International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B3, 2012
XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia

SPECTRUM-BASED OBJECT DETECTION AND TRACKING TECHNIQUE FOR DIGITAL VIDEO SURVEILLANCE

Boris Vishnyakov, Yury Vizilter and Vladimir Knyaz

The Federal State Unitary Enterprise "State Research Institute of Aviation Systems"
Moscow, Viktorenko 7, Russia
vishnyakov@gosniias.ru, viz@gosniias.ru, knyaz@gosniias.ru
http://gosniias.ru

Commission III/4: Complex Scene Analysis and 3D Reconstruction; III/5: Image Sequence Analysis

KEY WORDS: object, detection, tracking, monitoring, video, image

ABSTRACT:

This paper presents a motion detection and object tracking technique for digital video surveillance applications. Motion analysis algorithms are based on processing of multiple-regression pseudospectrums. Complete object detection and tracking scheme is described. Results of testing on public PETS and ETISEO test beds are outlined.

1 INTRODUCTION

The video surveillance is one of the key technologies of modern security systems. Digital video surveillance presumes the visual control of some territory with one or more video cameras, that allows storing and viewing digital video data, continuously evaluating the state of controlled region and detecting some changes in observed scene as "security events".

The main drawback of traditional video surveillance systems providing raw video to a human operator is a serious decreasing of operator's response capability, while the system is growing in size. This problem is especially urgent in case of city-level surveillance systems. Well-known business case is an implementation of video surveillance system in London, Great Britain including tens of thousands of cameras in a single network and more than half a million cameras in the whole city. Unfortunately, it did not provide a serious reduction of crime incidents or increasing of crime detection rate. Now we know that it is not enough just to broadcast cameras' video to the surveillance center. Video should be processed and alarms should be generated in real-time to attract the attention of operator in critical situations.

So, the design of high-performance intellectual video analytic systems is a very actual practical task. Moreover, such intelligent systems can address both security and counterterrorism objectives, and can be of use in some business applications. For example, they can collect statistical information about the attendance of observed object, distribution of visitors over time, main routes of movement, etc. Other possible application is a traffic monitoring and so on.

The Motion analysis is a basis of all intelligent video surveillance technologies. In particular, it provides the fundamentals for automatic detection and tracking of moving objects and automatic detection of new or disappeared objects of observed scene. It is the well-studied area of computer vision including many different techniques. The brief overview of these techniques is given in next section.

This paper contains a description of proposed technique accompanied with testing results on PETS (PETS video database, n.d.) and ETISEO (ETISEO video database, n.d.) public video test beds.

2 RELATED WORKS

The motion detection and tracking problem is widely studied all around the world. There are lots of methods and algorithms, that detect motion and trace moving objects. Let us dwell on main approaches in video analysis task. First one is the optical flow approach (Horn and Schunck, 1981, Nagel, 1983, Barron et al., 1994). It was the first mentioned in (Horn and Schunck, 1981). This approach is based on finding the pixel speed from previous to current frames. Let $I(k)$ be an input image pixel matrix with width w and height h on frame number k. It is assumed that the brightness of a point remains constant during a short period of time, which is expressed by the equation

$$\frac{dI(k)_{x,y}}{dk} = 0.$$

Hence we get an equation

$$\nabla I(k)_{x,y} \cdot (u,v)^T + \frac{\partial I(k)_{x,y}}{\partial k} = 0,$$

where $(u,v)^T$ – vector of pixel movement.

Hence optical flow speed $(u,v)^T$ can be found via iteration method from (Horn and Schunck, 1981, Barron et al., 1994). In different books and papers the number of required iterations varies, but to achieve a good result you have to make over 100 iterations over full image, what is very time consuming.

The optical flow approach is useless if image sequence contains large amount of pixel noise. The next correlation approach (Anandan, 1989, Singh, 1992) is based on computing correlation function of some area and minimizing it in surrounding region to find the best match for it and speed vector $(u,v)^T$. Most of correlation algorithms are based on minimization of SSD-function (Sum of Squares Difference):

$$SSD_k(x,y,u,v) =$$

$$= \sum_{i=-n}^{i=n} \sum_{j=-n}^{j=n} W_{i,j} \left(I_{x+u+i,y+v+i}(k+1) - I_{x+i,y+i}(k) \right),$$

where $W_{i,j}$ is weight function for the area.

In (Anandan, 1989) SSD-function is sequentially optimized by the Laplacian pyramid. Minimum is found for all levels of the pyramid to begin with the highest level (the smallest image) and dropping to the lowest level (the whole image). Speed vector is being obtained more accurate on each level. In (Singh, 1992) minimum of SSD-function is found through iteration process.

But correlation approach is not robust too because it strongly depends on invariability of scene brightness. In (Heeger, 1988) frequency approach is proposed. This approach is based on "power" function, evaluated as the Gabor filter (Gabor, 1946) with frequencies L_x, L_y, ω:

$$R(u,v) = \exp\left\{\frac{-4\pi^2\sigma_x^2\sigma_y^2\sigma_k^2(uL_x + vL_y + \omega)}{(u\sigma_x\sigma_k)^2 + (v\sigma_y\sigma_k)^2 + (\sigma_x\sigma_y)^2}\right\},$$

where $\sigma_x, \sigma_y, \sigma_k$ – standard Gabor filter derivatives.

Speed vector $(u,v)^T$ is found during minimization of function

$$f(u,v) = \sum_{i=1}^{12} m_i - \overline{m}_i \frac{R_i(u,v)}{\overline{R}_i(u,v)}$$

with respect to u and v, where m_i – measured power value, R_i – predicted power value, $\overline{m}_i$ and $\overline{R}_i$ are average power values.

3 REGRESSION PSEUDOSPECTRUMS

In this section we introduce the notion of multiple-regression pseudospectrums.

Let again $I(k)$ be an input image pixel matrix with width w and height h on frame number k, $I(k) \in \mathbb{R}^{w\times h}$. It is assumed that $I(k)$ is a grayscale image, so $0 \le I(k)_{x,y} \le 255 \quad \forall x = 1\ldots w, \quad y = 1\ldots h$. Let us call $M_n(k)$ an regression accumulator of n frames with parameter α, calculated on frame k. It will be a matrix $M_n(k) \in \mathbb{R}^{w\times h}$(Box et al., 1994):

$$M_n(k+1) = \alpha M_n(k) + (1-\alpha)I(k). \tag{1}$$

You can calculate the accumulator value $M_n(k)$ on frame k by adding each older member in series (1):

$$M_n(k) = (1-\alpha)\sum_{i=0}^{k-1} \alpha^{k-1-i} I(i). \tag{2}$$

Let us assume that $l(k)$ is an element of the image matrix $I(k)$ and $m_n(k)$ is an element of the accumulator matrix $M_n(k)$ with the same coordinates, as $l(k)$. Let us suppose that on an initially zero input of accumulator (2) since some moment k_0 (without loss of generality, let $k_0 = 0$), during enough long time some signal with intensity l is being given:

$$m_n(k) = l(1-\alpha)\sum_{i=0}^{k-1} \alpha^{k-1-i} = l(1-\alpha^k). \tag{3}$$

Now it's quite simple to find such α, so that $m_n(k)$ would surely exceed β share of signal l after n frames:

$$m_n(n) = l(1-\alpha^n) = \beta l.$$

Hence

$$\alpha_n = \sqrt[n]{1-\beta}. \tag{4}$$

Thus, α_n is such time averaging parameter, at which the accumulator sum will be equal to $m_n(n) = \beta l$ through n frames. At the same time n here can be called β memory length or, simply, length of accumulator memory with the corresponding averaging parameter $\alpha_n = \sqrt[n]{1-\beta}$.

Given α_n can be found as (4), the whole accumulator sum in one pixel at variable frame k can be found as

$$m_n(k) = l(1-\alpha_n^k) = l\left(1-(1-\beta)^{k/n}\right). \tag{5}$$

The $m_n(k)$ graphs for different $\alpha_n, n = 4, 8, 16, 32$ values are shown in Figure 1, supposed $\beta = 0.5, l = 100, k_0 = 10$.

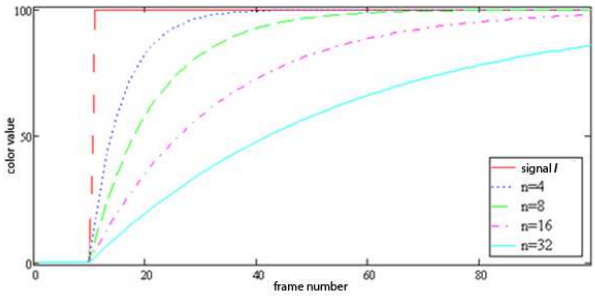

Figure 1: Accumulated pixel $m_n(k)$ for different α_n.

Thus, time averaging parameter α_n, defined in (4), is in fact the satiety parameter of the filter response function. It allows to judge after which time (in frames) n accumulated sum will be equal to βl.

According to (5), α_n possesses the multiplicity property:

$$\alpha_n = \alpha_{n\cdot s}^s. \tag{6}$$

Indeed, $\alpha_{n\cdot s}^s = \left(\sqrt[n\cdot s]{1-\beta}\right)^s = \sqrt[n]{1-\beta} = \alpha_n$. Let us call

$$D_{n,s}(k) = m_n(k) - m_{n\cdot s}(k)$$

a difference between the responses of accumulators with multiple smoothing parameter n and $n \cdot s$. By (6) and assuming that some signal with intensity l is being given from time $k_0 = 0$, this difference will possess a very interesting property:

$$D_{n,s}(n\cdot s) = m_n(n\cdot s) - m_{n\cdot s}(n\cdot s) = \tag{7}$$

$$= l\left(1-(1-\beta)^{\frac{n\cdot s}{n}}\right) - l\left(1-(1-\beta)^{\frac{n\cdot s}{n\cdot s}}\right) =$$

$$= l(1-\beta)\left(1-(1-\beta)^{s-1}\right) = l(1-\beta)\sum_{i=1}^{s-1}\beta^i.$$

Consider the behaviour of derivatives $D_{n,s}(k)$ function. Let $s = 2$ and $\beta = 0.5$. Then, according to (7), difference between accumulator with memory of $2n$ frames and accumulator with memory of n frames will be equal to

$$D_{n,2}(2n) = 0.25l. \tag{8}$$

Figure 2 shows differences $D_{n,s}$ between accumulators with variable n and $s = 2, l = 100$.

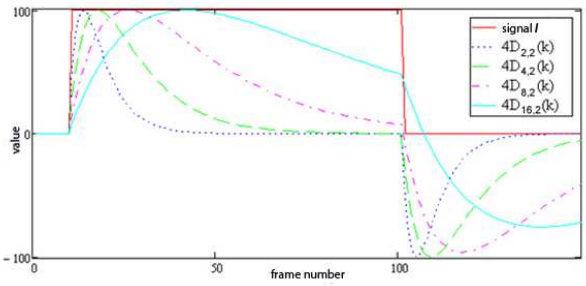

Figure 2: Pseudospectrum: accumulator derivatives $D_{n,s}(k)$.

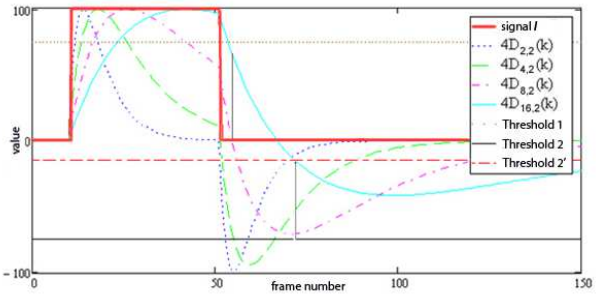

Figure 3: Dynamic brightness threshold correction based on pseudospectrum.

As you can see, quadruplicate difference of multiple accumulators $4 \cdot D_{n,2}(k)$ is a partially convex function on the segment of the signal presence. This maximum is single and equal to l, moreover, it is reached on the frame with number $2n$ (if this maximum can be reached at all).

Thus, first order regression derivatives behaviour with multiple memory length recalls spectral decomposition, or rather signal wavelet transformation. Let us call a multiple-regression pseudospectrum – set of differences of first-order regressive accumulators (7) with multiple characteristics of memory length by a sequence of powers of two: $1, 2, 4, 8, \ldots$ (view Figure 2). This pseudospectrum allows to qualitatively and quantitatively investigate both the duration and amplitude of the input time signal such as "meander."

If the maximum of differences between the responses has been consistently achieved for all accumulators with memory length N, but for accumulator with memory length $n = N+1$ predicted signal maximum was not reached, it means that a constant input signal had a length of $2N$ frames, and then began to decrease or was otherwise dramatically changed.

Similarly, we can make conclusions about the magnitude of the signal. Cause $D_{n,2}(2n) = 0.25l$, for all n whose maximum was reached,

$$l = 4D_{n,2}(2n). \tag{9}$$

Expected maximum value of $D_{n,2}(k)$ can be easily found, for example, for $n = 1$. Further it should be compared with the value of differences between accumulators $D_{n,2}(k)$ for other n until maximum on frame $k = 2m$ will be less than all previous maximums for $n < m$.

Now consider the problem of determining the sensitivity threshold of the algorithm, detecting the changes of brightness in images. Figure 3 shows the shape of multiple-regression pseudospectrum for the case of shorter time of signal presence on the image sequence.

Apparently, for lesser duration of the signal, lower frequency components of pseudospectrum start to move in the negative direction from higher initial values (after a reaction to the passage of the front edge of the signal) and thus achieve the appropriate extremum (in this case it will be minimum) at values lower in magnitude than the specified threshold, based on the expected drop estimate (8). Figure 3 illustrates it well by the function $D_{16,2}(k)$ (the lowest frequency component of the presented pseudospectrum). However, this problem can be solved if we jointly consider a pair of consecutive pseudospectrum components.

Consider previous $D_{8,2}(k)$ to $D_{16,2}(k)$ pseudospectrum component on Figure 3. Since its response to input signal change is much faster, it crosses the zero line much earlier, according to signal disappearance. At this point, the value of current $D_{16,2}(k)$ component still significantly greater than zero. This value (the value of the $D_{16,2}(k)$ pseudospectrum component when preceding component $D_{8,2}(k)$ crosses zero line) is proposed to memorize for each pixel and then to use in dynamic corrections to the threshold that detects brightness changes. As shown in Figure 3, detection of the back front of the signal with the threshold with dynamic correction is successful even in case of significantly short, compared with the characteristic time of accumulation of this pseudospectrum component, input signal.

Analysis of the introduced multiple-regression pseudospectrums is particularly useful in the case of image analysis that studies moving objects or left/missing items. Since, on the one hand, the object's motion relative to the background due to the effect of image pixels obstruction generates in each individual pixel temporal "meander" signal, which has clearly defined leading and trailing edges (brightness fluctuations over time). On the other hand, the possibility of signal analysis based on the difference between the accumulators with multiple memory lets you significantly decrease processing time of machine vision systems. Since estimates of the time signal characteristics must be obtained independently for each image pixel, in the case of using more complex statistics than the accumulated sums, the necessity to calculate the corresponding parameters estimates of the time signal directly leads to a huge increase of either computation time, or use of the program memory, or both.

4 ALGORITHMIC SCHEME

In this section we introduce the algorithmic scheme, which includes image preprocessing, motion detection and object tracking.

Objects detection and tracking are implemented as a modular three-stage procedure:

1. Detection of moving pixel groups based on pseudospectrum analysis.
2. Forming of object hypotheses and interframe object tracking.
3. Spatiotemporal filtration of object motion parameters.

Let us consider first and second stages of this procedure.

Detection of moving pixel groups is performed as follows:

- Calculate $D_{n,2}(k)$ pseudo-spectrums for various n, for example, $n = 2, 4, 8, 16$ in each pixel of the image on frame number k.
- If signal exits in some pixels, then $|D_{n,2}(k)|$ in them will be greater than zero. It can be or a signal from the object, or some noise on the image sequence. To make an algorithm more robust, we should filter the noise with some threshold. This threshold can be found adaptively on each frame using methods described above.
- Divide the whole accumulator image on many square parts using grid. Assume each small square as moving if its value is greater than threshold and not moving (background) otherwise. Let us call these small image squares moving image elements $\omega_1 \dots \omega_m$.

Moving object is created from moving image elements $\omega_1 \dots \omega_m$. Various moving elements exist for all values of n (or don't exist if there's no moving objects on video sequence on current frame). It's obvious that pseudospectrums with longer memory are more robust to noise, but it takes longer to react for them, when a signal in some pixels starts being received. Pseudospectrums with shorter memory react to a pixel signal much faster, but they react to noise as well as to a real signal. So if an element is a moving one, its signal should exist on most of faster pseudospectrums. And if it is a new or disappeared object, its signal should exist on most of slower pseudospectrums. Let us suppose that we have a set of moving objects $\Lambda_1 \dots \Lambda_{s1}$ and set of new or disappeared objects $\Delta_1 \dots \Delta_{s2}$ on a previous frame, set of moving image elements $\omega_1 \dots \omega_{m1}$ and elements that concern to new or disappeared objects $\omega_1 \dots \omega_{m2}$ on current frame. So we must somehow associate all objects with their new regions. Let us see hypotheses forming for moving objects:

- No object associates with the moving element. So this moving element belongs to a new object.
- No moving element associates with the object. This object is treated as lost on this frame. Maybe it will be found in future.
- Several moving elements are associated with the object. This object is treated as found on this frame. New position is calculated for it.
- Several objects are associated with one moving element. This case is called a "collision". It's the most difficult case, it should be treated very carefully. We have to use additional algorithms to parse this conflict.

As a result, on each frame we have a number of moving objects with their unique IDs and a number of new or disappeared objects with their unique IDs too.

5 EXPERIMENTAL RESULTS

Described algorithms were tested using the private video bases and public domain video bases like PETS (PETS video database, n.d.), ETISEO (ETISEO video database, n.d.). Typical screenshot of object tracking visualization is presented on Figure 4.

We created an algorithm analyzing and testing block that is based on comparison of automatic object detection and tracking results with results of manual object marking. Performance is measured in FPS (frames per second processed). Detection probability is estimated in terms of "precision" and "recall".

The "Precision" is a percentage ratio of real (human-marked) objects traced by the algorithm to all number of objects traced by algorithm. Simply put, 100% minus precision is a percentage of outliers provided by algorithm. The "Recall" equals is a percentage ratio of human-marked objects found by the algorithm to all number of human-marked objects in a sequence, i.e. 100% minus recall means percentage of real objects that were not found by the algorithm somehow.

The table 1 contains some video sequences from PETS and ETISEO databases and corresponding processing results. FPS was especially estimated for budget PC configuration: Intel Atom N270 1600 MHz processor and 1 Gb of RAM memory.

6 CONCLUSION

The problem of automatic video analysis for object detection and tracking is the most significant algorithmic topic in the digital video surveillance. The new motion analysis and object tracking technique is presented. Motion analysis algorithms are based on forming and processing of multiple-regression pseudospectrums. The object detection and tracking scheme contains: detection of moving pixel groups based on pseudospectrum analysis; forming of object hypotheses and interframe object tracking; spatiotemporal filtration of object motion parameters. Results of testing on public domain PETS and ETISEO video test beds are outlined.

REFERENCES

Anandan, P., 1989. A computational framework and an algorithm for the measurement of visual motion. Int. J. Comp. Vision 2, pp. 283–310.

Barron, J., Fleet, D. and Beauchemin, S., 1994. Performance of optical flow techniques. Internat. Jour. of Computer Vision 12(1), pp. 43–77.

Box, G., Jenkins, G. M. and Reinsel, G., 1994. Time series analysis: Forecasting and control (3rd edition).

ETISEO video database, n.d. http://www-sop.inria.fr/orion/ETISEO/.

Gabor, D., 1946. Theory of communication. Journal of the Institute of Electrical Engineers 93, pp. 429–457.

Heeger, D. J., 1988. Optical flow using spatiotemporal filters. Int. J. Comp. Vision 1, pp. 279–302.

Horn, B. K. P. and Schunck, B. G., 1981. Determining optical flow. Artificial Intelligence 17, pp. 185–203.

Nagel, H., 1983. Displacement vectors derived from second-order intensity variations in image sequences. CGIP 9, pp. 85–117.

PETS video database, n.d. http://www.cvg.rdg.ac.uk/slides/pets.html.

Singh, A., 1992. Optic flow computation. a unified perspective. IEEE Computer Society Press pp. 168–177.

Video name	Frame dimensions	Detection Type	Precision	Recall	FPS
PETS-2001-SEQ1-CAM1	768x576	Moving objects	80% (8/10)	100% (8/8)	110
PETS-2001-SEQ1-CAM2	768x576	Moving objects	88% (8/9)	100% (8/8)	109
PETS-2006-S1-T1-C3	720x576	Moving objects	74% (26/35)	85% (24/28)	115
ETISEO-VS2-BE-19-C2	768x576	Moving objects	100% (4/4)	100% (4/4)	110
ETISEO-VS1-AP-5-C5	720x576	Moving objects	88% (8/9)	100% (6/6)	124
ETISEO-VS1-AP-5-C7	720x576	Moving objects	100% (9/9)	100% (7/7)	124
ETISEO-VS2-BC-17-C1	640x480	New/diss. objects	66% (2/3)	100% (2/2)	142
ETISEO-VS1-BC-12-C1	640x480	New/diss. objects	100% (1/1)	100% (1/1)	144

Table 1: Video analysis algorithms testing results

Figure 4: Sample frame of PETS-2001-SEQ1-CAM1 video sequence. Objects, tracked on this frame: walking man (7), moving car (5) and parking car (3).