Active Object Detection in Dynamic Scenes

Sérgio Miguel dos Santos Copeto
sergio.copeto@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

April 2012

Abstract

This paper presents an object detection algorithm for video surveillance systems that can handle sudden illumination changes. The proposed system keeps multiple background models, each one representing a certain lightning condition. The system performs the identification of the active lighting condition for each incoming frame. The system can also identify unknown lighting conditions and add them to the background model collection, as well as perform updates to the existing models. For this paper, a set of test videos, recorded at the Culturgest auditorium, were captured and studied using the RGB, HSV and L*a*b* colorspaces. The system was tested in terms of the active background model identification and active region segmentation, for both constant and varying illumination. It was also made a comparison with a classical model, which concludes that the proposed system presents better performance, particularly in the case of sudden light changes. **Keywords:** Video Surveillance Systems, Dynamic Ambient, Sudden Illumination Changes

1. Introduction

This paper presents an object detection system for fixed cameras that is robust to changing sudden illumination changes. The slow changes, such as the position of the sun throughout the day are well compensated by the different methods proposed before. However, the main difficulty of such systems has to do with abrupt changes in lighting conditions that occur, in most cases, in indoor environments, where different illumination sources are turned on or off Existing methods are not robust enough when it becomes necessary to deal with such situations. The solution proposed in this paper is to collect multiple models for the background image, each representing a different lighting condition. The system is able to identify the illumination scheme at each incoming frame, and select the best background model to be used in the segmentation phase. Furthermore, the system is able to identify changes in lighting conditions, and if it is an unknown type of lighting start a learning process of the new scene, automatically managing the set of existing models. This approach differs from existing models as the background image is interpreted globally, with a contribution of all pixels to identify the type of lighting in the scene, and also because there are two ways to update existing images, one serving to update the existing background, the second for background learning purposes.

2. Related Work

This section summarizes the various techniques for detection of active objects in video sequences captured by static cameras. The proposed problem can be divided into several subsections: background modeling, background updating and foreground segmentation.

2.1. Background Modeling

In the video-surveillance applications, the first step is to obtain a model for the scene (frequently called background image), which will then be used to perform the foreground segmentation. The easiest way is to acquire an image representing the static elements of the scene. However, several other background modeling methods were studied, being statistical modeling the most widely used technique. In statistical modeling, the simplest situation is to represent each pixel information by means of a random variable with gaussian probability distribution [7]. The use of probabilistic distribution allows noise to be represented as a random variable whose variance can be estimated. Moreover, the single gaussian model was found not robust enough and was extended. In [5], each pixel is modeled using a weighted mixture of gaussian distributions. This model is more suited to outdoor scenes, where there is the presence of waving tree leafs or periodic reflections caused by water rippling. Another alternative is to propose that the probability density function of a given pixel is estimated using a kernel density

function through a set of training images [2]. This model is better suited to outdoor environments, but exhibits bad performance when dealing with sudden light changes.

2.2. Background Model Update

The images acquired by the surveillance systems change over time due to illumination changes, presence of clouds, or scene variations (cars in a parking lot as an example). Thus, the background model must be updated so the system can handle the scene variations. In [7], each pixel in the background model is updated computing a weighted mean of the actual background model and each acquired frame. [5] proposes a method for the gaussian mixture model where the incoming pixel value is matched with the most probable gaussian, which is then updated. Another method is proposed in [1], in which the background model is updated using a temporal median filtering over a set of frames acquired over time, sequentially or with a certain sampling frequency, increasing the model stability.

2.3. Object Segmentation

Foreground segmentation can be achieved using the background model, finding significant deviations between the pixel values of the incoming frame and the corresponding value of the background model, which is frequently called Background Subtraction. These methods became popular with the work developed in [7], where a certain pixel is classified as foreground if it's probability related to the background image is not sufficiently high. Regarding this idea, [5] proposes that a pixel is classified as foreground if it's value doesn't fit any of the gaussians of the mixture. Besides these methods, other approaches were used using kernel density estimation [2] [3] or state space models [4] [6].

3. Implementation

The existing background modeling methods uses only one image for the background. Even using a gaussian mixture model, they are not sufficiently robust when there are sudden changes of lighting conditions. As such, this paper uses a multiple model approach for the background, assuming that the system comprises a set of J background models, M^j, each one representing a certain lightning condition. Each pixel of the model is represented using a single gaussian distribution

$$M^j(x) \sim \mathcal{N}(B^j(x), (\sigma^2)^j(x)) \tag{1}$$

where $B^j(x)$ is the mean value of $M^j(x)$, called the background image j, and $\sigma^2(x)$ is the variance of the distribution of $M^j(x)$. The mean and the variance of the probability distribution are estimated using previous acquired images. For each incoming frame, the system selects the background model that better describes the lightning conditions of the scene. The identification method is described in the following section.

3.1. Background Identification

Given the J background models, the probability that a given pixel of the incoming frame in a certain time instance, $I(t,x)$, contains the same information than a given background model, M^j, for the color channel c is given by:

$$\begin{aligned} P_c^j(x) &= P\{I_c(t,x)|M^j(x)\} \\ &= \mathcal{N}\{I_c(t,x)|B_c^j(x), (\sigma_c^2)^j(x)\} \end{aligned} \tag{2}$$

The probability given by equation 2 is computed for every pixel in all three color channels. The next step is to obtain a similarity measure between the aquired frame and the model M^j, named $d^j(x)$. This value is computed by a probabilistic combination of P_c^j, assuming that the probabilities computed are independent for all three color channels.

$$d^j(x) = P_{c_1}^j(x) P_{c_2}^j(x) P_{c_3}^j(x) \tag{3}$$

where c_1, c_2 and c_3 are the three color channels of the colorspace.

Given the value of d^j for all pixels, the similarity measure between $I(t)$ and the model M^j is defined as

$$D^j = \prod_X d^j(x) \tag{4}$$

and the active background model for the acquired frame, $B(t)$, is computed as:

$$B(t) = B^j : j = \arg\max_j D^j \tag{5}$$

3.2. Background Update

Having multiple models for the background image allows the system to identify the kind of illumination in a scene. However, these models must be updated due to slow changes in the background illumination scene. The models M^j are updated by temporal median filtering considering a moving window, W_N, consisting of N images acquired with a T_a sampling time. for each pixel, the value of B^j is updated in a pixel-by-pixel basis:

$$B_c^j(x) = \text{median}_{n=1,\ldots,N}(W_n)_c^j(x) \tag{6}$$

Considering the variance $(\sigma^2)^j$ and having in mind that the images in W_N may have active objects in the scene, it's value can't be updated using the most common methods for the variance computations. As such, the value of $(\sigma^2)^j$ is obtain through a median absolute deviation estimator, MAD, for each pixel

$$MAD_c^j(x) = \text{median}_{n=1,\ldots,N} \mid (W_n)_c^j(x) - B_c^j(x) \mid \tag{7}$$

and the final value for the variance of each pixel distribution is given by

$$(\sigma^2)_c^j(x) = (1.4826[MAD_c^j(x)])^2 \tag{8}$$

3.3. New Model Creation

The background model selection described in 3.1 is valid when the best background model M^j sufficiently describes the scene's illumination conditions. However, it may occur that a certain light scheme isn't considered in any background model. This situation is evaluated using

$$\forall_{j \in J} : D^j < (\eta_I)^{XC} \tag{9}$$

where η_I is the minimum similarity threshold between the best background model and the incoming frame, X and C are the total number of pixels and the total number of color channels. In this case, the system creates a new background model to better describe the new lightning condition. Then, the system starts a background learning phase, where all the needed information for the new model estimation is stored in a temporal background model, M^*, always selected during the background learning process. Since the model must be built in the shortest possible time interval, the frames are stored in the temporary window, W_N^*, with a lower sampling time than in the normal model update case

$$T_b < T_a \tag{10}$$

until all the frames needed to fill the temporary moving window are being acquired, the B^* value is updated using a weighted mean computation

$$B_c^*(x) = \alpha I_c(t,x) + (1-\alpha) B_c^*(x) \tag{11}$$

where α, $\alpha \in [0,1]$, is a learning rate. This way it's guaranteed that the segmentation phase is processed using the best possible background model in the learning phase.

Once all the needed frames are stored in the temporary window, the model (B^* and $(\sigma^2)^*$) is then estimated using the model update described in section 3.2. The temporary model is then compared with the existing models in the collection using

$$P^j = P\{B^* | M^j\} = \mathcal{N}\{B^* | B^j, (\sigma^2)^j\} \tag{12}$$

Computing the value of D^j for all the background models (see section 3.1), the temporary model is added to the collection if

$$\forall^{j \in J} : D^j < (\eta_M)^{XC} \tag{13}$$

where η_M is the minimum similarity threshold between background models. After that, the learning phase is terminated, having the sampling rate to the moving window back to it's original value, T_a.

3.4. Foreground Segmentation

Object segmentation is performed using the basic background subtraction method between the incoming frame $I(t)$ and the active background model $B(t)$. For the sake of better segmentation performance, subtraction is done separately for each color channel of the images. The subtraction between the two images is computed as:

$$S_c(x) = |B_c(t,x) - I_c(t,x)| \tag{14}$$

The subtraction result for the three channels is then combined using logical combination. First, the binarization of S_c is computed

$$F_c(x) = \begin{cases} 1, & \text{if } S_c(x) \geq \delta_c \\ 0, & \text{if } S_c(x) < \delta_c \end{cases} \tag{15}$$

where δ_c is the segmentation threshold for the image channel c. The final segmentation mask is then computed assuming that a certain pixel is classified as foreground if it's classified as foreground in, at least, one of the image channels.

$$F(x) = F_{c_1}(x) \vee F_{c_2}(x) \vee F_{c_3}(x) \tag{16}$$

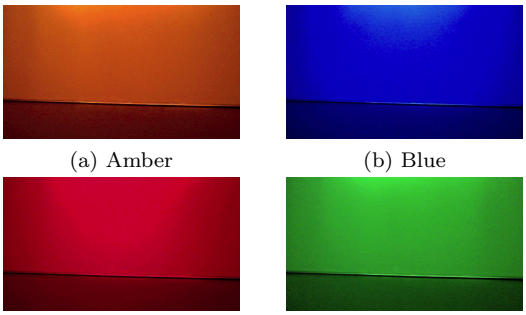
4. Dataset

This section presents the dataset used to test the proposed system, composed of five video sequences recorded at the Culturgest's auditorium stage, and describes some tests made to the recorded sequences, regarding color stability, noise estimation and pixel value variation in presence of active objects.

4.1. General Description of the Dataset

The scenario is constituted by the base of the stage and a white screen. Within are designed different types of lighting, illustrated in Figure 1, changing instantly over time, with in total four different illumination types: amber, blue, red and green.

The video sequences were acquired considering increasing difficulty objectives. As such, the first sequence consist in the study of the background model identification, as there are no active objects in the scene during all the sequence. The second and third sequences evaluates the segmentation with static background with the presence of one and two active objects, respectively. The last two sequences evaluates foreground segmentation with sudden light changes during active objects presence in the scene. Table 1 resumes the main characteristics of the acquired video sequences.



(a) Amber (b) Blue

(c) Red (d) Green

Figure 1: Lightning conditions used in the video sequences

Sequence Number	Number of frames	Number of Objects	Light transitions
1	6060	0	No
2	6109	1	No
3	2762	2	No
4	6112	1	Yes
5	2648	2	Yes

Table 1: Summary description of the recorded video sequences

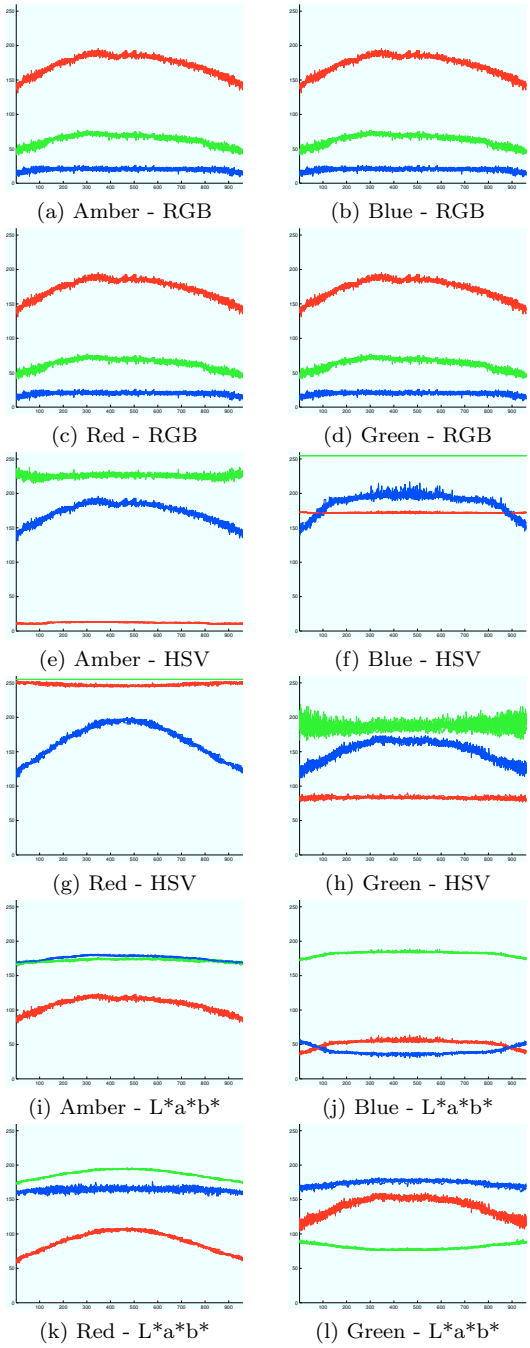
4.2. Color Stability

Studying color stability gives an idea of which color space can provide a more robust selection of background model. For each illumination type, a set of 10 frames were acquired with a sampling time of 1 second. From each image set, the color values of three horizontal lines were plotted for each color space, has been chosen lines 50 (top of the white screen), 250 (middle of the image) and 480 (stage base).

An example of the results is shown in figure 2, where a higher vertical dispersion represents less color stability. Considering vertical dispersion, it's possible to say that the L*a*b* color space is the one with better color stability. Regarding the four light conditions, the one with better results are seen in the red illumination scheme, and the green scheme is the less stable. One important thing to take in account is that in cases when the color saturation is lower, the uncertainty of the color value increases, causing higher noise. Still, L*a*b* color space gets better results in this case.

4.3. Noise Estimation

The purpose of this section is to examine how the noise varies across space, and if this variation is related to certain areas or color spaces. This test can also be useful to estimate a reasonable segmentation threshold, described in section 3.4. The noise uncertainty is obtained by computing the pixel by



(a) Amber - RGB (b) Blue - RGB

(c) Red - RGB (d) Green - RGB

(e) Amber - HSV (f) Blue - HSV

(g) Red - HSV (h) Green - HSV

(i) Amber - L*a*b* (j) Blue - L*a*b*

(k) Red - L*a*b* (l) Green - L*a*b*

Figure 2: Example of the color stability for line 270. The red, green and blue lines represent the information on the first, second and third color channels of each color space

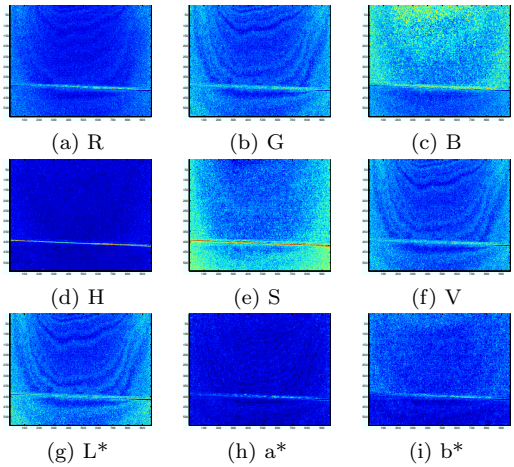


Figure 3: Noise standard deviation for the green illumination scheme. Dark-blue represents low noise values, while red values represent high noise values.

pixel standard deviation of the acquired images. The same set of images acquired for the previous section was used. For each pixel, an average of the test images is computed, i.e.:

$$\overline{I_C(x)} = \frac{1}{N}\sum_{n=0}^{N} I_c(n,x) \tag{17}$$

The variance for each pixel is obtained through

$$(\sigma_c)^2(x) = \frac{1}{N}\sum_{n=0}^{N}(I_c(n,x) - \overline{I_c(x)})^2 \tag{18}$$

and the standard deviation is obtained using the variance.

$$\sigma_c(x) = \sqrt{(\sigma_c)^2(x)} \tag{19}$$

An example of the noise distribution is shown in figure 3. It is concluded that the noise values are not uniform across the image, and it also depends on the illumination scheme. Furthermore, there is less noise in areas with an higher value of color saturation or light intensity.

Since the images do not give us a quantitative measure of the noise value, the mean values of the noise for each color space and each lighting situation are computed, according to

$$\overline{\sigma_c} = \frac{1}{X}\sum_{x=1}^{X}\sigma_c(x). \tag{20}$$

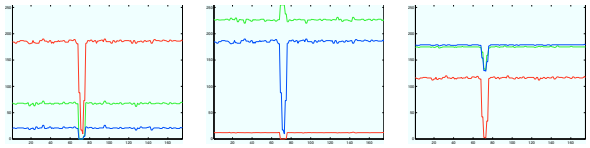


Figure 4: Change in one pixel values due to the presence of an object for the amber illumination scheme. The red, green and blue lines represent the information on the first, second and third color channels of each color space

Color space	RGB	HSV	L*a*b*
Amber	0.093	0.138	0.047
Blue	0.094	0.071	0.054
Red	0.076	0.044	0.051
Green	0.136	0.167	0.074

Table 2: Noise mean values for each color space and illumination scheme.

Table 2 shows the noise values computed using (20) and normalized between 0 and 1. L*a*b* colorspace is the one that gets the best results, it is expected that foreground segmentation also obtain best results for this color space. From the table values, it's possible to define a value for the segmentation threshold, σ_c, defined in section 3.4. Since digital information for the pixel values are between 0 and 255, the value of 25 was chosen for σ_c.

4.4. Active Object Reaction

For this step, four video sequences were analyzed, in which there is the presence of active objects. The sequences are subsequences of the second video stream, each one representing a different lighting condition. All subsequences having a length of 175 frames.

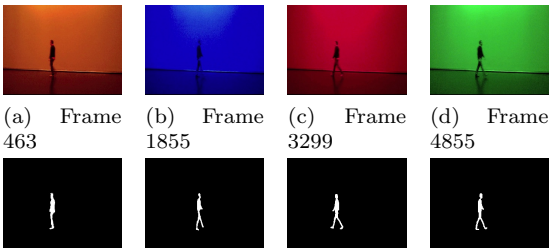
A pixel is then selected, common to all the video sequences, and a temporal representation of it's value is made for each type of illumination and each color space. The pixel of coordinates (300,480) was chosen, as it is located substantially in the middle of the image.

Figure 4 shows the results for the amber illumination scheme. It appears that there are considerable differences in the pixel's value, so object detection can be done using any of the color spaces. However, it is possible to note that the color space L*a*b* may be more appropriate for that purpose, since it's value doesn't saturate in the case of object detection.

5. Results

This section aims to evaluate the proposed system for non-stationary light conditions. The first test

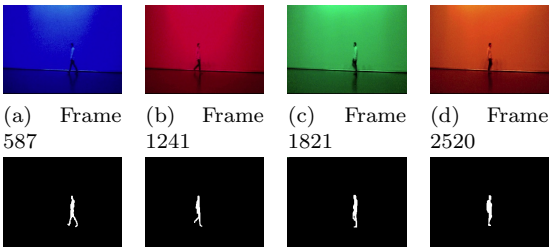
consists in the analysis of the lighting condition identified by the system for each frame, comparing with the expected value. Then, the foreground segmentation results are evaluated for the case of constant illumination and for the case where there are changes in illumination. In both tests, a series of images corresponding to segmentation result performed by the system at different time instants are collected, for all lighting conditions and color spaces. The last step is to compare the segmentation result of the proposed system with the results from the Gaussian mixture model [5]. In each case, the system output is compared with a ground truth segmentation, whose examples used are illustrated in figures 5 and 6. The associated labels are evaluated using a confusion matrix. It is important to note that in these tests, the system has no prior knowledge of any background image and it is assumed that the maximum number of background models that the system can handle is 4.



(a) Frame 463 (b) Frame 1855 (c) Frame 3299 (d) Frame 4855

(e) Ground Truth 463 (f) Ground Truth 1855 (g) Ground Truth 3299 (h) Ground Truth 4855

Figure 5: Ground truth for the static illumination examples



(a) Frame 587 (b) Frame 1241 (c) Frame 1821 (d) Frame 2520

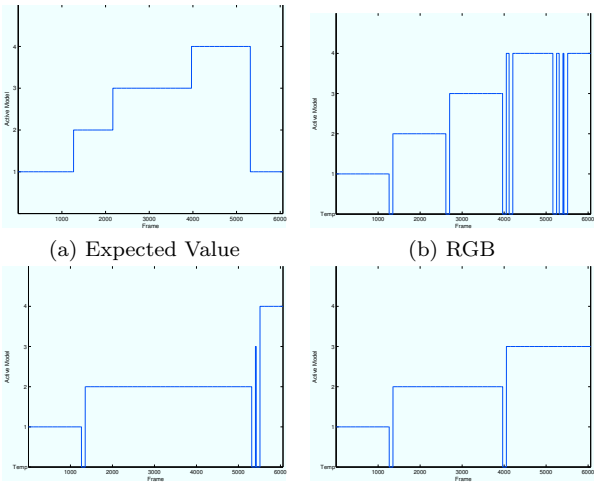
(e) Ground Truth 587 (f) Ground Truth 1241 (g) Ground Truth 1821 (h) Ground Truth 2520

Figure 6: Ground truth for the sudden illumination change examples

5.1. Active Background Model Identification

In this test, the first video stream was used, since there is no presence of active elements. This sequence is analyzed using all the color spaces in the study. There are four important parameters for the active background management algorithm, as shown in section 3

- T_a - Sampling time for the existing background model update.



(a) Expected Value (b) RGB

(c) HSV (d) L*a*b*

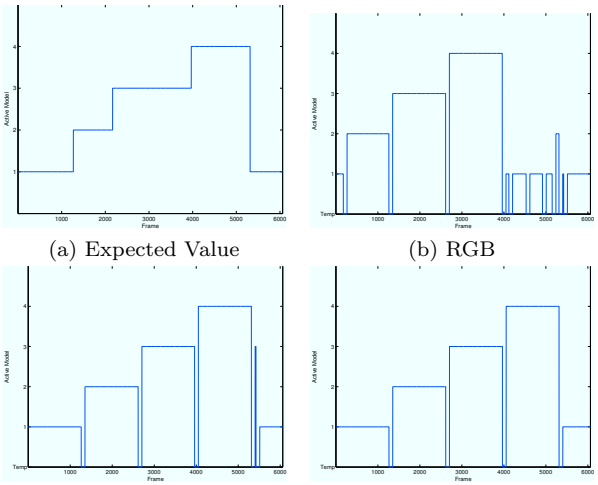
Figure 7: Active background model indentified at each frame, for $\eta_I = 70\%$ and $\eta_M = 90\%$

- T_b - Sampling time for the temporary background model creation.
- η_I - Minimum similarity threshold between the best background model and the incoming frame.
- η_M - Minimum similarity threshold between the temporary background model and existing background models.

We chose a sampling interval T_a of 15 frames for the case of models belonging to the collection and a sampling interval T_b of 8 frames for the case of the temporary background creation. Regarding η_I and η_M, two test were made, one of them using the following values:

- η_I - 70%
- η_M - 90%

Results for the fist test are shown in figure 7, where values between 1 and 4 described in the graphs indicate the active model at each moment, and the value 0 indicates that the system detects a change in lighting conditions, starting the background learning phase. Aparently, when there is a change in lighting conditions from blue to red and from red to green, these changes are not recognized in the HSV and L*a*b* color spaces. This means that the value chosen for η_I is low, making the system insensitive to significant lightning changes. In addition, the transition from green to amber in the RGB color space, the system interprets as a new illumination scheme, which means that the value chosen for η_M must be reduced.



(a) Expected Value (b) RGB

(c) HSV (d) L*a*b*

Figure 8: Active background model indentified at each frame, for $\eta_I = 80\%$ and $\eta_M = 80\%$

As such, a second test was made using the following values:

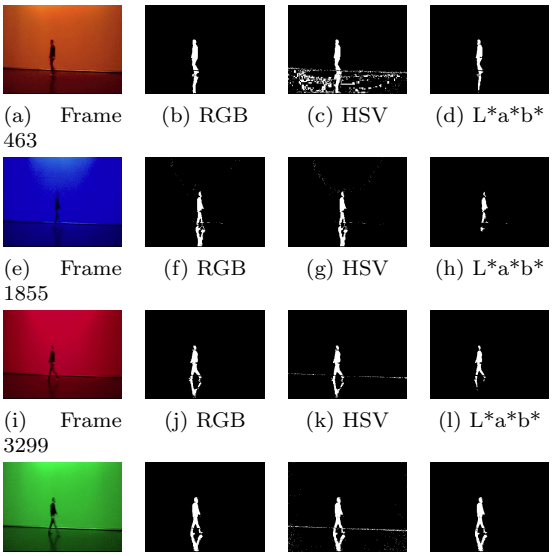
- η_I - 80%
- η_M - 80%

Results are shown in figure 8. Given the results for both tests, we conclude that the second set of values produces better results. It's visible that the RGB color space has a lot of instability, mostly due to the presence of noise. The first emerging identification problem occurs between frames 1 and 1000, during which the algorithm identifies a second background image, despite the light conditions are constant. The system correctly detects the active background model when using the L*a*b* color space with the used parameters. Regarding the HSV color space, the identification is also done properly, except for the last transition (green - amber), which is subsequently corrected and the expected model is identified.

5.2. Object Segmentation with Static Background

The frames to be analyzed must be chosen so that the active background is fully estimated and the object to identify must be fully visible, preferably in the middle of the frame. Having said that, frames 463, 1855, 3299 and 4855 were chosen.

Segmentation results are shown in figure 9. As provided in the previous chapter, the L*a*b* color space is one that gets the best results, mainly because it has less noise than the other color spaces under study, but also because it is more stable at the color interpretation. The RGB color space also produces good results except for the blue illumination scheme, in which noise is recognized as foreground. However, this situation can be solved using morphological operations on the binary image obtained.



(a) Frame 463 (b) RGB (c) HSV (d) L*a*b*

(e) Frame 1855 (f) RGB (g) HSV (h) L*a*b*

(i) Frame 3299 (j) RGB (k) HSV (l) L*a*b*

(m) Frame 4855 (n) RGB (o) HSV (p) L*a*b*

Figure 9: Segmentation results for static background.

5.3. Object Segmentation Under Illumination Changes

In the choice of frames to be collected, it is necessary to take into account that they must belong to the background learning phase. Frames were collected about a second after having a change in illumination. Therefore, we chose frames 587, 1241, 1821 and 2520.

An important variable for this test is learning rate α for the temporary background image as defined in 3.3, since it is necessary that light variations are quickly included in the background image, suggesting large value for α, but not too high so that active objects are also included in the background. Therefore, we chose 0.15 as the learning rate α.

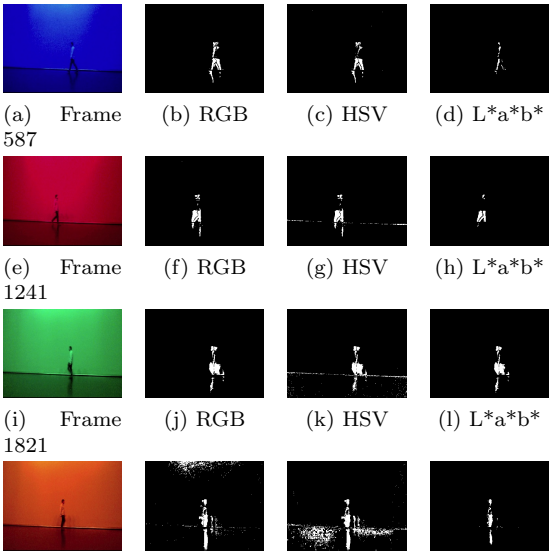
Figure 10 shows the segmentation results under illumination changes. Since moving object are present at illumination change, it is natural that the segmentation does not provide as good results as in the case of constant illumination. Still, most of the pixels identified as foreground belong to the area where the object is located, although this is not fully segmented. Another positive fact to take into account is that, in most cases, the regions where there is no presence of objects contain no pixels classified as foreground, taking cases of frame 2520 for the RGB and HSV color spaces, and the frames 1241 and 1821 for the HSV color space. Considering our results, it can be considered that the chosen value for α is suitable for this situation.

		RGB Classification		HSV Classification		L*a*b* Classification	
		Negative	Positive	Negative	Positive	Negative	Positive
Expected Value	Negative	98.84 %	1.16 %	97.37 %	2.63 %	99.35 %	0.65 %
	Positive	5.88 %	94.12 %	5.65 %	94.35 %	12.24 %	87.76 %

Table 3: Confusion matrices for the proposed method under static illumination.

		RGB Classification		HSV Classification		L*a*b* Classification	
		Negative	Positive	Negative	Positive	Negative	Positive
Expected Value	Negative	99.83 %	0.17 %	99.60 %	0.40 %	99.88 %	0.12 %
	Positive	20.08 %	79.92 %	27.93 %	72.07 %	41.26 %	58.74 %

Table 4: Confusion matrices for the Gaussian Mixture Model under static illumination.



(a) Frame 587 (b) RGB (c) HSV (d) L*a*b*

(e) Frame 1241 (f) RGB (g) HSV (h) L*a*b*

(i) Frame 1821 (j) RGB (k) HSV (l) L*a*b*

(m) Frame 2520 (n) RGB (o) HSV (p) L*a*b*

Figure 10: Segmentation results under illumination change.

5.4. Comparison with the Gaussian Mixture Model

In order to evaluate system performance, a comparison between our method and a classical method is made. The chosen model was the Gaussian Mixture Model [5], as it is widely used and show good performance. All the parameters that have to do with the foreground segmentation are initialized with the same value, ie, threshold value of 25 is used both for the developed system and the Gaussian Mixture Model. Two tests are made, in which a set of foreground masks is acquired. The first test uses the second and the third video sequences, selecting 8 foreground masks, 2 per light condition considering a static background situation. The second test regards the fourth and fifth video sequences and, for each light scheme, a foreground mask is acquired corresponding to 1 and 2 seconds after a light change. For each test, the system output is compared with a ground truth segmentation, which returns 1 for foreground pixels and 0 for background pixels. Then, a confusion matrix is obtained, computing the true positive, true negative, false positive and false negative rates.

Tables 3 to 6 shows the confusion matrices for the tests made. For the case of static background, it's possible to see that the segmentation resulting from the purposed method has a better performance regarding the identification of active elements, as it's true positive rates are higher than the ones obtained with the classical method. On the other hand, the false positive rates is lower in the classical method, due to its lower sensibility. Regarding segmentation under light changes, the results are satisfactory, as the true positive rates are clearly higher in the proposed method, and because the false positive rates are also lower, which means that most of the possible identifications due to changes in light conditions are absorved by the temporary background model.

		RGB Classification		HSV Classification		L*a*b* Classification	
		Negative	Positive	Negative	Positive	Negative	Positive
Expected Value	Negative	98.64 %	1.36 %	97.76 %	2.24 %	99.55 %	0.45 %
	Positive	42.36 %	57.64 %	44.98 %	55.02 %	55.69 %	44.31 %

Table 5: Confusion matrices for the proposed method under illumination changes.

		RGB Classification		HSV Classification		L*a*b* Classification	
		Negative	Positive	Negative	Positive	Negative	Positive
Expected Value	Negative	94.95 %	5.05 %	99.47 %	0.53 %	87.20 %	12.80 %
	Positive	73.59 %	26.41 %	78.17 %	21.83 %	77.96 %	22.04 %

Table 6: Confusion matrices for the Gaussian Mixture Model under illumination changes.

6. Conclusions

In this paper, an object detection system robust to sudden illumination changes was proposed. Several tests were made, the first one considering the identification of the active background model for each incoming frame. This test concluded that the L*a*b color space is the best choice considering this situation. The second test had the objective of evaluating the foreground segmentation algorithm of the proposed system, for both static background and sudden illumination changes. In the first case, the foreground segmentation produced good results, except for the HSV color space, which produced a higher false positive rate. Once again, the best results were achieved using the L*a*b* color space. Under illumination changes, one can consider that the foreground segmentation is reasonable, despite the higher false negative rates obtained in all color spaces. Finally, the proposed system was compared with a gaussian mixture model [5]. Regarding the obtained results, it's possible to conclude that the proposed method is more efficient than the classical model, particularly in the case of sudden light changes.

As a general view, it's possible to say that the proposed system has a more efficient way of dealing with the background model management, which results in a good foreground segmentation performance, even in the presence of sudden light changes.

Acknowledgements

Firstly, I would like to thank my advisor, Professor Jorge Salvador Marques, for having accepted my thesis proposal and, above all, for the support provided throughout the time. My co-supervisor, Professor João Miranda Lemos, also for his support in this work and also for the opportunity he gave me to participate in ARGUS Project, which made me learn a lot. I would also like to thank the Fundação para a Ciência e Tecnologia for the research grant I was assigned to for the ARGUS project PTDC/EEA-CRO/098550/2008. I also have to thank Dr. Miguel Lobo Antunes, to Dr. Paulo Prata Ramos and to the whole Culturgest team for the help they gave me in the recording of the test videos used, giving a great contribution to this work.

References

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

[2] A. Elgammal, D. Harwood, and L. Davis. Nonparametric model for background subtraction. 1843:751–767, 2000.

[3] L. Li and M. Leung. Integrating intensity and texture differences for robust change detection. *Image Processing, IEEE Transactions on*, 11(2):105 –112, feb 2002.

[4] J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. pages 336–350, 2000.

[5] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. 2:637–663, 1999.

[6] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. M. Buhmann. Topology free hidden markov models: Application to background modeling. In *ICCV'2001*, pages 294–301, 2001.

[7] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 51 –56, oct 1996.