraction method.

[1]The work described in this Chapter does not make use of the silhouette extraction algorithm introduced in Chapter 2. As explained in Chapter 1, the work described in this Chapter has been achieved before the work described in Chapter 2. However, our later work has shown that ViBe is the best choice for background subtraction.

Background segmentation methods are numerous. The method we have chosen is based on an adaptive modeling of each pixel as a mixture of Gaussians, each of which corresponds to the probability of observing a particular intensity or color for this pixel. In each Gaussian cluster, the mean accounts for the average color or intensity of the pixel, whereas the variance is used to model illumination variations, surface texture, and camera noise. The whole algorithm relies on the assumptions that the background is visible more frequently than the foreground and that the variance of the background is relatively low. These are common assumptions for most background subtraction techniques. An extensive description of the algorithm can be found in [109, 110] and a tutorial is available in [97]. The technical description is given hereafter.

If X_t is the color or intensity value observed at time t for a particular pixel in the image, the history of the pixel $\{X_1, \ldots, X_t\}$ is modeled as a mixture of K Gaussian distributions. The probability of observing a particular color or intensity value at time t is expressed as

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \tag{3.1}$$

where

- K is the number of Gaussian clusters used to model the history of the pixel,
- $\omega_{i,t}$ is the weight associated with the ith cluster at time t –it models the amount of data represented by the ith Gaussian–,
- $\mu_{i,t}$ and $\Sigma_{i,t}$ are the mean and covariance matrix of the ith Gaussian, and
- η is a Gaussian probability density function.

For computational efficiency reasons, the covariance matrix $\Sigma_{i,t}$ is assumed to be isotropic and diagonal

$$\Sigma_{i,t} = \sigma_k^2 \mathbf{I}. \tag{3.2}$$

The Gaussian distributions are sorted in decreasing order of the ratio $\frac{\omega_{i,t}}{\sigma_{i,t}}$. The j first Gaussians are considered to account for the background, while the rest of them accounts for the foreground. The j factor is dynamically estimated by accumulating the $\omega_{i,t}$ values, according to the computed order of the Gaussians, until a given threshold value T is reached. As stated above, this algorithm assumes that the background is visible more often than the foreground and that the variance of the background is relatively low.

Every new pixel value is checked against the K distributions until a match is found, in which case the pixel receives a class label (background or foreground) according to that of the matched distribution. A match is defined as a pixel value within 2.5 times the standard deviation of a distribution. If no match is found, the pixel is considered as belonging to the foreground. In this case, a new distribution, centered on the pixel color or intensity, is initialized to replace the weakest distribution present in the mixture model. This new distribution is of high initial variance and low prior weight.

Once the new pixel value is classified, the model has to be updated. A standard method would be to use the *expectation maximization* algorithm. Unfortunately, that would be prohibitively computationally expensive. In [109, 110], Stauffer and Grimson give an on-line K-means approximation efficient enough to be performed in real-time on a standard VGA image (640×480 pixels).

After the computation of the foreground, foreground pixels are aggregated by a 8-connected component algorithm. This guarantees that a unique label is assigned to each connected region. Then each connected region is considered as a distinctive input for both the silhouettes description and silhouettes classification modules. Examples of candidate silhouettes extracted by the mixture of Gaussians algorithm are shown in Figure 3.2.

In the silhouettes description module, each candidate silhouette will be handled as if it was the unique region in the image. There are thus as many silhouettes as connected regions for which an algorithm has to decide whether or not the silhouette belongs to a known shape pattern.

Figure 3.2: Examples of silhouettes extracted with the Gaussian mixture model background subtraction technique.

3.4 Features based on a granulometric description by rectangles

Most surfacic descriptors can be described in terms of the theory of mathematical morphology. Therefore we will use this framework to describe our new feature set.

After a brief introduction to some notations, we will present the framework of granulometries that proved to be the starting point of our development. Then we will provide a formal description of our new operator.

3.4.1 Morphological operators on sets

Hereafter we briefly recall some definitions and notations used in mathematical morphology that serves as the framework to define our new feature space. Consider a space $\mathcal{E}$, which is the continuous Euclidean space $\mathbb{R}^n$ or the discrete space $\mathbb{Z}^n$, where $n \geq 1$ is an integer. Given a set $X \subseteq \mathcal{E}$ and a vector $b \in \mathcal{E}$, the translate X_b is defined by $X_b = \{x + b |\, x \in X\}$.

Let us take two subsets X and B of $\mathcal{E}$. We define the $X \oplus B$ (the *dilation* of X by B) and $X \ominus B$ (the *erosion* of X by B) respectively as

$$X \oplus B = \bigcup_{b \in B} X_b = \bigcup_{x \in X} B_x = \{x + b |\, x \in X,\, b \in B\} \tag{3.3}$$

$$X \ominus B = \bigcap_{b \in B} X_{-b} = \{p \in \mathcal{E} |\, B_p \subseteq X\}. \tag{3.4}$$

where B is referred to as the *structuring element*.

When X is eroded by B and then dilated by B, one may end up with a smaller set than the original set X. This set, denoted by $X \circ B$, is called the *opening* of X by B and defined by $X \circ B = (X \ominus B) \oplus B$. The geometric interpretation of an opening is that it is the union of all translated versions B included in X, or in mathematical terms, $X \circ B = \{B_p |\, p \in \mathcal{E},\, B_p \subseteq X\}$. Note that this geometrical interpretation is valid for a given set of fixed size. We have to enlarge it to encompass the notion of size or family of structuring elements, which leads us to granulometries.

3.4.2 Granulometries

The concept of granulometry, introduced by Matheron [78], is based on the following definition.

Let $\Psi = (\psi_\lambda)_{\lambda \geq 0}$ be a family of image transformations, depending on a parameter λ, that apply the following mapping: $\mathcal{E} \to \mathcal{E} : x \to \psi_\lambda(x)$. This family constitutes a granulometry if and only if the following properties are satisfied[2]:

$$\forall \lambda \geq 0, \qquad \psi_\lambda \text{ is increasing} \tag{3.5}$$

$$\forall \lambda \geq 0, \qquad \psi_\lambda \text{ is anti-extensive} \tag{3.6}$$

$$\forall \lambda \geq 0,\, \mu \geq 0, \qquad \psi_\mu \psi_\lambda = \psi_\lambda \psi_\mu = \psi_{\max(\lambda,\mu)}. \tag{3.7}$$

[2]An image transformation ψ_λ is *increasing* if for sets A and B of $\mathcal{E}$ such that $A \subseteq B$, $\psi_\lambda(A) \subseteq \psi_\lambda(B)$, and *anti-extensive* if for every $A \subseteq \mathcal{E}$, $\psi_\lambda(A) \subseteq A$.

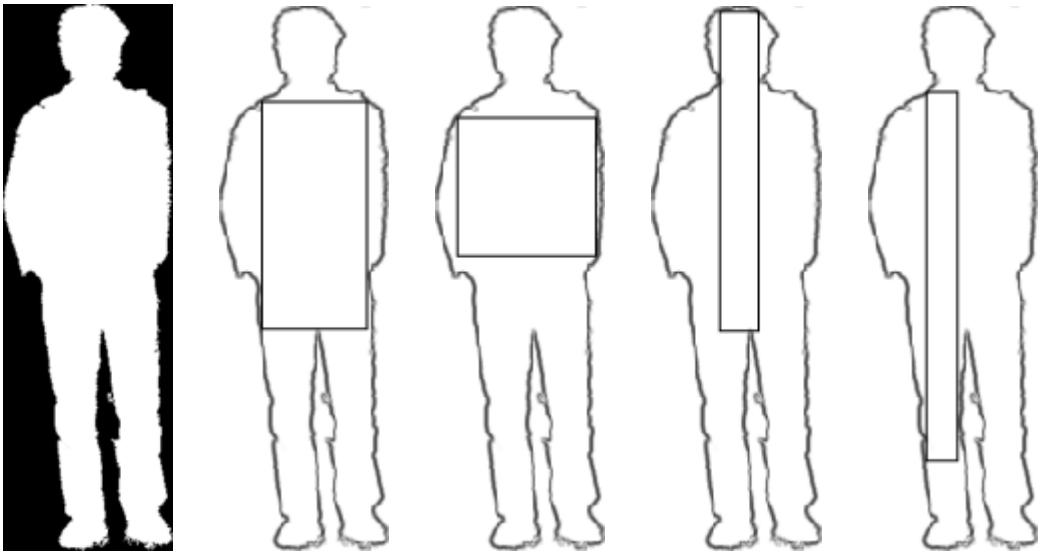


Figure 3.3: Examples of wedged maximal rectangles contained in a human silhouette.

The third property implies that, for every $\lambda \geq 0$, ψ_λ is an idempotent transformation, that is: $\psi_\lambda \psi_\lambda = \psi_\lambda$. As these properties reflect those of an opening, openings fit nicely in this framework as long as we can order the openings with a scalar. For example, assume that $X \circ rB$ is the opening by a ball of radius r. Then $\Psi = (\psi_r)_{r \geq 0} = (X \circ rB)_{r \geq 0}$ is a granulometry. Of particular interest are granulometries generated by openings by scaled versions of a convex structuring element.

Granulometries, and some measures taken of them, have been applied to the problems of texture classification [76], image segmentation, and more recently to the analysis of document images [3].

3.4.3 Granulometric curves and features

Maragos [76] has described several useful measurements for granulometries defined by a single scale factor: the *size distribution* and the *pattern spectrum*. The size distribution is a curve that gives the probability of a point belonging to an object to remain into that object after openings with respect to a size factor. The pattern spectrum is defined likewise as the derivative of the size distribution. All these measures are taken on operator residues driven by a one-dimensional criterion. They are neither applicable to a family of arbitrary structuring elements, nor able to produce uncorrelated multi-dimensional features. Therefore, we define a new operator that produces a *cover*.

Definition 1. ***[Cover]*** *Let $\mathcal{S}$ be a family of I arbitrary structuring elements $\mathcal{S} = \left\{S^{i \leq I}\right\}$. A* cover of a set X by $\mathcal{S}$ *is defined as the union of translated elements of $\mathcal{S}$ that are included in X such that*

$$C(X) = \left\{S_z^j |\, z \in \mathcal{E} \text{ and } S^j \in \mathcal{S}\right\} \tag{3.8}$$

where, if $S_{z'}^{j'}$ and $S_{z''}^{j''}$ both belong to $C(X)$, none of them is totally included in the other one.

In our application we consider the simplest two-dimensional opening which is of practical interest and practically tractable: an opening by a rectangular structuring element $B = mH \oplus nV$ where mH, nV respectively are m-wide horizontal and n-wide vertical segments. Based on the family $\mathcal{B}$ of all possible rectangle sizes, $C(X)$ will be the union of all the maximal[3] rectangles included in X. Such rectangles are shown in Figure 3.3. A fast algorithm for computing this cover is given in [115].

[3]A rectangle $S \subseteq X$ is *maximal* in X if $S = S'$ for every rectangle S' such that $S \subseteq S' \subseteq X$.

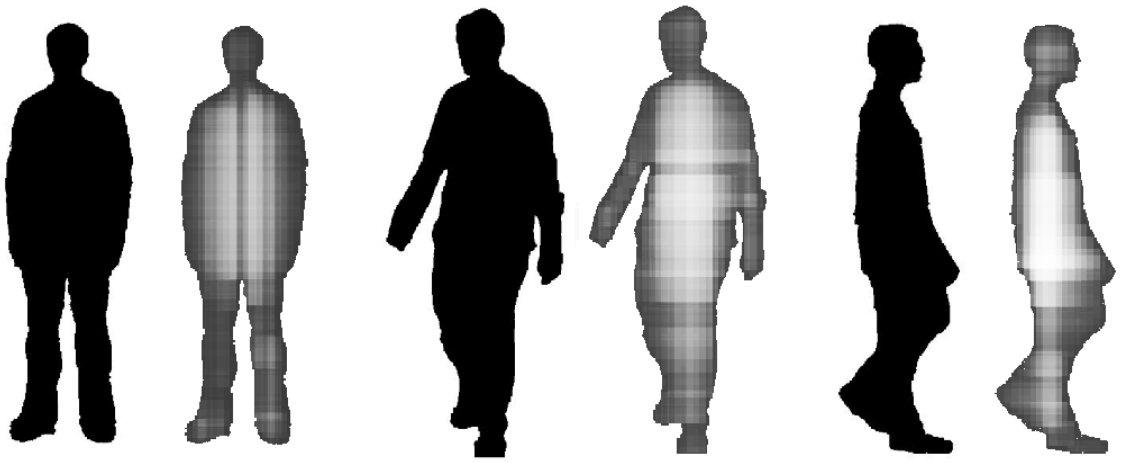


Figure 3.4: Examples of rectangles size distributions for human shaped silhouettes. The pixel intensities account for the number of overlapping rectangles that cover each location in the image.

The main advantage of using a cover is that we have a family of structuring elements, describing the surface of an object X, whose members might overlap but all of them uniquely fit somewhere inside X.

The next step is to extract features from the cover. Since B are rectangles, features like width, height, perimeter, and area spring to mind. However, for classification purposes, care should be taken to avoid redundant features. Redundancy would not increase the performance and could even be counterproductive. For example, Hadwiger [40] has shown that any continuous, additive, and translation and rotation invariant measure on a set X must be a linear combination of the perimeter, the area, and the Euler-Poincaré number of X.

Since translations and scales have some significance in the analysis of silhouettes, we reintroduce them. In our set of features, we take the positions, relative to the center of X, of the elements of $C(X)$. Finally, we select 6 features on any *element* of $C(X)$[4]:

- 2 coordinates of the center of the rectangle, computed relatively to the barycenter of the silhouette,
- width of the rectangle, which is normalized with respect to the width of the bounding box of the silhouette,
- height of the rectangle, which is normalized with respect to the height of the bounding box of the silhouette,
- area of the rectangle, which is normalized with respect to the area of the whole silhouette, and
- percentage, to the area of the considered rectangle, of pixels that are covered by the considered rectangle exclusively.

3.5 Silhouettes classification

Once the set of all the features describing a silhouette has been extracted (see Figure 3.4 for an illustration of the density of covered pixels), it becomes possible to exploit a machine learning algorithm to map this set into a class. Indeed, such mappings are especially hard to derive by hand and should be *learned* by the system. In our framework, as we are interested in the detection of

[4]Note that the perimeter derive from the width and height of a rectangle so that it is unnecessary to add it to the list of features.

human silhouettes, only two classes of interest are considered: the class of the human silhouettes, and the class of any other silhouette.

The machine learning approach requires to take two difficulties into account: (1) there is a need of a classifier which have excellent generalization abilities and which is not subject to overfitting, and (2) we must define a way to apply this classifier on a set of rectangles, the number of which may widely vary between silhouettes.

To this aim, we propose to use *EXTremely RAndomized* trees (extra-trees). The extra-trees algorithm is a fast, yet accurate and versatile, machine learning algorithm [36]. The reasons for using extra-trees in our context are threefold: (1) extra-trees have proven to be successful for solving some color image classification tasks [77], (2) they form a non-parametric function approximation architecture, which do not require previous knowledge, and (3) they have a low bias, a low variance, and good performances in generalization.

3.5.1 Classification based on extremely randomized trees

In this section, we first describe how extra-trees can be used to map a single rectangle to a class. We then explain how to map a *set* of rectangles to a class. We restrict our study of extra-trees to the case where all the input attributes are numerals, which is obviously the case for our rectangular features. Indeed, as mentioned earlier, the input attributes for the rectangles are their width, height, area, two relative center coordinates, and an information about the cover.

Extra-trees are an extension of ensemble methods such as *bagging* [14] and *random forests* [15]. They consist of a forest of N independent binary decision trees. These trees are built with a highly (extremely) random induction algorithm. Each of their internal nodes is labeled by a threshold on one of the input attributes, which is to be tested in that node. As for the leaves, they are labeled by the classification output labels. To classify a rectangle with an extra-trees model, this rectangle is independently classified by each tree. This is achieved by starting at the root node, then progressing down the tree according to the result of the tests on the threshold found during the descent, until a leaf is reached. Doing so, each tree votes for a class. Finally, the class that obtains the majority of votes is assigned to the rectangle.

We use the implementation described in [49]. The trees are built in a top-down fashion, by successively splitting the leaf nodes where the output variable does vary. If a leaf node does not comprise a sufficient number of training samples (at least 6 in our case), the node is labeled according to the majority class among its training samples. The split of a node consists of, for each input variable, computing the variation bounds of the variable and choosing uniformly one random threshold between those bounds. Once a threshold has been chosen for every input variable, the split that gives the best information-theoretic score on the classification output is kept. This guarantees that the variance in the model is reduced (thanks to the presence of a forest of independent trees), as well as the bias (thanks to the random selection of the thresholds), while taking advantage of an information measure that guides the search for good splits.

3.5.2 Classification of the silhouettes

We have just described the process of classifying *one* rectangle. But we describe a silhouette X by its cover $C(X)$, which is a *set* of rectangles. Furthermore, two distinct silhouettes can have a different number of rectangles in their cover. We must therefore introduce a meta-rule over the extra-trees for mapping a set $C(X)$ to a class. In this work, we exploit an idea that is similar to that of Marée *et al.*, which was used in the context of image classification [77].

Let M be a fixed positive integer. Given the set $C(X)$ of rectangles that shapes the silhouette X, we select M rectangles inside this set, which induces a subset $C_M(X) \subseteq C(X)$. Then, we apply the extra-trees model on each rectangle inside $C_M(X)$. This process generates one vote per rectangle. Finally, the silhouette X is assigned to the class that has obtained either the majority of the votes, or a sufficient ratio of the votes. This minimal ratio can be chosen to be greater than the inverse of the number of classes, if a high level of confidence is required. Below, we refer to this minimal ratio of the votes to be reached as "classification threshold".

Figure 3.5: A few examples of negative instances contained in the training dataset.

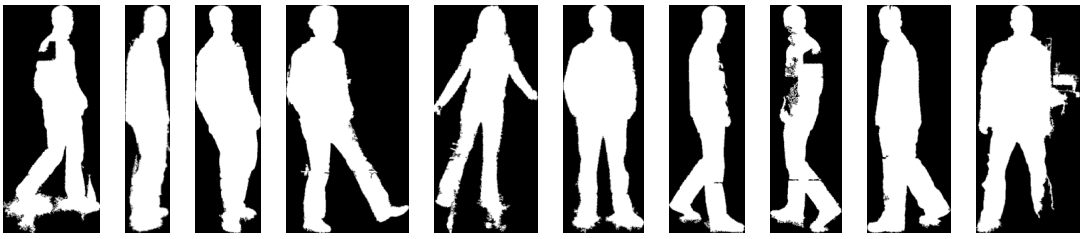


Figure 3.6: Subset of the positive instances contained in the training dataset.

3.6 Experimental results

3.6.1 Dataset collection

As mentioned in the introduction, we focus our experiments on the detection of human silhouettes in a video stream. The extra-trees are trained on a dataset of silhouettes that contains both silhouettes of human bodies and silhouettes of other kinds of objects. We feed the learning set with a large number of instances for each of these two classes.

Some instances of non-human silhouettes, called negative instances, are shown in Figure 3.5.

The negative samples are the union of the non-human silhouettes that were extracted from a live video stream by the background subtraction technique presented in Section 3.3, and of images that were taken from the COIL-100 database [85]. There are about 12,000 images in this dataset. As for the positive instances, we have about 3,500 human silhouettes. Some of them are represented in Figure 3.6.

Those two datasets are converted to a database that is fed into the extra-trees learning algorithm (cf. Section 3.5).

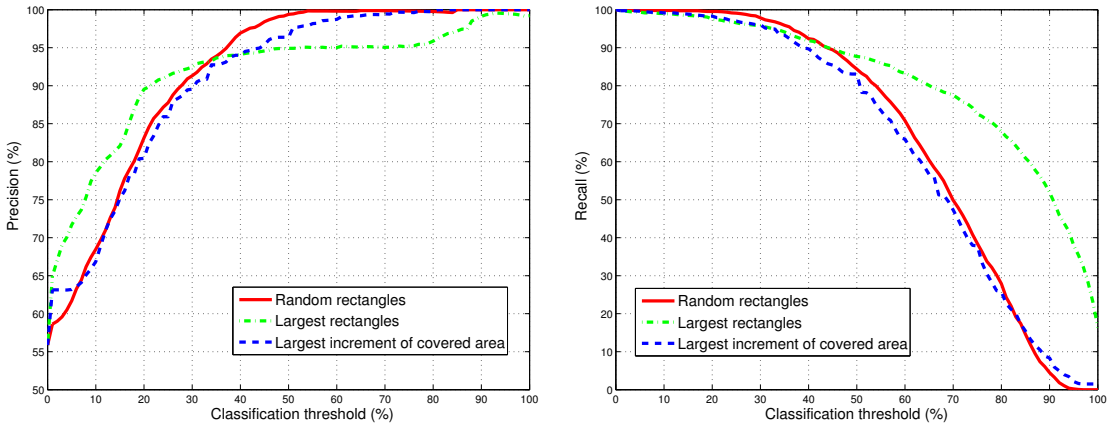
3.6.2 Choice of a rectangle selection policy

As stated in Section 3.5.2, we select and classify a subset of M rectangles chosen in the cover $C(X)$ of a silhouette X. Consequently, we have to determine a rectangle selection policy. Below, we consider three selection policies and evaluate them using our database that we divided into two parts of equal size. We use the first part to train the system and the second part to evaluate its performance.

The three selection policies that we consider are:

1. the selection of the M largest rectangles of $C(X)$,
2. the selection of the M rectangles of $C(X)$ that contain the maximum number of pixels not contained in any other rectangle of $C(X)$,
3. and the selection of M rectangles of $C(X)$ at random.

The precision and recall curves for the three rectangle selection policies are shown in Figure 3.7. We see that if a high precision is required, the best option is to use the random selection policy. Furthermore, from a computational point of view, the random selection of the rectangles is faster.



(a) Precision (b) Recall

Figure 3.7: Precision and recall of three rectangle selection policies for a number of selected rectangles $M = 100$ and for a classification threshold ranging from 0 to 100%.

If a high recall is required, the selection of the M largest rectangles is a good alternative, but at the cost of large number of false human detections. In this work, we choose to minimize the number of false person detections and opt for the random policy.

3.6.3 Choice of an appropriate number of rectangles

In order to determine an educated value for the number M of rectangles randomly selected in $C(X)$, we tested our technique for M ranging from 10 to 500. The results are shown in Figure 3.8. We observe that the best results are obtained for a number M of selected rectangles ranging from 50 to 200. It must be noted that the required number of rectangles needed to cover a significant part of the silhouette X is a function of the total number of rectangles contained in $C(X)$, which is itself a function of the area of X. The silhouettes considered in these tests were mostly extracted from video streams of 640×480 pixels.

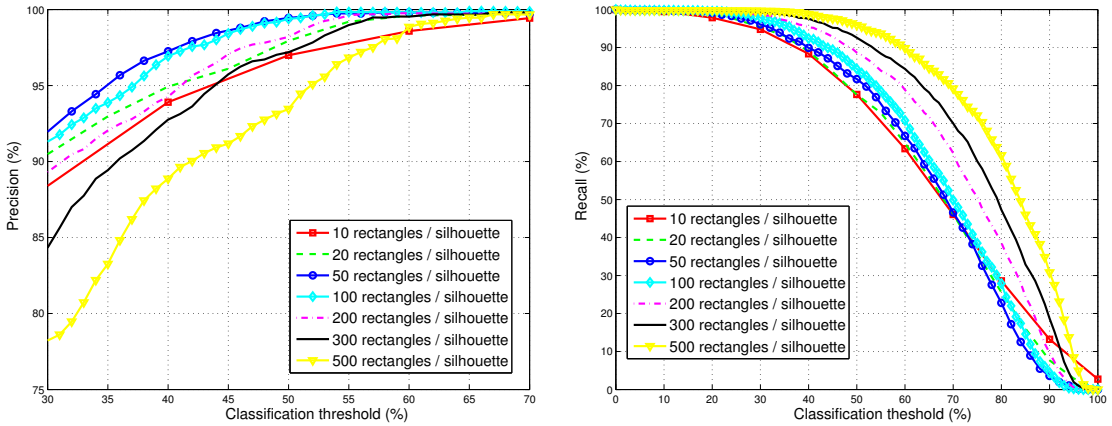
3.6.4 Comparison with another surfacic silhouette descriptor

In order to demonstrate the benefit of the use of the cover by rectangles, we compare the results of our technique with those of an alternative technique. In [44], Hu introduced a set of 7 image moments that are invariant under translation, changes in scale, and rotation. We compare our technique with a method that uses the set of Hu's image moments as a silhouette descriptor.

This alternative technique uses the same silhouettes as our technique. Furthermore, the sets of moments extracted from the silhouettes are classified with the same machine algorithm as our method. The results of both techniques are shown in Figure 3.9. We see that our method based on the cover by rectangles of the silhouettes outperforms the alternative technique based on Hu's image moments. This demonstrates the interest of using the cover by rectangles as a silhouette descriptor and convinced us to use the cover by rectangles during the design of our gait recognition technique described in chapter 4.

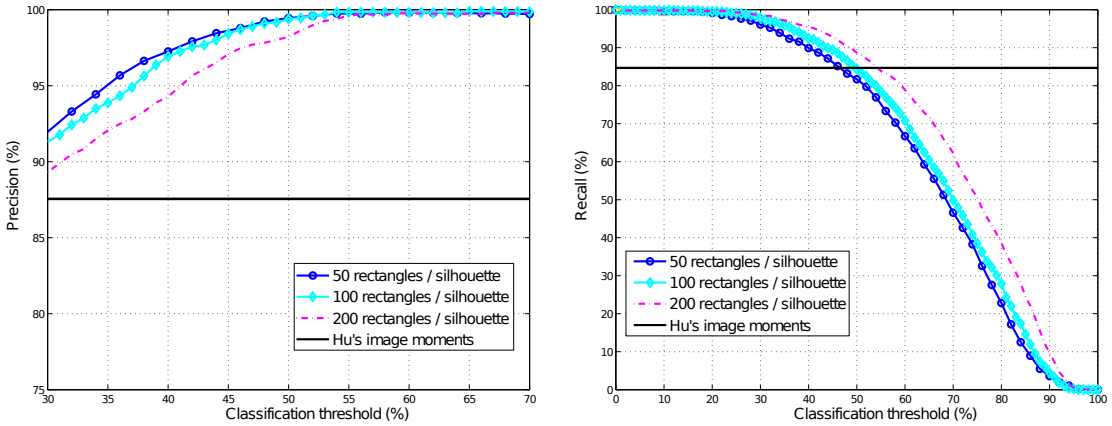
3.6.5 Tests on real-world images

We have tested our algorithms on a color video stream of 640×480 pixels that was captured with a FireWire CCD camera. The whole process (including silhouettes extraction, description, and classification) was carried out at approximately five frames per second on a 3.4 GHz Pentium IV computer.



(a) Precision (b) Recall

Figure 3.8: Precision and recall curves of a random rectangle selection policy for M ranging from 10 to 500 and for a classification threshold ranging from 0 to 100%.



(a) Precision (b) Recall

Figure 3.9: Precision and recall curves for a random rectangle selection policy with M ranging from 50 to 200 and for an alternative technique that uses Hu's image moments as a silhouette descriptor.

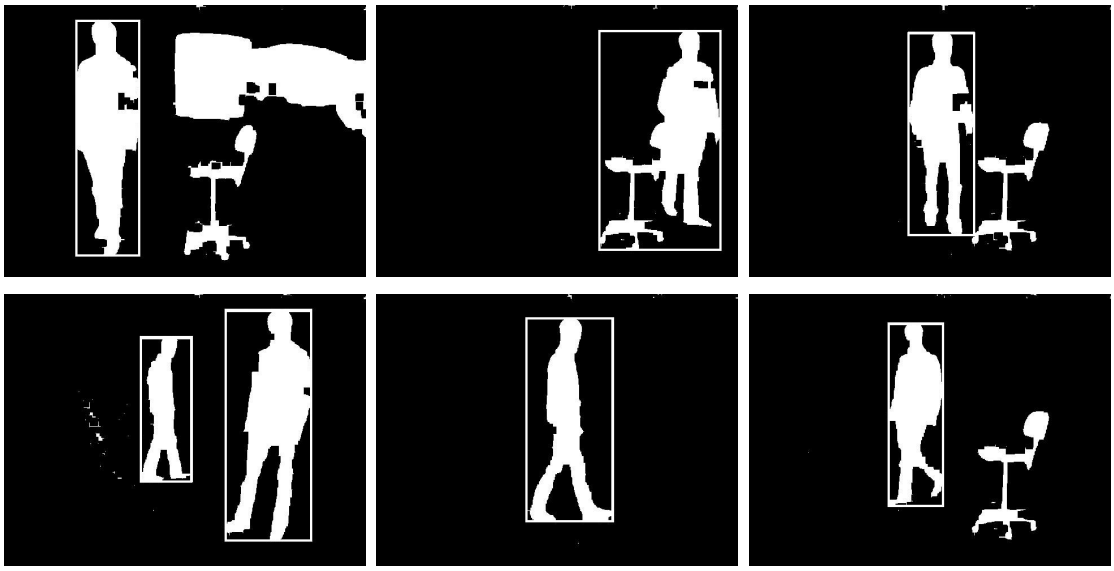


Figure 3.10: Examples of silhouettes classified correctly. A white frame around an object indicates that the system classifies it as a human silhouette.

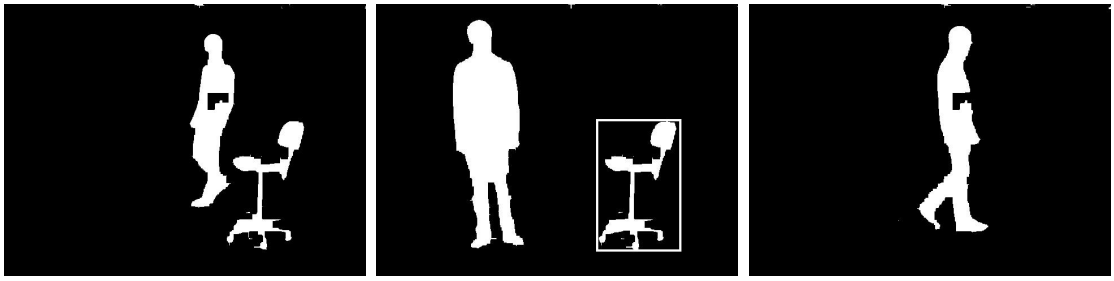


Figure 3.11: Examples of misclassified silhouettes.

The detection of human silhouettes is very robust since the number of correct classifications largely outnumbers misclassifications, although we ignored any correlation between successive frames. Example images of correct (resp. wrong) classifications are shown in Figure 3.10 (resp. in Figure 3.11). These results demonstrate that on single images, our system is able to recognize specific silhouettes in a semi-controlled environment. In the next chapter, we will introduce an algorithm able to classify *series* of silhouettes on the basis of their cover by rectangles.

3.7 Conclusions

In this chapter, we proposed a new system for the real-time detection and classification of binary silhouettes. Using a standard background subtraction algorithm, the silhouettes are extracted from an input video stream. Each silhouette is then treated by a new kind of granulometric filter that produces a morphological cover of the silhouette. This filter characterizes the silhouette as the set of all the maximal rectangles that can be wedged inside it. One of our major achievements is to have managed to implement the feature extraction step in real-time, which is uncommon for surface-based descriptors. The rectangle features are then fed into an extra-trees classifier that assigns a class label to each detected silhouette. Thanks to the simple tree-based structure of the extra-trees model, the classification step is also very fast. Consequently, the whole process that consists of the detection, analysis, and classification of the silhouettes can be carried out in real-time on a common computer.

In our experiments, we determined an appropriate rectangle selection policy and compared our results with those of an alternative technique based on a widely used surfacic silhouette descriptor. We showed that the use of the cover by rectangles to characterize the silhouettes leads to a better classification. Finally, empirical results that consisted in the application of our method to images captured with a CCD camera, which was put in an environment unknown to the learning process, show that our method manages to detect human silhouettes with a high level of confidence.

Chapter 4

Gait Recognition: Frontal-view gait recognition by intra- and inter-frame rectangle size distribution

This chapter is an extended version of our article on frontal gait recognition published in Pattern Recognition Letters in 2009 [6]. The first part of this chapter proposes a method able to recognize people from their gait using a frontal camera. It must be stated that the first implementation of the technique was made with the invaluable help of Steve Frécinaux. The last part of this chapter relates an original application of the method to the control of the access to a secure area that we carried out in collaboration with BEA. This work has been achieved in close collaboration with Sébastien Pierard and will be published in the proceedings of the 2010 Advanced Concepts for Intelligent Vision Systems conference [96].

Abstract

Current trends seem to accredit gait as a sensible biometric feature for human identification, at least in a multimodal system. In addition to being a robust feature, gait is hard to fake and requires no cooperation from the user. As in many video systems, the recognition confidence relies on the angle of view of the camera and on the lightening conditions, inducing a sensitivity to operational conditions that one may wish to lower.

In this chapter we present an efficient approach capable of recognizing people in frontal-view video sequences. The approach uses an intra-frame description of silhouettes which consists of a set of rectangles that will fit into any closed silhouette. A dynamic, inter-frame, dimension is then added by aggregating the size distributions of these rectangles over multiple successive frames. For each new frame, the inter-frame gait signature is updated and used to estimate the identity of the person detected in the scene. Finally, in order to smooth the decision on the identity, a majority vote is applied to previous results. We provide experimental results and discuss the accuracy of the classification for our own database of 21 known persons, and for a public database of 25 persons. In the last part of this chapter, we describe, and provide the results, of an original application of our algorithm to the monitoring of the access to a secure area.

4.1 Introduction

The number of video-surveillance cameras has increased dramatically over the last few years. It has therefore become unrealistic to process manually or even visually the gigantic amount of information gathered by surveillance cameras, which explains why the automation of real-time visual surveillance tasks is currently one of the most active topics in computer vision. Visual surveillance has a wide spectrum of promising applications, including control of access to certain areas, human identification, crowd flux statistics, detection of anomalous behaviors, etc [45]. This chapter focuses on one of these tasks: automatic human identification.

Automatic human identification can be achieved through a variety of biometrics using different kinds of sensors: fingerprint readers, iris scanners, microphones for voice recognition, and video cameras. One advantage of video cameras is that they are not intrusive; also subjects can be filmed without their cooperation. Face recognition through the use of a video camera is a widely used biometric, although its efficiency is conditioned by the need for a relatively constrained image of the person's face. Unconstrained face recognition is possible (see [133]) but is almost useless for strong identification in practice. Asking a person to cooperate can also be an issue; not everyone is going to help the system. Gait recognition is therefore a viable alternative; in this case, it is neither necessary to restrict the field of view to constrained environment, nor to ask for cooperation. Gait recognition is not (yet?) as effective as the best face recognition algorithm but, acting as a complementary form of identification, it might reinforce a decision made in a multi-modal biometric system.

Gait as a biometric is quite a recent topic for discussion, which has gained in popularity since its introduction in [89]. Its robustness against poor imaging conditions makes it applicable to a wide range of real-world scenarios. Images can be acquired from a great distance, or in changing illumination conditions (even outdoor, as shown in [68]). Furthermore, absolutely no kind of cooperation from the subjects is required. Gait is also difficult to fake: it has been shown in [35] that impersonation attacks are not a real threat unless attackers have the knowledge of their closest match in the database of the gait authentification system. Yet, gait recognition techniques are still not accurate enough to use gait as the sole biometric of a real surveillance system. These recognition techniques are better used to reinforce a decision in a multi-modal biometric system (see [68, 87, 88, 103, 134]).

Gait recognition techniques are usually classified in two categories: model-based and holistic/silhouette approaches [86].

Model-based approaches make use of explicit gait models whose parameters are to be estimated by processing sequences of images, hereafter referred to as image frames or frames. The identification is performed entirely on the basis of the estimated values of the explicit gait model. Model-based approaches are generally scale and view invariant, as long as the parameters estimation is feasible given the imaging configuration. This is a major advantage, since training conditions are likely to differ from conditions of practical use. On the other hand, these methods often need high definition images in order to work properly. They also exhibit a significantly higher computational cost. An articulated model is used in [135] to introduce strong prior knowledge in a Bayesian framework for extracting human gait. High quality laboratory images are needed to estimate the statistics of the parameters but the the Bayesian framework allows accurate gait extraction from noisy outdoor scenes. Techniques in this category include modeling the thighs as a pair of thick lines, as in [25], modeling the silhouette of a walking person as a group of seven ellipses as in [37], or modeling the legs as two penduli joined in series, as in [130].

Holistic approaches do not assume any explicit model for the walking human. They extract information directly from the gait image sequences. Gait signatures are, for example derived from time series of binary silhouettes extracted from the original sequence with a background subtraction algorithm. This brings a suitable invariance to color, texture or illumination conditions (assuming that the used background subtraction algorithm is robust). A simple approach that uses areas of raw (re-sized) silhouettes as a gait signature is described in [34]. In [41], silhouettes are averaged over complete gait cycles to compute Gait Energy Images (GEI) and in [111], Gabor-function-based image decompositions of averaged silhouettes are used for gait representation. In

both [41] and [111], a linear subspace method is then used to extract features vectors from the average templates. Other dimensionality reduction algorithms are presented in [129] for average-template-based gait recognition. Modified versions of the GEI are proposed in the literature such as the *frame difference GEI* in [19] that copes with incomplete silhouettes, or the *enhanced GEI* that focuses on the parts of the GEI that reflects the walking manner of an individual [131]. In [60], average templates are combined with contour templates to perform recognition.

The contours of silhouettes have also been used, either directly [123] or through their Fourier descriptors [84]. An angular transform of the silhouette is proposed in [12]. This is said to be more robust than the raw contour descriptions. Procrustes shape analysis of the contours is used in [122] to obtain the mean shape of a series of silhouettes which serves as a gait signature. Gait dynamics can be captured using principal components analysis of self-similarity plots [8]. In [67], gait dynamics normalization is performed using a population HMM whose states represent gait stances over a complete gait cycle. Feature vectors derived from the binary silhouettes can also be used to train HMM's, as in [54] and [20]. In [118], a nonparametric model of the shape deformation of a person's silhouette is proposed and used as a discriminative feature for gait recognition.

Other authors have used horizontal and vertical projections of the silhouettes [53]. In [10] the Radon transform is used to consider simultaneously all the projections along lines of every directions. Feature vectors are derived from Radon template using Linear Discriminant Analysis (LDA). A genetic algorithm is used in [47] to fuse information from different kind of features: two generalized Radon transforms and the weighted Krawtchouk moments. In [66], time series of horizontal and vertical projections of silhouettes are treated as *frieze* patterns. The framework of frieze patterns leads the authors to estimate the viewing direction of the walking humans and to align gait sequences from similar viewpoints both spatially and over time. The identification is then performed using cross-correlation and nearest neighbor classification between frieze patterns. In [62], a similar algorithm is used to compare frieze patterns of frame differences between a key silhouette and a series of successive silhouettes. The method is claimed to be more robust to silhouette differences between the training and test sets.

Gait recognition is formulated as a channel coding problem with noisy side information at the decoder in [2]. Error correcting codes are employed to recognize users using features based the radial and the circular integration transforms of their silhouettes.

Nearly all silhouette-based approaches are designed to deal with image frames captured from the side of a person. While it is reasonable to assume that the lateral view captures an appropriate amount of gait and walking information, it is not easy to capture these image frames in practical scenarios. In order to obtain a sufficiently long sequence of images of a person walking (containing several gait cycles), cameras need to be put at a long distance. This hinders recognition, since small silhouettes are hard to discriminate. In hallways, frames are rarely captured from the side, but from the front or the back of the walker. Front-view cameras, as opposed to lateral-view cameras, capture longer sequences of walkers, which results in more gait cycles. However front-view cameras are thought to be less efficient for gait recognition as they capture geometric and scale transformations of the silhouettes. But the human capacity to recognize people using only a frontal view of their walking silhouettes tends to prove that a frontal view contains enough information to perform automatic recognition. This is confirmed by Soriano et al. [107]. In an article in which gait signatures are derived from series of Freeman encoding of the re-sized silhouette shape, these authors showed that frontal view gait recognition is possible [107].

In [46], the gait template of a walking human is computed by averaging the corresponding binary silhouettes. The classification is then achieved using a nearest neighbor technique. The authors use the MoBo database [38] from the CMU to compare the classification results obtained by their method with sequences captured from different viewpoints. The best single viewpoint results are obtained using the frontal view. But better classification scores are achieved by combining the frontal view with the lateral view.

This chapter presents a gait recognition algorithm capable of recognizing persons from image frames captured in real-time with surveillance cameras located in hallways. Unlike many techniques in the literature which process complete gait sequences, our algorithm identifies a previously known person as soon as it obtains a complete gait cycle, which accounts for about 1 second or 25

(a) Lateral view (b) Frontal view

Figure 4.1: Lateral and frontal views of a walker.

frames. Requirements for our method are that (1) low image resolution (like 640×480) suffice, (2) walkers can wander at quite a long distance from the cameras, and (3) the algorithm should run in real time on any computer.

For noisy surveillance video frames, a precise detection of moving objects and their contours is difficult. In order to achieve a better resilience to noise, we chose a surfacic representation of the silhouettes in terms of a descriptor called "Cover by Rectangles", introduced in [4]. This descriptor provides a piecewise surfacic description of silhouettes which, unlike horizontal and vertical projections, is reversible and therefore does not induce any information loss. In addition, covers by rectangles limits the effect of noise to a local neighborhood as noise will impact locally on the description of the silhouette, in contrast with global surfacic measures. Section 4.2 derives a new silhouette representation based on the cover by rectangles approach. This representation serves to characterize gait silhouettes for each frame separately; we therefore call this an intra-frame descriptor. Section 4.2 also explains how we consider temporal and dynamic information by introducing inter-frame dependencies in order to derive a gait signature. We describe the complete gait identification algorithm in Section 4.3. Experimental results and an evaluation of our method are presented in Section 4.4. We show that gait recognition is possible, efficient, and achievable in real time, even for front-view video frames. In Section 4.5, we describe an original application of our technique in the context of the control of access to a secure area.

4.2 A surfacic gait representation

In order to identify a walking person, a time series of his silhouettes is extracted from the raw video frames, at a rate of one silhouette per frame. For each frame, the silhouette is converted into a set of features, which are used to update a gait signature. The gait signature is fed into a classifier which will output the class label corresponding to a particular person. Hereafter we present the intra-frame description of a silhouette.

4.2.1 Cover by rectangles of a binary silhouette

The cover by rectangles, proposed in chapter 3, is a morphological descriptor. Consider a binary silhouette S. The cover by rectangles, denoted $C(S)$, is defined as the union of all the maximal

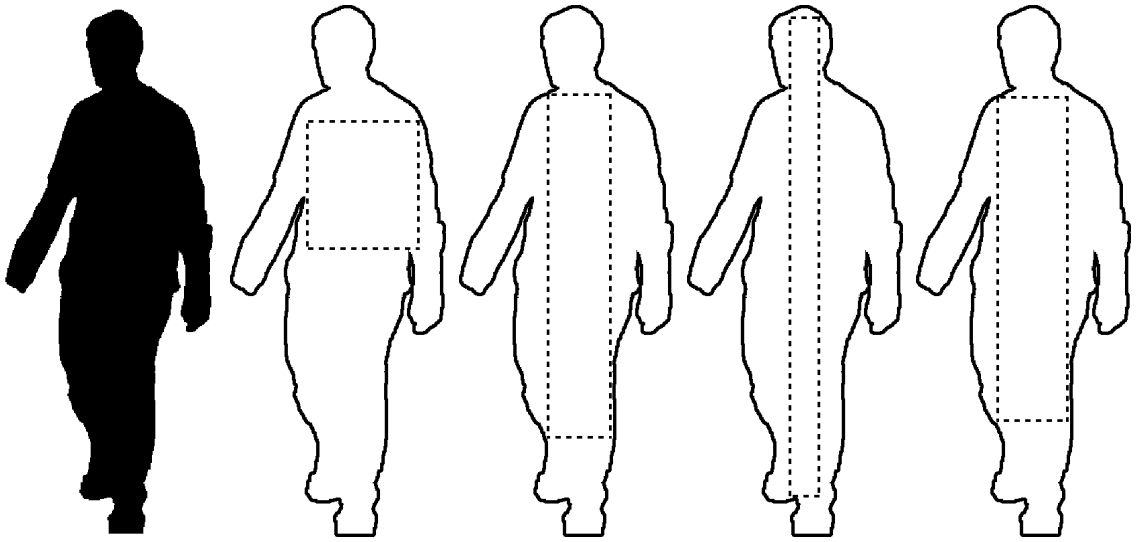


Figure 4.2: The cover by rectangles $C(S)$ is the union of all the maximal rectangles that can be wedged inside the silhouette.

rectangles that can fit inside S (see Figure 4.2 for an example). This union is unique and the cover $C(S)$ has the two following useful properties: (1) the elements of the set overlap each other, introducing redundancy (robustness), and (2) when displayed in the frame, the union of all rectangles reconstructs S so that no information is ever lost.

Other morphological surfacic descriptors, such as the morphological skeleton [102], have been developed to represent shapes. However, since they provide an isotropic description of the silhouettes through, for example, the union of open balls included in S, they are unsuited for the description of gait. Moreover, it is important to ensure that a local modification of the silhouette does not lead to a global change in its description. Figure 4.3 compares the effect of a slight modification of the shape in the case of the skeleton and features (widths or heights) derived from the rectangles of $C(S)$. In Section 4.4.2, we show that a gait signature based on the cover by rectangles of the silhouettes of a walking human is robust and allows the correct identification of people from noisy silhouettes (see Figure 4.6) through a set of experiments.

4.2.2 Rectangle size probability distributions

The number of wedged maximal rectangles that will fit inside a binary silhouette can be very high (more than a thousand). It is thus impractical to use all the rectangles directly as a set of features. In order to find a more compact representation, we can operate on one of the size distribution densities, as shown in Figure 4.4. These distributions offer different but suitable interpretations of a silhouette. For example, the largest number of rectangles containing a given pixel is to be found inside the torso (Figure 4.4(b)), and the tallest rectangles pass through both the legs and the head (Figure 4.4(d)).

As can be seen, much of the information resides in the distributions of the normalized sizes (width or height). These distributions can be estimated as a discrete histogram whose bins correspond to the ratios of rectangles that fall within given size intervals.

From a formal point of view, let α be the cardinality of a cover by rectangles $C(S)$, *that is* $\alpha = \sharp\{C(S)\}$. We index the rectangles of $C(S)$ with a parameter d, so that R_d $(d = 1, \ldots, \alpha)$ are the rectangles of $C(S)$. The width and height of R_d are respectively denoted by w_d and h_d; they are upper-bounded by w^{max} and h^{max}: $\forall d$, $w_d \leq w^{max}$ and $h_d \leq h^{max}$. In order to build histograms, we partition the widths and heights of the rectangles R_d respectively into M bins

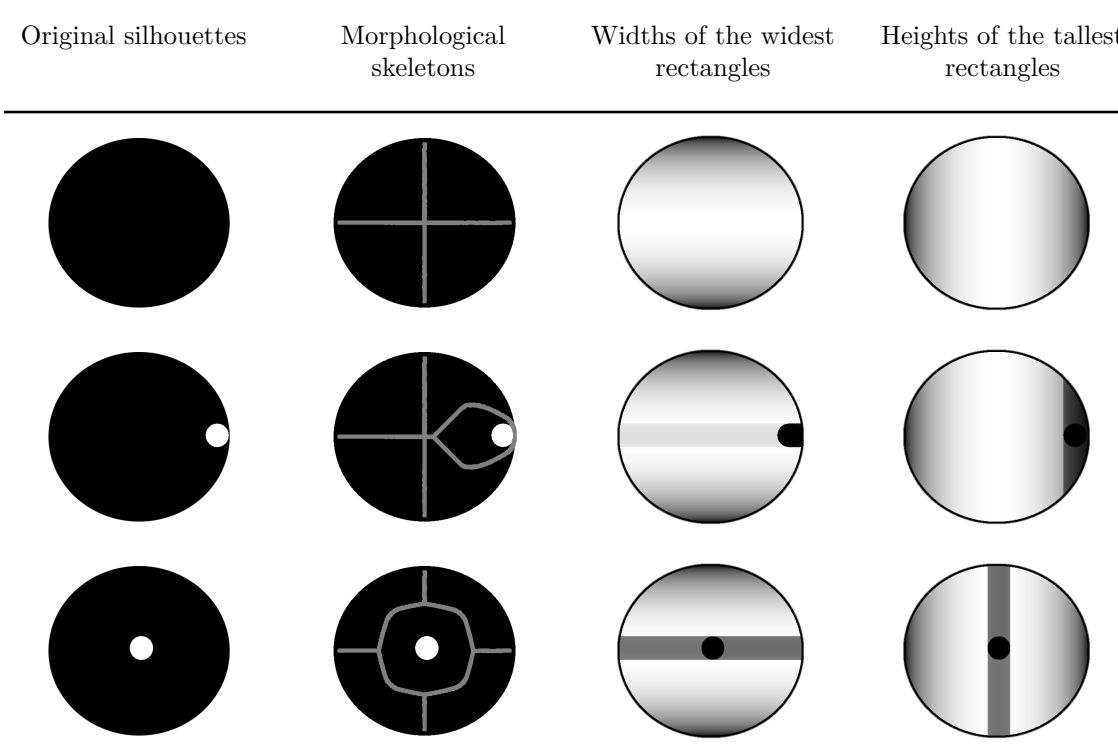
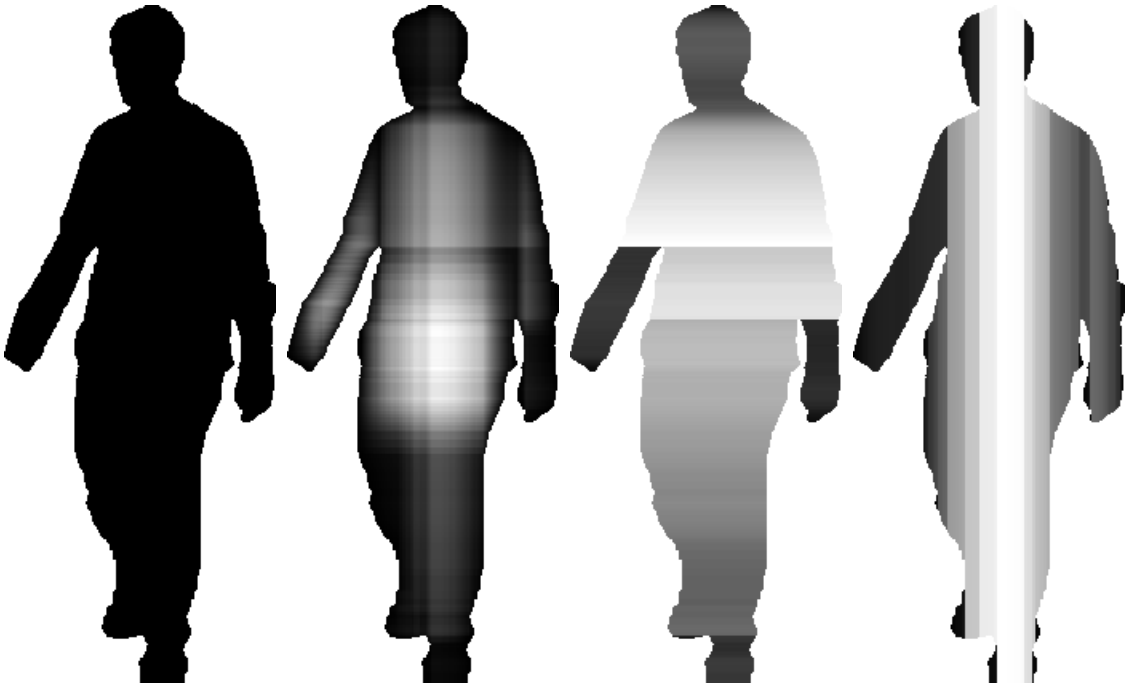


Figure 4.3: The first column shows three original images. The morphological skeletons (shown in gray in the second column) are modified by the presence of a small hole in the silhouette: a local perturbation leads to a global modification of the skeleton. The images the two right-hand columns represent the size distributions of the rectangles contained in $C(S)$. In these images, the gray level of pixels is proportional to the width (resp. height) of the widest (resp. tallest) rectangle comprising the given pixel.



(a) (b) (c) (d)

Figure 4.4: Illustration of several size distributions based on the description provided by the cover $C(S)$ of a binary silhouette S. A gray level of pixel p in images (b), (c), and (d) displays respectively the density of rectangles, the width of the widest rectangle, and the height of the tallest rectangle where all these rectangles contain pixel p.

$B^W(i)$ and N bins $B^H(j)$

$$B^W(i) = \left]i\frac{w^{max}}{M},\ (i+1)\frac{w^{max}}{M}\right] \tag{4.1}$$

$$B^H(j) = \left]j\frac{h^{max}}{N},\ (j+1)\frac{h^{max}}{N}\right] \tag{4.2}$$

where $i = 0, \ldots, M-1$ and $j = 0, \ldots, N-1$.

Following the above notations, we define the histogram $\text{hist}^W(i)$ of the normalized widths as

$$\text{hist}^W(i) = \frac{1}{\alpha} \sharp \left\{R_d |\, w_d \in B^W(i)\right\}, \tag{4.3}$$

the histogram of the normalized heights similarly as

$$\text{hist}^H(j) = \frac{1}{\alpha} \sharp \left\{R_d |\, h_d \in B^H(j)\right\}, \tag{4.4}$$

and the two-dimensional histogram $\text{hist}^{W\times H}(i,j)$ as

$$\text{hist}^{W\times H}(i,j) = \frac{1}{\alpha} \sharp \left\{R_d | w_d \in B^W(i),\, h_d \in B^H(j)\right\}. \tag{4.5}$$

Note that, according to equations 4.1 and 4.2, these histograms are normalized with respect to either the largest rectangle of the cover of the silhouette, or the widest, or both. In a continuous space, they would be scale invariant. Such a normalization might seem counter-intuitive; much of the interpretation of the motion of a gait derives from the size of a silhouette, and it would not be good for frontal cameras to lose motion information. A finer analysis shows however that size information is still present in a normalized histogram. Indeed the cover of a scaled down version of a silhouette S contains fewer rectangles (α is always lower than the number of contour points) than its original counterpart. Therefore the histograms have a distribution that adapts to both the shape and the size of a silhouette. In addition, if noise is added to the contour of the silhouette, it will modify the positions of the rectangles but not so much their size or number.

Of the three $\text{hist}^W(i)$, $\text{hist}^H(j)$, $\text{hist}^{W\times H}(i,j)$ histograms, the last one best describes S. However, its dimensionality is proportional to the product of the numbers of bins ($M \times N$), which is acceptable for an intra-frame description but might be too high for embedded systems if the features are to be fed into a classifier for inter-frame gait recognition. In order to solve this tractability issue, we introduce the composite histogram $\text{hist}^{W+H}(k)$ with $k = 0, \ldots, M+N-1$ defined as the strict concatenation of $\text{hist}^W(i)$ and $\text{hist}^H(j)$. $\text{hist}^{W+H}(k)$ has a dimensionality of $M+N$, and accounts for both the vertical and horizontal characteristics of the silhouette. Experiments detailed in Section 4.4 show that both $\text{hist}^{W\times H}(i,j)$ and $\text{hist}^{W+H}(k)$ are suitable descriptors.

4.2.3 Gait as an inter-frame rectangle distribution

So far we have considered a single intra-frame silhouette, but a gait sequence is a temporal series of binary silhouettes. In order to capture the dynamics of a walking person we introduce an inter-frame dependency by defining a gait signature based on the temporal series of the silhouettes S of a walker. We assume that t refers to the time of the current frame, and that $\text{hist}(i,\ j,\ t)$ is a histogram for S at time t. We introduce two gait signatures, denoted $\mathcal{G}$, which consist of n-uples of L consecutive histograms. We propose the following gait signature

$$\mathcal{G}^{W\times H}(i,j,t) = \left\{\text{hist}^{W\times H}(i,\ j,\ t-(L-1)), \ldots,\ \text{hist}^{W\times H}(i,\ j,\ t-1), \text{hist}^{W\times H}(i,\ j,\ t)\right\}, \tag{4.6}$$

and a shortened version as

$$\mathcal{G}^{W+H}(k,t) = \left\{\text{hist}^{W+H}(k,\ t-(L-1)), \ldots,\ \text{hist}^{W+H}(k,\ t-1), \text{hist}^{W+H}(k,\ t)\right\}. \tag{4.7}$$

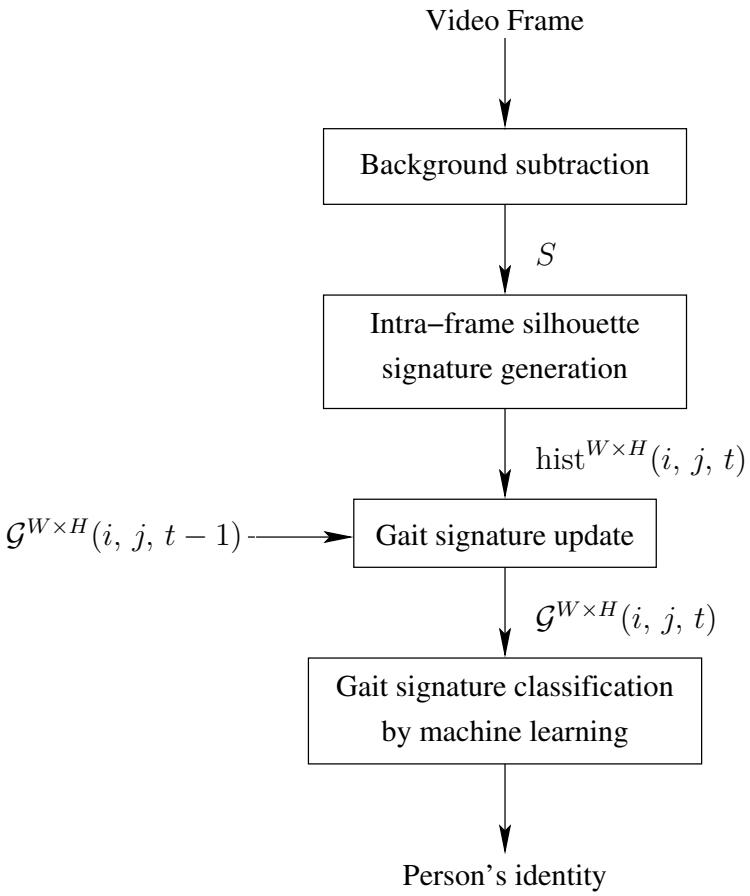


Figure 4.5: Steps of our gait recognition algorithm.

4.3 Gait recognition algorithm

The gait recognition process is shown in Figure 4.5. For every frame of a gait sequence, it predicts the identity of the walking human. The algorithm consists of three steps, further detailed in this section:

1. extraction of a silhouette by a background subtraction technique at time t,
2. computation of a histogram at time t, which is used to update the gait signature, and
3. classification of a gait signature by a machine learning algorithm which outputs the identity of one of the persons known to the system.

4.3.1 Silhouette extraction[1]

The quality and the changing nature of the illumination conditions encountered when using real surveillance cameras led us to adopt an advanced background subtraction technique which can deal with changing illumination, noisy sensors and cast shadows. This background technique was

[1]The work described in this Chapter does not make use of the silhouette extraction algorithm introduced in Chapter 2. As explained in Chapter 1, this is due to the fact that the work described in this Chapter has been achieved before the work described in Chapter 2. However, our later work has shown that ViBe is the best choice for background subtraction.

Figure 4.6: Example of binary silhouette extracted with the algorithm of Zivkovic, as described in [136].

proposed by Zivkovic in [136]. It extends the widely used Mixture Of Gaussian algorithm ([110]) by selecting automatically and dynamically the optimal number of Gaussian distributions to use for each pixel. The result of this background extraction technique is illustrated in Figure 4.6. It can be seen that despite the use of an advanced background subtraction technique, the silhouette is not perfectly detected. Much of the gait recognition efficiency will therefore rely on the robustness of the gait signature.

4.3.2 Intra-frame silhouette description and gait signature by rectangle size distributions

In order to characterize a gait, we use one of the gait signatures introduced in Section 4.2.3. These are updated frame by frame, as soon as a silhouette histogram is computed at time t. Figure 4.7 displays a graphical representation of $\mathcal{G}^{W+H}(k,\, t)$ to show the quantity of information gathered in the signature. Since we do not perform any kind of tracking, we restrict ourselves to only one person being present at a time in the field of view of the camera. The choice of using $\text{hist}^{W+H}()$ or $\text{hist}^{W\times H}()$ depends on the amount of training data available as the dimensionality of $\text{hist}^{W\times H}()$ is usually larger than the one of $\text{hist}^{W+H}()$.

It is important to note that our method comprises no gait cycle detection or normalization algorithm, unlike many techniques described in the literature (see [11]); our tests have proven that these techniques can be unnecessary.

4.3.3 Gait classification

The gait signature obtained at time t is the feature set used for recognition. There is no special difficulty involved in mapping a gait signature to a class label, except that it must be fast, versatile, and accurate. Another criterion for the classifier is its ability to handle sets of features having high dimensionalities ($(M + N) \times L$ or even $M \times N \times L$ in our case). We chose a classifier, called extra-trees (for EXtremely RAndomized TREES) for its ability to handle features spaces of high dimensionality. Without going into detail, extra-trees is an ensemble method related to *bagging* [14] and *random forests* [15]. The goal of extra-trees is to reduce the variance by using

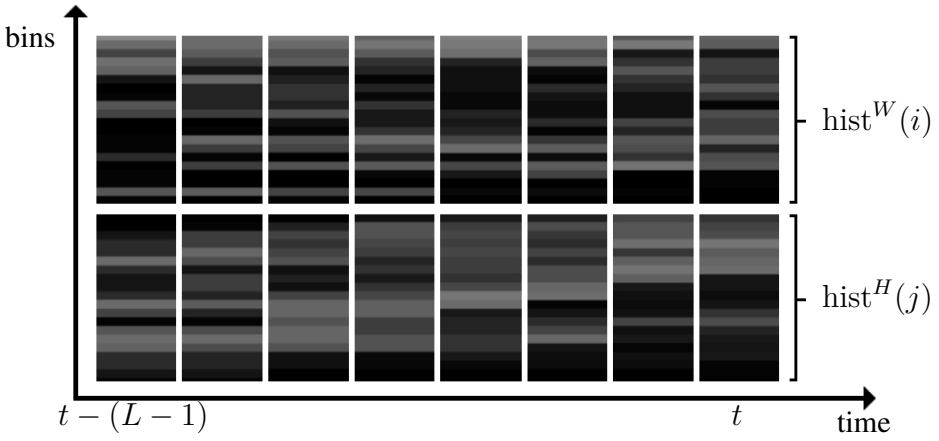


Figure 4.7: A graphical representation of $\mathcal{G}^{W+H}(k,\, t)$. All these displayed bin values are part of the feature set given to the gait classification algorithm.

a forest of independent trees instead of a single tree, and to reduce the bias by using a random selection of the thresholds at the splits of the trees (see [36] for a full description).

4.3.3.1 Majority vote policy on a sliding temporal window

Our gait recognition algorithm is synchronous: it provides the name for the person in the field of view whatever the time t might be. This is less restrictive than many techniques described in the literature which have to process the *complete* gait sequence before producing a single class label. On the other hand, this guarantees no temporal consistency, and a new, possibly different, class label might be computed by the system for each new frame, on the basis of the previous L frames. In order to smooth the result over time, we add a step that performs a majority vote on the previous V class labels produced by the classifier. Since the gait signatures already account for the information contained in the previous L frames, this brings a total delay of $L + V$ frames in achieving a reliable identification of a person once he has entered into the field of view of a camera.

4.4 Experimental results

In this section, we present results of multiple experiments that were run in real time on 640×480 pixels wide video sequences. Our algorithm can handle higher resolutions as well, but we haven't noticed any significant performance improvements when using higher resolutions.

Let us first determine appropriate values for all the parameters of the method. Afterward, we will present the precision of the classification on our database of 21 persons and then test our algorithm on a public database comprising videos of 25 persons.

We ran a first series of experiments on a dataset, hereafter called LAB5, which contains 4 sets of walking sequences for 5 persons. These sequences of the LAB5 data set were captured in our lab (see Figure 4.8) under strict and constant illumination. Videos were obtained from a consumer market webcam in order to get a realistic noise level and to ensure similar acquisition conditions to those of common situations. The goal of this set-up and this first series of videos was to determine appropriate values for the few parameters of our system.

The parameters to be refined were:

- which gait signature to use: either $\mathcal{G}^{W\times H}(i,\ j,\ t)$ or $\mathcal{G}^{W+H}(k,\, t)$,
- the numbers of bins M and N,
- the number of frames L aggregated in a single gait signature, and

Figure 4.8: Examples of frames of the LAB5 and LAB21 datasets captured in our lab.

- the length V of the sliding temporal window used for the majority vote policy.

The decision to use $\mathcal{G}^{W\times H}(i,\ j,\ t)$ or $\mathcal{G}^{W+H}(k,\ t)$ depends on the amount of training data and memory available to the classification process. If all other parameters are kept unchanged, the use of $\mathcal{G}^{W\times H}(i,\ j,\ t)$ generally leads to better results. However, the dimensionality of the corresponding feature space is $M \times N \times L$ instead of $(M + N) \times L$. As a result, a larger amount of data is necessary to train the system and the resulting extra-trees model that has to be loaded into memory at run-time is significantly larger.

In order to determine M and N, the numbers of bins, we tested values ranging from 2 to 40. It was observed that higher values of M or N (or both) generally leads to a better performance. However, the performance starts to be acceptable for 10 bins and then saturates with 20 bins and above. It is therefore recommended to use a value in the interval range [10, 20] for M and N. Depending on the size of the training dataset and the dimensions of its silhouettes, the statistical significance of all the bins of the histograms needs to be taken into account. Indeed, from small training sets of small silhouettes, it is impossible to populate a large histogram with enough statistical significance. Consequently a value closer to 10 needs to be chosen. By contrast, larger training sets of larger silhouettes would incline us to take values of closer to 20.

A similar reasoning applies to the number of silhouettes L aggregated in a gait signature: the higher, the better. Since the value of L impacts on the reactivity of the system and no significant gain in performance is observed for values of L larger than 20, taking $L = 20$ offers an appropriate compromise. Note that this parameter may be refined according the framerate of the cameras used. Typical cameras have a framerate of 25 images per second: $L = 20$ corresponds to a signature of about 1 second which roughly matches the length of a gait cycle. For slower framerates, L has to be adapted.

The discussion regarding the appropriate value for V, the length of the sliding temporal window used for the majority vote policy, is again similar to the one regarding L. With V at a high level, the results are better but the drawback is that this increases the number of frames needed to identify a person. From a practical point of view, a majority vote regarding 10 consecutive frames is sufficient; it improves the performance of the system to a satisfactory level. If $L = 20$ and $V = 10$, the algorithm delays its answer for 30 frames, *that is* 1 second for commonly-used cameras.

4.4.1 Tests on a database of 21 persons

In order to estimate performance of our system, we used a second dataset, called LAB21, which was composed of 4 sets of laboratory sequences of 21 different subjects. All the classification tests were conducted by training the algorithm using 3 of the 4 sequences available for each subject

and testing it on the left out one. We used the ratio of correctly classified gait signatures as a performance criterion. This ratio was computed for different numbers of frame per gait signature and for different histogram resolutions. For the sake of simplicity, we restricted ourselves to the case where $M = N$, and disabled the majority vote on the previous V frames (or to equivalently set V to 1) in the first instance. This allowed us to assess the raw classification precision of the system, regardless of whether the majority vote improved the performance, as shown further on.

The results of the first series of tests are shown in Figure 4.9. The ratio of correctly classified gait signatures reached 74% for $\text{hist}^{W\times H}()$ and 72% for $\text{hist}^{W+H}()$. Both $\text{hist}^{W+H}()$ and $\text{hist}^{W\times H}()$ obtained the best results for a number of bins of 10 and a number of frames per gait signature (L) of 20. We also noticed that the performance of $\text{hist}^{W\times H}()$ was generally better than that of $\text{hist}^{W+H}()$, especially for small values of the parameters M, N, and L.

One could be misled by the relatively average examples of performance given by figures around the 75% mark. Remember that the examples of performance reflect all the synchronous decisions individually. Should a single class label be assigned to a test sequence as the majority decision among the complete set of individual decisions, the performance ratio would overstep 95% of correctly classified gait sequences!

The second series of tests was limited to $\text{hist}^{W\times H}()$ in order to focus on the performance improvement brought about by the majority vote on the previous V frames. The curves displayed in Figure 4.10 show that the use of the majority vote improves the performance of the system. For high values of V, the ratio of correct classifications peaks at 97%. In the same way as in the discussion on parameter L, we observe that an increase in the length of the majority vote time window improves precision. Interestingly, we also noticed that the choice of $M = N = 15$ outperformed the results of the choice of $M = N = 20$. This presumably originates from the small size of some silhouettes, which only contained a few wedged rectangles α. If α is too small, which typically occurs when a person stands too far from the camera, it is impossible to estimate a histogram split into 20×20 bins with a good statistical significance; this poor estimation negatively impacts on performance.

4.4.2 Tests on frames acquired with surveillance cameras

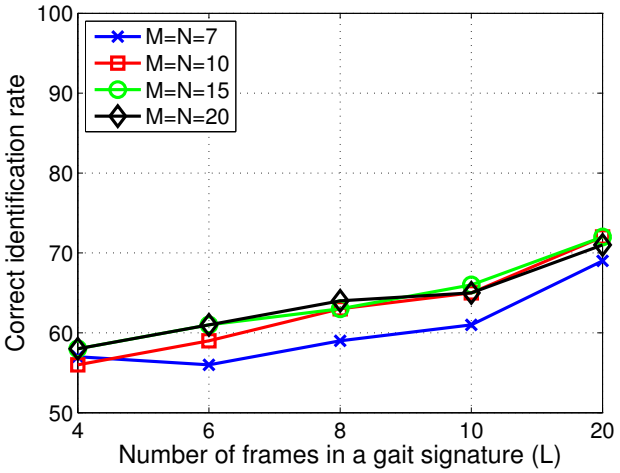
The third data set used was named HW5. This consisted of frames captured with surveillance cameras located in hallways for five different persons and involving 3 sequences per person. In contrast with the previous sequences, the environment was totally unconstrained and some frames had a poor signal to noise ratio.

The results of this last series of experiments are shown in Figure 4.11. As expected, the precision of the classification suffered from the poor quality of the extracted silhouettes (remember the example of Figure 4.6). Nevertheless, thanks to the robustness of the proposed gait signature, the system still managed to identify correctly the persons in up to 81% of cases (one should compare this with the previous 97%). The 81% of correct classifications were obtained for a majority vote window of 55 frames, which corresponded to an identification delay of 2 seconds (or $L + V = 65$ frames).

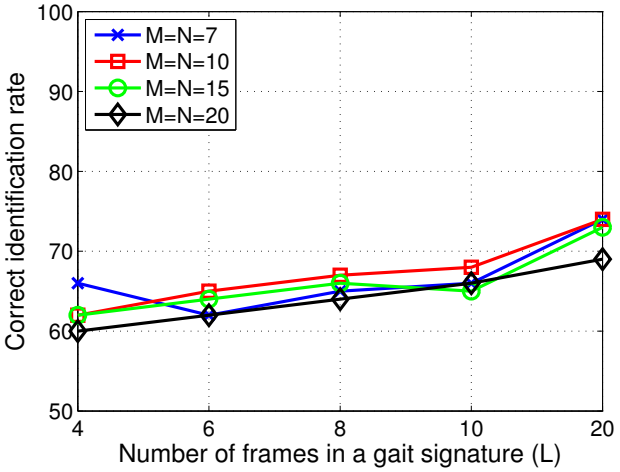
4.4.3 Tests on the CMU MoBo database

To further evaluate the performance, our algorithm was tested on the publicly available MoBo database [38]. The MoBo database consists in video sequences of 25 subjects walking on a treadmill. Six calibrated and synchronized cameras were used to capture the subjects from six different viewpoints performing four different walking activities: slow walk, fast walk, incline walk, and walk with a ball. The database also comprises binary segmentation maps for each sequence. By using these segmentation maps, we are able to assess the performances of the features extraction and classification process exclusively (without any interference from the background subtraction algorithm).

To achieve a fair comparison with other techniques evaluated on the MoBo database, we used exactly the same experimental set-up. For example, each complete walking sequence is given a



(a) $\text{hist}^{W+H}(k)$



(b) $\text{hist}^{W\times H}(i,\, j)$

Figure 4.9: Performance of $\mathcal{G}^{W\times H}(i,\, j,\, t)$ on the LAB21 dataset with no majority vote policy (more precisely $V = 1$) using (a) $\text{hist}^{W+H}(k)$ and (b) $\text{hist}^{W\times H}(i,\, j)$.

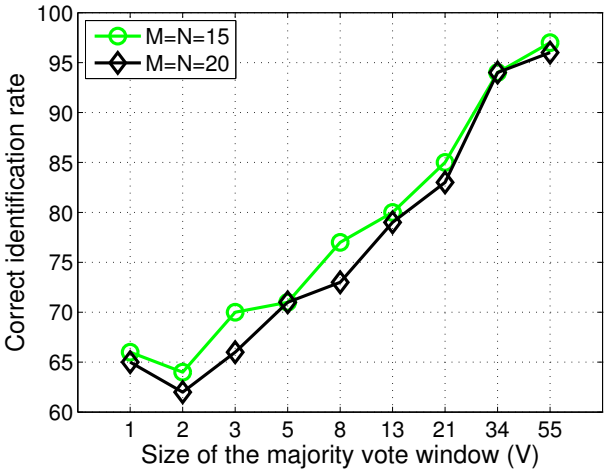


Figure 4.10: Performance of $\mathcal{G}^{W\times H}(i,\, j,\, t)$ on the LAB21 database using $\text{hist}^{W\times H}(i,j)$ for different lengths V of the majority vote window (L is set to 10).

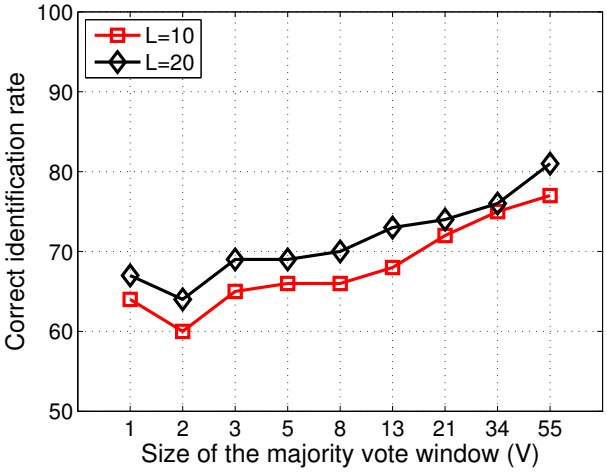


Figure 4.11: Performance on the HW5 dataset, which contained frames acquired with cameras located in hallways (M and N are set to 20).

Our algorithm	**Slow**	**Fast**
$\mathcal{G}^{W\times H}(i,\, j,\, t)$ with $M = N = 10,\, L = 10$	100%	100%
$\mathcal{G}^{W\times H}(i,\, j,\, t)$ with $M = N = 10,\, L = 20$	100%	100%
$\mathcal{G}^{W\times H}(i,\, j,\, t)$ with $M = N = 20,\, L = 10$	100%	100%
$\mathcal{G}^{W\times H}(i,\, j,\, t)$ with $M = N = 20,\, L = 20$	100%	100%

Table 4.1: Results obtained on non-overlapping parts of sequences from the same category of activity (training and testing sequences are both taken in the "slow walk" or "fast walk" subparts of the MoBo database).

Comparison of two methods	**Slow/Fast**	**Fast/Slow**
Our algorithm:		
- $\mathcal{G}^{W\times H}(i,\, j,\, t)$ with $M = N = 10,\, L = 10$	96%	96%
- $\mathcal{G}^{W\times H}(i,\, j,\, t)$ with $M = N = 10,\, L = 20$	96%	96%
- $\mathcal{G}^{W\times H}(i,\, j,\, t)$ with $M = N = 20,\, L = 10$	96%	96%
- $\mathcal{G}^{W\times H}(i,\, j,\, t)$ with $M = N = 20,\, L = 20$	96%	96%
Algorithm proposed in [46]:		
- frontal view	$\emptyset$	88%
- 6 views	$\emptyset$	92%
- frontal and lateral views	$\emptyset$	96%

Table 4.2: Results when training on one category of activity and testing on the other. Slow/Fast means that slow walking sequences were used for training while the tests were performed on fast walking sequences, and vice versa.

unique class label; this is equivalent to setting V to the total number of frames contained in the corresponding video sequence. Additionally, each sequence is divided in two non-overlapping parts of equal size. One part serves to train the algorithm, the other is used to evaluate it. We tested the method against the "slow walk" and the "fast walk" sequences separately. The results given in Table 4.1 show that the algorithm is able to successfully recognize every single person present in the database across the whole advised ranges of values of its parameters. For the sake of completeness, we also tested the method (with no adaptations) on the *lateral* sequences contained in the MoBo database using the same procedure. Interestingly, we observed identical scores (100% in all the cases). Future work will investigate the performance of our algorithm on lateral-view sequences.

Finally, we checked if the method was able to deal with greater discrepancies between training and test sequences on frontal views. Therefore our algorithm was trained on all the "slow walk" sequences and evaluated against all the "fast walk" sequences, and vice versa. From the results provided in Table 4.2, we see that the algorithm is able to successfully recognize persons even if the walking speed changes between the training and the testing steps. We also notice that our method outperforms that of [46] when using a single frontal camera; the best classification score presented in [46] was obtained by combining the frontal and the lateral views. In our case, sequences acquired with a single frontal camera suffice to produce the best recognition scores.

4.5 Application to an intelligent access control system

In this Section, we describe an original application of our inter-frame gait signature in the context of the design of a system for the intelligent control of access to a secure area. Commonly, access to restricted areas is monitored by a door with an electrical lock or a revolving door activated by the swipe of an access control card. In this context, we aim for a system able to automatically inform the security when the usual scenario of a single person crossing the revolving door is not confirmed. This occurs for example when an unauthorized (or even an authorized) person enters

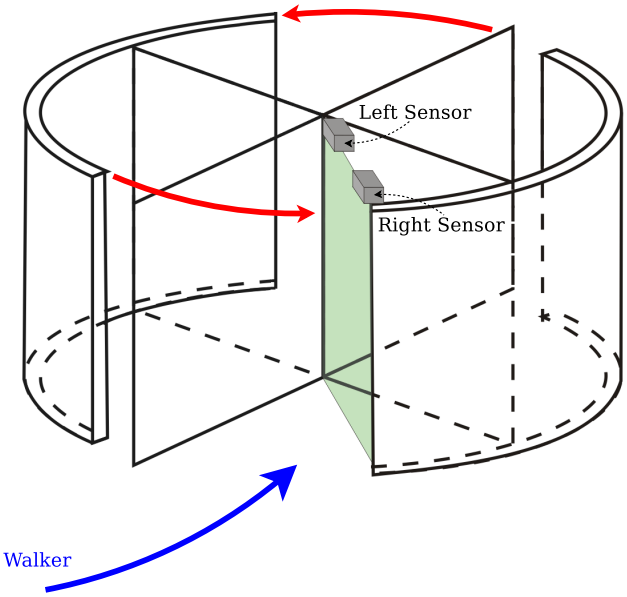


Figure 4.12: Embedding of the sensors in the frame of a revolving door (this figure is a modified version of an illustration provided by BEA).

a restricted area by passing through the door at the same time as another person (this is often referred to as *"piggybacking"*).

Hereafter, we will describe how we use our gait classification technique for the design of a system able to automatically raise an alert when more than one person are trying to pass through a door simultaneously. To get an absolute robustness to the lighting conditions, we do not use cameras. Instead, we use distance measures coming from radial laser sensors (or *scanners*) that can be directly embedded in a door frame (see Figure 4.12). We use the distance measures that these sensors provide to reconstruct the time series of the binary silhouettes of the person(s) walking through the door. We then classify these series using our gait recognition algorithm, which was trained to determine which series correspond to a *single* person crossing the door.

We describe below the sensors we use and their physical arrangement. We then explain how to recover binary silhouettes from the distance signals provided by these sensors. Finally, we give our experimental results, which were obtained using a database of more than 800 sequences.

4.5.1 Experimental set-up

Hereafter, we describe which specific kind of laser range sensors we use and their physical arrangement.

4.5.1.1 Sensors

To be usable in a real security framework, our system has to be completely independent of the lighting conditions. Consequently, we cannot afford using a background subtracted video stream to recover the binary silhouettes of the walkers. Instead, we use sensors that rely exclusively on their own light source, such as the rotating laser sensors manufactured by BEA (*BEA LZR P-200*, see Figure 4.13).

These laser range sensors are completely independent of the lighting conditions. They are able to measure the distance between the scanner and the surrounding objects by sending and

Figure 4.13: BEA LZR P-200 rotating laser range sensor.

receiving laser pulses in a plane. The measurement process is discrete. It samples the angles with an angular precision of 0.35° and covers an angular aperture of 96°. These sensors deliver a signal $d_t(\theta)$ where d is the distance between the sensor and the object hit by the laser, θ denotes the angle in the scanning plane ($0 \leq \theta \leq 96°$) and t is the time. This signal is determined alternatively in 4 planes shifted by an angle of $\phi = 1°$. Moreover, each one of the 4 planes is scanned 15 times per second. In practical terms, these sensors deliver a signal $d_{\phi,t}(\theta)$ where $t = \frac{k}{15}s$ ($k = 0, 1, 2, \ldots$) and $\phi \in \{0, 1, 2, 3\}°$. In Section 4.5.2, we will show how to use these signals to reconstruct the time series of the binary silhouettes of the person(s) crossing the door.

4.5.1.2 Physical arrangement of the sensors

Theoretically, a single scanner is sufficient to build a 2D shape. However, we decided to use two scanners to reduce the shadowing effects resulting from a single scanner. The sensing system is made of two laser scanners located on the two upper corners of the frame of a door (see Figure 4.12 and Figure 4.14). Consequently, the distance measures are performed in planes that comprise the vertical of the gravity (possibly shifted by 1, 2, or 3°) and the straight line joining the two sensors.

It must be noted that the collected signals from the two sensors are not synchronized. This impacts on the system. Assume for example that the laser scanners acquire the border of an object in two different planes and that the planes are shifted by 3 degrees (this is the worst case). If the scanners are located at a height of 2.5 m, then the top of the head (say at a height of 1.8 m) is seen with a horizontal shift of 4 cm (= 0.7 sin(3)). At the height of the knee (say 0.6 m), it accounts for a shift of 10 cm, which is about the width of the knee. Therefore the physical significance of a point increases with the height in the reconstructed plane. In other words, the horizontal imprecision, due to the de-synchronization of the scanners, decreases with the height of a point.

Furthermore, as measures correspond to the distance between the sensor and the first point hit by the laser along its course, they account for a linear information related to the central projection of the silhouette of the moving objects passing through the door. It implies that points located beyond the first point are invisible and that widths are impossible to measure with a single scanner. With two scanners, there are less ambiguities but some of them remain, for example in the bottom of the silhouette. Furthermore, a hole in the silhouette cannot be detected. In practice, the subsequent classification algorithm has to be robust enough to be able to deal with these ambiguities.

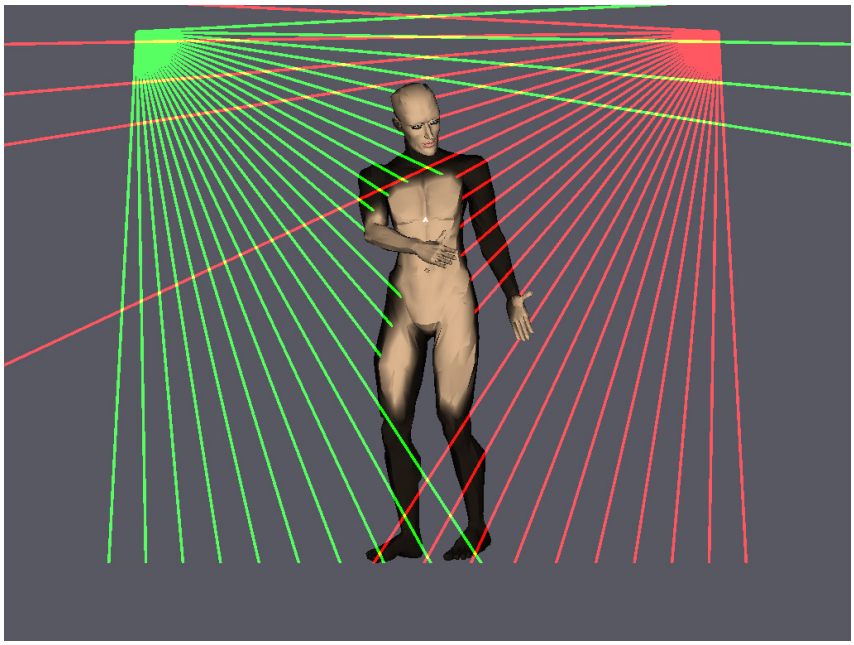


Figure 4.14: Arrangement of the sensors. The two rotating laser range sensors are located on the two upper corners of the frame of a door.

4.5.2 Silhouette reconstruction

We now describe how to recover the silhouettes of the walking human passing through the door from the polar information coming from the sensors.

4.5.2.1 Polar transformation and registration of the two signals

Since the information given by the laser scanners is polar, the first step towards the reconstruction of an image related to the shape of the scanned objects is a polar transformation of the raw signals:

$$x_{\phi,\,t}(\theta) = d_{\phi,\,t}(\theta)\,cos(\theta) \tag{4.8}$$

$$y_{\phi,\,t}(\theta) = d_{\phi\,,t}(\theta)\,sin(\theta) \tag{4.9}$$

Every $\frac{1}{15}s$, we get two new captured signals $d_{\phi,\,t}(\theta)$, one for each sensor, and apply a polar transformation to them. We then combine pairs of signals *sharing identical* ϕ, connect their points, and register them according to the width of the frame of the door. Thanks to the calibration of the sensors and the real physical distances they deliver, the registration process is simple as it relies exclusively on the physical dimensions of the frame of the door. The resulting signal after polar transformation and registration of the two sensors is shown on Figure 4.15a.

It should be noted that since the signals provided by the two sensors are not synchronized, the registration of the sensor signals will be affected by a time jitter that impacts on the overall signal to noise ratio.

4.5.2.2 Flood fill and intersection

For each laser scanner, we now have a continuous line that outlines one side of the silhouette of the object currently passing through the door. We need to recover one complete silhouette (see Figure 4.15b) from the two contours displayed on Figure 4.15a. We obtain the complete silhouette by reconstruction and intersection of two half-silhouettes. The reconstruction of a half-silhouette is achieved by closing the contour and applying a flood fill algorithm to it.

(a) (b)

Figure 4.15: (a) Resulting signal after polar transformation and registration of the two sensors. The signal captured by the left (right) sensor is displayed in green (red). (b) Resulting silhouette obtained after flood fill and intersection of the two contours.

The whole reconstruction process is illustrated on Figure 4.16. We see that the upper part of the silhouette (in principle, the shoulders and the head) are better represented than the lower parts of the silhouette because the lasers are closer to the upper part and thus do sample this part of the shape with a superior precision. As a matter of fact, the legs are almost absent from the reconstructed silhouettes. Furthermore, we showed in Section 4.5.1.2 that the lack of synchronization of the sensors causes a horizontal imprecision that decreases with the height of a point.

4.5.3 Crossings detection and classification

When a person enters the frame of a door, we reconstruct the time series of its binary silhouettes and assign a class to it with our gait classification algorithm. In this particular application, only two classes can be assigned to a series of silhouettes:

- "0" (that is "false") which denotes the fact that a sole person has passed through the door,
- and "1" (that is "true") which denotes that more than one person has passed through the door.

We tested both $\mathcal{G}^{W+H}(i,\, j,\, t)$ and $\mathcal{G}^{W\times H}(i,\, j,\, t)$ and tried different reading order of the scanning planes. The order in which the scanning planes are processed is a delicate question since the sensors deliver them in a deterministic but non-increasing order. Due to a balance constraint on the rotating device, signals are given in the order of $\phi = 0,\, 2,\, 1,\, 3°$. Three processing orders were considered:

- "chronological" which considers the scanning planes in the order delivered by the sensors,
- "angular" which considers them in order of increasing ϕ,
- and "anti-angular" which considers them in order of decreasing ϕ.

4.5.4 Results

To be able to evaluate the performance of the application of our algorithm to this problem, we collected and labeled a database that contains 349 walking sequences of a single person (class "0") and 517 sequences that contain two walkers (class "1"). We use these sequences to build databases

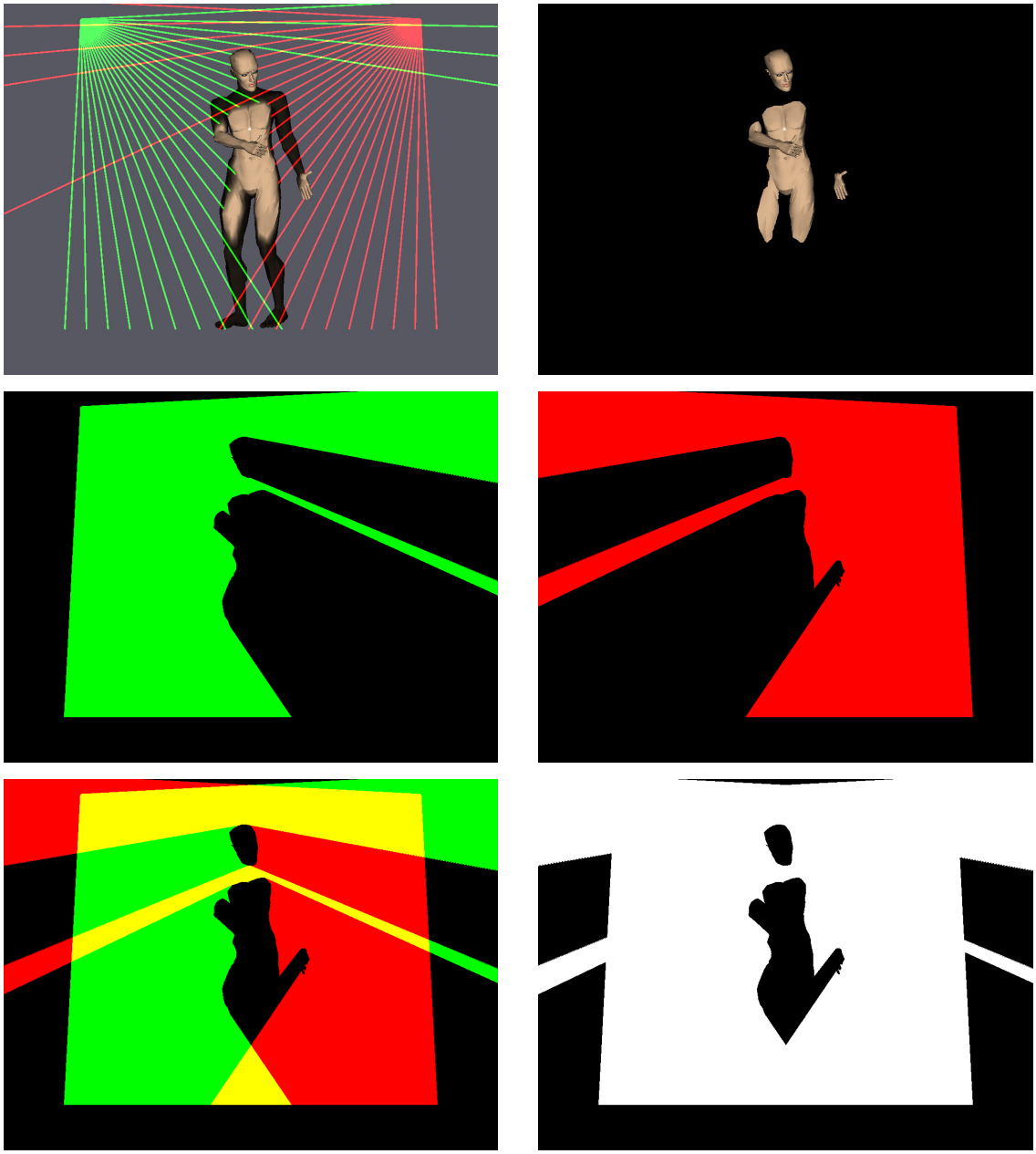


Figure 4.16: Illustration of the silhouette reconstruction process.

E [%]	$M = N = 2$	$M = N = 4$	$M = N = 6$	$M = N = 8$	$M = N = 10$
$L = 40$	15.99	14.43	14.37	14.68	14.76
$L = 60$	9.77	8.34	8.23	8.86	8.55
$L = 70$	7.89	7.04	6.70	6.86	7.17
$L = 80$	6.98	**6.05**	**5.68**	6.19	**5.91**
$L = 90$	7.51	7.65	7.44	7.09	7.51
$L = 100$	9.84	11.58	10.04	10.86	10.97
$L = 120$	18.15	18.15	18.16	16.34	17.50

Table 4.3: Error rates obtained using $\mathcal{G}^{W+H}(i,\, j,\, t)$ and a chronological processing of the scanning planes.

E [%]	$M = N = 2$	$M = N = 4$	$M = N = 6$	$M = N = 8$	$M = N = 10$
$L = 40$	16.06	14.05	14.07	14.04	14.76
$L = 60$	9.83	8.96	9.79	10.25	10.79
$L = 70$	8.01	7.82	8.35	8.35	8.57
$L = 80$	**7.12**	**7.45**	7.5	**7.26**	7.64
$L = 90$	8.14	8.62	8.62	8.76	8.62
$L = 100$	10.45	9.43	9.84	9.73	11.79
$L = 120$	19.81	17.33	16.01	16.01	18.32

Table 4.4: Error rates obtained using $\mathcal{G}^{W\times H}(i,\, j,\, t)$ and a chronological processing of the scanning planes.

of labeled gait signatures for different sets of parameters (signature type, processing order, M, N, and L). For each set of parameters, we employ 5-fold cross-validation on the corresponding database to assess the precision of the classification according to the error rate (E) defined as

$$E = \frac{FP + FN}{TP + TN + FP + FN}$$

where TP is the number of true positives, TN the number of true negatives, FP the number of false positives, and FN the number of false negatives. Note that due to the short crossing times observed in this application, we do not employ the majority vote policy on a sliding temporal window of Section 4.3.3. This is equivalent to setting the parameter V of Section 4.3.3 to 1, that is $V = 1$.

Two series of results are shown in Table 4.3 and in Table 4.4. They show that we manage to obtain an error rate as low as 5.68% using the $\mathcal{G}^{W+H}(i,\, j,\, t)$ signature with $M = N = 6$, $L = 80$, and a chronological processing order of the scanning planes. We also observe that for this particular problem, the most influential parameter is L. In our tests, the best results are obtained for a signature length L of 80 frames, a number that roughly matches the average crossing time of the walker(s). For M and N, the choice of a value is less critical but in our tests, it appears that $M = N = 4$ is an educated choice. We also notice that for this particular problem, $\mathcal{G}^{W+H}(i,\, j,\, t)$ has slightly better results than $\mathcal{G}^{W\times H}(i,\, j,\, t)$, while having a reduced computational cost. One explanation to this is that the shadowing effect in the lower part of the silhouette adds more noise on $\mathcal{G}^{W\times H}(i,\, j,\, t)$ than on $\mathcal{G}^{W+H}(i,\, j,\, t)$.

In a second series of cross-validation tests, we investigated the influence of the processing order of the scanning planes. From the results shown in Table 4.5, it comes out that the processing order does not have a significant influence on the global error rate, provided that appropriate values are assigned to M, N, and L.

Finally, it must be noted that during our tests, we observed that for low global error rates, the number of FN is vastly lower than the number of FP. In other words, the system is naturally more inclined to reject a single person than to allow to a group of two persons to pass the door.

PCC [%]	$M = N = 2$	$M = N = 4$	$M = N = 6$	$M = N = 8$	$M = N = 10$
chronological	6.98	5.96	**5.73**	**6.38**	6.47
angular	6.94	**5.75**	6.37	6.63	6.43
anti-angular	**6.72**	6.12	6.2	6.37	**6.12**

(a) $\mathcal{G}^{W+H}(i,\, j,\, t)$

PCC [%]	$M = N = 2$	$M = N = 4$	$M = N = 6$	$M = N = 8$	$M = N = 10$
chronological	**6.94**	**7.08**	**7.22**	7.59	8.19
angular	6.99	7.15	7.36	**7.10**	**7.62**
anti-angular	7.54	7.54	7.50	7.45	8.14

(b) $\mathcal{G}^{W\times H}(i,\, j,\, t)$

Table 4.5: Error rates obtained using different processing order of the scanning planes.

For an access control system, it is a welcomed property since the final decision of granting access to the secure area can be taken by a security employee when an unauthorized passing is detected.

4.6 Conclusions

Gait identification is currently an intensive topic for research. Most techniques described in the literature are based on lateral views of walking persons. It is known that lateral views contain appropriate information regarding the gait. However, using lateral views in indoor environments might be unfeasible, especially in hallways where a frontal view is almost inevitable.

This chapter proposes a real-time *frontal-view* gait recognition system. A major contribution is introduced by defining a gait signature of a walking person. Successive binary silhouettes are extracted with a background subtraction algorithm. Each silhouette is then converted to an intra-frame histogram which compacts the width and height distributions of the set of all the rectangles that can be wedged inside the silhouette. Afterward, a given number L of successive histograms is combined into a single spatio-temporal (inter-frame) gait signature. The identification of the persons is then computed by a classification of this signature by a machine learning algorithm called extra-trees. Finally, successive decisions are combined along several frames using a majority vote policy to determine the identity of the person currently present in the field of view of the camera.

Four series of experiments were conducted on different databases. The first series helped to determine the parameter values needed to optimize the performance of the overall system. The second series was intended to evaluate the precision of the classification for different ranges of values of the parameters. It was shown that the ratio of correct classifications could reach 97% for a database of 21 persons. The third series of experiments served as a showcase for a practical scenario. Frames were captured with hallway surveillance cameras at our institute. Despite the noise and the unavoidable phenomena in such an unconstrained environment, the system was still able to identify the persons correctly in up to 81% of cases. We also tested our algorithm on the publicly available MoBo database. Our method was able to successfully recognize the persons from video sequences taken in the MoBo database, reaching a score as high as 96% to 100%, depending on the training and testing conditions.

Finally, we described a complete platform (hardware and software) for the intelligent control of access to a secure area, which make use of our inter-frame gait signature. Its purpose is to raise an alarm when unauthorized scenarios are detected. Despite the shortcomings (such as occlusion or shadowing) resulting from the use of rotating laser scanners to recover the silhouettes of the walkers, we managed to reach an error rate as low as 5.68% for the detection of groups of persons trying to cross the door simultaneously.

Chapter 5

Conclusions

In this manuscript, we introduced three algorithms. Together, they constitute a complete framework for the detection and recognition of human in video sequences acquired with a static camera. These three algorithms are: (1) a background subtraction algorithm called "ViBe", (2) a human silhouette detection technique, and (3) a silhouette-based gait recognition algorithm.

ViBe is a pixel-based background subtraction algorithm that uses a classification model based on a small number of correspondences between a candidate value and a collection of samples which constitutes the background pixel model. ViBe incorporates an innovative model update mechanism that ensures both a smooth decaying lifespan for the samples stored in the pixel models and a spatial consistency of the background model. This update mechanism brings a reduced memory usage and relieves from the need to post-process the segmentation maps produced by ViBe. The update mechanism of ViBe also enables the use of a conservative update policy: no foreground value is never included in any background pixel model. Furthermore, ViBe can be instantaneously initialized using a single frame. In our experiments, we showed that ViBe is fast enough to be directly embedded in a portable device. Comparisons with six other state-of-the-art algorithms established that ViBe produces the most precise results while being faster that five out of the six methods that were used for comparison. We finally proposed a scaled-down version of ViBe which only requires one byte of memory and one comparison per pixel. This scaled-down version still produces better results than several other state-of-the-art techniques.

To detect the presence and the location of persons, we proposed a silhouette classification algorithm that can be applied directly to the binary motion detection maps produced by ViBe. This algorithm is based on the classification of features extracted from the silhouettes. These features are based on a new type of morphological operator which can be seen as an extension of the granulometric filters: the cover by rectangles. Intuitively, the cover by rectangles of a silhouette is the set of all the maximal rectangles than can be wedged inside a silhouette. Features are extracted from these sets of rectangles and classified with a machine learning algorithm. A real-time implementation of the method showed that the method is both stable and computationally effective.

To be able to recover the identity of the detected persons, we introduced a frontal silhouette-based gait recognition algorithm also based on the cover by rectangles. From the time series of the binary silhouettes of a walking person, we infer the identity of the person by extracting and classifying a gait signature. For each silhouette of the series, we compact the width and height distributions of all the rectangles contained in its cover by rectangles using a histogram. Histograms of a given number of consecutive silhouettes are concatenated to form the gait signature of the walker. The gait signature is eventually classified by a machine learning algorithm. We tested the method against a database of 21 persons and obtained a ratio of correct classifications of 97%. When tested on the publicly available MoBo database, our algorithm managed to recognize persons in up to 100% of the cases.

Finally, in collaboration with BEA, we used our algorithm in the context of the design of an intelligent access control system. The purpose of the system is to raise an alarm when groups of

persons are trying to pass through a door simultaneously. In this context, we used range measures from rotating laser scanners to reconstruct the silhouettes of the walkers. Despite the unavoidable occlusion and shadowing artifacts caused by the use of laser scanners, our gait classification algorithm was robust enough to reach an error rate as low as 5.68%.

At several crucial points of the algorithms we have described in this manuscript, we resorted to random decisions. One conclusion of this thesis is that in the absence of a meaningful deterministic heuristic, the best policy could be to roll the dice.

Appendix A

Patent application for ViBe: "Visual Background Extractor"

This appendix provides the final text of the patent that covers the background subtraction algorithm introduced in chapter 2. The patent proposes several innovations that are not described in chapter 2. These innovations are:

- the applicability of the algorithm to any kind of multi-dimensional data;
- a multi-layer model that allow to sort the data according to their evolution rate;
- the use of an additional background layer to speed-up the absorption of the ghosts into the background model;
- a method to generate a synthetic image that depicts the background;
- a random spatial subsampling technique that can be used to further reduce the computational cost of the algorithm;
- and a technique that reallocates the pixel models in order to account for either a panning, or a zooming in, or a zooming out of the camera.

A.1 Abstract

The present invention relates to a Visual Background Extractor (ViBe) consisting in a method for detecting a background in an image selected from a plurality of related images. Each one of said set of images is formed by a set of pixels, and captured by an imaging device. This background detection method comprising the steps of:

- establishing, for a determined pixel position in said plurality of images, a background history comprising a plurality of addresses, in such a manner as to have a sample pixel value stored in each address;
- comparing the pixel value corresponding to said determined pixel position in the selected image with said background history, and, if said pixel value from the selected image substantially matches at least a predetermined number of said sample pixel values:
  - classifying said determined pixel position as belonging to the image background; and
  - updating said background history by replacing the sample pixel values in one randomly chosen address of said background history with said pixel value from the selected image. The method of the invention is applicable a.o. for video surveillance purposes, videogame interaction and imaging devices with embedded data processors.

A.2 Description

The present invention relates to a method for detecting a background in an image selected from a plurality of related images, each formed by a set of pixels, and captured by an imaging device.

One of the major research areas in computer vision is the field of motion tracking in video sequences. The aim of motion tracking is to extract the movement of an object in a scene and sometimes the movement of the imaging device generating images of said scene. The human eye and brain carry out this kind of operation easily, but in computer vision it is a difficult problem to solve, and it involves several tasks, such as:

- separating the moving objects from the static or pseudo-static background, wherein a pseudo-static background is defined as a background with a slight apparent motion due to a motion of the imaging device;
- solving the occlusion problem, which involves re-identifying an object that is momentarily at least partially lost to view;
- eliminating so-called ghost objects, resulting from falsely considering as part of the background real objects present in the first image frame of a sequence of images; and
- identifying and removing shadow effects.

A video sequence is formed by a sequence of images I, captured successively in time, wherein each image I comprises a plurality of picture elements, or pixels, each single pixel in an image having a position x in the image and a pixel value $I(x)$. The position x may have any number of dimensions. Although pixels with three-dimensional positions are also known as "voxels" (for "volume elements") in the domain of 3D imaging, the expression pixel should also be understood throughout the present text as also including such "voxels". The image can thus be a conventional two-dimensional image, but also a 3D image and/or a multispectral image. This position x may, or may not be limited to a finite domain. In the case of images captured by a moving imaging device, such as, for example, a satellite on-board camera, the position x will not be limited to a finite domain. Depending on the type of image, the pixel value $I(x)$ may be scalar, as in monochromatic images, or multidimensional, as in polychromatic images, such as red-green-blue (RGB) component video images or hue saturation value (HSV) images.

The step of separating the moving objects from the background is generally called background subtraction. Background subtraction involves the recognition of which pixels in each image belong to the background and removing them from the image.

Background subtraction is also used in other applications, such as medical imaging systems. A medical image may be compared with a background generated using images of the same region in other patients to show outliers possibly indicating a pathological condition.

To be efficient, a background subtraction technique should:

- be universal, that is, it should be capable to handle any type of video sequences;
- be robust with respect to noise;
- be simple, since simple techniques are easier to implement and need fewer parameter adjustments depending on the scene content;
- be fast, involving as few computations as possible;
- be accurate in shape detection;
- have a small footprint, although with the decreasing cost of memory, it is not crucial to minimize the memory space needed;
- be directly usable, requiring only a fast initialization, if any;

- be adaptable to gradual or fast illumination changes, due, for instance to the changing time of day or to clouds, motion of the imaging device, high frequency motion of background objects, such as tree leaves or branches, and changes in the background geometry, due, for instance, to parked cars.

There are numerous existing background subtraction methods. Surveys of such methods have been disclosed by M. Piccardi in "Background subtraction techniques: a review", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 4, 2004, pages 3099-3104, and by R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam in "Image change detection algorithms: A systematic survey", IEEE Transactions on Image Processing, 14(3): 294-307, 3 2005.

A first approach to background subtraction is the so-called na[iota]ve approach. Following this na[iota]ve approach, a simple background model assumes that background pixel values are constant over time. The background and foreground are then recognized by comparing the values of the pixels in the image with those of an earlier image of the same sequence, as disclosed, for instance, in US Patent 6,061,476. Differences can be caused by noise in the image data, illumination changes in the scene, or motion. Noise and global illuminations are dealt with by applying a filter or by using a detector targeting illumination changes. However, discarding the motion of background objects is more difficult, especially if one keeps in mind the requirements given above.

In practice, the difference between two images is thus inefficient and background recognition methods have to allow for a distribution of background pixel values to cope with noise or illumination changes. Therefore, extended techniques have been proposed. These methods, hereafter called model-based approaches, build a model $m(x)$ for the value $I(x)$ of each pixel. The model is used to compare a pixel value with previous pixel values for the same pixel position. For example, the W^4 algorithm disclosed by I. Haritaoglu, D. Harwood, and L. Davis in "W^4: Real-time surveillance of people and their activities", IEEE Transactions on Pattern Analysis and Machine Intelligence", 22(8): 809- 830, 2000, models the background by the minimum and maximum pixel values, and the maximum difference between two consecutive images during a training stage. More advanced model-based approaches comprise the following steps:

- Initialization: The purpose of initialization is to build a valid estimate of the model parameters from the first images of a sequence.
- Comparison: The current pixel value is checked against the model.
- Model update: If the pixel value fits reasonably well with the model, then the parameters and thus the model are updated. If the pixel value does not match the model, the new value is not used to update the model, as otherwise the model would diverge.

Depending on the type of model, comparison strategy or updating process, several categories of model-based background subtraction methods have been proposed.

In a first category of model-based background subtraction method, the background pixel model is a predicted background pixel value, based on a statistical measure, such as, for instance a weighted average of pixel values. Background subtraction methods of this category have been disclosed, for instance, in US Patent 7,136,525, US Patent 4,350,998, International Patent Application WO 01/16885, US Patent Application US 2006/0120619 or US Patent Application 2002/0064382. In one of its simplest forms, the background pixel model is based on a weighted average of previous pixel values in the same pixel position. If $m_t(x)$ denotes the pixel value provided by the model for the pixel position x at time t, and $m_{t+1}(x)$ the pixel value provided by the model for the same pixel position x at the next time $t+1$, the update formula will then be:

$$m_{t+1}(x) = \alpha l_t(x) + (1-\alpha) m_t(x) \tag{A.1}$$

where α is a weighting factor between 0 and 1. The image formed by the model pixel values $m_t(x)$ constitutes an estimate of the background.

While the calculation of a weighted-average background pixel model is comparatively simple, in some situations a single predicted pixel value, such as a simple weighted average, does not provide

a comprehensive model of the background. For this reason, in a second category of model-based background subtraction method, so-called parametric models are used which provide a probability density function (pdf) of background pixel values at a certain pixel position, wherein the pdf is characterized by a set of parameters. For example, in C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland disclosed in "Pfinder: Real-time tracking of the human body", IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):780-785,1997, a background model which assumes that the pixel values over a time window at a given position are Gaussian distributed. A Gaussian background model is also proposed in US Patent Application US 2006/0222205. Complex methods, such as the background subtraction method disclosed in International Patent Application WO 03/036557, estimate a mean vector and covahance matrix for each pixel position. Sometimes this is combined with the use of Kalman filtering to adapt their values, as disclosed, for example, by D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell in "Towards robust automatic traffic scene analysis in real time", Proceedings of the International Conference on Pattern Recognition, Israel, 1994. However, the main drawback of the method remains: the uni-modal nature of the model prevents it to deal correctly with the multi- modal appearance caused by the motion present in the background of a dynamic environment, such as in tree leaves, flickering monitors, etc.

To solve this problem, and still within the category of parametric models, the use of a weighed mixture of Gaussian distributions to model the background was first proposed by C. Stauffer and E. Grimson in "Adaptive background mixture models for real-time tracking", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 246-252, 6 1999, and in "Learning patterns of activity using real-time tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):747-757, 2000. Since its introduction, this Gaussian Mixture Model (GMM) has gained in popularity among the computer vision community, as can be seen in the abovementioned surveys, as well as in the article by P. Power and J. Schoonees, "Understanding background mixture models for foreground segmentation", Proc. Images and Vision Computing, Auckland, NZ, 11 2002, and is still raising a lot of interest, as seen in the disclosures by Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction", Proceedings of the International Conference on Pattern Recognition, 2004, by Qi Zang and R. Klette, "Robust background subtraction and maintenance", ICPR '04: Proceedings of the 17th International Conference on Pattern Recognition, Volume 2, pages 90-93, Washington DC, USA, 2004, and by D.S. Lee, "Effective Gaussian mixture learning for video background subtraction", IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(5):827-832, May 2005, as well as in the International Patent Application WO 2005/024651. Nevertheless, GMM has some fundamental drawbacks as it strongly rests on the assumptions that the background is more frequently visible than the foreground and that its variance is significantly lower. Neither of these assumptions is valid for every time window. Furthermore, if highland low-frequency changes are present in the background, its sensitivity can't be accurately tuned and the model may adapt itself to the targets themselves or miss the detection of some high speed targets, as noted by A. Elgammal, R. Duraiswami, D. Harwood and L. S. Davis in "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance", Proceedings of the IEEE, Vol. 90, No. 7, July 2002, pages 1151 -1163. Finally, the estimation of the model parameters, especially of the variance, can become problematic in real-world noisy environments.

These problems have been addressed by a third category of background models, namely, so-called non-parametric models. These models are quite similar to parametric models except that they do not try to estimate parameters of predefined probability density functions but instead estimate a kernel density function directly from real pixel values without any assumptions about their underlying distribution. This avoids having to choose a function and estimating its distribution parameters.

One of the strengths of non-parametric kernel density estimation methods, as disclosed by A. Elgammal et al in their abovementioned paper, and by Z. Zivkovic and F. van der Heijden in "Efficient adaptive density estimation per image pixel for the task of background subtraction", Pattern Recognition Letters, 27(7):773-780, 2006, is their ability to circumvent a part of the parameter estimation step as they are based on actually observed pixel values. They model the

kernel density of a background pixel using a set of real values sampled from its recent history and can thus provide fast responses to high- frequency events in the background by directly including newly observed values in the pixel model. However, their ability to successfully handle concomitant events evolving at various speeds is debatable since they update their pixel and models in a first-in first-out manner. As a matter of fact, they often use two sub-models for each pixel: a short term model and a long term model.

While this can be a convenient solution, it seems artificial and requires fine tuning to work properly in a given situation. Finding a smoother lifespan policy for the sampled values that constitute the pixel models would be a salient improvement.

The use of real observed pixel values in non-parametric background pixel models, besides the fact that it reduces the number of parameters to estimate, makes also sense from a stochastic point of view since already observed values often have a higher probability to be observed again than never encountered ones. However, a whole set of parameters remains: the parameters of the kernel functions, unless using a significant amount of fixed-size kernels. As a matter of fact, a kernel density estimation method can also suffer from incorrect parameter estimation related problems.

Since all of the above mentioned methods work at pixel level, they have the additional drawback of requiring additional postprocessing to obtain spatial consistency.

To address these drawbacks, a new background detection method called SAmple CONsensus, or SACON, has been proposed by H. Wang and D. Suter in "A consensus-based method for tracking: Modelling background scenario and foreground appearance", Pattern Recognition, 40(3):1091-1105, March 2007. This article, which appears to constitute the closest prior art, discloses a method for detecting a background in an image selected from a plurality of related images, each formed by a set of pixels, and captured by an imaging device, this background detection method comprising the steps of:

- establishing, for a determined pixel position in said plurality of images, a background history comprising a plurality of addresses, in such a manner as to have a sample pixel value stored in each address;
- comparing the pixel value corresponding to said determined pixel position in the selected image with said background history, and, if said pixel value from the selected image substantially matches at least a predetermined number of sample pixel values of said background history:
  - classifying said determined pixel position as belonging to the image background; and
  - updating said background history by replacing one of its sample pixel values with said pixel value from the selected image.

More precisely, the SACON method keeps a history of N background sample pixel values for each pixel position. These samples are simply the N last observed values that fit within the model. N ranges from 20 to 200, but the authors concluded that N=20 is the absolute minimum that must be used in most practical tasks. However, diagrams provided in the paper indicate that N=60 should lead to the best performance.

The article does not mention any initialization process. It can however be assumed that the first N samples are used to fill the history, so that an initialization of 1 to 2 seconds, at a frame rate of 30 images per second, is necessary before the algorithm produces appropriate results.

The method simply counts the number of times that sample pixel values stored in the background history for a determined pixel position substantially match with the pixel value for that pixel position in the selected image. By "substantially match" it is meant throughout this text that the sample pixel value is within a given range around the pixel value of the selected image. In this SACON method, this range is adaptive and depends on the sample pixel values. For the predetermined minimum number of sample pixel values in the background history with which the pixel value of the selected image should match to be classified as belonging to the image background, the article mentions that it is determined empirically and that it depends on N.

The background history update in this closest prior art is carried out on a first-in, first-out basis, that is, the current pixel value will replace the oldest pixel value in the background history. Moreover, in order to remove ghost objects, once a pixel position has been classified as foreground for a predetermined number of times, it will be recognized as belonging to the background and the background history for that pixel position accordingly updated with the last N pixel values for that pixel position. Furthermore, to ensure spatial coherence, there is an additional updating mechanism at the blob level, which handles sets of connected (neighbouring) foreground pixels. Foreground blobs that are static, that is, whose size and centre are substantially constant, are also incorporated into the background. Such a method of providing spatial coherence does however require significant additional computing capacity, and will only be as reliable as the blob detection.

The SACON method does however present some significant drawbacks. Foremost among them is that, due to its first-in, first-out background history updating mechanism, the sample background pixel values are stored in the history for a fixed lapse of time. Because of this, the behaviour of this background detection method will vary significantly depending on the background evolution rate and the number of sample values in the background history.

An objective of the present invention is thus that of providing a background detection method adapted to a wider range of background evolution rates.

In order to achieve this objective, in the method of the present invention the address of the sample pixel value to be replaced is randomly chosen. This results in a smoother lifespan policy whose induced exponential decay for the expected remaining lifespan of the sample values in the history is better from a theoretical point of view and enables an appropriate behaviour of the technique for a wider range of background evolution rates even with a small history size.

Moreover, while the deterministic methods of the state of the art are vulnerable to bias and adversely affected by noise which has to be suppressed by previous filtering, this random technique has inherent noise resilience, making such filtering of the pixel values unnecessary. The method of the invention thus has the additional surprising advantages of bringing sharper segmentation results at the borders of the foreground objects, where decision boundaries of deterministic processes are often overlapping, and strongly reinforcing the robustness of the method, especially when dealing with noisy image data.

Furthermore, while the SACON method requires an adaptive comparison threshold between the current pixel value and the sample pixel values, the method of the invention also has the additional surprising advantage of achieving very satisfactory results with the same fixed comparison range for all pixels. This removes the need of the SACON method for estimating complex sets of threshold parameters for an effective background detection for all pixels in the image in a variety of environmental circumstances, which negatively affects the simplicity and practicality of the SACON method.

Advantageously, the method of the invention may further comprise a random update selection step, wherein it is randomly determined with a predetermined background history update probability whether said updating step is to be carried out or not. This allows a reduction of the number of necessary computations, while increasing the noise resilience and improving the time behaviour and decreasing the bias of the background detection method of the invention.

A further objective of the present invention is that of completing the background history without a lengthy initialisation. In a particular embodiment of the method of the invention, the step of establishing said background history may comprise reading pixel values in neighbouring pixel positions in at least one image of said plurality of images, other than said selected image, and storing these pixel values, randomly sorted in the plurality of addresses of said background history, as sample pixel values. This can provide a complete background history initialisation with a single image, greatly reducing the time necessary for this method to start providing reliable background detection. For increased noise and bias resilience, said neighbouring pixel positions may be randomly chosen within a neighbourhood of the determined pixel position.

A further objective of the present invention is that of providing a degree of spatial coherence without post-processing and without having to define blobs of purportedly interconnected pixels. In order to achieve this, in an advantageous embodiment of the background recognition method of the present invention, if said pixel value in said determined pixel position of said selected image

substantially matches at least said predetermined number of sample pixel values in the history of said determined pixel position, a background history corresponding to a neighbouring pixel, comprising another plurality of addresses, and established in such a manner as to have a sample pixel value stored in each address, is also updated by replacing the sample pixel value in a randomly chosen address in said background history of the neighbouring pixel with the pixel value of the determined pixel position in the selected image. Said neighbouring pixel background history may advantageously be randomly chosen within a neighbourhood of said determined pixel, so as to avoid bias. Also advantageously, the method may further comprise a random neighbouring pixel history update selection step, wherein it is randomly determined with a predetermined neighbouring pixel background history update probability whether a neighbouring pixel history is to be updated or not, so as to reduce computing operations while avoiding bias.

Since the background histories contain a plurality of sample pixel values, irrelevant information accidentally inserted into the neighbouring pixel background history does not affect the accuracy of the method. Furthermore, the erroneous diffusion of irrelevant information is blocked by the need to match an observed value before it can further propagate. This natural limitation inhibits error propagation.

In order to broaden even further the range of background evolution rates to which the background recognition system of the present invention is adapted, in an advantageous embodiment of this method at least one auxiliary history, corresponding to said determined pixel position and also comprising a plurality of addresses, is also established in such a manner as to have a sample pixel value stored in each address, and, if said pixel value from the selected image does not substantially match at least said predetermined number of sample pixel values of the background history, the method may also comprise the following steps:

- comparing said pixel value from the selected image with said auxiliary history, and, if it substantially matches at least a further predetermined number of sample pixel values of said auxiliary history:
  - randomly determining with an auxiliary history update probability higher than the background history update probability whether the auxiliary history is to be updated or not-; and, if yes,
  - updating said auxiliary history by replacing a sample pixel value from a randomly selected address in the auxiliary history with said pixel value from the selected image.

This provides at least one auxiliary image layer with a different time response characteristic to which the current pixel values can be compared, allowing discriminating between background changes and motion at various speeds.

The at least one auxiliary history may advantageously be used for ghost suppression. At the first observation of a previously unseen part of the scene, the corresponding pixels are wrongly classified as foreground and constitute so-called ghost data. In an embodiment of the invention with only the background layer, an insertion in the background history of a determined pixel of neighbouring pixel values for ensuring spatial consistency, as described above, will progressively move these ghost pixels to the background. However, this spatial diffusion process can be slow or even ineffective when neighbouring pixels exhibit major dissimilarities, such as -at the boundaries of a background object.

In order to address this problem, if said pixel value from the selected image substantially matches at least said further predetermined number of sample pixel values of said auxiliary history, the method may further comprise the steps of:

- randomly determining with a predetermined background seeding probability lower than the auxiliary history update probability whether the background history is also to be updated with the pixel value from the selected image; and, if yes,
- updating the background history by replacing a sample pixel value from a randomly selected address in the background history with said pixel value from the selected image.

These additional steps seed the background history with new values which can accelerate the spatial diffusion process for ghost suppression inside the objects.

To further reduce the processing necessary for the background detection, in an advantageous embodiment the background detection method of the invention may also comprise the steps of dividing said selected image into a plurality of sub-sampling areas, and randomly selecting one such determined pixel position in each sub-sampling area as representative of the whole sub-sampling area. In previous methods it has been proposed to select sample pixels as located on a regular grid. When using a linear model, this required a preliminary filtering in order to be compliant with the Nyquist stability, which had the inconvenient of introducing new pixel values not actually present in the original image, and thus possibly introduce unwanted artefacts. By randomly selecting the sampled pixels, preliminary filtering becomes unnecessary, and such artefacts as may result from such filtering are prevented.

Advantageously, the method further comprises a step of reallocating the background history from another pixel position within said image to said determined pixel position in response to image motion, such as panning or zooming. This provides a compensation for apparent background motion due to image motion, in particular for large imaging device motion, such as imaging device travelling.

Advantageously, since existing imaging devices often incorporate embedded motion sensors, the pixel position from which the background history is reallocated may be based on the motion detected by a motion sensor of the imaging device.

Advantageously, since existing imaging devices may be equipped with zoom devices with zoom value sensors, the pixel position from which the background history is reallocated may be based on a zoom value of the imaging device.

Advantageously, since existing imaging devices may be equipped with image processors for carrying out block matching algorithms, the pixel position from which the background history is reallocated may be determined based on a correspondence obtained by a block matching algorithm, such as, for example, that used in the standard MPEG video encoding process.

The present invention relates also to a data processing device programmed so as to carry out the image background recognition method of the invention; to a data storage medium comprising a set of instructions for a data processing device to carry out an image background recognition method according to the invention; to a set of signals in magnetic, electromagnetic, electric and/or mechanical form, comprising a set of instructions for a data processing device to carry out an image background recognition method according to the invention; and/or to a process of transmitting, via magnetic, electromagnetic, electric and/or mechanical means, a set of instructions for a data processing device to carry out an image background recognition method according to the invention.

In particular, said data processing device programmed so as to carry out the image background recognition method of the invention may be embedded in an imaging device, such as, for example, a digital camera.

Said data processing device programmed so as to carry out the image background recognition method of the invention may also belong to a video-game system or video-surveillance system further comprising an imaging device.

As "data storage medium" is understood any physical medium capable of containing data readable by a reading device for at least a certain period of time. Examples of such data storage media are magnetic tapes and discs, optical discs (read-only as well as recordable or re-writable), logical circuit memories, such as read-only memory chips, random-access memory chips and flash memory chips, and even more exotic data storage media, such as chemical, biochemical or mechanical memories.

As "electromagnetic" any part of the electromagnetic spectrum is understood, from radio to UV and beyond, including microwave, infrared and visible light, in coherent (LASER, MASER) or incoherent form.

As "object" is understood any observable element of the real world, including animals and/or humans. A particular embodiment of the invention will now be described in an illustrative, but not restrictive form, with reference to the following figures:

- Fig. 1 shows a schematic view of a set of images comprising an image including a determined pixel with a background history and neighbouring pixels;
- Fig. 2 shows a flowchart illustration of an embodiment of the background detection method of the invention;
- Fig. 3 shows a diagram of a pixel value and sample pixel values of the background history of that determined pixel;
- Figs. 4a-4c shows diagrams with the sample pixel values of possible alternative background histories resulting from updating the background history of Fig. 3 by replacing a randomly chosen one of its sample pixel values with the current pixel value;
- Figs. 5a-5c shows a diagram showing the reallocation of background histories in response to, respectively, panning, zooming in and zooming out;
- Fig. 6 shows a schematic view of an image divided in sub- sampling areas;
- Fig. 7 shows a diagram of a background history and a plurality of auxiliary histories for a determined pixel position;
- Fig. 8 shows a flowchart illustration of part of an alternative background detection method of the invention;
- Fig. 9 shows an imaging device with an embedded data processing device programmed to carry out an embodiment of the method of the invention; and
- Fig. 10 shows an imaging device connected to a data processing device programmed to carry out an embodiment of the method of the invention.

Fig. 1 shows a set of images I. These images may have been, for example, successively captured by an imaging device. Each image I is formed by a plurality of pixels, each single pixel in an image I having a dedicated pixel position x and a pixel value $I(x)$. For ease of understanding, in the accompanying drawings, the pixel position x is shown as two-dimensional, but it could have any number of dimensions. For 3D images, for instance, the pixel position x may have three dimensions. The pixel value $I(x)$ in the illustrated embodiment is three- dimensional, in the form of RGB- or HSV-triplets for obtaining a polychromatic image. In alternative embodiments, it could however have any other number of dimensions. A determined pixel 101 will have a number of neighbouring pixels 102.

In the selected image I_t corresponding to observation t, pixel 101 has a pixel value $I_t(x)$. For the pixel position x of pixel 101, there is a background history 103 with N addresses 104-1 to 104-N, in which sample pixel values $b_i(x)$ to $b_N(x)$ are stored, representing a background model for that pixel position x. Turning now to the flowchart represented in Fig. 2, it can be determined whether pixel 101 in the selected image I_t belongs to the background by reading: in step 201, its value $I_t(x)$ in said selected image I_t, and comparing it, in step 202, with the background history 103. This comparison step 202 is illustrated in Fig. 3 for a background history 103 of N=6 addresses containing sample pixel values $b_i(x)$ to $b_6(x)$, and the pixel value $I_t(x)$ in a three-dimensional space. Instead of estimating a probability density function, the pixel value $I_t(x)$ is compared to the closest sample pixel values within the background history 103, so as to limit the influence of outliers in said background history 103. To achieve this, a sphere $S_{R,t}(x)$ of radius R centred on the pixel value $I_t(x)$ is defined. Then a cardinality, that is, the number C of set elements of the intersection set of this sphere and the set of sample pixel values $\{b_1(x), b_2(x), ..., b_N(x)\}$ is determined:

$$C = \sharp\{S_{R,t}(x) \cap \{b_1(x), b_2(x), ..., b_N(x)\}\} \tag{A.2}$$

Pixel 101 will be classified in step 203 as belonging to the background if this cardinality C is higher than or equal to a minimum number. The two parameters determining the accuracy of the background recognition are thus the radius R, forming the comparison threshIold, and said

minimum matching cardinality C. Experiments have shown that a single fixed radius R and a minimum cardinality of 2 for N=6 sample pixel values offer excellent performances. There is no need to adapt these parameters during the background detection nor is it necessary to change them for different pixel positions x within the image I. The sensitivity of the model can be adjusted with the ratio between the minimum matching cardinality C and the total number N of sample pixel values $b_1(x), b_2(x), ..., b_N(x)$.

While in the illustrated embodiment N=6, in alternative embodiments N could be a different number according to circumstances. Advantageously, but not necessarily, N may be in a range between 3 and 50. A preferred range would be between 5 and 30, and in a particularly preferred embodiment N=20.

Turning back to the flowchart of Fig. 2, the next step 204 is that of randomly determining whether the pixel value $I_t(x)$ is to be used in updating the background history 103. If this is the case, the update will take place in the next step 205. In this background recognition method a conservative update scheme is used, that is, only if the pixel 101 is confirmed as belonging to the background in the comparison step 202 may its current pixel value $I_t(x)$ be used to update its background history 103. To achieve accurate results over time and to handle new objects appearing in a scene, the background history 103 has to be updated frequently. Since in this method the pixel value $I_t(x)$ is compared directly to the sample pixel values $b_t(x)$ to $b_N(x)$ in the background history 103, and there is thus no smoothing effect on the sample pixel values, the question of which sample pixel values are to be kept in the background history 103, and how, is important.

In the random update selection step 204, a random technique is used to determine whether a sample pixel value will be effectively replaced in the background history 103. Even if the pixel 101 has been confirmed as belonging to the background, its pixel value $I_t(x)$ will only randomly be used to replace one of the sample pixel values in the background history 103, for example with a random history update probability of 1 in 16. Then, in updating step 205, the pixel value $I_t(x)$ will replace a sample pixel value in a randomly chosen address 104-i, wherein i is a random number between 1 and N, of the background history 103. This is illustrated in Figures 4a to 4c, showing three alternative updates to the background history 103 illustrated in Fig. 3. In Figure 4a, the pixel value $I_t(x)$ has replaced sample pixel value $b_3(x)$. In Figure 4b, it has replaced $b_2(x)$ instead, whereas in Fig. 4c, it has replaced $b_5(x)$.

When the background history 103 is updated, the probability of a sample pixel value to be conserved is:

$$\frac{N-1}{N} \tag{A.3}$$

After the updating step 205, the method may be repeated again for the pixel value $I_{t+1}(x)$ of pixel 101 in the next image I_{t+1}. With every successive effective update, the expected remaining lifespan of each sample pixel value in the background history 103 will decay exponentially.

To respond to some situations, such as small movements and oscillations of the imaging system, or slowly evolving background objects, it is favourable to introduce some spatial consistency between the background history 103 of pixel 101, and those of pixels 102 in a spatial neighbourhood $NG(x)$ of pixel 101. These pixels 102 may include directly neighbouring pixels, as illustrated in Fig. 1, but also those in a broader spatial neighbourhood $NG(x)$. To achieve this, in a neighbouring pixel background history update selection step 206 it may be randomly decided, possibly with a different probability than for the update selection step 204, whether, in a second updating step 207, the pixel value $I_t(x)$ of pixel 101 is to be used to update the background history of a randomly chosen neighbouring pixel 102. As in the previous updating step 205, the background history 103 is updated by replacing a sample pixel value at a randomly chosen address 104-i of said background history 103.

To carry out the earlier comparison step 202 it is however necessary to establish the background history 103 first. While, after a certain time, all the sample pixel values $b_1(x)$ to $b_N(x)$ in the background history 103 can be expected to originate from previous updates, at the beginning of a sequence of images there are no previous pixel values for pixel 101 available as sample pixel

values. To establish the background history 103 an initialisation step 208 is carried out using, for example, the first image I_1 of the set of images I. In this initialisation step 208, the pixel values of randomly chosen neighbouring pixels 102 in, for example, said first image I_1, are selected as the initial sample pixel values $b_i(x)$ to $b_N(x)$ in the N addresses 104-1 to 104-N of the background history 103 of pixel 101. A complete background model can thus be established from the first image I_1 in the sequence without a lengthy initialisation period. While this initialisation step 208 has the drawback of generating ghost objects in response to moving objects in the first image I_1, later updates will ensure that they fade over time.

Neighbouring pixels 102 from differently-sized spatial neighbourhoods $NG(x)$ may be used for the initialisation step 208 and for the second updating step 207. For instance, to initialise a background history 103 comprising N=20 sample pixel values, a broader spatial neighbourhood $NG(x)$ of 5x5, or even 7x7 pixels around said determined pixel 101 may be chosen. On the other hand, for the second updating step 207 for improving spatial consistency, a narrow spatial neighbourhood $NG(x)$ of 3x3 pixels, or even of only the four closest neighbouring pixels 102 (left, right, top and bottom) may be chosen.

A synthetic background image B_t can be created for a given image I_t using:

- the pixel values of the pixels 101 in said image I_t which are classified as belonging to the background;
- for the pixels 101 in said image I_t which are not classified as belonging to the background, one of the sample pixel values in its background history 103, preferably from a randomly selected address 104-i in said background history 103.

This synthetic background image B_t may be used, for instance, in a shadow detection post-processing technique in which the image I_t will be compared with the background image B_t and/or, for the detection of abandoned objects in video-surveillance, where said background image B_t will be used to regularly update a reference image R which will be compared with current images in a manner known to the skilled person to detect abandoned objects.

In practical situations, neither the imaging device nor the objects in the scene are necessarily fixed. Some imaging devices may provide motion parameters, such as a translation vector, parameters of an affine transformation, or parameters of axial transformations, such as zoom values. If at least a minimum such motion is detected in a step 209, this allows for the reallocation of the background history from another pixel 105 to the determined pixel 101, in a motion compensation step 210 before the comparison step 202, as illustrated in Figures 5a to 5c.

Said motion parameters determine the relative position of pixel 105 to pixel 101, and may be obtained by motion sensors and/or zoom sensors embedded in the imaging device, and/or by techniques establishing a correspondence between areas in different images I directly based on the image data. For example, many consumer market digital cameras have embedded motion sensors for digital motion compensation, which may also be used to obtain those motion parameters. Also, encoders compliant with the ISO MPEG standards apply a block matching algorithm for building a one-to-one correspondence between areas of two images which may also be used to obtain those motion parameters.

In this reallocation process, if two pixel positions are uniquely paired to each other, the background history 103 is simply reallocated from its old location at pixel 105 to its new location at pixel 101. However, if an image I_t contains at least one pixel not present in the previous image I_{t-1}, sample pixel values from the background history of the closest pixel in the previous image I_{t-1}, or, alternatively, random values, may be used to fill its background history 103.

Fig. 5a illustrates background history reallocation in the case of a panning motion of the imaging device. As the imaging device travels from right to left on an axis parallel to the image plane, the background histories of pixels 105 will be reallocated to the corresponding pixels 101 to their right, leading to consistent results of the background detection method of the invention for those pixels 101, despite the fact that the imaging device is moving. However, while in two consecutive images I_{t-1}, I_t a majority of pixels may correspond to the intersection between the scenes seen in both images I_{t-1}, I_t, some pixels from I_{t-1} will move out of the image frame, while

others will appear in I_t corresponding to parts of the scene which were visible in the previous image I_{t-1}. The background histories of the former (in the illustrated example, those on the right side) may be discarded or saved for latter use. The background histories of the latter (in the illustrated example, those on the left side) may be instantaneously initialised or recovered from a previous memorisation, for example if the imaging device had previously travelled in the opposite direction.

Fig. 5b illustrates background history reallocation in the case of a closing-up motion or a zoom-in of the imaging device. In this case, the reallocated background histories will move away from the centre of the image I_t. The background histories of the pixels closest to the edge of the previous image I may be discarded or saved for later use. As the number of pixels corresponding to a given part of the scene will increase in the image I_t with respect to the previous image I_{t-1}, the background history of a pixel 105 in image I_{t-1} may be reallocated to several pixels 101 in image I_t.

Fig. 5c illustrates background history reallocation in the case of a retreating motion or a zoom-out of the imaging device. In this case, the reallocated background histories will move towards the centre of the image I_t. As the number of pixels corresponding to a given part of the scene will decrease, not all background histories from pixels 105 in the previous image I_{t-1} will however find a corresponding pixel 101 in image I_t to be reallocated to. Close to the edge of the image I_t will appear pixels corresponding to parts of the scene that were not visible in the previous image I_{t-1}. The histories of these pixels may be instantaneously initialised or recovered from a previous memorisation, for example if the imaging device had previously zoomed in.

If the block-matching algorithm detects a moving object, its corresponding pixel values $I_t(x)$ in the image I_t may be excluded from the background history updates.

In a first embodiment of the invention, the abovementioned method will be carried out on every pixel of each image I of the set of images. However, in some situations, processing speed may be more important than using the whole definition of the image. Since the method offers good performances at any scale, it is then possible to detect the image background by carrying out the method of the invention on a selected subset of pixels from the set of pixels of each image I, that is, by sub-sampling the original image data set, reducing the image to a lower resolution. The pixels in this subset may be randomly chosen. Figure 5 shows an example of how this may be carried out. In the embodiment of Figure 6, each image I of the sequence is divided into a plurality of sub-sampling areas 601. In each sub-sampling area 601, a single determined pixel 101 is randomly chosen as representing the whole sub-sampling area 601. For each image I in the set of images, this random pixel selection is carried out once again.

In another alternative embodiment of the invention, illustrated in Figs. 7 and 8, a set of auxiliary histories 103',103",103"' and 103^{IV} are established for the pixel position x of each determined pixel 101. In alternative embodiments of the invention, different numbers of auxiliary histories may be used.

Each one of these auxiliary histories 103', 103", 103"' and 103^{IV} has, like background history 103, the following parameters:

- number N of available addresses 104',104",104"' or 104^{IV} for sample pixel values in, respectively, said additional history 103', 103", 103"' or 103^{IV};
- maximum distance R between a current pixel value $I_t(x)$ and a sample pixel value in said auxiliary history 103', 103", 103"' or 103^{IV} for the pixel value $I_t(x)$ to be considered to substantially match said sample pixel value;
- number C of sample values in said auxiliary history 103', 103", 103"' or 103^{IV} which a pixel value $I_t(x)$ must substantially match to be considered as belonging to said auxiliary history 103', 103", 103"' or 103^{IV};
- and history update probability defining the probability of using a pixel value $I_t(x)$ for updating said auxiliary history 103', 103"... or 103^{IV} when it is considered for an update.

In the illustrated embodiment, each auxiliary history 103', 103", 103"' and 103^{IV} of a pixel 101 may have the same number N, maximum distance R and threshold cardinality C than the background history 103, even if different parameters could also be chosen. However, each auxiliary history 103', 103", 103"' and 103^{IV} is characterised by a different auxiliary history update probability higher than the background history update probability. An increasing auxiliary history update probability decreases the inertia of each auxiliary history, defined as its slowness to adapt to changes in successive images I. A history with high inertia will thus account for pixel values observed over a large number of images I, while a history with low inertia will be able to model fast changing values. For example, with a background history update probability of 1/16, the auxiliary history update probabilities used for updating the auxiliary histories 103',103",103"' and 103^{IV} of the illustrated embodiment may be, respectively, 1/8,1/4,1/2 and 1.

As in the previously described embodiment, a background history 103 may be initialised by filling its addresses 104-1 to 104-N with randomly sorted values of randomly chosen neighbouring pixels 102 in a first image I_1. However, as it may be assumed that all pixels in said first image I_1 belong to the background, each one of the auxiliary histories 103', 103", 103"' and 103^{IV} may be initialised with random values. The auxiliary histories 103', 103", 103"' and 103^{IV} will then adapt as soon as more information becomes available.

The current pixel value $I_t(x)$ will be hierarchically compared to the corresponding history 103, 103', 103", 103"' and 103^{IV}, as illustrated in Fig. 8. Starting from the background history 103, the value $I_t(x)$ will be recursively checked against each history 103, 103', 103", 103"' and 103^{IV}.

So, if the pixel value $I_t(x)$ of the selected image I_t did not match the background history 103 in the abovementioned comparison step 202, it will then be compared with the first auxiliary history 103' in a first auxiliary history comparison step 202'. If it substantially matches at least C sample pixel values stored in the addresses 1O4'-1 to 1O4'-N of said first auxiliary history 103', it will be considered, in a random update selection step 204' with a first auxiliary history update probability such as, for example, the abovementioned 1/8, whether to update the first auxiliary history 103' with the pixel value $I_t(x)$ in a first auxiliary history update step 205', wherein said pixel value $I_t(x)$ will replace a sample pixel value in a randomly chosen address 104'-i of the first auxiliary history 103'.

If the pixel value $I_t(x)$ does however not match said at least C sample pixel values stored in said first auxiliary history 103', it will then be compared in a second auxiliary history comparison step 202" with the second auxiliary history 103". Again, if it substantially matches at least C sample pixel values stored in said second auxiliary history 103", it will proceed to a similar random update selection step (not shown), with a higher update probability such as, for example, the abovementioned 1/8, and eventually a similar updating step (not shown) of the second auxiliary history 103". If the pixel value $I_t(x)$ does not substantially match at least C sample pixel values stored in said second auxiliary history 103", an analogous procedure will be executed for the third auxiliary history 103"', with an even higher update probability such as, for example, the abovementioned 1/2, and, if the pixel value $I_t(x)$ does not substantially match at least C sample pixel values stored in said third auxiliary history 103"', an analogous procedure will be executed for the fourth auxiliary history 103^{IV}, with an even higher update probability such as, for example, the abovementioned 1.

The method of the invention may also incorporate for each auxiliary history, such as the four auxiliary histories 103', 103", 103"' and 103^{IV} of the illustrated embodiment, the same techniques as described above for the background history 103 for ensuring spatial consistency, compensating image motion and/or reducing processing time by spatial sub-sampling.

When a block-matching algorithm is used, at least one auxiliary history of a pixel corresponding to a moving object detected by said black-matching algorithm may be reallocated according to its movement to increase the consistency of this auxiliary history.

In an image I_t, the pixels 101 with pixel values $I_t(x)$ that have been matched to an auxiliary history may be used to generate an image layer possibly intermediary between the background and the foreground.

At least one auxiliary history may be used for ghost suppression. For this purpose, the pixel value $I_t(x)$ of a pixel 101 matched to an auxiliary history may also be used to update the back-

ground history 103, or a lower-ranked auxiliary history. This is also illustrated in the embodiment shown in Fig. 8, wherein, if the pixel value $I_t(x)$ is matched with the first auxiliary history 103', it is determined, in another update selection step 801 with a lower probability, such as, for example 1/2048, than the probability of updating the first auxiliary history 103', whether to update, in a different updating step 802, the background history 103 with said pixel value $I_t(x)$. This will insert small background seeds inside the ghost data which will help the abovementioned spatial diffusion process to remove ghosts. Eventually, the same procedure may be used to update the first auxiliary history 103' with a pixel value $I_t(x)$ that has been matched in step 202" to the second auxiliary history 103", to update the second auxiliary history 103" with a pixel value $I_t(x)$ that has been matched in step 202"' to the third auxiliary history 103"', and so on.

The background detection method of the invention may be carried out with assistance of a data processing device, such as, for example, a programmable computer, connected to the imaging device. In such a case, the data processing device may receive instructions for carrying out this background detection method using a data storage medium, or as signals in magnetic, electromagnetic, electric and/or mechanical form.

In the various embodiments of the invention, there is extensive use of random numbers. These numbers may be provided by a random number generator. However, since such random numbers cannot be provided by a deterministic computer, a pseudorandom number generator may be used instead with properties similar to those of a true random number generator. Another alternative is the use of a large look- up list of previously generated random or pseudorandom numbers.

The background detection method of the invention may, for example, be applied to video-surveillance, professional and/or consumer digital still and/or video cameras, computer and video-game devices using image capture interfaces, satellite imaging and Earth observation, automatic image analysis and/or medical imaging systems.

Fig. 9 illustrates a possible application of the invention with an imaging device 901 in the particular form of a digital camera with an embedded data processing device 902 programmed to carry out the method of the invention. Fig. 10 illustrates another possible application of the invention with an imaging device 901 connected for video surveillance or interaction with a video-game system with a data processing device 902 programmed to carry out the method of the invention.

Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader scope of the invention as set forth in the claims. Accordingly, the description and drawings are to be regarded in an illustrative sense rather than a restrictive sense.

A.3 Claims

1. A method for detecting a background in an image selected from a plurality of related images, each formed by a set of pixels, and captured by an imaging device, this background detection method comprising the steps of:

    - establishing, for a determined pixel position in said plurality of images, a background history comprising a plurality of addresses, in such a manner as to have a sample pixel value stored in each address;
    - comparing the pixel value corresponding to said determined pixel position in the selected image with said background history, and, if said pixel value from the selected image substantially matches at least a predetermined number of sample pixel values of said background history:
        - classifying said determined pixel position as belonging to the image background; and
        - updating said background history by replacing one of its sample pixel values with said pixel value from the selected image; and characterized in that:

- the address of the sample pixel value to be replaced is randomly chosen.

2. A method according to claim 1, further comprising a random update selection step, wherein it is randomly determined with a predetermined background history update probability whether said updating step is to be carried out or not.

3. A method according to claim 2, wherein at least one auxiliary history, corresponding to said determined pixel position and also comprising a plurality of addresses, is also established in such a manner as to have a sample pixel value stored in each address, and further comprising the steps, if said pixel value from the selected image does not substantially match at least said predetermined number of sample pixel values of the background history, of:

   - comparing said pixel value from the selected image with said auxiliary history, and, if it substantially matches at least a further predetermined number of said sample pixel values in said auxiliary history:
     - randomly determining with an auxiliary history update probability factor higher than the background history update probability factor whether the additional history is to be updated or not; and, if yes,
       * updating said auxiliary history by replacing a sample pixel value from a randomly selected address in the auxiliary history with said pixel value from the selected image.

4. A method according to claim 3, which, if said pixel value from the selected image substantially matches at least said further predetermined number of sample pixel values of said auxiliary history, further comprises the steps of:

   - randomly determining with a predetermined background seeding probability lower than the auxiliary history update probability whether the background history is also to be updated with the pixel value from the selected image; and, if yes,
   - updating the background history by replacing a sample pixel value from a randomly selected address in the background history with said pixel value from the selected image.

5. A method according to any one of the previous claims, wherein the step of establishing said background history comprises reading pixel values in neighbouring pixel positions in at least one image of said plurality of related images other than said selected image, and storing these pixel values, randomly sorted in the plurality of addresses of said background history, as sample pixel values.

6. A method according to claim 5, wherein said neighbouring pixel positions are randomly chosen within a neighbourhood of the determined pixel position.

7. A method according to any one of the previous claims, wherein, if said pixel value in said determined pixel position of said selected image substantially matches at least said predetermined number of sample pixel values in the background history of said determined pixel position, a background history corresponding to a neighbouring pixel, comprising another plurality of addresses, and established in such a manner as to have a sample pixel value stored in each address, is also updated by replacing the sample pixel value in a randomly chosen address in said background history of the neighbouring pixel with the pixel value of the determined pixel position in the selected image.

8. A method according to claim 7, wherein the position of the neighbouring pixel is randomly chosen within a neighbourhood of the determined pixel position.

9. A method according to any one of claims 7 or 8, further comprising a random neighbouring pixel history update selection step, wherein it is randomly determined with a predetermined neighbouring pixel background history update probability whether a neighbouring pixel history is to be updated or not.

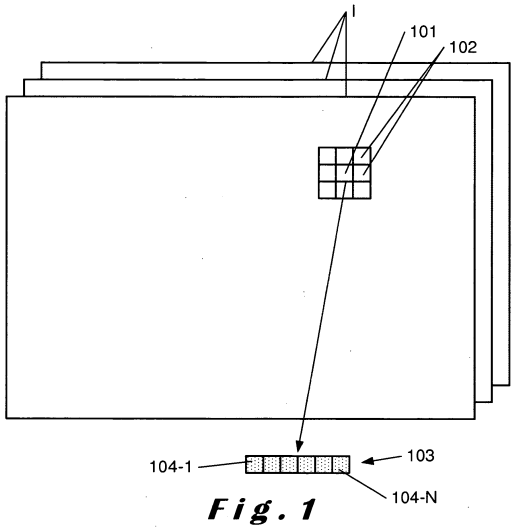


Fig. 1

10. A method according to any one of the previous claims, also comprising the steps of:

    - dividing said selected image into a plurality of sub-sampling areas; and
    - randomly selecting one such determined pixel position in each sub- sampling area as representative of the whole sub-sampling area.

11. A method according to any one of the previous claims, further comprising a step of reallocating a background history of another pixel position to said determined pixel position within said image in response to image motion, such as panning or zooming.

12. A method according to claim 11, wherein the pixel position from which the background history is reallocated is based on the motion detected by a motion sensor of the imaging device.

13. A method according to claims 11 or 12, wherein the pixel position from which the background history is reallocated is based on a zoom value of the imaging device.

14. A method according to claim 11, wherein the pixel position from which the background history is reallocated is determined based on a correspondence obtained by a block matching algorithm.

15. A data processing device programmed so as to carry out an image background recognition method according to any one of the previous claims.

16. An imaging device with an embedded data processing device according to claim 15.

17. A video-surveillance system comprising at least one imaging device and a data processing device according to claim 15.

18. A video-game system comprising at least one imaging device and a data processing device according to claim 15.

19. A data storage medium comprising a set of instructions for a data processing device to carry out an image background recognition method according to any one of claims 1 to 15.

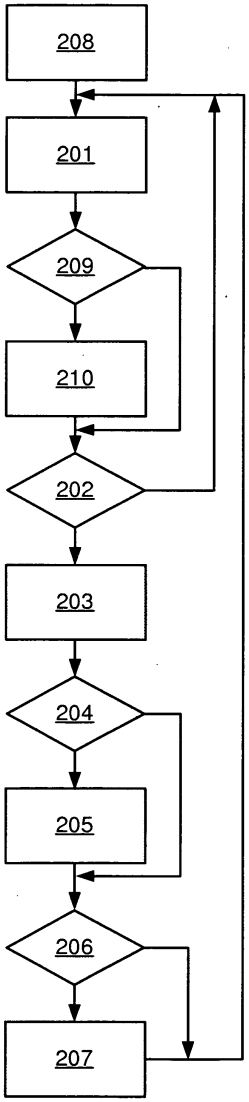

208
201
209
210
202
203
204
205
206
207

Fig. 2

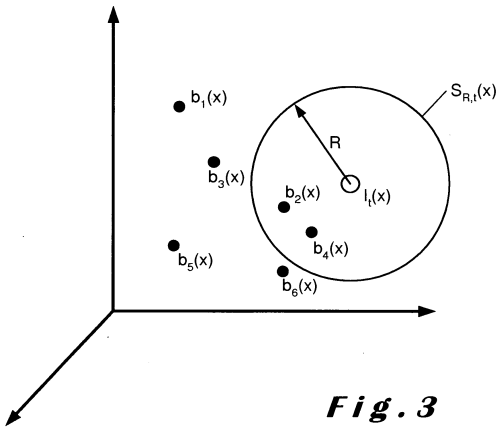

$b_1(x)$
$b_3(x)$
$b_2(x)$
$I_t(x)$
R
$S_{R,t}(x)$
$b_4(x)$
$b_5(x)$
$b_6(x)$

Fig. 3

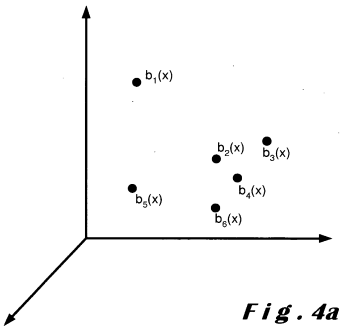


Fig. 4a

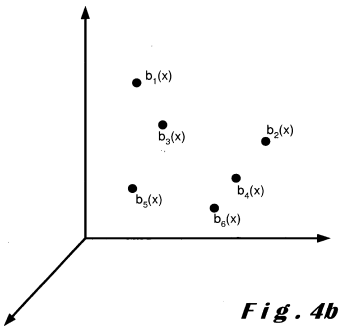


Fig. 4b

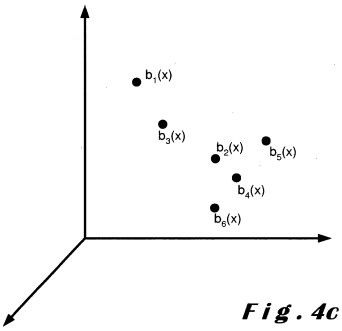


Fig. 4c

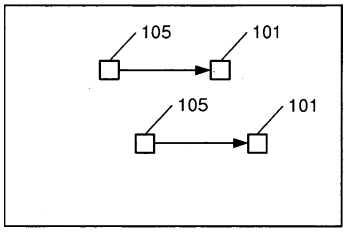


Fig. 5a

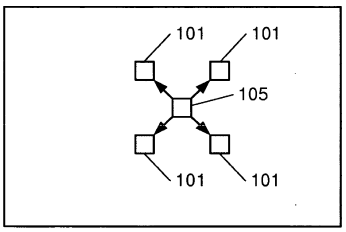


Fig. 5b

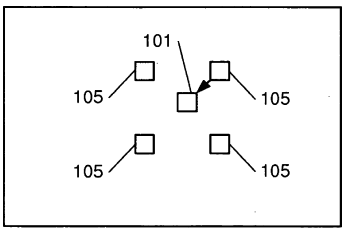


Fig. 5c

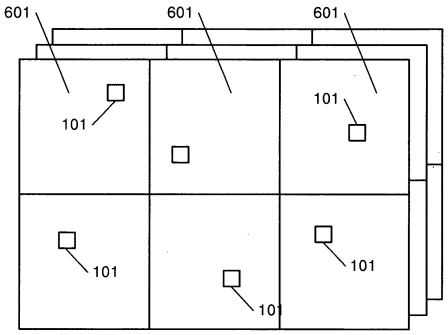


Fig. 6

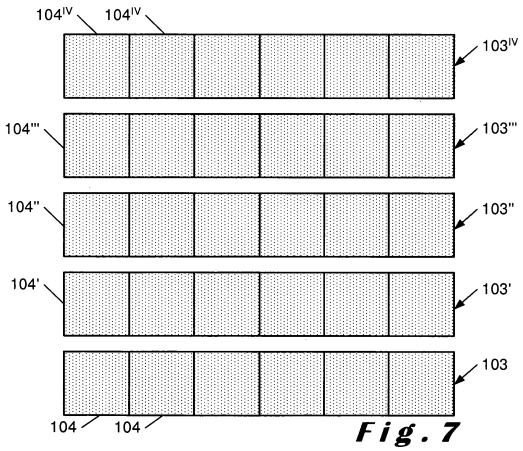


Fig. 7

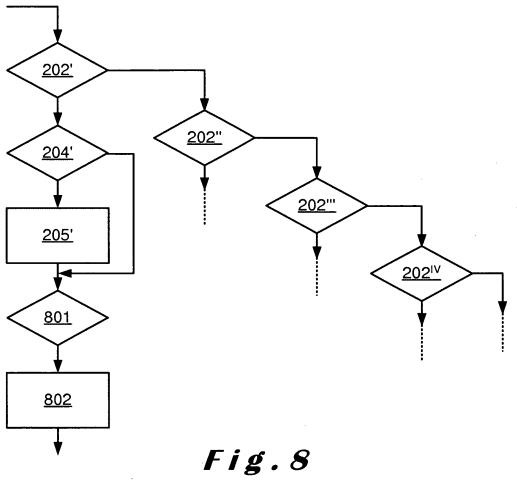


Fig. 8

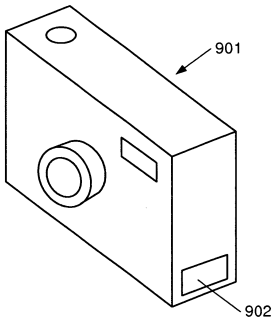


Fig. 9

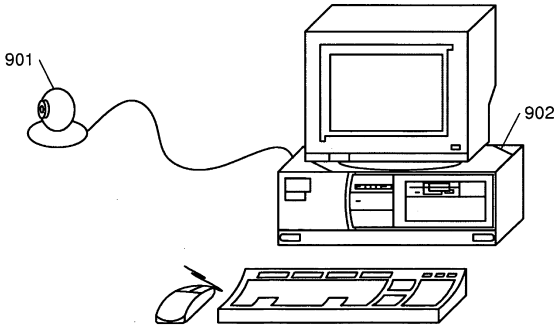


Fig. 10

List of publications

Articles in journals

1. A. Borghgraef, O. Barnich, F. Lapierre, M. Van Droogenbroeck, W. Philips, and M. Acheroy. **An evaluation of pixel-based methods for the detection of floating objects on the sea surface.** EURASIP Journal on Advances in Signal Processing, 2010:11 pages, 2010.

2. O. Barnich and M. Van Droogenbroeck. **Frontal-view gait recognition by intra- and inter-frame rectangle size distribution.** Pattern Recognition Letters, 30(10):893-901, July 2009.

Patents

1. M. Van Droogenbroeck and O. Barnich. **Visual background extractor.** International patent application published under the Patent Cooperation Treaty (PCT), WO 2009/007198 A1, January 2009.

2. M. Van Droogenbroeck and O. Barnich. **Visual background extractor.** European patent application, EP 2015252 B1, February 2010.

Conference articles

1. S. Piérard, O. Barnich, and M. Van Droogenbroeck. **A virtual curtain for the detection of humans and access control.** In Advanced Concepts for Intelligent Vision Systems (ACIVS 2010), Sydney, Australia, 12 pages, December 2010.

2. S. Piérard, V. Pierlot, O. Barnich, and M. Van Droogenbroeck, and J. Verly. **A platform for the fast interpretation of movements and localization of users in 3D applications driven by a range camera.** In 3DTV Conference, Tampere, Finland, June 2010.

3. O. Barnich and M. Van Droogenbroeck. **Design of a morphological moving object signature and application to human identification.** In International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009), pages 853-856, April 2009.

4. O. Barnich and M. Van Droogenbroeck. **ViBe: a powerful random technique to estimate the background in video sequences.** In International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009), pages 945-948, April 2009.

5. J. Leens, S. Piérard, O. Barnich, M. Van Droogenbroeck, and J.-M. Wagner. **Combining Color, Depth, and Motion for Video Segmentation.** Volume 5815 of Lecture Notes in Computer Science, pages 104-113. Springer Verlag, 2009.

6. O. Barnich, S. Jodogne, and M. Van Droogenbroeck. **Robust analysis of silhouettes by morphological size distributions.** Volume 4179 of Lecture Notes on Computer Science, pages 734-745. Springer Verlag, 2006.

7. M. Van Droogenbroeck and O. Barnich. **Design of Statistical Measures for the Assessment of Image Segmentation Schemes.** Volume 3691 of Lecture Notes in Computer Science, pages 280-287. Springer Verlag, 2005.

Bibliography

[1] R. Abbott and L. Williams. Multiple target tracking with lazy background subtraction and connected components analysis. *Machine Vision and Applications*, 20(2):93–101, February 2009.

[2] S. Argyropoulos, D. Tzovaras, D. Ioannidis., and M.G. Strintzis. A channel coding approach for human authentication from gait sequences. *IEEE Transactions on Information Forensics and Security*, 4(3):428–440, September 2009.

[3] A. Bagdanov and M. Worring. Granulometric analysis of document images. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 478–481, Québec, Canada, August 2002.

[4] O. Barnich, S. Jodogne, and M. Van Droogenbroeck. Robust analysis of silhouettes by morphological size distributions. In *Advanced Concepts for Intelligent Vision Systems (ACIVS 2006)*, volume 4179 of *Lecture Notes on Computer Science*, pages 734–745. Springer, September 2006.

[5] O. Barnich and M. Van Droogenbroeck. Design of a morphological moving object signature and application to human identification. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009)*, pages 853–856, April 2009.

[6] O. Barnich and M. Van Droogenbroeck. Frontal-view gait recognition by intra- and inter-frame rectangle size distribution. *Pattern Recognition Letters*, 30(10):893–901, July 2009.

[7] O. Barnich and M. Van Droogenbroeck. ViBe: a powerful random technique to estimate the background in video sequences. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 945–948, April 2009.

[8] C. BenAbdelkader, R. Cutler R., and L. Davis. Motion-based recognition of people in Eigen-Gait space. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, pages 267–272, May 2002.

[9] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 1–4, December 2008.

[10] N. Boulgouris and Z. Chi. Gait recognition using radon transform and linear discriminant analysis. *IEEE Transactions on Image Processing*, 16(3):731–740, March 2007.

[11] N. Boulgouris, D. Hatzinakos, and K. Plataniotis. Gait recognition: a challenging signal processing technology for biometric identification. *IEEE Signal Processing Magazine*, 22(6):78–90, November 2005.

[12] N. Boulgouris, K. Plataniotis, and D. Hatzinakos. Gait recognition using linear time normalization. *Pattern Recognition*, 39(5):969–979, May 2006.

[13] T. Bouwmans, F. El Baf, and B. Vachon. Statistical background modeling for foreground detection: A survey. In *Handbook of Pattern Recognition and Computer Vision (volume 4)*, chapter 3, pages 181–199. World Scientific Publishing, January 2010.

[14] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, August 1996.

[15] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, January 2001.

[16] A. Cavallaro and T. Ebrahimi. Video object extraction based on adaptive background and statistical change detection. In *Visual Communications and Image Processing*, volume 4310 of *Proceedings of SPIE*, pages 465–475. SPIE, January 2001.

[17] V. Cevher, A. Sankaranarayanan, M. Duarte, D. Reddy, R. Baraniuk, and R. Chellappa. Compressive sensing for background subtraction. In *European Conference on Computer Vision*, volume 5303 of *Lecture Notes in Computer Science*, pages 155–168. Springer, October 2008.

[18] J. Cezar, C. Rosito, and S. Musse. A background subtraction model adapted to illumination changes. In *IEEE International Conference on Image Processing (ICIP)*, pages 1817–1820, October 2006.

[19] C. Chen, J. Liang, H. Zhao, H. Hu, and J. Tian. Frame difference energy image for gait recognition with incomplete silhouettes. *Pattern Recognition Letters*, 30(11):977–984, August 2009.

[20] M. Cheng, M. Ho, and C. Huang. Gait analysis for human identification through manifold learning and hmm. *Pattern Recognition*, 41(8):2541–2553, August 2008.

[21] C.-C. Chiu, M.-Y. Ku, and L.-W. Liang. A robust object segmentation system using a probability-based background extraction algorithm. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(4):518–528, April 2010.

[22] S. Cohen. Background estimation as a labeling problem. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1034–1041, Beijing, China, October 2005.

[23] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, September 2004.

[24] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, October 2003.

[25] D. Cunado, M. Nixon, and J. Carter. Automatic gait recognition via model-based evidence gathering. In *Proceedings of IEEE Workshop on Automated ID Technologies (AutoID)*, pages 27–30, Morristown, USA, November 1999.

[26] N. Dalal and B. Triggs. Hog, histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, San Diego, USA, June 2005.

[27] J. Davis and V. Sharma. Robust background-subtraction for person detection in thermal imagery. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 8, page 128, Washington, USA, June 2004.

[28] J. Davis and V. Sharma. Background-subtraction in thermal imagery using contour saliency. *International Journal of Computer Vision*, 71(2):161–181, February 2007.

[29] M. Dikmen and T. Huang. Robust estimation of foreground in surveillance videos by sparse error estimation. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 1–4, Tampa, USA, December 2008.

[30] A. El Maadi and X. Maldague. Outdoor infrared video surveillance: A novel dynamic technique for the subtraction of a changing background of IR images. *Infrared Physics & Technology*, 49(3):261–265, January 2007.

[31] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, July 2002.

[32] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision-Part II*, volume 1843 of *Lecture Notes in Computer Science*, pages 751–767, London, UK, June-July 2000. Springer.

[33] S. Elhabian, K. El-Sayed, and S. Ahmed. Moving object detection in spatial domain using background removal techniques – State-of-art. *Recent Patents on Computer Science*, 1:32–54, January 2008.

[34] J. Foster, M. Nixon, and A. Prügel-Bennett. Automatic gait recognition using area-based metrics. *Pattern Recognition Letters*, 24(14):2489–2497, October 2003.

[35] D. Gafurov, E. Snekkenes, and P. Bours. Spoof attacks on gait authentication system. *IEEE Transactions on Information Forensics and Security*, 2(3):491–502, September 2007.

[36] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, April 2006.

[37] W. Grimson. Gait analysis for recognition and classification. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, pages 155–161, Washington, USA, 2002.

[38] R. Gross and J. Shi. The CMU motion of body (MoBo) database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Pittsburgh, PA, June 2001.

[39] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, and A. Jain. A background model initialization algorithm for video surveillance. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 733–740, Vancouver, Canada, July 2001.

[40] H. Hadwiger. *Vorlesungen über inhalt, oberfläche and isoperimetric*. Springer Verlag, 1957.

[41] J. Han and B. Bhanu. Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):316–322, February 2006.

[42] I. Haritaoglu, D. Harwood, and L. Davis. W^4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, August 2000.

[43] J.-S. Hu and T.-M. Su. Robust background subtraction with shadow and highlight removal for indoor surveillance. *EURASIP Journal on Applied Signal Processing*, 2007(1):108–108, January 2007.

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

[45] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(3):334–352, August 2004.

[46] X. Huang and N. Boulgouris. Human gait recognition based on multiview gait sequences. *EURASIP Journal on Advances in Signal Processing*, 2008:8 pages, January 2008.

[47] D. Ioannidis, D. Tzovaras, I.G. Damousis, S. Argyropoulos, and K. Moustakas. Gait recognition using compact feature extraction transforms and depth information. *IEEE Transactions on Information Forensics and Security*, 2(3):623–630, September 2007.

[48] J. Jacques, C. Jung, and S. Musse. Background subtraction and shadow detection in grayscale video sequences. In *Brazilian Symposium on Computer Graphics and Image Processing*, pages 189–196, Natal, Brazil, October 2005.

[49] S. Jodogne, C. Briquet, and J. Piater. Approximate policy iteration for closed-loop learning of visual tasks. In *Proceedings of the 17th European Conference on Machine Learning (ECML)*, volume 4212 of *Lecture Notes in Computer Science*, pages 222–233. Springer Verlag, September 2006.

[50] P.-M. Jodoin, M. Mignotte, and J. Konrad. Statistical background subtraction using spatial cues. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(12):1758–1763, December 2007.

[51] C. Jung. Efficient background subtraction and shadow removal for monochromatic video sequences. *IEEE Transactions on Multimedia*, 11(3):571–577, April 2009.

[52] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *European Workshop on Advanced Video Based Surveillance Systems*, London, UK, September 2001.

[53] A. Kale, N. Cuntoor, B. Yegnanarayana, A. Rajagopalan, and R. Chellappa. Gait analysis for human identification. In *Proceedings of the International Conference on Audio-and Video-Based Person Authentication*, pages 706–714, Guildford, UK, 2003.

[54] A. Kale, A. Sundaresan, A. Rajagopalan, N. Cuntoor, A. Roy-Chowdhury, V. Kruger, and R. Chellappa. Identification of humans using gait. *IEEE Transactions on Image Processing*, 13(9):1163–1173, September 2004.

[55] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *IEEE International Conference on Image Processing (ICIP)*, volume 5, pages 3061–3064, Singapore, October 2004.

[56] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, June 2005. Special Issue on Video Object Processing.

[57] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *European Conference on Computer Vision (ECCV)*, pages 189–196, Stockholm, Sweden, May 1994. Springer-Verlag.

[58] L. Lacassagne, A. Manzanera, J. Denoulet, and A. Mérigot. High performance motion detection: some trends toward new embedded architectures for vision systems. *Journal of Real-Time Image Processing*, 4(2):127–146, June 2009.

[59] L. Lacassagne, A. Manzanera, and A. Dupret. Motion detection: Fast and robust algorithms for embedded systems. In *IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, November 2009.

[60] T. Lam, R. Lee, and D. Zhang. Human gait recognition by the fusion of motion and static spatio-temporal templates. *Pattern Recognition*, 40(9):2563–2573, September 2007.

[61] D. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):827–832, May 2005.

[62] S. Lee, Y. Liu, and R. Collins. Shape variation-based frieze pattern for robust gait recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.

[63] B. Lei and L. Xu. Real-time outdoor video surveillance with robust foreground extraction and object tracking via multi-state transition management. *Pattern Recognition Letters*, 27(15):1816–1825, November 2006.

[64] L. Li, W. Huang, I. Gu, and Q. Tian. Foreground object detection from videos containing complex background. In *ACM International Conference on Multimedia*, pages 2–10, Berkeley, USA, November 2003. ACM.

[65] H.-H. Lin, T.-L. Liu, and J.-C. Chuang. Learning a scene background model via classification. *IEEE Signal Processing Magazine*, 57(5):1641–1654, May 2009.

[66] Y. Liu, R. Collins, and Y. Tsin. Gait sequence analysis using frieze patterns. In *Proceedings of the 7th European Conference on Computer Vision - Part II*, pages 657–671, London, UK, 2002. Springer-Verlag.

[67] Z. Liu and S. Sarkar. Improved gait recognition by gait dynamics normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):863–876, June 2006.

[68] Z. Liu and S. Sarkar. Outdoor recognition at a distance by fusing gait and face. *Image Vision Computing*, 25(6):817–832, June 2007.

[69] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[70] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the international joint conference on Artificial intelligence (IJCAI)*, pages 674–679, Vancouver, Canada, April 1981.

[71] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, July 2008.

[72] V. Mahadevan and N. Vasconcelos. Spatiotemporal saliency in dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):171–177, January 2010.

[73] A. Manzanera. Σ-Δ background subtraction and the Zipf law. In *Progress in Pattern Recognition, Image Analysis and Applications*, volume 4756 of *Lecture Notes in Computer Science*, pages 42–51. Springer, November 2007.

[74] A. Manzanera and J. Richefeu. A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 46–51, Kolkata, India, December 2004.

[75] A. Manzanera and J. Richefeu. A new motion detection algorithm based on Σ-Δ background estimation. *Pattern Recognition Letters*, 28(3):320–328, February 2007.

[76] P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):701–716, July 1989.

[77] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 34–40, San Diego, USA, June 2005.

[78] G. Matheron. *Eléments pour une théorie des milieux poreux*. Masson, Paris, 1967.

[79] T. Mathes and J. Piater. Robust non-rigid object tracking using point distribution models. In *Proceedings of the British Machine Vision Conference*, pages 849–858, Oxford, UK, September 2005.

[80] A. McIvor. Background subtraction techniques. In *Proc. of Image and Vision Computing*, Auckland, New Zealand, November 2000.

[81] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 257–263, Madison, USA, June 2003.

[82] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 302–309, Los Alamitos, USA, June-July 2004.

[83] E. Monari and C. Pasqual. Fusion of background estimation approaches for motion detection in non-static backgrounds. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 347–352, London, UK, September 2007.

[84] S. Mowbray and M. Nixon. Automatic gait recognition via fourier descriptors of deformable objects. In J. Kittler and M. Nixon, editors, *Audio Visual Biometric Person Authentication*, pages 566–573, Guildford, UK, June 2003. Springer.

[85] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical report, Columbia University, February 1996.

[86] M. Nixon. Gait biometrics. *Biometric Technology Today*, 16(7-8):8–9, July 2008.

[87] M. Nixon, J. Carter, J. Shutler, and M. Grant. New advances in automatic gait recognition. *Elsevier Information Security Technical Report*, 7(4):23–35, 2002.

[88] M. Nixon, T. Tan, and R. Chellappa. *Human identification based on gait*. Springer, 2006.

[89] S. Niyogi and E. Adelson. Analyzing and recognizing walking figures in xyt. In IEEE Computer Society, editor, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 469–474, Seattle (Washington, USA), June 1994.

[90] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, August 2000.

[91] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In IEEE, editor, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 1997.

[92] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, June 2000.

[93] A. Papoulis. *Probability, random variables, and stochastic processes*. McGraw-Hill, 1984.

[94] D. Parks and S. Fels. Evaluation of background subtraction algorithms with post-processing. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 192–199, Santa Fe (New Mexico, USA), September 2008.

[95] M. Piccardi. Background subtraction techniques: a review. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 4, The Hague, The Netherlands, October 2004.

[96] S. Piérard, O. Barnich, and M. Van Droogenbroeck. A virtual curtain for the detection of humans and access control. In *Advanced Concepts for Intelligent Vision Systems (ACIVS 2010)*, page 12, Sydney, Australia, December 2010.

[97] P. Power and J. Schoonees. Understanding background mixture models for foreground segmentation. In *Proceedings of Image and Vision Computing*, pages 267–271, Auckland, New Zealand, November 2002.

[98] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, March 2005.

[99] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997.

[100] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[101] M. Seki, T. Wada, H. Fujiwara, and K. Sumi. Background subtraction based on cooccurrence of image variations. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 65–72, Los Alamitos, USA, June 2003.

[102] J. Serra. *Image analysis and mathematical morphology*. Academic Press, New York, 1982.

[103] C. Shan, S. Gong, and P. McOwan. Fusing gait and face cues for human gender recognition. *Neurocomputing*, 71(10-12):1931–1938, June 2008.

[104] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1778–1792, November 2005.

[105] B. Shoushtarian and H. Bez. A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking. *Pattern Recognition Letters*, 26(1):5–26, January 2005.

[106] M. Sivabalakrishnan and D. Manjula. An efficient foreground detection algorithm for visual surveillance system. *International Journal of Computer Science and Network Security*, 9(5):221–227, May 2009.

[107] M. Soriano, A. Araullo, and C. Saloma. Curve spreads: a biometric from front-view gait video. *Pattern Recognition Letters*, 25(14):1595–1602, 2004.

[108] A. Srivastava, A. Lee, E. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33, January 2003.

[109] C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 246–252, Ft. Collins, USA, June 1999.

[110] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[111] D. Tao, X. Li, X. Wu, and S.J. Maybank. General tensor discriminant analysis and gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1700–1715, October 2007.

[112] A. Tavakkoli, M. Nicolescu, G. Bebis, and M. Nicolescu. Non-parametric statistical background modeling for efficient foreground region detection. *Machine Vision and Applications*, 20(6):395–409, October 2008.

[113] K. Toyama, J. Krumm, B. Brumitt, and M. Meyers. Wallflower: Principles and practice of background maintenance. In *International Conference on Computer Vision (ICCV)*, pages 255–261, Kerkyra, Greece, September 1999.

[114] D.-M. Tsai and S.-C. Lai. Independent component analysis-based background subtraction for indoor surveillance. *IEEE Transactions on Image Processing*, 18(1):158–167, January 2009.

[115] M. Van Droogenbroeck. Algorithms for openings of binary and label images with rectangular structuring elements. In H. Talbot and R. Beare, editors, *Mathematical morphology*, pages 197–207. CSIRO Publishing, Sydney, Australia, April 2002.

[116] M. Van Droogenbroeck and O. Barnich. Visual background extractor. World Intellectual Property Organization, WO 2009/007198, 36 pages, January 2009.

[117] P. Varcheie, M. Sills-Lavoie, and G.-A. Bilodeau. A multiscale region-based motion detection and background subtraction algorithm. *Sensors*, 10(2):1041–1061, January 2010.

[118] A. Veeraraghavan, A.K. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909, December 2005.

[119] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[120] H. Wang and D. Suter. Background subtraction based on a robust consensus method. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 223–226, Washington, USA, August 2006.

[121] H. Wang and D. Suter. A consensus-based method for tracking: Modelling background scenario and foreground appearance. *Pattern Recognition*, 40(3):1091–1105, March 2007.

[122] L. Wang, T. Tan, W. Hu, and H. Ning. Automatic gait recognition based on statistical shape analysis. *IEEE Transactions on Image Processing*, 12(9):1120–1131, September 2003.

[123] L. Wang, T. Tan, H. Ning, and W. Hu. Silhouette analysis-based gait recognition for human identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1505–1518, December 2003.

[124] Y. Wang, K. Loe, and J. Wu. A dynamic conditional random field model for foreground and shadow segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):279–289, February 2006.

[125] Y. Wang, T. Tan, K. Loe, and J. Wu. A probabilistic approach for foreground and shadow segmentation in monocular image sequences. *Pattern Recognition*, 38(11):1937–1946, November 2005.

[126] B. White and M. Shah. Automatically tuning background subtraction parameters using particle swarm optimization. In *IEEE International Conference on Multimedia and Expo*, pages 1826–1829, Beijing, China, July 2007.

[127] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[128] M. Wu and X. Peng. Spatio-temporal context for codebook-based dynamic background subtraction. *International Journal of Electronics and Communications*, In Press, 2009.

[129] D. Xu, S. Yan, D. Tao, S. Lin, and H. Zhang. Marginal fisher analysis and its variants for human gait recognition and content- based image retrieval. *IEEE Transactions on Image Processing*, 16(11):2811–2821, November 2007.

[130] C. Yam, M. Nixon, and J. Carter. Automated person recognition by walking and running via model-based approaches. *Pattern Recognition Letters*, 37(5):1057–1072, May 2004.

[131] X. Yang, Y. Zhou, T. Zhang, G. Shu, and J. Yang. Gait recognition based on dynamic region analysis. *Signal Processing*, 88(9):2350–2356, September 2008.

[132] Q. Zang and R. Klette. Robust background subtraction and maintenance. In *IEEE International Conference on Pattern Recognition (ICPR)*, volume 2, pages 90–93, Washington, USA, August 2004.

[133] S. Zhou, R. Chellappa, and W. Zhao. *Unconstrained Face Recognition*. Springer, 2006.

[134] X. Zhou and B. Bhanu. Feature fusion of side face and gait for video-based human identification. *Pattern Recognition*, 41(3):778–795, March 2008.

[135] Z. Zhou, A. Prugel-Bennett, and R.I. Damper. A bayesian framework for extracting human gait using strong prior knowledge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1738–1752, November 2006.

[136] Z. Zivkovic. Improved adaptive gausian mixture model for background subtraction. In *IEEE International Conference on Pattern Recognition (ICPR)*, volume 2, pages 28–31, Cambridge, UK, August 2004.

[137] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, May 2006.