

Combining Template Matching and Background Subtraction Techniques to Detect Objects in Infrared Video Sequences

I.Q. Pham^{*}, R. Jalovecký and M. Polášek

Department of Air Electrical Systems University of Defence, Brno, Czech Republic

The manuscript was received on 9 March 2016 and was accepted after revision for publication on 2 September 2016.

Abstract:

In this paper, an algorithm is introduced combining a template matching technique with background subtraction to detect moving targets on a road captured on video using a static camera. The designed algorithm based on combining the two techniques is to reduce the difficulties in background estimation and improve the processing time, etc. The proposed algorithm was tested in two different video sequences. The first video sequence was filmed with stabilized camera, while the second video was acquired with unstable camera. The results in both cases indicated that the background subtraction algorithm is successful in detecting moving objects and the template matching technique is suitable for object replacement from background frame to acquire the standard background frame.

Keywords:

background estimation, background subtraction, infrared images, normalized 2-D cross correlation, object detection, template matching technique, video surveillance

1. Introduction

Background subtraction is an important technique in many computer vision applications such as video surveillance, people counting, human gesture recognition, moving object detection and tracking, traffic monitoring or in military application like imaging seekers. Many background subtraction methods have been proposed in publications such as running Gaussian average, temporal median filter, mixture of Gaussians, Kernel density estimation, etc. There are three representative approaches for background subtraction methods. First, they are pixel-based methods that extract foregrounds using each pixel independently. The second group of background subtraction techniques are the block-based methods. The third class of background subtraction approaches are the texture-based methods [1-4].

^{*} *Corresponding author: Department of Air Electrical Systems, University of Defence, Kounicova 65, CZ – 61200 Brno, Czech Republic, Telephone: +420 973 445 009, E-mail: mylemonwater.piq86@gmail.com.*

In this paper, the presented method allows the user to choose the background frame at the start. Then, the algorithm automatically updates the background frame to eliminate environmental changes such as daytime, weather changes, etc. However, the background frame can contain objects of interest. So, a template matching technique needs to be employed to delete these objects from picture and set suitable intensity to pixels of removed objects.

There are many similarity criteria for template matching that are the sum of absolute differences (SAD), the locally scaled sum of absolute difference (LSAD), the zero - mean sum of absolute differences (ZSAD), the sum of squared differences (SSD), the locally scaled sum of squared differences (LSSD), the zero - mean sum of squared differences (ZSSD), the normalized cross correlation (NCC), the zero - mean normalized cross correlation (ZNCC), and the sum of hamming distances (SHD) [6]. In this paper, similarity criteria of the normalized cross correlation were employed. And the normalized cross correlation coefficient used is 0.8 [6, 7].

The algorithm employed the following formula to calculate the correlation coefficient $c(u, v)$ [8, 9]:

$$c(u, v) = \frac{\sum_{x,y} [f(x, y) - \bar{f}_{u,v}] [g(x-u, y-v) - \bar{g}]}{\sqrt{\sum_{x,y} [f(x, y) - \bar{f}_{u,v}]^2 \sum_{x,y} [g(x-u, y-v) - \bar{g}]^2}}, \quad (1)$$

where

- $f(x, y)$ is the pixel value at coordinates x, y of image,
- $g(x, y)$ is the pixel value at coordinates x, y of template image,
- \bar{g} is the mean of the template,
- $\bar{f}_{u,v}$ is the mean of $f(x, y)$ in the region under the template.

2. Principle of the Algorithm

Fig. 1 outlines the principle of the algorithm. There are $NumFram$ frames in the video sequence. The parameter $step$ is an increment between the frames of video sequence. In this study, the $step$ value is set as 1. The algorithm begins at frame $i = a$ and finishes at frame $i = b$. In case of $i = a$, the unreal colour background frame $B = F_a$, and then the background frame B is converted to grey scale.

The parameter $flag$ has the initial value equal to 0. If $flag = 1$, the background frame is changed, and if $flag = 2$, then the moving objects in background frame are removed. Parameters c_1, c_2 are calculated by the following formula.

$$c_1 = a + k, \quad (2)$$

$$c_2 = i + k, \quad (3)$$

where i is the frame number, when the background frame has chosen or has changed to suit the environmental conditions. The value of k is based on the experience that there are some moving objects in the background, so all objects will leave initial positions after k frames. The constant k is dependent on the velocity of the objects. In this paper, we used $k = 60$ equivalent to 2 s. At this time, it is easy to determine objects location and accurate shape. The acquired results help us to achieve more accurate results when

the template matching technique is employed. In the template matching block, the algorithm also removes moving objects on the background frame. We use the normalized cross correlation coefficient with the value equal to 0.8 in order to define clearly that new objects do not appear in the position of the moving objects in background frame. In a case that the cross correlation coefficient is less than 0.8, the algorithm will

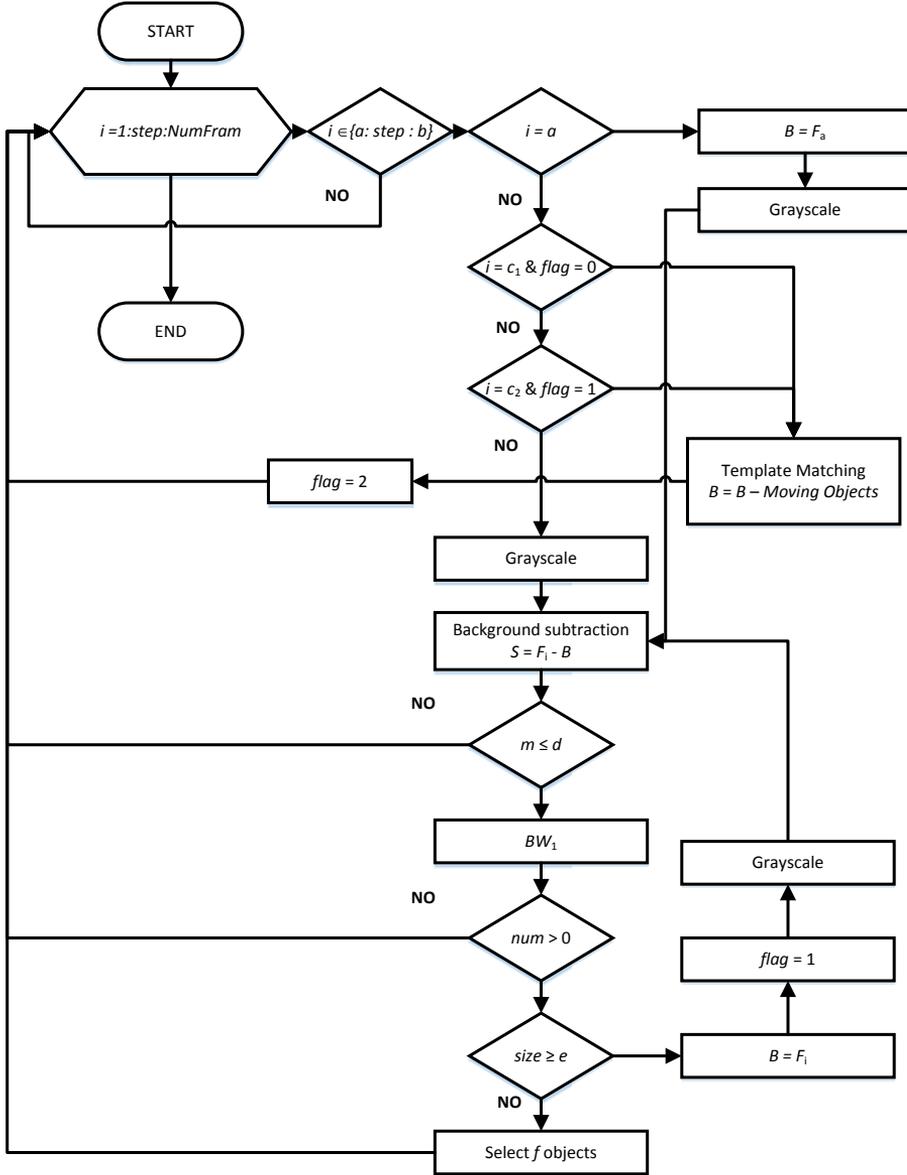


Fig. 1 Flowchart of the algorithm

use the intensity of pixels in the current frame to replace the intensity of pixels in the background frame. The algorithm also converts each current colour frame F_i to grayscale frame, then it employs the background subtraction technique so as to find a matrix

S . On the basis of matrix S , the mean value m can be calculated. It can be empirically predicted that the value m cannot exceed d . In this paper, d is equal to 14. In a case of exceeding the value of 14, the algorithm will automatically switch to the next frame F_i . In other cases, the algorithm will compute the binary image BW_1 . The objects of interest should be then in the BW_1 which is calculated by the following formula:

$$BW_1 \equiv S, \quad (4)$$

$$S = F_i - B, \quad (5)$$

where every element value of matrix S must be equal or greater than d_1 . In this paper, minimum pixel value d_1 is 10. On BW_1 , we will find the number of objects (num) and the size of each object. If the BW_1 does not contain any object ($num = 0$), the algorithm will automatically switch to the next frame. In other cases, the algorithm will check the size of the largest object in BW_1 , which is stored in variable $size$. If this value exceeds a limit e , the algorithm automatically selects the current frame as new background frame meaning $B = F_i$ and the value of $flag$ will be 1. In other cases, the algorithm will only select f objects and show them. In this paper, f is 20 and e is computed by the following equation:

$$e = \frac{S_1}{6}, \quad (6)$$

where S_1 is a size of the matrix S , as multiplication of the width and height.

3. Experimental Results

In this paper, results of the designed algorithm are presented in two video sequences. The first video sequence has been acquired in stable air, while the second video is little blurred in initial frames of the video sequence. In both cases we obtained credible results and also the proposed algorithm proved its effectiveness when interference occurred. In each video sequence the following results are presented: the initial background frame F_a , the changed background frame F_i , the background frame at the time when the algorithm removed the moving objects containing F_{i+k} . 95 frames in each video sequence were tested, where the initial frame $a = 5$ and b is 100, but only the results at 6th, 46th and 86th frames are shown. The size of the frame in all video sequences is 120×160 pixels.

3.1. First Video Sequence

Fig. 2 shows a change of background frame during the video sequence. At the start (initial time), the algorithm chooses the background frame $B = F_5$. Then the algorithm continues to track two conditions to change the background frame, but these conditions were not fulfilled. Therefore, the algorithm employed the template matching technique at frame F_{65} and then it also removed moving objects to acquire a new background frame (on the right). Moving objects were omitted completely by the algorithm.

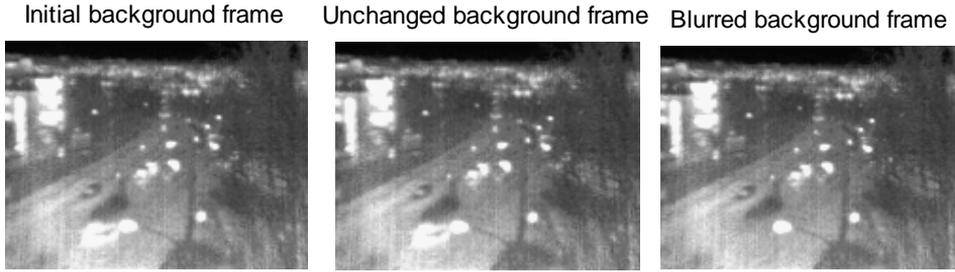
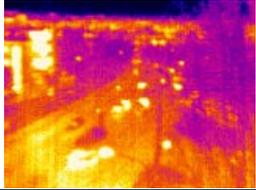
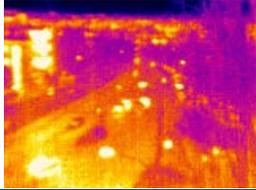
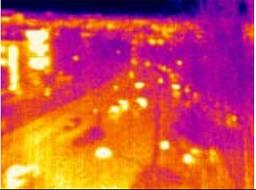
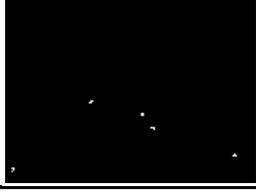


Fig. 2 Background frame of 1st video changed according to the time

The following table shows results for a few selected frames (see Tab. 1).

Tab. 1 Results of the algorithm in a few frames of 1st video

Type	6 th	46 th	86 th
Colour			
Grey scale			
Bi-nary			

Tab. 1 indicates the result of the proposed algorithm for 3 frames. For all three frames the algorithm did not detect objects in the middle of frame. These objects belong to the objects of interest, but they did not change their position after 86 frames. If they changed their position at the next frame, the algorithm would successfully detect them and replace their intensity of pixels by more appropriate intensity such as background intensity.

3.2. Second Video Sequence

As in the previous case, Fig. 3 shows changes of background frame. At the start (initial time), the algorithm chose background frame $B = F_5$. Then the algorithm continued to check two conditions to change the background frame. One of these conditions had happened and therefore, the background frame changed at the frame F_8 . From this point,

the algorithm employed the template matching technique at frame F_{68} and then the algorithm also removed the moving objects to acquire a new background frame (on the right). The background frame still contained heads of objects after being removed. Rectangle number 1 denotes the head of the first object after being removed. Rectangle number 2 denotes the head of the second moving object. Two objects were removed from the background.

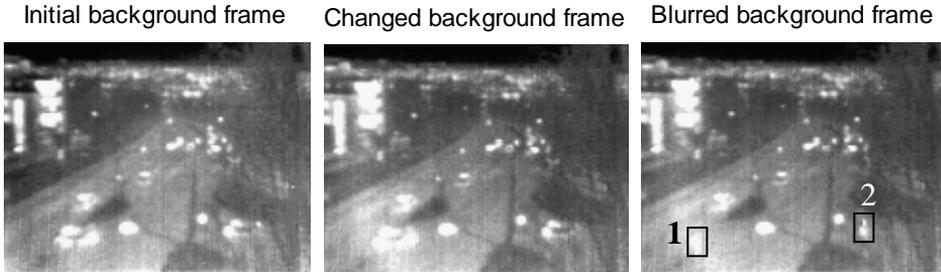


Fig. 3 Background frame of the 2nd video changed according to the time

The Tab. 2 shows the results for a few selected frames.

Tab. 2 Results of the algorithm in a few frames of 2nd video

Type	6 th	46 th	86 th
Colour			
Grey scale			
Binary			

Tab. 2 indicates the result of the proposed algorithm for three frames. For the same reason as in the previous case, the algorithm did not detect objects in the middle of the frame in all the three frames. In this video sequence, the noise was clearly visible in the 6th frame. Because the new background frame still contained parts of the object, the result in frame 46 was unclear about the size of the object. In frame 86 the noise was visible, but still the algorithm was able to detect objects.

In this study, the focus was on moving object detection and objects that did not move were temporarily ignored. These objects stood still on the road for a few seconds. As soon as these objects started moving again, the proposed algorithm successfully found the accurate background frame and detected all the objects.

4. Conclusion

The following conclusions can be drawn from the results we have acquired. The proposed algorithm employs both techniques (background subtraction and template matching technique) to solve two main problems. The background subtraction technique solved the first problem with moving objects detection. However, there were some difficulties with predicting a standard background meaning where the background frame contained all objects that were not needed to be detected. This was due to the fact that the background subtraction is based on the principle of frame differences, i.e. difference of the current frame and the background frame, to acquire a foreground frame that would contain the objects of interest. If the background frame contained the objects of interest, then the effectiveness of background subtraction was reduced.

In order to solve the effectiveness drop mentioned above, the algorithm used the second technique—template matching technique to compare sub-images of objects on the background with sub-images on the current frame. The algorithm removed these objects from the background and replaced inappropriate intensities that were not similar to the intensities of the real background to get a more suitable background frame.

In this paper, the new algorithm was tested in two different video sequences. The first video sequence was filmed with stabilized camera, while the second video was acquired with unstable camera. The results in both cases indicated that the background subtraction algorithm is successful in detecting moving objects and the template matching technique is suitable for object replacement from background frame. However, there were some objects of interest that were not moving for a short period of time on the background. These objects were removed from the background after a period of time that depended on the time in which the object was motionless. After a certain time, a perfect background frame was acquired again.

Based on the above mentioned results, the following conclusions can be also drawn. Firstly, the proposed algorithm automatically accommodates the requirements such as timing change, environment change, the impact of noise or the change of lighting conditions. A user can choose the background at any time and the algorithm automatically adjusts the background to respond to the environmental conditions. Secondly, the algorithm can also solve difficulties with memory. The algorithm requires a memory with capacity of about the size of one frame. Thirdly, the algorithm does not require too much time for background frame creation. Therefore, this algorithm could be used for real time applications.

Finally, the proposed algorithm did not solve the problem of stand still objects detection in the background frame. This insufficiency will be covered in our future research.

Acknowledgement

The work presented in this article has been supported by the Czech Republic Ministry of Defence – University of Defence development program “Research of sensor and control systems to achieve battlefield information superiority“.

References

- [1] LEE, J. and PARK, M. An Adaptive Background Subtraction Method Based on Kernel Density Estimation. *Sensors*, 2012, vol. 9, no. 9, p. 12279-12300. ISSN 1424-8220. DOI: 10.3390/s120912279.
- [2] SZELISKI, R. *Computer Vision: Algorithms and Applications*. London: Springer, Texts in computer science. 2010. 812 p. ISBN 978-1-84882-934-3.
- [3] BOURDONNAYE, A., DOSKOČIL, R., KŘIVÁNEK, V. and ŠTEFEK, A. Practical Experience with Distance Measurement Based on the Single Visual Camera. *Advances in Military Technology*, 2012, vol. 7, no. 2, p. 51-58. ISSN 1802-2308.
- [4] POLÁŠEK, M. and NĚMEČEK, J. Methods to Increase Resistivity against Jamming in Selected Types of Optoelectronic Seekers. *Advances in Military Technology*, 2010, vol. 5, no. 2, p. 45-56. ISSN 1802-2308.
- [5] AHUJA, S. *Normalized Cross Correlation*. Wordpress.com [online]. [cited. 2016-03-03]. Available from: <<http://siddhantahuja.wordpress.com/tag/normalized-cross-correlation/>>.
- [6] PHAM, Q.I. and POLÁŠEK, M. Using template matching technique for object detection in infrared images. In: *Transport Means 2014*. Kaunas: Kaunas University of Technology, 2014, p. 257-260. ISSN 1822-296X.
- [7] PHAM, Q.I., JALOVECKÝ, R. and POLÁŠEK, M. Using template matching for object recognition in infrared video sequences. In: *Designing an Air transportation system with multi-level resilience*. Prague: ALR International, 2015, p. „3D3-1“-“3D3-15“. ISBN 978-1-4799-8939-3.
- [8] BRUNELLI, R. *Template matching techniques in computer vision: theory and practice*. Chichester, U.K.: Wiley, 2009, 338 p. ISBN 978-0-470-51706-2.
- [9] *Normalized 2-D cross-correlation - MATLAB normxcorr2* [online]. United States: MathWorks, 2015, [cited 2016-01-03]. Available from: <<http://www.mathworks.com/help/images/ref/normxcorr2.html>>.