

Chapter 2

Motion Detection in Static Backgrounds

Abstract Motion detection plays a fundamental role in any object tracking or video surveillance algorithm, to the extent that nearly all such algorithms start with motion detection. Actually, the reliability with which potential foreground objects in movement can be identified, directly impacts on the efficiency and performance level achievable by subsequent processing stages of tracking and/or object recognition. However, detecting regions of change in images of the same scene is not a straightforward task since it does not only depend on the features of the foreground elements, but also on the characteristics of the background such as, for instance, the presence of vacillating elements. So, in this chapter, we have focused on the motion detection problem in the basic case, i.e., when all background elements are motionless. The goal is to solve different issues referred to the use of different imaging sensors, the adaptation to different environments, different motion speed, the shape changes of the targets, or some uncontrolled dynamic factors such as, for instance, gradual/sudden illumination changes. So, first, a brief overview of previous related approaches is presented by analyzing factors which can make the system fail. Then, we propose a motion segmentation algorithm that successfully deals with all the arisen problems. Finally, performance evaluation, analysis, and discussion are carried out.

Keywords Motion detection • Background subtraction • Visual surveillance • Image segmentation • Computer vision

2.1 State of the Art

Motion detection plays a fundamental role in any object tracking or video surveillance algorithm, to the extent that nearly all such algorithms start with motion detection. Actually, the reliability with which potential foreground objects in movement can be identified, directly impacts on the efficiency and performance level achievable

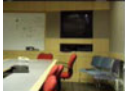

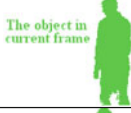



Case	Reference Frame	Current Frame	Background Subtraction Result
Ideal			 The object in current frame
General			 The object in reference frame (ghosting) The object in current frame

Fig. 2.1 Background subtraction results by depending on foreground presence/absence in the reference frame when a background(-frame) subtraction technique is used

by subsequent processing stages of tracking and/or recognition. However, detecting regions of change in images of the same scene is not a straightforward task since it does not only depend on the features of the foreground elements, but also on the characteristics of the background such as, for instance, the presence of vacillating elements. In this chapter we will study the motion detection on static scenes, that is, the only elements in movement will be the targets. In that way, it is possible to analyze and solve issues relative to the use of different imaging sensors, the adaptation to different environments, and to some dynamic, uncontrolled factors such as (gradual or global) changes in illumination.

From this starting point, any detected changed pixel will be considered as part of a foreground object. For that reason, techniques based on temporal information by using a thresholded frame difference could be fitted. By depending on the temporal relationship between frames implied in the difference, two different approaches can be defined. On the one hand, *background(-frame) subtraction* uses a reference frame to represent the scene background. That frame is usually set to the first captured image. Thus, a pixel is classified as foreground if its current value is considerably different from its value in the reference frame. Although it could seem the perfect solution, it is worth noting that two different situations can take place in real environments (see Fig. 2.1):

1. *Ideal situation.* There are no foreground objects in the reference frame. In this case, the resulting image would be the same as the desired segmentation result
2. *General situation.* Foreground objects may appear in the reference frame. Their presence makes background subtraction fail by providing false positives due to their position in the reference frame.

On the other hand, *techniques based on temporally adjacent frames* could be considered. Basically, this time-differencing approach suggests that a pixel is moving if its intensity has significantly changed between the current frame and the previous one. That is, a pixel x belongs to a moving object if

$$|I_t(x) - I_{t-1}(x)| < \tau \quad (2.1)$$

Fig. 2.2 Drawbacks of adjacent frame difference approach

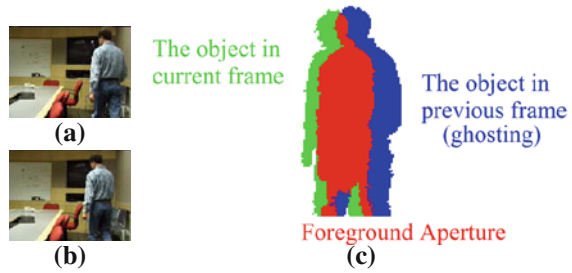
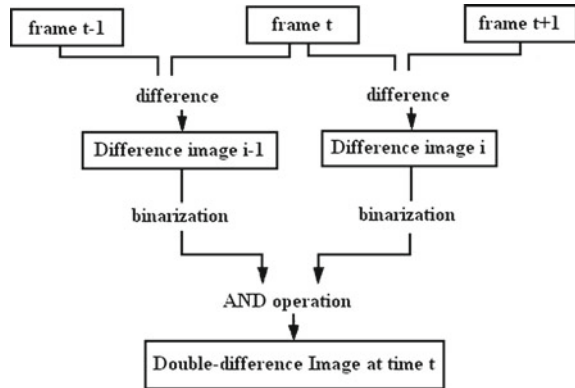


Fig. 2.3 Double-difference image generation [2]



where $I_t(x)$ represents the intensity value at pixel position x at time t and τ corresponds to a threshold describing a significant intensity change.

Nevertheless, in spite of the fact that this method provides an easy, fast moving object detection, it only works on particular conditions of object's speed and frame rate because they generate its two well-known difference drawbacks [1]: *ghosting* and *foreground aperture*. So, as depicted in Fig. 2.2, the presence of an object in the previous frame generates false alarms (*ghosting*), while the similarity between pixels when object's speed is too low or it becomes motionless, generates holes in the segmentation result (*foreground aperture*).

Thus, as a solution, Kameda and Minoh [2] proposed a variation of this method: a *double-difference* image. This approach operates a thresholded difference between frames at time t and $t - 1$ and between frames at time t and $t + 1$, by combining them with a logical **AND** (see Fig. 2.3). However, the object's position is not estimated in real time, an accurate motion detection is not allowed if the moving objects have no enough texture, and the situation in which targets become motionless is not considered.

In the *VSAMproject*, Collins et al. [1] described a different hybrid algorithm for motion detection. Basically, a three-frame differencing operation, based on image difference between frames at time t and $t - 1$ and the difference between t and $t - 2$, is performed to determine regions of legitimate motion and to erase *ghosting*

problem. Then, an adaptive background(-frame) subtraction, proposed by Kanade et al. [3], was used to solve the *foreground aperture* problem. Nevertheless, although the proposed algorithm solves issues of image difference and gives good results in motion detection, the background update procedure fails when objects begin or end their motion and/or there are luminance variations in the scene. Moreover, it suffers a few drawbacks on variable depth shots since it was widely used in outdoor environments with a low depth of field images.

With the aim of solving these problems, many techniques for a proper background update have been developed. The simplest ones update the background by a convex composition of background pixels a time $t - 1$ and those at time t such that the update weight for background pixels is eventually variable with pixel classification (light weight for background pixels and heavy weight for foreground pixels).

On the contrary, Migliore et al. [4] claimed that it is possible to obtain a robust pixel foreground classification without the need of previous background learning. For that, they exploited a joint background subtraction and frame-by-frame difference to properly classify pixels (see Algorithm 1). Then, the background model is selectively updated according to such classification as pointed out by Wren et al. [5], by using the following formula:

$$B_t = (1 - \alpha)B_{t-1} + \alpha F_t \quad (2.2)$$

where the α value is different depending on pixel classification. So, it is set to 0 if the pixel is classified as foreground by avoiding the background corruption; a low, non-zero value is used to slowly update the model; and, finally, in the case when any background element starts moving, a high α will allow to quickly restore the background model.

Algorithm 1 Joint Difference Algorithm [4]

```

if ( $(|F_t(x) - B_{t-1}(x)| > \tau_B)$  AND ( $|F_t(x) - F_{t-1}(x)| > \tau_A$ )) then
  Foreground Pixel;
else if ( $(|F_t(x) - B_{t-1}(x)| > \tau_B)$  AND ( $|F_t(x) - F_{t-1}(x)| < \tau_A$ )) then
  Collect pixels in blobs;
  if ( $\# \text{ Foreground Pixels} \geq (\gamma * (\# \text{ Total Pixels}))$ ) then
    Foreground Pixel; //foreground aperture problem solution
  else
    Background Pixel; //a background object suddenly starts moving at time  $t$ 
  end if
else if ( $(|F_t(x) - B_{t-1}(x)| < \tau_B)$  AND ( $|F_t(x) - F_{t-1}(x)| > \tau_A$ )) then
  Background Pixel; //ghosting problem solution
else
  //  $|F_t(x) - B_{t-1}(x)| < \tau_B$  AND  $|F_t(x) - F_{t-1}(x)| < \tau_A$ ;
  Background Pixel;
end if

```

However, despite its good performance, it has two important handicaps. On the one hand, this method fails when a target stops or their speed is low. On the other hand, given that difference thresholds are established for the whole image, various

factors, such as nonstationary and correlated noise, ambient illumination, inadequate contrast, and/or an object's size not commensurate with the scene, can make the approach fail.

2.2 Combination of Difference Approach

Our contribution at this stage is to provide a real-time algorithm for robust motion detection. For that, a combination of difference (CoD) techniques is proposed since it was proven that they provide a good performance. As presented by Migliore et al. [4], it is possible to overcome *adjacent difference* problems by using a *background(-frame) subtraction*.

The first issue to be solved is how to properly choose the threshold value because it is a key parameter in the segmentation process since it can affect quite critically the performance of successive steps. Although users can manually set a threshold value, it is not a valid solution when autonomous systems are designed. In this context, a common solution is to use a thresholding algorithm that automatically computes that value.

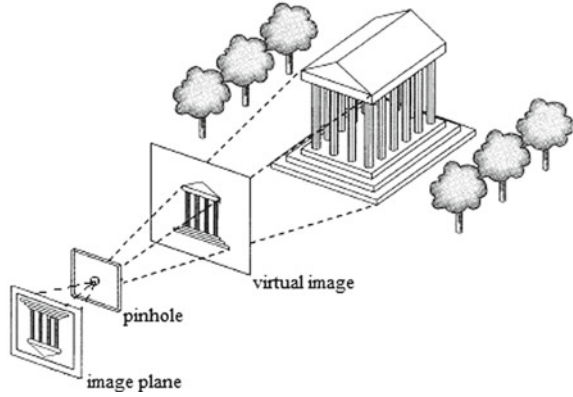
Sezgin and Sankur [6] categorized automatic thresholding methods according to the information they are exploiting, in:

- *Histogram shape-based methods*, where, for example, the peaks, valleys, and curvatures of the smoothed histogram are analyzed
- *Clustering-based methods* divide the gray-level samples into two parts (background and foreground), or alternately are modeled as a mixture of Gaussians
- *Entropy-based methods* result in algorithms that use the entropy of the foreground and background regions, the cross-entropy between the original and binarized image, etc.
- *Object attribute-based methods* search a similarity measurement between the gray-level and the binarized images such as fuzzy shape similarity, edge coincidence, etc.
- *The spatial methods* use higher order probability distributions and/or correlation between pixels
- *Local methods* adapt the threshold value on each pixel to the local image characteristics.

Despite the wide variety of possibilities, the existing methods only work well when the images to be thresholded satisfy their assumptions about the distribution of the gray-level values over the image. So, situations such as shape deformations of the interest object, the relationship of the foreground object's size with respect to the background, or overlapping of background and target gray-level distributions, make them fail. For all that, it was necessary to design a new way to automatically obtain the threshold value.

Our contribution at this point is an adaptive *dynamic* thresholding method such that it is capable of adapting to non-uniform-distributed resolution, inadequate illu-

Fig. 2.4 Perspective projection model



mination gradient in the scene, shadows, and gradual as well as sudden changes in illumination. The main idea is to divide each captured image in regions such that a threshold is obtained for each described area. A key issue is the way regions are defined since it is resolution-dependent and, therefore, camera-dependent. In this manuscript, two different kind of cameras are considered:

1. *Perspective cameras*, often referred as *pinhole cameras*, are optical imaging devices which follow the perspective projection model in order to obtain an image (see Fig. 2.4). Basically, the beams of light bouncing off an object are redirected by a lens to the image plane as if it was a rectilinear propagation of light through a small hole
2. *Fisheye cameras*, on the contrary, are imaging systems combining a fisheye lens with a conventional camera. They are usually used to present a small display of a large structure. For that, they use a lens with a wide field of view (*fisheye lens*) that allows them to take a hemispherical image. Their main advantages with respect to the catadioptric sensors (i.e., the combination of a conventional camera and mirrors) are, first, that they do not exhibit a dead area, and, second, a fisheye lens does not increase the size and the weakness (in the sense of the complete scene is visible, without loss of information due to dead areas) of the imaging system with respect to a conventional camera. In this case, as shown in Fig. 2.5, the projection model consists of a projection onto a virtual unitary sphere, followed by a perspective projection onto an image plane.

Therefore, the image generated in both cases is different as depicted in Fig. 2.6. Perspective cameras obtained a rectangular image that, in most practical situations, accurately satisfies the extrinsic perspective assumptions. So, for each pixel, the set of 3D points projecting to the pixel (i.e., whose possibly-blurred images are centred on the pixel) is a straight line in 3D space, and all the light rays meet at a single 3D point (the optical centre). On the contrary, fisheye cameras capture circular images such that objects close to the focal point are clear and distinguishable to the user, while the level of detail decreases as objects move further away from the point of

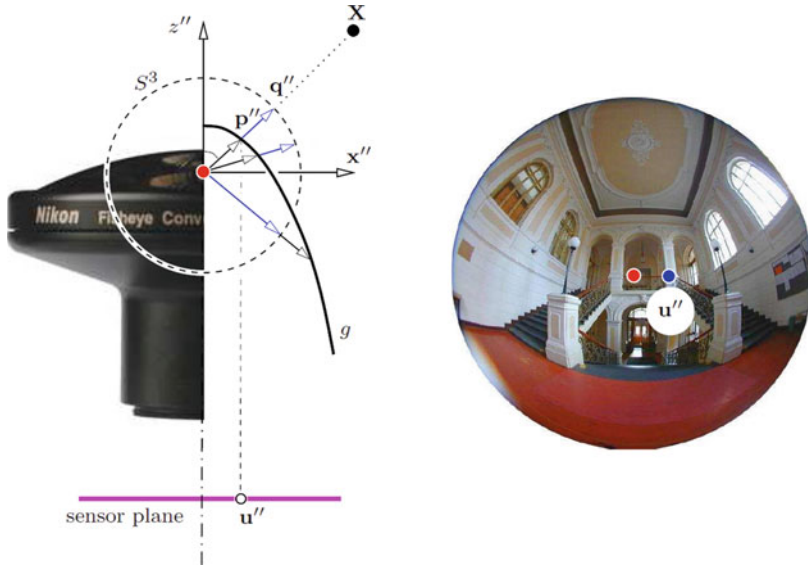


Fig. 2.5 The mapping of a scene point X into a sensor plane to a point u'' for a fisheye lens (courtesy of Mičušík [7])

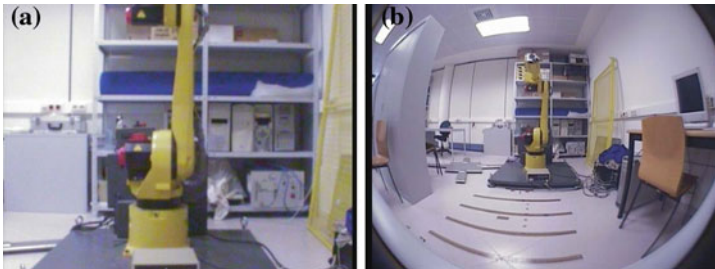


Fig. 2.6 Sample images captured at the same position by a perspective camera (*left*) and a fisheye camera (*right*)

interest. Thus, the fisheye strategy magnifies the area of interest (located in the focal point) to show the detail, whereas the context is maintained by preserving continuity at the periphery. As a consequence, image distribution is not homogeneous in terms of resolution.

The key concept is to divide an image into the proper regions such that the resulting subimages keep the features of the original images, specially in terms of resolution distribution. So, rectangular regions are described for perspective images, where resolution is approximately uniform along the whole image while a fisheye image is divided into sector portions (see Fig. 2.7). Note that circular regions are not used, even though resolution is laid out in that way. It is because a circular region would cover a 180° 3D area and the neighborhood similarity could not be exploited.

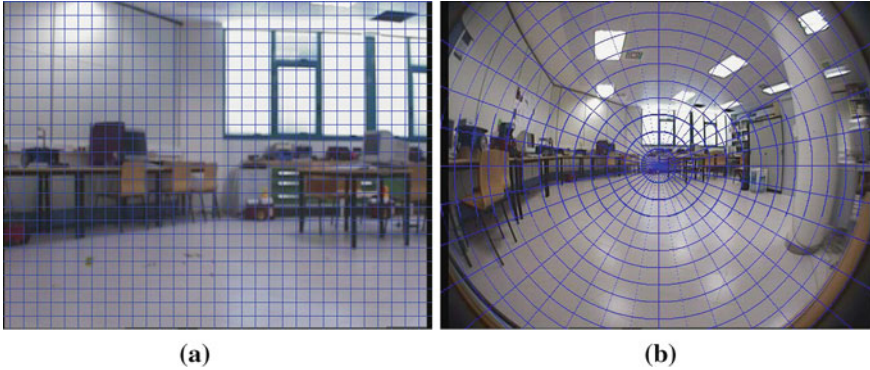


Fig. 2.7 Example of image division depending on the kind of camera used

Once the shape of the image regions is determined, a new issue arises: their size. This parameter is important in terms of noise influence as well as uniformity in illumination and gray level. It mainly depends on the position of the camera with respect to the scene such that when the further a camera is, the smaller regions have to be defined. This is because the size of scene elements is proportional to the distance between the camera and those elements.

Note that each region should be identified by a unique value that allows to properly choose the threshold. In particular, statistic functions are commonly used. The statistic which is the most appropriate, largely depends on the input image, although simple and fast functions include the mean, median, or mean of minimum and maximum values of the local intensity distribution. In our case, we have used the mean value to describe each image region.

The next step is to determine the proper CoD techniques to achieve our goal, i.e., an accurate segmentation. As depicted in Fig. 2.8 and sketched in Algorithm 2, different situations have been studied:

1. *The ideal case.* The reference frame for the *background(-frame) subtraction* is free of foreground objects. Thus, three different situations could be faced by the *adjacent frame differencing*:
 - There are no foreground objects in the previous frame. In this case, pixels are classified as foreground when both the *adjacent difference* and the *background(-frame) subtraction* are greater than or equal to their corresponding thresholds
 - A foreground object appears in the previous frame. So, one or both of the *adjacent difference* drawbacks can occur. The *ghosting* problem is solved by taking into account that the *background(-frame) subtraction* does not identify those pixels as foreground. With regard to the *foreground aperture* drawback, it will be solved as follows. When the foreground objects have a similar texture, they are correctly identify by the *background(-frame) subtraction*, but not by

the *adjacent difference*. The way we have solved this situation is considering what makes a pixel be in this situation. Therefore, on the one hand, they must satisfy a gray-level similarity relationship because they belong to the same homogeneous texture. And, on the other hand, the other requirement refers to those pixels that are classified as a foreground object in the previous frame. So, it is necessary to use both constraints to obtain a successful result since whether the similarity criterion was only used, many false alarms could be generated

- The last case takes place when a foreground object stops moving. Again, this situation has been solved by means of a similarity criterion
2. *The general case.* Any foreground object appears in the reference frame. The difference between this case and the previous one is that an element initially considered as a background element can become foreground at any time. Thus, when it starts moving, it leaves behind a *hole* which will be wrongly classified as foreground. Taking advantage of this knowledge, a new method to detect and solve this situation has been designed. Mainly, it consists of a comparison between the segmentation results of the current frame and the content of that blob in the reference frame. Each time a blob results from a *background* element movement, it has been identified that a *hole* should be removed. So, those pixels are now reclassified as background and background frame is updated with the new information. Again, performance in the different conditions considered in the ideal case was also analyzed by using similar solutions for them (see Fig. 2.8).

Note that color images have been used as input in the examples depicted in Fig. 2.8. However, with the aim of obtaining a general solution that is able to run over both color and gray-level images, a preprocessing takes place. Basically, this preprocessing consists of obtaining a gray-level image composed of the intensity channel in the Hue-Saturation-Intensity (HSI) system. Despite other color spaces are available such as, for instance, *Lab*, *YUV*, *XYZ*, etc. HSI is used because it encodes color information by separating an overall intensity value *I* from two values encoding ‘chromaticity’—hue *H* and saturation *S*—(see Appendix A for further information). This might also provide better support for computer vision algorithms because it can normalize small lighting changes and focus on the two chromaticity parameters that are more associated with the intrinsic character of a surface rather than the source that is lightning it. In addition, with the purpose of reducing lighting influence on the algorithm’s performance, a difference normalization is carried out. Mathematically, it can be expressed as follows:

$$\left| \left(\frac{\sigma_P^2}{\sigma_C^2} * (Ngray_C - \mu_C) \right) - Ngray_P \right| \quad (2.3)$$

where the indexes *C* and *P*, respectively, correspond to current and previous (reference image in case of the *background subtraction*) frame; *Ngray* represents

the gray level for the considered pixel, while σ^2 and μ , respectively, refer to the standard deviation and the average of the image intensities.

In addition, take into account that two consecutive morphological operations are applied on the binary image resulting from the segmentation process in order to suppress small errors in the background/foreground classification method. So, first, a 3×3 erode filter is used to erase isolated points or lines caused by different dynamic factors such as sensor noise, non-uniform attenuation, or blinking of the lights. Then, a foreground region recovery is achieved by means of a 3×3 expand filter. It is specially useful when two different parts of the same interest object appear divided due to capture and/or segmentation errors.

Another important issue is sudden, global changes in illumination. A common way to solve this situation consists of generating an alarm when a considerable part of the image (usually two thirds of the image) has changed, that is, has been classified as foreground. Although it works well in most cases, it fails when that change is due to target's proximity to the camera. This situation has been solved by comparing the amount of foreground pixels detected in the current frame and in the previous one. So, it is assumed that a global illumination has occurred when more than two-thirds of the image are classified as foreground and the amount of foreground pixels detected in the current frame is greater than 1.5 times the amount detected in the previous frame. Note that this kind of illumination change makes necessary to set a new reference frame for the *background(-frame) subtraction* technique. Moreover,

Algorithm 2 Combination of Differences (CoD)

```

for each pixel  $x$  do
  if ( $|F_t(x) - B(x)| \geq \tau_B(x)$ ) then
    if ( $|F_t(x) - F_{t-1}(x)| \geq \tau_A(x)$ ) then
      Foreground Pixel;
    else if ( $(|F_t(x) - F_{t-1}(x)| < \tau_S) \text{ AND } (Foreground(F_{t-1}(x)))$ ) then
      Foreground Pixel; // Foreground Aperture
    else
      Background Pixel;
    end if
  else if ( $(|F_t(x) - F_{t-1}(x)| \geq \tau_A(x)) \text{ AND } (|F_t(x) - B(x)| < \tau_B(x))$ ) then
    Background Pixel; // Ghosting problem
  else
    Background Pixel;
  end if
end for
Collect pixels in blobs;
for each pixel  $x$  do
  if ( $Foreground(F_{t-1}(x))$ ) then
    if (NO ( $|F_t(x) - F_{t-1}(x)| < \tau_S$ )) then
      Pixel Re-classification (From Foreground to Background); //It is a hole
      Update background reference frame
    end if
  end if
end for

```


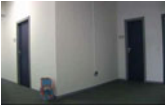


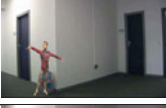








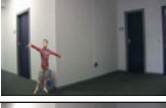

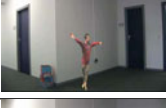
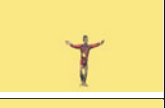


Frame at t	Reference frame	Frame at $t - 1$	Result
			
			
			
			
			
			
			
			

Fig. 2.8 Performance result samples over different situations considered in order to determine the proper combination of difference techniques for an accurate segmentation

some lighting sources require several milliseconds to stabilize. For that reason, when a global illumination change has been detected, the system waits for some frames (typically five frames in our experiments) before resetting all its parameters.

2.3 Experimental Results

In this section, we evaluate the performance of the proposed segmentation procedure. For that, two different kind of experiments have been carried out. First, the CoD's performance is assessed by using the video images provided by three different image

datasets in the literature: the Wallflower Dataset [8], the image dataset developed for the Forth ACM International Workshop on Video Surveillance & Sensor Networks (VSSN06) [9], and the Audiovisual People Dataset, courtesy of Engineering and Physical Sciences Research Council funded MOTINAS project (EP/D033772/1) [10]. Although there exist other datasets such as, for instance, PETS 2006 Benchmark Data [11], to name any, they have not been used here since they aim at a different goal such as the identification of an unattended luggage. Finally, the results over our own dataset composed of both perspective and fisheye images are presented. Note that the qualitative results are displayed as binary images where pixels of interest are coded by the white color, while the background is identified by the black color.

2.3.1 Principles for Performance Evaluation and Comparison

The performance of a motion segmentation technique can be evaluated visually and quantitatively based on the task requirements. So, on the one hand, a qualitative/visual evaluation can be achieved by displaying a flicker animation [12] or a short movie file containing a registered pair of images that are played in fast succession at intervals of about a second each, among others. In that way, in the absence of change, one perceives a steady image, while when changes are present, the changed regions appear to flicker. The estimated change mask can be also superimposed on each image (e.g., as a semitransparent overlay, with different colors for different types of change).

On the other hand, a quantitative evaluation is more challenging. First, because of the difficulty of establishing a valid ground truth, that is, the process of defining the *correct answer* for what *exactly* the algorithm is expected to produce. Arriving at the ground truth is an image analysis that is known to be difficult and time-consuming [13], since it is usually done by human beings and the same human observer can generate different segmentations for the same data at two different times. A secondary issue is to define the relative importance of the different types of errors. There are several standard methods for comparing the ground truth to a candidate binary change mask. The following amounts are generally involved:

- True positives (TP): the number of foreground pixels correctly detected
- False positives (FP): the average of false alarms per frame, i.e., the number of background pixels incorrectly detected as foreground
- True negatives (TN): the number of background pixels correctly detected
- False negatives (FN): the average of false misses, that is, the number of foreground pixels incorrectly detected as background.

From them, Rosin and Ioannidis [14] described three methods for quantifying method's performance:

- *The Percentage Correct Classification (PCC)*, also called *accuracy*, is used as a statistical measurement of how well the segmentation process identifies or excludes foreground pixels. Mathematically, it can be expressed as follows:

$$PCC = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.4)$$

So, an accuracy of 100 % means that the measured values are exactly the same as the given values in the ground truth.

- The Jaccard coefficient (JC) is a statistic used for comparing the similarity and diversity of sample sets and is defined as:

$$JC = \frac{TP}{TP + FP + FN} \quad (2.5)$$

- The Yule coefficient (YC) is a statistic summarizing the extent to which two variables are independent or not, as in the case of the correlation coefficient. It is obtained as follows:

$$YC = \left| \frac{TP}{TP + FP} + \frac{TN}{TN + FN} - 1 \right| \quad (2.6)$$

On the contrary, other authors quantify how well an algorithm matches the ground truth by means of *recall* and *precision* measurements [15, 16]. Thus, *recall* (also known as true positive rate (TPR) or *sensitivity*) [17] is computed as the ratio of the number of foreground pixels correctly identified to the number of foreground pixels in the ground truth; whereas precision or positive predictive value (PPV) is obtained as the ratio of the number of foreground pixels properly identified to the number of foreground pixels detected. That is:

$$Recall = TPR = \frac{\# \text{ of foreground pixels correctly detected}}{\text{total } \# \text{ of ground-truth foreground pixels}} = \frac{TP}{TP + FN} \quad (2.7)$$

$$Precision = PPV = \frac{\# \text{ of foreground pixels correctly detected}}{\text{total } \# \text{ of foreground pixels detected}} = \frac{TP}{TP + FP} \quad (2.8)$$

Other metrics which can be used are:

- False Positive Rate (FPR) which measures background pixels misclassified as foreground such that:

$$FPR = \frac{FP}{(FP + TN)} \quad (2.9)$$

- False Negative Rate (FNR) that refers to foreground pixels erroneously tagged as background. Similar to the previous measurement, it is defined as follows:

$$FNR = \frac{FN}{(FN + TP)} \quad (2.10)$$

- Specificity (SPC) or True Negative Rate (TNR) which expresses the ratio of detected foreground pixels that are true positives. Thus, a specificity of 100 % means that the segmentation process recognizes all actual negatives, that is, 100 % specificity

means no positives are erroneously tagged. In a more formal way:

$$SPC = \frac{TN}{(FP + TN)} = 1 - FPR \quad (2.11)$$

- *Negative Predictive Value (NPV)* that quantifies the ratio of background pixels correctly identified. Its value is obtained as:

$$NPV = \frac{TN}{(TN + FN)} \quad (2.12)$$

- *False Discovery Rate (FDR)* or *False Alarm Rate (FAR)* which measures the foreground pixels misclassified as background:

$$FDR = FAR = \frac{FP}{(FP + TP)} \quad (2.13)$$

- *Mathews Correlation Coefficient (MCC)* that is used as a measurement of the quality of binary classifications. That is, MCC is, in essence, a correlation coefficient between the observed and the predicted binary classifications. Actually, it takes into account true and false positives and is generally regarded as a balanced measurement. Its value oscillates between -1 and 1 such that a coefficient of 1 represents a perfect prediction, 0 an average random prediction, and -1 an inverse prediction. Mathematically, it is defined as follows:

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FN) * (TP + FP) * (FN + TP) * (FN + TN)}} \quad (2.14)$$

Note that if any of the four sums in the denominator is zero, the denominator will be arbitrarily set to 1; this results in a Mathews correlation coefficient of zero, which can be shown to be the correct limiting value

- *F₁ score* which is a measurement of a process' accuracy. It considers both precision and recall of the test to compute the score as follows:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \quad (2.15)$$

The F_1 score can be interpreted as a weighted average of the precision and recall, where an F_1 score reaches its best value at 1 and the worst score at 0

Table 2.1 Parameter values used for evaluating the **CoD** performance over the three considered image datasets

Subimage Size	10x10
Nframes for Initial Background Model	200
Erosion Mask	0 1 0
	0 1 1
	0 0 0
Dilation Mask	1 1 1
	1 1 1
	1 1 1

2.3.2 Experimental Results Over Image Datasets

In this section, three different image datasets are used to evaluate CoD's performance. For that, the set of parameters, listed in Table 2.1, has been the same over all of the considered image sequences.

2.3.2.1 Wallflower Dataset

This dataset was created to evaluate background modeling algorithms from the definition of ten canonical problems that an ideal background maintenance system should overcome:

- *Moved objects*. When a background object is moved, it should not be considered as foreground
- *Time of day*: The passage of time generates gradual illumination changes that alter the background appearance
- *Light switch*. Sudden changes in illumination such as switching on/off lights or opening/closing a window modify the background appearance
- *Bootstrap*. A frame without foreground objects is not available in some environments
- *Foreground Aperture*. When the entire target does not appear as foreground because it is homogeneously colored and the change in the interior pixels cannot be detected
- *Waving trees*. Some background elements can vacillate (e.g., swaying branches, blinking of screens, etc) by requiring models which can represent those disjoint sets of pixel values
- *Camouflage*. Foreground object's pixel characteristics can be subsumed by the modeled background
- *Sleeping person*. The distinction between a foreground object that becomes motionless and a background object that moves and then becomes motionless
- *Waking person*. When an object initially in the background moves, both it and the newly revealed parts of the background appear to change

- *Shadows*. The foreground objects often cast shadows which appear different from the modeled background.

However, only seven real video sequences, with a test image and its corresponding ground truth, are included in this dataset by presenting typical critical situations. All of the test sequences were taken with a 3-CCD camera recording to digital tape at a size of 160×120 pixels, sampled at 4 Hz. Nevertheless, in this section, we have only used those video sequences corresponding to the case under study, that is, those with a static background. Therefore, the used video sequences are: *Time-of-day*, *Light-switch*, *Bootstrap*, and *Foreground-aperture*.

With the aim for evaluating the performance of our approach, it is qualitative and quantitative compared with previous algorithms that have provided results over this dataset. Those approaches can be briefly summarized as follows:

- *Mixture of Gaussians* [18]. A pixel-wise mixture of three Gaussians models the background. Each Gaussian is weighted according to the frequency with which it explains the observed background ($\pm 2\sigma$). The most heavily weighted Gaussians that together explain over 50 % of past data are considered background
- *Normalized Block Correlation* [19]. Images are split into blocks. The pixels in each block are represented as their respective medians over the training images. Each block is represented as its median template and the standard deviation of the block-wise normalized correlation from the median over the training images. For each incoming block, normalized correlation values that deviate too much from the expected deviations cause the block to be considered foreground
- *Temporal Derivative* [20]. In the training phase, for each pixel, the minimum and maximum values are saved along with the maximum interframe change in intensity. Any pixel that deviates from its minimum or maximum by more than the maximum interframe change is considered foreground. They additionally enforced a minimum interframe difference of 10 pixels after the regular training phase
- *Bayesian Decision* [21]. Pixel value probability densities, represented as normalized histograms, are accumulated over time, and backgrounds are determined by a straightforward maximum *a posteriori* criterion
- *Eigenbackground* [22]. Images of motionless backgrounds are collected. Principle Component Analysis (PCA) is used to determine means and variances over the entire sequence (whole images as vectors). So, the incoming images are projected onto the PCA subspace. The differences between the projection and the current image greater than a threshold are considered foreground
- *Wallflower* [23]. Images are processed at three different spatial scales:
 - pixel level, which makes the preliminary classification foreground-background as well as the adaptation to changing backgrounds
 - region level that refines the raw classification of the pixel level based on inter-pixel relationships
 - frame level, designed for dealing with the light switch problem
- *Tracey LAB LP* [24]. The background is represented by a set of codebook vectors locally modeling the background intensities in the spatial-range domain. Thus,

the image pixels not fitting that set are classified as foreground. In addition, as in the *Wallflower* algorithm, a frame-level analysis is used to discriminate between global light changes, noise, and objects of interest. Moreover, the foreground is also represented by a set of codebook vectors in order to obtain a more accurate foreground segmentation. Note that images are treated in the *CIE Lab* color space and filtered with a 2×2 mean low-pass filter as preprocessing

- *RGT* [25]. Image processing is carried out at region level, where background is modeled at different scales, from large to small rectangular regions, by using the color histogram and a texture measurement. So, motion is detected by comparing the corresponding rectangular regions from the coarsest scale to the finest one such that the comparisons are done at a finer scale only if motion was detected at a coarser scale. Furthermore, a Gaussian mixture background subtraction in combination with Minimum Difference of Pair Assignments (MDPA) distance [26] is used at the finest scale
- *Joint Difference* [4]. Motion is detected by means of an hybrid technique that uses both frame-by-frame difference and background subtraction. This technique integrates a selective updating method of the background model to tune background adaptation. In addition to a pixel-by-pixel difference in the *RGB* color space, a shadow filter in the *HSV* space is used to improve segmentation process.

A qualitative analysis highlights a good performance of the proposed approach (Fig. 2.9). It is worth noting that both *Time-of-day* and *Foreground-aperture* results present some false positives around the real foreground element due to the criteria designed to detect stop-moving foreground elements. Nevertheless, targets are correctly detected in all images, even though in *Bootstrapping* video sequence, where foreground elements appear from the first frame. This means that our approach successfully deals with the two well-known difference drawbacks (i.e. *ghosting* and *foreground aperture*).

From a quantitative point of view, two different statistical measurements have been considered: TPR and FPR. As previously introduced, the TPR evaluates foreground pixel classification. Thus, a high TPR means that the number of foreground pixels correctly classified is much larger than the number of foreground pixels misclassified as background. Nevertheless, it is also necessary to investigate the influence of the real background pixels in the extracted foreground, since TPR is just about the actual foreground pixels. For that, FPR is used to measure how many background pixels are classified as background. Note that the best technique should have the highest TPR value, but the lowest FPR value because a high FPR value means that most parts of the image are detected as foreground by making the background subtraction technique under study not appropriate to achieve our final goal.

Focusing on the results presented in Fig. 2.10, our approach has the best results except for the *Foreground-aperture* video sequence. The reason is that it has been influenced by the criteria established for detecting the situation when an object stops moving, and they have produced some false positives in the previous location of the target's head. On the other hand, the *FPR* is low (less than 5%) what means most of the image pixels are correctly classified (see Fig. 2.11). So, as the proposed approach



























	time of day	light switch	bootstrap	foreground aperture
Test Image				
Problem	Light gradually brightened	Light just switched on	No clean background training	Interior motion undetectable
Frame	1850	1865	299	489
Ground-truth				
Wallflower [23]				
Tracey Lab LP [24]				
RGT[25]				
Joint Difference [4]				
Combination of Differences (CoD)				

Fig. 2.9 Tests of different background maintenance algorithms for four canonical background problems contained in the *Wallflower* dataset [8] such that each column represents one considered image sequence. The *top row* shows the image in the sequence at which the processing was stopped. The *second row* shows hand-segmented images of the foreground used as *ground truth* for a quantitative comparison. The *rest rows* show the results of one algorithm

combines the highest TPR value and a low FPR value in most of the video sequences, it can be concluded that our segmentation process outperforms previous algorithm results.

Moreover, a quantitative analysis in terms of *recall* and *precision* is presented in Table 2.2. It can be observed that although another algorithm achieves a better result for a measurement in any situation, it is at the cost of obtaining a bad result

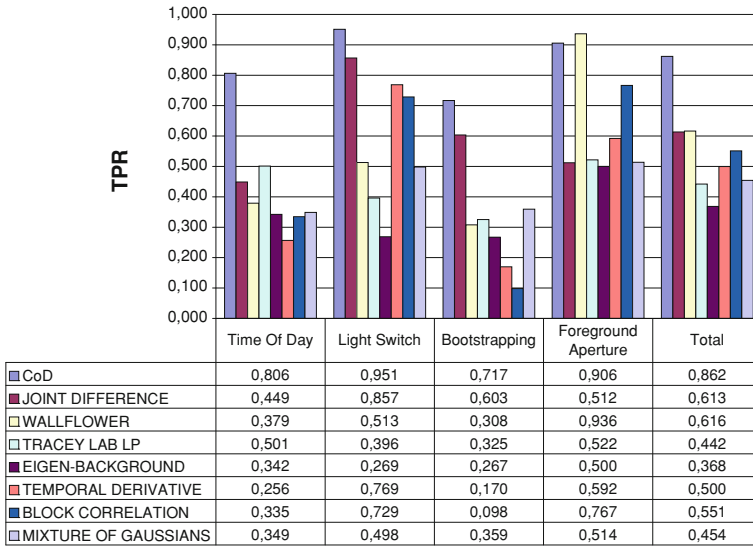


Fig. 2.10 True Positive Rate (TPR) of different background subtraction techniques for some video sequences of the *Wallflower* dataset [8], by including an extra column, *Total* that represents the result obtained for all the videos combined together. Note that, as TPR evaluates the foreground pixel classification, a high TPR value means that the number of foreground pixels correctly classified is much larger than the number of foreground pixels misclassified as background

for the other measurement. That is, there is no video sequence for which a previous algorithm overcomes the performance of CoD in both *recall* and *precision*.

A deeper quantitative study reveals the good performance of the proposed approach (see Table 2.3). So, first, TNR which expresses how many positive are wrongly tagged is presented. A high value of this measurement means a more accurate image segmentation since less background pixels were wrongly tagged as foreground. As it can be seen, the obtained results are close to 100%. Nevertheless, as in the case of FPR, TNR cannot be the only criterion for the evaluation of a segmentation technique. As a complementary measurement, NPV is used for evaluating how many foreground pixels have been wrongly classified as background. Again, a high value of this parameter refers to a more accurate performance. The results are also near 100%. Moreover, TNR measurement can be complemented with the FDR measurement when we are more interested in evaluating the error rate with respect to misclassified background pixels, that is, the percentage of the background pixels erroneously tagged as foreground. From its definition, a good performance will provide a low value for this parameter. As it can be checked in Table 2.3, FDR is lower than 17%, except for the *Bootstrap* sequence, which is slightly higher. The reason lies on the continuous presence of foreground elements in some parts of the image. This fact makes more difficult the segmentation problem.

On the other hand, the following measurements provide a global evaluation for the algorithm's performance. The first considered one is *accuracy* since it takes

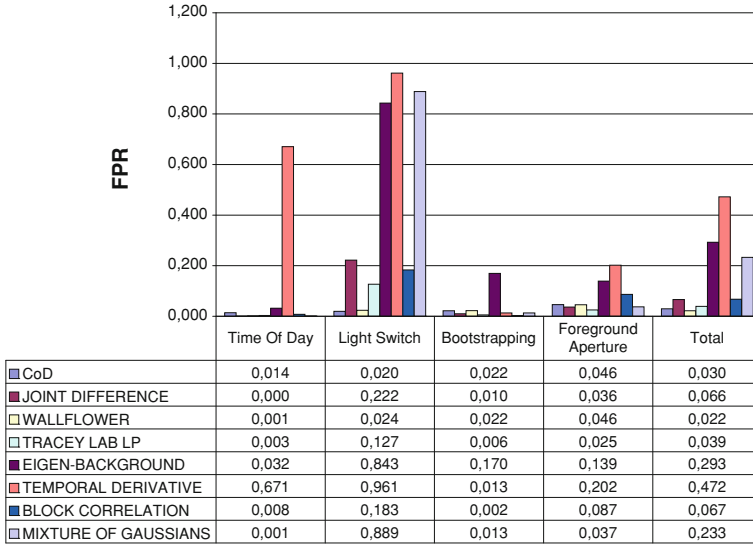


Fig. 2.11 False Positive Rate (FPR) of different background subtraction techniques for some video sequences of the *Wallflower* dataset [8], by including an additional column, *Total*, which contains the FPR for all the videos combined together. FPR is used to measuring how many background pixels are classified as background. Therefore, the best segmentation technique will have the lowest FPR value

into account pixels both correctly and wrongly classified. Mathematically speaking, *accuracy* will be higher when lower classification errors are made. Thus, an *accuracy* of 100 % is desired. The obtained results are >90 % in all the considered image sequences. Another way to assess the system's performance is based on the MCC value. Basically, it is a balance measurement that provides information about the correlation between pixels correctly tagged and those wrongly classified. So, values near 1, as the ones obtained for the proposed approach, result in an accurate segmentation. Note that a lower value is again obtained for the *Bootstrap* sequence. The reason is the continuous presence of foreground elements in some parts of the image. Regarding F_1 Score, it is a kind of average of *precision* and *recall* measurements. The F_1 Score reaches its best value at 1. Note that all the obtained results are close to the unit. The JC evaluates the algorithm's accuracy when foreground pixels are considered. So, a low error rate will provide JC values around 1. In this case, the *Bootstrap* and the *Time-of-day* sequences have obtained the lowest values. Finally, the YC value expresses the relationship between foreground and background pixels correctly tagged and its value oscillates between 1 and -1 , by providing a better performance when it is around 1. All values are positive and close to the unit. Again, the *Bootstrap* sequence has obtained the worst result, although it is nearly 0.70.

Table 2.2 Comparison of the experimental quantitative results, in terms of *recall* and *precision*, obtained for different segmentation methods on some *Wallflower* benchmarks [8]

Algorithm	Measurement	Time of day	Light switch	Bootstrap	Foreground aperture	Total
Mixture of Gaussians [18]	recall	34.88	49.82	35.93	51.37	45.43
	precision	96.43	10.27	82.89	82.96	27.94
Block correlation [19]	recall	33.46	72.86	9.81	76.66	55.11
	precision	79.33	44.82	89.13	75.79	61.93
Temporal derivative [20]	recall	25.65	76.89	16.99	59.20	49.96
	precision	20.26	17.44	13.21	93.20	17.39
Bayesian decision [21]	recall	34.24	26.86	26.74	50.00	36.84
	precision	48.53	6.11	22.05	55.97	20.04
Eigenback ground [22]	recall	43.22	70.44	89.61	51.39	64.03
	precision	97.66	86.36	29.95	82.78	53.68
Wallflower [23]	recall	37.92	51.29	30.77	93.63	61.64
	precision	95.92	81.65	71.15	87.87	84.75
Tracey LAB LP [24]	recall	50.13	39.61	32.51	52.15	44.20
	precision	93.49	38.91	91.18	88.03	69.05
RGT [25]	recall	29.09	51.19	57.76	50.18	47.06
	precision	99.00	44.04	91.40	83.34	64.89
Joint difference [4]	recall	44.90	85.68	60.34	51.20	61.13
	precision	99.00	44.04	91.40	83.34	64.89
Combination of differences	recall	80.62	95.11	71.69	90.56	86.19
	precision	83.31	90.82	73.73	88.55	85.27

2.3.2.2 VSSN06 Dataset

This dataset was developed for an algorithm competition in Foreground/Background Segmentation within the *Forth ACM International Workshop on Video Surveillance and Sensor Networks*. Their motivation was based on the results reported in the literature that did not provide a direct comparison among algorithms because each

Table 2.3 Quantitative results obtained for the **CoD** approach over some video sequences of the *Wallflower* dataset [8] such that the first three measurements (TNR, NPV, and FDR) provide a performance evaluation related to misclassified/correctly classified pixels, while the rest of measurements provide a global assessment of the algorithm performance

	Time of day	Light switch	Bootstrap	Foreground aperture	Total
TNR	98.58	98.04	95.41	95.85	97.04
NPV	98.31	98.99	94.94	96.63	97.24
FDR	16.69	9.18	26.27	11.45	14.73
Accuracy	97.14	97.54	91.80	94.45	95.24
MCC	0.80	0.91	0.68	0.86	0.83
F_1 Score	0.82	0.93	0.73	0.90	0.86
JC	0.69	0.87	0.57	0.81	0.75
YC	0.82	0.90	0.69	0.85	0.83

So, high values for TNR, NPV, accuracy, MCC, F_1 score, JC, and YC, and low values of FDR result in an accurate segmentation

researcher reports results using different assumptions, evaluation methods, and test sequences.

Each of its 12 test videos consists of a video that illustrates a background with dynamic elements sometimes, and one or more virtual foreground objects, taken from [27, 28] together with a foreground mask video (ground-truth video), in most of the video sequences, by specifying each pixel belonging to a foreground object. These color videos evaluate algorithm's performance in view of the different canonical problems mentioned above. Particularly, the considered problems here are:

- oscillating background
- gradual illumination changes
- sudden changes in illumination
- bootstrapping
- shadows

Again, we have concentrated on those video sequences that evaluate algorithm's performance when background elements are motionless. Therefore, only four videos are considered. So, in *video sequence 1*, an indoor scene without oscillating elements is the background where a virtual girl is moving around. No ground truth information has been provided for this video sequence. That is why only qualitative results are presented. So, as it can be observed in Fig. 2.12, the target element was detected in all frames, even when the target is partially visible (e.g. frame at time 143). However, some false positives appear around the foreground object. That is because the similarity criterion is defined at pixel level and no extra information is used. Therefore, a more accurate segmentation could be obtained if any cognitive knowledge is integrated in the system.

In a similar way, the *video sequence 2* represents an indoor scene where no oscillating elements appear. In this case, the target elements are two boys who are dancing along the whole scene. Again, the qualitative results, presented in Fig. 2.13, highlight the proper identification of the background and foreground pixels. As in the

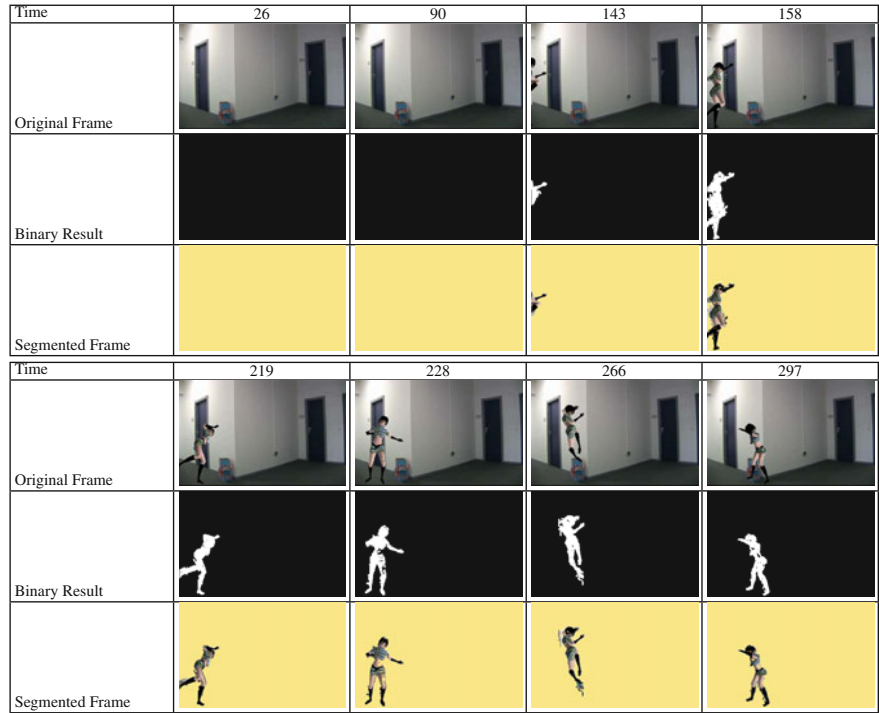


Fig. 2.12 Qualitative results for the *video sequence 1* of the *VSSN06* dataset [9], where a virtual girl is moving around an indoor scene without oscillating elements. So, the *first row* of each block shows the original frame of the sequence, while the *other rows* depict the segmentation result obtained by the **CoD** approach: a binary image representing the background/foreground classification carried out, such that the background is represented by *black color* and the foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas the background is coded in an artificial, homogeneous color

previous case, a few false alarms have been detected in the surrounding target borders. Nevertheless, the quantitative results reflect the good performance of the algorithm with high values for measurements corresponding to correctly tagged pixels, whereas those related to misclassified pixels have low values (see Table 2.4).

With regard to the *video sequence 5*, as in *video sequences 1* and 2, a static indoor scene is used as background. The particularity of this sequence is the presence of foreground elements from the first frame. In addition, two different kinds of target elements are considered. On the one hand, a virtual human being who is dancing around the scene. On the other hand, a cat which is walking around the scene. Thus, this video sequence evaluates both the presence of foreground elements during the whole experiment and the detection of target elements different from human beings. As depicted in Fig. 2.14, both elements of interest were successfully identified. It is worth noting that the presence of a foreground element in the reference frame has been properly detected when it started to move by solving the *ghosting*

Table 2.4 Quantitative results obtained for the **CoD** approach over some video sequences of the *VSSN06* dataset [9] such that the first seven measurements provide a performance evaluation related to misclassified/correctly classified pixels, while the rest of measurements provide a global assessment

	Recall	Precision	TPR	FPR	TNR	NPV	FDR	Accuracy	MCC	F_1 Score	JC	YC
Video 2	80.07	86.25	80.07	0.39	99.61	99.43	13.75	99.09	0.82	0.82	0.70	0.86
Video 5	79.22	73.90	73.90	2.86	97.14	98.73	25.02	96.18	0.73	0.73	0.60	0.70
Video 6	71.71	87.81	71.17	0.81	99.19	97.92	10.85	97.29	0.77	0.76	0.65	0.83

So, high values for recall, precision, TPR, TNR, NPV, accuracy, MCC, F_1 score, JC and YC, and low values of FPR and FDR, result in an accurate segmentation

problem. Nevertheless, the no detection of the foreground element in the early frames has influenced the quantitative results, summarized in Table 2.4. So, although the relationship between the measurements to show a good performance is kept, the values are a little bit lower, or higher in the case of the pixel misclassification, than it could be expected.

Again, the problem of lacking a frame free of foreground elements is considered in the *video sequence 6*. The background scene is similar to the one in *video sequences 1, 2 and 5*, an indoor scene with constant illumination conditions where one or more foreground elements are moving around. In particular, a virtual boy is in the scene in the first captured frame, moves around, leaves, and re-enters the scene, while a little girl enters and leaves the scene during the whole experiment. Both qualitative and quantitative results are presented (see Fig. 2.15 and Table 2.4). Analyzing the qualitative results, the foreground element is not detected in the early frames, since it is not moving and it is initially classified as background. Then, it starts moving and the proposed approach has detected this situation by properly updating its reference frame. That is why there is no *ghost* presence in the frame at time 21. Later, the girl enters the scene and both targets are detected without any problem, even when they partially appear, as in frame at time 367 or at time 380. From a quantitative point of view, bootstrapping event has had less influence on the results than on the previous video sequence by showing a better performance, i.e., a lower error rate.

2.3.2.3 Audiovisual People Dataset

This dataset, courtesy of EPSRC funded MOTINAS project (EP/D033772/1), for uni-modal and multi-modal (audio and visual) people detection tracking, consists of three video sequences recorded in different scenarios with a video camera and two microphones, although, in our case, only the image sequences have been used.

The 8-bit color AVI sequences were recorded by using a KOBİ KF-31CD analog CCD surveillance camera in the Department of Electronic Engineering—Queen Mary University of London. Two of the image sequences were recorded in rooms with reverberations, whereas the third one was recorded in a room with reduced

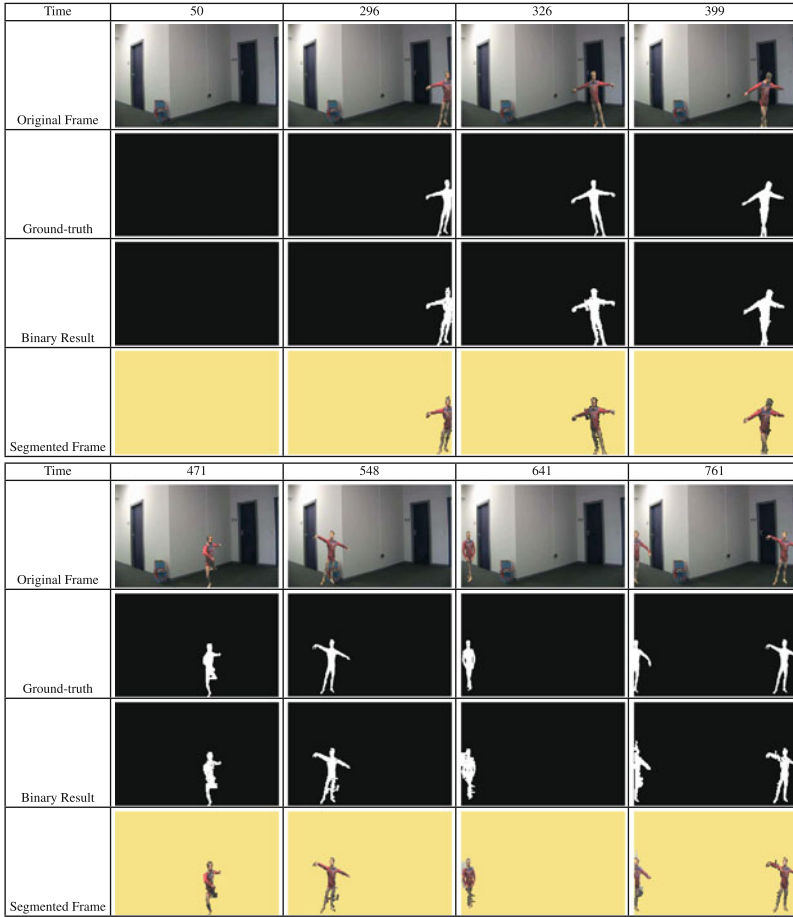


Fig. 2.13 Qualitative results for the *video sequence 2* of the *VSSN06* dataset [9], where two virtual boys are dancing along an indoor scene where no oscillating elements appear. So, the *first row* shows the original frame of the sequence, the *second row* depicts the ground truth frame and, the last *two rows* illustrate the segmentation result obtained by the **CoD** approach: a binary image representing the background/foreground classification carried out, such that the background is represented by *black color* and the foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas the background is coded in an artificial, homogeneous color

reverberations, although all of them were captured at a frame rate of 25 Hz with a 360×288 resolution.

Unlike the previous study cases, no quantitative results are presented since no ground truth is provided for this dataset. However, it is used because it considers some issues that are missed in the previous datasets such as occlusions, the change in targets' speed, and/or in the camera pose with respect to the scene.



Fig. 2.14 Qualitative results for the *video sequence 5* of the *VSSN06* dataset [9], where the lack of frames without foreground elements is analyzed. So, the *first row* shows the original frame of the sequence, the *second row* depicts the ground truth frame and, the last *two rows* illustrate the segmentation result obtained by the **CoD** approach: a binary image representing the background/foreground classification carried out, such that the background is represented by *black color* and the foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas the background is coded in an artificial, homogeneous color

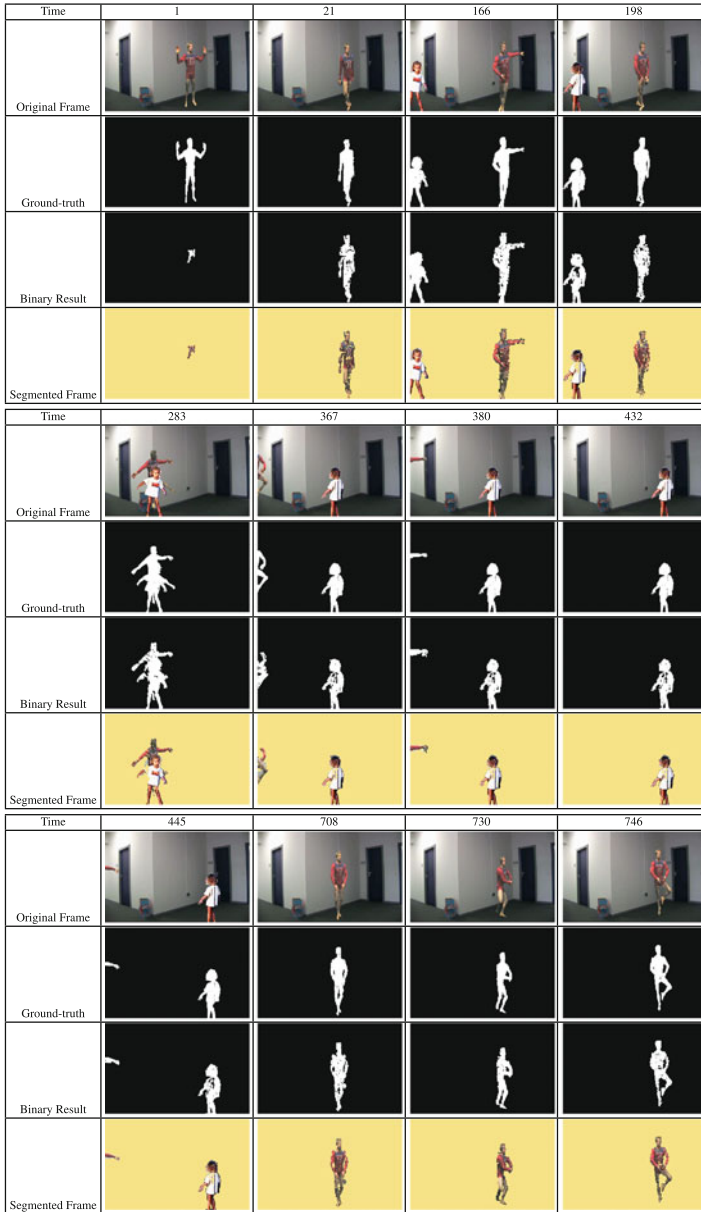


Fig. 2.15 Qualitative results for the *video sequence 6* of the *VSSN06* dataset [9] such that the problem of the absence of a frame free of foreground elements is studied. So, the *first row* shows the original frame of the sequence, the *second row* depicts the ground-truth frame and, the last *two rows* illustrate the segmentation result obtained by the **CoD** approach: a binary image representing the background/foreground classification carried out, such that the background is represented by *black color* and the foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas the background is coded in an artificial, homogeneous color

At first instance, a classroom where a person is moving around is observed. Again, the problem of the foreground element presence in the initial frames and its corresponding presence in the reference frame for the *background(-frame) subtraction* is considered. Moreover, in this video sequence, the occlusion problem is also taken into account. So, the target of the sequence, a guy, appears on the left of the image in the initial frames and, while he is moving toward the right side, is occluded. Then, he reappears in the scene and walks around it by approaching and going away until he again disappears of the image as a consequence of a new occlusion. Finally, he re-enters the scene and moves around it. Figure 2.16 shows some frame samples of the obtained result. It is worth noting that the person of interest was successfully detected in all the frames although he was occluded and his distance with respect to the camera was considerable and variable.

In a similar way, the second considered video sequence observes a computer room where two people are constantly entering and leaving it. Nevertheless, in this case, the initial frames are free of foreground pixels which means that the reference frame for the *background(-frame) subtraction*, initially set to the first captured frame, is an exact model of scene background. So, the interest in this video sequence lies on the number of targets, i.e., two individuals, and the fact that they are crossing and overlapping several times during the whole experiment. In the resulting frames, depicted in Fig. 2.17, different situations are analyzed: (1) the absence of foreground elements; (2) the partial presence and subsequent appearance of one of the interest people; (3) the presence of one or both of them, even when they cross (e.g. frame at time 810); and, (4) the scene without any foreground element. Again, from a qualitative point of view, the proposed approach presents a good performance.

The last video sequence was recorded in a room with reduced reverberations. Basically, there are two people who continuously enter and leave the scene such that they change their speed and trajectories all the time. As it can be observed in Fig. 2.18, the individuals are properly detected in all frames. Note the presence of shadows in some of the resulting images, since no processing to erase them has been applied at this point.

2.3.3 Experimental Results Over Our Own Dataset

In this section, we present some results obtained from different experiments carried out in our laboratory. Mainly, they consist of locating an imaging device at different places of our laboratory room. Although the lab contains some dynamic factors (e.g., blinking of computer screens), they have been avoided for this section. Furthermore, as previously pointed out, two different kinds of imaging devices have been used. So, results for both perspective and fisheye devices are presented.

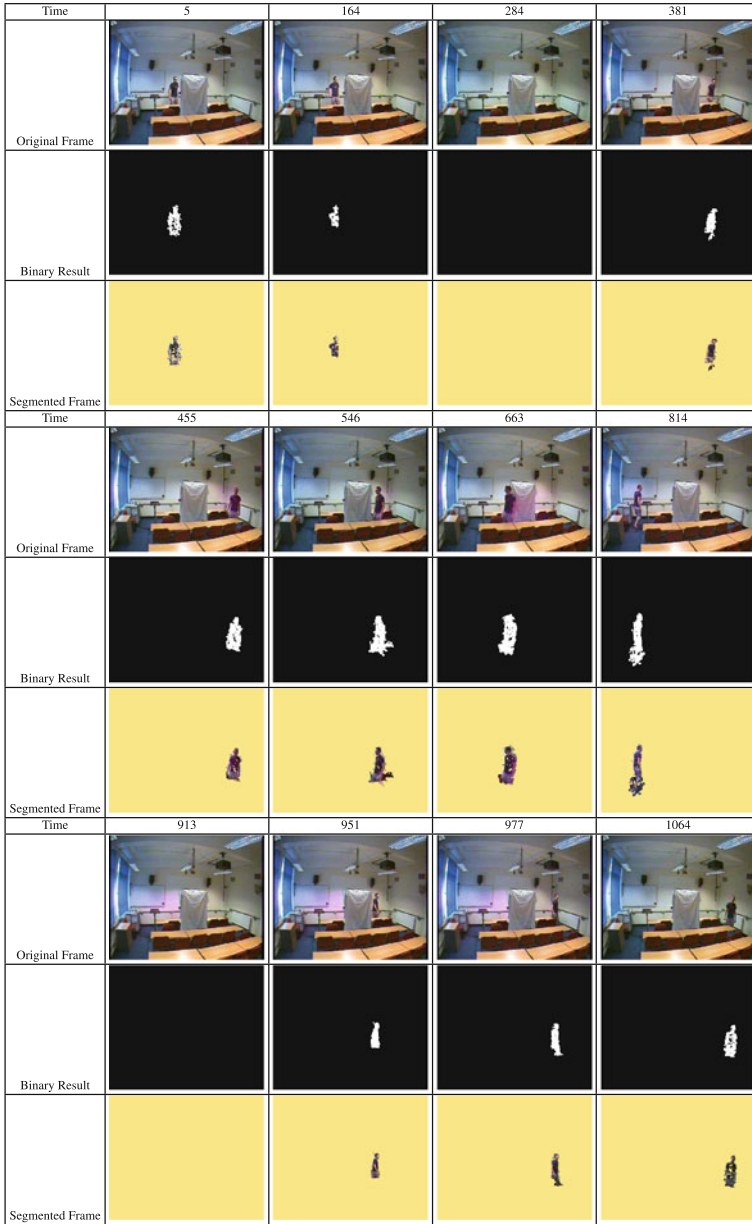


Fig. 2.16 Qualitative results over *Room 105* image sequence of Audiovisual People dataset [10] where a person is moving around a classroom. So, the *first row* of each block shows the original frame of the sequence, whereas the last *two rows* illustrate the segmentation result obtained by the **CoD** approach: a binary image representing the background/foreground classification carried out, such that the background is represented by *black color* and the foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas the background is coded in an artificial, homogeneous color

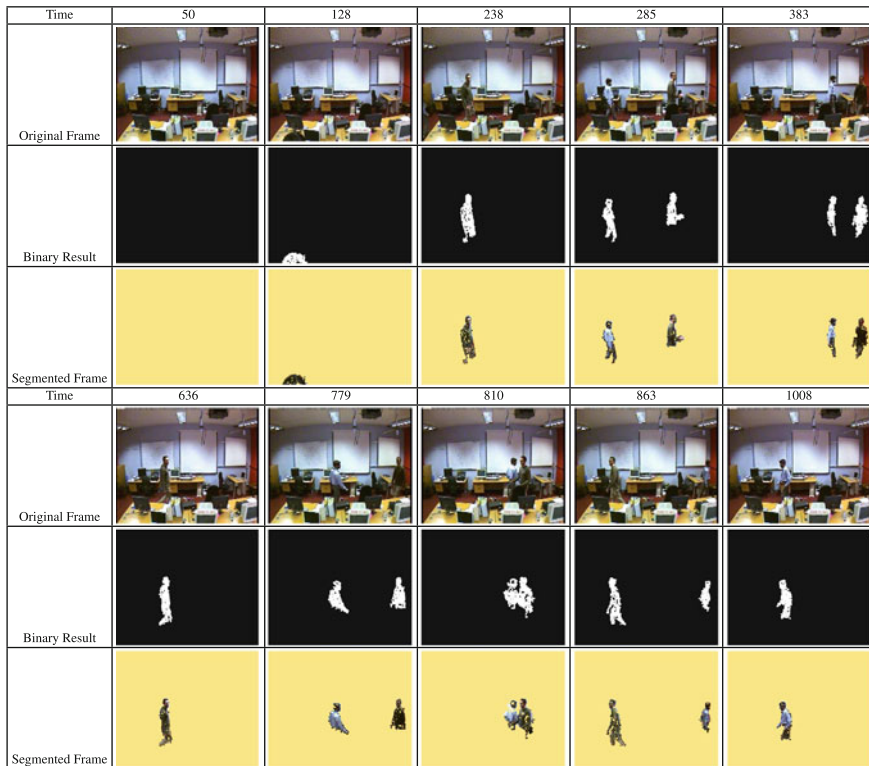


Fig. 2.17 Qualitative results over *Room 160* video sequence of Audiovisual People dataset [10] where two people are moving around a computer room. So, the *first row* of each group shows the original frame of the sequence, whereas the last *two rows* illustrate the segmentation result obtained by the CoD approach: a binary image representing the background/foreground classification carried out, such that the background is represented by *black color* and the foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas the background is coded in an artificial, homogeneous color

2.3.3.1 Perspective Image Experiments

First, a perspective imaging device has been used. In particular, a *STH-DCSG Stereo head* by using one of its two C-mount lenses was employed [29]. Basically, it is a synchronized digital stereo head camera with two global shutter CMOS imagers, offering VGA resolutions at 30 fps. Nevertheless, different features have been tested. So, on the one hand, images were acquired in *monochrome* mode with a 320×240 resolution and, on the other hand, 640×480 , 24-bit RGB color images are considered.

In both experiments, the goal is to properly detect the presence of a person in the scene, continuously entering and leaving the observed space. However, experimental conditions have been changed. In the first experiment, illumination changes do not occur. An individual enters and moves around the scene by approaching and moving



Fig. 2.18 Qualitative results over *Chamber* video sequence of Audiovisual People dataset [10] where two people are continuously entering and leaving a room with reduced reverberations. So, the *first row* of each group shows the original frame of the sequence, whereas the last *two rows* illustrate the segmentation result obtained by the **CoD** approach: a binary image representing the background/foreground classification carried out, such that the background is represented by *black color* and the foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas the background is coded in an artificial, homogeneous color

away the camera until a distance of 9 m. Furthermore, occlusions and stop motions have been also analyzed with this image sequence. A good performance is obtained in all those situations as depicted in Fig. 2.19.

Regarding the second experiment, the camera was located at a different place in our laboratory room. Again, a person is continuously entering, moving around, and leaving the scene. However, in this case, global illumination changes take place. So, initially, the visual system is observing a very bright scene. As shown in the first row of Fig. 2.20, the target individual is successfully detected at several positions, in spite of some internal pixels are misclassified as background. The main reason lies on the similarity between the pixel intensities since CoD algorithm works at pixel level. Then, a global illumination change takes place by slightly darkening the scene.





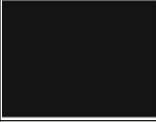



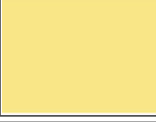
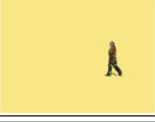

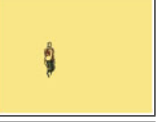







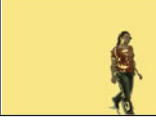
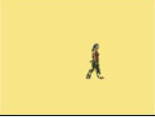
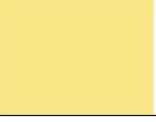

Time	50	240	244	277
Original Frame				
Binary Result				
Segmented Frame				
Time	311	481	555	639
Original Frame				
Binary Result				
Segmented Frame				

Fig. 2.19 Qualitative results over color perspective images such that a person is continuously entering and leaving our laboratory room. So, the *first row* shows the original frame of the sequence, whereas the last *two rows* illustrate the segmentation result obtained by the **CoD** approach: a binary image representing the background/foreground classification carried out, such that background is represented by *black color* and foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas background is coded in an artificial, homogeneous color

At this point, the approach’s performance is more accurate by providing less false negatives. Finally, another global illumination change makes the scene very dark. Although the illumination is poor, the proposed approach is capable of detecting the individual. Note that the darker the scene is, the higher the shadow presence is. That is, because the intensity of the shadow pixels is more affected by this phenomenon, their value is different enough from the background pixel brightness to be wrongly labeled as foreground. Moreover, a background element (a chair) is moved. Note that it is not correctly detected both in the new position and the old one. The reason is that when it is moved, it is identified that it is a background element that has started to move. So, the left *hole* is properly covered with the new background. Then, when it is located at the new position, it is adequately identified as a background element, as it can be observed.

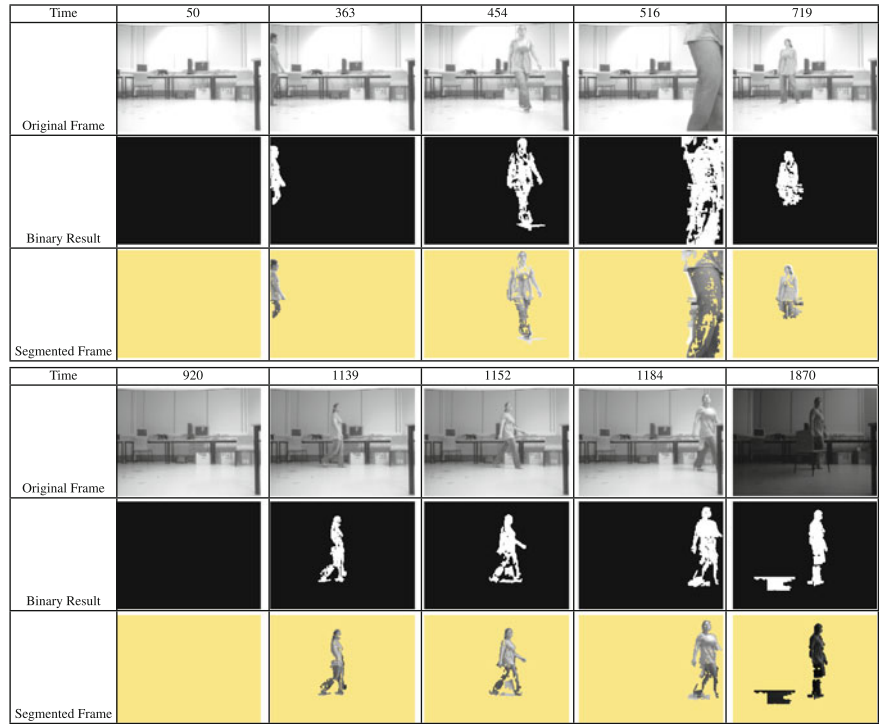


Fig. 2.20 Qualitative results over gray-scale perspective images such that a person is continuously entering and leaving our laboratory room. So, the *first row* shows the original frame of the sequence, whereas the last *two rows* illustrate the segmentation result obtained by the **CoD** approach: a binary image representing the background/foreground classification carried out, such that the background is represented by *black color* and the foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas the background is coded in an artificial, homogeneous color

2.3.3.2 Fisheye Image Experiments

In this section, the CoD’s performance is assessed over fisheye images. For that, a *DR2-COL-CSBOX* camera with a *Fujinon YV2.2 × 1.4A-2 1/3” 1.4–3.1 mm CS-Mount* lens was used [30, 31]. At this instance, the fisheye camera was located at the center of another laboratory room, pointing upwards, by monitoring the presence of an individual around the visual system. Some examples of the algorithm’s performance over this image sequence are depicted in Fig. 2.21, while the set of parameters used is summarized in Table 2.5.

As it can be observed, the performance results are even better than those over perspective images, in spite of the lightning source blink, which is properly corrected by avoiding false positives due to it. Also note that, unlike the perspective images, the proximity to the camera affects in large extent to the pixel intensity values. A

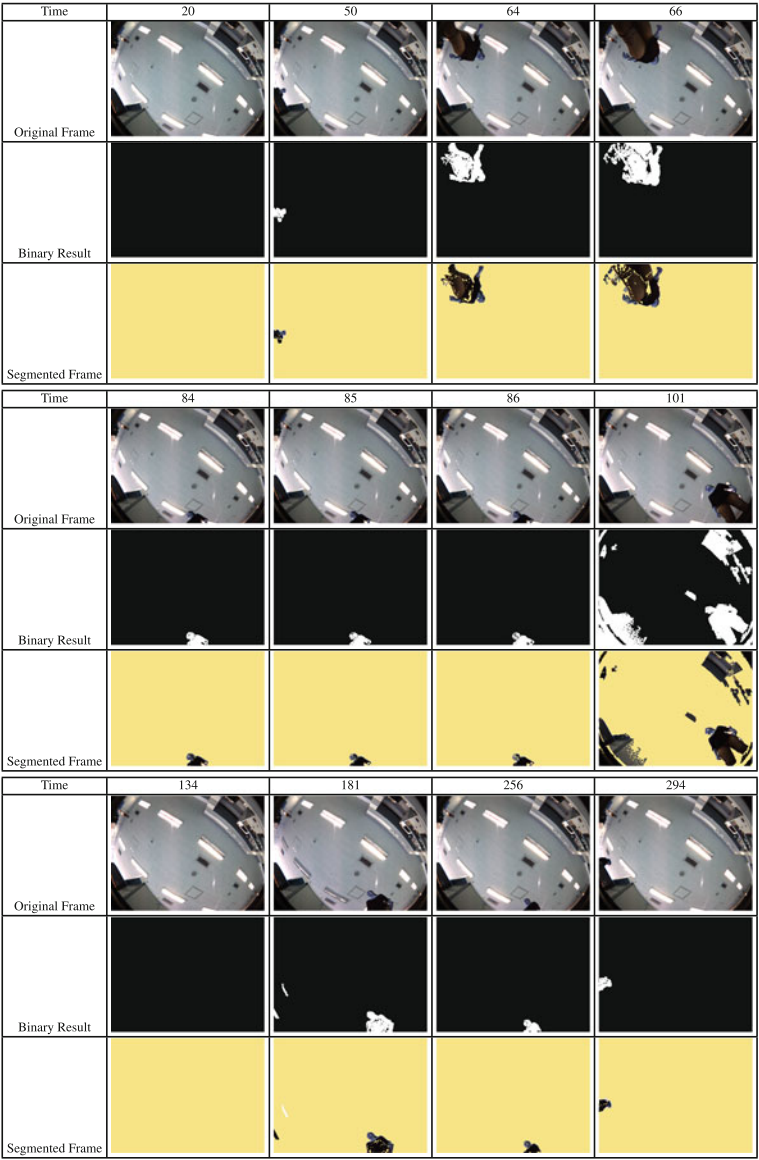
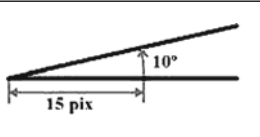


Fig. 2.21 Qualitative results for fisheye images for a video sequence where a person is continuously entering and leaving our laboratory room. So, the *first row* shows the original frame of the sequence, whereas the last *two rows* illustrate the segmentation result obtained by the **CoD** approach: a binary image representing the background/foreground classification carried out, such that the background is represented by *black color* and the foreground pixels are coded in *white*; and a color image, where the foreground elements appear as in the original frame, whereas the background is coded in an artificial, homogeneous color

Table 2.5 Parameter values used for fisheye images when the **CoD** is performed

										
Subimage Size										
Erosion Mask	<table><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	1	0	0	1	1	0	0	0
0	1	0								
0	1	1								
0	0	0								
Dilation Mask	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	1	1
1	1	1								
1	1	1								
1	1	1								

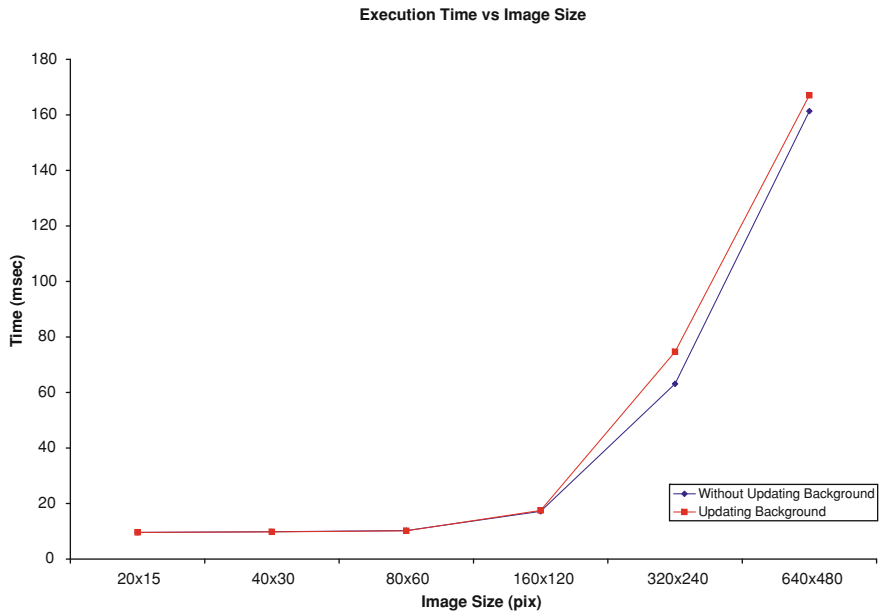


Fig. 2.22 Execution time analysis of the **CoD** approach based on the image size and the background frame update on an Intel(R) Core(TM) Duo CPU P8700 at 2.53 GHz

sample of this can be observed in frame at time 101 or at time 181. Nevertheless, this temporary change is rightly not considered as a global illumination change. That is why classification in consecutive frames was again successful.

To conclude this section, an execution time analysis is carried out. For that, two different parameters are considered: the image size and the process to update the reference frame for the *background(-frame) subtraction*. The CoD C++ implementation was run on an Intel(R) Core(TM) Duo CPU P8700 at 2.53 GHz by obtaining

the execution time depicted in Fig. 2.22. As it can be observed, the execution time is slightly higher when the updating operation is used. However, it is worth noting that it was considered the worst case in which the updating operation was required. In addition, real-time performance is obtained for a 320×240 image resolution.

2.4 Conclusions

In this chapter we have studied the basic case of motion detection, that is, motion detection in scenes with background motionless, aiming at analyzing and solving different issues referred to the use of different imaging sensors, the adaptation to different environments, different motion speed, the shape changes of the targets, or some uncontrolled dynamic factors such as, for instance, gradual/sudden illumination changes. As a solution, a CoD techniques has been proposed. Mainly, it combines a *frame-by-frame difference* together with a *background(-frame) subtraction* with the purpose of overcoming the two well-known difference drawbacks (i.e., *ghosting* and *foreground aperture*). Moreover, on the way to autonomous, robust visual systems, it has also been necessary to study the automatic threshold estimation. For that, a *dynamic* thresholding method based on resolution distribution in an image has been presented. This technique automatically divides the captured images and sets the proper thresholding parameters for two different kinds of cameras: perspective and fisheye. So, problems such as non-uniform-distributed resolution, inadequate illumination gradient in the scene, unsuitable contrast, or the overlapping of the background and the target gray-level distributions, are overcome.

In addition, some experiments over public image datasets and our own image datasets were carried out. Both quantitative and qualitative results have been provided in order to assess the CoD's performance under different conditions. As the experimental results have highlighted, the proposed approach is able to deal with different imaging devices, variable target's speeds or types of interest elements such as people or animals. Furthermore, a comparative analysis with some well-known techniques (those that have provided results on these image datasets) has demonstrated that our approach outperforms them.

Finally, a time-execution analysis has been presented. In that study, two different parameters were considered: the image size and the process to update the reference frame for the *background(-frame) subtraction*. It highlights that the execution time depends on the image size, although a real-time performance is obtained for a 320×240 image resolution. So, the proposed approach can be used for real-time robotic tasks.

References

1. Collins, R., Lipton, A., Kanade, T., Fijiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt, P., Wixson, L.: A system for video surveillance and monitoring. Tech. rep., Carnegie Mellon University, Pittsburgh, PA (2000)

2. Kameda, Y., Minoh, M.: A human motion estimation method using 3-successive video frames. In: International Conference on Virtual Systems and Multimedia (VSMM), pp. 135–140. Gifu, Japan (1996)
3. Kanade, T., Collins, R., Lipton, A., Burt, P., Wixson, L.: Advances in cooperative multi-sensor video surveillance. In: Darpa Image Understanding Workshop, vol. I, pp. 3–24. Morgan Kaufmann (1998)
4. Migliore, D., Matteucci, M., Naccari, M.: A revaluation of frame difference in fast and robust motion detection. In: 4th ACM International Workshop on Video Surveillance and Sensor Networks (VSSN), pp. 215–218. Santa Barbara, California (2006)
5. Wren, C., Azarbeyejani, A., Darrell, T., Pentland, A.: Pfunder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **19**(7), 780–785 (1997)
6. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging* **13**(1), 146–168 (2004)
7. Mičušík, B.: Two view geometry of omnidirectional cameras. Ph.D. thesis, Center for Machine Perception, Czech Technical University in Prague (2004)
8. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: <http://research.microsoft.com/en-us/um/people/jckrumm/WallFlower/TestImages.htm> (1999)
9. Hörster, E., Lienhart, R.: http://mmc36.informatik.uni-augsburg.de/VSSN06_OSAC/ (2006)
10. Taj, M.: Surveillance performance evaluation initiative (spevi)—audiovisual people dataset. http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007_d.html (2007)
11. Ferryman, J.: <http://www.cvg.rdg.ac.uk/PETS2006/data.html> (2006)
12. Berger, J., Patel, T., Shin, D., Piltz, J., Stone, R.: Computerized stereochronoscopy and alteration flicker to detect optic nerve head contour change. *Ophthalmology* **107**(7) (2000)
13. Hu, J., Kahsi, R., Lopresti, D., Nagy, G., Wilfong, G.: Why table ground-truthing is hard. In: Sixth International Conference on Document Analysis and Recognition, pp. 129–133. Seattle, WA, USA (2001)
14. Rosin, P., Ioannidis, E.: Evaluation of global image thresholding for change detection. *Pattern Recognition Letters* **24**(14), 2345–2356 (2003)
15. Cheung, S., Kamath, C.: Robust techniques for background subtraction in urban traffic video. *Electronic Imaging: Video Communications and Image Processing* **5308**(1), 881–892 (2004)
16. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H., Rosenberger, C.: Review and evaluation of commonly-implemented background subtraction algorithms. In: 19th International Conference on Pattern Recognition (ICPR), pp. 1–4. Tampa, Florida (2008)
17. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: 23rd International Conference on Machine Learning, pp. 233–240. Pittsburgh, Pennsylvania (2006)
18. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 246–252 (1999)
19. Matsuyama, T., Ohya, T., Habe, H.: Background subtraction for non-stationary scenes. In: Fourth Asian Conference on Computer Vision, pp. 662–667. Singapore (2000)
20. Haritaoglu, I., Harwood, D., Davis, L.: W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **22**(8), 809–830 (2000)
21. Nakai, H.: Non-parameterized bayes decision method for moving object detection. In: Asian Conference on Computer Vision. Singapore (1995)
22. Oliver, N., Rosario, B., Pentland, A.: A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **22**(8), 831–843 (2000)
23. Toyama, K., Krum, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Seventh IEEE International Conference on Computer Vision (ICCV), vol. 1, pp. 255–261. Kerkyra, Greece (1999)
24. Kottow, D., Koppen, M., del Solar, J.R.: A background maintenance model in the spatial-range domain. In: 2nd ECCV Workshop on Statistical Methods in Video Processing, pp. 141–152. Prague, Czech Republic (2004)

25. Varcheie, P., Sills-Lavoie, M., Bilodeau, G.A.: An efficient region-based background subtraction technique. In: Canadian Conference on Computer and Robot Vision, pp. 71–78 (2008)
26. Cha, S., Srihari, S.: On measuring the distance between histograms. *Pattern Recognition* **35**(6), 1355–1370 (2002)
27. Max-Planck-Institut-Informatik: <http://www.mpi-inf.mpg.de/departments/irg3/software.html> (2005)
28. <http://www.gifart.de/> (2002)
29. VidereDesign: <http://198.144.193.48/index.php?id=31>
30. PointGrey: <http://www.ptgrey.com/products/dragonfly2/index.asp> (2009)
31. Fujinon: <http://www.fujinon.com/Security/Product.aspx?cat=1019\&id=74> (2009)

Robust Motion Detection in Real-Life Scenarios

Martínez-Martín, E.; Pobil, Á.P.d.

2012, XII, 108 p. 70 illus., 61 illus. in color., Softcover

ISBN: 978-1-4471-4215-7