2

Background modelling with associated confidence

J. Rosell-Ortega and G. Andreu-García
Computer Vision Group. DISCA. Technical University of Valencia
Spain

In this chapter we introduce an algorithm aimed to create background models which associates a confidence value to the obtained model. Our algorithm creates the model based on motion criteria of the secene. The goal of this value is to quantify the quality of the model after a number of frames have been used to build it. The algorithm is first designed in gray tones and for unimodal background models and through the chapter is extended for colour scenarios and with the possibility of using several models per pixel. Quantitative and qualitative experimental results are obtained with a well-known benchmark.

1. Introduction

Visual analysis of human motion (Wang .et. al., 2003) is currently one of the most active research topics in computer vision. This strong interest is driven by a wide spectrum of promising applications in many areas such as *virtual reality*, *smart surveillance* and *perceptual interface*, just to mention the most representative.
Visual analysis concerns the detection, tracking and recognition of objects in general, and particularly, people. Also the understanding of human behaviour in the case of image streams involving humans. Visual analysis of a scene starts from a segmentation of the scene in order to classify pixels as foreground or background; then, other steps may be taken depending on the application, such as motion analysis, object detection, object classification, tracking and activity understanding.
Background subtraction is usually mentioned in the literature concerning smart surveillance, as one of the most popular methods to detect regions of interest in frames. This technique consists in subtracting the acquired frame from a background model and classifying as foreground all those pixels whose difference with the background is over a threshold. Thus, the importance of producing an accurate background model and choosing a precise threshold is obvious.

1.1 Related work

A large number of different methods have been proposed in recent years and many different features have been used to maintain the background model and perform the background subtraction. Most of the methods rely on describing the background model with pixel's

intensity or colour information (Sshoustarian & Bez, 2005), (Hu et. al., 2004), (Elgammal et. al. 2000). But some others rely on other kind of information, for instance, edge detection, optical flow or textures, (Mason and Duric, 2001), (Wixson, 2000), (Heikkilä & Pietikäinen, 2006).

One of the methods, which rely on the pixel's intensity, consists in modelling each pixel in a video frame with a Gaussian distribution. This is the underlying model for many background subtraction algorithms. A simple technique is to calculate an average image of the scene, to subtract each new video frame from it and to threshold the result. The adaptive version of this algorithm updates the model parameters recursively by using a simple adaptive filter. This single Gaussian model can be found in (Wren et. al., 2004).

This model however, does not work well when the background is not static. For instance, waves, clouds or any movement which does also belong to the background cannot be properly described using one Gaussian distribution. A solution is using more than one Gaussian to model the background, as proposed in (Stauffer & Grimson, 1999]. In (Zang & Klette, 2004) methods for shadow detection and per-pixel adaptation of the parameters of the Gaussians are developed.

Following with the methods based on mixture of Gaussians, in (Elgammal et. al. 2000), it is proposed to build a statistical representation of the background. This is done by estimating directly from data the probability density function, with no previous assumptions about the underlying distribution.

Other approaches which do not rely on Gaussian distributions to model the background can be found, for instance, in (Mason and Duric, 2001). In this paper, the algorithm proposed computes a histogram of edges in a block basis. These histograms are constructed using pixel-specific edge directions. A fusion of this approach with intensity information may be found in (Jabri et. al., 2000).

Motion may also be used to model the background. Authors of (Wixson, 2000) propose an algorithm that detects salient motion by integrating frame-to-frame optical flow over time. Salient motion is considered to be motion that tends to move in a consistent direction over time.

Radically different is the approach introduced in (Heikkilä & Pietikäinen, 2006). These authors propose using features bases on textures to model the scene and to detect moving objects. The features used are LBP (local binary pattern) and the algorithm models the background using these features by assigning each pixel with a set of LBP histograms. As authors state, their algorithm has a lot of parameters to tune.

Though the aforementioned approaches obtain good results in the tested scenarios, in general, all these approaches expect working in scenarios with low or null activity to build their first model. One of aspects which we miss in these approaches is that there is no measure of when a suitable background is achieved.

Besides the different approaches to background modelling, another issue related to this technique is the detection of corrupt models, that is, models which are not useful any more for surveillance purposes.

Few papers in the literature address this issue, as far as authors of this chapter are concerned. In the literature it is generally assumed that changes in the background will occur smoothly and abrupt changes are not considered. In (Toyama et.al., 1999), authors propose maintaining a database of models. In the case the background model is considered

to be corrupt, by whichever the mean, a search in the database should be enough to find the most suitable model.
In our opinion, this is a very time consuming process, and does not solve completely the problem. We consider that recovering a corrupt background model is the same as creating a new one. In this chapter, we explore the possibility of giving a unique solution to both problems. Thus, only a method to detect corrupt models must be defined and the model recovery may just be considered as a restart of the system, building a new background model.

1.2 Goals

Our developments are constrained to concrete situations. We focus our attention specially on demanding scenarios, which are those in which there is always a significant activity level, making it difficult to obtain a clean model with traditional techniques, such as mean, mode... These scenarios may be found in public buildings or outdoor areas, for instance, airports, subway or railway stations, entrance of buildings and so on, in which there are always people walking or standing.
Scenarios such as airports or railway stations are on duty 24 hours a day with a constant activity. In this kind of scenario it is difficult obtaining images without people of the areas under surveillance, in the case it had to be done in a concrete moment. But it is also desirable to get as soon as possible a good model in the case of model corruption.
Hardware is another of our constraints. Algorithms discussed in the following sections are designed to be implemented in a DSP-based hardware with a limited memory. Thus, storing a big amount of background models is not possible.
Two questions arise when talking about corrupt background models. How can a model be considered to be corrupt? From the algorithmic point of view, a measure of the quality of a model is needed in order to be able to detect how the process of model recovery evolves and a mechanism to detect corrupt models is also needed.
And, how may the quality of a model be measured? These questions are not yet given an answer in the literature. We propose measuring the quality by taking into account for how long a model has properly described a pixel.
Our approach tries to obtain a background model which can provide the system with a suitable segmentation and a correct classification of objects in the scene.
The solution we propose tries to answer the two questions aforementioned and also, give a general technique for background reconstruction. We propose a mathematical definition of model corruption in terms of number of pixels classified as foreground with respect to total number of pixels contained in the scene. This definition may, of course, be tailored for any situation.
The aims of the algorithm are:

(1) Construct a background model by acquiring frames no matter how many objects appear in the scene.
(2) Define a measure of the quality of the background model obtained and a confidence of the pixels classified as foreground.
(3) Avoid storing background models, in case of failure, the model will be recomputed on the fly.

(4) Compute a confidence value associated to the model for each pixel $b \in B$, in order to evaluate the security with which this pixel is classified as belonging to background. The higher the confidence of the model, the better the background model is.

In the following sections we develop an algorithm which may quickly reconstruct a background model in the case it is corrupt. Our scope is being able to build it even if people are present in the scene, meeting the first four constraints.

The fourth requirement is achieved by means of a definition for background model quality. One of the problems of the methods mentioned in the introduction is that they cannot determine whether a suitable background model is built or not. For instance, averaging 50 images of a scene with no people present in any frames (or present in a little amount of them) will produce, more or less, the same result as taking the average of just a couple of images.

Using simple statistical methods gives no hint of the quality of the model constructed. If moving objects are present in the frames used to construct the background, blurred areas may appear as a mixture of the values of the objects and the values of the background will be done. The quality index we propose, is mainly used to give a quantitative measure of the background model's quality. However, its use is not only limited to this, but also may be helpful when defining a segmentation quality using this model.

This chapter is organized as follows, section 2 introduces BAC the background adaptive modelling algorithm (Rosell –Ortega et. al. 2008). This first version of the algorithm uses gray tones to describe the scene. In section 3, we explore the possibility of using the same segmentation schema of BAC with RGB coordinates. In section 4 we compare BAC with the Stauffer's approach. Finally, section 5 is devoted to conclusions and future works.

2. Background adaptive modelling algorithm (BAC)

Background models are traditionally generated using statistical measures. In this chapter, we propose not to use only statistical properties of pixels, but also their behaviour, to build the model. As stated in the introduction, the aim of the algorithm is reconstructing or creating a background model from the scratch, with no previous assumption about the scene activity. Similarity with the background and motion criteria are used to determine how the model must be updated.

We propose an algorithm that considers consecutive gray scale frames F(0), F(1), ... F(n), in which any pixel $p_{x,y} \in F(i)$ must belong either to foreground or to background and builds a background model B starting from a frame $F(i), i \geq 0$. In this first frame it is impossible to classify pixels as background or foreground, as no further information is given. To decide which pixels may be used to update the background model and which not, a new similarity and motion criteria is defined in next sections.

Section 2.2 describes the notion of similarity with the background and motion of a pixel. We use the previous knowledge of how a background pixel should behave to discriminate which values in each incoming frame belong to background and which do not. In section 2.3 we explain the algorithm. Section 2.4 explains the experiments we made with different real videos, comparing the result of using our method with mean, mode and median to construct a background model and shows the results we obtained.

2.1 Similarity criteria

Similarity between two pixels is usually tested by comparing the difference of their gray levels with a threshold. We propose to translate into a function the intuitive idea behind "very similar" or "similar" by using a continuous function defined as,

$$S(p,q) = e^{\frac{-|p-q|}{\kappa}} \tag{1}$$

with $S : \Re \to [0,1]$, p and q are gray levels of two pixels, κ is a constant determined experimentally. This way, a difference degree and not an absolute value is calculated for pixels similarity. Figure 1 shows the evolution of this function.

2.2 Motion and similarity with the background

By using equation 1, it can be measured the similarity of each pixel with the background. Similarity between a pixel $q_{x,y} \in F(i)$ with a background pixel is then given by $S(q_{x,y}, b_{x,y})$, being $b_{x,y} \in B(i)$ the pixel in the background model.

Also, motion can be computed using equation 1. Motion of a pixel can be defined as its similarity with previous values of the pixel. Being $q_{x,y} \in F(t)$ a pixel in the current frame, $p_{x,y} \in F(i-1)$ and $r_{x,y} \in F(i-2)$; we define the motion of $q_{x,y}$ as,

$$M(q) = \frac{((1 - S(p_{x,y}, q_{x,y})) + (1 - S(r_{x,y}, q_{x,y}))}{2} \tag{2}$$

This way, motion in the scene is detected by considering similarities of three consecutive frames.

2.3 Segmentation process

The background algorithm with confidence (BAC) starts by taking a frame F(i) to be the initial background model B(i) (the model in time i), and sets,

$$\forall b_{x,y} \in B(i) \ , c_{x,y}(i) = 0 \wedge \sigma_{x,y}(i) = 0 \tag{3}$$

being $c_{x,y}(i)$ the confidence value of pixel $b_{(x,y)}$ and $\sigma_{x,y}(i)$ the filtered probability in time i.

Next two frames, F(i+1) and F(i+2), are ignored and used only to detect motion in frame F(i+3). For all the next incoming frames F(i), motion and similarities with B(i-1) are sought for. We define then the probability that any pixel $q \in F(i)$ belongs to foreground as,

$$P_{fore}(q) = \max(M(q), 1 - S(q,b)) \tag{4}$$

because pixels will belong to foreground if either their motion value is high or their difference with the background is high. This way, we can include in the foreground set all pixels which, even being similar to the background but show significant motion and vice versa.

On the other side, the following expression,

$$P_{back}(q) = \max(1 - M(q), S(q,b)) \tag{5}$$

defines the probability that a pixel $q \in F(i)$ belongs to background if both its motion value is low and its similarity to current background is high (as stated in the constraints described before). It must be noted that the relationship,

$$P_{back} + P_{fore} = 1 \tag{6}$$

does necessary verify. According to definitions of both probabilities, it is easy to see that the chosen value for each probability is complementary of the other one.

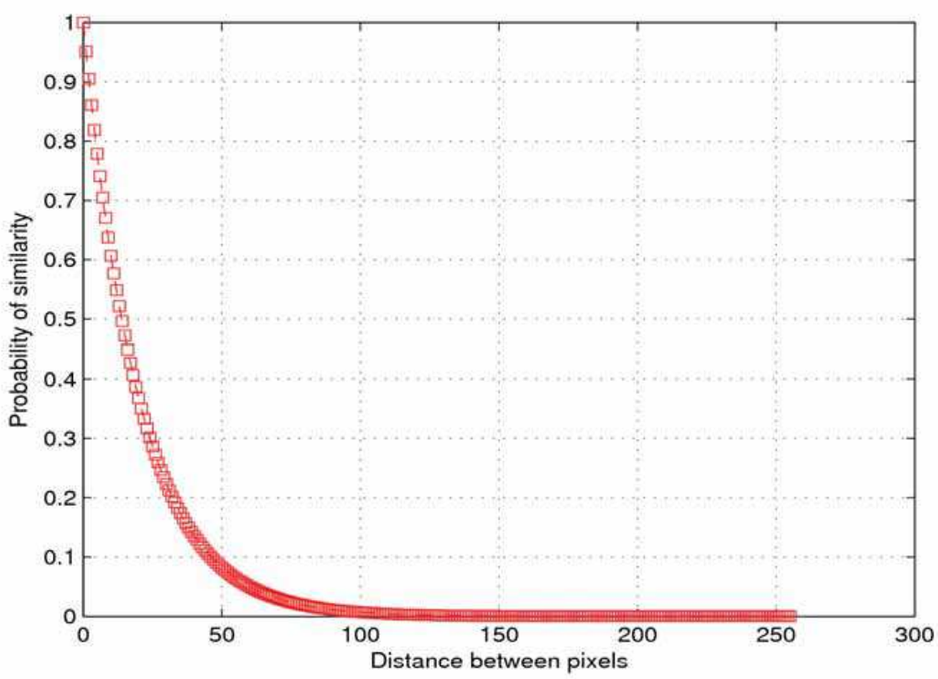

Fig. 1. Plot of function similarity for different distances.

Once F(i) is segmented, we must update the model B(i-1) to obtain B(i), using pixels in F(i). Not all pixels $b_{x,y} \in B(i-1)$ are updated in the same way, it depends on $P_{back}(b_{x,y})$, $c_{x,y}(i-1)$ and $\sigma_{x,y}(i)$. The segmentation separates pixels in two different sets; the foreground set (fSet) and the background set (bSet).

$$fSet = \{p_{x,y} \in F(\mathrm{i}) : P_{fore}(p_{x,y}) > 0.7\} \tag{7}$$

$$bSet = \{p_{x,y} \in F(\mathrm{i}) : p_{x,y} \notin fSet\} \tag{8}$$

We reduce the amount of pixels classified as foreground to only those whose foreground probability is high. This way, we get sure most of shadows will not be considered as foreground. On the other hand, anything which is not considered to be foreground, is classified as background.

The model probabilities are then updated,

$$\forall b_{x,y} \in B(i) : \sigma_{x,y}(i) = \frac{\sigma_{x,y}(i-1) \cdot c_{x,y}(i-1) + P_{back}(p_{x,y})}{(c_{x,y}(i-1)+1)} \quad (9)$$

being $\sigma_b(i)$ the filtered certainty of a pixel of belonging to background in time t. This probability is used, together with other measurements, to avoid that an object captured in the first model stays forever in the model. This filter accumulates the different background background probabilities obtained by a model pixel over time.

After filtering background probabilities, it may be seen that there are pixels whose probabilities diminish over time. For instance, this may be due to the fact that these pixels were still at the beginning of the process, and started to move later. But it may be also the case, that objects are moving over these pixels but they recover their values again after few frames.

In order to distinguish these two cases, two sets are added to the previous set definitions: pixels labelled as doubtful (dSet) and pixels in B(i) whose gray level will be replaced by the gray level of pixels in F(i) (cSet).

Doubtful pixels are those whose filtered background probability is under a threshold but whose confidence is still over a minimum. Recalling that the algorithm starts with zero knowledge about the scene, special care must be taken with such behaviour, in order to quickly change pixels which do not describe the background properly.

On the other side, pixels in the cSet represent pixels, whose confidence is under a threshold, and will be replaced by values from the current frame.

Equations 10 and 11 show how these sets are built. Doubtful pixels, dSet in equation 10, will be those which have a low filtered probability but still their confidence is high (at least, 80%). Equation 11 shows which are considered to be changed, those whose confidence is under 80% and also their filtered probability is low.

$$dSet = \left\{ p_{x,y} \in fSet : \sigma_{x,y}(i) < 0.8 \wedge \frac{c_{x,y}(i-1)}{c_{x,y}(i-1)+1} \geq 0.8 \right\} \quad (10)$$

$$cSet = \left\{ p_{x,y} \in fSet : \sigma_{x,y}(i) < 0.8 \wedge \frac{c_{x,y}(i-1)}{c_{x,y}(i-1)+1} < 0.8 \right\} \quad (11)$$

Values defining the previous sets were chosen to be very restrictive, this way, pixels which may yield low background similarity are quickly replaced. The regions of interest of frame F(i) are then defined by fSet. We define the following set for convenience,

$$C = bSet \cup dSet \quad (12)$$

We update pixels in a different way, depending on their observed behaviour. Those which have a high confidence and high filtered probability or their confidence is over a threshold, i.e., those which do not belong to cSet, are updated using the incoming values to cope with light changes.
Pixels which belong to the cSet are directly changed by values in the incoming frame. This way, regions which were labelled as foreground, become part of the background.

Being $q_{x,y} \in F(i)$,the model B(i) is updated as follows,

$$\forall b_{x,y} \in cSet : b_{x,y}(i) = q_{x,y}(i) \tag{13}$$

$$\forall b_{x,y} \in C : b_{x,y}(i) = q_{x,y}(i) + \alpha_{x,y}(i) \cdot \left(b_{x,y}(i-1) - q_{x,y}(i)\right) \tag{14}$$

Confidences of pixels are also updated distinguishing the set to which each pixel belongs to. In this case, pixels which do describe the background increase their confidence. Pixels whose $\sigma_{x,y}(i)$ reduces over time, do also reduce their confidence. On the other side, pixels which are copied from the image, take a confidence equal to zero.
As this operation is performed in a frame by frame basis, and pixels are reclassified after segmentation, any pixel whose confidence is reduced by a temporal occlusion by a foreground pixel will recover its previous confidence as soon as the occlusion finishes.

The confidence of pixels in B(i) is updated according to the following expressions,

$$\forall p_{x,y} \in bSet : c_{x,y}(i) = c_{x,y}(i-1) + 1 \tag{15}$$

$$\forall p_{x,y} \in dSet : c_{x,y}(i) = c_{x,y}(i-1) - 1 \tag{16}$$

$$\forall p_{x,y} \in cSet : c_{x,y}(i) = 0 \tag{17}$$

A difference with respect to other algorithms, is that we propose using a different adaption coefficient α for each pixel depending on the confidence they show. This way, we expect pixels which strongly described the scenario to update smoothly. On the other side, pixels whose confidence diminishes, recalling this means their background probability is descending, take a lower adaptation coefficient.
It is computed taking into account the confidence of the pixel in time i according to the following equations,

$$\forall p_{x,y} \in C : \alpha_{x,y}(i) = 0.98 \cdot \frac{c_{x,y}(i)}{\left(c_{x,y}(i) + 1\right)} \tag{18}$$

$$\forall p_{x,y} \in cSet : \alpha_{x,y}(i) = 0 \tag{19}$$

The adaptation coefficient takes values in the range [0,0.98). Being 0.98 the value which corresponds to pixels with a high confidence and 0 the value which corresponds to pixels which have been changed.

As said before, together with its gray level value, each pixel $b_{x,y} \in B(i)$ provides a confidence value which may be used to weight the quality of the segmentation. We define the segmentation confidence of the model B(i) as,

$$sc = \frac{1}{m \cdot n} \cdot \sum \frac{c_b(i)}{c_b(i)+1}, \forall b \in B(i) \tag{20}$$

being $m \cdot n$ the number of pixels of the model. The segmentation confidence (sc) is calculated for a target T_i with a size l in pixels of frame F(i), by particularizing this expression considering only the pixels segmented for this target.

Finally, in order to test when the background model is stable the mean square quadratic difference (msqd) between two consecutive models is calculated; the algorithm finishes if the following condition verifies,

$$msqd(B(i), B(i-1)) < 10^{-3} \wedge sc > 0.995 \tag{21}$$

2.4 Experiments

We made experiments to test two different issues. First, several random frames from test videos were chosen as the base to reconstruct the background model. We then compared the background model obtained with the BAC algorithm, with the one obtained by using median, mean and mode with the same frames used by BAC. Next experiments were aimed to control how accurate the segmentation was, by using BAC to segment frames while the algorithm was under construction.

Videos from different sources were used with the aim of reproducing different situations; videos recorded by ourselves, real videos from the airport and Wallflower benchmark. Videos had different lengths and were converted into grayscale when needed.

We compared the BAC's segmentation with a supervised segmentation in order to quantify the true positives (TP), which are pixels classified as foreground in the control image and by the algorithm, and the true negatives (TN), which are are pixels classified as background in the control image and by the algorithm. False positives (FP) and false negatives (FN) are defined as the complementary of the previous ones.

Good results were obtained with BAC, they may be found together with resulting models using mean aplied to test videos at www.vxc.upv.es/vision/proyectos/BAC.

A representative situation aim of our developments is analysed in this section. The video starts in F(0) with several people in a scene, simulating a surveillance system, in that moment B(0) is created with targets with sc=0, see figure 2 (a). In order to evaluate quantitatively the evolution of BAC, we segmented manually 22 frames randomly selected.

In table 1, segmentation results for frames F(90) and F(390) obtained with BAC and mean are compared; sc of pixels found in each target's segmentation with BAC is shown under column "confidence", for pixels not correctly segmented, sc was under 0.001. The original frames, together with segmentation result and the background model used may be found in figures 2,3 and 4.

	Frame 90			Frame 390		
target	BAC	mean	Model background confidence	BAC	mean	confidence
1 st	0.69	0.74	0.946	0.96	0.88	0.997
2 nd	0.90	0.70	0.977	0.89	0.71	0.996
3 rd	0.37	0.49	0.986	0.51	0.69	0.997
4 th	0.47	0.49	0.933	-	-	-

Table. 1. Percentage of pixels found for each hand-segmented target in control frames 90 and 390. Targets are not the same in both frames.

In F(90), the four objects present in the scene are segmented with BAC and mean; only those dark objects which are far in the field of view of the camera are segmented more poorly (target 3); something similar happens with target 4, which is a group of two people moving still in the same area they occupied at the beginning of the movie. We consider that with at least 45% of the total size of pixels detected of a target is sufficient to continue with classification and tracking tasks, if they are grouped in an only blob.

In figure 2 (a) and 2 (b) B(0) and B(89) are shown, it may be seen that in B(89), background model has achieved c = 0.982 and some targets have been removed. Improvement over time is evident as B(389) contains no target. This improvement manifests in F(390) with a better segmentation and a model with sc = 0.997.

Evolution of BAC's confidence, TP and TN, of BAC and mean are shown in figure 2. Objects standing still for long periods of time influence negatively the value of TP. The plot shows that BAC segments correctly more pixels than mean. In F(201) several objects leave the scene and others start coming in and in F(680) some objects stand still; this explains some foreground pixels not found. On the other side, TN, easily reach a high level as area of quiet targets is small compared to the image.

a) B(0)=0 b) B(90) = 0.982 c) B(390) = 0.997

Fig. 2. Background evolution, first figure correspond to background model in F (1). Figure (b) corresponds to the background model updated until F (90). Figure (c) corresponds to the background model in F (390).

(a) (b) (c)

Fig. 3. Different frames showing the evolution of people in the scenario. From left to right, images correspond to frames 1, 90 and 390.

(a) (b) (c)

Fig. 4. From left to right, result of background subtraction of frames 1, 90 and 390 using the background models computed so far.

(a) (b) (c)

Fig. 5. From left to right, the expected result for background subtraction for frames 1, 90 and 390. These images are segmented manually, labeling with white expected foreground and in black the background.

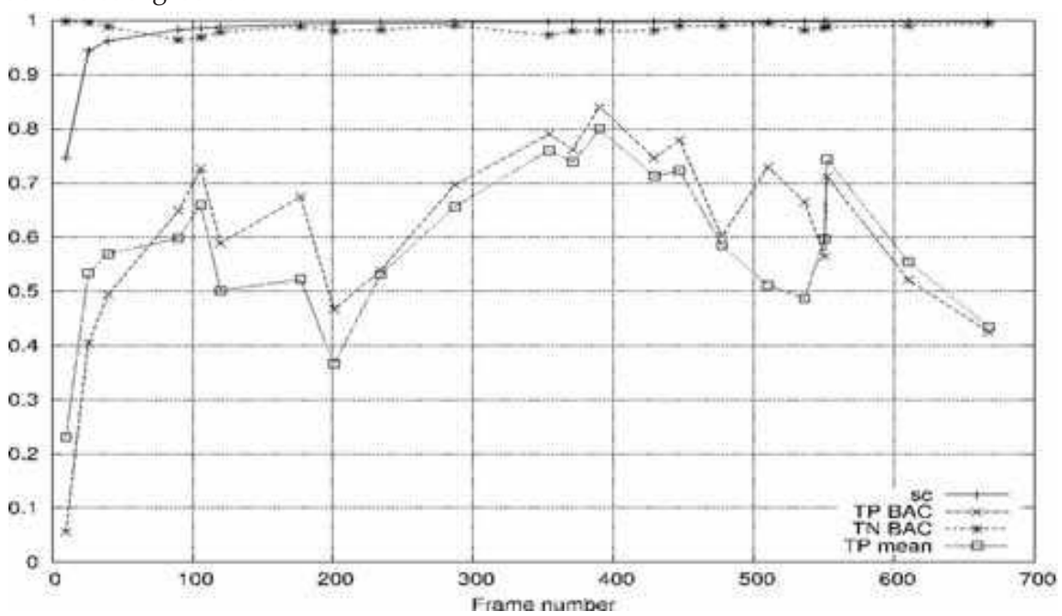

Fig. 6. Evolution of confidence, true positives and true negatives for the discussed video

Table 2 shows results for Wallflower benchmark. Values of true negatives are high, though videos were too short for BAC to converge. For sequence "wavingTree" sequence, fails due to the movement of the tree. In "lightSwitch", BAC was started in the moment lights were switched on.

Finally, in "bootstrap", brightness makes BAC fail to find correctly the targets, though it find most of the pixels associated to them.

Experiments show that BAC obtains background models equal to those obtained by using any statistical technique; with the added benefit of permitting segmentation from the very beginning of the process; as was the goal and together with a confidence measure of the obtained model. Also, the experiments performed with the Wallflower test set result promising. Our efforts should be address in near future to improve the response of the algorithm to shadows and brightness.

In the following sections, we extend the algorithm to improve segmentation results. We start by adding colour to the segmentation schema.

	bootstrap	camouflage	foreground	Light Switch	Moved Object	Time Of Day	Waving Trees
TP	0.48	0.73	0.51	0.44	------	0.36	0.73
TN	0.96	0.86	0.95	0.98	0.99	0.99	0.74

Table. 2. True positives and true negatives for the Wallflower benchmark.

3. BAC with colour

In order to improve results, one of the basic things that can be done is adding colour to the description of pixels. By doing this, better results for foreground and background are expected.

Choosing the colour coordinates depends on several factors. On one side, usually, cameras can produce images in RGB or YUV coordinates. Though mathematical methods exist that can convert coordinates of one system to another, this conversion may be very time consuming and have a severe impact on the system performance.

In the following sections, we develop BAC with colour by adding the use of RGB coordinates to the previous algorithm. Other systems exists, such as CIEL*a*b* or HSI which could be claimed to have better properties than RGB. We chose RGB system because it is a widely used system and it is easy to find cameras which reproduce images using this system.

3.1 Colour coordinates

There are different colour models that can be used in order to describe the colour of a pixel, RGB, HSI, CIEL*a*b*. CIEL*a*b*, for instance, has the advantage that is perceptually uniform. That means, that a change of the same amount in a colour value should produce a change of about the same visual importance. The distance between two colours represented in CIEL*a*b* coordinates is just the Euclidean difference of the two vectors representing them.

RGB on the other side, is not perceptually uniform because it was designed from the perspective of devices and not from a human perspective and CIEL*a*b*. Methods exist to convert the RGB coordinates into CIEL*a*b* coordinates and vice versa. In fact, the most common coordinates may be converted into each other through mathematical conversions.

In our case, we will use RGB coordinates. The similarity between two pixels, p and q is now given by a similar function as with gray tones. The only difference is that the distance between the pixels is extended to use the three RGB coordinates.

In this case, equation (1) is modified and the similarity function is given by,

$$S(p,q) = e^{-|dist|/\kappa} : \Re \to [0,1] \tag{22}$$

where the colour distance is computed as the Euclidean distance of two colours as in equation 23 and κ is a constant determined experimentally.

$$dist = \sqrt{(p_R - q_R)^2 + (p_G - q_G)^2 + (p_B - q_B)^2} \quad (23)$$

3.2 Experiments

Experiments were performed with the same benchmark as with BAC in order to compare results. It is obvious that adding colour to the image processing will improve results, as more information is being used in the segmentation.

Results for the Wallflower benchmark are shown in table 3. In this case, the improvement is evident, a bigger rate of foreground pixels is obtained in all sequences. The chosen segmentation seems to be very sensitive and a lower rate of background pixel is achieved in the sequences.

	Bootstrap	Camouflage	Foreground	Light Switch	Moved Object	Time Of Day	Waving Trees
TP	0.67	0.78	0.58	0.46	-	0.53	0.93
TN	0.85	0.70	0.87	0.97	0.99	0.98	0.59

Table. 3. True positives and true negatives for the Wallflower benchmark using colour in the pixels' description.

4. Comparison with other approaches

We compared the performance of the original BAC algorithm with another approach introduced in the paper by Stauffer and Grimson (Stauffer & Grimson, 1999). We implemented their algorithm and executed it with different parameters in order to seek for best results.

As in the other experiments, we used the Wallflower test for comparisons. We found some difficulties when trying to deal with shadows with this algorithm. Also, we used the parameters which seemed to be the best, keeping them the same for all sequences, as we made with BAC.

	Bootstrap	Camouflage	Foreground	Light Switch	Moved Object	Time Of Day	Waving Trees
TP	0.33	0.80	0.59	0.76	---	0.24	0.66
TN	0.97	0.62	0.55	0.08	1.00	0.99	0.85

Table. 4. True positives and true negatives for the Wallflower benchmark using the Stauffer & Grimson algorithm with a maximum number of models equal to 5, and T= 0.8.

Stauffer's algorithm uses several models per pixel to model the scene, and that is a big difference when motion in the background appears as in camouflage sequence or wavingTree sequence. Results for these two sequences outperform clearly BAC.
Despite results with Stauffer's algorithm could be improved with another set of parameters or initialization, the sequence that better illustrates the performance of BAC is lightSwitch.
In this sequence, a sudden change in light in the scenario is applied. The background rapidly changes from a dark room to an illuminated room.
This sudden corruption of the scene is caught by BAC, which quickly restarts the model. Stauffer's algorithm is slower when reducing the weights of the model to include the new model.

5. Conclusions and future work

A different approach to background modelling was introduced in this chapter. The aim of the algorithms developed is trying to give a quick response to two different problems with a common solution: building a background model and recovering a background model in demanding scenarios.
These scenarios are characterized by having always a significant activity level, making it difficult to obtain a clean model with traditional techniques. Results for the Wallflower benchmark and for the test videos result promising.
Several algorithms have been developed in order to meet the constraints we were facing. First algorithm, MBAC is the simplest of them. Its aim is building a background model and associates a confidence to it, in order to have a numerical description of how good the model is. This algorithm uses gray tone levels to describe the scene.
By adding colour to the BAC algorithm, more accuracy in the background description and the segmentation process is achieved. Results show that, as it was expected, the version of BAC with colour improves results. Other colour systems exists, such as CIEL*a*b* or HSI which could be claimed to have better properties than RGB. We chose RGB system because it is a widely used system and it is easy to find cameras which reproduce images using this system.
The reconstruction of background models on the fly proved to be useful for demanding scenarios, in which it may be difficult achieving good quality background models with traditional techniques. We tested the algorithm in several situations with test videos from different sources, we set a web-site where videos showing algorithm evolution is illustrated.
Further research should be done in improving the segmentation to include also shadows, which proved to be very difficult to classify with our method. Also, a review of the segmentation process should be done. Maybe the fact that RGB coordinates are not perceptual uniform affect the computation of distances and produces a high amount of missed background pixels

6. References

A. Elgammal, D. Harwood & L.S. Davis "Non-parametric model for background subtraction". *ECCV 2000*, pages 751 – 767. 2000.

I. Haritaoglu, D. Harwood & L. S. Davis "W4: Real-time Surveillance of people and their activities". . I*EEE Transactions on PAMI*. vol 22, num. 8, pages 809 – 830. 2000

M. M. Heikkilä & M. Pietikäinen. "A texture-based method for modeling the background

and detecting moving objects" *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 28(4):657 - 662, April 2006.

W. Hu, T. Tan, L. Wang & S. Maybank.] "A Survey on Visual Surveillance of Object Motion and Behaviors". I*EEE Transactions on Systems, man, and Cybernetics*, vol 34, num.3. 2004

S. Jabri, Z. Duric, H. Wechsler & A. Rosenfeld. " Detection and location of people in video images unsing adaptive fusion of color and edge information" *Proc. International Conference Pattern Recognition,* pages 627 - 630, 2000.

M. Mason & Z. Duric " Using histograms to detect and track objects in color video". *Proc. Applied Imaginery Pattern Recognition Workshop,* pages 154 - 159, 2001.

J. Rosell-Ortega, G. García-Andreu, A. Rodas-Jordà, V. Atienza-Vanacloig. "Background modelling in demanding situations with confidence measure" *International Conference on Pattern Recognition*. ICPR08. 2008.

B. Shoushtarian & H.E. Bez. "A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking". *Pattern Recognition Letter*s 26, pages. 5 – 26. 2005

C. Stauffer & W. E. L. Grimson. " Adaptive background mixture models for real-time tracking" *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition,* pages 246 - 252, 1999.

K. Toyama, J. Krumm, B. Brumitt & B. Meyers. "Wallflower: Principles and Practice of Background Maintenance". *7th Intl. Conf. on Computer Vision*, Kerkyra, Greece. pages 255-261. 1999

L. Wixson. " Detecting salient motion by accumulating directionally-consistent flow". *IEEE Trans. Pattern Analysis and Machine Intelligence*, (8):774 - 780, 2000.

L. Wang, W. Hu & T. Tan. "Recent developments in human motion analysis". *Pattern Recognition*. pages 585-601,Vol .3 2003.

C. R. Wren, T. D. A. Azarbayenjani, & A. P. Pentland " Pfinder: rel-time tracking of the human body". *IEEE Trans. Pattern Analysis and Machine Intelligence*, (7):780 - 785, 1997.

Q. Zang & R. Klette. " Robust background subtraction and maintenance**"** *Proc. International Cong. Pattern Recognition***,** pages 90 – 93, 2004.

Pattern Recognition
Edited by Peng-Yeng Yin

ISBN 978-953-307-014-8
Hard cover, 568 pages
Publisher InTech
Published online 01, October, 2009
Published in print edition October, 2009

For more than 40 years, pattern recognition approaches are continuingly improving and have been used in an increasing number of areas with great success. This book discloses recent advances and new ideas in approaches and applications for pattern recognition. The 30 chapters selected in this book cover the major topics in pattern recognition. These chapters propose state-of-the-art approaches and cutting-edge research results. I could not thank enough to the contributions of the authors. This book would not have been possible without their support.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen