

Rafał Jachowicz*, Piotr Duch*

Real Time Algorithm for Defined Simple Templates Detection and Tracking

1. Introduction

Objects detection and tracking algorithms are one of the main regions of research joined with computer image processing. Solutions based on image analysis are used with success in many fields of industry: production process oversight, computer tomography, surveillance cameras image analysis etc. As a matter of fact, there is a significant increase of usage of template detection algorithms in day-to-day life: road signs recognition implemented on board computers of modern cars or detection and analysis of plates on images acquired from traffic enforcement cameras. Irrespective of the way of applying all detection and tracking algorithms can be group to at least one of the three main categories: shape analysis [2], edge analysis [3] and color analysis [1].

In this paper a real-time algorithm for predefined template detection and tracking will be presented. The algorithm can be numbered to two of the previously mentioned categories because it analyzes the image in order to search for given shape and simultaneously controls color relations (values of analysed pixels of monochromatic image) on this image. That kind of approach assures high precision of detection and fair resistance to false-negative matching with little computing power.

2. Mathematical Theory

Proposed solution of objects detection and tracking on the sequence of images is based on the comparison of the image fragment and the given template. There are several methods of two image similarity coefficient computing. The simplest one is the sum of squared differences calculation of the values on both images [4]:

* Computer Engineering Department on Technical University of Lodz, Poland

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2 \quad (1)$$

where:

- I – input image,
- T – pattern,
- R – result image
- x, y – horizontal and vertical coordinates of the image (I),
- x', y' – horizontal and vertical coordinates of the template (T).

This method of computing differences between two images is relatively fast but its inaccuracy is unacceptable in that type of algorithms. After many experiments and tests of accuracy and speed dependency a different similarity coefficient determination method was chosen – Pearson's correlation coefficient [5] calculation:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2)$$

where:

- n – size of collections,
- x, y – elements in collection,
- \bar{x}, \bar{y} – arithmetical averages of elements in collections,
- r_{xy} – result Pearson's coefficient.

The equation above refers to one-dimensional collections correlation. Therefore in the proposed algorithm a modified two-dimensional version of this equation was used:

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}} \quad (3)$$

where:

$$T'(x', y') = T(x', y') - \frac{1}{w \cdot h} \sum_{x'', y''} T(x'', y'')$$

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{w \cdot h} \sum_{x'', y''} I(x + x'', y + y'')$$

- I – input image,
- T – pattern,
- R – result image,
- x, y – horizontal and vertical coordinates of the image (I),
- x', y' – horizontal and vertical coordinates of the template (T),
- x'', y'' – local horizontal and vertical coordinates of the analysed image/template fragment,
- w, h – width and height of the image or the template.

3. Proposed Algorithm

3.1. Preprocessing

Proposed algorithm can be divided into several main stages. In the first step preliminary processing of the input image is made. To significantly increase speed of execution the input image is scaled down to minimum acceptable resolution and the algorithm sets a region of interest where probability of template finding is the highest. Location of this region is evaluated according to positions of the patterns found on the previously analyzed images. That kind of approach saves a lot of precious CPU's computing power, however it takes effect in maximum detection distance diminish.

3.2. Template preparing and detection

In the next step a new pattern is prepared. It constitutes a fragment of the previous image where the previous template was found. If there were no detection yet or in the case there were no positive match for determined number of frames which means tracking is lost, the algorithm searches for generated template. The full range of template scales is scanned on the image. It is possible because the mathematical description and its size of the template is known a priori. Its dimensions are evaluated based on the calculated size of the previously found template (and if there were no detection yet based on determined calibration size) and its colors based on current brightness of the image.

In the second step a correlation coefficient between just prepared template and each fragment of the image that is the same size as the template is calculated and the best match is chosen.

3.3. Match confirmation

At the beginning of this step brightness of the image in the closest neighborhood of the pattern position is calculated. Then the area is thresholded using calculated brightness value

in order to separate desired shape from the background. Then exact dimensions of a template which hypothetically should be placed in the match location are evaluated. Assumption that there is a desired template on the searched area and proceeding with the image analysis procedure regardless to real occurrence of a template in the analyzed point is preponderant solution in presented algorithm. Next an artificial template is created based on just evaluated dimensions and colors and matched to a fragment of the current image on which template should be placed. Again Pearson's correlation coefficient is used, but this time more restrictive threshold is taken into account. If on the analyzed area desired template really occurs the evaluated dimensions and colors will be correct and value of the correlation coefficient will be exceptionally high. This stage of proposed algorithm allows to minimize a probability of false-negative errors occurrence that is failing to reject a match which is not a desired template.

3.4. Post-processing

If confirmation of template presence in analyzed point is successful, window parameters which will be used to limit the search area on the next frame will be evaluated. Size and location of mentioned window depends on the size of the found template and an information given by template movement from prediction module. This part of algorithm remembers locations of templates found on the previous images and analyzes them calculating movement tendency. There are many methods to evaluate movement tendency on image sequence, inter alia, based on partial derivative calculation or extrapolation of template positions. In the presented algorithm the extrapolation method was used which bases on the locations coordinates on the previous images with the most important pattern position on the last image. That kind of analysis makes algorithm to proceed faster, because only part of the image on which the probability of template presence is highest is analyzed.

In the case no template was found on some determined number of recently analyzed images, the algorithm resigns from usage of the narrowing window and starts to analyze whole image.

3.5. Block Diagram

Stages of the algorithm mentioned above and relations between them are presented in the block diagram (Fig. 1).

4. Results

Presented algorithm was implemented and tested on a PC with Intel(R) Core(TM) i5 CPU (2,67GHz) and NVIDIA GeForce GT 240 graphic card. In addition to make results comparable the mentioned implementation was moved into a PC with Intel(R) Atom Z530 CPU (1,6 GHz). In the Table 1 the times per frame of the algorithm execution in two different stages are presented.

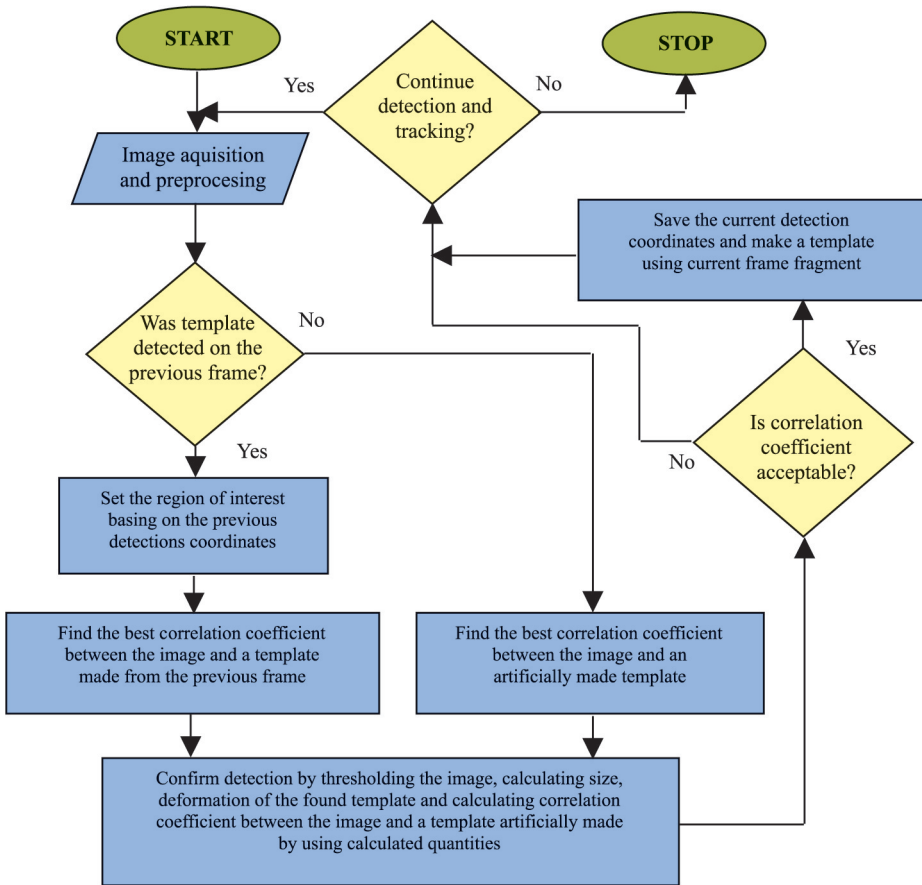


Fig. 1. Block diagram of the proposed algorithm

Table 1
Average times of single image processing

	Intel(R) Core(TM) i5 CPU average time [s]	Intel(R) Atom Z530 average time [s]
Whole image template detection	0.17	1.7
Found template tracking (limited area processing)	0.04	0.19

On the figures below (Fig. 2) the chosen frames taken from a sequence of images presenting detection and tracking of the example template are presented. For tests a black circle on white background was chosen as a template. A thin broken line drawn on each figure is a track of the template movement.

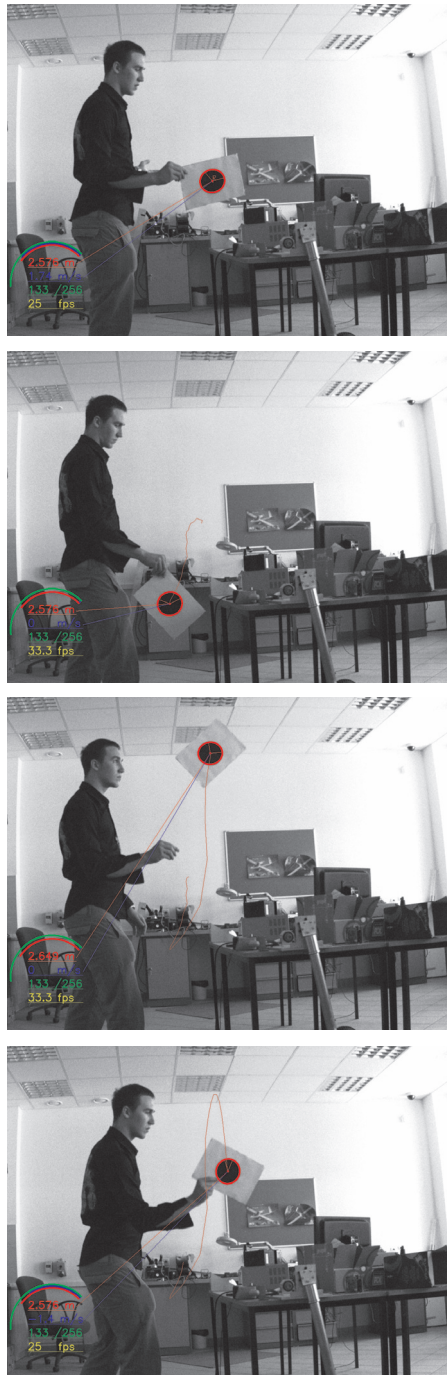


Fig. 2. Chosen frames of image sequence showing template in move

An advantage of the presented algorithm is fact that it tracks not only templates that are perpendicular to the camera axis (without deformations) but also templates that are deformed on the image by the spatial 3D rotation of almost any angle. It can be done because the algorithm has a mathematical description of the template and can calculate its obliquity if one is found. Detection and tracking of deformed by spatial 3D rotation templates is presented on the figures below (Fig. 3).

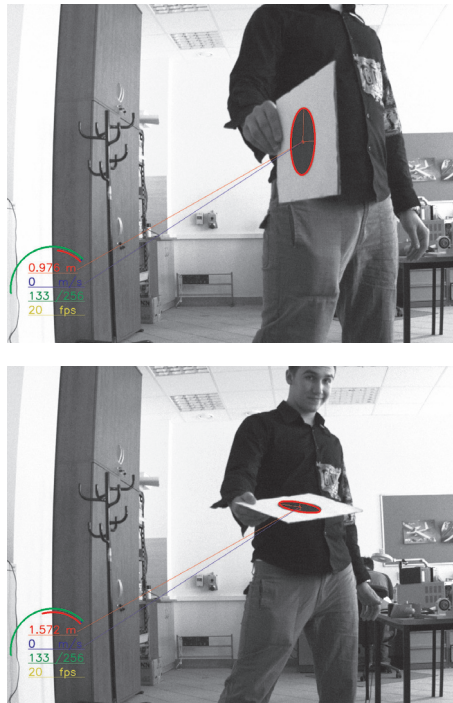


Fig. 3. Detection and tracking of the template deformed by 3D spatial rotation

5. Conclusion and future work

In this paper a robust method of detection and tracking of simple templates which can be easily mathematically described on the image sequence was presented. In comparison to other similar detection and tracking methods the improvements are: possibility to detect and track templates that are deformed by spatial 3D rotation and innovative technique of template presence confirmation which minimizes the probability of detecting objects that are not templates.

Performed experiments showed that it is possible to detect and track the template on the sequence of images nearly regardless to its location and rotation in space in the real

time. Future work and tests will be proceeded in two different directions: distance from camera to template calculation based on the template real size and detection and tracking of objects which are more difficult to describe mathematically.

References

- [1] Lopez Luis David, Fuentes Olac, *Color-Based Road Sign Detection and Tracking*. Lecture Notes in Computer Science, vol. 4633, 2007, 1138–1147.
- [2] Viola Paul, Jones M.J., *Rapid Object Detection using a Boosted Cascade of Simple Features*. Proc. IEEE Conf. On Computer Vision and Pattern Recognition, 2001.
- [3] Rizon Mohamed *et al.*, *Object Detection using Circular Hough Transform*. American Journal of Applied Sciences, 2 (12), 2005, 1606–1609.
- [4] Bendat J.S., Piersol A.G., *Random Data Analysis and Measurements Procedure*. John Wiley, New York 1971.
- [5] Rice A.J., *Mathematical Statistics and Data Analysis*. Duxbury Press India, 2006.