2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

Real-time moving object segmentation algorithm implemented on the Eye-RIS focal plane sensor-processor system

Tamás Fülöp[†] and Ákos Zarándy[‡†]

†Peter Pazmany Catholic University
Budapest Hungary
[‡]Computer and Automation research Institute of the Hungarian Academy of Sciences
Budapest, Hungary
Email: zarandy@sztaki.hu

Abstract– Novel real-time object segmentation algorithm, optimized for the Eye-RIS sensor-processor device is introduced. The special feature of the new method is that it is based on spatial-temporal information rather than spatial purely. Since that it provides smaller sensitivity for illumination changes, handles shadows correctly even on monochrome images, requires simple computational framework.

1. Introduction

Moving object segmentation is the key initial step of practically all security-surveillance and other monitoring algorithms. It is also called foreground-background separation, where the moving objects should be distinguished from the stationary or quasi-stationary background. There are two classes of these algorithms. The first assumes stationary camera platform, while the second considers moving camera platform. Our method belongs to the first one.

The naive approach of this problem is to capture a reference image, when there is no moving object in the scene, and compare it to the forthcoming images. In case of absolutely stable camera, strictly unchanging illumination conditions and capturing parameters, and stable background, the difference indicates the position of the moving objects only. However, these conditions are practically never fulfilled, especially not in outdoor environment, which leads to false object identifications. The reason of this failure is that the background is slowly changing on the pixel level due to the changing of illuminations (sun shines from different angle, shadows are moving, clouds are coming and going, etc), which accumulates significant changes over time.

To overcome this situation, one has to continuously update the reference image (also called background image) to introduce the latest slow changes. This immediately brings up the stability/plasticity dilemma, because quick update leads to loosing slow moving objects, while slow update cannot follow the illumination changes caused by clouds. One has certainly find a trade-off in each applications. However it can be seen clearly, that those algorithms requires less fine-tune, which are less sensitive to the illumination variation.

State-of-the-art methods [1][2] nowadays are based on the temporal statistics of the individual pixel values. They build up a background model, which contains the statistical description of the distribution of the history of the last few hundred pixel values in each pixel position. The temporal pixel value sequence is clustered, and the number of pixels in each cluster is permanently updated. If a new pixel value belongs to an existing cluster with many pixels, it is considered background. However, if it belongs to a new cluster, or to an old cluster but with few pixels from the history, it is considered to be object. All of this clusterization is done pixel wise, and is done for each pixel.

The method performs very well, however it requires per pixel complex Gaussian classifications and storage of the statistical descriptors, which is roughly 18-20 bytes per pixel. To reduce the illumination sensitivity, the method uses color images, which requires extra processing and needs more memory space for parameters. This high memory requirement does not allow the implementation of this method on morphic analog focal processor array device, like the Eye-RIS system [3], where there is a processor behind each pixel with limited memory.

As a contrast, we propose to use gradient based segmentation method. Since the gradient is not illumination dependent our method performs better under changing illuminating conditions even in monochromatic case. Some gradient based segmentation methods were introduced in the literature previously [5]. The specialty of our method is, that it is based on subtraction and threshold operations only, which are very easy to implement on analog morphic computer array. We will introduce a real-time efficient implementation of the algorithm on the Eye-RIS system [3].

After the introduction, the algorithm will be described in Section 2. Then, the Eye-RIS implementation is introduced in Section 3. Finally we conclude the paper.

2. Description of the algorithm

The main motivations behind the algorithm are (i) to reduce the dependency from illumination changes, and (ii) to be able to distinguish shadows using monochromatic sensor and finally (iii) to use simple processing steps only. Illumination changes modify the pixel values, hence non-spatial, intensity based segmentation methods are expected to generate false segmentation output (artifacts) when illumination changes rapidly in time, especially in case of shadows of quickly moving objects (Fig. 1).

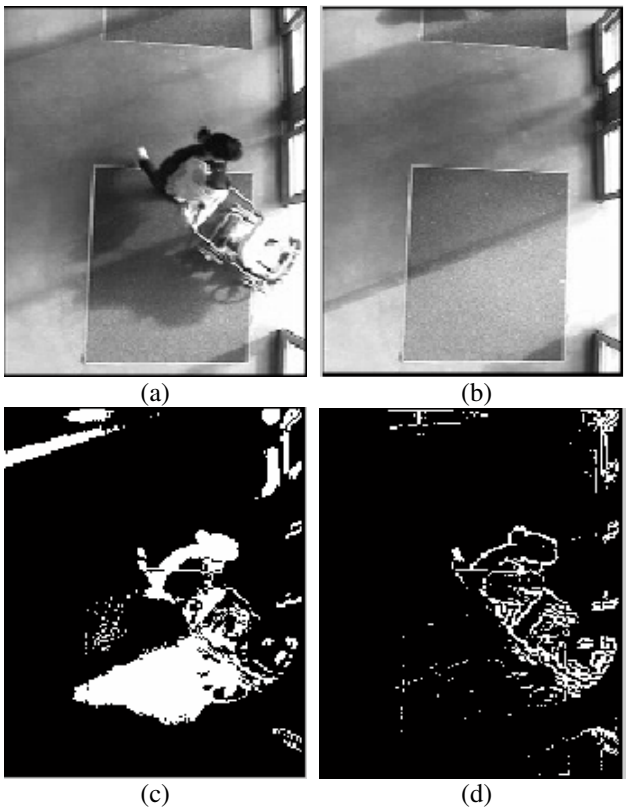


Figure 1. Comparison of the intensity based and gradient based methods. (a) current image, (b) reference image, (c) foreground using intensity based method, (d) foreground using gradient based method. As it can be seen, the intensity based method finds the shadows also, not just the moving objects.

In our proposed method, rather than monitoring the individual pixel intensity levels, we follow the relations (local differences) between the neighboring pixel levels. The idea is based on the assumption that if a location is lighter than its neighboring location, it will remain lighter even if the global illumination level is changed, because the local lightness relations depends mostly on the reflectance of the surface and not on the illumination level. In case of a logarithmic image sensor, these differences lead to the lightness map of the image as Land defined it back to 1983 [4].

To simplify the method, we do not calculate the exact gradient. Rather than that, we analyze the local difference composition in each pixel. This means, that we test to each direction, whether the neighbor is significantly greater, roughly equal, or significantly smaller. In this way, we get the following neighborhood index.

$$D_{N\,i,j} = \begin{cases} if\ (I_{i,j} - I_{i,j-1}) < -\varepsilon\ \ then & -1 \\ if\ \left|I_{i,j} - I_{i,j-1}\right| \le \varepsilon\ \ then & 0 \\ if\ (I_{i,j} - I_{i,j-1}) > \varepsilon\ \ then & 1 \end{cases}$$

$$D_{S\,i,j} = \begin{cases} if\ (I_{i,j} - I_{i,j+1}) < -\varepsilon\ \ then & -1 \\ if\ \left|I_{i,j} - I_{i,j+1}\right| \le \varepsilon\ \ then & 0 \\ if\ (I_{i,j} - I_{i,j+1}) > \varepsilon\ \ then & 1 \end{cases}$$

$$D_{E\,i,j} = \begin{cases} if\ (I_{i,j} - I_{i+1,j}) < -\varepsilon\ \ then & -1 \\ if\ \left|I_{i,j} - I_{i+1,j}\right| \le \varepsilon\ \ then & 0 \\ if\ (I_{i,j} - I_{i+1,j}) > \varepsilon\ \ then & 1 \end{cases}$$

$$D_{W\,i,j} = \begin{cases} if\ (I_{i,j} - I_{i-1,j}) < -\varepsilon\ \ then & -1 \\ if\ \left|I_{i,j} - I_{i-1,j}\right| \le \varepsilon\ \ then & 0 \\ if\ (I_{i,j} - I_{i-1,j}) > \varepsilon\ \ then & 1 \end{cases}$$

The reason, why three levels (two steps functions) are proposed is the following. In many images in artificial environment (indoor, or urban outdoor), there are large flat areas with the same lightness (color). The pixel value differences in these areas are close to zero. If we do not use the middle class (roughly equal) just the simple step function only, these values will belong to +1 and -1 according to some sensor disturbance or light illumination gradient changes.

By calculating these maps from both the current input and from the reference image, the maps should be compared using the following rule:

$$G_{i,j} = \sum_{X=N,W,S,E} \left| D_{RX\,i,j} - D_{CX\,i,j} \right|$$

where D_{RX} is the neighborhood index of the reference image in the X direction (X can be North, West, South, East). Similarly D_{CX} is the neighborhood index on the currently captured image.

We considered a pixel is to be a foreground, if larger than a threshold:

$$G_{i,j} > t_h .$$

In this way, we get a binary foreground map. To delete small noises, it is recommended to apply some morphological filtering to the foreground map.

The reference image (background) can be generated in different ways. The simplest method is to use the previous frame as the reference. Though this already works quite well for structured objects, there are several problems of this approach, because (i) it looses the object immediately when it stops, moreover (ii) it generates a leg (tale) behind the objects.

Other method is to do an exponential averaging; however, this accumulates traces of frequently appearing objects. More advanced, but still simple method is to apply larger weights, if a pixel is considered to be background, and smaller, when it is a foreground. In this way, the background is updated faster than the foreground, however misclassified foreground pieces does not remain foreground forever.

3. Implementation of the algorithm on the Eye-RIS system

Efficient implementation of any algorithm on a particular hardware requires resource considerations. We have to understand, what operations are "cheap", which ones are "expensive", and what the system constraints are. On a fine-grain focal-plane sensor-processor chip, where each pixel is corresponding to a processor, typically those processing operations are cheap, which are directly supported by the hardware. On the other hand, the hard constraint is the memory limitation, and sometimes the IO bottleneck.

In the case of the Q-Eye chip, the image capturing, the arithmetic operations, and the logic operations are very cheap, and the IO is fast. There 9 grayscale and 4 binary memories in each processor (pixel), which makes possible to store the sub-results while processing a frame. Certainly, due to the analog nature of its internal image memories, the background image cannot be kept on-chip for longer time. However, this is not a significant problem, because the fast IO enables to load it in each cycle, and save it, when it was updated.

The flowchart of the algorithm is shown in Fig. 2. The algorithm starts with image capturing. Then, the neighborhood index is calculated on the captured image, according to the shown pseudo code. It is followed by the comparison of the neighborhood indexes on the input image and the reference image. On the result, we have performed morphological filtering, to get the foreground image, which contains the moving objects. This is needed to the reference image update also for the selective weighting. Finally the neighborhood index is calculated for the reference image.

Fig. 3 and Fig. 4 show two typical snapshots of a video sequence. We have modified the opening of the iris of the lens on the Eye-RIS system, which generated the over illuminated situation. As it can be seen, the system was capable to adapt to the strongly changed illuminations values, and provide good results. The segmentation took only 5 ms on the Eye-RIS system.

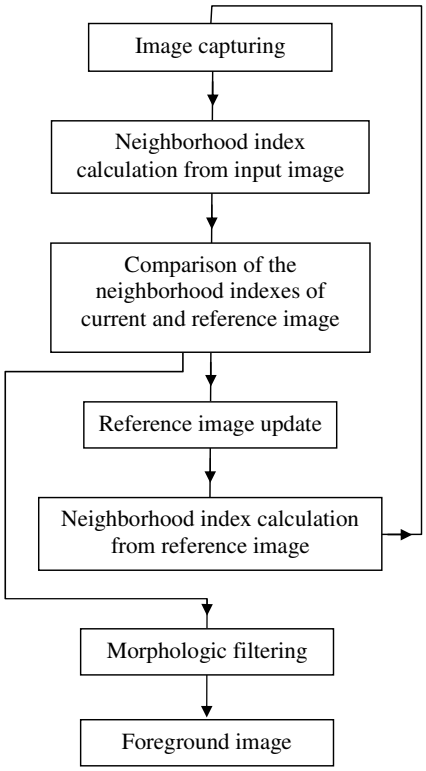
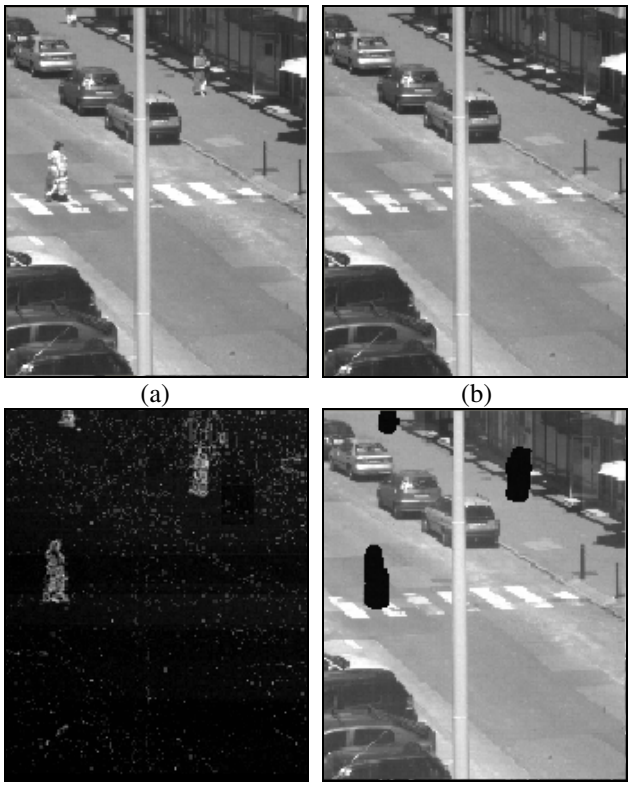


Figure 2. Flowchart of the algorithm

Algorithm 1 Neighborhood index calculation

```
1.  function NEGIHBOUR_INDEX (I_input, I_reference, th)
2.  result=0;
3.   for dir=['right','down','left','up'] do
4.
5.     I_input_shifted=shiftimage(I_input,dir)
6.     I_th_neg_diff_inp =threshold(I_input- I_input_shifted, th)
7.     I_th_pos_diff_inp = threshold(I_input_shifted - I_input, th)
8.
9.     I_reference_shifted=shiftimage(I_reference,dir)
10.    I_th_neg_diff_ref=threshold(I_reference - I_reference_shifted, th)
11.    I_th_pos_diff_ref = threshold(I_reference _shifted - I_reference, th)
12.
13.    result=result+( I_th_neg_diff_inp  XOR  I_th_neg_diff_ref)
14.    result=result+( I_th_pos_diff_inp  XOR  I_th_pos_diff_ref)
15.
16.  end
17.
18. return result;
```



(a) (b)

(c) (d)

Figure 3. Snapshot of a video flow captured during normal daylight illumination. (a): captured image; (b): actual background; (c) difference of the neighborhood indexes ($G_{i,j}$) before thresholding; (d): foreground (thresholded $G_{i,j}$) superimposing back to the original image.

Conclusion

In this paper, we have shown a new gradient based foreground-background segmentation method. The new method enables real-time implementation on embedded focal-plane sensor-processor devices, like the Eye-RIS system.

Acknowledgments

The authors would like to thank to AnaFocus to provide the Eye-RIS system and technical support for this work. The support of the ONR Award No.: Grant N00014-07-1-0350 is greatly acknowledged.

References

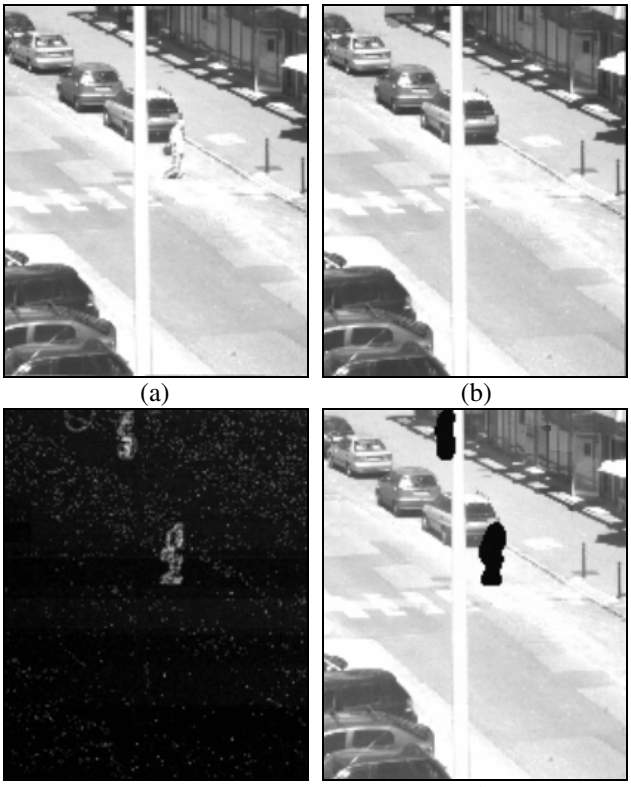
[1] A. Elgammal, R. Duraiswami, D. Harwood and L. S. Davis "*Background and Foreground Modeling using Non-parametric Kernel Density Estimation for Visual Surveillance*", Proceedings of the IEEE, July 2002

[2] A. Elgammal, D. Harwood, L. S. Davis, "*Non-parametric Model for Background Subtraction*", 6th European Conference on Computer Vision. Dublin, Ireland, June/July 2000

[3] www.anafocus.com

[4] E.H. Land, "Recent advantages in retinex theory and some implications for cortical computations: Color vision and the natural image", Proc. Natl. Acad. Sci. USA Vol. 80, pp. 5163-5169, August, 1983, Physics.

[5] A. Câmara Lara, R. Hirata Jr, "A morphological gradient-based method to motion segmentation", Proceedings of the 8th International Symposium on Mathematical Morphology, Rio de Janeiro, Brazil, Oct. 10 –13, 2007, MCT/INPE, v. 2, p. 71–72.



(a) (b)

(c) (d)

Figure 4. Snapshot of a video flow captured in an over illuminated situation. As it can be seen, the algorithm has adapted to the extreme illuminations, and extracted the moving objects only. (a): captured image; (b): actual background; (c) difference of the neighborhood indexes ($G_{i,j}$) before thresholding; (d): foreground (thresholded $G_{i,j}$) superimposing back to the original image.