ASTRONOMY AND SPACE SCIENCE
eds. M.K. Tsvetkov, L.G. Filipov, M.S. Dimitrijević, L.Č. Popović,
Heron Press Ltd, Sofia 2007

Spot Object Motion Detection Algorithm

P.M. Gugutkov[1], **H.D. Lukarski**[2]

[1]Department of Electronics, Technical University, Sofia, Bulgaria

[2]Space Research Institute, Bulgarian Academy of Sciences, Sofia, Bulgaria

Abstract. Moving objects in image sequences can be detected by analyzing the pixel–level difference between successive frames. This paper describes a method for motion detection that is strong in small size (spot) objects. It is based on combining an image processing with a motion detection algorithm. The method is realized on an embedded DSP, which gives the opportunity for excellent flexibility in adaptation of the system to different environments.

1 Introduction

One of the most difficult tasks of Motion Detection Systems (MDS) is to detect small objects with low contrast where signal to noise is poor, and to be able to track such targets long enough to identify it. We are interested in locating moving objects in sequential images where the object are a few pixels in size and has low contrast with its background. Simple methods of image differentiation are not appropriate. We needed more robust algorithm that is fairly simple to compute (because of our main aim to implement it in embedded DSP), has the potential to run in real-time, and is relatively immune to noise. After some research the authors decided that there is a need for an effective decision theoretic approach to the detection of small low-contrast objects. In this paper we expand upon the ideas of [2], exploiting useful properties of wavelet filters to provide detection of the motion of these small, low-contrast objects. An additional effort was to try to improve the signal to noise ratio by removing clutter from the image using useful properties of wavelet filters to provide a robust method for moving objects detection even when the object size is as small as 6-9 pixels. The processing time for this algorithm is typically few scores of millisecond per frame with some optimization of the computational time, and the method is highly parallelisable making it ideal for real time implementations. In the next section we describe the wavelet filtering, which we use for image preprocessing to decrease noise in the image and in such way to improve the small low contrast objects detection algorithm.

2 Decision

Our emphasis is on the detection of small targets only a few pixels across. For such objects a wavelet description is most suitable as wavelets encapsulate information on the discontinuities in a very tight way. We could describe the object in terms of wavelet decomposition and then apply a matched filter to it. However, the objects we seek have very little structure associated with them. The classic way to detect moving objects in the image sequence taken with a camera is by differentiation neighboring images. The problem with this approach is that many noise variations are also detected, and if our object is small it can prove hard to separate out the moving target from the noise.

Wavelet filters have been proven to be superior to Fourier methods for detecting localized high frequency behavior in an otherwise as scaled and dilated copies of a single function $\psi(x)$ having the property:

$$\int_{-\infty}^{\infty} \psi(x)dx = 0\,.$$

The wavelet functions are then defined by:

$$\psi_{mn}(x) = 2^{-m}\psi(2^{-m}x - n)$$

where m is the dilation factor and n represents the shift. Wavelets are associated with a *scaling* function ϕ(x) via the *two-scale equation*:

$$\phi(x) = \sum_n h_n \phi(2x - n)$$
$$\psi(x) = \sum_n g_n \phi(2x - n)$$

The finite set of coefficients h_n and g_n constitute low pass and high pass filters, respectively. For our purposes we have used wavelet filters of lengths 4 to 8 that provide the generalization of image differentiation to combining 4 up to 8 images in a sequence together. The high pass nature of the filter then detects the passage of a grey level boundary across each pixel and so highlights the outline of the object. The advantage of using these wavelet filters is that they are robust to noise. By combining image data over several frames the motion stands out above the noise, providing a much higher signal to noise ratio. For each filtered image we compute channel energy in local neighborhoods by averaging the absolute values of the wavelet coefficients.

After wavelet filtering we apply the change detection algorithm. The goal of a change detection system is to generate a change mask q consisting of binary labels $q(k)$ for each pixel k on the image grid. The labels either take the value "u" ('unchanged') or "c" ('changed'). In order to determine the label $q(k =$

i) for pixel i we start with the grey-level difference image $d(k)$ between two successive frames, and compare the sum of absolute differences Δ_i within a sliding window w_i with N pixels and center i to a threshold T:

$$\Delta_i = \frac{2\sqrt{2}}{\sigma_u} \sum_{k \in w_i} |d(k)| \underset{\text{u}}{\overset{\text{c}}{\gtrless}} T.$$

Here, σ_u is the noise standard deviation of the grey level differences in stationary areas, which is assumed to be constant over space. Normalization by σ_u – which is known for a given camera or easily estimated – makes T insensitive to different noise levels. Given the null hypothesis H0 (grey-level differences $d(k)$ in w_i only due to noise), and modelling the grey level differences $d(k)$ as independent and Laplacian distributed, Δ_i obeys a χ^2 distribution with $2N$ degrees of freedom. Change detection can then be formulated as a significance test, where the threshold value is determined in terms of an acceptable false alarm rate α. The "significance" α is equivalent to the probability that Δ_i exceeds the threshold T, given H0:

$$\alpha = \text{Prob}(\Delta_i > T|\text{H0}).$$

For a given false alarm rate α, the threshold T is determined from tables of the χ^2-distribution. The decision rule then is

$$\Delta_i \underset{\text{u}}{\overset{\text{c}}{\gtrless}} T.$$

Whenever Δ_i exceeds T, we decide $q(i) = $ c, otherwise $q(i) = $ u. As mentioned above, this global-threshold-based decision procedure is prone to two kinds of errors, false positives and false negatives. The basic idea to reduce these is to decrease the decision threshold inside changed areas, and to increase it outside. This exploits the prior knowledge that changed and unchanged regions correspond to objects or background, which usually are of compact shape. Formally, this prior knowledge can be expressed by modeling the change masks as realizations of Gibbs fields. This leads to a variable threshold t which adapts to the label constellation within a pixel's neighborhood when making a decision. The higher the number n_i of "changed" pixels found in this neighborhood, the lower the threshold is:

$$t(n_i) = T + (4 - n_i)B$$

with $0 \leq n_i \leq 8$ when using a 3×3 neighborhood. The parameter B is a positive-valued potential, which determines the range of $t(n_i)$. If $n_i = 0$, the threshold t reaches its maximum value of $T{+}4B$. The minimum value of $T{-}4B$

results if $n_i = 8$, i.e. all neighbors of pixel i are labelled as "changed". If there are as many "changed" as "unchanged" labels, we have $n_i = 4$, and $t = T$. Clearly, the threshold $t(n_i)$ favors the emergence of compact, smoothly shaped object masks, and reduces scattered decision errors caused by noise. The labels $q(k)$ necessary to calculate $t(n_i)$ can be obtained as follows: assuming a raster scan from the upper left to the lower right image corner, the labels in the causal part of the 3×3-neighbourhood of pixel i are already known. The labels in the no causal part of the neighborhood are approximated by keeping labels from the previous change mask. Note that when overwriting the previous change mask during the raster scan, this constellation emerges automatically.

The combination of wavelet and applying change detection algorithm after that gave us very good results in order to determine the goals of our experiment. Our team implemented this algorithm in embedded DSP system and made standalone digital camera independent from use of dedicated personal computer for calculation. Of course the system can be connected to PC for visualization of image sequence and image archive. This feature is very helpful but all the calculations are made outside the PC and this provide multiple systems connection to only one computer. The system and algorithm is very suitable for star observation and telescope guide.

3 Results

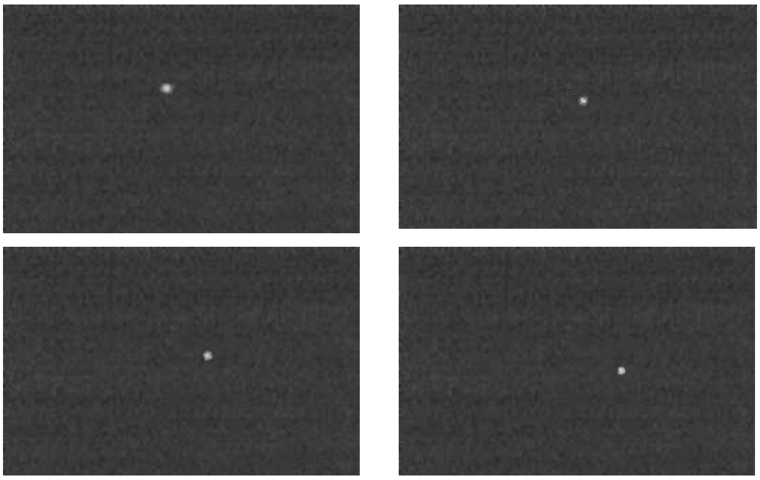

Figure 1. Original images before processing.

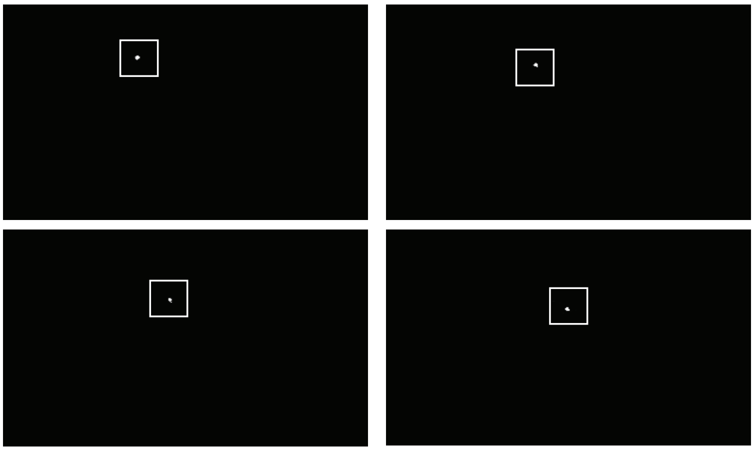

Figure 2. Output images after processing.

References

[1] *Open Source Computer Vision Library*, Intel Corporation, 2001.

[2] S. Blostein and T. Huang. *Detecting small, moving objects in image sequences using sequential hypothesis testing*. *IEEE Transactions on Signal Processing*, 39(7):1611–1629, 1991.