Computer Vision: Fourth Assignment

Students Giacomo Lugano (5400573), Claudio Tomaiuolo (5630055)

Abstract

Detecting regions of change in multiple images of the same scene taken at different times is of widespread interest due to a large number of applications in diverse disciplines, including remote sensing, surveillance, medical diagnosis and treatment, civil infrastructure, and underwater sensing [1]. This article presents a comparison of the effectiveness of different ways of change detection implemented using MATLAB. For change detection, techniques such as Image Differencing and running average were used to update the background model. The objective of this work is to develop an approach to efficiently and quantitatively evaluate threshold algorithms for change detection in a surveillance environment.

Introduction

Video surveillance systems originated in the analogue environment in the not too distant past. The first monitoring experiments were developed and deployed during the Second World War to monitor the German army's missile launches. It was not until the early 2000s that these systems had an application in video surveillance. Its widespread diffusion and increasing use in the private sphere revolutionised fields such as investigation and criminology, facilitating investigations by the judiciary and speeding up their timescales. Unfortunately, the task of detecting anomalous events monitored by these systems requires the continuous vigilance of security personnel. As many video surveillance installations employ a large number of strategically placed cameras, this task places a heavy burden on the operator, with inevitable loss of concentration and thus poor performance. One solution is to automate change detection and object tracking. This information can be made available to alert the operator if illegal or abnormal activity is detected. Change detection is the first step needed to solve this problem. Change detection involves the quantification of temporal effects using multi-temporal data sets. Frame differentiation and/or background subtraction followed by a thresholding is a commonly used method for change detection if images are co-registered. A surveillance camera recording at 30 frames per second does not have to deal with rapid changes in lighting and illumination conditions. Therefore, an almost constant environment can often be assumed between adjacent frames. For the rest, there are many more or less sophisticated algorithms [2][3] that can be used to generate and maintain background images, so it is preferable to use simple image differentiation. Thresholding is a fundamental technique applied in many image processing applications.

Change Detection

Most digital change detection techniques used in this study require the selection of a threshold value to determine areas of change. The greyscale difference map is usually converted to binary form and subjected to a predetermined threshold value to obtain a change/no-change classification. However, the threshold value is critical, as too low a value will swamp the difference map with spurious changes, while too high a value will suppress significant changes [4].
In the case where the camera is moving and an I_{ref} reference image is assumed, image differentiation is widely used due to its simplicity of implementation and interpretation. It involves subtracting the I_t frame from the I_{ref} reference image, pixel by pixel. A critical aspect of the image differentiation method is to decide where to place the threshold boundaries between the changing and non-changing pixels displayed in the binary image M_t. Thus, mathematically,

$$M_t(x,y) = \begin{cases} 1 & if \quad |I_t(x,y) - I_{ref}(x,y)| > \tau \\ 0 & otherwise \end{cases} \tag{1}$$

where I_t is the image acquired at time t, I_{ref} is the reference image and τ is the threshold.

The reference image or, more generally, the background model, is a 'picture' of the empty scene. The easiest way to calculate it is to have the first N frames, assuming that at the beginning the scene is stationary:

$$B = I_{ref} = \frac{1}{N} \sum_{t=1}^{N} I_t \tag{2}$$

A single constant background model may not be sufficient because the scene may change over time, e.g. when the light changes. A solution to these problems is to use a background model that is updated from time to time with the current averaging method, so the previous formula can be rewritten as follows:

$$M_t(x,y) = \begin{cases} 1 & if \quad |I_t(x,y) - B_{t-1}(x,y)| > \tau \\ 0 & otherwise \end{cases} \tag{3}$$

where B_{t-1} is the dependent background model at the previous instant.

To update the background, the condition must be executed is:

$$B_t(x,y) = \begin{cases} B_{t-1}(x,y) & if\ the\ pixel\ is\ moving \\ (1-\alpha)B_{t-1}(x,y) + \alpha I_t(x,y) & if\ the\ pixel\ is\ static \end{cases} \tag{4}$$

with $0 < \alpha < 1$.

The running average of the background model is fairly robust to moving objects in the scene, incorporates stable changes (at a rate proportional to α) and is simple and computationally efficient.

Results

Task 1

This is an example of working on the video surveillance sequence using a simple background obtained as an average between two empty frames. After loading the two empty images of the scene and calculating a simple background model, each image of interest was loaded sequentially, performing change detection by showing the Frame, the Background Model and the Binary Map. As can be seen in the three figures, the threshold is of crucial importance for change detection and implementation of the binary map.

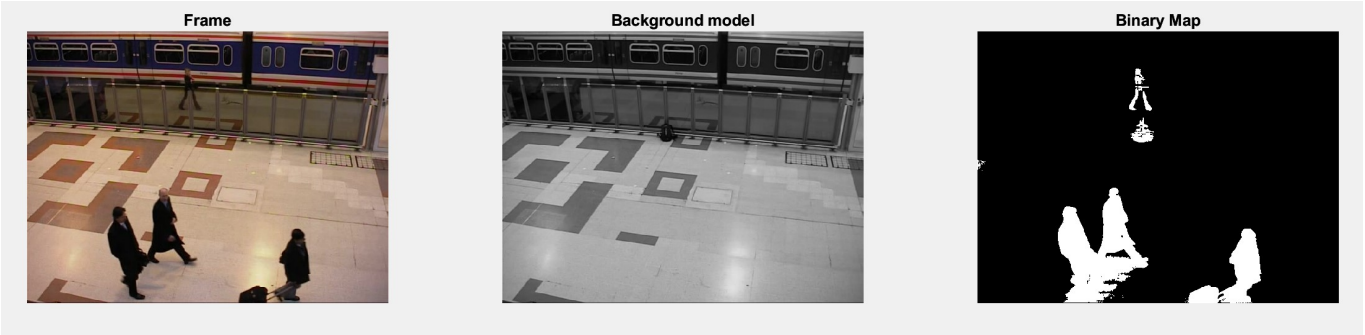

Figure 1: Results with $\tau = 20$.

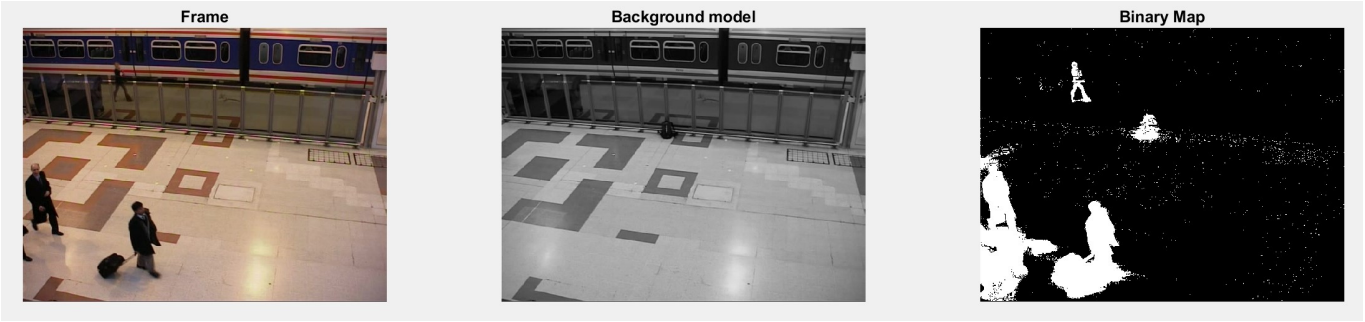

Figure 2: Results with $\tau = 10$.

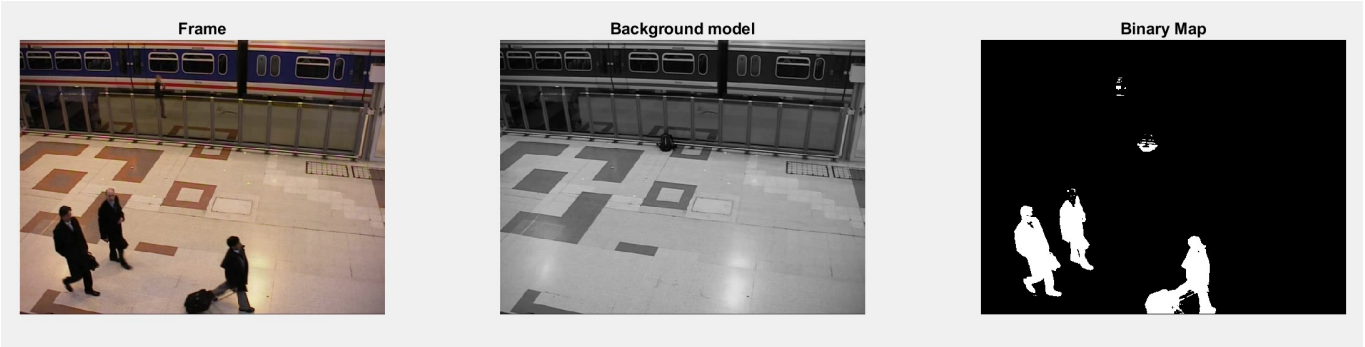

Figure 3: Results with $\tau = 60$.

It is necessary to find the right threshold value, depending on the scene of interest. As we can see in Figure 2, with a value of tau=10, the recognised changes also include the shadows of passers-by. In Figure 3, with tau=60 on the other hand, a person's head is not recognised, because the colour of their skin is very similar to that of the floor. The same happens with the body of the person in the background, obscured by the glass.

Task 2

This is another example of working on the video surveillance sequence, but this time using a background model based on running average to incorporate changes in the scene. To initialise the background, the first N (e.g. 5) frames are used. Then, after initiating the detection of changes in the sequence of frames, the background can be updated using the running average method. Figure 4 shows the result obtained with $\tau = 20$ and $\alpha = 0.1$.

Figure 4: Results with $\tau = 20$ and $\alpha = 0.1$.

Task 3

The previous example is repeated with the implementation of the algorithm on frames concerning a different video surveillance scene.

Figure 5: Results with $\tau = 20$ and $\alpha = 0.1$.

Task 4

Example of implementation of a simple tracking system using a change detection using a background model based or running average. The MATLAB function *bwconncomp()* is used to identify the connected components in the binary map. A trajectory line is also drawn on the images, taking into account the centre of the bounding boxes.

Figure 6: Tracking System implementation example.

Conclusions

A variety of change detection techniques always produce different accuracies depending on the nature of the study area. The purpose of this study was to examine the effectiveness of two different background models. It was concluded that of the two methods tested, the latter is more reliable in real-life applications, while the former is more suitable for the implemented algorithm. In general, we can say that the different change detection algorithms in Matlab are simple and user-friendly, but they must be compared to find the best method for specific applications and regions. In addition, the methods tested in this work are very simple. More efficient and complex algorithms are already available and widely used. The opensource library OpenCV (from which it takes its name: Open Source Computer Vision) already makes many of these available, especially with regard to tracking algorithms.

References

[1] R. J. Radke, S. Andra, O. Al-Kofahi and B. Roysam, *Image change detection algorithms: a systematic survey*, in IEEE Transactions on Image Processing, vol. 14, no. 3, pp. 294-307, March 2005, doi: 10.1109/TIP.2004.838698.

[2] Stauffer C., Grimson W., (1999) *Adaptive background mixture models for real-time tracking*, Conf. Computer Vision Pattern Recognition, pp. II:246–252.

[3] Y Ren, C Chua and Y Ho, (2003) *Pattern Recognition*, Lett., 24 (1–3), pp. 183-196.

[4] Paul, L.R., (1998) *Thresholding for Change Detection*, Sixth International Conference of Computer Vision, 4–7 Jan 1998, Bombay, 274–279.