JACET Journal of Advances in Computer Engineering and Technology, 1(1) 2015

A New RSTB Invariant Image Template Matching Based on Log-Spectrum and Modified ICA

Mehran Yazdi[1], Narjes Pourjafarian[2], Mehrnaz Fani[3] and Elahe Taherianfard[4]

Received (22, 8. 2011)
Accepted (17, 12. 2011)

***Abstract* — Template matching is a widely used technique in many of image processing and machine vision applications. In this paper we propose a new as well as a fast and reliable template matching algorithm which is invariant to Rotation, Scale, Translation and Brightness (RSTB) changes. For this purpose, we adopt the idea of ring projection transform (RPT) of image. In the proposed algorithm, two novel suggestions are offered that significantly increase the precision and performance of the previous methods. First, our algorithm works with Log-Spectrum of image instead of the image itself, this change increases the accuracy of matching, and secondly for boosting the speed of the searching strategy, a new and modified version of Imperialist Competitive Algorithm, MICA, is presented. This matching procedure avoids the searching algorithm from being trapped in local minimum by taking advantage of adding a modification step to ICA. The simulation results show the superiority of proposed method in comparison with the previous ones.**

***Index Terms* - RSTB invariant template matching; Logarithmic Spectrum; Ring Projection Transform (RPT); Imperialist Competitive Algorithm (ICA)**

1- Mehran Yazdi, Dept. of Communications and Electronics, School of Electrical Engineering, Shiraz University, Shiraz, Iran (yazdi@shirazu.ac.ir).
2- Narjes Pourjafarian, School of Electrical Engineering, Shiraz University, Shiraz, Iran (n.pourjafarian@gmail.com)
3- Mehran Fani, Dept. of Communications and Electronics, School of Electrical Engineering, Shiraz University, Shiraz, Iran (fani.mehrnaz@gmail.com)
4- Elahe Taherianfard, School of Electrical Engineering, Shiraz University, Shiraz, Iran (etaherianfard@yahoo.com)

I. INTRODUCTION

Template matching is a technique for finding small parts of an image, which match a query template image. It is one of the most important challenges in object recognition, stereo matching, image registration, feature tracking, remote sensing and computer vision.

There are two main challenges in template matching, the first one is choosing a proper similarity measurement and the other one is finding an efficient searching strategy.

Normalized Cross Correlation (NCC) is widely used as a standard method of template matching, but it has some pitfalls, like being sensitive to rotation and scaling. It's also computationally expensive, because in this technique all the possible blocks of image must be tested for finding the best match.

In more recent works some template matching approaches are suggested that are invariant to Rotation, Scaling, Translation and Brightness (RSTB) and use faster searching strategies. For instance in [1] the authors have used tree cascade filters(Circular, Radial and template matching filters) which provide a RSTB invariant template matching technique that refines the possible matching samples in each step to finally locate the best match. They also have worked on template matching with illumination changes in the case of color images in [2]. In fact in their latter work, the authors have used the same procedure as in the former, except for introducing a new similarity measure that uses the CIELAB color space. This similarity measure is defined as the weighted geometric mean of two other similarity criteria (the Euclidean distance of chromatic components, and the correlation of the lightness components) [2].

In order to speed up the searching step, in the template matching, a coarse to fine procedure has been used in [3]. In this method, the local entropy differences between the template image and the sub-images are calculated in the coarse stage. Then a candidates-table is built by using a threshold on the entropy differences. Afterwards, in the fine searching stage, a new similarity measure is introduced to find the best match among the candidates.

In another work done by Duan et al. [4], NCC has been used as the similarity measurement, and in order to accelerate the template locating process, chaotic Imperialist Competitive Algorithm (ICA) has been used as a searching strategy. This method is fast but not invariant to possible changes of template image like rotation and scaling.

In this work we propose a novel, fast and precise template matching algorithm which is invariant to transformation, rotation, scaling and brightness changes. In the suggested algorithm we compute the logarithmic spectrum of a set of differently scaled versions of the template image, then the ring projection transform (RPT) [5], is applied to results. In this way the scaling and rotation invariability is obtained. Afterwards, we modify the ICA method [6] to be used as the searching strategy for locating the best match.

The paper is organized as follows, in section II we explain our proposed method for achieving a simulation measurement which is invariant to rotation, scaling and brightness, we also briefly review the RPT method that is used as a part of this algorithm. In section III a suggested searching strategy that is based on ICA is described. In section IV the combination of searching strategy with the proposed template matching algorithm is explained. In section V the simulation results of proposed method with comparison to other algorithms are given. Finally, in section VI we draw final conclusions.

II. PROPOSED METHOD FOR FINDING AN RSTB INVARIANT SIMILARITY MEASURMENT

In the first stage of template matching, we do the histogram equalize and then we convert the query image and each test-block of the source image, to a set of vectors that are invariant to rotation, scaling and brightness. The required steps for reaching this purpose are as follow.

1. Histogram Equalization

To make our matching method invariant to brightness changes, as the first step we equalize the histogram of the template image and any sub-image of the source image that is intended to be checked for finding the best match.

The resulted images have uniform histograms. It means that the gray levels in the images are uniformly distributed.

Hence, different versions of an image with different brightness levels will almost be the same, after passing a histogram equalization block. This has been shown in Fig.1.

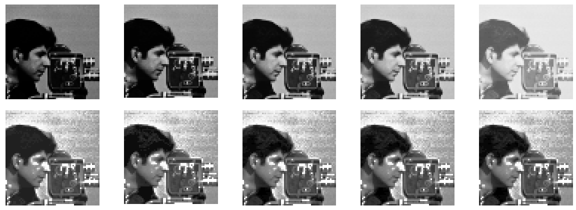


Figure1. The cameraman image with different brightness levels, in the first row, and the histogram equalized versions of each image, in the second row.

In this figure, the images of the first row have different brightness levels while the histogram equalized versions of them, shown in the second row, are almost the same.

1.Computing the Logarithmic Spectrum of an Image

Consider that we have a template image t(x,y) of size , as it is shown in Fig. 2(a). At First the Tukey window is applied to the image to make it circularly symmetric [7], like what can be seen in Fig. 2(b). The formula of Tukey window which is a cosine-tapered window is given in (1).

$$w(k)=\begin{cases}1 & 0\le |k|\le r_f\,\dfrac{\mathbf{K}}{2}\\[2mm] \dfrac{1}{2}\left(1+\cos\left(\pi\dfrac{k-r_f\frac{\mathbf{K}}{2}}{2(1-r_f)\frac{\mathbf{K}}{2}}\right)\right) & r_f\,\dfrac{\mathbf{K}}{2}\le |k|\le \dfrac{\mathbf{K}}{2}\end{cases} \qquad (1)$$

In which, is the ratio between the fixed and varying parts of window and K+1 is the length of window. Afterwards by using the FFT, the windowed image, is transferred to the Fourier domain to get into T(u,v), then the logarithm of the magnitude of T(u,v) is computed and finally normalized. The resulted, which is called the normalized log-spectrum of image, is depicted in Fig. 2(c) and the whole process is formulated as follows.

Figure 2. a) Template image. b) Template image after applying the Tukey window. c) Normalized Log-Spectrum of image.

$$t_w(x,y) = t(x,y)[w(x)w(y)] \quad x \in [1,M] \,\&\, y \in [1,N] \tag{2}$$

$$T(u,v) = f\{t_w(x,y)\} \tag{3}$$

$$S(u,v) = Log(1 + |T(u,v)|) \tag{4}$$

$$S_{norm}(u,v) = \frac{S(u,v)}{\max(S(u,v))} \tag{5}$$

In the next step, we apply the ring projection transform on this image in order to convert it to a 1D vector which becomes invariant to rotation of the image.

2. The Ring Projection Transform (RPT) Process

This stage of algorithm is the adaptation of the method, first introduced by Lin et al. [5]. In fact we reformulate all procedure for working with log-spectrum of image instead of the image itself. So at first $S_{norm}(u,v)$ is transformed into a polar frame according to the following relations:

$$u = r\cos\theta, \quad v = r\sin\theta \tag{6}$$

Where: $r = \text{int}(\sqrt{(u-u_c)^2 + (v-v_c)^2})$

$$r \in [0 \quad R], R = \min(M,N), \theta \in [0 \quad 2\pi]$$

and (u_c, v_c) is the central point of $S_{norm}(u,v)$.

Next, the RPT of $S_{norm}(u,v)$ at radius r is computed by:

$$RPT(r) = \frac{1}{C_r}\sum_k S_{norm}(r\cos\theta_k, r\sin\theta_k) \tag{7}$$

In which C_r is the number of pixels along the circle with radius r and center of (u_c, v_c). For better understanding the RPT, implementation of this projection on log-spectrum of an image is depicted in Fig. 3 where the mean value of pixels on each radius must be computed.

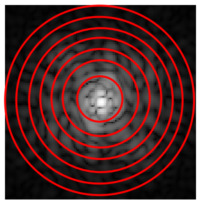


Figure 3. RPT on the Log-Spectrum of an image

Therefore, we can form a vector using all rings of an image as it is presented in (8). This 1D ring-projection is invariant to the rotation of its corresponding 2D image.

$$RPT = [RPT(0), RPT(1), \ldots\ldots RPT(R)] \tag{8}$$

3. Normalized Cross Correlation as Similarity Measurment

To have a similarity criterion which is invariant to scaling, rotation and brightness, a procedure based on [1] is used. For this purpose we build a set of differently scaled versions of template image $(t_1, t_2, \ldots\ldots, t_n)$, and repeat sections 1, 2 and 3 for each scale, to get into set of RPT vectors:

$$\begin{aligned} RPT_{t_1} &= [RPT_{t_1}(0), RPT_{t_1}(1), \ldots\ldots RPT_{t_1}(R)] \\ RPT_{t_2} &= [RPT_{t_2}(0), RPT_{t_2}(1), \ldots\ldots RPT_{t_2}(R)] \\ &\vdots \\ RPT_{t_n} &= [RPT_{t_n}(0), RPT_{t_n}(1), \ldots\ldots RPT_{t_n}(R)] \end{aligned} \tag{9}$$

We also compute the RPT for the test block of source image or the sub-image:

$$RPT_s = [RPT_s(0), RPT_s(1), \ldots\ldots RPT_s(R)] \tag{10}$$

For finding the best match, Normalized Cross Correlation (NCC) between the ring- projection of subimage (RPT_s) and the ring-projection of different scales of template image (RPT_t) is used. The maximum of these correlations is considered as similarity value. Formulas of NCC and similarity value are respectively given in (11), (12).

$$NCC = \frac{\sum_{r=0}^{R}(RPT_s - \overline{RPT_s})(RPT_t - \overline{RPT_t})}{\left[\sum_{r=0}^{R}(RPT_s - \overline{RPT_s})\right]^2 \left[\sum_{r=0}^{R}(RPT_t - \overline{RPT_t})\right]^2} \tag{11}$$

$$Sim\_val = \underset{i=1}{\overset{n}{Max}}\{NCC(RPT_s, RPT_{t_i})\} \tag{12}$$

Where in (11), $\overline{RPT_s}$ and $\overline{RPT_t}$, represent means of their corresponding vectors (RPT_s, RPT_t) and in (12), n is the considered numbers of image scales.

Till now we have obtained a proper similarity measurement and now we want to offer an efficient searching strategy, which can find the best subimage that matches the query image without being forced to check all the possible blocks of the source image. To reach this purpose we modify the Imperialistic Competitive Algorithm (ICA) and use it as our searching strategy. In the next section ICA and its modification are explained.

III. IMPERIALIST COMPETITIVE ALGORITHM

1. Original ICA

Imperialist Competitive Algorithm (ICA) [6], proposed by Atashpaz and Lucas (2007), is a novel global evolutionary optimization algorithm which uses the socio-political competition process as a source of inspiration. The ability of converging to global optimum solutions in a reasonable time with appropriate precision has caused it to be used in different kinds of optimization problems.

Like other population-based optimization algorithm, ICA starts with a randomly generated initial population which is called countries. Countries with the best cost are selected to be the imperialist states and the rest, form colonies of these imperialists. Based on the power of imperialists, all the colonies are divided among them. So each empire will consist of an imperialist and its colonies. This division is shown in Fig. 4.

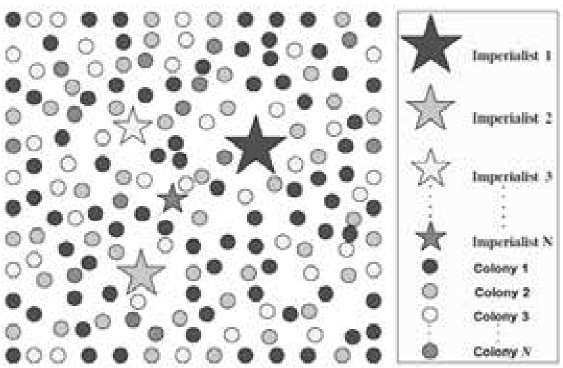


Figure 4. Initial empires [6]

When all empires were created, imperialist states start to assimilate their colonies. So, each colony moves toward its relevant imperialist state. Fig.5 shows the new position of each colony. In this movement, l is a random variable with uniform (or any proper) distribution and it is generated as follows:

$$l \sim U(0, \beta.d) \qquad (13)$$

In which β is a number greater than 1 and d is the distance between a colony and an imperialist. A random amount of deviation, θ, is applied to the direction of movement of each colony to extend searching area around the imperialists.

$$\theta \sim U(-\gamma, \gamma) \qquad (14)$$

Where θ is a variable with uniform (or any proper) distribution and γ is a parameter that modifies the deviation.

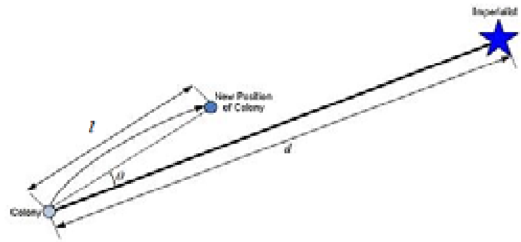


Figure5. Movement of colonies toward their relevant imperialist in a randomly deviated direction.

If after the assimilation policy, a colony reaches to a position with lower cost than its relevant imperialist, this colony and its imperialist will exchange their position. To start an imperialist competition between empires, total power of each empire is needed. The total power of an empire is defined as the power of imperialist country plus a percentage of the mean power of its colonies. Then Competition among empires will begin, the weakest empire loses one of its possessions (colony) and the others try to get it. The powerful Empires will have more chances to get this colony. During the procedure of the algorithm, empires which have lost all their colonies will be eliminated. This competition continues until the most powerful empire takes possessions of all other empires and there exists just one empire. In this situation, the position and power of all the colonies are the same as the remained imperialist and we stop the algorithm. The main steps in the algorithm are reviewed in the pseudo code shown in Fig.6.

1) *Generate an initial population randomly and calculate their objective function.*
2) *Select imperialist states and initialize empires.*
3) *Move the colonies toward their relevant imperialist.*
4) *If there is a colony with lower cost than its related imperialist, exchange their position.*
5) *Calculate the total cost of an empire based on the power of imperialist and its colonies.*
6) *Pick the weakest colony from the weakest empire and give it to the empire that has the most likelihood to possess it (Imperialistic competition).*
7) *If there is an empire without colony removes it.*
8) *If there is one empire left, stop algorithm, if not go to step3.*

Figure 6. Pseudo code of imperialist competitive algorithm

2. Modified ICA(MICA)

In bsic ICA, each empire uses assimilation policy to improve its colonies. This policy causes colonies to move toward their related imperialists. To diversify searching points around the imperialist, a random amount of deviation is added to the direction of these moving colonies as it is illustrated in Fig.5. But this deviation doesn't help to get rid of trapping in local optimum, in template matching. So, we use a small probability perturbation and a mutation operator to modify ICA.

Zhang, Wang and Peng in their paper [8] have suggested an improved ICA. According to their suggestion, in the assimilation policy when colonies moved toward their relevant imperialists, some colonies of each empire will be selected randomly. Then the new position of these selected colonies will be computed by adding a linear perturbation α to their present positions. α is a random variable with uniform distribution:

$$\alpha \sim \mathrm{U}(-\tau, \tau) \qquad (15)$$

Moreover, τ shows the search space of decision variables. To improve our modified algorithm, the combination of this method with a mutation operator is applied. When selected colonies obtained their new positions, the objective function is calculated. If these new positions have better costs than the old ones, they will be replaced; otherwise, we use mutation operator to determine next position of the current colony.

To generate a mutant colony, three different colonies from the empire, which contains the selected colony, are chosen randomly. Then the new position is calculated as follows:

$$colony = C = [c_1, c_2, ...c_{N_{var}}]$$

$$C_{mu\tan t} = C_1 + rand \times (C_2 - C_3) \qquad (16)$$

Where colony or C is a $1 \times N_{var}$ array, C1, C2 and C3 are three different colonies and C_{mutant} is the new position of the selected colony after mutation. rand is a random number between 0 and 1[9].

IV. APPLYING MODIFIED ICA TO RSTB INVARIANT TEMPLATE MATCHIN

In the following paragraphs we are going to give a systematic procedure for applying the MICA method to RSTB invariant template matching based on logarithmic spectrum. First, the histogram equalization is applied on template image and the test block of the source image to compensate the possible brightness changes in them. Then five differently scaled versions of the template image are produced $(s_1 = 0.5, s_2 = 0.7, s_3 = 1, s_4 = 1.3, s_5 = 1.5)$.

Next, the logarithmic spectrums of all templates and the test block are computed. To have a rotationally insensitive algorithm, the RPT process is applied on the achieved images. In this step, the selected number of radiuses does not have a significant effect on the final result, so we consider all the rings. To check the similarity between different versions of query image and the test block of the source image, Normalized Cross correlation is used. For avoiding the search of all pixels of the source image to find the best match, we apply Modified Imperialist Competitive Algorithm. In this algorithm, centers of randomly selected test blocks are considered as individuals of population. As our optimization algorithm (MICA) is based on minimizing the cost function, the objective function is defined as follows:

$$ObjectiveFunction = 1 - NCC \qquad (17)$$

The algorithm starts with creation of empires. The competition among empires improves them and the final result will be the center of the best-matched subimage.

V. EXPERIMENTAL RESULTS

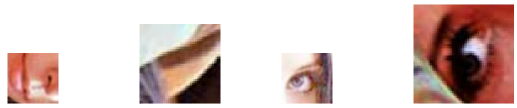
In order to evaluate the performance of the proposed algorithm in comparison with the other ones a set of template images with different rotation, scaling, translation and brightness variations have been used and three experiments have been conducted using these images. In the first experiment the superiority of the proposed similarity criterion (Log-Spectrum RPT) over RPT is checked, while using the same searching strategy (MICA) for matching in both cases, and the obtained results are given in tables II and III. In another experiment the simulation results for the method introduced in [5] is shown in Table IV. In this method, similarity criterion and matching strategy are both different from what we have proposed. In fact in [5] simple RPT along with NCC is applied for template matching. Therefore comparison of this table with Table II shows the efficiency and precision of our algorithm clearly. In the third experiment for checking the efficiency of the suggested searching strategy (MICA) in comparison with ICA, a fix similarity measurement (Log Spectrum RPT) is used in both simulations, and the results of this part are depicted in curve of Fig.9. Then in another experiment, some larger source images with more detail information are used to demonstrate the time efficiency and the precision of the proposed matching method in the more complicated cases.

The required parameters for implementation of proposed algorithm are given in Table I. The values of these parameters were achieved experimentally and by several runs of the algorithm to have a tradeoff between run-time of algorithm and the accuracy of results.

TABLE I
VALUES OF REQUIRED PARAMETERS.

M ICA		Log Spectrum RPT	
Parameter	**Value**	**Parameter**	**Value**
Number of Population(N_{pop})	100	Number of Scales	5
Number of imperialist(N_{imp})	8	Number of Rings	Optional(>5)
β	2	Taper to constant section of Tukey window	0.5
Number of iteration	50		
θ	$\pi/4$		

Fig.7 shows a series of templates, extracted from the original image in Fig.8. A rotated template with arbitrary angle is displayed in Fig.7 (a). Templates in Figs.7 (b) and 6(c) have scale and brightness changes, respectively. Finally the whole variations (RSTB) have been applied simultaneously on the template image in Fig.7 (d). As it is shown in Fig.8, the proposed algorithm has located all the template images precisely and it was invariant to the RSTB changes.



(a) (b) (c) (d)

Figure7. a) Rotation. b) Scaling. c) Brightness change. d) RSTB changes.

Figure8. Locating all template images correctly in source image.

In order to assess the performance of the suggested Log-Spectrum RPT, it is compared with simple RPT along with MICA approach and then with simple RPT along with simple NCC.

For this purpose we have found the location of the query image in the source image by running each of the above mentioned algorithms 100 times for each of the templates and the numbers of correct matching for Log-Spectrum RPT, RPT with MICA and without it are given respectively in Tables II, III and IV. Note that in these experiments the rotation, scale and brightness factors of the sample templates were arbitrary.

TABLE II
NUMBER OF CORRECT ANSWERS IN 100 RUNS FOR LOG SPECTRUM RPT WITH MICA.

MICA with logarithmic spectrum RPT	**Number of Correct Answers in 100 Runs**		
	$N_{pop=100}, N_{imp=8}$	$N_{pop=150}, N_{imp=10}$	$N_{pop=50}, N_{imp=6}$
Rotation	100	100	100
Scaling	100	100	100
Brightness	100	100	100
RSTB	100	100	100

TABLE III

NUMBER OF CORRECT ANSWERS IN 100 RUNS FOR RPT WITH MICA.

MICA with RPT	Number of Correct Answers in 100 Runs		
	$N_{pop=100}$, $N_{imp=8}$	$N_{pop=150}$, $N_{imp=10}$	$N_{pop=50}$, $N_{imp=6}$
Rotation	68	65	71
Scaling	64	60	52
Brightness	70	72	65
RSTB	53	57	49

TABLE IV
NUMBER OF CORRECT ANSWERS IN 100 RUNS FOR RPT.

Number of Correct Answers in 100 Runs			
Rotation	Scaling	Brightness	RSTB
62	54	47	42

From Tables II, III and IV it can be concluded that the suggested algorithm is more reliable and accurate than the previous simple RPT method. The Log Spectrum RPT is also not sensitive to the variation of most important parameters of MICA while the other method, RPT with MICA, is not invariant to these changes and cannot find the correct answer efficiently.

To compare the convergence speed and accuracy of proposed Modified ICA with original ICA, the convergence characteristics of these algorithms for one solution are illustrated in Fig.9.

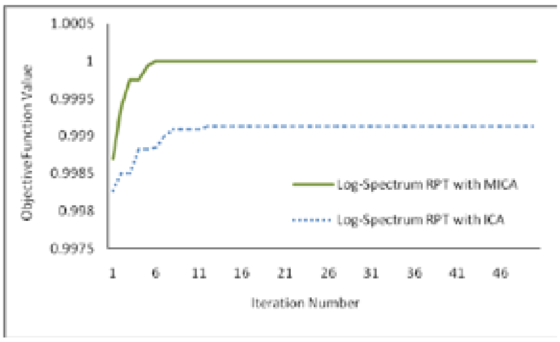


Figure9. Comparison between the performance of ICA and MICA.

As it can be seen in the above figure, our algorithm converges to the global solution faster, while the other algorithm has trapped in the local optimum.

From the computational point of view it can be said that our algorithm is more efficient in comparison with the ones that use simple cross correlation for finding the matching subimage. In fact our algorithm converges to the best match in less than 20 iterations for most of the cases. While in simple NCC approach all the blocks of the image must be checked for finding the best match. As an example we have compared our method to RPT along with the ICA searching strategy, and the simple RPT that checks all the possible blocks of the image in order to get into the best match. For this purpose we have used Matlab 7.8 on a Pentium IV computer with 3GB RAM and 2.26 GHz processor. In this experiment we have used two complicated source images with relatively large sizes, to be searched for finding the best matching block. The two source images along with their corresponding template images are shown in Figs.10 and 11.

(a)

(b)

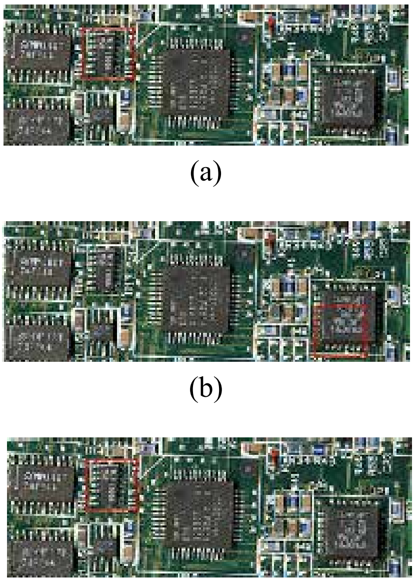
Figure10. a) Template image with size of 59x65 pixels. b) Source image (Board) with size of 306x648.

(a)

(b)

Figure 11. a) Template image with size of 114x108 pixels. b) Source image (Concord aerial) with size of 1018x1530.

The results of applying the proposed algorithm in comparison to two other methods are given in the following figures and tables.



(a)

(b)

(c)

Figure12. a) Result of applying the proposed method (Log-Spectrum RPT along with MICA). b) Result of applying RPT along with ICA. c) Result of applying RPT and searching all of the blocks.

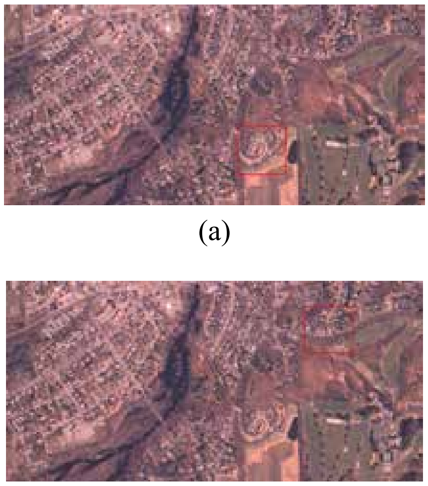
TABLE V
RESULTS OF APPLYING 3 METHODS ON BOARD IMAGE.

Algorithm	Board image		
	Max. No. iteration	*Time(sec)*	*No. of Success in 50 runs*
Proposed	10	21	50
RPT & ICA	40	21	6
RPT & all blocks searching	All the blocks	1.04e+003	50

Considering Fig.12 and Table V, you can see that our method works better than the two other algorithms. The simulation results provided in Table V are obtained by running each of the algorithms 50 times. To have a better comparison between the proposed method and the RPT method along with ICA, we have set the run time of each algorithm equal to 21 sec. In this time spell, our algorithm goes through the maximum number of 10 iterations and the second method (RPT & ICA) experiences 40 iterations. As you can see the former method, our algorithm, converges to the correct answer, in all 50 runs; however, the later is successful in only 6 runs out of the total number of 50. Note that by increasing the number of iterations and therefore the run time of this algorithm, the chance of correct matches increases as well.

According to Table V, the last method that checks all the blocks of the source image finds the right answer in all the cases, but this method is not time efficient at all. For larger images the required time increases dramatically and makes the use of this algorithm irrational or totally vain.

We have repeated the above experiment for the image of Fig.11. The results are demonstrated in Fig.13 and Table VI.



(a)

(b)

Figure13. a) Result of applying the reposed method (Log-Spectrum RPT along with MICA). b) Result of applying RPT along with ICA.

TABLE VI
RESULTS OF APPLYING 2 METHODS ON CONCORD AERIAL IMAGE.

Algorithm	Concord aerial		
	Max .No. iteration	*Time(sec)*	*No. of Success in 50 runs*
Proposed	10	75	50
RPT & ICA	35	75	4

Note that in this case we have refused applying the last method, as it is very time consuming due to size of the source image.

Simulation results show using Log Spectrum RPT increases the precision and convergence speed. Combination of MICA with this algorithm reduces the number of test blocks which are searched to find the best match, especially when a large source image is available. As a result, the proposed algorithm is robust, fast and suitable for solving template matching problems.

VI. CONCLUSION

The main problems in template matching are defining a proper similarity measurement and finding an efficient searching strategy. With our proposed Log Spectrum RPT method and normalized cross correlation, we dealt with the first issue. In this paper, the modified imperialist competitive algorithm was suggested to promote the matching process. The combination of these suggested methods provided an efficient algorithm that was invariant to rotation, scaling, translation and brightness changes. The speed, precision and reliability of this novel template matching algorithm were verified in simulation results.

Introducing a successful template matching technique which is robust to noisy and blurred images can be considered as a future work.

REFERENCES

[1] H. Y. Kim and S. A. de Araújo, " Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast," IEEE Pacific-Rim Symposium on Image and Video Technology, Lecture Notes in Computer Science(From Springer), vol. 4872, pp. 100-113, 2007.

[2] S. A. de Araújo, H. Y. Kim, "Color-Ciratefi: A color-based RST-invariant template matching algorithm," proceedings of IWSSIP, 17th International Conference on Systems, Signals and Image Processing, pp. 101-104, Sep. 2010.

[3] L. Wu-xing, W. Wei, H. Bin-bin, R. He, Sh. Tie-lin, " Fast Template Matching Based on Local Entropy Difference," proceeding of CIP, 2nd international congress on signal and image processing, pp.1-5, Oct. 2009.

[4] H. Duan, Ch. Xu, S. Liu , Sh. Shao, "Template matching using chaotic imperialist competitive algorithm," Pattern Recognition Letters, vol. 31, pp. 1868-1875, Oct. 2010.

[5] Y. H. Lin, Ch. H. Chen and Ch. Ch. Wei ,"Translation, Rotaion, and Scale-invariant Template Matching," International Computer Symposium, pp. 1190-1194, Dec. 2006.

[6] Atashpaz-Gargari, E. and Lucas," Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition," proceedings of IEEE Congress on Evolutionary Computation, pp. 4661-4667, Spt. 2007.

[7] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, " Proceedings of the IEEE, vol. 66, pp. 66-67, Jan.1978.

[8] Y. Zhang, Y. Wang and C. Peng, "Improved imperialist competitive algorithm for constrained optimization," Proceedings of the International Forum on Computer Science-Technology and Applications, Chongqing, China, vol. 1, pp. 204-207, 2009.

[9] N. Pourjafarian, A. A. Safavi, "Application of Imperialist competitive algorithm in e-commerce negotiation," 5th International Conference on e-Commerce in Developing Countries, in press.

[10] L. A. Torres-Méndez, J. C. Ruiz-Suárez, Luis E. Sucar, and G. Gómez, "Translation, Rotation, and Scale-Invariant Object Recognition, " IEEE Transactions on Systems, Man, and Cybernetics, vol. 30, pp. 125-130, Feb. 2000.

[11] M. Roshanaei, E. Atashpaz-Gargari, C. Lucas, "Adaptive beamforming using colonial competitive algorithm," 2nd International Joint Conference on Computational Engineering, Vancouver, Canada, 2008.

[12] A. M. Jasour, E. Atashpaz-Gargari, C. Lucas, "Vehicle fuzzy controller design using imperialist competitive algorithm," Second First Iranian Joint Congress on Fuzzy and Intelligent Systems, Tehran, Iran, 2008.

[13] H. Bahrami, K. Faez, M. Abdechiri, "Imperialist competitive algorithm using chaos theory for optimization (CICA)," Proceedings of the 12th International Conference on Computer Modeling and Simulation, Cambridge, UK, pp. 98-103, 2010.

[14] E. Shokrollahpour, M.Zandieh, B. Dorri, "A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flowshop problem," International Journal of Production Research, May 2010.