Università degli studi di Genova

DIBRIS

Informatics, Bioengineering
Systems Engineering, Robotics and System Engineering

Computer Vision

Assigment 4

Report Laboratory 7

Student:
Andrea Chiappe,
Fabio Guelfi

January 7, 2024

Report

The aim of this laboratory is to compare the results of two motion estimation algorithms applied to the same sequence of images. The algorithms used are: the *optical flow estimation's algorithm*, Lucas-Kanade, and the *change detection algorithm* with running average. For all these two algorithms, we have to make a strong assumption about the brightness constancy of the image (because they both suffer the change in light intensity seen as a change in the image implying a false positive).
The Lucas Kanade is an algorithm used, in sequence of images, to study the motion of the key points, and to give back the motion field of the object, the direction and the speed of the object of an image with respect to the previous one. This approach relies on the assumption that within a small local area of an image, the optical flow remains relatively constant. By establishing individual windows around specific points of interest and deriving a system of equations, we aim to capture and analyze the variations in optical flow across these neighboring points. This is a linear system, so we can solve it and estimate the optical flow without a difficult computation. The output of our function is a vector $u = [U, V]$, where U and V are the shift in x and y of the point.

For the change detection algorithm we also consider the assumption that the camera is stationary and only the scene is moving. This algorithm is used to find significant changes between sequence of images, makes a pixel-by-pixel comparison of the following image to a background model.
In this assignment we use two different computation of the background: a *static* background (it will stays always the same), and a *running average* version, where the background is upgraded during the sequence making him change. The first model (static background) is based to the idea that if the pixel is static (has not changed with respect to the background), it will colored **black** (0),if the pixel is moving with respect to the background image, it becomes **white**.
It is base on this formula:

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

Where I_x is the current image, I_ref is the background image, and τ is a threshold.
We try lots of value of τ and in our opinion the best is in $30 < \tau < 40$ for the video-surveillance sequence of images. We decide to use $\tau = 35$. We use this value because is the best compromise for remove the shadows of the man in the front and find the man on the background. The man in the background is not completely detected because the railing to his right is dark like its suits, and also his head is impossible to find because correspond with the white line of the train and the difference of the magnitude of the pixel is less than τ.

The other algorithm used is the change detection with *running average*. In

this case the background will upgrade during the sequence depending on the scale factor α.
The formula that we used is:

$$M_t(x,y) = \begin{cases} 1 & if \;\; |I_x(x,y) - B_{t-1}(x,y)| > \tau \\ 0 & if \;\; otherwise \end{cases} \quad (2)$$

where:

$$B_t(x,y) = \begin{cases} B_{t-1}(x,y) & if \quad D_t(x,y) > \tau' \\ (1-\alpha)B_{t-1}(x,y) + \alpha I_t(x,y) & if \quad otherwise \end{cases} \quad (3)$$

where:

$$D_t(x,y) = |I_t(x,y) - I_{t-1}(x,y)|$$

D_t give us the information about the motion of the pixel with respect to the previous image, not with respect to the background.
With this formula we built another binary map where all the pixel which are moving are colored in white.
All the used parameters are:

- B_{t-1} is the Background at the previous instant time;
- τ is the same threshold used for the case without running average, This is a threshold to build a binary map comparing the image through the relationship with the previous background. If this difference is $> \tau$ it means that the pixel is different with respect to the background and so it is set to white. If not, the pixel is set on black;
- τ' in a threshold used to understand if the pixel in the current image is moving with respect to the previous one;
- α' is a value $0 < \alpha' < 1$. This parameter control the amount of the information that will change in the background. If $\alpha' = 1$ the next background will be the actual image. If $\alpha = 0$ the background consider is the one consider before, this case is the same without the running average.

In our first case, in the video **surveillance sequence**, we set this value:

$\tau = 40$, to capture all the pixel differences between the background and the actual image. We decide a not too low value because in the scene there is a quite clear difference in color between the pixel moving and the pixel of the scene. The man who are walking have a black suits, and the scene in background have lighter color.
$\tau' = 35$, This parameter is used as a threshold to capture the moving pixel between the previous and the actual image. We decided to use a value that is not too high for the same reason of τ. In the scene, the people who move have a much darker color than the parts of the scene that remain static. Thanks to

this there is a big difference between the pixels that move and the others.
$\alpha = 0.8$, we decide to use a high value for this parameter. We do this choice according to our intention to detect the motion of the people moving in the scene. In this case the people are constantly moving so it is important that the background will be update fast, that is, taking more into account the previous nearby images, and not all the previous images in our sequence. In practice we want our background to have "little memory" and therefore update quickly in accordance with the scene.

Despite the choice of these parameters, we still have some imperfections in our binary image. In fact, the movement creates white trails behind the pixels that are moving. To avoid this problem we could raise the value of α, or the value of τ', but in doing this we lose the movement of the person behind the railing since much of it is covered by the railing and its head coincides with the white line of the train, therefore little detectable due to the similar colors.

For this sequence of images, we also used the algorithm to estimate the optical flow, Lucas Kanade. In this case, unlike the previous algorithm that returned a binary image, it provides us a vector field containing both magnitude and direction of the optical flow. As expected, the larger magnitude of the optical flow is concentrated in the region where people in the foreground are walking towards the right. However, we also observe field vectors in the background corresponding to the person walking in the background. Additionally, we encounter small disturbances, namely field vectors appearing in static parts of the scene. These could be due to reflections of light, or possibly, being a simple algorithm, there might be some small computation errors.

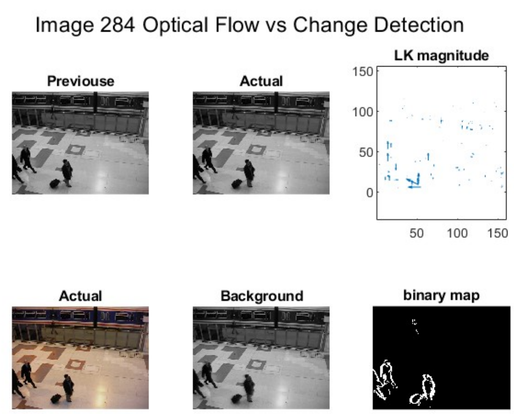

Figure 1: Video surveillance changes example

In conclusion, for this type of application, the change detection algorithm with running average is more suitable, as it involves the movement tracking of people against a constant background. As we will see later, the Lucas Kanade algorithm makes more sense in the case of tracking an object in a scene, effectively capturing the motion field of that specific object.

In the second case, analyzing the sequence of the **sphere**, we notice that the rotating image has little color contrast. Therefore, for the change detection algorithm with a running average, we use the following parameter values: $\tau = 10$, $\tau' = 5$, with $\alpha = 0.3$.
This way, we manage to capture the movement of the sphere. Also in this case, we notice a slight trailing effect due to the color shades present on the sphere. In this context, the Lucas Kanade algorithm is also effective as it maps the motion field of the sphere.
Both algorithms are effective in this case. The choice between the two depends on the type of processing needed and on the specific application of our motion detection. For example, if we want to track the movement, we could use the Lucas Kanade algorithm. If, on the other hand, we are more interested in understanding the movement in a more graphical and human-readable way, we could use the change detection algorithm with a running average, and obtain a binary map.

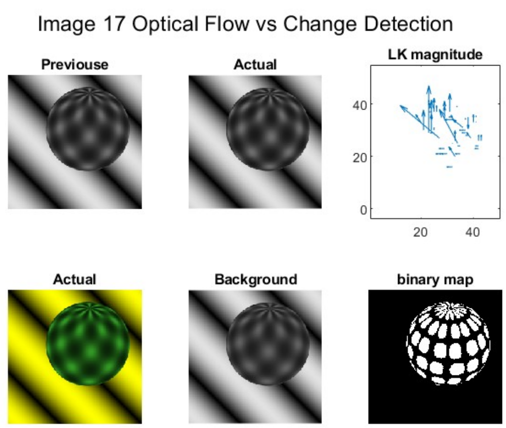

Figure 2: Sphere frame of changes for the 16 images