BACKGROUND SUBTRACTION USING RUNNING GAUSSIAN AVERAGE: A COLOR CHANNEL COMPARISON

Cerman Martin
0625040

Seminar aus Bildverarbeitung und Mustererkennung
183.065

ABSTRACT

Background subtraction methods are widely used to detect moving objects from static cameras. Many different methods have been proposed, reviewed and categorised based on their complexity, speed, memory requirements and accuracy. The Running Gaussian Average is a simple method offering acceptable accuracy and high frame rate while having low memory requirements. Originally this method was proposed for intensity images, but extensions to other color spaces such as (R,G,B), (Y,U,V) and others can be made.
In this paper, we adapt this method to the (R,G,B) color space, compare the results with the single-component intensity color space. Further we discuss the results of the comparison and provide an analysis of the relation between both color models.

Keywords: background subtraction, moving object detection, running gaussian average, color model comparison

1. INTRODUCTION

Background subtraction is a simple but effective method to detect moving objects in a static scene. The main principle of this approach is to detect moving objects, or more generally, motion from the difference of the current frame and a reference frame, which is called a ”background image” or ”background model”. This background image should be acquired by computation during an initialization phase and must be a representation of the scene without moving objects. As stated in [1] a good background model should be characterized by the following features: high precision; flexibility in different scenarios (indoor, outdoor) or different light conditions; and efficiency, in order for detection to be provided in real-time.

Many background subtraction methods have been proposed, trying to effectively estimate the background model from a temporal sequence of frames. The main methods aiming at real-time performance, including Running Gaussian Average, Temporal Median Filter, Mixture of Gaussians, Kernel Density Estimation (KDE) and Eigenbackgrounds, are reviewed, analyzed and categorized based on speed, memory requirements and accuracy in [2]. The Running Gaussian Average is a simple method offering acceptable accuracy and high frame rate while having low memory requirements and therefore has been chosen to be used in this paper. The simple design of this method though limits its functionality to frame sequences provided by a static camera. Over time several methods have been proposed to improve the performance of the Running Gaussian Average method. Koller et al. in [3] proposed to update the background model selectively, based on a binary mask of hypothetized moving objects in the current frame. Tang et al. in [4] combined this method with a frame difference method and proposed a new technique to fill small gaps that the foreground or the moving object may contain. Jabri et al.

in [5] used edge information combined with this method to improve the accuracy of the results. In this paper we use the approach as proposed by Wren et al. in [6] combined with a confidence normalization step. This way we keep the simplicity of the Running Gaussian Average method and enhance We apply this approach on a test frame sequence in the (R,G,B)- and the Grayscale-color model and compare the results using the Receiver Operating Characteristic curves. This paper also provides an original correlation analysis of the mentioned color models based on the comparison results.

The rest of the paper is organized as follows: Section 2 describes the background modelling and background subtraction method in detail. Section 3 briefly describes the (R,G,B) color space and a conversion formula of an intensity image derived. In section 4 we provide a comparison of the Running Gaussian Average method in the mentioned color models. Section 5 presents solutions for shadow suppression and conclusive remarks are addressed at the end of this paper.

2. RUNNING GAUSSIAN AVERAGE

This approach is divided into three main parts: (1) building the background model, (2) performing background subtraction and classifying the scene and (3) updating the background model.

2.1. Building the Background Model

The background is modelled separately for each color channel in the (R,G,B) color space and once for the intensity image. The rationale of this approach is that of fitting a Gaussian probability density function (pdf) on the last n values of each pixel. This results in two images which hold the mean and standard deviation for each channel. The approach builds the background image during an initialization phase and further only updates its parameters instead of fitting the pdf from scratch at each new frame time for increased speed and accuracy. During this phase the method is provided a temporal sequence of frames taken from a static camera containing no moving objects.
The mean is estimated for each pixel from *N* initialization frames as:

$$\mu = \frac{1}{N} \sum_{t=1}^{N} x_t \tag{1}$$

where x_t is the pixel value at frame *t*. After acquiring the average for each pixel, the standard deviation can be estimated as:

$$\sigma^2 = \frac{1}{N-1} \sum_{t=1}^{N} (x_t - \mu)^2 \tag{2}$$

where *N* is the number of initialization frames, x_t is the pixel value at frame *t* and μ is the mean.

2.2. Background Subtraction

We subtract the current frame from the previously estimated mean image, resulting in a difference image for each channel.
Each difference image indicates how much the value of the current frame changed compared to the mean image in the corresponding channel. It is assumed that the camera does not provide a perfect image and the noise matches the Gaussian probability density function, therefore we perform a confidence normalization step for every channel using two thresholds. These thresholds are multiples of the standard deviation of the background model and divide the space of possible pixel values into two intervals, where a distinct classification can be done, and one interval, where further analysis is necessary.
The result of this confidence normalization step represents the confidence of each pixel being classified as foreground. If a value of the difference is below the threshold mσ, we can assume that the corresponding pixel's value changed due to camera noise and therefore it belongs to the background. In this case the confidence is set to 0%. Contrary, if the value of the difference is above the threshold Mσ, that means the value is much more different from the mean value and cannot

be produced by anything else then a moving object, the confidence is set to 100%.

For intermediate values of the difference, the confidence is scaled linearly:

$$C = \frac{D - m\sigma}{M\sigma - m\sigma} * 100 \quad (3)$$

where *D* is the difference value.

When using a multi-component color model, a single confidence image is derived from the maximum of the confidence images of the corresponding channels. Finally a segmentation is achieved by applying a threshold function on the confidence image.

2.3. Updating the Background Model

Cucchiara et al. in [1] stated that "the background model should immediately reflect sudden scene changes such as the start or stop of objects, so as to allow detection of only the actual moving objects with high reactivity." For this reason we introduce a new variable α, the learning rate of the model. For a pixel classified as foreground a low value for α should be chosen to prevent an integration of the moving object into the background model. For a pixel classified as background the height of the value for α should be chosen based on the need for stability (lower value) or quick update (higher value).
At each pixel, the running average is computed for every new frame as:

$$\mu_t = \alpha x_t + (1 - \alpha)\mu_{t-1} \quad (4)$$

where μ_t is the mean computed up to frame *t*, α is the learning rate of the model, and x_t is the pixel value in frame *t*. As we can see, the mean is computed as a weighted mean of all the previous values of the pixel and the current pixel value.
Along with the average, the running standard deviation σ_t is computed as:

$$\sigma_t^2 = \alpha(x_t - \mu_t)^2 + (1 - \alpha)\sigma_{t-1}^2 \quad (5)$$

As stated in [2] the background model consists for each pixel of two parameters (μ_t, σ_t) instead of a buffer with the last *n* pixel values. The on-line cumulative computation of the average and standard deviation and the storage of only two parameters for each pixel explains the high speed and low memory requirements.

3. RGB- TO GRAYSCALE-MODEL CONVERSION

In this section the conversion formula from the (R,G,B) color model to the monochrome color model is discussed. Also a closer look is taken on how the relation between these two models affects the classification rate of the proposed method.

3.1. Conversion formula

A pixel's color representation in the (R,G,B) color space is a 3 dimensional vector (r,g,b) associated with each of the primary colors - red, green and blue. As Sonka et al. in [7] presented, the vector (0,0,0) represents the color black and (*k*,*k*,*k*) is white, where *k* is the quantization granularity for each color channel. Most common number for *k* is 256, which is 2^8, meaning that each color channel has 8 bits of color depth. This implies a color space of 2^{24} distinct color combinations, called Truecolor.

The monochrome color space is represented by an one-dimensional vector with 8 bits per pixel. Sonka et al. in [7] mentioned the YIQ model as being useful in color TV broadcasting, and a simple linear transform of an RGB representation. The YIQ model consists of one luminance channel (Y) and two chrominance channels (I and Q). This model is useful since the Y component provides all the necessary information for a monochrome representation of an image.
The conversion from the RGB color model used in this paper is defined as:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (6)$$

Over time many other Color-to-Grayscale conversions have been proposed and are summarized and well analyzed in [8] by Cadik.

3.2. Relation between the Grayscale- and the RGB-model

Formula (6) is a linear combination, which maps 256^3 RGB-values to 256 Grayscale-values, i.e the 3-dimensional RGB-color space is reduced to a 1-dimensional Monochrome-color space. Visually, all 3-dimensional points in the RGB-color space, which lie on a plane, are projected to one single point in the Monochrome-color space. There is a total number of 256 parallel planes, which cut the RGB-space, each plane for one grayscale value.

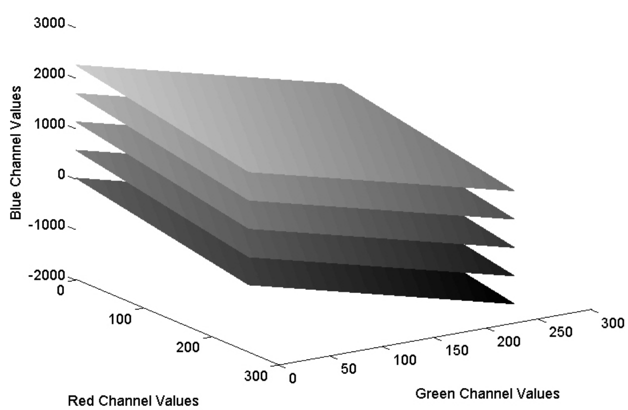

Figure 1: 5 Planes cutting the RGB-color space

Figure 1 shows the planes for the grayscale values 0, 64 ,127, 192 and 255. Here the red and green values range from 0 to 255, and blue from -2000 to 3000 to illustrate the planes with numbers outside the RGB-space, for better understanding.

Figure 2 shows all 256 planes cutting the RGB-space, with only valid RGB values. As can be seen, the planes are not parallel to either of the axis. The consequence is, that the number of RGB values transformed to grayscale values by each plane are different. The plot, showing the possible RGB values for each grayscale value, is illustrated by Figure 3.

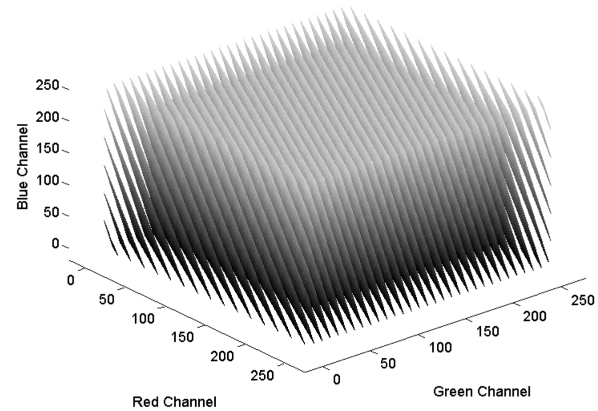

Figure 2: All 256 planes for each grayscale value

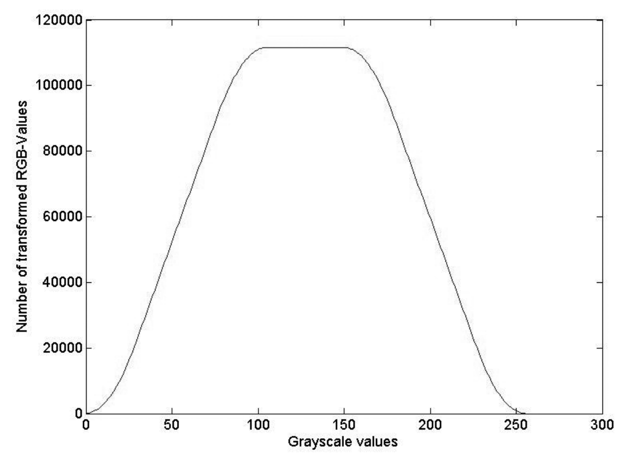

Figure 3: Number of RGB values lying in each grayscale transformation plane

It can be seen, that exactly 1 RGB value (0,0,0) can be transformed with Formula (6) to the grayscale value 0. The same applies for the grayscale value 255, where only the RGB value (255,255,255) results in the correct monochrome value. On the other hand, more than 110000 different RGB values can be used, to get the grayscale value 128.

4. COMPARISON

In this section we first present the Receiver Operating Characteristic (ROC) as an analysis tool for the Running Gaussian Average method. Next we will compare the RGB and the Grayscale color models using ROC curves and finally we discuss the results.

4.1. Receiver Operating Characteristic (ROC Curves)

The result of the Running Gaussian Average method is a binary classification of the image into foreground (positive) and background (negative). The ROC curve is a graphical plot, which visualizes the true positive rate against the false positive rate based on a varied threshold. Since in most cases the results of a binary classificator are not perfect, 4 different outcomes of classification exist:

- True Positive (TP) - predicted and actual value are both positive
- True Negative (TN) - predicted and actual value are both negative
- False Positive (FP) - predicted value is positive and actual value is negative
- False Negative (FN) - predicted value is negative and actual value is positive

From these classification outcomes the True Positive Rate (TPR) and the False Positive Rate (FPR) can be computed as:

$$TPR = \frac{TP}{TP + FN} \tag{7}$$

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

Another important metric representing the overall accuracy of the classificator can be computed as:

$$ACC = \frac{TP + TN}{P + N} \tag{9}$$

4.2. Color model comparison

In this section the proposed methods were implemented and tested to see how the choice of a color model affected the accuracy of the classification. To get a most precise comparison basis, the Ground Truth was created manually for every 40 frames of the video sequence. Further the ROC-Curves were used as a means for measuring the accuracy of the classification.

(a) Frame 45 (b) Frame 630

Figure 4: Frames 45 and 630 from a 700-frame sequence of an airport scene

For the experiment a temporal frame sequence taken from an outdoor static camera was chosen, presenting an airport scene during daytime, shown in Figure 4. The first 35 frames were used for the background initialization during training phase. For both color models, every 40 frames the results of the Running Gaussian Average method has been compared with the Ground Truth, which represents the binary classification by a human. During the confidence normalization step, following parameters were used:

$$m{=}3.5$$
$$M{=}10.0$$

Figure 5 shows the ROC curves of the method results for the Grayscale- and RGB-color model, tested with 8 different threshold values.

The results of the RGB-color model show a higher True Positive Rate and a slightly higher rate of False Positives. Generally less pixels were classified as foreground with the RGB-color model. A lower threshold resulted in proportionally less false classified pixels leading to a slightly higher accuracy. After a certain threshold value has been reached, no more change in the accuracy of the method can be noticed.

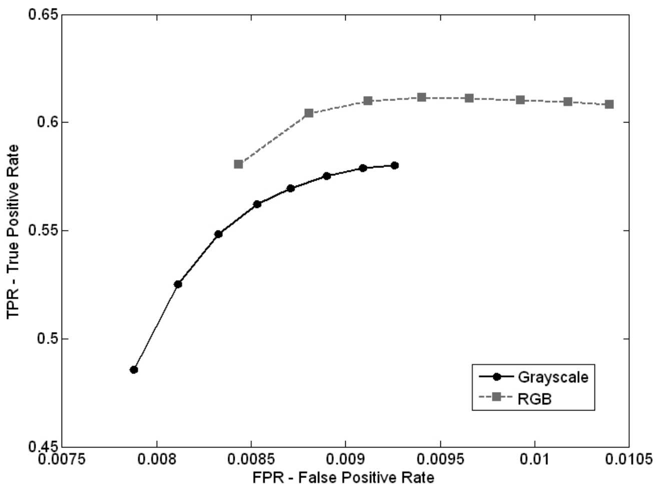

Figure 5: ROC-Curves of the tested data

Table 1 shows the accuracy of both color-models with 8 different thresholds, where the best results have been underlined:

Threshold	Grayscale	RGB
0.05	0.97680	0.98228
0.10	0.97930	0.98280
0.15	0.98058	**0.98283**
0.20	0.98128	0.98275
0.25	0.98157	0.98263
0.30	0.98178	0.98248
0.35	**0.98188**	0.98233
0.40	**0.98188**	0.98219
0.45	**0.98188**	0.98219

Table 1: Accuracy of the color models

For an analysis of falsely classified pixels for each color model the threshold with the highest accuracy has been chosen. Both the number of False Positive and False Negative pixels have been counted for both color models at each frame and plotted, as seen in Figure 6 and 7.

4.3. Discussion of the results

Table 1 showed for the test video sequence, that the accuracy of both color models was almost identical, when the right threshold after the confidence normalization step was chosen. Generally in the RGB-color model less pixels were classified as foreground. When chosing the best threshold for each color model, the ratio of the True Positive Rate and the False Positive Rate remained for both nearly the same, resulting in the almost identical accuracy. The RGB-model had the tendency to produce more False Positives as the threshold value was raised. Figure 6 and 7 show the comparison of the False Negatives and False Positives of both used models. The two rates were during the test video sequence almost the same. At approximately frame 500 more moving objects started to enter the scene. This resulted in higher False Negative values in the Grayscale model and higher False Positive values in the RGB model.

Section 3.2 covered the relation between the RGB- and the Grayscale-model, when Formula (6) was used for the transformation. It was shown, that RGB-values lying on the same plane are transformed to the same grayscale value. If the RGB-image contains colors, which are grouped together around a few planes, a distinction between these colors is very hard in the Grayscale-image and can very easily lead to bad accuracy of the classification. If the colors in the RGB-color space, seen as 3-dimensional points, are equally scattered in the whole color space, the classification will provide for both color spaces similar results. The verification of this theory is thought a subject for further research.

5. CONCLUSION

In this paper, we have presented the Running Gaussian Average, which is a simple Background subtraction method offering acceptable accuracy and high frame rate while having low memory requirements. Further a Receiver Operating Characteristic (ROC) analysis of the presented method in the Grayscale- and RGB-color space has been performed and visualized. Our testing data showed, that the accuracy of the proposed method was almost identical for both color spaces. This knowledge could be further used, to improve the performance of real-time surveillance systems and systems, where a high frame rate is crucial.

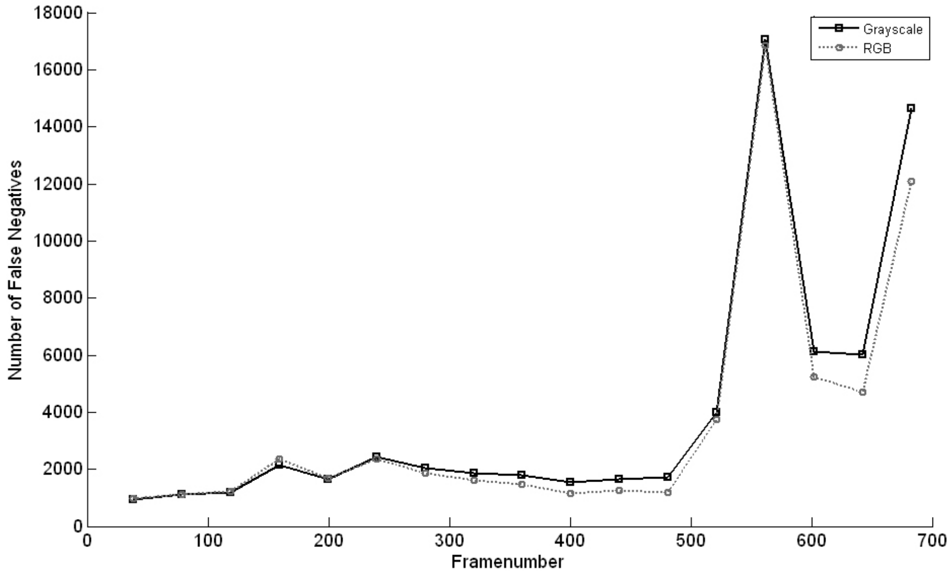

Figure 6: False Negative comparison of the Grayscale- and RGB-Color model.

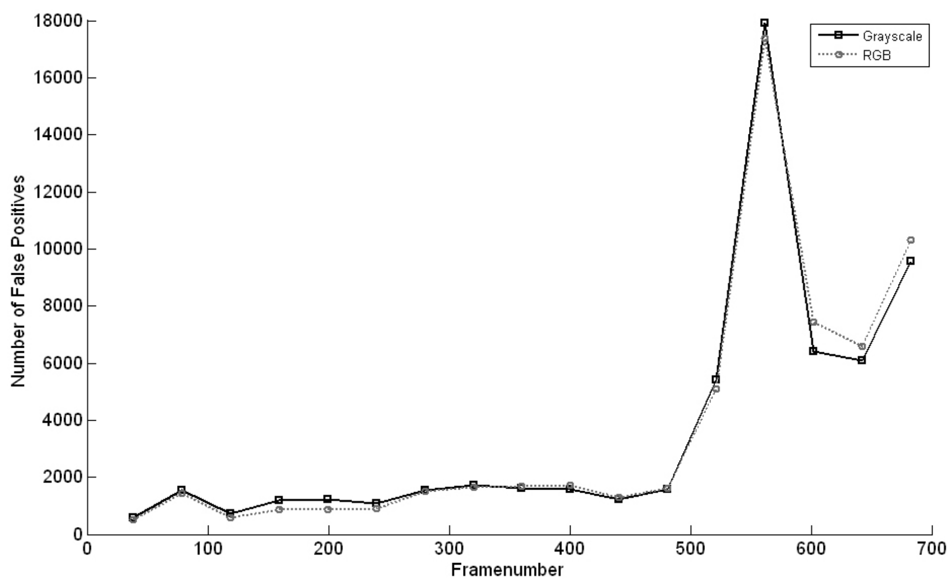

Figure 7: False Positive comparison of the Grayscale- and RGB-Color model.

6. REFERENCES

[1] Cucchiara, R., Piccardi, M., Prati, A.: "Detecting Moving Objects, Ghost, and Shadows in Video Streams." IEEE Transactions on Analysis and Machine Intelligence 25(10), 1337-1342. (2003)

[2] Piccardi, M., "Background subtraction techniques: A review." in Proc. IEEE Conf. Syst., Man Cybern., 2004, vol. 4, p. 3099-3104.

[3] Koller, D., Weber, J., Huang, T., Ma, J., Ogasawara, G., Rao, B., Russell, S.: "Towards Robust Automatic Traffic Scene Analysis in Real-time." in Proc. ICPR ,1994, p. 126-131.

[4] Tang, Z., Miao, Z., Wan, Y.: "Background Subtraction Using Running Gaussian Average and Frame Difference" in ICEC ,2007, LNCS 4740, p. 411-414.

[5] Jabri, S., Durie, Z., Wechsler, H., Rosenfield, A.: "Detection and Location of People in Video Images Using Adaptive Fusion of Color and Edge Information." in ICPR, 15th International Conference on Pattern Recognition (ICPR'00), vol. 4, p. 4627-4631. (2000)

[6] Wren, C. R., Azarbayejani , A., Darrell , T., Pentland, A.: "Pfinder: Real-time tracking of the human body." IEEE PAMI, 19: p.780-785, 1997

[7] Sonka, M., Hlavac, V., Boyle, R.: "Image Processing, Analysis, and Machine Vision", 2nd ed. lbany, Brooks/Cole, 1999, ISBN 0-534-95393-X, p. 23-25.

[8] Cadik, M.: "Perceptual Evaluation of Color-to-Grayscale Image Conversions", Pacific Graphics 2008, vol. 27, Nr. 7