ICEST 2010 23 - 26 June 2010, Ohrid, Macedonia

A Fast Template Matching Algorithm

Yulka Petkova[1], Nuri Nuri[2] and Milena Karova[3]

Abstract – **Fast template localization in an image has ever been a great challenge to the researchers. The here presented algorithm uses a combination of different approaches for run-time acceleration including computations over extracted sets of representative points, fast elimination of unpromising positions and parallel computations. Experimental results prove the reliability and precision of localization and a significant run-time acceleration comparing with the classic brute-force algorithms for template matching.**

Keywords – **Template Matching, Template Localization, Image Processing, Representative Points, Parallel Computing**

I. Introduction

One of the most frequently solved tasks in the field of image processing is the task for template matching. It consists of finding approximately or fully coinsidence of the template T^M in the image I^N, i.e. coinsidence with any of sub-images I_i^M, as it is shown in fig. 1.

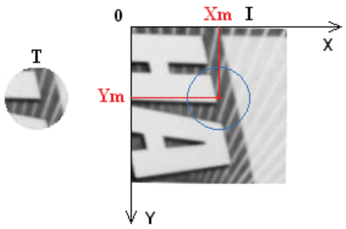

Fig. 1. Illustration of template matching problem

A wide variety of applications need the solution of the pointed problem. Some of them are:

- Fiducial recognition for PCB assembly by pick-and-place robot;
- Registration of alignment marks on printed material to be inspected;
- Robot vision guidance, locating objects on conveyor belts, pallets and trays;
- Template (model) based object recognition, etc.

The problem of template matching is connected with finding of any matching measure and algorithm to evaluate the extent of coincidence of the template with the correspondent part of the image. A great number of matching criteria and searching algorithms are published [4]. One of the basic purposes is the high reliability - the matching criterion has to cope successfully with any type of distortions – translation, rotation, scaling, nonlinear changes in the intensity, etc. The other important goal is the real time processing. The second goal is harder to achieve than the first one, because of the great data sets, which are to be processed.

Some of ways to speed-up the computations for template localization in a greater image are: making computations in the frequency domain [5]; searching around the most probable position [6]; reducing the number of sub-windows for making computations [4, 6]; skipping the unpromising positions [1]; reducing sets of points taking part in comparisons [4, 6]; using pyramids of data for image and template representation [2, 4], etc.

An alternative approach, to speed-up the computations for template localization, is to use effectively the hardware capacity of the modern computers, especially on systems with multicore central processing units (CPUs). This approach is based on scalable parallel computations which will guarantee effective usage of computer hardware.

In this paper we propose to combine the usage of different optimization techniques in order to increase the speed of template localization process in a greater image. We combine:

- usage of extracted sets of representative points;
- fast elimination of unpromising positions;
- usage of parallel computations.

As a result of this combination the run-time of template localization is highly accelerated. The precision and reliability of the localization are preserved.

II. Sets of representative points

One of the ways for speeding-up the algorithms is to use sets of representative points and compute the similarity measures taking intensities of only these points. A great variety of methods for representative point extraction are using in practice. They use different criteria for making selection. For the goals of our study, we have used three types of representative points:

- Points, selected by Sobel edge detector are the first type;
- The second type are selected, using the earlier proposed by us TyPe edge detector [7]. It extracts edge points, determined by the points of local extremes of the second derivative of the intensity function according to the following criteria:

$$\text{If} \quad \left| \underset{i}{extr}\left(\frac{d^2 I(x)}{dx^2} \right) \right| > \theta, \quad \text{then} \quad i \in E, \qquad (1)$$

where $I(x)$ is the intensity function in a row or in a column of the image, θ is a threshold. E is the set of extracted edge points.

[1]Yulka Petkova is with the Department of Computer Sciences and Technologies at Technical University of Varna, 1 Studentska Str., Varna 9010, Bulgaria, E-mail: jppet@abv.bg.

[2]Nuri Nuri is with the Department of Computer Sciences and Technologies at Technical University of Varna, 1 Studentska Str., Varna 9010, Bulgaria, E-mail: nuri.ismail@gmail.bg.

[3]Milena Karova is with the Department of Computer Sciences and Technologies at Technical University of Varna, 1 Studentska Str., Varna 9010, Bulgaria, E-mail: milena.karova@gmail.com.

- The third type of points are selected, using another earlier proposed by us method for Equipotential planes [8]. We interpret the image as a three-dimensional object and we propose the choice to be made by equipotential planes, which are parallel to the plane *xOy* and which cut the image (relief) on proper intensity levels. Thus, the extracted points outline the horizontal contours of the local “hollows” and “hills” from the three-dimensional image profile.

We define the following rule (Eq. 2) to extract representative points:

$$\text{If}\left(\begin{array}{l}\left(\begin{array}{l}\left[(PI(i,j)\le Pt_{max})\cap(PI(i+1,j)>Pt_{max})\right]\cup\\ \cup\left[(PI(i,j)\ge Pt_{max})\cap(PI(i+1,j)<Pt_{max})\right]\end{array}\right)\cup\\ \cup\left(\begin{array}{l}\left[(PI(i,j)\le Pt_{min})\cap(PI(i+1,j)>Pt_{min})\right]\cup\\ \cup\left[(PI(i,j)\ge Pt_{min})\cap(PI(i+1,j)<Pt_{min})\right]\end{array}\right)\cup\\ \cup\left(\begin{array}{l}\left[(PI(i,j)\le Pt_{max})\cap(PI(i,j+1)>Pt_{max})\right]\cup\\ \cup\left[(PI(i,j)\ge Pt_{max})\cap(PI(i,j+1)<Pt_{max})\right]\end{array}\right)\cup\\ \cup\left(\begin{array}{l}\left[(PI(i,j)\le Pt_{min})\cap(PI(i,j+1)>Pt_{min})\right]\cup\\ \cup\left[(PI(i,j)\ge Pt_{min})\cap(PI(i,j+1)<Pt_{min})\right]\end{array}\right)\end{array}\right)\text{Then}\quad (i,j)\in EP \qquad (2)$$

In this equation $PI(i,j)$ is the intensity at point (i,j); Pt_{min} and Pt_{max} are intensity thresholds - low and high respectively; *EP* is the extracted set of representative points.

We use the three-dimensional image profile only to extract coordinates of representative points. In the next computations (for template localization) these points take part with their intensities.

III. Similarity Measures

Similarity measures are also very important part of the localization acceleration process because their computational complexity reflects directly on the speed of localization. We focus on two kinds of similarity measures because of their high reliability and low computational complexity:

- Normalized Cross Correlation (NCC);
- M-estimators combined with “sum of robust differences” (SRD).

For the goals of the study, NCC has been computed over sets of representative points:

$$NCC(x,y)=\frac{\sum_{q=1}^{P} I(x+i_q,y+j_q).T(i_q,j_q)}{\sqrt{\sum_{q=1}^{P}\left(I\left(x+i_q,y+j_q\right)\right)^2}.\sqrt{\sum_{q=1}^{P}\left(T\left(i_q,j_q\right)\right)^2}} \qquad (5)$$

where $T\left(i_q,j_q\right)$ for $q = 1,2,\ldots,P$ is the intensity at q-th representative point of the template point with coordinates (i_q,j_q), $I\left(x+i_q,y+j_q\right)$ is the intensity at the corresponding point of the image.

A lot of M-estimators have been discussed in literature - Huber’s, Tukey’s, Geman’s and McClure’s, Lorentzian’s, etc. [1], [3]. We have investigated all of them, but the best results we have received using the first two. They also have been computed over sets of representative points.

In the following equations ρ is the robust error measure, r is the difference between intensities of compared points and σ is a threshold, which can vary from 0 to 255.

- Huber’s estimator:

$$\rho_1(r,\sigma)=\begin{cases}\dfrac{r^2}{2} & r\le\sigma\\[2mm] \sigma\left(r-\dfrac{\sigma}{2}\right) & r>\sigma\end{cases} \qquad (6)$$

- Tukey's estimator:

$$\rho_2(r,\sigma)=\begin{cases}\dfrac{\sigma^2}{6}\left[1-\left(1-\left(\dfrac{r}{\sigma}\right)^2\right)^3\right] & r\le\sigma\\[2mm] \dfrac{\sigma^2}{6} & r>\sigma\end{cases} \qquad (7)$$

The similarity measure is “sum of robust differences - SRD”, as it is in [1]:

$$SRD_{\rho,\sigma}(T,I)=\sum_{i=0}^{n}\sum_{j=0}^{n}\rho\left(r,\sigma\right) \qquad (8)$$

IV. Search Strategy

For additional acceleration of the localization we use a search strategy based on a regular set of initial positions of the template in the image and consecutive search around the most promising position [9]. This strategy belongs to the so called “coarse-to-fine” searching algorithms and guarantees a significant run-time acceleration comparing with the classic brute-force algorithms.

V. Parallel Computing

In recent years parallel computing has become a dominant paradigm in computer architecture, mainly in form of multicore processors. In order to use the advantages of these processors we have implemented parallel versions of the localization algorithms, using a cross platform multithreading library designed and implemented by us. This library is designed as general purpose library and is used in template localization algorithms at different levels:

- for parallelization of single template localization (parallel localization);
- for parallelization of multiple localizations (parallel experiments).

The strategy used for parallelization of single localization is shown in Fig.2.

The first step is to detect the number of processors (cores) for the current system on which the program is running. If this number is N, then we divide the image in N sectors and create a worker thread for each sector. In the parallel part of the algorithm each worker thread executes computations for template matching for its own sector and finds the best match. After that these results are analyzed and compared in order to find the best candidate for template position in the image.

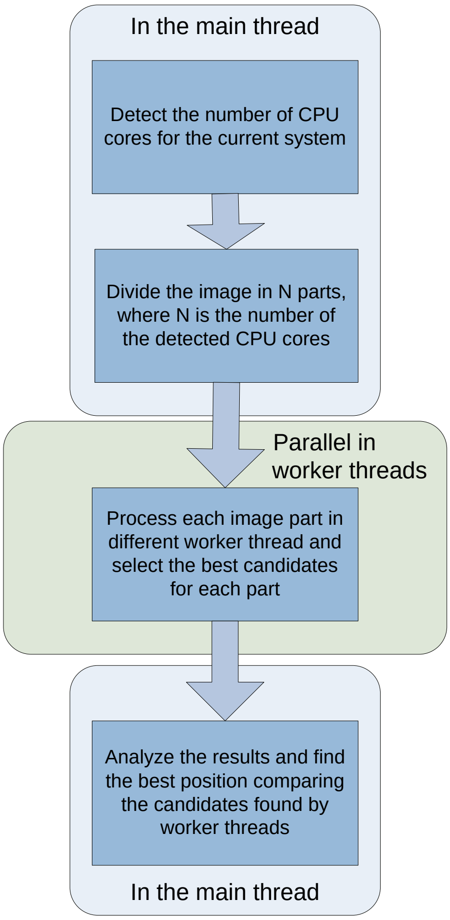

Fig.2. Parallel single template localization algorithm

The strategy used for parallelization of multiple localizations on same image or on different images is very similar to the previous presented parallel single template localization algorithm. The most significant difference is that for multiple localizations we group pending localization tasks in order to distribute them uniformly on worker threads. The number of worker threads is equal to the number of processors (cores) in the system and each worker thread executes one or more localization tasks. Each thread keeps the results for further analysis and registration. The illustration of this strategy is shown in Fig.3.

We successfully combine both of these parallelization strategies with the algorithms presented in previous sections and get significant run-time acceleration.

VI. Experimental Results

Series of experiments have been made to estimate proposed approach for fast template localization. These experiments include using different sets of representative points, different template shapes and dimensions, different searching strategies and parallel computing techniques as well. All the series include 100 experiments. A short summary of these experiments is shown in the next tables.

- **Localization reliability in translated and noised images**

The experiments are provided for different sets of representative points, using fast elimination of unpromising positions and parallel computing techniques, presented in previous sections.

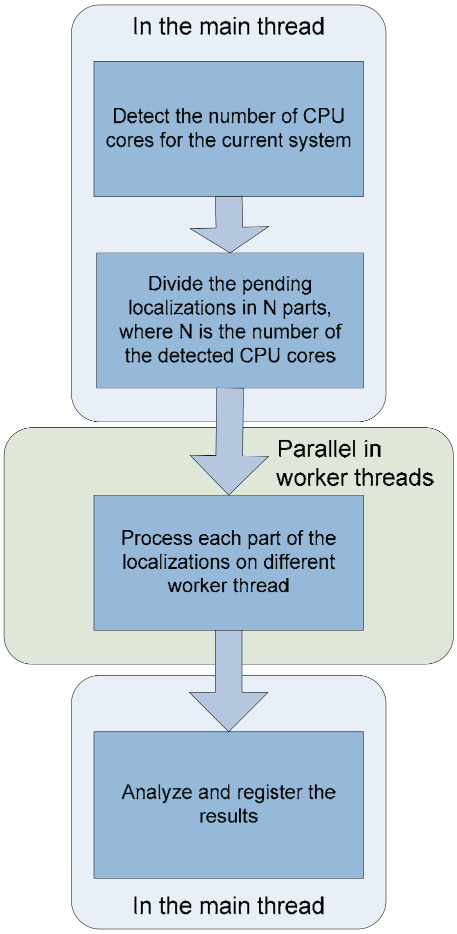

Fig.3. Parallel algorithm for multiple localizations

The results in Table I present the successive template localizations in the image in presence of noise. Table II is about the precision of template matching in presence of noise and present mean square error values for different noise levels. Both of these experiments have been designed and implemented on same conditions.

Experimental results confirm the high reliability of the template localization using the proposed approach of combining the different acceleration techniques with parallel computing.

Table I
Reliability in the presence of noise [%]

Noise [%]	5	10	30	50	70
M-estimators with Equipotential Points	100	100	100	99	99
M-estimators with TyPe edge points	100	100	100	100	99
M-estimators with Sobel edge points	100	100	100	99	98
NCC over all points	100	100	100	100	100

Table II

TEMPLATE MATCHING IN THE PRESENCE OF NOISE - PRECISION

Mean Square Error [$*10^{-6}$]

Noise [%]	5	10	30	50	70
M-estimators with Equipotential Points	0,02	0,02	0,02	0,03	0,05
M-estimators with TyPe edge points	0,02	0,02	0,02	0,04	0,06
M-estimators with Sobel edge points	0,02	0,02	0,02	0,04	0,06
NCC over all points	0,006	0,01	0,01	0,02	0,02

- **Run-time optimization**

All of the presented experiments in this section have been made on the following PC configuration:

- OS: Microsoft Windows XP Professional SP2;
- CPU: Intel Core 2 Quad Q6600, 2400 MHz (9x267);
- System Memory: 2 x 1 GB DDR2-667 MHz;
- VGA: Intel(R) Q33 Express Chipset Family (128 MB);

The results in Table III present the run-time comparison of classical template localization technique and the combinative parallel acceleration approach presented in this paper.

Table III

RUN-TIME COMPARISON

		Searching algorithm	**brute-force**	**coarse-to-fine**
Total duration [ms]	**Sequential**	**M-Estimators**	10078	844
		Correlation	11060	1078
	Parallel	**M-Estimators**	2880	275
		Correlation	3210	305

The processor of the test system is multicore (4 cores) and the usage of parallel algorithm results in great performance improvement (at about 350%) for both classical brute-force and accelerated coarse-to-fine search strategies.

The next images show the CPU usage for single-threaded and for multi-threaded parallel algorithm on the test system.

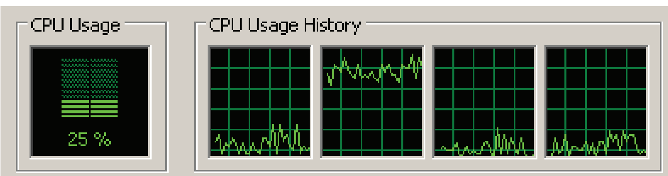

Fig.4. CPU usage for sequential single-threaded algorithm

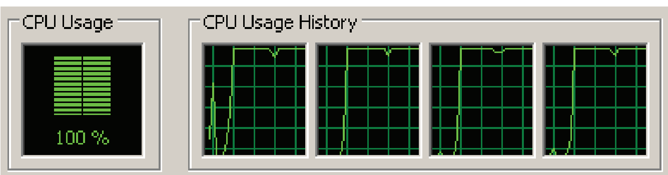

Fig.5. CPU usage for parallel multi-threaded algorithm

It is seen in the images that the parallel computing techniques guarantee effective usage of computer hardware.

VII. CONCLUSION

To conclude we can point out the advantages of our approach:

- It uses a combination of different strategies in order to decrease the computational complexity of template matching.
- It uses parallel computing techniques for effective usage of computer hardware.
- Decreasing the computational complexity and using parallel versions of the algorithms does not lead to significant reliability and precision decrease.

We can generalize that the received results are good enough, and we can use the proposed method to solve tasks for template matching in applications for which the received precision is satisfactory.

ACKNOLEDGEMENTS

The work presented in this paper was partially supported within the project BG 051PO001-3.3.04/13 of the HR Development OP of the European Social Fund 2007-2013.

REFERENCES

[1] J-H. Chen, C-S. Chen, Y-S. Chen, “Fast Algorithm for Robust Template Matching with M-estimators”, IEEE Transactions on Signal Processing, Vol. 51, Issue 1, Jan. 2003, pp. 230 - 243

[2] R. Dahyot, P. Charbonnier, F. Heitz, “Robust Visual Recognition of Colour Images“, CVPR-2000 International conference on Computer Vision and Pattern Recognition - pp. 685-690 - Hilton Head Island, USA

[3] K.V. Arya, P. Gupta, P.K. Kalra, P. Mitra, “Image registration using robust M-estimators”, Pattern Recognition Letters 28, pp. 1957–1968, 2007

[4] B. Zitova, J. Flusser, “Image Registration Methods: A Survey”, Image and Vision Computing, 21, pp. 977 – 1000, 2003

[5] S. Yoshimura, T. Kanade, “Fast Template Matching Based on the Normalized Correlation by Using Multiresolution Eigenimages”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No 12, pp. 2086-2093, 1993

[6] Lee Chang-Hsing, Ling-Hwei Chen, “A Fast Motion Estimation Algorithm Based on the Block Sum Pyramid”, IEEE Transactions on Image Processing, Vol. 6, No11, pp. 1587-1591, 1997

[7] D. S. Tyanev, Y. P. Petkova, “About the Possibilities of a New Edge Definition”, 28th ISSE 2005, 19th – 22nd May 2005, Wiener Neustadt, Austria, IEEE Catalog Number 05EX1142C, ISBN: 0-7803-9325-2, pp. 273 - 378

[8] D. Tyanev, Y. P. Petkova, “A New Method for Important Points Extraction”, International Conference on Computer Systems and Technologies CompSysTech’2004, 17th – 18th June 2004, Rousse, Bulgaria, ISBN 954-9641-384, pp. IIIA.11-1 – IIIA11-8

[9] Y. Petkova, N. Nuri, “A Strategy for Fast Template Localization in an Image”, Fourth International Conference “Computer Science” - 2008, Kavala, Greece, Sep. 2008, pp. 675 - 680.