REAL TIME OBJECT TRACKING ALGORITHM

B. A. Zalesky, E. N. Seredin

United Institute of Informatics Problems of the NAS Belarus
Minsk, Belarus
E-mail: zalesky@newman.bas-net.by

A description of fast realization of the object tracking algorithm is presented. It is designed to track objects via a video camcorder mounted on an aircraft. The algorithm is based on a comparison of the correlation between the object image and regions of the current video frame together with analysis of inter-frame information. The video card programming technology CUDA has been used to provide real time computations on a standard PC. Also, the special robust version of the Kalman filter was developed to make the algorithm more stable. Tests showed applicability of the algorithm for solving tasks of the object tracking.

Key words: object tracking, video camcorder, aircraft, correlation, CUDA.

Introduction

In recent years, object tracking algorithms and technologies have been widespread to solve various problems in society, such as protection of the environment, change detection, cartography, navigation etc.

The problem of object tracking is understood as follows. The object is selected on the current frame of video sequence made from the board of the aircraft by the *non-stabilized camcorder*. The task is to find automatically this object in subsequent frames in real or quasi real time. The formulated task is now among most popular problems of image processing. A huge number of articles on the subject contain variety different methods and algorithms designed to solve it. Many of them are successfully applied in practice. Different techniques have been exploited to solve the problem. Among them are histogram, key point, optical flow, correlation, active contours and other methods [1, 2, 3, 4, 5, 6]. Overall difficulty for all current approaches is the small available run-time, which is limited by milliseconds needed for one shot. Usually it is in the range between 25 and 40 milliseconds. Some known reliable and accurate algorithms cannot be executed in such a short time.

Until recently, it can be said about correlation algorithms, which are ones of the most reliable and accurate but rather time consuming. A direct calculation of position of the tracked object in one video-frame of the standard size (without taking into account any prior information about its location) requires a few minutes of a modern PC processor. Even use of special fast computations such as the Fast Fourier Transform and SSE instructions has not made possible real time execution of correlation algorithms.

However, situation changed after appearance of the parallel computing platform CUDA and a new generation of video cards providing opportunities of working with 3D textures. Two these things allowed real time computations of correlation algorithms. In the paper results of such computations are presented together with comparison of the correlation approach to the task of object tracking with several other known approaches.

Moreover, the robust version of the Kalman filter has been included in the designed algorithm to increase stability of tracking.

In Section 2 brief description of the developed algorithmic approach is done. Section 3 contains the outlines of the designed program realization. Results of experiments are also placed in Section 3.

Brief description of tracking algorithm

After preliminary testing of several known approaches to the task of tracking objects in video sequences frames made by a non-stabilized camcorder it was decided to try real time realization of a few types of correlation algorithms.

To start with let us denote by $\mathbf{I} - m \times n$ gray scale either RGB image with pixels $p = (x, y)$ and brightness I_p or color $(I_{R,p}, I_{G,p}, I_{B,p})$. Frame of the video-sequence at time t is denoted by $\mathbf{I}_t$.

As usual, tracking process begins with selection of an object of interest. So, $\mathbf{O}_0(p_{obj})$ means sub-image of the initial frame $\mathbf{I}_0$ of the square form and size $k \times k$, $k < \min(m, n)$ with the center in pixel p_{obj}, which contains the object to track. Square $k \times k$ sub-images of the current frame $\mathbf{I}_t$ rotated by an angle α are denoted as $\mathbf{O}_t(p, \alpha)$, where p means the center of $\mathbf{O}_t$ with respect to $\mathbf{I}_t$. For simplicity, we do not consider here a scaling pyramid since according to our experience it is sufficient either not to use scaling at all or consider only two nearest levels of the pyramid.

Two types of correlation were used to estimate position of the tracked object. They are the standard non-normalized correlation (called in mathematical statistics covariance)

$$r_1(p,\alpha) = r_1(\mathbf{O}_0, \mathbf{O}_t(p,\alpha)) = \frac{1}{k^2}\sum\nolimits_{x,y=0}^{k} O_{0,x,y} O_{t,x,y}(p,\alpha) - \overline{O}_0 \overline{O}_t(p,\alpha),$$

where $\overline{O}_0$ and $\overline{O}_t(p,\alpha)$ are mean values of images $\mathbf{O}_0$ and $\mathbf{O}_t(p,\alpha)$, and the Pearson correlation

$$r_2(p,\alpha) = r_2(\mathbf{O}_0, \mathbf{O}_t(p,\alpha)) = \frac{r(p,\alpha)}{SSQ(\mathbf{O}_0)SSQ(\mathbf{O}_t(p,\alpha))},$$

where $SSQ(\mathbf{I})$ is the standard deviation of brightness (or color) of the image $\mathbf{I}$.

The estimators of the current position p_t of the tracked object at time t are

$$\hat{p}_{1,t} = \arg\max_{p,\alpha} r_1(p,\alpha) \text{ and } \hat{p}_{2,t} = \arg\max_{p,\alpha} r_2(p,\alpha).$$

Maximum is taken over all pixel of $\mathbf{I}$ and all allowable angles α. These estimators are well known not only in image processing, and therefore their use cannot be a surprise.

The novelty of the result is in the fast realization of the estimators that made possible computation of r_1 or r_2 in real time mode (details its fast program realization are placed in Section 3). It is also in exploiting a robust version of the linear Kalman filter, which makes tracking process more stable and accurate in comparison with application of the classic its version. Description of the Kalman filter can be found in many articles and books [7]. Very briefly, the principle of operation of the filter can be represented by two equations

$$\hat{x}_t^- = F_{k-1}\hat{x}_{t-1}^+ \qquad (1)$$

$$\hat{x}_t^+ = \hat{x}_t^- + K_t(y_t - H_t\hat{x}_t^-),$$

where $\hat{x}_t^-$ is called forecasting, $\hat{x}_t^+$ is the resulting estimate, and y_t is input data. Forms of matrices F_{k-1}, K_t and H_t is given in all details in [7]. The filter was exploited to stabilize the estimators $\hat{p}_{i,t}, i = 1, 2$. Therefore, $\hat{p}_{i,t}$ would have been used as input of the filter, i.e. for the classic version of the filter it should be $y_t = \hat{p}_{i,t}$. However, such choice generate significant error when the estimator $\hat{p}_{i,t}$ found false position of the tracked object in some frame $\mathbf{I}_t$. Thus, for some previously fixed square $S(\hat{x}_{t-1})$ of pixels with the center in pixel $\hat{x}_{t-1}$ we check whether $\hat{p}_{i,t}$ gets to $S(\hat{x}_{t-1})$ and set the input of the filter

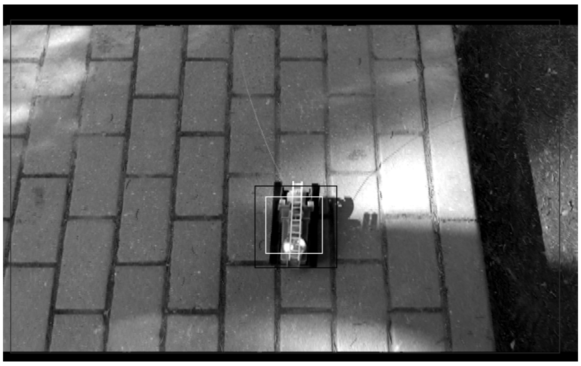
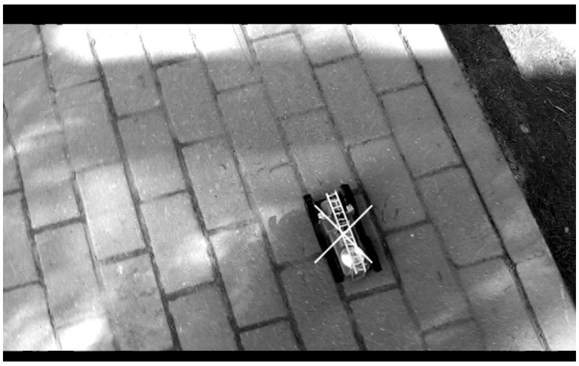
$$y_t = \begin{cases} \hat{p}_{i,t}, & if\ \hat{p}_{i,t} \in S(\hat{x}_{t-1}) \\ \hat{x}_t^-, & otherwise \end{cases}.$$

Besides that, for some previously fixes integer N_{out} we count how many successive estimates $\hat{p}_{i,t}$ turned out $S(\hat{x}_t^-)$. As soon as number these critical solutions becomes greater then N_{out} we start initialization of a new Kalman estimator in the following way. For some another previously fixed integer N_{in} we set $\hat{x}_t^+ = \hat{p}_{i,t}$ and check, when number successive estimates $\hat{p}_{i,t+k}$ turned in squares $S(\hat{x}_t^+)$ and only after this number becomes greater than N_{in} the real Kalman estimator is started.

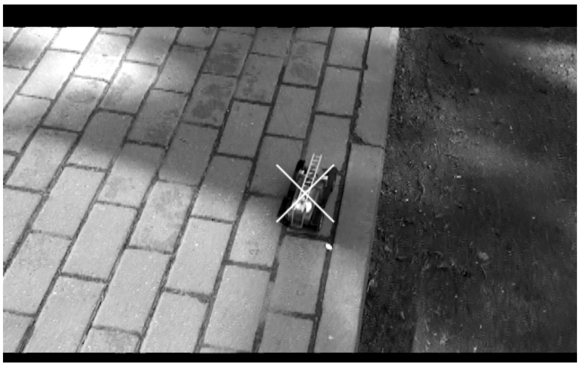
Detailes of program realization and experiments

Correlation type algorithms are one the most accurate tools to solve object tracking tasks. However, application of those algorithms till now has been limited due to the high computational cost required for their execution. For instance, finding the object of interest in a video sequence frame with help of direct realization of the Pearson correlation for PC CPU takes more than one minute. In order to overcome this difficulty NVIDIA's parallel computing architecture has been exploited. It enabled dramatic increases in computing performance by harnessing the power of the GPU.

Calculation of correlations has been optimized for the video card architecture Fermi that is now available in all last GPU of NVIDEA. To accelerate execution time all 6 types of the GPU memory were used together with the specially developed approach to parallel GPU computations. It enabled realization of the object tracking in real time mode for grayscale videos and almost real time computations for RGB videos of the standard size (that are 640×480 pixels). So, to find the object of interest in a gray scale frame the designed program realization needs less 30 millisecond.

a *b*




c *d*

Fig. 1. Examples of results of object tracking for videos made by the HDV video camera Sony HVR-HD1000E: *a* – manual object detection; *b* – *d* – object found by implemented algorithm

The developed approach to the formulated above object tracking task has been compared with tracking techniques, which are based on comparison of histograms of the oriented gradient and a few schemes of the texture analysis. The offered solution showed visibly more accurate results especially in the case of noisy video sequences. So, for instance, for a noisy video sequence made by the non-stabilized camcorder mounted on board the aircraft the object of interest was found in 97 % of frames by non-normalized correlation, and only 62 % by normalized one; simple two pixel texture analysis allowed founding the object in 74 % of frames; the mean square difference gave only 32 % of correct responses.

Application of Gauss, median and other filters have not given improvements because of non-stationary noise and distortions of video frames. Use of the corrections of brightness and contrast also has not improved reliability of the object tracking. Actually, the used standard preprocessing algorithms improved solutions for some frames but worsen for others.

In the future we plan investigation of fast recognition methods in order to increase probability of detection of frame background.

Conclution

A fast realization of the object tracking algorithm is presented. It allows real time tracking of objects observed by a non-stabilized camcorder installed on board the aircraft. Application of the Pearson and non-normalized correlations provided the most accurate tracking results in comparison with a few simple texture methods. Novelty of the offered realization consists in highly parallel computation of the correlation with help of the programming technology CUDA, which enable real time mode. Besides, the robust modification of the Kalman filter improved tracking stability. Experiments showed the correlation measure of image proximity together with robust modification of the Kalman filter provide reliable real time solution of the task.

Bibliography

1. *Yilmaz A., Javed O., Shah M.* Object tracking: A survey // ACM Computing Surveys. 2006. V. 38. № 4. Article №. 13.
2. *Marimon D., Ebrahimi T.* Orientation histogram-based matching for region tracking // Proc. 8th Int. Workshop on Image Analysis for Multimedia Interactive Services WIAMIS, Santorini, 2007. P. 8–12.
3. *Lowe D.* Object recognition from local scale invariant features // Proc. Int. Conf. on Computer Vision ICCV, Corfu, 1999. P. 1150–1157.
4. *Bay H., Tuytelaars T., Van Gool L.* Surf: Speeded up robust features // Proc. 9th Europ. Conf. on Computer Vision ECCV, Graz, 2006. P. 404–417.
5. *Altmann J., Reitböck H. J.* A Fast Correlation Method for Scale-and Translation-Invariant Pattern Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 1984. V. 6. № 1. P. 46–57.
6. *Mihaylova L., Brasnett P., Canagarajan N., Bull D.* Object Tracking by Particle Filtering Techniques in Video Sequences // Advances and Challenges in Multisensor Data and Information. NATO Security Through Science Series. Netherlands: IOS Press. 2007. P. 260–268.
7. *Simon D.* Optimal State Estimation. Kalman, H∞, and Nonlinear Approaches // New Jersey : John Wiley & Sons, Inc., Publication. 2006. P. 526.