

Motion Segmentation of Color Images

Adaptive background modelling by using a mixture of Gaussians, improved by collinearity and maximum criterion decision

Joan Borràs Nogués and Tomás Svoboda

Czech Technical University in Prague

joan.borras@estudiants.urv.cat

11th April 2008

Segmentation of color images is a common research area, where good algorithms have been developed. In this project the algorithm will process each input frame to decide which pixels belong to the background or to an object detection. After an input image is read, it uses the adaptive background modelling by using a mixture of Gaussians, GMM [6]. It creates a model with several Gaussians. After that it decides which Gaussians belong to the background. There can be more than one Gaussians that belongs to the background, but only one has to be selected. That one will be the most similar to the intensity values of the pixel. Once it has the intensity values of the most similar background and the current data, it compares that values and its neighbourhood to decide if the point is labelled as foreground or background. The comparison can be made by collinearity [1] or maximum distance criterion. If that distance is smaller than a threshold, then the pixel will be labelled as a background, otherwise it will be labelled as foreground.

The threshold has to be specified. So a significance test is running to a percentage of the pixels in each frame. It uses the GMM algorithm to know which pixels match background. With those pixels it will process an histogram to specify the threshold. The GMM can be run with gray or color intensity values, and also the distance criterion can be selected to use gray or color. So it means that different results can be experimented. Two sequences of images were tested, and the results of the segmentation improved. However the processing times are increased with respect to the source algorithms.

1. Introduction

The objective of this project is to find new methods for the motion segmentation research area. The segmentation is used to find movement in some scene. The scene would be a static camera observing a room, street, or similar situation. This observation is considered to be the background. Other unimportant movements such as image noise, leaves or sunlight should also be considered as a background. Then the goal would be the detection of a moving object within the scene, for example a person or car. These are extracted to become the foreground of the scene, and it is important to avoid false detection. Many video surveillance systems employ a static or quasi-static camera, which uses these kind of algorithms for motion detection. Other examples of practical applications include robots, vehicle navigation and video indexing.

Some aspects to improve are the robustness and the processing time. A fast algorithm is required to process frame-by-frame especially for Real-time systems. At the same time it is essential for the surveillance systems to have a very low false alarm rate.

2. Algorithm

GMM will be used to create the background image from the input image. It will be used also to create a mask of the image, that indicates which pixels matches foreground and background. The points selected in that mask are those that will do the significance test. Once the background image is known, the algorithm will create a matrix containing the intensity values of the neighbourhood for each pixel. That matrix is created both for the input image and for the background image. So, each point of the matrix has a vector with its neighbourhood. Then the vector of the input image has to be compared with the vector of the background image. The method of the comparison can be either collinearity distance or maximum distance.

The first method will test the collinearity of that two observed vectors and compare this to a threshold. If that distance is less than the threshold the pixel will be considered foreground. The maximum method takes the two vectors, and applies the absolute distance between them. Thus a third vector with the result is created. The maximum value of the vector is compared to a threshold to decide if the pixel is background or foreground. The threshold used for the comparison is specified on the training mode. It will be run on a percentage of pixels in each frame. On training mode it will create an histogram with the results of the distance criterion (collinearity or maximum) for some pixels. The matrix of matches created in GMM indicates those pixels which will be used for the significance test. The histogram accumulates the values of all the the frames and pixels used in the training mode, and specifies the threshold.

2.1. Adaptive background modelling by using a mixture of Gaussians GGM

The method GGM [2] processes pixel by pixel and creates a model for each one, which will be updated with each new frame. The model would be represented by the following matrix, every row will be a new Gaussian, where K is the number of rows (works well with $K=5$):

$$\begin{bmatrix} \boldsymbol{\mu}_1^T & \omega_1 & \sigma_1 & b_1 \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\mu}_K^T & \omega_K & \sigma_K & b_K \end{bmatrix},$$

$\boldsymbol{\mu}$ is the estimation of the observation, could be a vector of 3 dimensions for a color mode, or a scalar for a gray mode (this report will refer in color mode); ω is the weight of each Gaussian; σ the covariance; and b are the flags whether the i^{th} Gaussian belongs to the background $b=1$ or foreground $b=0$. The next step is to initialize a new Gaussian if no match is found, find the Gaussian with minimal weight and replace it with a new one. Or if a match is found, then decrease the weights of non-matching Gaussians and re-normalize. Then update the model. After the update of the model the next step is to decide which Gaussians represent the background. It is important to note that the background may change due to lighting changes, object movement, etc. The set of background Gaussians has to change accordingly, it will be labelled in the value b of the model.

Afterwards the algorithm GMM compares the actual observation x with all Gaussians and selects the one that matches, which uses the covariance values to recognize it. Then it creates a mask with all pixels. The pixel will be labelled as a match foreground if there was no match at all or the matched Gaussian(s) does not belong to the background ($b=1$). This mask of matches are exclusively used to compute the significance test.

2.2. Background image creation

After update of the complete model it is necessary to create the background image. It selects the Gaussians that belongs to the background, the same as $b=1$. There can be more than one, but only one has to be selected. It will select the one which the intensity values $\boldsymbol{\mu}^T$ are the most similar to the current data \mathbf{x}^T . So, for a Gaussian $_i$ that belongs to the background it computes the absolute distance between the vectors $\boldsymbol{\mu}_i^T$ and \mathbf{x}^T and calculate the average value over color channels. For all the possible results it selects the minimal $D_bckg_i(1)$. It means that it selects the Gaussian that is most similar to the current data \mathbf{x}^T .

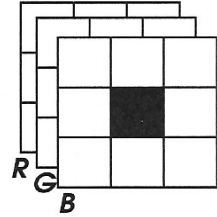
$$i^* = \underset{i}{\operatorname{argmin}} \quad \mathbf{D_bckg}_i = \sum_c \operatorname{abs}(\boldsymbol{\mu}_{ic}^T - \mathbf{x}_c^T) \quad (1)$$

where c are the number of color components and i are the indexes of the Gaussians that belongs to the background. For all the values $\mathbf{D_bckg}_i$ it computes the average between the color components and then the index of the minimum value will be chosen. This index i^* indicates the Gaussian to use as a background. Now it can create the background image with the intensity values of the Gaussian selected. This background image will be used in the distance criterion, see 2.3.

2.3. Distance criterion decision: Collinearity and maximum

The technique used to calculate the distance creates a neighbourhood for every pixel both for the intensity values of the input image and the background image created in

GMM algorithm. The neighbourhood is a 3x3 window. If it is using gray scale intensity values, the neighbourhood becomes a 9 dimension vector, or 27 dimension vector if it uses the color intensity values. The figure on the right illustrates the generalization for different color channels, here red, green and blue, but it could be only one channel with the gray values.



To create the neighbourhood in that images for each pixel p_{ij} it creates a vector \mathbf{c}_{ij} for the current input image and \mathbf{b}_{ij} for the background image with the intensity values of its neighbourhood. Where the length of that vectors can be 9 or 27. The creation of the vector of one pixel located in the i^{th} row and the j^{th} column will be like this:

$$\mathbf{c}_{ij} = [\mathbf{x}_{(i-1)(j-1)}^T \quad \mathbf{x}_{(i-1)j}^T \quad \mathbf{x}_{(i-1)(j+1)}^T \quad \mathbf{x}_{i(j-1)}^T \quad \mathbf{x}_{ij}^T \quad \mathbf{x}_{i(j+1)}^T \quad \mathbf{x}_{(i+1)(j-1)}^T \quad \mathbf{x}_{(i+1)j}^T \quad \mathbf{x}_{(i+1)(j+1)}^T], \quad (2)$$

where \mathbf{x}^T are the the intensity values of the pixel selected. The creation of the \mathbf{b}_{ij} will be the same as \mathbf{c}_{ij} with the intensity values of the background image. Once both neighbourhood vectors are created, they have to be compared according to the distance criterion. For each pixel p_{ij} the comparison will be made between the neighbourhood vector of the input image \mathbf{c}_{ij} and the background image \mathbf{b}_{ij} . The comparison of those two vectors is used to decide if the pixel is finally labelled as a background or foreground. There are two different methods to run the comparison: collinearity or maximum criterion.

a) Collinearity criterion:

The procedure of the collinearity criterion [1] for every pixel p_{ij} , this vector \mathbf{c}_{ij} in the current frame is compared to the corresponding vector \mathbf{b}_{ij} of the background and checked for collinearity. Figure 1 shows this principle for 2 dimensions.

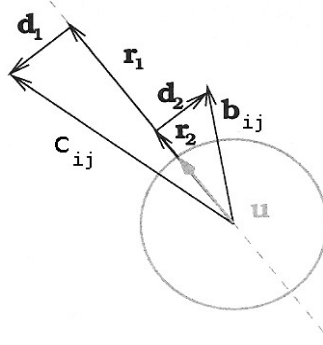


Figure1. Geometrical interpretation of testing the collinearity of two vectors, in 2 dimensions.

The collinearity distance is given by minimizing the sum $D^2 = |\mathbf{d}_1|^2 + |\mathbf{d}_2|^2$ of the square distances of the observed vectors \mathbf{c}_{ij} and \mathbf{b}_{ij} to the axis given by vector \mathbf{u} . The projections \mathbf{r}_1 and \mathbf{r}_2 are estimates of the corresponding signal vectors. Where it is something like:

$$|\mathbf{d}_1|^2 = |\mathbf{c}_{ij} - \mathbf{r}_1|^2, \quad (3)$$

$$|\mathbf{d}_2|^2 = |\mathbf{b}_{ij} - \mathbf{r}_2|^2, \quad (4)$$

$$|\mathbf{r}_1| = |\mathbf{c}_{ij}| \cdot \cos\phi_1 = |\mathbf{c}_{ij}^T \cdot \mathbf{u}| \quad (|\mathbf{u}| = 1!), \quad (5)$$

$$|\mathbf{r}_2| = |\mathbf{b}_{ij}| \cdot \cos\phi_2 = |\mathbf{b}_{ij}^T \cdot \mathbf{u}|, \quad (6)$$

with the above equations it derives to:

$$|\mathbf{d}_1|^2 = |\mathbf{c}_{ij} - (\mathbf{c}_{ij}^T \cdot \mathbf{u})|^2, \quad (7)$$

$$|\mathbf{d}_2|^2 = |\mathbf{b}_{ij} - (\mathbf{b}_{ij}^T \cdot \mathbf{u})|^2, \quad (8)$$

$$D^2 = |\mathbf{d}_1|^2 + |\mathbf{d}_2|^2 = |\mathbf{c}_{ij}|^2 + |\mathbf{b}_{ij}|^2 - |\mathbf{c}_{ij}^T \cdot \mathbf{u}|^2 - |\mathbf{b}_{ij}^T \cdot \mathbf{u}|^2. \quad (9)$$

Now it can form the matrix \mathbf{X} , containing two rows. One row is the vector of the neighbourhood matrix of the input image \mathbf{c}_{ij} , and the other is the vector of the background image \mathbf{b}_{ij} . It would be like (10):

$$\mathbf{X} = \begin{bmatrix} \mathbf{c}_{ij}^T \\ \mathbf{b}_{ij}^T \end{bmatrix} \quad \mathbf{X} \cdot \mathbf{u} = \begin{bmatrix} \mathbf{c}_{ij}^T \cdot \mathbf{u} \\ \mathbf{b}_{ij}^T \cdot \mathbf{u} \end{bmatrix}, \quad (10)$$

then it can derive

$$|\mathbf{X} \cdot \mathbf{u}|^2 = \mathbf{u}^T \cdot \mathbf{X}^T \cdot \mathbf{X} \cdot \mathbf{u} = |\mathbf{c}_{ij}^T \cdot \mathbf{u}|^2 + |\mathbf{b}_{ij}^T \cdot \mathbf{u}|^2, \quad (11)$$

it turns out that

$$D^2 = |\mathbf{d}_1|^2 + |\mathbf{d}_2|^2 = |\mathbf{c}_{ij}|^2 + |\mathbf{b}_{ij}|^2 - \mathbf{u}^T \cdot \mathbf{X}^T \cdot \mathbf{X} \cdot \mathbf{u}, \quad (12)$$

and the vector \mathbf{u} that minimizes D^2 is the same vector that maximizes.

$$\mathbf{u}^T \cdot \mathbf{X}^T \cdot \mathbf{X} \cdot \mathbf{u} \rightarrow \max \quad \text{with } |\mathbf{u}| = 1, \quad (13)$$

which is an eigenvalue problem with respect to matrix $\mathbf{X}^T \cdot \mathbf{X}$. The non-zero eigenvalues of $\mathbf{X}^T \cdot \mathbf{X}$ and $\mathbf{X} \cdot \mathbf{X}^T$ are identical. Thus, the sought eigenvalue is identical to the smaller one of the two eigenvalues of the 2 x 2 matrix $\mathbf{X} \cdot \mathbf{X}^T$, which can be computed in closed form without using iterative numerical techniques. So you can use the eigenvalues of $\mathbf{X} \cdot \mathbf{X}^T$ as a distance measure. The product is:

$$\mathbf{X} \cdot \mathbf{X}^T = \begin{bmatrix} \text{fore} & \text{cross} \\ \text{cross} & \text{back} \end{bmatrix}, \quad (14)$$

and the final collinearity distance between the two vectors would be [1,4]:

$$D_col^2 = \frac{1}{2} \left(\text{fore} + \text{back} - \sqrt{(\text{fore} - \text{back})^2 + 4 \cdot \text{cross}^2} \right). \quad (15)$$

Let D_col^2 be the threshold of distance t , to avoid another square root, which has a long processing time. The last equation will be used for obtain the D_col^2 threshold on training mode. Once the threshold is known, with some expansions of the last equation to avoid the square root, it becomes:

$$\begin{aligned} (\text{fore} - t) \cdot (\text{back} - t) &> \text{cross}^2 \\ \text{fore} + \text{back} &> 2 \cdot t \end{aligned} \quad (16)$$

Then finally, when t (the threshold in D_col^2) is known, it just needs to check both inequalities (16) to set the pixel as foreground if they are positive, or otherwise as background. This is computationally cheaper than calculating the square root (15). It means that on the training mode the algorithm will use the equation number (15) because it is studying the threshold with the significance test. When it is not on training mode it can use (16) which is computationally faster. Would be better if (16) could be used for the training mode, but there is no way to get the D_col^2 , so is precisely necessary to use (15) for the significance test.

b) Maximum criterion:

The technique consists in take the two vectors \mathbf{c}_{ij} and \mathbf{b}_{ij} to compare, and apply the absolute distance between them (17).

$$D_max = \max (\text{abs} (\mathbf{c}_{ij} - \mathbf{b}_{ij})), \quad (17)$$

$$D_max < t, \quad (18)$$

the result of the difference is a vector, then take the maximum value of the vector (17) and if it is bigger than the threshold it will be considered foreground (18).

2.3 Training mode

The training mode consists in making a significance test of some pixels for each frame. The number of pixels to test is a percentage of the total pixels of one frame. The value N_{train} indicates the percentage of the pixels that will be used. For the pixels selected to test will be ruled out those which its neighbourhood have at least one pixel that matches foreground in GMM, see 2.1. The significance test will be done for each pixel accepted. This idea is the same for both collinearity and maximum criterion. So, for each pixel selected the algorithm checks the value of the distances, with equation (15) or equation (17) for collinearity or maximum mode respectively. It means that the value D_{col^2} and D_{max} will construct the histogram, creating the frequencies of each value to compose it. The histogram obtained in one frame will be kept for the next frame. In other words, the frequency of that values in one frame will accumulate the values of all the previous frames. Finally, the histogram will be constructed for all the frames of the sequence, and for only those pixels selected in each frame.

After the histogram is created it performs a significance test with level $alpha_dist$. It consists in normalising the histogram with the accumulation sum. To assign the threshold it takes the value when $alpha_dist \cdot 100$ % of background pixels will be misclassified as foreground (false positives):

$$p\langle D_{col^2} > T | B \rangle = alpha_dist . \quad (19)$$

All this process will be done in each new frame. Accumulating the histogram of the previous frames. So it will study a new threshold for each frame. It means that the threshold will be adapted as the sequence progresses. And the threshold obtained at that moment will be the one used for the segmentation in the current frame.

3. User's guide

The implementation strongly builds on [5] implementation of GMM. Notation and system of configuration are kept intentionally similar.

3.1. Running the `segm_run.m`

The `segm_run.m` is a function prepared to run the final algorithm, the configure values are already set with the most probably optimal result. The call to a function would be like this:

```
param.inputDir = 'images/Office/';
param.outputDir = '/output_images/Office/DcolD3G3/';
param.imagePatron = 'img01_%04d';
param.format = 'jpg';
param.startFrame = 1;
param.endFrame = 1070;
param.D2_method = 'col';
param.D2_dim = 3;
param.GMM_dim = 3;
param.nosave = 0;

segm_run (param)
```

The input parameters are:

<code>inputDir</code>	String	Path of the frames location directory.
<code>outputDir</code>	String	Path of the results location directory.
<code>imagePatron</code>	String	Base name of the frames.
<code>format</code>	String	Format of the images.
<code>startFrame</code>	1x1	The index of the first frame.
<code>endFrame</code>	1x1	The index of the last frame.
<code>D2_method</code>	String	Method for the distance: - 'col' : Collinearity criterion. - 'max': Maximum criterion.
<code>D2_dim</code>	1x1	Distance dimension: - 1: Gray version, 9 dimensional neighbourhood. - 3: Color version, 27 dimensional neighbourhood.
<code>GMM_dim</code>	1x1	Model Gaussian dimension: - 1: Gray version, 'u' (mean) of pixel model is scalar. - 3: Colour version, 'u' of pixel model is a vector of 3D.
<code>nosave</code>	1x1	Save the files on the disk or not.

The `segm_run` function essentially calls the `segm_frame` function. In case the application requires some modifications of the parameters it can be done by calling the `segm_frame` function.

3.2. Running the segm_frame.m

If you want to change some parameters, there is the possibility to do it by preparing the values and then running directly the function segm_frame.m. The call would be:

```
D2_method = 'col';
D2_dim = 3;
GMM_dim = 3;

cfg.input.dir = 'images/Office/';
cfg.output.dir = '/output_images/Office/DcolD3G3/';
cfg.input.basename = 'img01_%04d';
cfg.input.fmt = 'jpg';
cfg.output.fmt = 'png';
cfg.output.dirs = {'01Input/', '02Segm/', '03ImageSegm/'};

% Set the configuration data for GMM
cfg.alpha = 0.05; %
cfg.K = 5; % Number of Gaussians
cfg.var = 30; % initial variance
cfg.idx = [1:1:1070];
cfg.thr = 2.5;
cfg.T = 0.6;
cfg.D = 0;
cfg.Ntrain = 10;
cfg.alpha_dist = 0.01;
cfg.D2_method = 'col';
cfg.D2_dim = 3;
cfg.GMM_dim = 3;

model = [];
idx = [];

for i=cfg.idx
    im = imread([cfg.input.dir, sprintf(cfg.input.basename,i), ...
        ' ',cfg.input.fmt]);

    [model,idx,imSegm,cfg,D_hist] = segm_frame(im,model,idx,cfg,...
        D2_method,D2_dim,GMM_dim,D_hist)
end
```

The input parameters are:

im4seg	[r x c]	Input image. This is one frame from a video sequence.
model	[r x c x K x n]	Multidimensional array containing all statistics about pixels. If the model is empty it is initialized, see model_init for details.
idx	struct	Structure containing indexes (pointers) to the model. Currently, four index groups are used - idx.mu indexes means, idx.vars variances, idx.w weights, and idx.bckg background flags.
cfg	struct	Structure containing indexes (pointers) to the model .K: 1x1 Number of Gaussians in the model. .var: 1x1 Initial variance. .thr: 1x1 Multiplicative coefficient that multiplies standard deviations for thresholding differences. .alpha: 1x1 Learning coefficient. .T: 1x1 Threshold decision about background/foreground. .D: 1x1 Threshold decision about distance between current vector and closest background. .Ntrain: 1x1 Percentage of used pixels for the training. .alpha_dist: 1x1 Level of significance test
D_hist	[1 x 10000]	The distance histogram of the first 'maxIterD' frames. The first call to the function has to be a vector of zeros. Then it becomes the sum of data from all previous frames.

The output parameters are:

model	[r x c x K x n]	Updated model of the background.
idx	struct	Structure containing indexes (pointers) to model. The indexes are not changed throughout the sequence: they are included among output parameters for the sake of consistency since bckggm also initializes the model.
imSegm	[r x c]	Is the segmented image.
cfg	struct	The same as the input, it needs to keep some modified config data for the next frames.
D_hist	[1 x 10000]	The same as the input, it needs to accumulate the histogram for the first cfg.maxIterD frames.

4. Results

4.1. Experiment 1: “Office”

a) Segmentation results

The next figures show some experimental results obtained by using the techniques described above. The images display the segmentation of a frame sequence. The results showed are the best method of the collinearity and maximum, which is the one that uses the color of intensity values.

In the the first row there is the input image and the GMM segmentation, and in the second row we have the collinearity and the maximum segmentation.

The following results are configured with $\alpha = 0.005$ for the significance test, the number of Gaussians K is 5 and the 10% of pixels will be used for the significance test.

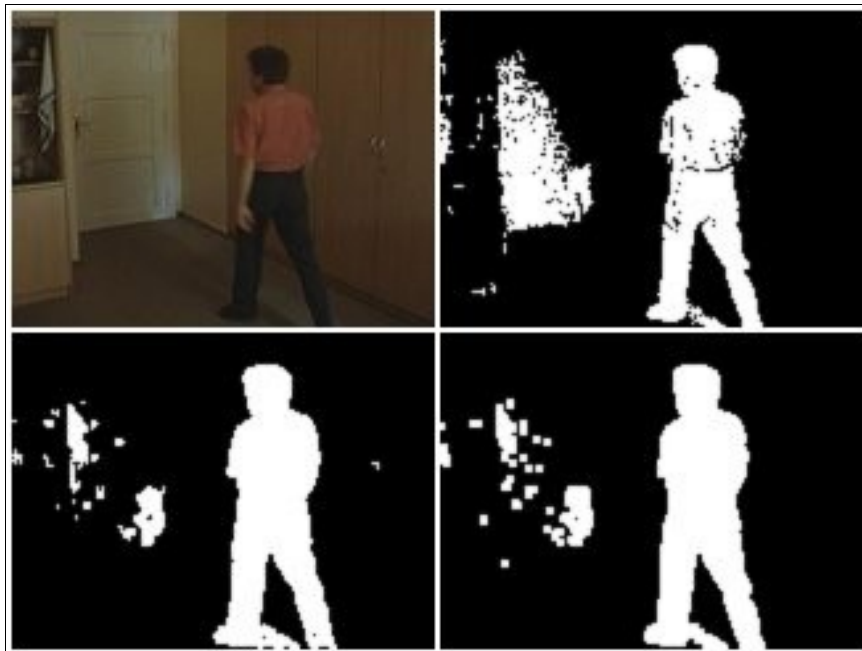


Figure 2. Frame number 75. A person comes into an empty room and causes some shadows, which create false positives. The effect is especially noticeable over light colors like the white wall, due to the high contrast.

With the collinearity model, the false positives and negatives are reduced with respect to the GMM. The maximum method gives a clear segmentation of the body because it uses the maximum distance. But it provokes a higher index of false positives.



Figure 3. Frame number 145. Now the shadows provoked for the body are in the wardrobe.

That is why the false positives are increased in the collinearity method. When the critical zones are in an heterogeneous area, the distance of the collinearity method increases. That provokes false positives. There are also some false negatives, but less than the GMM method.

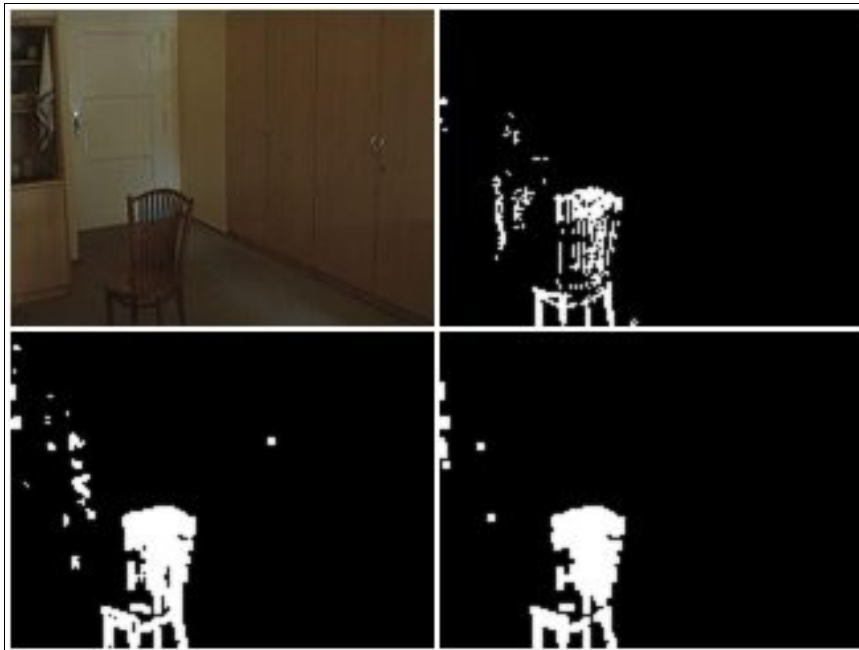


Figure 4. Frame number 335. A chair is put in the room. The segmentation is quite clear for all the methods. GMM is more precisely because does not look the neighbourhood.

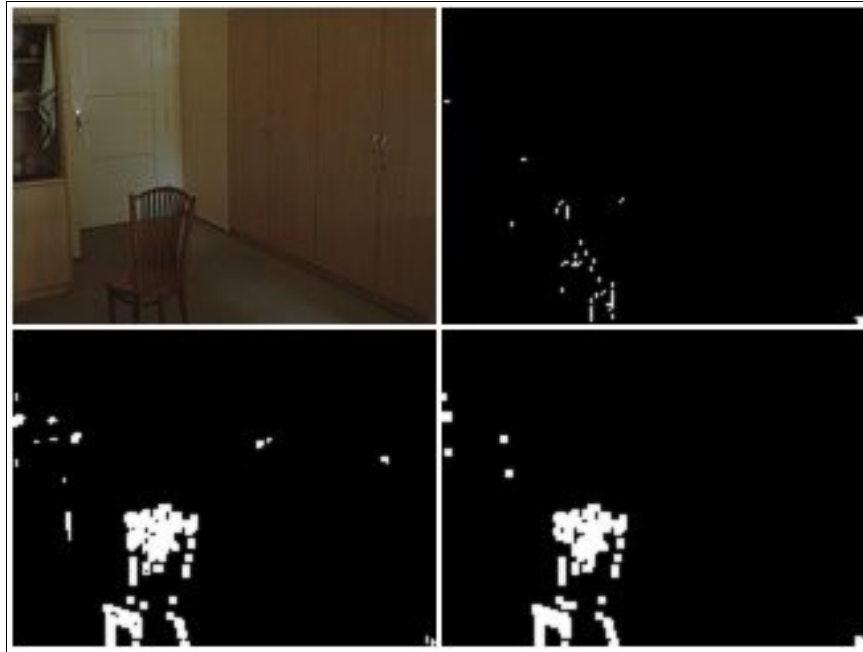


Figure 5. Frame number 456. After some seconds the chair becomes background. The collinearity and maximum method have a delay with respect to GMM. That happens because the heterogeneity (the holes) of the chair. Looking the part of the chair where there are not holes, it becomes background properly.

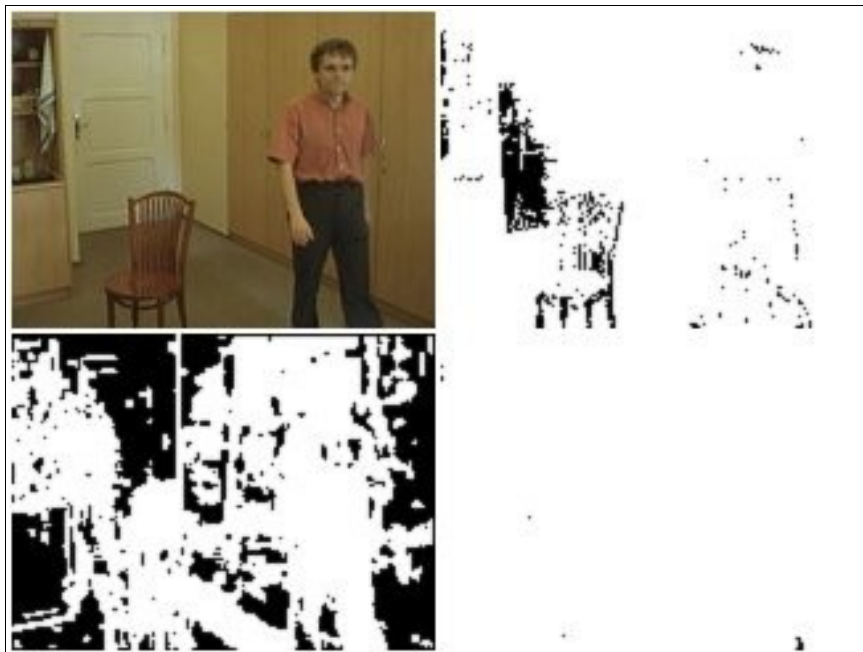


Figure 6. Frame number 598. The lights are turned on. Using the maximum method, almost all the pixels become foreground. Also has the same reaction for the collinearity method. The collinearity method only detects the heterogeneous zones as foreground. The corners, body and chair are the only pixels to be segmented. The smooth colors, like the wardrobe, are misclassified background. That happens as explained previously.

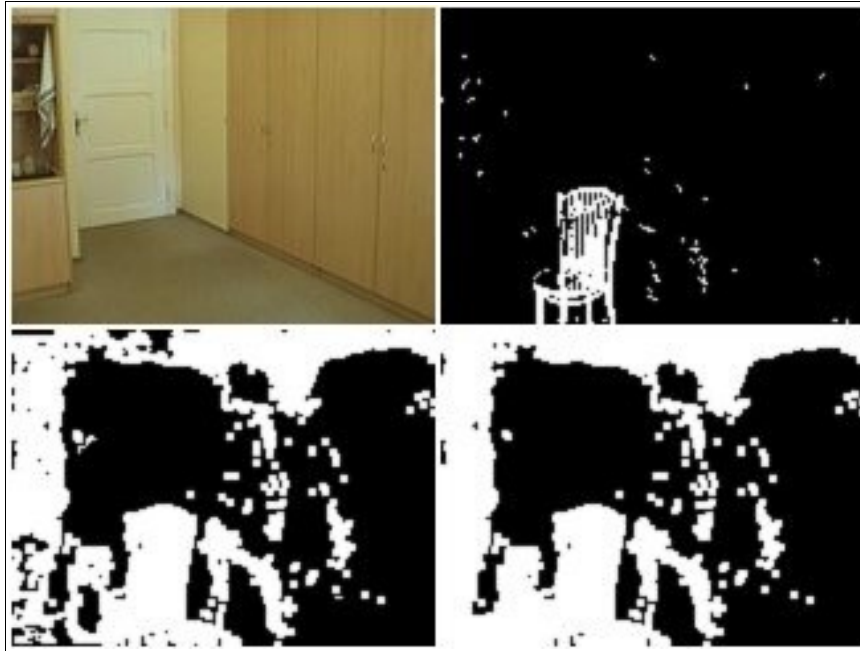


Figure 7. Frame number 1070. After a while, the room with the light on, adapts to the background. The algorithms are expecting the chair as a foreground, even though it is not there. It will take more time to adapt to the background without the chair. The collinearity and maximum method still have false positives specially in the heterogeneous areas. After the changing of the illumination the adaptation of the methods does not run properly as GMM.

b) Significance test

During the experiments tested, the threshold of the distance both collinearity and maximum method is updating in each frame. An area of the image is used to compute the significance test. In figure 8 and 10 we can see the histogram accumulated at the end of the sequence. In figures 9 and 11 there is the evolution of the threshold during the sequence.

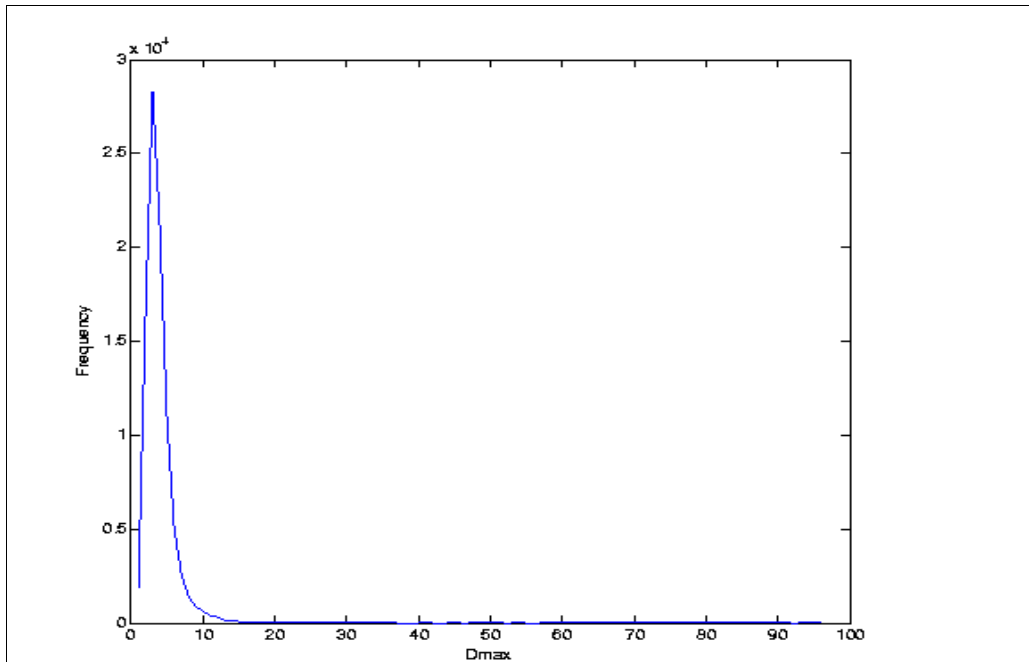


Figure 8. Histogram of the of the collinearity distance value for all the frames. During all the frames, those pixels selected for the training mode are accumulating its values to create an histogram.

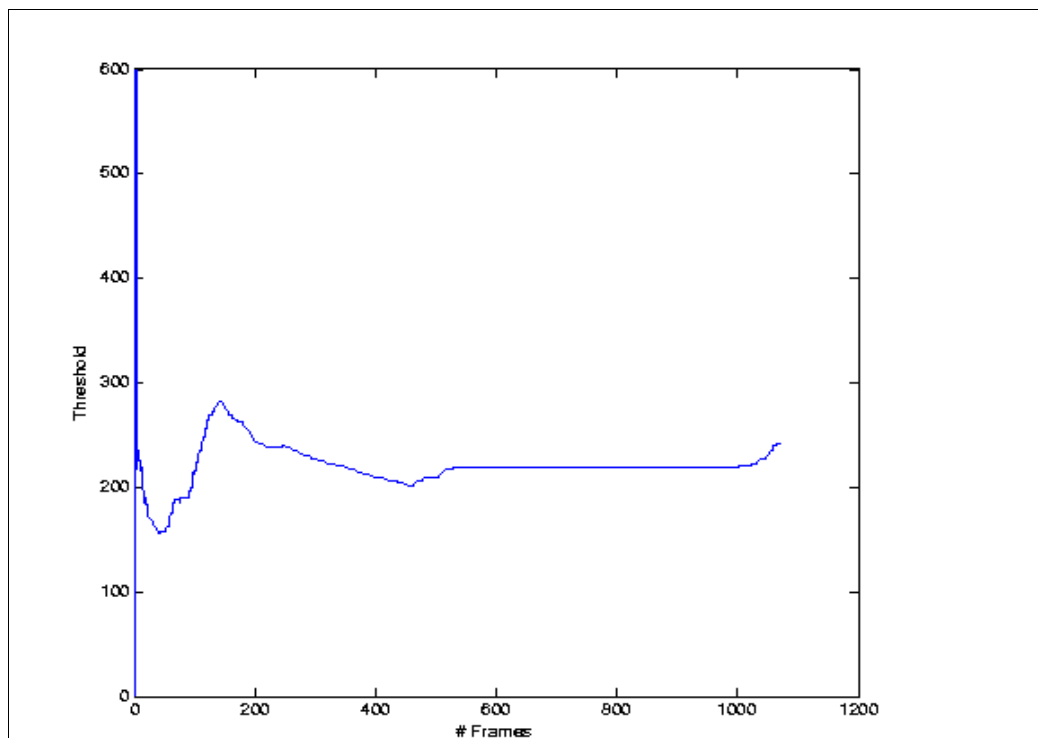


Figure 9. The evolution of the collinearity threshold during the sequence of the frames. The threshold is updated in each frame using the normalised histogram for each one. After the frame 545 the threshold does not change due the lights are on. It means that almost all the pixels are match foreground in GMM, then just few pixels are used for the significance test.

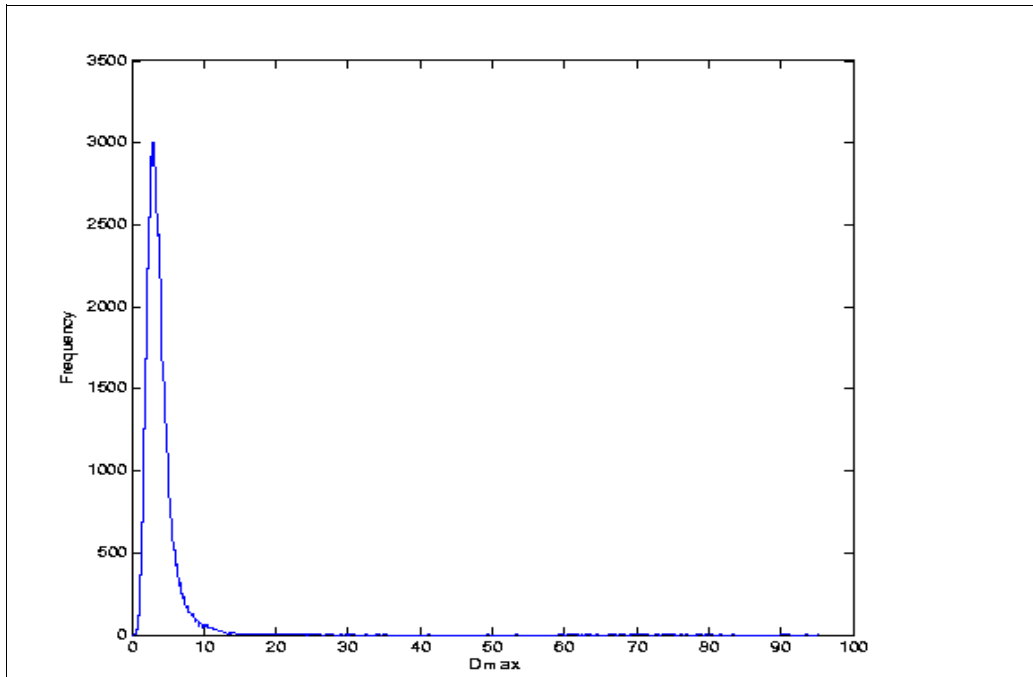


Figure 10. Histogram of the of the maximum distance value for all the frames. During all the frames, those pixels selected for the training mode are accumulating its values to create an histogram.

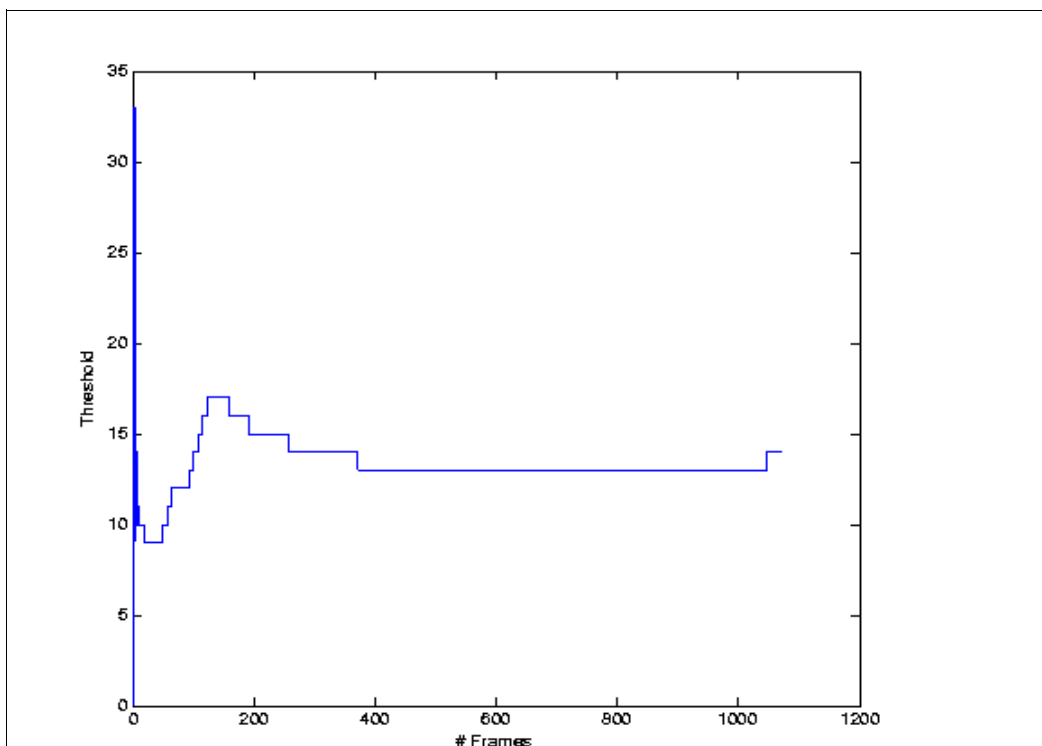


Figure 11. The evolution of the maximum threshold during the sequence of the frames. The threshold is updated in each frame using the normalised histogram for each one.

As the figure 9 and 11 show, the threshold is quite stable. It does not change with high values. So we can consider that the pixels selected for the significance test does not have big changes, it means that just few errors exist.

d) Processing times

Figure 12 shows the comparison of the processing times between the different methods. As we can see the first algorithm GMM is the fastest one. After adding the collinearity or maximum method the processing time increases, as expected. Looking the segmentation results, we could decide that the best improvement of the GMM is the collinearity method with the color version both Gaussian and distance, but is also the slowest one. The time is increased up to 52% of the fastest one.

We can notice also that the difference of the time between use a color or gray version in the distance decision is insignificant. A larger increase occurs when using the Gaussian method for color as opposed to gray images, but when using Gaussian for gray images the result is not satisfactory.

Table 1: Time in seconds that takes each version to compute the previous experiment.

	Method	G.M.M. Dim.	Distance dim.	Sec./frame	Total seconds
1	G.M.M	1	-	2.9	3109
2	G.M.M	3	-	3.08	3297
3	Maximum	1	1	3.18	3402
4	Maximum	1	3	3.18	3403
5	Maximum	3	1	3.72	3985
6	Maximum	3	3	3.73	3998
7	Collinearity	1	1	3.89	4163
8	Collinearity	1	3	3.97	4249
9	Collinearity	3	1	4.39	4712
10	Collinearity	3	3	4.41	4718

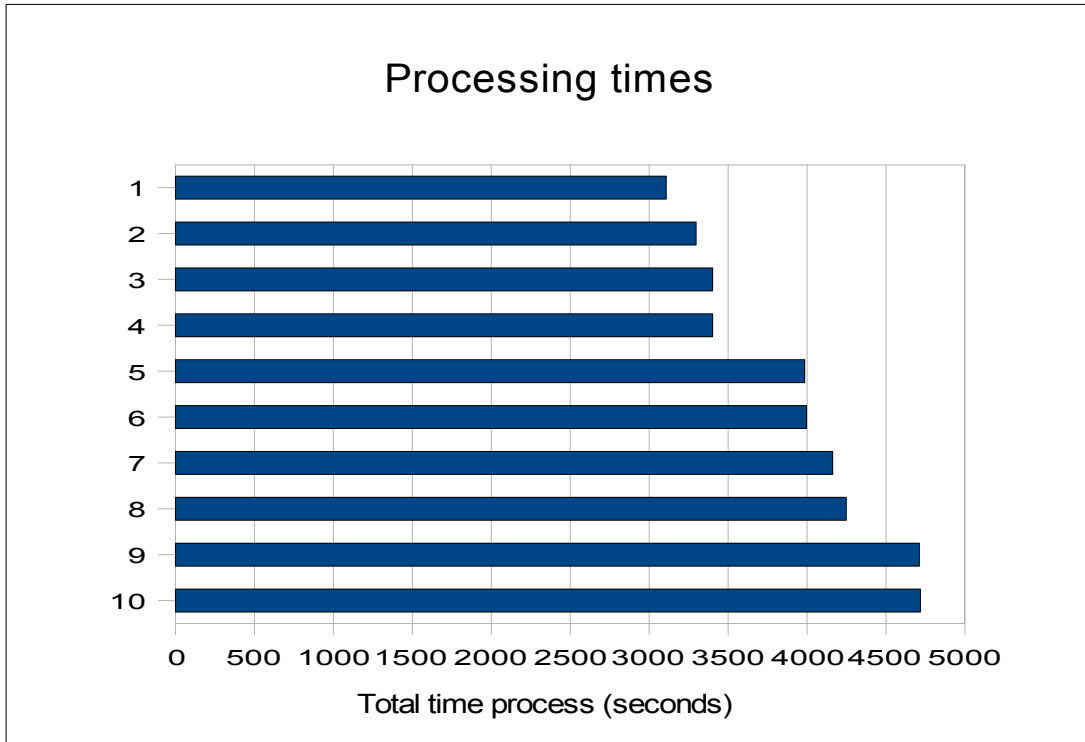


Figure 12. Bar graph of processing times.

4.2. Experiment 2: “Red Chair”

The following results are configured with $\alpha = 0.005$ for the significance test, the number of Gaussians K is 5 and the 10% of pixels will be used for the significance test.



Figure 13. Frame number 29. In that empty room a person is moving. The body is clearly better segmented with the newer versions than the GMM. There are also fewer false positives.

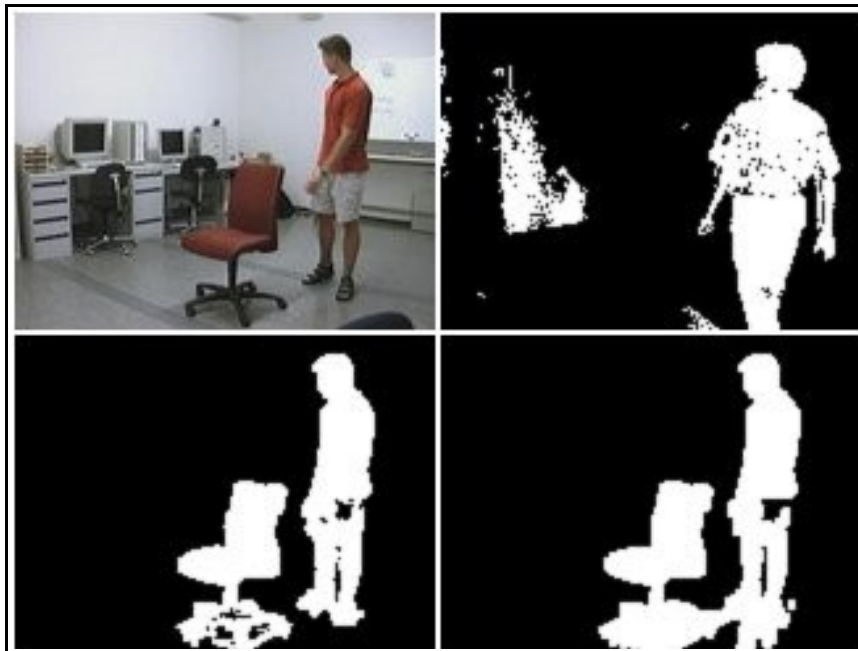


Figure 14. Frame number 62. The person puts a chair in the room.

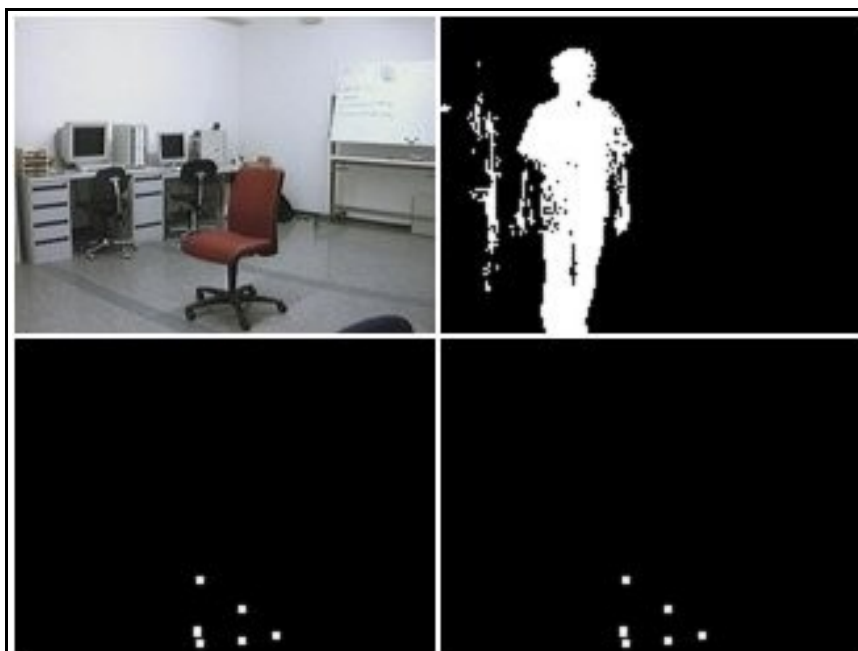


Figure 15. Frame number 113. Adaptation to the new background after some seconds.

Figure 14 shows how a new object appears. For the first seconds the chair will have to be considered as foreground. But after a while, as in figure 15, the background would be adapted including the chair. The adaptation of the new versions is faster than the GMM one.



Figure 16. Frame number 203. The lights are turned on, and almost all the pixels are considered foreground, except the smooth areas.

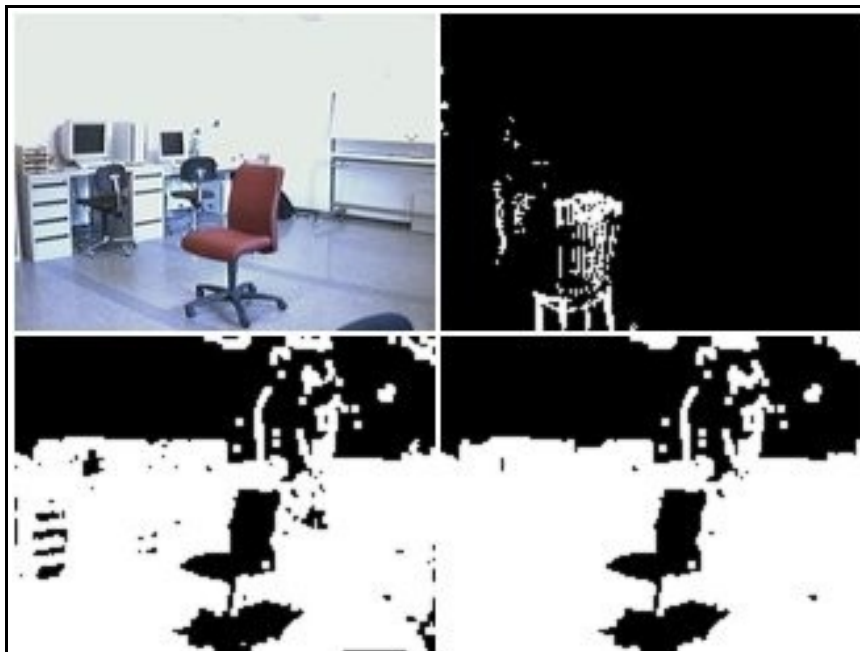


Figure 17. Frame number 338. After some seconds, some versions do not adapt as fast as GMM. Because of the heterogeneity of some zones. But the smooth areas like the wall and the chair become background properly.

b) Significance test.

The same as the first experiment, the values of the threshold are studied.

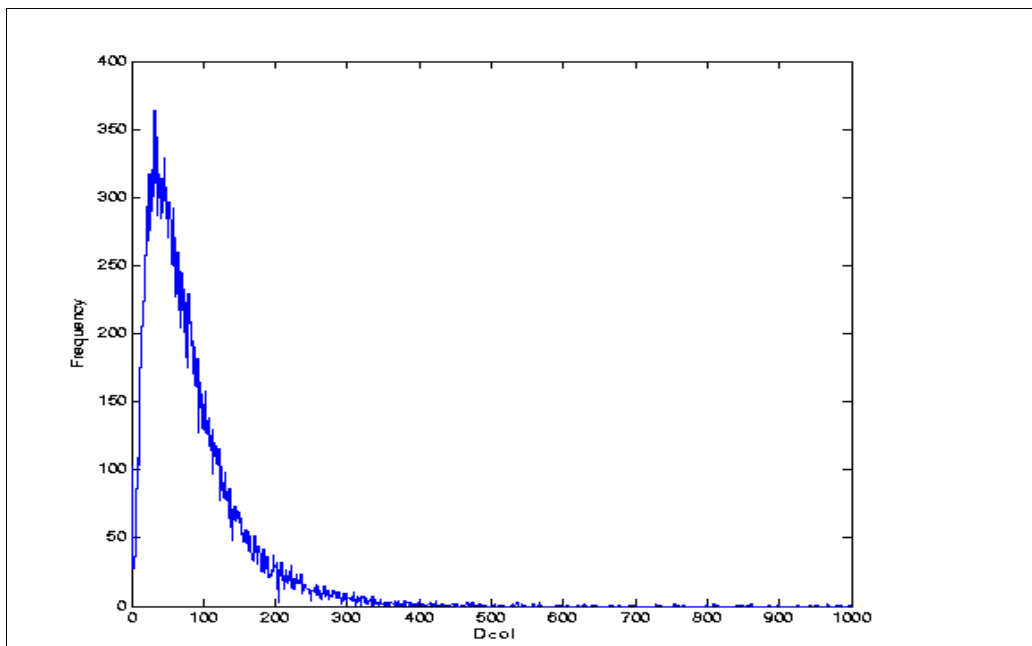


Figure 18. Histogram of the of the collinearity distance value for all the frames.

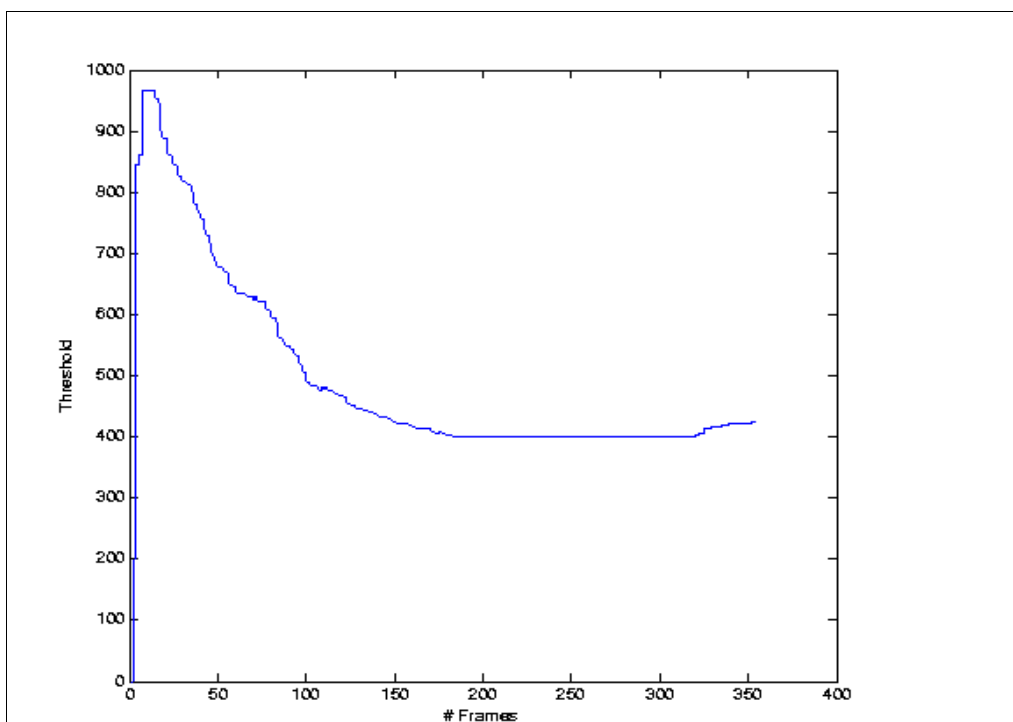


Figure 19. The evolution of the collinearity threshold during the sequence of the frames.

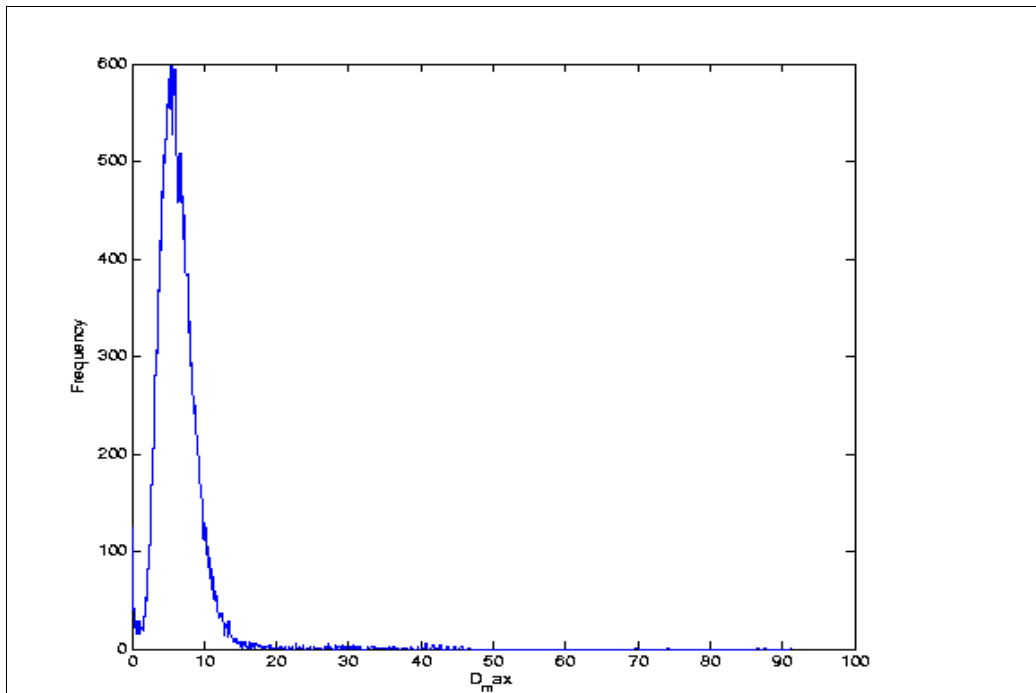


Figure 20. Histogram of the of the maximum distance value for all the frames.

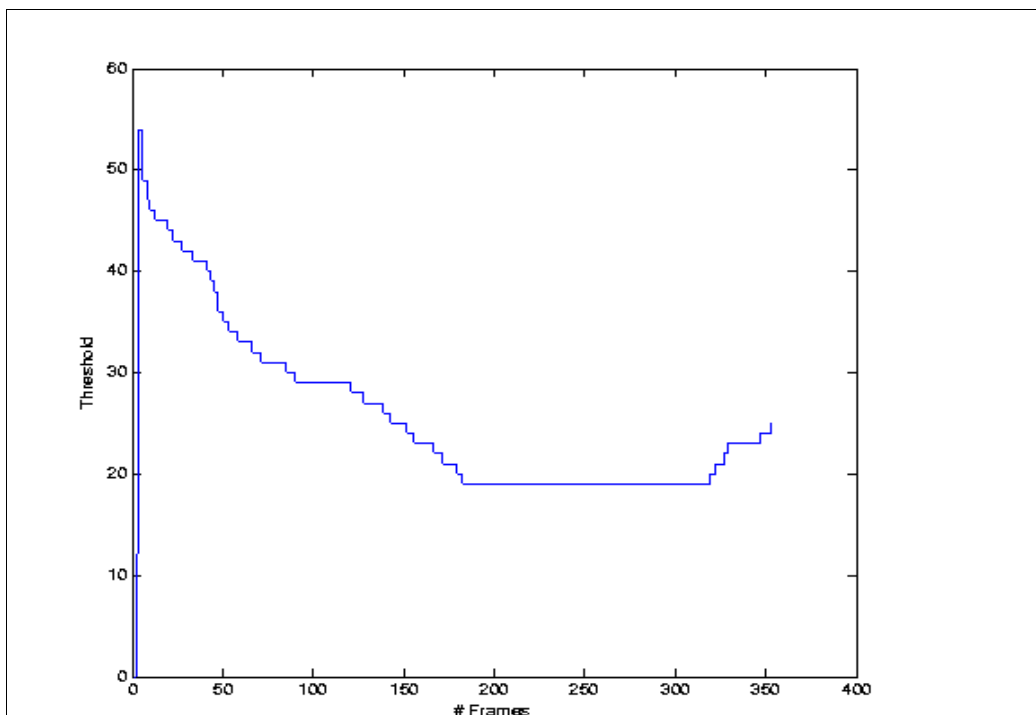


Figure 21. The evolution of the maximum threshold during the sequence of the frames.

c) Processing Times

The processing times are the same for both experiments, since the algorithms are not image dependent.

5. Conclusions

The modifications made in this project have produced improvements in the GMM. algorithm, as shown by the experimental results. Still, the new algorithms can be further improved.

The processing times are longer than before. For real-time applications it would be interesting to reduce that time.

Regarding segmentation, the results are satisfactory. The collinearity method with color version shows the best segmentation results. However it still shows problems with adaptations for the new backgrounds in heterogeneous areas, for example when a chair with holes is becoming background or when the lights have been turned on for a while.

The final algorithm could be used to support some applications, if the GMM result is not considered good enough. Furthermore, some modifications could improve the algorithm.

6. References

- [1] Rudolf Mester, Til Aach and Lutz Dümbgen. Illumination-invariant change detection using a statistical collinearity criterion. In B. Rading and S. Florczyk, editors, Pattern Recognition: Proceedings 23rd DAGM Symposium, Lecture Notes in Computer Science 2191, pages 170-177. Springer Verlag.
- [2] Image Processing, Analysis and Machine Vision. Milan Sonka, Vaclav Hlavac and Roger Boyle. Thomson 3rd edition, 2007. Chapter 16: Motion analysis.
- [3] Segmentation of color images. Adam Chwedyk. Czech Technical University, Prague. Version 1.0, 25th January 2007.
- [4] Real-time Segmentation of color images. Implementation and practical issues in the Blue-C project. Nicolas Galoppo von Borries, Tomas Svoboda, and Stefaan de Roeck. Swiss Federal Institute of Technology, Zürich, Computer Vision Lab. BiWi-TR-261, Version 0.9, 21st August 2003.
- [5] Image Processing, Analysis & Machine Vision – A MATLAB Companion, by Tomas Svoboda, Jan Kybic and Vaclav Hlavac. August 2007.
- [6] Adaptive background mixture models for real-time tracking. Stauffer, Chris and Grimson, W.E.L. IEEE Computer Vision and Pattern Recognition, June 1999.