

Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art

Shireen Y. Elhabian*, Khaled M. El-Sayed* and Sumaya H. Ahmed*

Information Technology Department, Faculty of Computers and Information, Cairo University, 5 Dr. Ahmed Zweel Street, Doki, Giza, 12613, Egypt

Received: July 6, 2007; Accepted: August 28, 2007; Revised: September 15, 2007

Abstract: Identifying moving objects is a critical task for many computer vision applications; it provides a classification of the pixels into either foreground or background. A common approach used to achieve such classification is background removal. Even though there exist numerous of background removal algorithms in the literature, most of them follow a simple flow diagram, passing through four major steps, which are pre-processing, background modelling, foreground detection and data validation. In this paper, we survey many existing schemes in the literature of background removal, surveying the common pre-processing algorithms used in different situations, presenting different background models, and the most commonly used ways to update such models and how they can be initialized. We also survey how to measure the performance of any moving object detection algorithm, whether the ground truth data is available or not, presenting performance metrics commonly used in both cases.

Key Words: Background modelling, foreground detection, performance evaluation, background subtraction, moving object detection.

INTRODUCTION

Computer vision systems have been developed to simulate most biological systems which have the ability to cope up with changing environments such as moving objects, changing illumination and changing viewpoints. Detection and tracking of moving objects can be viewed as lower level vision tasks to achieve higher level event understanding. Identifying moving objects is a critical task for video segmentation, which is used in many computer vision applications such as remote sensing, video surveillance and traffic monitoring.

Moving object detection provides a classification of the pixels in the video sequence into either foreground (moving objects) or background. A common approach used to achieve such classification is background removal, sometimes referred to as background subtraction, where each video frame is compared against a reference or background model, pixels that deviate significantly from the background are considered to be moving objects.

The general requirements for a background removal algorithm are the accuracy in object contour detection (spatial accuracy) and temporal stability of the detection (temporal coherency). Moreover, the ability to detect changes of small magnitude (sensitivity) and providing good accuracy under varying conditions such as illumination changes (robustness).

Despite of its importance, moving object detection in complex environments is still far from being completely

solved. As noted by Toyama *et al.* [1], Elgammal *et al.* [2] and Harville *et al.* [3], there are several problems that must be addressed by a good background removal algorithm to correctly detect moving objects. A good background removal algorithm should handle the relocation of background objects, non-stationary background objects e.g. waving trees, and image changes due to camera motion which is common in outdoor applications e.g. wind load. A background removal system should adapt to illumination changes whether gradual changes (time of day) or sudden changes (light switch), whether global or local changes such as shadows and inter-reflections. A foreground object might have similar characteristics as the background, it become difficult to distinguish between them (camouflage). A foreground object that becomes motionless can not be distinguished from a background object that moves and then becomes motionless (sleeping person). A common problem faced in the background initialization phase is the existence of foreground objects in the training period, which occlude the actual background, and on the other hand often it is impossible to clear an area to get a clear view of the background, this puts serious limitations on system to be used in high traffic areas. Some of these problems can be handled by very computationally expensive methods, but in many applications, a short processing time is required.

Even though there exist numerous of background removal algorithms in the literature, most of them follow a simple flow diagram, defined by Cheung and Kamath [4], passing through four major steps, which are (1) pre-processing (simple image processing tasks that change the raw input video into a format that can be processed by subsequent steps), (2) background modelling (also known as background maintenance), (3) foreground detection (also known as background subtraction) and (4) data validation (also referred to as post-processing, used to eliminate those

*Address correspondence to these authors at the Computer Vision and Image Processing Laboratory, Lutz Hall, Univ. of Louisville, Louisville, KY, 40292; Tel: (+2010) 5376133 - (502) 417-9445; E-mails: s.elhabian@fci-cu.edu.eg; syelha01@louisville.edu; s.ahmed@fci-cu.edu.eg

pixels that do not correspond to actual moving objects). Although the terms background subtraction and background modelling are often used interchangeably, they are separate and distinct processes. Background modelling refers to the process of creating, and subsequently maintaining, a model of the appearance of the background in the field of view of a camera. Background subtraction refers to the process in which an image frame is compared to the background model in order to determine whether individual pixels are part of the background or the foreground. So it is also referred to as foreground detection.

In this paper, we focus on the problem of moving object detection using background removal techniques. We survey many existing schemes in the literature from the viewpoint of the generic flow diagram introduced by Cheung and Kamath [4]. Section 2 discusses the pre-processing phase, surveying the common algorithms used in different situations. Section 3 discusses the background modelling phase, presenting different background models used to represent the background, the most commonly used ways to update such models and how those models can be initialized. Foreground detection and data validation are presented in section 4 and 5 respectively. While section 6 is dedicated to how to measure the performance of any moving object detection algorithm, whether the ground truth data is available or not, presenting performance metrics commonly used in both cases.

2. PRE-PROCESSING

The goal of a moving object detection algorithm is to detect significant changes occurring throughout the video sequence while rejecting unimportant ones. The following describes pre-processing steps used to filter out common types of unimportant changes before making the object detection decision. These steps generally involve geometric and radiometric (i.e. intensity) adjustments [5]. Others involve using image derivatives or depth information as an information source to the moving object detection algorithm. For real-time systems, frame-size and frame-rate reduction are commonly used to reduce the data processing rate. Simple temporal and/or spatial smoothing is often used in the early stage of pre-processing to reduce camera noise and to remove transient environmental noise such as rain and snow captured in outdoor applications.

2.1. Geometric Adjustments

Apparent intensity changes at a pixel resulting from camera motion alone are virtually never desired to be detected as real changes. A method of processing video images includes capturing a first image with a camera having a first field of view. The capturing occurs at a first point in time. Commands are transmitted to the camera to make pan, tilt and zoom movements. A second image is captured with the camera at a second point in time. The second point in time is after the movements have commenced. A second field of view of the camera is calculated at the second point in time. The calculating is based upon the pan, tilt and zoom commands. The second image is processed based upon the first field of view and the calculated second field of view. The field of view of the camera may be calculated as a function of time. The calculated field of view may be output with a qualification based upon a point in time associated with the

calculated field of view. The processing may comprise determining a mask location or tracking an object of interest [6] [p1]. Hence, *frame registration* is used to align several frames into the same coordinate frame. When the scenes of interest are mostly rigid in nature and the camera motion is small, registration can often be performed using low-dimensional spatial transformations such as similarity, affine, or projective transformations. Excellent surveys [7-10], and software implementations (e.g., the Insight toolkit [11]) are available. Stewart *et al.* [12] switch automatically to higher-order transformations after being initialized with a low-order similarity transformation. In some scenarios a non-global transformation may need to be estimated to determine corresponding points between two frames, e.g. *via* optical flow [13], active contour tracking [14], object recognition and pose estimation [15, 16], or structure-from-motion [17, 18] algorithms.

2.2. Radiometric/Intensity Adjustments

There are several techniques that attempt to pre-compensate for illumination variations between frames caused by changes in the strength or position of light sources in the scene. Some of the earliest attempts to deal with illumination changes used intensity normalization [19-21], i.e. normalizing the pixel intensity values to have the same mean and variance as those in the estimated background. Alternatively, both, the current frame and the background can be normalized to have zero mean and unit variance. This allows the use of decision thresholds that are independent of the original intensity values [5]. Instead of using global statistics, the frames can be divided into corresponding disjoint blocks, and the normalization independently performed using the local statistics of each block. This can achieve better local performance at the expense of introducing blocking artifacts [5]. However, algorithms which generally employ normalized colours typically work poorly in dark areas of the image [22]. For scenes containing Lambertian surfaces, it is possible to extract the reflectance component (albedo image which contains mainly physical object information) by applying homomorphic filter [23] to the input intensities. The reflectance component can be provided as input to the decision rule step of a moving object detection process (see, e.g. [24, 25]). Modelling and compensating for local radiometric variation that deviates from the Lambertian assumption is necessary in several applications (e.g. underwater imagery [26]). Can and Singh [27] modelled the illumination component as a piecewise polynomial function. Hager and Belhumeur [28] used principal component analysis (PCA) to extract a set of basis images that represent the views of a scene under all possible lighting conditions. According to Radke *et al.* [5], these sophisticated models of illumination compensation are not commonly used in the context of foreground detection.

Pre-processing may include feature extraction, i.e. transforming the input frame into the most appropriate feature space derived from the current frame only, i.e. it doesn't contain motion information. The feature space may represent the data format used by a particular background removal algorithm. Most of the algorithms handle luminance intensity, which is one scalar value per each pixel [29-35]. However, colour image is becoming more popular in the back-

ground removal literature [17, 36-44]. These papers argue that colour is better than luminance at identifying objects in low-contrast areas and suppressing shadow cast by moving objects [4]. The colour space used is generally the RGB space, since RGB values are readily provided by most frame grabbers. However, it is not well behaved with respect to colour perception, as a distance computed between two colours in RGB space does not reflect their perceptual similarity. Pfister [17] uses the YUV colour space, which separates intensity (Y) and chromaticity (U,V) in the pixel measurement. Similarly, the HSV model separates the intensity (V) from the chromatic components (H,S). However, the UV subspace representation of chromaticity, based on linear combinations of R, G and B channels, is not as intuitive as the radial HS subspace representation [45]. Elgammal *et al.* [37] use the chromaticity coordinates as $r = R/s$, $g = G/s$ and $b = B/s$ where $s = (R+G+B)$ and $r+g+b=1$. This has the advantage of being more insensitive to small changes in illumination that arise due to shadows. However, they have the disadvantage of losing lightness information which is related to the differences in whiteness, blackness, and grey-ness between different objects [46]. To address this problem, Elgammal *et al.* [2] use s a measure of lightness at each pixel.

2.3. Image Derivatives

Pixel-based image features such as spatial and temporal derivatives are sometimes used to incorporate edges and motion information, to have a representation of the scene background that is invariant to illumination changes. For example, intensity values and spatial derivatives can be combined to form a single state space for background tracking with the Kalman filter [47]. Pless *et al.* [48] combine both spatial and temporal derivatives to form a constant velocity background model for detecting speeding vehicles. Level lines can also be employed [49] exploiting the fact that a global illumination variations changes the number, but not the geometry of the level lines which are the boundary of the level sets extracted from each frame in the sequence in hand. The use of edges provides good spatial accuracy [50]. In addition, since edge maps are bi-level (binary) images, they are convenient from computation and storage viewpoints. However, the main drawback of adding colour or derived features in background modelling is the extra complexity for model parameter estimation. The increase in complexity is often significant as most background modelling techniques maintain an independent model for each pixel (pixel-based modelling).

2.4. Depth Information

Using depth information presents fewer problems for segmentation, since depth information from stereo video is relatively unaffected by lighting conditions or extraneous motion. However, depth data does not produce valid results in scene locations with little visual texture, low contrast regions or that is not visible to all cameras, and tends to be noisy even where it is available. Hence, background removal methods based on depth alone [51,52] produce unreliable answers in substantial portions of the scene, and often fail to find foreground objects in close proximity to the background, such as hands placed on walls or feet on a floor. A method using both depth and colour has been proposed via

Gordon *et al.* [53], but it lacks time adaptivity and uses a relatively simple statistical model of the background. A significant advantage of the use of colour and depth space in the background estimation process is that, at pixels for which depth is usually valid, we can correctly estimate depth and colour of the background when the background is represented in only a minority of the frames. For pixels which have significant invalid range, we fall back to the same majority requirement as colour-only methods [53]. In the general case where depth measurements at the pixel are largely valid, the background is simply represented by the mode which is farthest in depth and covers at least T% of the data temporally. Eveland *et al.* [51] work with disparities rather than depth because the error statistics are constant over the range of disparities. A disparity image can be extracted from each stereo pair, using the area correlation method described in [54].

3. BACKGROUND MODELLING

Background modelling, also referred to as background maintenance, is at the heart of any background removal algorithm. Toyama *et al.* [1] propose a set of principles to which background modelling modules should adhere. The module performing background modelling should not attempt to extract the semantics of foreground objects on its own, since it is not an end by itself, larger systems use it as a component. One can evaluate background modelling by how closely it comes to finding all foreground pixels (as defined by the end task) when a foreground object *first* appears in the scene, while simultaneously ignoring all others. Backgrounds are not necessarily defined by absence of motion, e.g. waving trees. No unimodal distribution of pixel values can adequately capture such a background, because these models implicitly assume that the background is static. An appropriate pixel-level stationarity criterion should be defined. Pixels that satisfy this criterion are declared background and ignored. The background model must adapt to both sudden and gradual changes in the background. Toyama [1] suggests that fast adaptation works quite well if the foreground consists of moving people. Some parts of a scene may remain in the foreground unnecessarily long if adaptation is slow, but other parts will disappear too rapidly into the background if adaptation is fast. Neither approach is inherently better than the other - a point that emphasizes the inadequacy of background modelling for all but the initialization of tracking. Most background modelling techniques operate at the pixel-level. Background models should take into account changes at differing spatial scales which are necessary to solve many of background modelling problems. Toyama *et al.* [1] and Javed *et al.* [55] process images at the pixel, region, and frame levels, where the global illumination changes can be handled in the frame level.

According to Cristani *et al.* [56], the issues characterizing a background modelling process are usually three; model representation, model initialization, and model adaptation. The first describes the kind of model used to represent the background; the second one regards the initialization of this model, and the third one relies to the mechanism used for adapting the model to the background changes (e.g. illumination changes).

3.1. Background Representation

The simplest background model is a known background. This occurs often in the entertainment or broadcast television industry in which the environment can be engineered to simplify background removal algorithms. This includes the use of “blue screens”, backdrops with a constant colour which are designed to be easy to segment. The measurement used is the colour of a given pixel. If the camera is fixed and the background can be expected to stay relatively constant, we can model the background as a single static image that may be easily identified and ignored. The required measurement in this case is the intensity in case of gray level images and the colour components in case of colour images. If the background is not actually constant, then modelling both the mean intensity at a pixel and its variance gives an adaptive tolerance for some variation in the background. If a scene contains motion that should be considered part of the background, more tolerant models are required. One solution is to model measurements with a single multivariate Gaussian distribution. The parameters of this model are the mean and covariance matrix. When a single Gaussian is insufficient to model the distribution of pixel values, a finite mixture of Gaussians (MOG) may be used instead. Having a mixture model containing k Gaussians for some $k \in \mathbb{N}$. The parameters of this model are then k mean values, k covariance matrices, and k scaling factors to weight the relevance importance of each Gaussian.

Elgammal *et al.* [37] estimate the density function of pixel's distribution at any moment of time given only very recent history information hoping to obtain sensitive detection. The measurement of this model is a recent sample of intensity values for a pixel. Using this sample, the probability density function that this pixel will have a certain intensity value at time t can be non-parametrically estimated [57] using the kernel estimator.

A particular distribution of spatio-temporal image derivatives arises at points which always follow a constant optic flow. In this case, the image derivatives should fit the optic flow constraint equation: $I_x u + I_y v + I_t = 0$, for an optic flow vector (u, v) which remains constant through time for background pixels. Although motion-based approaches allows predicting the motion pattern of each pixel, this approach might fail when there is no obvious difference between the motion fields of the foreground and background [58].

The fundamental background model used by Toyama *et al.* [1] is a one step Wiener filter which is linear predictor of the intensity at a pixel based upon the time history of intensity at that particular pixel. This can account for periodic variations of pixel intensity. The measurement includes two parts, the intensity at the current frame, and the recent time history of intensity values at a given pixel.

Hidden Markov Models can also be used to represent the pixel process where its states can represent different states that might occur in the pixel process, such as background, foreground, shadows, day and night illumination. It can also be used to handle the sudden changes in illumination where the change from a status to another, such as the change from dark to light, day to night, indoor to outdoor, can be repre-

sented as the transition from state to state in the HMM. Wang *et al.* [59] represent pixel process with HMM, where they use three-states HMM to represent background, shadows and foreground. They modelled both background and shadows as single Gaussian distribution.

The background can also be represented by a group of clusters which are ordered according to the likelihood that they model the background and are adapted to deal with background and lighting variations. Incoming pixels are matched against the corresponding cluster group and are classified according to whether the matching cluster is considered part of the background. Butler *et al.* [60] model each pixel by a group of K clusters where each cluster consists of a weight w_k and an average pixel value or centroid c_k . Kim *et al.* [61, 62] quantize sample background values at each pixel into codebooks which represent a compressed form of background model for a long image sequence. They adopt a quantization/clustering technique, inspired by Kohonen [63, 64]. Their method can handle scenes containing moving backgrounds or illumination variations, and it achieves robust detection for different types of videos. Mixed backgrounds can be modelled by multiple codewords. Unlike MOG, Kim *et al.* do not assume that backgrounds are multimode Gaussians. Also, in contrast to Kernel, Kim does not store raw samples to maintain the background model.

3.2. Background Adaptation

According to Mittal and Paragios [65], existing methods for background adaptation may be classified as either predictive or non-predictive. Predictive methods model the scene as a time series and develop a dynamical model to recover the current input based on past observations. The magnitude of the deviation between the predicted and actual observation can then be used as a measure of change, while non-predictive methods neglect the order of the input observations and build a probabilistic representation (pdf) of the observations at a particular pixel.

Cheung and Kamath [4] suggest another way of classification; they classify background adaptation techniques into two broad categories - non-recursive and recursive. A non-recursive technique uses a sliding-window approach for background estimation. It stores a buffer of the previous L video frames, and estimates the background image based on the temporal variation of each pixel within the buffer. Non-recursive techniques are highly adaptive as they do not depend on the history beyond those frames stored in the buffer. On the other hand, the storage requirement can be significant if a large buffer is needed to cope with slow-moving objects. Given a fixed-size buffer, this problem can be partially alleviated by storing the video frames at a lower frame-rate. Recursive techniques do not maintain a buffer for background estimation. Instead, they recursively update either a single or multiple background model(s) based on each input frame. As a result, input frames from distant past could have an effect on the current background model. Compared with non-recursive techniques, recursive techniques require less storage, but any error in the background model can linger for a much longer period of time. Most schemes include exponential weighting to discount the past, and incorporate positive decision feedback to use only background pixels for updating.

3.2.1. Non-recursive Techniques

Frame differencing, also known as temporal difference, uses the video frame at time $t-1$ as the background model for the frame at time t . This technique is sensitive to noise and variations in illumination, and does not consider local consistency properties of the change mask [5]. This method also fails to segment the non-background objects if they stop moving [45, 66]. Since it uses only a single previous frame, frame differencing may not be able to identify the interior pixels of a large, uniformly-colored moving object. This is commonly known as the aperture problem.

Average filter averages the images over time, creating a background approximation which is similar to the current static scene except where motion occurs. However, this is not robust to scenes with many moving objects particularly if they move slowly. It also cannot handle bimodal backgrounds, recovers slowly when the background is uncovered, and has a single, predetermined threshold for the entire scene [67]. Koller *et al.* [68] handle the change of lighting condition using a moving-window average method, where exponential forgetting is used. An obvious problem with this technique is that all information coming from both background and foreground is used to update the background model. If some objects move slowly, these algorithms will fail. The solution to this problem is that only those pixels not identified as moving objects are used to update background model.

Median filtering defines the background to be the median at each pixel location of all the frames in the buffer [36, 42, 69-71]. It assumes that the pixel stays in the background for more than half of the frames in the buffer. Median filtering has been extended to colour by replacing the median with the medoid [42]. The complexity of computing the median is $O(L \log L)$ for each pixel.

Minimum-Maximum filter, three values are estimated for each pixel using the training sequence without foreground objects: minimum intensity (Min), maximum intensity (Max), and the maximum intensity difference between consecutive frames (D) [39]. These values are estimated over several frames and are periodically updated for background regions. Boulton *et al.* [72] uses two gray level background images B_1 , B_2 to cope with intensity variations due to noise or fluttering objects, moving in the scene.

Linear predictive filter, Toyama *et al.* [1] compute the current background estimate by applying a linear predictive filter on the pixels in the buffer using a Wiener filter to predict a pixel's current value from a linear combination of its k previous values. Pixels whose prediction error is several times worse than the expected error are classified as foreground pixels. The filter coefficients are estimated at each frame time based on the sample covariance, making this technique difficult to apply in real-time. Linear prediction using the Kalman filter was also used in [68, 73, 74]. Monnet *et al.* [75] present an autoregressive form to predict the frame to be observed. Two different techniques are studied to maintain the model, one that update the states in an incremental manner and one that replaces the modes of variation using the latest observation map. Other techniques can be considered to determine such prediction model. Principal

component analysis [76, 77] refers to a linear transformation of variables that retains - for a given number n of operators - the largest amount of variation within the training data. Estimation of the basis vectors from the observed data set can be performed through singular value decomposition. Computing the basis components for large vectors is a time consuming operation. Optimal algorithms for singular value decomposition of an $m \times n$ matrix take $O(m^2n + n^3)$ time [78]. A simple way to deal with such complexity is by considering the process at a block level. To this end, Monnet *et al.* [75] divide the image into blocks and run the algorithm independently on each block.

Non-parametric modelling models the pixel as a random variable in a feature space with an associated probability density function (pdf). Nonparametric approaches estimate the density function directly from the data without any assumptions about the underlying distribution, avoiding having to choose a model and estimating its distribution parameters. Kernel density estimators asymptotically converge to any density function [79, 80]. In fact, all other nonparametric density estimation methods, e.g., histograms, can be shown to be asymptotically kernel methods [79]. However, the major drawback with colour histograms is the lack of convergence to the right density function if the data set is small; also they are not suitable for higher dimensional features [80]. Unlike histograms, even with a small number of samples, kernel density estimation leads to a smooth, continuous and differentiable density estimate. Kernel density estimation does not assume any specific underlying distribution and, theoretically, the estimate can converge to any density shape with enough samples [57]. Therefore, this approach is suitable to model the colour distribution of regions with patterns and mixture of colours. Unlike parametric fitting of a mixture of Gaussians, kernel density estimation is a more general approach that does not require the selection of the number of Gaussians to be fitted; also the adaptation of the model is trivial and can be achieved by adding new samples. The advantage of using the full density function over a single estimate is the ability to handle multi-modal background distribution, i.e. pixels from a swinging tree or near high-contrast edges where they flicker under small camera movement.

One major issue that needs to be addressed when using kernel density estimation technique is the choice of suitable kernel bandwidth (scale). Theoretically, as the number of samples reaches infinity, the choice of the bandwidth is insignificant and the estimate will approach the actual density. Practically, since only a finite number of samples are used and the computation must be performed in real time, the choice of suitable bandwidth is essential. Too small a bandwidth will lead to a ragged density estimate, while too wide a bandwidth will lead to an over-smoothed density estimate [79]. Since the expected variations in pixel intensity over time are different from one location to another in the image, a different kernel bandwidth is used for each pixel. Also, a different kernel bandwidth is used for each colour channel. Elgammal *et al.* [37] use the median of the absolute differences between successive frames as the width of the kernel. Thus, the complexity of building the model is the same as median filtering.

A variety of kernel functions with different properties have been used in the literature. Typically the Gaussian kernel is used for its continuity, differentiability, and locality properties. Note that choosing the Gaussian as a kernel function is different from fitting the distribution to a Gaussian model. Here, the Gaussian is only used as a function to weight the data points. A good discussion of kernel estimation techniques can be found in [79].

The major drawback of using the nonparametric kernel density estimator is its computational cost. This becomes less of a problem as the available computational power increases and as efficient computational methods have become available recently [80, 81]. However, several pre-calculated lookup tables for the kernel function values can be used to reduce the burden of computation of this approach. Also, this method can not resist the influence of foreground objects in the training stage (background initialization) [82]. In general, given N original data samples and M target points at which the density need to be evaluated, the complexity is $O(NM)$ evaluations of the kernel function, multiplications and additions [80, 83]. Elgammal *et al.* [80, 83] present a computational framework for efficient density estimation, introducing the use of Fast Gauss Transform (FGT) for efficient computation of colour densities, allowing the summation of a mixture of M Gaussians at N evaluation points in $O(M+N)$ time as opposed to $O(MN)$ time for a naive evaluation, and can be used to considerably speed up kernel density estimation.

Given a new pixel sample, according to Elgammal *et al.* [37], there are two alternative mechanisms to update the background; *selective update*: add the new sample to the model only if it is classified as a background sample and *blind update*: just add the new sample to the model. There are tradeoffs to these two approaches. The first enhance detection of the targets, since target pixels are not added to the model, however, any incorrect detection decision will result in persistent incorrect detection later, which is a deadlock situation. The second approach does not suffer from this deadlock situation since it does not involve any update decisions allowing intensity values that do not belong to the background to be added to the model. This leads to bad detection of the targets (more false negatives) as they erroneously become part of the model. This effect is reduced as we increase the time window over which the sample are taken [37], as a smaller proportion of target pixels will be included in the sample. But as we increase the time window more false positives will occur because the adaptation to changes is slower and rare events are not as well represented in the sample. Elgammal [37] presents a way to combine the results of two background models (a long term and a short term) in such a way to achieve better update decisions and avoid the tradeoffs discussed above. *Short-term model* is a very recent model of the scene. It adapts to changes quickly to allow very sensitive detection. The sample is updated using a selective-update mechanism, where the update decision is based on the final result of combining the two models. *Long-term model* captures a more stable representation of the scene background and adapts to changes slowly. The sample is updated using a blind-update mechanism. Computing the intersection of the two detection results will eliminate the persistence false positives from the short term model and

will eliminate as well extra false positives that occur in the long term model results. The only false positives that will remain will be rare events not represented in either model. If this rare event persists over time in the scene then the long term model will adapt to it, and it will be suppressed from the result later. Taking the intersection will, unfortunately, suppress true positives in the first model result that are false negatives in the second, because the long term model adapts to targets as well if they are stationary or moving slowly. To address this problem, all pixels detected by the short term model that are adjacent to pixels detected by the combination are included in the final result.

3.2.2. Recursive Techniques

Approximated Median Filter

Due to the success of non-recursive median filtering, McFarlane and Schofield [84] propose a simple recursive filter to estimate the median. This technique has also been used in background modelling for urban traffic monitoring [85] where the running estimate of the median is incremented by one if the input pixel is larger than the estimate, and decreased by one if smaller. This estimate eventually converges to a value for which half of the input pixels are larger than and half are smaller than this value, that is, the median. The only drawback of the approximated median filter, as seen by Cheung and Kamath [4], is that it adapts slowly toward a large change in background. It needs many frames to learn the new background region revealed by an object that moves away after being stationary for a long time.

Single Gaussian

One of the simplest background removal techniques is to calculate an average image of the scene, subtract each new frame from this image, and threshold the result. This basic Gaussian model can adapt to slow changes in the scene (for example, gradual illumination changes) by recursively updating the model using a simple adaptive filter. This basic adaptive model is used in [17]; also, Kalman filtering for adaptation is used in [68, 73, 74]. The main feature of modelling the probability distribution of the pixel intensity that differentiate it from other ways such as predictive filters is that it ignores the order in which observations are made and focuses on the distribution of the pixel intensities [86].

Gordon *et al.* [53] model each pixel as an independent statistical process, recoding the (R, G, B, Z) observations at each pixel over a sequence of frames in a multidimensional histogram (depth and colour information). Then they use a clustering method to fit the data with an approximation of a mixture of Gaussians. At each pixel, one of the clusters (Gaussians) is selected as the background process, the others are considered to be caused by foreground processes. They are working on extensions which will allow dynamic background estimation based on the previous N frames (allowing modelling slow changes in the background), they are also working on the estimation of multiple background processes at each pixel, similar to [67] but using higher dimensional Gaussians.

Jabri *et al.* [87] model the background in two distinct parts, the colour model and the edge model. For each colour

channel, each pixel is represented by its mean and standard deviation throughout time. The edge model is built by applying the Sobel edge operator to each colour channel yielding a horizontal difference image and a vertical difference image. Weighted means and standard deviations are computed as in the colour model. This model is used to locate changes in the structure of the scene as edges appear, disappear, or change direction. However, their method can not deal with sudden changes in illumination. Moreover, this algorithm doesn't present a solution to the relocation of background object problem [55].

Kalman filter is a widely-used recursive technique for tracking linear dynamical systems under Gaussian noise. It can be viewed as the simplest background model assuming that the intensity values of a pixel can be modelled by a Gaussian distribution $N(\mu, \sigma^2)$, where the mean and variance of the background are updated using simple adaptive filters to accommodate changes in lighting or objects that become part of the background. This basic model is used in [17, 88]. Ridder *et al.* [89] modelled each pixel with a Kalman Filter which made their system more robust to lighting changes in the scene. While this method does have a pixel-wise automatic threshold, it still recovers slowly and does not handle bimodal backgrounds well. Koller *et al.* [90] have successfully integrated this method in an automatic traffic monitoring application. Many different versions of Kalman filter have been proposed for background modelling, differing mainly in the state spaces used for tracking. The simplest version uses only the luminance intensity [17, 91-93]. Karman and von Brandt use both the intensity and its temporal derivative [73], while Koller, Weber, and Malik use the intensity and its spatial derivatives [47].

Zhong and Sclaroff [58] propose an algorithm that explicitly models the dynamic, textured background via an Autoregressive Moving Average (ARMA) model [94]. Although ARMA is a first-order linear model many dynamic textures can be well captured by it [94]. A robust Kalman filter algorithm is used in estimating the intrinsic appearance of the dynamic texture [95]. The foreground object regions are then obtained by thresholding the weighting function used in the robust Kalman filter. In their current implementation, the Kalman filter model is only trained using the empty scenes; however, they could use the Robust PCA method in [77] to train the ARMA model on non-empty scenes in the future.

After visual analysis of Kalman filter results, Cheung and Kamath [4] conclude that Kalman filter produces the worst foreground masks when compares with other schemes. Even with a large foreground threshold and slow adapting rates, the background model in Kalman filter is easily affected by the foreground pixels. As a result, it typically leaves a long trail after a moving object.

Mixture of Gaussians (MoG)

The background of the scene contains many non-static objects such as tree branches and bushes whose movement depends on the wind in the scene. This kind of background motion causes the pixel intensity values to vary significantly with time. So a single Gaussian assumption for the pdf of the

pixel intensity will not hold. Instead, a generalization based on a mixture of Gaussians has been used in [67, 96, 97] to model such variations. In [97] and [67], the pixel intensity was modelled by a mixture of K Gaussian distributions (K is a small number from 3 to 5). In [97], a mixture of three Gaussian distributions was used to model the pixel value for traffic surveillance applications, corresponding to road, shadow, and vehicle distribution. Adaptation of the Gaussian mixture models can be achieved using an incremental version of the EM algorithm. Although, in this case, the pixel intensity is modelled with three distributions, still unimodal distribution assumption is used for the scene background, i.e. the road distribution. Unlike Kalman filter which tracks the evolution of a single Gaussian, the MoG method tracks multiple Gaussian distributions simultaneously. MoG has enjoyed tremendous popularity since it was first proposed for background modelling in [97]. The generalized mixture of Gaussians (MoG) has been used to model complex, non-static backgrounds [40, 67].

Stauffer and Grimson [38] allow the background model to be a mixture of several Gaussians. Every pixel value is compared against the existing set of models at that location to find a match. The parameters for the matched model are updated based on a learning factor. If there is no match, the least-likely model is discarded and replaced by a new Gaussian with statistics initialized by the current pixel value. The models that account for some predefined fraction of the recent data are deemed "background" and the rest "foreground". There are additional steps to cluster and classify foreground pixels into semantic objects and track the objects over time. Javed *et al.* [55] use a mixture of Gaussians method, slightly modified from the version presented by Stauffer and Grimson [38] to perform background subtraction in the colour domain, where the covariance matrix is assumed to be diagonal to reduce the computational cost. A K-means approximation of the EM algorithm is used to update the mixture model. Harville *et al.* [3] propose a method for modelling the background that uses per-pixel, time-adaptive, Gaussian mixtures in the combined input space of depth and luminance-invariant colour. They improve such combination by introducing the ideas of 1) modulating the background model learning rate based on scene activity, and 2) making colour-based segmentation criteria dependent on depth observations. The input to the algorithm is a time series of spatially registered, time-synchronized pairs of colour (YUV space) and depth images obtained by static cameras.

"Background subtraction" is an old technique for finding moving objects in a video sequence---for example, cars driving on a freeway. The idea is that subtracting the current image from a time-averaged background image will leave only non-stationary objects. It is, however, a crude approximation to the task of classifying each pixel of the current image; it fails with slow-moving objects and does not distinguish shadows from moving objects. The basic idea of [98] is that they can classify each pixel using a model of how that pixel looks when it is part of different classes. They learn a mixture-of-Gaussians classification model for each pixel using an unsupervised technique---an efficient, incremental version of EM. Unlike the standard image-averaging approach, this automatically updates the mixture component

for each class according to likelihood of membership; hence slow-moving objects are handled perfectly. Their approach also identifies and eliminates shadows much more effectively than other techniques such as thresholding. Application of this method as part of the Roadwatch traffic surveillance project is expected to result in significant improvements in vehicle identification and tracking.

Eveland *et al.* [51] uses background statistics to model disparity images derived from stereo video using a bimodal distribution, with a Gaussian representing good correlations between left and right images. Eveland proposes an algorithm for updating background statistics on-the-fly called *gated background adaptation*, starting by assuming that the background will appear at some point during the video sequence. Eveland's idea of the gate is to filter acceptable values for the update equations, where any value larger than $G = \mu_t + 3 \cdot 0.9$ above the current mean is rejected in subsequent applications of the update. The gate has the effect of excluding the outlying foreground readings, gradually reducing the estimated background statistics to their true value. To deal with changing background over time, i.e. background objects start to move, or some objects come and remain, the gate is relaxed to consider foreground objects to be background if they stay for a certain length of time.

Francois and Medioni [44] model the background pixel values as multi-dimensional Gaussian distributions in HSV colour space. The distribution for each background pixel is updated using the latest observation, in order to take into account changes in the scene background. They initialize the means using the values of the first frame, and set the standard deviations to zero. The actual distributions are learned as the incoming frames are processed. Lee *et al.* [99] present a Bayesian formulation of the background segmentation problem at the pixel level based on Gaussian mixture modelling since its analytical form fits well into a statistical framework.

However, the MoG has its own drawbacks. First, it is computationally intensive and its parameters require careful tuning. Second, it is very sensitive to sudden changes in global illumination. If a scene remains stationary for a long period of time, the variances of the background components may become very small. A sudden change in global illumination can then turn the entire frame into foreground. Backgrounds having fast variations cannot be modelled with just a few Gaussians accurately, so it fails to provide sensitive detection [82, 99]. However, when foreground objects are included in the training frames, MOG will misclassify [1]. In addition, depending on the learning rate to adapt to background changes, MoG faces trade-off problems. For a low learning rate, it produces then a very wide and inaccurate model that will have low detection sensitivity, not able to detect a sudden change to the background. On the other hand, if the model adapts too quickly, slowly moving foregrounds will be absorbed into the background model, resulting in a high false negative rate. This is the foreground aperture problem described in [1]. However, MoG maintains a density function for each pixel. Thus, it is capable of handling multimodal background distributions. On the other hand, since MoG is parametric, the model parameters can be

adaptively updated without keeping a large buffer of video frames.

Clustering-Based

Butler *et al.* [60] propose a moving object segmentation algorithm with a similar premise to that of Stauffer and Grimson [67] but having the capability of processing 320 x 240 video in real-time on modest hardware. The premise of their algorithm is the more often a pixel takes a particular colour, the more likely that it belongs to the background. Therefore, this requires a technique for maintaining information regarding the history of pixel values. They model each pixel by a group of K clusters where each cluster consists of a weight and an average pixel value or centroid c_k . additionally; the algorithm assumes that the background region is stationary. Incoming pixels are compared against the corresponding cluster group. The matching cluster with the highest weight is sought and so the clusters are compared in order of decreasing weight. A matching cluster is defined to have a Manhattan distance (i.e. sum of absolute differences) between its centroid and the incoming pixel below a user prescribed threshold T . If no matching cluster is found, the cluster with the minimum weight is replaced by a new cluster having the incoming pixel as its centroid and a low initial weight. If a matching cluster is found, then the weights of all clusters in the group are adjusted. The centroid of the matching cluster must also be adjusted according to the incoming pixel. Previous approaches adjust the centroid based on a fraction of the difference between the centroid and the incoming pixel. However, doing so results in fractional centroids and inefficient implementations. Butler chooses instead to accumulate the error between the incoming pixel and the centroid. After adaptation, the weights of all clusters in the group are normalised so that they sum up to one. The normalised clusters are next sorted in order of decreasing weight to aid both the initial cluster comparisons and the final classification step.

Kim *et al.* [61, 62] quantize the background values of each pixel into group of code words constituting a codebook for each pixel, where each pixel might have a different codebook size than the other, according to the pixel variation throughout the time. In order to make their technique more practically useful in a visual surveillance system, they improve their basic algorithm by layered modelling/detection multiple background layers and adaptive codebook updating, to handle changing backgrounds.

Hidden Markov Models (HMM)

All of the previously mentioned models can adapt to gradual changes in illumination. On the other hand, a sudden change in illumination presents a challenge to such models. Another approach to model a wide range of variations in the pixel intensity is to represent these variations as discrete states corresponding to modes of the environment, e.g., lights on/off or cloudy/sunny skies. Hidden Markov models (HMMs) have been used for this purpose in [100,101]. In [100], a three-state HMM has been used to model the intensity of a pixel for a traffic-monitoring application where the three states correspond to the background, shadow, and foreground. In [102], the topology of the HMM representing

global image intensity is learned while learning the background. At each global intensity state, the pixel intensity is modelled using a single Gaussian. It was shown that the model is able to learn simple scenarios like switching the lights on and off.

Major issues in the use of HMMs in real world applications involve two points: real-time computation, and topology modification to address non-stationarity due to dynamically varying conditions. Automatic selection of HMM topology during batch-learning phase has been addressed in the literature. These methods use state-splitting [102, 103], or merging, [104], criteria to iteratively estimate the optimal number of states. More recently, a maximum a posteriori estimation scheme utilizing an entropic prior is presented wherein a redundant number of states is initially assumed and weak states satisfying a given probability threshold are eliminated iteratively [105]. However, dynamic update of the model topology is not addressed in the work. Besides, state-merging schemes are computationally intensive and online updates are not practical [101]. Another interesting piece of work for real time background modelling based on Hidden Markov Models with fixed topology is recently introduced in [100].

Unlike the other methods where the model is fixed and its parameters are updated, Stenger *et al.* [101] present a dynamic framework (e.g. it can change naturally the topology over time) and can deal with sudden as well as gradual illumination changes. They describe a solution to the batch and online estimation of HMM topology. For computer vision problems where the online estimation is critical, they believe that state-splitting is the most reasonable approach since it is computationally efficient. Stenger compares several state splitting criteria such as the chi-squared goodness-of-fit test, the cross-validation criterion and the MDL and AIC criteria. It is seen that the MDL criterion performed the best in terms of minimal variance on the number of states estimated. The topologies were compact and provided the right trade-off between approximation error and model simplicity. An online version of the HMM topology estimation algorithm is also presented. Both the online versions and offline versions are tested on real data. A system and method for coding text data wherein a first group of text data is coded using a Viterbi algorithm using a Hidden Markov model. The Hidden Markov Model computes a probable coding responsive to the first group of text data. A second group of text data is coded using the Viterbi algorithm using a corrected Hidden Markov Model. The Hidden Markov Model is based upon the coding of the first group of text data. Coding the first group of text data includes assigning word concepts to groups of at least one word in the first group of text data and assigning propositions to groups of the assigned word concepts [p2] [106].

Wang *et al.* [59] use an offline Baum-Welch algorithm to learn the parameters of HMM, and use an online algorithm to adapt the parameters, assuming that no obvious sudden change of illumination takes place, the background is assumed to be approximately stationary (to avoid adding a module of recognition of background motion), and the speed of the moving objects doesn't vary greatly. These assumptions are the nature property of video sequence on freeway,

obtaining such video sequence is very easy. Details about offline Baum-Welch algorithm can be found in [107]. Theo-

retical detail of online learning algorithm can be found in [98, 108, 109] as incremental version of the EM algorithm. Wang *et al.* [59] use an exponential forgetting where each pixel value's contribution is weighted so as to decrease exponentially as it recedes into the past.

3.3. Background Initialization

Recently, there has been a large amount of work addressing the issues of background model representation and adaptation (maintenance) [1, 17, 37, 39, 67, 73, 110, 111]. However, a third problem which has received little attention is model initialization (also called bootstrapping in [1]). Actually, most of the background models are built on a set of initial parameters that comes out from a short sequence, in which no foreground objects are present [112]. This is a too strong assumption, because in some situations it is difficult or impossible to control the area being monitored (e.g., public zones), which are characterized by a continuous presence of moving objects, or other disturbing effects. In such cases it may be necessary to train the model using a sequence which contains foreground objects.

According to Gutchess [113], several assumptions are necessary to make the task feasible. Each pixel in the image will reveal the background for at least a short interval of the sequence during the training phase to avoid randomly choosing background appearance. The background is approximately stationary; only small background motion may occur. A short processing delay is allowed subsequent to acquiring the training sequence. Wang and Suter [82] add other assumptions to the above mentioned ones; a foreground object can remain stationary for a short interval in the training sequence. However, the interval should be no longer than the interval from the revealed static background. The background scene remains relatively stable.

Median Filtration

Background initialization using the median intensity value for each pixel is used in a traffic monitoring system [112], relying on the assumption that the background at every pixel will be visible more than fifty percent of the time during the training sequence [113]. However, this may not be always satisfied. A method of creating an image difference overlay comprises identifying a loop of reference images of a subject and identifying a loop of data images of the subject. The loop of image data can be identified after an event, such as the administration of contrast agent to the subject. A reference loop image frame is compared to one or more data loop image frames and the reference loop frame is associated with a data loop image frame which closely resembles the data loop image frame. Each of the associated frames can then be processed and used to create an image difference overlay frame [114]. The advantage of using the median rather than the mean is that it avoids blending pixel values. The mean of a pixel's intensity over time may not correspond to any of the pixel's actual values during that time, in which case it is likely to be in error. Dawson-Howe proposed a similar technique called dynamic background subtraction

(DBS) [115] using a fixed window of three frames to recursively update the background, avoiding the rather expensive cost of computing the median.

Stable Intensity Extraction

Long and Yang [111] propose an algorithm, called the adaptive smoothness method which also avoids the problem of blending, it finds intervals of stable intensity, and uses a heuristic which chooses the longest, most stable interval as the one most likely to represent the background. Their method performs well when all foreground objects are in motion throughout the sequence [113]. However, for sequences in which foreground objects were stationary for a long period of time (sleeping person), many pixels are incorrectly classified. Like most other algorithms, adaptive smoothness makes the decision for each pixel independently, and does not utilize other information from the sequence. Wang *et al.* [82] find one problem of this method is that when the data include multi-modal distributions (i.e., some modes from foreground objects and some modes from background), and when the modes from foreground objects tend to be relatively stable, this method can not differentiate these modes from those from the background. Kurita *et al.* [116] find the value which stays unchanged for a long period of time from the mixture of the stationary signal (the background) and the non-stationary signal caused by the moving objects such as cars, they use a technique from robust statistic [117, 118] by considering the non-stationary signal as outliers. The method works well if the background appears in the sequence more than 50% of the training period.

Relative Constant Intensity Extraction

Gutchess *et al.* [113], motivated by Long and Yang's adaptive smoothness detector [111], add background motion to his algorithm. Their algorithm, called the Local Image Flow algorithm, is similar to adaptive smoothness in that they generate hypotheses by locating intervals of relatively constant intensity. To overcome the sleeping person problem, he considers the optical flow in the neighbourhood surrounding each pixel. If the direction of optical flow in the neighbourhood is toward the pixel, then there is likely to be a moving object approaching the pixel. However, if the majority of optical flow is directed away from the pixel, it is likely that the moving object is leaving the area. Since Gutchess uses only low-level motion information to construct the background. Like the median filter and adaptive smoothness methods, it avoids the problem of blending pixel values present in many current methods. However, the heuristics used in his technique, based on local image flow, are stronger than those used by the median filter and adaptive smoothness. The main strength of the algorithm is that while the decision at each pixel is independent of its neighbours, it is not only based on past values observed at that pixel, but also local motion information. While using optical flow information potentially adds valuable information, most optical flow computation methods themselves are computationally complex and very sensitive to noise [82].

Background with A Mixture of Gaussian Distributions

There are several methods available for building such a mixture model. A widely used algorithm is expectation

maximization, which uses an iterative process to find the best-fitting mixture of Gaussians for a particular dataset [119]. However, the parameters must be updated and calculated offline; therefore, expectation maximization cannot be used when online methods are required. One possible solution is to use the adaptive mixture method proposed in [120], which uses a data-driven approach to estimate the parameters of an underlying mixture model. In Pless's simulations this method has proven to be almost as effective as expectation maximization while operating online [48].

Background Model Based on Kernel Density Estimation

Elgammal *et al.* [37] estimate the density function of this distribution at any moment of time given only very recent history information hoping to obtain sensitive detection, the probability density function that a pixel will have intensity value x_t at time t can be non-parametrically estimated [55] using a kernel estimator. At the first glance, Elgammal chooses his kernel estimator function, K , to be a Normal function $N(0, \Sigma)$, where Σ represents the kernel function bandwidth. Normal kernel function is a generalization of the Gaussian mixture model, where each single sample of the N samples is considered to be a Gaussian distribution $N(0, \Sigma)$ by itself, enabling the model to quickly "forget" about the past and concentrate more on recent observation.

Hidden Markov Models (HMMs)

Numerous methods for estimation of the HMM model parameters exist in the literature. These methods can be classified into two major categories: batch [107] and incremental [121, 122]. Some examples of the batch methods include: the EM algorithm (i.e. Baum-Welch) and segmental K-means algorithm. Offline methods guarantee that the parameter estimates correspond to local maxima, but are computationally expensive. The online methods cited attempt to incrementally update HMM parameters (without topology modifications) with initial starting points determined by batch estimation. These methods can deal with slowly varying drifts in model parameters with the risk of not being able to track the real parameters under sudden changes. Cristani *et al.* [56] propose an initialization algorithm, able to bootstrap an integrated pixel-and region-based background modelling algorithm, where moving objects are present; the output is a pixel- and region-level statistical background model describing the static information of a scene. At the pixel level, multiple hypotheses of the background values are generated by modelling the intensity of each pixel with a Hidden Markov Model (HMM), also capturing the sequentiality of the different colour (or gray-level) intensities. At the region level, the resulting HMMs are clustered with a novel similarity measure, able to remove moving objects from a sequence, and obtaining a segmented image of the observed scene, in which each region is characterized by a similar spatio-temporal evolution. The main drawback of this method is the strong computational effort, but an on-line computation for an initialization algorithm is indeed allowed. Nevertheless, a parallel computational architecture may solve this problem, permitting a very quickly and useful batch mode scheme. Wang *et al.* [59] use an easier method with the help of modified version of the algorithm from Horprasert *et al.* [110]. The

algorithm is based on a heuristic colour model, which separates the brightness from the chromaticity component. The pixel is labelled as background, shadow or foregrounds by a decision procedure whose threshold values are used to determine the similarities of chromaticity and brightness between background image and the current observed image [110]. If chromaticity difference is large, pixel is labelled as foreground, and then if brightness difference is large, pixel is labelled as shadow. The result is then used to initialize their HMM model. An image processing system useful for facial recognition and security identification obtains an array of observation vectors from a facial image to be identified. A Viterbi algorithm is applied to the observation vectors given the parameters of a hierarchical statistical model for each object, and a face is identified by finding a highest matching score between an observation sequence and the hierarchical statistical model [123].

Codebook-Based

Kim *et al.* [61, 62] quantize the background values of each pixel into group of codewords constituting a codebook for each pixel. At the very beginning the codebook of a pixel is empty with no codewords, when a new sample for the pixel encountered (in the training period), if no codewords exist in the codebook, it is assumed to be a codeword, and its brightness value is used to estimate the measures used to represent the codeword, if there are codewords in the codebook, this new pixel sample is compared with each codeword in the codebook using colour distortion measure and brightness bound, if it is matched with a codeword, its brightness value is used to update the measures of this codeword, if there is no match, it is assumed to be a new codeword, and so on till the end of the training period. One way to improve the speed of the algorithm is to relocate the most recently updated codewords to the front of the codebook list. Since most of the time, the matched codeword was the first codeword thus relocated, making the matching step efficient. In the temporal filtering step, he refines the fat codebook by separating the codewords that might contain moving foreground objects from the true background codewords, thus allowing moving foreground objects during the initial training period.

Statistical Approach

Initializing a background model might be approached statistically as the task should be robust against random occurrences of foreground objects, as well as against general image noise. The major advantage of this approach is that it can tolerate over 50% of noise in the data (including foreground pixels), in contrast with methods using the Median statistic which will break down totally when background constitutes less than 50% of the training data [82]. Wang and Suter [82, 124] introduce a consensus-based robust method of background initialization to overcome the problems inherent in methods based on the median filtration, by employing a two-step framework; first all non-overlapping stable subsequences of pixel values are located; using a sliding window with a minimum length, if candidate subsequence with the predefined minimum length can not be found, another minimum length is used, noting that even after this step, the chosen subsequences can contain pixels from foreground,

background, shadows, highlights, etc. The second step is considered a crucial step where the most reliable subsequence is chosen, where the reliability definition is motivated by RANSAC [125], and use the mean value of either the grey-level intensities or the colour intensities over that subsequence as the model background value.

4. FOREGROUND DETECTION

Foreground detection compares the input video frame with the background model, and identifies candidate foreground pixels from the input frame. To obtain this classification, the difference map is usually binarized by thresholding. The correct value of the threshold depends on the scene, on the camera noise, and on the illumination conditions. In the following subsections we will discuss first how to generate the difference map given the background model and the current frame, and then we will discuss the thresholding techniques to obtain foreground-background classification.

4.1. Foreground Detection Techniques

4.1.1. Difference-Based

The most trivial method to perform foreground detection is by taking the difference between two images. Locations of changes correspond to large values in the difference map which can be computed as the absolute values of the difference between corresponding pixels in the two images, relative or normalized difference also can be used [39, 126]. Another approach to introduce spatial variability is to use two thresholds with hysteresis [42, 93]. The basic idea is to first identify "strong" foreground pixels whose absolute differences with the background estimates exceeded a large threshold. Then, foreground regions are grown from strong foreground pixels by including neighbouring pixels with absolute differences larger than a smaller threshold. The region growing can be performed by using a two-pass, connected-component grouping algorithm [127]. Boulton *et al.* [74] keep track with the minimum and maximum values for each pixel throughout the past N frames, such algorithm uses two thresholds, T_L and T_H . The difference between each pixel and the closest background image is computed. If the difference exceeds a low threshold T_L , the pixel is considered as foreground. A target is a set of connected foreground pixels such that a subset of them exceeds the high threshold. The low and high thresholds as well as the background images are recursively updated in a fully automatic way (see [74] for details). Jabri *et al.* [85] perform foreground detection by subtracting the colour channels and the edge channels separately from their corresponding model (mean and standard deviation images) and then combining their results.

Absolute Difference Edge-Based

Getting away from thresholds dilemma, Cavallaro and Ebrahimi [43] apply Sobel edge detector over the absolute difference image between the current frame and the reference frame, a Sobel edge extractor provides thick edges which are useful for application in foreground detection by allowing lightening the post-processing for filling the contours. In a monochrome image, an edge is defined as an intensity discontinuity. In case of colour images, the additional variation in colour may be considered in order to obtain

more complete edge information. Monochrome edge detection, in fact, may not be sufficient for certain scenes. For instance, an object with different hue value from the background but equal intensity can be detected only by taking into account colour information. There are different possibilities to use colour for edge detection purposes. The most straightforward approaches to colour edge detection represent extensions from monochrome edge detection. These techniques are applied to three colour channels independently and then the results (an edge map for each colour channel) are combined by using a certain logical operator. Cavallo adopts this approach which has the advantage of speeding up the computations if the different channels are processed in parallel. The edge information of the three channels is then fused by means of an or logical operator.

Relative Difference

Ideally, the threshold should be a function of the spatial location (x, y) . For example, the threshold should be smaller for regions with low contrast. One possible modification is proposed by Fuentes and Velastin [128]. They use the relative difference rather than absolute difference to emphasize the contrast in dark areas such as shadow. Nevertheless, this technique cannot be used to enhance contrast in bright images such as an outdoor scene under heavy fog.

Normalized Difference

Another popular foreground detection scheme is to threshold based on the normalized statistics.

Predictive-Based

In case of predictive-based background modelling, differences in the state space between the prediction and the observation quantify the amount of change and are considered to perform detection [75]. So a simple mechanism to perform detection is by comparing the prediction with the actual observation. Under the assumption that the autoregressive model is built using background samples, such technique will provide poor prediction for objects while being able to capture the background. Two types of changes in the signal may be considered for detection: (1) "structural" change in the appearance of pixel intensities in a given region, and (2) change in the motion characteristics of the signal. Measures are developed in order to detect each of these types of changes.

4.1.2. Statistical-Based

To calculate an adaptive and local threshold, a region-based statistical analysis can be used if the probability density function of the camera noise is known [50]. The statistical analysis is based on modelling the intensity distribution of noise [29, 31, 32, 34, 35]. Instead of thresholding the difference image, this approach compares the statistical behaviour of a small neighbourhood at each pixel position in the difference image to a model of the noise that could affect the difference image. The comparison is based on a significant test.

Single Gaussian-Based

Pixels in the current frame are compared with the background by measuring the log likelihood in colour space. If a

small likelihood is computed, the pixel is classified as foreground. Otherwise, it is classified as background.

MOG-Based

Francois and Medioni [45] model the background pixel values as multi-dimensional Gaussian distributions in HSV colour space. When a new frame is processed, the value observed for each pixel is compared to the current corresponding distribution in order to decide whether the value is a measurement of the background or of an occluding element. Lee *et al.* [99] present a Bayesian formulation of the background segmentation problem at the pixel level based on Gaussian mixture modelling. From a Bayesian perspective, the foreground decision should be based on the posterior probability of the pixel being background $P(B|x)$ where x denotes the pixel observed in the frame at time t and B denotes the background class. Without giving a precise definition of foreground and background, which is most likely application dependent and requires higher level semantics, Lee proceeds by considering them as two mutually exclusive classes as defined by some oracle. Considering the value observed at a pixel over time is usually resulted from different real world processes, a Gaussian mixture is appropriate to model the distribution, with each Gaussian representing an underlying process.

Kernel Density Estimation-Based

Using the probability estimate of the pixel, the pixel is considered a foreground pixel if $\Pr(x_t) < th$ where the threshold th is a global threshold over all the image that can be adjusted to achieve a desired percentage of false positives.

MRF-Based

Migdal and Grimson [129] have developed an approach to the process of foreground detection that exploits the spatial and temporal dependencies objects in motion impose on their images. This is achieved through the development and use of MRFs during the subtraction process. In total, three MRFs (M_1 , M_2 and M_3) are used, each with different properties. M_1 , being the simplest of the three, only encodes spatial relationships. M_2 , an extension of M_1 , takes advantage of temporal constraints by looking at past segmentations. M_3 , the only field to use a batch computation model, fully exploits the temporal constraint. Segmentations at each time t , S^t , are obtained by estimating the maximum a posteriori (MAP) configuration of the MRF. The MAP estimate is computed using the Gibbs sampler algorithm [130] with a modified (linear) annealing schedule and vastly reduced numbers of iterations. This approach is not tied to any particular background model. In fact, this approach will work with any background model in which a cost function $\delta(d_s)$ can be defined for every pixel d_s . This implies (but does not necessitate) that the background models must be per-pixel based. This is not such a heavy constraint, however. In [48], it is concluded that per-pixel models are sufficient to handle the complex motions present in real world environments.

4.1.3. Clustering-Based

Kim *et al.* [61, 62] quantize background values at each pixel into codebooks; they test the difference of the current image from the background model with respect to colour and

brightness differences. If an incoming pixel meets two conditions, it is classified as background (1) the colour distortion to some codeword is less than the detection threshold, and (2) its brightness lies within the brightness range of that codeword. Otherwise, it is classified as foreground. The codebook method does not evaluate probabilities, which is very computationally expensive. Kim just calculates the distance from the cluster means. That makes the operations fast.

Butler *et al.* [60] classify pixels by summing the weights of all clusters used to model each pixel that are weighted higher than the matched cluster. The result, P is the total proportion of the background accounted for by the higher weighted clusters and is an estimate of the probability of the incoming pixel belonging to the foreground. Larger values of P are evidence the pixel belongs to the foreground and smaller values are evidence that it belongs to the background. This value can be thresholded to obtain a binary decision or can be scaled to produce a gray scale alpha map.

4.2. Thresholding Algorithms

Thresholding is a fundamental method to convert a gray scale image into a binary mask, so that the objects of interest are separated from the background [131]. In the difference image, the gray levels of pixels belonging to the foreground object should be different from the pixels belonging to the background. Thus, finding an appropriate threshold will solve the localization of the moving object problem. The output of the thresholding operation will be a binary image whose gray level of 0 (black) will indicate a pixel belonging to the background and a gray level of 1 (white) will indicate the object.

The efficiency of the foreground detection partially depends on the threshold selection, as clearly observed in the previous schemes. The threshold can be set empirically [30, 132, 133] or computed adaptively [23, 29, 31, 32, 34, 35, 44, 134]. In the former case, the threshold is fixed for all pixels in the frame and all the frames in the sequence. The value is usually determined experimentally based on a large database. In the latter case, the threshold is adapted according to some rules.

Rosin [135, 136] surveyed and reported experiments on many different criteria for choosing the threshold. Smits and Annoni [137] discussed how the threshold can be chosen to achieve application-specific requirements for false alarms and misses (i.e. the choice of point on a receiver-operating-characteristics curve [138]). According to Sirtkaya [139], thresholding algorithms can be divided into 6 major groups [140]. These algorithms can be distinguished based on the exploitation of (1) histogram entropy information, (2) histogram shape information, (3) image attribution information, (4) clustering gray level information, (5) local characteristics and (6) spatial information.

The entropy based methods result in different algorithms which use the entropy of the foreground-background regions or the cross-entropy between the original and binarized image, etc. Assuming that the histogram of an image gives some indication about this probabilistic behaviour, the entropy is tried to be maximized, since the maximization of the entropy of the thresholded image is interpreted as indicative

of maximum information transfer [140, 141]. Histogram shape based methods analyze the peaks, valleys and curvatures of the image histogram and set the threshold according to these morphological parameters. Image attribute methods select the threshold by comparing the original image with its binarized version. The method looks for similarities like edges, curves, number of objects or more complex fuzzy similarities. Iteratively searching for a threshold value that maximizes the matching between the edge map of the gray level and the boundaries of binarized images and penalizing the excess original edges can be a typical example of this approach. Clustering based algorithms initially divide the gray level data into two segments and apply the analysis afterwards. For example, the gray level distribution is initially modelled as a mixture of two Gaussian distributions representing the background and the foreground and the threshold is refined iteratively such that it maximizes the existence probability of these two Gaussian distributions. Locally adaptive methods simply determine thresholds for each pixel or a group of pixel, instead of finding a global threshold. The local characteristics of the pixels or pixel groups, such as local mean, variance, surface fitting parameters etc. is used to identify these thresholds. The spatial methods utilizes the spatial information of the foreground and background pixels, such as context probabilities, correlation functions, co-occurrence probabilities, local linear dependence models of pixels etc. All these methods are tested on difference images of thermal camera sequences [139]. Experiments showed that, the entropy-based approaches [141] give best results for the tested dataset.

5. DATA VALIDATION

The output of a foreground detection algorithm where decisions are made independently at each pixel will generally be noisy, with isolated foreground pixels, holes in the middle of connected foreground components, and jagged boundaries. Cheung and Kamath [4] define data validation as the process of improving the candidate foreground mask based on information obtained from outside the background model. Data validation phase is sometimes referred to as the post-processing phase of the foreground mask (pixels).

There are two kinds of misclassifications that may occur in segmentation results. False positives occur when background regions are incorrectly labelled as foreground. Conversely, false negatives occur when foreground regions are classified as background. Data validation aims to reduce the number of such misclassifications without an appreciable degradation in classification speed.

The simplest techniques simply post-process the foreground mask with standard binary image processing operations, such as median filters to remove small groups of pixels that differ from their neighbours' labels (salt and pepper noise) [87] or morphological operations to smooth object boundaries. Post-processing can be applied either to the binary image representing the foreground map $F(x, y)$ resulted from the foreground detection phase only, or to both the binary image and the original frame. The former case has the advantage of reducing the false alarm probability at low computational cost. However, the a priori topological assumptions (compactness and regular contours) on which they

are based may not always be valid. For this reason, these techniques often result in blocky contours. To solve this problem, the original sequence may be used along with the detection result in the post-processing phase. Motion, colour and edge information are typical examples of features that are analyzed to improve the spatial accuracy of the detection result.

All the background models discussed earlier have three main limitations: first, they ignore any correlation between neighbouring pixels; second, the rate of adaptation may not match the moving speed of the foreground objects; and third, non-stationary pixels from moving leaves or shadow cast by moving objects are easily mistaken as true foreground objects. The first problem typically results in small false-positive or false-negative regions distributed randomly across the candidate mask. False positives resemble pepper noise and are typically attributed to camera noise [60]. That is, they are small (1-2 pixel), incorrectly classified regions surrounded by correctly classified background pixels. False negatives arise because of the existence of similarities between the colours of foreground objects and the background. They form holes in correctly classified foreground regions and can be quite large. Consequently, they are more difficult to remove than false positives. The most common approach is to combine morphological filtering and connected component grouping to eliminate these regions [1, 38, 87, 91], whilst preserving the contours of correctly classified regions. Applying morphological filtering on foreground masks eliminates isolated foreground pixels and merges nearby disconnected foreground regions. Many applications assume that all moving objects of interest must be larger than a certain size. Connected-component grouping can then be used to identify all connected foreground regions, and eliminates those that are too small to correspond to real moving objects [45].

6. PERFORMANCE EVALUATION

Many algorithms have been proposed for moving object detection in many applications, as a primary step towards video segmentation. However, the segmentation quality performance evaluation of those algorithms is often ad-hoc, and a well-established solution is not available [142]. In fact, the field of objective evaluation is still maturing. Performance evaluation allows the appropriate selection of segmentation algorithms as well as adjusts their parameters for optimal performance [143]. The current practice for evaluation involves a representative group of human viewers which is subjective, time consuming and expensive process [142]. Subjectivity can be minimized by following strict evaluation conditions, with the video quality evaluation recommendations developed by ITU providing valuable guidelines [144, 145]. Alternatively, objective evaluation methodologies can be used, to mimic, using an automatic procedure, the results that a formal subjective evaluation would produce. Depending on the availability, or not, of reference segmentation (the so-called ground truth), two alternatives can be considered for the evaluation of video segmentation quality; *standalone evaluation* when the reference segmentation is not available and *relative evaluation* when the reference segmentation is available for comparison.

According to Corriea and Pereira [142, 143], two types of measurements can be targeted when performing video segmentation evaluation; *individual object segmentation evaluation* when one of the objects identified by the segmentation algorithm is independently evaluated in terms of its segmentation quality, which is valuable when objects are independently manipulated, e.g. for reusing in different contexts, and *overall segmentation evaluation* when the complete set of objects identified by the segmentation algorithm is globally evaluated in terms of its segmentation quality. This requires the estimation of individual object evaluation, and the weighting of those values according to each object's relevancy in the scene, since segmentation errors in the more important objects are more noticeable to a human viewer. It may determine whether the segmentation algorithm is adequate for the application addressed.

6.1. Evaluation Methodology

Correia and Pereira [143] propose the methodology for performing individual object segmentation evaluation, which consists of three major steps: (1) *Segmentation*; the segmentation algorithm is applied to the test sequences selected as representative of the application domain in question. (2) *Object selection*; the object whose segmentation quality should be evaluated is selected. (3) *Segmentation evaluation*; the objective segmentation evaluation metric, as surveyed later, is computed. This metric differs for standalone and relative evaluation.

The methodology for objective overall segmentation evaluation follows a five-step approach as proposed by Correia and Pereira [145], both for the standalone and the relative evaluation cases. These steps are: (1) *Segmentation*; the segmentation algorithm is applied to the test sequences selected as representative of the application domain in question. (2) *Individual object segmentation evaluation*; for each object, the corresponding individual object segmentation quality, either standalone or relative, is evaluated. (3) *Object relevance evaluation*; the relevance of an object must be evaluated taking into account the context where it is found. This relevance metric reflects the importance of an object in terms of the human vision system and can be computed by the combination of a set of metrics expressing the features able to capture the viewers' attention. It can also be seen as a measure of the likeliness that one object will be further processed, manipulated and used, since users tend to reuse and manipulate more the objects which capture more their attention (see [146] for metric evaluation). (4) *Similarity of objects evaluation*; the correctness of the match between the objects identified by the segmentation algorithm and those relevant for the targeted application is evaluated (see [139] for details). This step is different depending on whether standalone or relative evaluation is being performed. (5) *Overall segmentation evaluation*; by weighting the individual segmentation evaluation for the various objects in the scene with their relevance values (see [139] for details).

6.2. Relative Performance Evaluation

Relative evaluation is expected to provide more reliable evaluation results as it has access to ground truth information. Three approaches have been recently considered [126]: pixel-based, template-based and object-based methods. Pixel

based methods assume that we wish to detect all the active pixels in a given image. Moving object detection is therefore formulated as a set of independent pixel detection problems. This is a classic binary detection problem provided that we know the ground truth. The algorithms can therefore be evaluated by standard measures used in Communication theory e.g., misdetection rate, false alarm rate and receiver operating characteristic (ROC) [147]. Several proposals have been made to improve the computation of the ROC in video segmentation problems e.g., using a perturbation detection rate analysis [148] or an equilibrium analysis [149]. However we are not interested in the detection of point targets but object regions instead. The computation of the ROC can also be performed using rectangular regions selected by the user, with and without moving objects [150] which improve the evaluation strategy since the statistics are based on templates instead of isolated pixels. A third class of methods is based on an object evaluation. Most of the works aim to characterize colour, shape and path fidelity by proposing figures of merit for each of these issues [151-153] or area based performance evaluation as in [154].

These approaches have three major drawbacks. First object detection is not a classic binary detection problem [126]. Several types of errors should be considered (not just misdetection and false alarms). For example, what should we do if a moving object is split into several active regions? or if two objects are merged into a single region? Second some methods are based on the selection of isolated pixels or rectangular regions with and without persons. This is an unrealistic assumption since practical algorithms have to segment the image into background and foreground and do not have to classify rectangular regions selected by the user. Third, it is not possible to define a unique ground truth. Many images admit several valid segmentations. If the image analysis algorithm produces a valid segmentation its output should be considered as correct.

6.2.1. Ground Truth Generation

Ground truth (or gold standard) generation can be viewed as the process of establishing the “correct answer” for what *exactly* the algorithm is expected to produce, which is generally application-specific [5]. For example, in video surveillance, it is generally undesirable to detect the “background” revealed as a consequence of camera and object motion as change, whereas in remote sensing, this change might be considered significant (e.g. different terrain is revealed as a forest recedes). Video motion detection apparatus in which successive output images of an output video signal are generated with respect to images of an input video signal comprises means for applying a motion test to detect inter-image motion between two or more images of the input video signal. The motion test provides 2 sets of one or more motion vectors for use in the generation of a respective output image and test output images using the respective sets of motion vectors are generated. Image areas in the test output images pointed to by the motion vectors in one or both sets, are compared and if a motion vector has less than a predetermined degree of similarity between the image areas, the image areas are divided into two or more smaller image areas and the comparison is applied again in respect of each of the two or more smaller areas. The use of motion vectors for

which the corresponding image areas in the test output images have less than a predetermined degree of similarity at the image area sizes tested is inhibited. The invention has application in standards conversion and polyphase interpolation [155]. It is an image analysis problem that is known to be difficult and time consuming [156]. Levine and Nazif [157] suggested assessing the quality of image segmentation without reference to a ground truth image, but by measuring for each region its internal homogeneity and its contrast along its boundaries. In a similar vein, Kitchen and Rosenfeld [158] assessed the quality of thresholded edge maps using edge continuity and thinness. However, the problem is that these criteria do not always reflect good results (Venkatesh and Rosin [159]).

In most cases, ground truth is essential for performing a quantitative analysis of an algorithm’s results. There are three main approaches to generating ground truth [136]. The first uses synthetic data; example applications are ellipse fitting (Fitzgibbon *et al.* [160]), edge detection (Venkatesh and Kitchen [161]), corner detection (Zheng *et al.* [162]), and optic flow (Barron *et al.* [163]). This method enables ground truth to be easily provided; the problem is that the synthetic data will probably not faithfully represent the full range of real data. Alternatively, real image data can be manually annotated, e.g. to mark edge and no-edge pixels. Usually, this is done by an expert human observer. As noted by Tan *et al.* [164], multiple expert human observers can differ considerably even when they are provided with a common set of guidelines. The same human observer can even generate different segmentations for the same data at two different times. So the algorithm designer may be faced with the need to establish ground truth from multiple conflicting observers. A third approach avoids explicitly determining a ground truth dataset, and relies instead on evaluating the algorithms’ outputs by a human panel. A conservative method is to compare the algorithm against the set intersection of all human observers’ segmentations. In other words, a foreground detected by the algorithm would be considered valid if every human observer considers it a foreground. A method that is less susceptible to a single overly conservative observer is to use a majority rule. The least conservative approach is to compare the algorithm against the set union of all human observers’ results. Finally, it is often possible to bring the observers together after an initial blind markup to develop a consensus markup. Two disadvantages are the time consuming nature of the exercise (more images need to be viewed), and the difficulty in incorporating additional algorithms into the evaluation results at a later date (unless the same panel is reconvened).

6.2.2. Pixel-based Performance Metrics

Once a ground truth has been established, there are several standard methods for comparing the ground truth to a candidate binary foreground map. The following quantities are generally involved:

- True positives (TP): the number of foreground pixels correctly detected;
- False positives (FP): the number of background pixels incorrectly detected as foreground (also known as false alarms);

- True negatives (TN): the number of background pixels correctly detected; and
- False negatives (FN): the number of foreground pixels incorrectly detected as background (also known as misses).

Based on the above mentioned quantities, Rosin [136] described three methods for quantifying a classifier's performance:

- The Percentage Correct Classification

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

- The Jaccard Coefficient $JC = \frac{TP}{TP + FP + FN}$
- The Yule Coefficient $YC = \left| \frac{TP}{TP + FP} + \frac{TN}{TN + FN} - 1 \right|$

Combining all four values to form the PCC is the most widespread method in computer vision for assessing a classifier's performance. However, it tends to give misleading estimates when the amount of change is small compared to the overall image [136]. The Yule and Jaccard coefficients overcome this problem to some degree by minimizing or eliminating the effect of the expected large volume of true negatives. Note that the Yule coefficient cannot be applied when the algorithm correctly detected no change in the image (since one denominator becomes zero). Within the sequences there might be frames in which no change occurs which can be analyzed separately to monitor the effects of noise and compression artifacts when no real activity exists in the sequence.

Cheung and Kamath [4] compare the performance of a number of popular background removal techniques using two information retrieval measurements, recall and precision, to quantify how well each algorithm matches the ground-truth [165]. They are defined in their context as; *Recall* is the ratio of the number of foreground pixels correctly identified by the algorithm to the number of foreground pixels in ground truth, while *Precision* is defined as the ratio of the number of foreground pixels correctly identified by the algorithm to the number of foreground pixels detected by the algorithm. Recall and precision values are both within the range of 0 and 1. Typically, there is a trade-off between recall and precision - recall usually increases with the number of foreground pixels detected, which in turn may lead to a decrease in precision. A good background algorithm should attain as high a recall value as possible without sacrificing precision [4].

Given the ground truth, Nascimento and Marques [126] detect several types of errors i) splits of foreground regions, ii) merges of foreground regions, iii) simultaneously split and merge of foreground regions, iv) false alarms (detection of false objects) and v) the detection failures (missing active regions). They then compute statistics for each type of error.

6.2.3. Object-based Performance Metrics

Object-based metrics usually involve both spatial and temporal accuracy metrics [143]. Additionally, some metrics like the criticality can be considered as spatio-temporal, since they simultaneously cover spatial and temporal aspects of the complexity of a sequence [166]. The results obtained with each metric are normalized to the range [0,1] (see [143] for details).

Spatial Accuracy Metrics

A good segmentation must have contours very similar to those of the reference segmentation. When a perfect shape match is not achieved, object features can be compared so that spatial segmentation errors contribute to lower the segmentation quality values. An image processing apparatus in which output pixels of an output image are generated from one or more input images using motion vectors having a sub-pixel accuracy. It comprises a motion vector allocator for allocating motion vectors to pixels of the output image, the motion vector allocator being arranged to compare a current output pixel with test image areas pointed to by motion vectors to detect a most suitable motion vector for the current output pixel. The motion vector allocator comprises a spatial filter for comparing the current output pixel and a test image area to sub-pixel accuracy and a pixel generator for generating the output pixels. The pixel generator comprising a spatial filter for generating an output pixel value at a required pixel position to a sub-pixel accuracy, in which the spatial filter of the motion vector allocator has fewer filter taps than the spatial filter of the pixel generator [p6] [167]. The spatial accuracy features selected for relative object-based evaluation include the following [143].

Shape fidelity; the number of misclassified shapes and their distances to the reference object's border are taken to compute the fidelity of the object shape. It can viewed also as the percentage of misclassified edge pixels, eventually weighting the edge pixels in error with a function of their distance to the reference edge, is used by several authors for the evaluation of edge detectors [168, 169]. Alternatively, the correlation between estimated and reference edge pixels has also been considered [170]. More recently, similar metrics were proposed for the evaluation of segmentation partitions composed of two objects (foreground and background), notably counting the misclassified shape pixels (shapels) [171] and weighting the erred shapels according to their distance to the reference [172].

Geometrical similarity; those features are based on the size, position and a combination of the elongation and compactness of the objects [139].

Edge content similarity; the similarity in terms of object edge content is evaluated using two metrics: the output of a Sobel edge detection filter and the instantaneous value of the spatial perceptual information (SI).

Statistical data similarity; since the human observer is especially sensitive to the brightness information and to image areas with red colour, a metric for evaluating the statistical similarity of brightness and redness of objects is used.

Number of objects comparison; the comparison of the number of objects identified in the estimated and reference segmentations also gives an indication about the correctness of the segmentation results. One such metric, called fragmentation has been proposed in [173].

Temporal and Spatio-Temporal Accuracy Metrics

When the estimated object shapes are not perfect, also the temporal dimension of video can be used to identify the more objectionable segmentation errors. As congestion increases in the X and Ku frequency bands an increasing number of communication systems are being designed using Ka and V frequency bands. However, at these higher frequencies the propagation impairments caused by meteorological phenomena (such as cloud, rain etc) become significant. The high levels of attenuation mean that a static fade margin is not practical and dynamic fade mitigation techniques must be used. Described herein is a method for the generation of radiowave propagation time series using estimates of the meteorological environment from Numerical Weather Prediction (NWP) systems. The resulting time series are shown to exhibit the correct first and second order characteristics, as well as the correct spatial correlation. The long term statistics are shown to compare well to the long term cumulative distribution functions produced using ITU-R recommendation P618-8 [174]. Temporal accuracy has received less attention in the literature than spatial accuracy. After a set of tests, two metrics have been selected for temporal accuracy evaluation [143].

Temporal perceptual information; the fidelity between the motion in the reference and the estimated objects is measured by the instantaneous value of the temporal perceptual information (TI) metric.

Criticality; simultaneously considers spatial and temporal characteristics of the objects and is used for evaluation of the temporal accuracy.

Temporal stability; a metric computing the difference in the number of object shapes, i.e., the object size, for consecutive time instants has been used for evaluating the temporal stability in comparison to a reference [171].

Composite Evaluation Metric

The metric for the relative evaluation of individual objects consists in a combination of the elementary metrics described before, capturing the effects of the various types of errors that may affect the segmentation quality. In this case, a single composite metric is used [143]. The weights for the various classes of features included in the composite metric have been selected taking into account both their strength in capturing the human visual attention and their ability to match subjective evaluation results.

According to Correia and Pereira [143], shape fidelity is given the largest weight, as it is the main indication of a mismatch with the reference. Recognizing the importance of the temporal information in terms of the HVS, the temporal fidelity metric receives the second highest weight. The remaining metrics account for a little over one third of the total weights, as they allow distinguishing the different types of spatial and temporal dissimilarities. The subjective tests per-

formed by Correia allowed the selection of weights for the different classes of features, as well as for the elementary metrics within each class of features. For the geometrical similarity class of metrics, it was observed that differences in size should have the largest contribution, followed by the differences in position and in elongation and compactness. The two metrics of the edge content similarity class were considered equally important, as none of them has clear advantages over the other.

6.2.4. Evaluation Methodology

In order to compare the output of the moving object detection algorithm with the ground truth segmentation, a region matching procedure is usually adopted [175] which allows establishing a correspondence between the detected objects and the ground truth, which is performed by computing a binary correspondence matrix defining the correspondence between the foreground regions in a pair of images. When associating ground truth regions with detected regions six cases can occur [175]: zero-to-one, one-to-zero, one-to-one, many-to-one, one-to-many, and many-to-many associations. According to Nascimento and Marques [175], these correspond to false alarm (the detected region has no correspondence), misdetection (the ground truth region has no correspondence), correct detection (the detected region matches one and only one region), merge (the detected region is associated to several ground truth regions), split (the ground truth region is associated to several detected regions) and split-merge.

The region based measures mentioned so far depends on an overlap requirement between the region of the ground truth and the detected region. Without this requirement, a single pixel overlap is enough for establishing a match between a detected region and a region in the ground truth segmentation, which does not make sense. A match is determined to occur if the overlap is at least as big as that overlap requirement. The bigger the overlap requirement, the more the pixels are required to overlap hence performance usually declines as the requirement reaches 100%. Nascimento [175] uses an overlap requirement of 10%. The match between pairs of the two regions is also considered to measure the performance of the algorithms. The higher is the percentage of the match size, the better are the active regions produced by the algorithm. This is done for all the correctly detected regions.

Sometimes the segmentation procedure is subjective, since each foreground region may contain several objects and it is not always easy to determine if it is a single connected region or several disjoint regions. Since we do not know how the algorithm behaves in terms of merging or splitting, every possible combination within elements, belonging to a group, must be taken into account. This suggests the use of multiple interpretations for the segmentation [175]. To accomplish this, the evaluation setup takes into account all possible merges of single regions belonging to the same group whenever multiple interpretations should be considered in a group, i.e., when there is a small overlap among the group members. The number of merges depends on the relative position of single regions. Instead of asking the user to identify all the possible merges in an ambiguous

situation, an algorithm is used to generate all the valid interpretations in two steps [126]. First we assign all the possible labels sequences to the group regions. If the same label is assigned to two different regions, these regions are considered as merged. The second step checks if the merged regions are close to each other and if there is another region in the middle. The invalid labelling configurations are removed. Nascimento and Marques [126] present a detailed description of the labelling method in appendix VII-A.

6.2.5. Receiver-Operating Characteristics (ROC) as Measures of Quality

The receiver-operating-characteristics (ROC) curve plots the detection probability versus the false alarm probability to determine a desired level of performance. However, since an ROC curve gives little understanding of the qualitative behaviour of a particular algorithm in different regions of an image sequence, visual inspection of the foreground maps is still recommended [62]. When comparing background removal algorithms [149] or evaluating computer vision systems [176, 177], ROC analysis is often employed when there are known background and foreground (target) distributions, i.e. ground truth data. ROC curves display the detection sensitivity for detecting a particular foreground against a particular background. These plots attempt to show the general performance of a classifier over the range of its possible threshold values. The quality of a particular algorithm can be inferred from its ROC plot in several ways. A common approach is to measure the area under the ROC curve. Another popular quality measure is the “distance” between the curve and the perfect classifier point. Gao *et al.* [149] use ROC curves to graph the probability of false alarm versus the probability of miss detection to set the system parameters; they combine multiple ROC plots for different values of some of the system’s fixed parameters.

The ROC curve approach has several weaknesses. One significant problem is that once an ROC plot is generated, it is impossible to infer the amount or nature of the data considered when creating the plot-information that is clearly important for ascertaining the relevance of the curve. Another problem concerns threshold selection. Many different thresholds are used to generate an ROC plot. From these plots, one can infer to some extent the general behaviour of the algorithm in hand and can perhaps gain basic insight into which values may be good thresholds. These plots can vary widely from one application to another, though, and the optimal threshold may drastically change with the situation. ROC also has some disadvantages when used to evaluate background removal algorithms in particular. There are as many ROC curves as there are possible different foreground targets [149]. In addition, it requires considerable experimentation and ground-truth evaluation to obtain accurate false alarm rates (FA) and the miss detection rates (MD).

6.2.6. Perturbation Detection Rate (PDR) Analysis

Horprasert *et al.* [148] present the perturbation method, called perturbation detection rate (PDR) analysis, which measures the sensitivity of a background removal algorithm without assuming knowledge of the actual foreground distribution. Rather, it measures the detection of a variable, small

(“just-noticeable”) difference from the background, obtaining a foreground distribution by assuming that the foreground might have a distribution locally similar in form to the background, but shifted or perturbed. The detection is measured as a function of contrast, the magnitude of the shift or perturbation in uniform random directions in RGB. The basic idea is to measure how far apart the two distributions must be in order to achieve a certain detection rate, or stated otherwise, given a false alarm rate (FA-rates), to determine detection rate as a function of the difference of the foreground from the background. It is similar to the *Just Noticeable Difference* (JND) typically used in comparing psychological magnitudes [148].

PDR analysis has two advantages over the commonly used ROC analysis [148]: (1) It does not depend on knowing foreground distributions; (2) It does not need the presence of foreground targets in the video in order to perform the analysis, while this is required in the ROC analysis. Because of these considerations, PDR analysis provides practical general information about the sensitivity of algorithms applied to a given video scene over a range of parameters and FA-rates. In ROC curves, we obtain one detection rate for a particular FA-rate for a particular foreground and contrast.

However, according to Kim [62], there are limitations. The method doesn’t model motion blur of moving foreground objects. Also in the case of mixed (moving) backgrounds, the simulated foreground distributions will be mixed (as plants or flags moving in the foreground); usually, though, foreground targets are from unimodal distributions. It should be noted, however, that the overall detection rates will be nearly the same if the clusters of the mixed distributions are well separated (compared to the usual small contrast delta). An important limitation is that foreground objects often will have shading and reflection effects on backgrounds and these are ignored although they are important for choosing a proper, practical false alarm rate for real video analysis. PDR does not predict the overall performance of a background removal algorithm, but shows detection rates for possible foreground targets given the background scene. ROC analysis is also very useful if a specific real target is known and crucial to the application. Kim [62] would not generally claim that one algorithm is better than another just from PDR analysis. There are other important performance criteria which are not compared, such as processing speed, memory capacity, online model update, etc.

The PDR method would seem to be useful for qualitative comparison of sensitivity of different algorithms, as well as comparison of choice of parameters for a particular algorithm with respect to sensitivity. In the future, the present method could be extended to measure local detection rates throughout the frame of the scene or varying over time [148, 62]. This might have application to localized parameter estimation, e.g. of detection/adaptation parameters in different parts of the frame of the scene. In the parameter space, every combination of parameters determines its FA-rate. One could select those combinations that produce the target FA-rate, and then plot a family of PDR graphs for them. One could then choose the algorithm parameters that provide best detection sensitivity with respect to the PDR analysis.

6.3. Standalone Performance Evaluation

Standalone evaluation is performed when no reference segmentation is available. It is not expected to provide as reliable results as the evaluation relative to reference segmentation; these results mainly provide qualitative information for the ranking of segmentation partitions and algorithms. Standalone objective evaluation algorithms work mainly with the a priori information available about the expected properties of objects and of their disparity to neighbours, in the context of the application addressed.

Metrics for individual object standalone evaluation can be established based on the expected feature values computed for each object (*intra-object metrics*), as well as on the observed disparity of some key features relative to the neighbours (*inter-object metrics*). These metrics are normalized to produce results in the interval [0, 1], with the highest values associated to the best segmentation results [143].

6.3.1. Intra-Object Homogeneity Metrics

Intra-object homogeneity can be evaluated by means of spatial and temporal object features. The spatial features selected for individual object evaluation and corresponding metrics, are: *Shape regularity*; the regularity of the object shapes can be evaluated by geometrical features such as the compactness, or a combination of circularity and elongation and thickness being defined as the number of morphological erosion steps [98] that can be applied to the object until it disappears. *Spatial uniformity*; it can be evaluated by such metrics as the spatial perceptual information (SI) and the texture variance [157].

The temporal features selected for individual object evaluation and the corresponding metrics, are: *Temporal stability*; a smooth temporal evolution of selected object features may be checked for the evaluation of temporal stability. Those features include size, position, TI, criticality, texture variance, circularity, elongation and compactness). *Motion uniformity*; may be evaluated by features such as the variance of the object's motion vector values, or the criticality.

6.3.2. Inter-Object Disparity Metrics

Inter-object disparity features give an indication if the objects were correctly identified as separate entities, i.e. the various objects are really different according to some criteria. These features can be computed either locally along the object boundaries, or for the complete object area. The selected inter-object disparity metrics are: *Local contrast to neighbours*; a local contrast metric may be used for evaluating if a significant contrast between the inside and outside of an object, along the object border, exists. *Neighbouring objects feature difference*; several features computed for the object area, can be compared between neighbours. Examples are the shape regularity, spatial uniformity, temporal stability and motion uniformity, whenever each of them is relevant for the targeted application.

6.3.3. Composite Metrics

Since the usefulness of the various standalone evaluation elementary metrics has a strong dependency on the characteristics of the content considered and thus on the application

addressed, it is not possible to establish a single general-purpose composite metric for standalone evaluation. Instead, the approach taken by Correia [143] is to select two major classes of content, differing in terms of their spatial and temporal characteristics and proposing different composite evaluation metrics for each of them. The two selected classes of content are: *Stable content*; which is temporally stable (scene background) and includes objects with reasonably regular shapes; additionally, neighbouring objects are expected to be contrasted. *Moving content*; for this type of content (foreground objects), the motion of objects can be strong and thus temporal stability is less relevant. Often, the motion of objects is uniform and neighbouring objects may be spatially less contrasted, while motion differences between neighbours are expected to be larger. Regular shapes are still expected, even if assuming a lower importance.

The stable content composite metric does not include the spatial and motion uniformity related elementary metrics, as arbitrary spatial patterns may be found in the expected objects (e.g., the clothing of people might produce misleading results) and the amount of expected motion in this case is very small leading to non significant values for motion related metrics. Correia and Pereira [143] propose composite metric for stable content including; *Shape regularity*, the two elementary metrics compact and elongation are combined with equal weights. *Temporal stability*; it is evaluated by combining the stabilities related to the size, elongation and criticality, all with equal weights. *Local contrast to neighbours*; the local contrast metric is selected for the evaluation of the contrast between neighbouring objects.

For moving content, the composite metric includes again only the relevant classes of elementary metrics. In this case, the content is not expected to be temporally stable, but the objects should have reasonably uniform motion and the neighbouring objects motion differences should be pronounced. Thus, the classes of metrics adopted for the standalone evaluation of moving content are [143]: *Shape regularity*; the two elementary metrics compactness and elongation are again combined with equal weights, as they complement each other without a clear advantage for any of them. *Motion uniformity*; the criticality metric is used to represent this class of features. *Local contrast to neighbours*; even if the local contrast is not so important in terms of segmentation evaluation as for stable content, the contrast metric is yet considered useful. *Neighbouring object features difference*; since neighbouring objects are expected to exhibit different motion characteristics, the motion uniformity difference metric is used.

CURRENT & FUTURE DEVELOPMENTS

We believe that “No perfect system exists. Background modelling and subtraction in itself is applications oriented”. Background modelling and subtraction in itself is only a pre-processing, absolutely not the ultimate task. A perfect system should solve many problems, such as “bootstrapping”, “moved objects”, shadows, gradually and suddenly change of illumination, “tree waving”, “camouflage” and so on. But some of these can't be solved very well simultaneously because differentiating of them needs semantic understanding of motion of foreground and of background, and it is impos-

sible if you have no information from the ultimate purpose. Further more, in a particular application, not all the problems will be encountered. A good system should use the knowledge derived from its purpose as possible as enough to solve the problems encountered. No perfect system exists, but a good framework will give background modelling and subtraction much help. When there is an expected semantic interpretation of the foreground pixels in a given application (e.g. an intruder in surveillance imagery), the algorithm designer should incorporate higher-level constraints that reflect the ability of a given algorithm to detect the “most important” changes, instead of treating every pixel equally.

REFERENCES

- [1] Kentaro T, John K, Barry B, Brian M. Wallflower: Principles and Practice of Background Maintenance, ICCV, *Seventh Int. Conf. on Com Vision (ICCV'99)* - 1999; 1: 255-261.
- [2] Elgammal A, Duraiswami R, Harwood D, Davis LS. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proc. IEEE Jul. 2002*; 90(7): 1151-1163.
- [3] Harville M., Gordon G., Woodfill J. Foreground Segmentation Using Adaptive Mixture Models in Color and Depth. event, *IEEE Workshop on Detection and Recog of Events in Video (EVENT'01)*, 2001; 3-12
- [4] Cheung S-C, Kamath C. Robust techniques for background subtraction in urban traffic video. In S. Panchanathan and B. Vasudev, editors, *Proc Elect Imaging: Visual Comm Image Proce 2004 (Part One) SPIE*, 2004; 5308: 881-892.
- [5] Radke RJ, Andra S, Al-Kofahi O, Roysam B. Image Change Detection Algorithms: A systematic survey, image processing, *IEEE Trans on March 2005*, 14(3): 294-307.
- [6] Henninger, P.E., Bell, M. S., Magrid, B.: GB2433173 (2007).
- [7] Brown LG. A survey of image registration techniques. *ACM Computer Surv 1992*; 24(4): 325 - 376.
- [8] Lavallee S. Registration for computer-integrated surgery: methodology, state of the art. MIT Press, 1995.
- [9] Maintz JBA, Viergever MA. A survey of medical image registration. *Med Image Anal 1998*; 2(10): 1-36.
- [10] Zitov 'a B, Flusser J. Image registration methods. a survey. *Image Vision Comput 2003*; 21: 977-1000.
- [11] Ibanez L, Schroeder W, Ng L, Cates J. The ITK Software Guide: The Insight Segmentation and Registration Toolkit (version 1.4). Kitware Inc., 2003.
- [12] Stewart CV, Tsai C-L, Roysam BA. Dual bootstrap iterative closest point (ICP) algorithm: Application to retinal image registration. *IEEE Trans Med Imaging - Special Issue on Medical Image Registration November 2003*; 22: 11.
- [13] Barron J, Fleet D, Beauchemin S. Performance of optical flow Techniques. *Int J Comp Vision 1994*; 12(1): 43-77.
- [14] Blake A, Isard M. Active Contours. Springer Verlag, 1999.
- [15] Lowe DG. Distinctive image features from scale-invariant key points. *Int J Comp Vision November 2004*; 60(2): 91-110(20).
- [16] Wells W. Statistical approaches to feature-based object Recog. *Int J Comp Vision 1997*; 21(1/2): 63-98.
- [17] Wren CR, Azarbayejani A, Darrell T, Pentland A, Pfinder: Real-time tracking of the human body. *IEEE Trans Pat Anal Mach Intel 1997*; 19(7): 780-785.
- [18] Hartley R., Zisserman A. Multiple view geometry in computer vision. Cambridge University Press, 2000.
- [19] Lillestrand R. Techniques for change detection. *IEEE Trans. On Computers 1972*; 21(7): 654-659.
- [20] Ulstad MS. An algorithm for estimating small scale differences between two digital images. *Pat Recog 1973*; 1(5): 323-333.
- [21] Dai X, Khorram S. The effects of image misregistration on the accuracy of remotely sensed change detection. *IEEE Trans. Geoscience Remote Sensing September 1998*; 36(5): 1566-1577.
- [22] Greiffenhagen M., Ramesh V., Comaniciu D., Niemann H., Statistical modelling and performance characterization of a real-time dual camera surveillance system. *Proc. Int. Conf. Comp Vision Pat Recog*, 2000; 2; 335-342.
- [23] Toth D., Aach T., Metzler V., Bayesian spatio-temporal motion detection under varying illumination illumination-invariant change detection. In *Proc. of X EUSIPCO*, Tampere, Finland, 2000; 3-7.
- [24] Toth TAD, Metzler V. Illumination-invariant change detection. In *The 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, April 2000; 3.
- [25] Aach T., D'umbgen L., Mester R., Toth D. Bayesian illumination-invariant motion detection. In *Proc. IEEE Int Conf Image Proces*, October 2001; 640-643.
- [26] Pizarro O, Singh H. Toward large-area mosaicing for underwater scientific applications. *IEEE J Oceanic Eng October 2003*; 28(4): 651-672.
- [27] Can A, Singh H. Methods for correcting lighting Patt and attenuation in underwater imagery. *IEEE J Oceanic Eng 2004*, in review.
- [28] Hager GD., Belhumeur PN. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Patt Anal. Machine Intell 1998*; 20(10): 1025-1039.
- [29] Neri A, Colonnese S, Russo G, Talone P. Automatic moving object and background separation. *Signal Processing 1998*; 66(2): 219-232.
- [30] Skifstad K, Jain R. Illumination independent change detection for real world image sequences. *Comp Vision, Graphics Image Process June 1989*; 46: 387-399.
- [31] Hotter M, Mester R, Muller F. Detection and description of moving objects by stochastic modelling and analysis of complex scenes. *Signal Proce: Image Comm 1996*; 8: 281-293.
- [32] Aach T, Kaup A, Mester R. Statistical model-based change detection in moving video. *Signal Process 1993*; 31: 165-180.
- [33] Liu SC, Fu CW, Chang S. Statistical change detection with moments under time-varying illumination. *IEEE Transactions on Image Processing 1993*; 7(9): 1258-1268.
- [34] Mech R, Wollborn M. A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera. *Signal Process 1998*; 66(2): 203-217.
- [35] Cavallaro A, Ebrahimi T. Video object extraction based on adaptive background and statistical change detection. In *Proc. of SPIE Visual Comm Image Process (VCIP) 2001*; 465-475.
- [36] Cutler R, Davis L. View-based detection and analysis of periodic motion in: *Int Conf Patt Recog Brisbane, Australia (Aug.) 1998*; 1: 495-500.
- [37] Elgammal A., Harwood D., Davis L. Non-parametric Model for Background Subtraction. *Proceedings of the 6th European Conf. on CompVision-Part II 2000*; 2: 751-767.
- [38] Stauffer C, Grimson W. Learning Patts of activity using realtime Tracking. In *IEEE Trans. on Patt Anal Mach Intel Aug 2000*; 22: 747-57.
- [39] Haritaoglu I., Harwood D., Davis LS. W4: Who? when? where? what? a real time system for detecting and tracking people in. *Third Face and Gesture Recog Conf. (Apr.) 1998*; 222-227.
- [40] Harville M. A framework for high-level feedback to adaptive, perpixel, mixture-of-gaussian background models. *European Conf. Comp Vision 2002*; 3: 543-560.
- [41] Magee DR. Tracking multiple vehicles using foreground, background, and motion models. In *Proceedings of the Statistical Methods in Video Processing Workshop*, (Copenhagen, Denmark), June 2002; 7-12.
- [42] Cucchiara R, Grana C, Piccardi M, Prati A. Detecting moving objects, ghosts and shadows in video streams. *IEEE Trans. Patt Anal Mach Intell, PAMI 2003*; 25(10): 1337-1342.
- [43] Cavallaro A, Ebrahimi T. Change detection based on color edges, circuits and systems. *The 2001 IEEE Int Symposium*, May 2001; 2: 141-144.
- [44] Costantini R., Ramponi G., Bracamonte J., et al. Countering illumination variations in a video urveillance environment. In *Proc. of SPIE, Electronic Imaging Conf.*, San Jose, USA, 2001; 4304: 85-97.
- [45] Francois AR, Medioni GG. Adaptive color background modeling for real-time segmentation of video streams. In *Proceedings of the Wireless Sensor Networks Recent Patents on CompScience*, 2008. *Int Imag Sci Syst Technol*, Las Vegas, NV, USA, (June) 1999; 1(121): 227-232.
- [46] Hall E L. *CompImage Processing and Recog*. New York: Academic, 1979.
- [47] Boul T., Ali Erkin A., Lewis P, et al. Frame-rate omnidirectional surveillance and tracking of camuaged and occluded targets. In

- Proceedings *Second IEEE Workshop on Visual Surveillance* June 1999; 48-55. (Fort Collins, CO).
- [48] Pless R., Larson J., Siebers S., Westover B. Evaluation of local models of dynamic background. In *Proceedings IEEE Conf Comp Vision Patt Recog*, 2003; 2: 73-78, (Madison, WI).
- [49] Aubert D. Passengers queue measurement. In *Proc. of 10th Int. Conf. on Image Anal Process*, Venice, Italy, 1999; 1132-1135.
- [50] Cavallaro A, Ebrahimi T. Classification of change detection algorithms for object-based applications. *Proc. of Workshop on Image Analysis For Multimedia Interactive Services (WIAMIS-2003)* April 2003; London (UK): 2003; 9-11.
- [51] Eveland C, Konolige K, Bolles RC. Background Modeling for Segmentation of Video-Rate Stereo Sequences. *CVPR, 1998 IEEE Comp Society Conf. on Comp Vision and Patt Recog (CVPR'98)* 1998; 266-271.
- [52] Ivanov Y, Bobick A, Liu J. Fast lighting independent background subtraction. *Int. J Comp Vision* 2000; 37(2): 199-207.
- [53] Gordon G., Darrell T., Harville M., Woodfill J. Background Estimation and Removal Based on Range and Color. *CVPR, 1999 IEEE Comp Society Conf. on Comp Vision and Patt Recog (CVPR'99)* 1999; 2: 2459.
- [54] Konolige K. Small vision systems: hardware and implementation. *Proc. ISRR*, Hayama, 1997.
- [55] Javed O., Shafique K., Shah M. A hierarchical approach to robust background subtraction using color and gradient information. motion. *Workshop on Motion and Video Computing (MOTION' 02)* 2002; 22-27.
- [56] Cristani M., Bicego M., Murino V. Multi-level background initialization using Hidden Markov Models. In *First ACM SIGMM Int. workshop on Video surveillance* 2003; 11-20.
- [57] Scott DW. Multivariate Density Estimation. Theory, Practice and Visualization. John Wiley & Sons, Inc., 1992.
- [58] Zhong J., Sclaroff S. Segmenting Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter. *ICCV, Ninth IEEE Int. Conf. on Comp Vision (ICCV'03)* 2003; 1: 44-50.
- [59] Wang D., Feng T., Shum H., Ma S. A novel probability model for background maintenance and subtraction. *The 15th Int. Conf. on Vision Interface* 2002; 109-117.
- [60] Butler D., Sridharan S., Bove VMJr. Real-time Adaptive Background Segmentation. Acoustics, Speech, and Signal Processing. 2003. *Proceedings. (ICASSP '03)*. 2003 *IEEE Int. Conf. on April* 2003; 3: 349-52.
- [61] Kim K, Chalidabhongse TH, Harwood D, Davis LS. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging* 2005; 11(3): 172-185.
- [62] Kim K. Algorithms and evaluation for object detection and tracking in comp vision, PhD Thesis, University of Maryland, College Park, 2005.
- [63] Kohonen T. Learning vector quantization, Neural Networks. Edition 1, 1988; 3-16.
- [64] Ripley BD. Patt Recog and neural networks. Cambridge: Cambridge University Press; 1996.
- [65] Mittal A., Paragios N. Motion-based background subtraction using adaptive kernel density estimation. In *Proceedings of the Int. Conf. Comp Vision and Patt Recog (CVPR)* 2004; 302-309.
- [66] Huwer S., Niemann H. Adaptive Change Detection for Real-Time Surveillance Applications. vs, *Third IEEE Int. Workshop on Visual Surveillance (VS'2000)*, 2000; 37-45.
- [67] Stauffer C., Grimson WEL. Adaptive Background Mixture Models for Real-Time Tracking. *CVPR, 1999 IEEE Comput Society Conf. on CompVision and Patt Recog (CVPR'99)* 1999; 2: 246-252.
- [68] Koller D., Weber J., Huang T., Malik J., Ogasawara G., Rao B. Russell S. Toward robust automatic traffic scene analysis in real-time. in *Proc. Int. Conf. Patt Recog*, 1994; 126-131.
- [69] Gloyer B, Aghajan H, Siu K-Y, Kailath T. Video-based freeway monitoring system using recursive vehicle tracking. In *Proceedings of SPIE* Feb 1995; 2421:173-180.
- [70] Lo B., Velastin S. Automatic congestion detection system for underground platforms. In *Proceedings of 2001 Int. Symposium on Intelligent Multimedia, Video, and Speech Processing*, (Hong Kong) May 2001; 158-161.
- [71] Zhou Q., Aggarwal J. Tracking and classifying moving objects from videos. In *Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance* 2001.
- [72] Boulton T, Micheals R, Gao X, Eckmann M. Into the woods: Visual surveillance of non-cooperative camouflaged targets in complex outdoor settings. In *Proceedings of the IEEE*, October 2001; 1382-1402.
- [73] Karmann K-P, Brandt A. Moving object Recog using and adaptive background memory. In *Time-Varying Image Processing and Moving Object Recog. V*. Cappellini, Ed. 2, Elsevier Science Publishers B.V. 1990; 289-307.
- [74] Karmann K-P, Brandt AV, Gerl R. Moving object segmentation based on adaptive reference images. In *Signal Processing V: Theories and Application*. Amsterdam. The Netherlands: Elsevier, 1990.
- [75] Monnet A., Mittal A., Paragios N., Ramesh V. Background modeling and subtraction of dynamic scenes. *Proc Int. Conf. Comput. Vision*, Nice, France, 2003; 2:1305-1312.
- [76] Jolliffe IT. Principal Component Analysis. Springer-Verlag, 1986.
- [77] de la Torre F., Black MJ. Robust principal component analysis for Compvision. In *ICCV*, Vancouver, Canada, July 2001; I: 362-369.
- [78] Golub GH, Van Loan CF. Matrix Computations. John Hopkins University Press, 1996.
- [79] Scott DW. Multivariate Density Estimation. New York: Wiley-Interscience, 1992.
- [80] Duda RO, Stork DG, Hart PE. Patt Classification. New York: Wiley, 2000.
- [81] Elgammal A, Duraiswami R, Davis LS. Efficient Non-Parametric Adaptive Color Modeling Using Fast Gauss Transform. *CVPR, IEEE Comp ociety Conf. on CompVision and Patt Recog (CVPR'01)* 2001; 2: 563-570.
- [82] Lambert C, Harrington S, Harvey C, Glodjo A. Efficient on-line nonparametric kernel density estimation. *Aorithmica* 1999; 25: 37-57.
- [83] Wang H., Suter D. Background Initialization with A New Robust Statistical Approach. *IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. 2005; 153-159.
- [84] Elgammal A, Duraiswami R, Davis LS. Efficient Kernel Density Estimation Using the Fast Gauss Transform with Applications to Color Modeling and Tracking. *Patt Analysis and Machine Intelligence*. IEEE Transactions 2003; 25(11): 1499- 1504.
- [85] McFarlane N, Schofield C. Segmentation and tracking of piglets in images, *Machine Vision Appl* 1995; 8(3): 187-193.
- [86] Remagnino P., Baumberg A. Grove T, et al. An Ingrated traffic and pedestrian modelbased vision system. *Proceedings of the Eighth British Machine ATED traffic and pedestrian Vision Conf*. 1997; 380-389.
- [87] Tuzel O, Porikli F, Meer P. A Bayesian Approach to Background Modeling. *IEEE Workshop on Machine Vision for Intelligent Vehicles (MVIV)* 2005; 3: 58.
- [88] Jabri S., Duric Z., Wechsler H., Rosenfeld A. Detection and Location of People in Video Images Using Adaptive Fusion of Color and Edge Information. *ICPR, 15th Int. Conf. on Patt Recog (ICPR'00)* 2000; 4: 4627-4630.
- [89] Horprasert T., Harwood D., Davis LS. A statistical approach for realtime robust background subtraction and shadow detection. *IEEE Frame-Rate Applications Workshop*, Kerkyra, Greece, 1999.
- [90] Ridder C., Munkelt O., Kirchner H. Adaptive Background Estimation and Foreground Detection using Kalman-Filtering. *Proces of Int. Conf. on recent Advances in Mechatronics. ICRAM'95*, UNESCO Chair on Mechatronics 1995; 193-199.
- [91] Koller D., Weber J., Huang T., Malik J., Ogasawara G., Rao B., Russel S. Towards robust automatic traffic scene analysis in real-time. In *Proc. of the Int. Conf. on Patt Recog*, Israel, November 1994.
- [92] Heikkila J., Silven O. A real-time system for monitoring of cyclists and pedestrians in: *Second IEEE Workshop on Visual Surveillance Fort Collins*, Colorado Jun. 1999; 74-81.
- [93] Halevy G, Weinshall D. Motion of disturbances: detection and tracking of multibody non-rigid motion. *Mach Vision Applications* 1999; 11: 122-137.
- [94] Koller D, Weber J, Malik J. Robust multiple car tracking with occlusion reasoning. Tech. Rep. UCB/CSD- 93-780, EECS Department, University of California, Berkeley, Oct 1993.
- [95] Soatto S, Doretto G, Wu YN. Dynamic textures. In *ICCV*, Vancouver, Canada July 2001; II: 439-446
- [96] Doretto G, Chiuso A, Wu YN, Soatto S. Dynamic textures. *IJCV* February 2003; 51(2): 91-109.

- [97] Grimson WEL, Stauffer C, Romano R, Lee L. Using adaptive tracking to classify and monitor activities in a site in: *Comp Vision and Patt Recog Santa Barbara, California, Jun. 1998*; 1-8.
- [98] Friedman N., Russell S. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth Annual Conf. on (UAI-1997)*, San Francisco, CA, Morgan Kaufmann Publishers. 1997; 175-181.
- [99] Lee D-S., Hull J., Erol B. A Bayesian framework for Gaussian mixture background modeling. In *Proceedings of IEEE Int. Conf. on Image Processing*, (Barcelona, Spain), Sept 2003; 3: 973-976.
- [100] Rittscher J., Kato J., Joga S., Blake A. A probabilistic background model for tracking. In *Proc. 6th Eur. Conf. CompVision*, 2000; 2: 336-350.
- [101] Stenger B., Ramesh V., Paragios N., Coetzee F., Buhmann J. M. Topology Free Hidden Markov Models: Application to Background Modeling. *ICCV, Eighth Int. Conf. on Computer Vision (ICCV'01) 2001*; 1: 294-301.
- [102] Montacie C, Caraty M-J, Barras C. Mixture Splitting Technic and Temporal Control in a HMM-Based Recog System. *Proc. Intl. Conf. on Spoken Language Processing (ICSLP) 1996*; 2: 977-980.
- [103] Ostendorf M, Singer H. HMM topology design using maximum likelihood successive state splitting. *Comp Speech Lang 1997*; 11(1): 17-41.
- [104] Brand M, Kettner V. Discovery and segmentation of activities I video. *IEEE Transactions on Patt Anal Mach Intel 22*: 844-851.
- [105] Brand M. An Entropic Estimator for Structure Discovery, 17 mitsubishi electric research labs. Technical report TR-98-19, August 1998.
- [106] Golden, R.M., Michale, A., Earwood, J.W.: US20077188064 (2007).
- [107] Rabiner LR. A Tutorial on hidden markov models and selected applications in speech recog. *Proceedings of the IEEE 1989*; 77(2): 257-286.
- [108] Neal RM, Hinton GE. A view of the em algorithm that justifies incremental, sparse, and other variants. In M.I. Jordan (Ed.), *In Learning in Graphical Models.*, Kluwer Academic Publishers 1998; 355-368.
- [109] Nowlan SJ. Soft competitive adaptation: neural network learning algorithms based on fitting statistical mixtures, Ph. D. thesis. School of Comp Sience. Carnegie Mellon University, Pittsburgh, 1991.
- [110] Horprasert T, Harwood D., Davis LS. A statistical approach for real-time robust background subtraction and shadow detection. *Proc. IEEE Int'l Conf. CompVision, Frame Rate Workshop 1999*; 1-19.
- [111] Long W, Yang YH. Stationary background generation: An alternative to the difference of two images, *Patt Recog 1990*; 23(12): 1351-1359.
- [112] Gloyer B., Aghajan HK, Siu KY, Kailath T. Video-based freeway monitoring system using recursive vehicle tracking. In *Proc. of IS&T-SPIE Symposium on Electronic Imaging: Image and Video Processing 1995*.
- [113] Gutches D, Trajkovic M, Cohen-Solal E, Lyons D, Jain AK. A background model initialization algorithm for video surveillance. *ICCV, Eighth Int. Conf. on CompVision (ICCV'01) 2001*; 1: 733-740.
- [114] White, C.A., Hirson, D., Poon, S., Mehi, J.: WO07058895 (2007).
- [115] Dawson-Howe K. Active surveillance using dynamic background subtraction. Tech. Rep. TCD-CS-, Trinity College, 1996; 96-06.
- [116] Kurita T, Shimai H, Umeyama S, Shigehara T. Estimation of Background from Image Sequence with Moving Objects, <http://coblitz.codeen.org:3125/citeseer.ist.psu.edu/cache/papers/cs/2945/http:zSzzSzwww.etl.go.jpzSzetlzSzsuzSzskuritzSzpaperszSzicip99.pdf/estimation-of-background-from.pdf>
- [117] Huber PJ. *Robust Statistics*, John Wiley & Sons, 1981.
- [118] Rousseeuw RJ, Leroy AM. *Robust Regression and Outlier Detection*, John Wiley & Sons, 1986.
- [119] Redner AR, Walker HF. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review 26*:195-239.
- [120] Priebe CE. Adaptive mixture density estimation. *J Amran Stat Asso 1989* :796-806.
- [121] Digalakis VV, Neumeyer LG. Speaker Adaptation using combined transformation and Bayesian methods, *IEEE Trans. Speech Audio Proces 1995*; 3: 357-366.
- [122] Huo Q, Lee CH. Online adaptive learning of the correlated continuous density hidden markov models for speech recog. *IEEE Trans. Speech Audio Processing 1998*; 6: 386-397.
- [123] Nefian, A. V.: US20077171043 (2007).
- [124] Wang H., Suter DA. Novel robust statistical method for background initialization and visual surveillance. *Asian Conf. on CompVision 2006*.
- [125] Fischler MA, Rolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 1981*; 24 (6): 381-395.
- [126] Nascimento J., Marques JS. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia 2005*.
- [127] Horn B. *Robot V*. The MIT Press 1986.
- [128] Fuentes L., Velastin S. From tracking to advanced surveillance. In *Proceedings of IEEE Int. Conf on Image Processing*, (Barcelona, Spain), Sept 2003.
- [129] Migdal J., Eric W., Grimson L. Background Subtraction Using Markov Thresholds, wacv-motion. *IEEE Workshop on Motion and Video Computing (WACV/MOTION'05) 2005*; 2: 58-65.
- [130] Geman S, Geman D. Stochastic relaxation, Gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Patt Analysis and Machine Intelligence*, 1984; 6(6): 721-741.
- [131] Jain R., Kasturi R., Schunk GB. *Machine Vision*. McGRAWHILL Int. Editions, 1995.
- [132] Durucan E, Ebrahimi T. Robust and illumination invariant change detection based on linear dependence for surveillance applications. In *Proc. of X EUSIPCO*, Tampere, Finland, 2000; 1041-1044.
- [133] Jain R, Nagel H. On the analysis of accumulative difference pictures from image sequences of real world scenes, *IEEE Transactions on Patt Analysis and Machine Intelligence 1979*; 1: 206-214.
- [134] Aach T, Kaup A. Bayesian algorithms for change detection in image sequences using Markov random fields. *Signal Processing: Image Comm 1995*; 7(2): 147-160.
- [135] Rosin P. Thresholding for change detection. *CompVision and Image Understanding May 2002*; 86(2): 79-95.
- [136] Rosin P, Ioannidis E. Evaluation of global image thresholding for change detection. *Patt Recog Letters October 2003*; 24(14): 2345-2356.
- [137] Smits P, Annoni A. Toward specification-driven change detection. *IEEE Trans. Geoscience and Remote Sensing May 2000*; 38(3): 1484-1488.
- [138] Poor H. V. *An introduction to signal detection and estimation*. 2nd ed. Springer-Verlag, 1994.
- [139] Sirtkaya S. Moving object detection in 2d and 3d scenes, Master Thesis, Electrical and electronic engineering, Graduate school of natural and applied science, Middle east technical university, Baltgat, Ankara, Turkey, September 2004.
- [140] Sankur B, Sezgin M. A survey over Image Thresholding Techniques and Quantitative Performance Evaluation. *J Elect Imag 2004*; 13: 146-165.
- [141] Kapur JN, Sahoo PK, Wong AKC. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision. Graphics Image Proc 1985*; 29: 273-285.
- [142] Correia P., Pereira F. Standalone objective evaluation of segmentation quality. *WIAMIS 2001 - Workshop on Image Analysis for Multimedia Interactive Services*, Tampere, Finland 2001.
- [143] Correia PL, Pereira F. Objective evaluation of video segmentation Quality. *IEEE Trans Image Proc 2003*; 12(2): 186-200.
- [144] ITU-R. Methodology for the subjective assessment of the quality of television pictures. Recommendation BT.500-7, 1995.
- [145] ITU-T. Recommendation - Subjective video quality assessment methods for multimedia applications August 1996; 910.
- [146] Correia P., Pereira F. Estimation of video object's relevance. In *EUSIPCO'2000-X European Signal Processing Conf.*, Tampere, Finland 2000.
- [147] Trees HV. *Detection, estimation, and modulation theory*. John Wiley and Sons, 2001.
- [148] Chalidabhongse TH., Kim K., Harwood D., Davis LS. A perturbation method for evaluating background subtraction algorithms. *Proceedings of the Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS 2003)* Nice, France, 2003.

- [149] Gao X., Boulton TE., Coetzee F., Ramesh V. Error analysis of background adaptation. *Comp Vision and Patt Recog. 2000. Proceedings. IEEE Conf 2000*; 1: 503-510.
- [150] Oberti F., Teschioni A., Regazzoni CS. Roc curves for performance evaluation of video sequences processing systems for surveillance applications. *In IEEE Int. Conf. on Image Processing 1999*; 2: 949-953.
- [151] Black J, Ellis T. Rosin P. A novel method for video tracking performance evaluation. *In Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, Nice, France, 2003; 125-132.
- [152] Correia P., Pereira F. Objective evaluation of relative segmentation quality. *In Int. Conf. on Image Processing 2000*; 308-311.
- [153] Erdem CE, Sankur B, Tekalp AM. Performance measures for video object segmentation and tracking. *IEEE Trans. Image Processing 2004*; 13(7): 937-951.
- [154] Toth D., Aach T., Metzler V. Illumination-invariant change detection, ssai. *4th IEEE Southwest Symposium on Image Analysis and Interpretation 2000*; 3-7.
- [155] Living, J.: GB2431805 (2007).
- [156] Hu J., Kahsi R., Lopresti D., Nagy G. and Wilfong G. Why table ground-truthing is hard. *In Proc. Sixth Int. Conf. Document Analysis and Recog 2001*; 129-133.
- [157] Levine M, Nazif A. Dynamic measurement of compgenerated image segmentations. *IEEE Trans. Patt Anal. Machine Intell. 1985*; 7: 155-164.
- [158] Kitchen L, Rosenfeld A. Edge evaluation using local edge coherence. *IEEE Trans. Systems Man Cybernet 1981*; 11, 597-605.
- [159] Venkatesh S, Rosin P. Dynamic threshold determination by local and global edge evaluation. *Comput. Vision Graphics Image Process 1995*; 57(2): 146-160.
- [160] Fitzgibbon A, Pilu M, Fisher R. Direct least square fitting of ellipses. *IEEE Trans 1999*; .21(5): 476-480.
- [161] Venkatesh S, Kitchen L. Edge evaluation using necessary components. *CVGIP 1992*; 54(1): 23-30.
- [162] Zheng Z, Wang H, Teoh E. Analysis of gray level corner detection. *Patt Recog Lett 1999*; 20(2): 149-162.
- [163] Barron J, Fleet D, Beauchemin S. Performance of optical flow techniques. *Int J CompVision 1994*; 12(1): 43-77.
- [164] Tan IL, van Schijndel R, Fazekas F, et al. Image registration and subtraction to detect active T2 lesions in MS: An interobserver study. *J. Neurol 2002*; 249: 767-773.
- [165] van Rijsbergen CJ. Information retrieval. Butterworth & Co (Publishers) Ltd, Second Ed. 1979.
- [166] Wolf S, Webster A. Subjective and objective measures of scene criticality. In ITU Experts Meeting on Subjective and Objective Audiovisual Quality Assessment Methods, Turin, Italy, Oct. 1997.
- [167] Living, J.: GB2431799 (2007).
- [168] Abdou I, Pratt W. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proc. IEEE May 1979*; 67: 753-763.
- [169] Fram J, Deutsch E. On the quantitative evaluation of edge detection schemes and their comparison with human performance. *IEEE Trans. Comput. June 1975*; 24(6): 616-628.
- [170] Bryant D., Bouldin D. Evaluation of edge operators using relative and absolute grading. *In Proc. IEEE Conf. Patt Recog Image*, Chicago, IL 1979; 138-145.
- [171] Wollborn M., Mech R., Refined procedure for objective evaluation of video object generation algorithms Doc. ISO/IEC JTC1/SC29/WG11 M3448, March 1998.
- [172] Villegas P, Marichal X, Salcedo A. Objective evaluation of segmentation masks in video sequences. *WIAMIS'99*, Germany 1999; 85-88.
- [173] Strasters K, Gebrands J. Three-dimensional image segmentation using a split, merge and group approach. *Patt Recognit Lett 1991*; 12: 307-325.
- [174] John, W.R., David, H.D.: EP1763154 (2007).
- [175] Nascimento JC, Marques JS. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on multimedia August 2006*; 8:4.
- [176] Jain AK, Duin RPW, Mao J. Statistical patt recog: A review. *IEEE Trans Patt Anal Mach Intel Jan 2000*; 22: 1.
- [177] Bowyer K, Kranenburg C, Dougherty S. Edge detector evaluation using empirical ROC curves. *In IEEE CompSociety Conf. on CompVision and Patt Recog 1999*; 1.