

Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast

Hae Yong Kim and Sidnei Alves de Araújo

Escola Politécnica, Universidade de São Paulo, Brazil
{hae, saraujo}@lps.usp.br

Abstract. In this paper, we consider the grayscale template-matching problem, invariant to rotation, scale, translation, brightness and contrast, without previous operations that discard grayscale information, like detection of edges, detection of interest points or segmentation/binarization of the images. The obvious “brute force” solution performs a series of conventional template matchings between the image to analyze and the template query shape rotated by every angle, translated to every position and scaled by every factor (within some specified range of scale factors). Clearly, this takes too long and thus is not practical. We propose a technique that substantially accelerates this searching, while obtaining the same result as the original brute force algorithm. In some experiments, our algorithm was 400 times faster than the brute force algorithm. Our algorithm consists of three cascaded filters. These filters successively exclude pixels that have no chance of matching the template from further processing.

Keywords: Template matching, RST-invariance, segmentation-free shape recognition.

1 Introduction

In this paper, we consider the problem of finding a query template grayscale image Q in another grayscale image to analyze A , invariant to rotation, scale, translation, brightness and contrast (RSTBC), without previous “simplification” of A and Q that discards grayscale information, like detection of edges, detection of interest points and segmentation/binarization. These image-simplifying operations throw away the rich grayscale information, are noise-sensitive and prone to errors, decreasing the robustness of the matching. Moreover, these simplifications cannot be used to find smooth grayscale templates.

The “brute force” solution to this problem performs a series of conventional (BC-invariant) template matchings between the image to analyze A and the query template Q . Image Q must be rotated by every angle, translated to every position and scaled by every factor (within some specified range of scale factors) and a conventional BC-invariant template matching is executed for each instance of the transformed Q . Possibly, the brute force algorithm yields the most precise solution to this problem. However, it takes too long and thus is not practical. Our technique, named Ciratefi, substantially accelerates this searching, while obtaining exactly the same result as the

original brute force algorithm (disregarding incidental numerical imprecision). In some experiments, our algorithm was 400 times faster than the brute force algorithm and obtained exactly the same results.

Fast grayscale RSTBC-invariant template matching is a useful basic operation for many image processing and computer vision tasks, such as visual control [1], image registration [2], and computation of visual motion [3]. Consequently, it has been the object of an intense and thorough study. However, surprisingly, we could not find any technique similar to Ciratefi in the literature.

Some approaches that achieve RST-invariance using detection of interest points and edges include: generalized Hough transform [4]; geometric hashing [5, 6]; graph matching [7]; and curvature scale space [8], adopted by MPEG-7 as standard shape descriptor. These operations and Ciratefi seems to occupy different hierarchies in image processing and computer vision. Indeed, low-level Ciratefi can be used to detect interest points, to be used later by high-level techniques such as geometric hashing and graph matching.

Techniques that achieve RST-invariance using previous segmentation/binarization are described, for example, in [9, 10]. They are in fact algorithms designed to search for binary templates in binary images. So, given a grayscale image to analyze A , they first convert it into a binary image using some thresholding algorithm. Then, they separate each connected component from the background and compute some RST-invariant features for each component. These features are compared with the template's features. The most commonly used rotation-invariant features include Hu's seven moments [11] and Zernike moments [12]. In recent years, many other rotation-invariant features have been developed [13, 14, 15, 16]. All these features are not truly RST-invariant, but only rotation-invariant. These features become scale-invariant by isolating each component and normalizing its area to one. Unfortunately, in many practical grayscale cases, the template Q and the analyzed image A cannot be converted into binary images and thus the above techniques cannot be applied. On the contrary, Ciratefi technique does not need to isolate individual shapes and can be used directly in grayscale (and also binary) template matchings.

Ullah and Kaneko [17] and Tsai and Tsai [18] present two different segmentation-free RTBC-invariant template-matching techniques. However, their techniques are not scale-invariant. Hence, the key problem seems to be: "How to obtain the scale-invariance without isolating the shapes or components?" Or, in other words: "How can we estimate the scale of a shape without determining first its boundaries?"

Our Ciratefi algorithm consists of three cascaded filters. Each filter successively excludes pixels that have no chance of matching the template from further processing, while keeping the "candidate pixels" that can match the template to further refined classifications. The first filter, called Cifi (circular sampling filter), uses the projections of images A and Q on circles to divide the pixels of A in two categories: those that have no chance of matching the template Q (to be discarded) and those that have some chance (called first grade candidate pixels). This filter is responsible for determining the scale without isolating the shapes. It determines a "probable scale factor" for each first grade candidate pixel. The second filter, called Rafi (radial sampling filter), uses the projections of images A and Q on radial lines and the "probable scale factors" computed by Cifi to upgrade some of the first grade candidate pixels to second grade. It also assigns a "probable rotation angle" to each second grade candidate

pixel. The pixels that are not upgraded are discarded. The third filter, called Tefi (template matching filter), is a conventional BC-invariant template matching. The second grade candidate pixels are usually few in number and Cifi and Rafi have already computed their probable scales and rotation angles. Thus, the template matching can quickly categorize all the second grade candidate pixels in true and false matchings.

There are some other papers that use circular or radial projections, like [19, 20]. However, their objectives (fingerprint and Chinese character recognition) are completely different from ours, and they intend to obtain neither scale-invariance nor segmentation-free recognition.

Ciratefi is not robust to occlusions (neither is the brute force algorithm). However, in the presence of occlusions, it appears that the template can be divided into smaller sub-templates and the results of the sub-matchings can be combined to detect the original template. Finally, Ciratefi (as well as the brute force algorithm) can easily be made parallel.

2 The Brute Force Algorithm

In this section, we describe the “brute force” algorithm. This algorithm makes use of the BC-invariant template matching.

2.1 BC-Invariant Template Matching

Template matching uses some difference measuring function to evaluate how well the template Q matches a given position of image A . Usually, sum of absolute differences, sum of squared differences, cross-correlation and correlation coefficient are used as difference measuring functions. We have adopted the correlation coefficient, because it always ranges from -1 to +1 and is BC-invariant. However, let us make the following reasoning to explicit the brightness/contrast-invariance.

Let \mathbf{x} be the columnwise vector obtained by copying the grayscales of Q 's pixels and let \mathbf{y} be the vector obtained by copying the grayscales of the pixels of A 's region to be correlated with Q . Then, the brightness/contrast correction can be written as a least squares problem:

$$\mathbf{y} = \beta\mathbf{x} + \gamma\mathbf{1} + \boldsymbol{\varepsilon} \quad (1)$$

where $\mathbf{1}$ is a vector of 1's, $\boldsymbol{\varepsilon}$ is the vector of residual errors, β is the contrast correction factor and γ is the brightness correction factor. The problem consists on finding β and γ that minimizes $\boldsymbol{\varepsilon}^2$. This problem has a computationally fast solution. Let $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$ be the mean-corrected vector, where $\bar{\mathbf{x}}$ is the mean of \mathbf{x} . Similar definitions are applicable to \mathbf{y} . Then:

$$\beta = \frac{\tilde{\mathbf{x}}\tilde{\mathbf{y}}}{\tilde{\mathbf{x}}^2}, \gamma = \bar{\mathbf{y}} - \beta\bar{\mathbf{x}}, \text{ and } \boldsymbol{\varepsilon} = \tilde{\mathbf{y}} - \beta\tilde{\mathbf{x}}. \quad (2)$$

The correlation coefficient r_{xy} can be computed:

$$r_{xy} = \frac{\tilde{x}\tilde{y}}{\|\tilde{x}\|\|\tilde{y}\|} = \frac{\beta\tilde{x}^2}{\|\tilde{x}\|\|\tilde{y}\|}. \tag{3}$$

We assume that the correlation is zero if a large brightness or contrast correction is required, because in this case the template and the image are likely quite different. The correlation is assumed to be zero if $|\beta| \leq t_\beta$ or $1/t_\beta \leq |\beta|$, where $0 < t_\beta \leq 1$ is a chosen contrast correction threshold. For example, $t_\beta = 0.5$ means that regions of A with contrast less than half or more than twice the Q 's contrast will be considered as not correlated with Q . This also avoids divisions by zero in regions of A with almost constant grayscale (where the values of $\|\tilde{y}\|$ are almost zero). The correlation is also assumed to be zero if $|\gamma| > t_\gamma$, where $0 < t_\gamma \leq 1$ is a chosen brightness correction threshold (we assume that the grayscales of the images are real numbers within the interval $[0, 1]$). We define Corr as the correlation that takes into account the contrast and brightness corrections:

$$\text{Corr}(x, y) = \begin{cases} 0, & \text{if } |\beta| \leq t_\beta, 1/t_\beta \leq |\beta| \text{ or } |\gamma| > t_\gamma \\ r_{xy}, & \text{otherwise} \end{cases} \tag{4}$$

Depending on the application, we can use either the absolute value $|\text{Corr}|$ (to allow matching negative instances of the template) or the value of Corr with signal (negative instances will not match the template).

2.2 RSTBC-Invariant Template Matching

To obtain RSTBC-invariant template matching, we said above that the query shape Q must be rotated by every angle and scaled by every factor. In practice, it is not possible to rotate and scale Q by every angle and scale, but only by some discrete set of angles and scales. Figure 1 depicts some of the ‘‘frog’’ template rotated in $m=36$ different angles ($\alpha_0=0, \alpha_1=10, \dots, \alpha_{35}=350$) and scaled by $n=6$ different factors ($s_0=0.6, s_1=0.7, \dots, s_5=1.1$). To avoid that a small misalignment may cause a large mismatching, a low-pass filter (for example, the Gaussian filter) smoothes both images A and Q . This low-pass filtering lessens the errors introduced by using discrete scales and angles.

Then, each pixel p of A is tested for matching against all the transformed templates ($6 \times 36 = 216$ templates, in our case). If the largest absolute value of the contrast/brightness-aware correlation Corr at pixel p is above some threshold t_f , the template is considered to be found at p . Figure 2 depicts the detection of the frog shape, using $t_f=0.9, t_\beta = 0.1$ and $t_\gamma = 1$. Besides detecting the shape, the brute force algorithm also returns the precise scale factor and rotation angle for each matching. The only problem is that this process takes 9173s, or two and half hours using a 3GHz-Pentium4 (image A has 465×338 pixels and image Q has 52×51 pixels). Our Ciratefi algorithm does the same task in only 22s.



Fig. 1. Some of the rotated and scaled templates

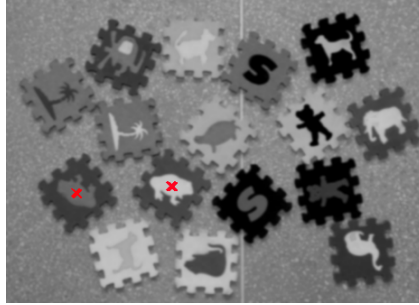


Fig. 2. Frog shapes detected by the brute force algorithm. Each matching is marked with a red “x”.

3 Circular Sampling Filter

Circular sampling filter (Cifi) uses the projections of the images A and Q on a set of rings (figure 3a) to detect the first grade candidate pixels and their probable scales. As we show experimentally in subsection 7.2, the correct choice of number of circles l is not essential to our algorithm, because Rafi and Tefi will further filter the first grade candidate pixels. Figure 3b depicts the output of Cifi filtering, where the first grade candidate pixels are depicted in magenta.

Given an image B , let us define the circular sampling $Cis_B(x, y, r)$ as the average grayscale of the pixels of B situated at distance r from the pixel (x, y) :

$$Cis_B(x, y, r) = \int_0^{2\pi} B(x + r \cos \theta, y + r \sin \theta) d\theta \tag{5}$$

In practice, a sum must replace the integral and a computer graphics algorithm for drawing circles, as [21], can be used to find efficiently all the pixels that belong to a specific circle. Given the template image Q and the set of n scales (in our example, $s_0=0.6, s_1=0.7, \dots, s_5=1.1$) the image Q is resized to each scale s_i , obtaining the resized templates Q_0, Q_1, \dots, Q_{n-1} . Then, each resized template Q_i is circularly sampled at a set of l predefined circle radii (in our example, $l=13$, and $r_0=0, r_1=2, \dots, r_{12}=24$ pixels), yielding a 2-D matrix of multi-scale rotation-invariant features C_Q with n rows (scales) and l columns (radii):

$$C_Q[i, k] = Cis_{Q_i}(x_0, y_0, r_k), \quad 0 \leq i < n \text{ and } 0 \leq k < l \tag{6}$$

where (x_0, y_0) is the central pixel of Q .

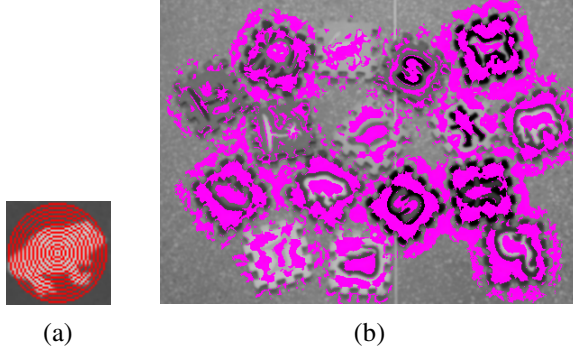


Fig. 3. Circular sampling filter Cifi. (a) Circles where the image is sampled. (b) The output of Cifi with the first grade candidate pixels in magenta.

Given the image to analyze A , we build a 3-D image $C_A[x,y,k]$:

$$C_A[x, y, k] = \text{Cis}_A(x, y, r_k), \quad 0 \leq k < l \text{ and } (x, y) \in \text{domain}(A) \quad (7)$$

Cifi uses matrices C_Q and C_A and the contrast and brightness thresholds t_β and t_γ to detect the circular sampling correlation CisCorr at the best matching scale for each pixel (x,y) :

$$\text{CisCorr}_{A,Q}(x, y) = \text{MAX}_{i=0}^{n-1} \left[\left| \text{Corr}(C_Q[i], C_A[x, y]) \right| \right] \quad (8)$$

A pixel (x,y) is classified as a first grade candidate pixel if $\text{CisCorr}_{A,Q}(x, y) \geq t_1$ for some threshold t_1 (in the example, $t_1=0.95$). As we show in subsection 7.1, the adequate choice of t_1 is not critical, provided that it is low enough to not discard the real matching pixels. The probable scale CisPS of a first grade candidate pixel (x,y) is the best matching scale:

$$\text{CisPS}_{A,Q}(x, y) = \text{ARGMAX}_{i=0}^{n-1} \left[\left| \text{Corr}(C_Q[i], C_A[x, y]) \right| \right] \quad (9)$$

In our example, the computation of the 3-D image $C_A[x, y, k]$ took 2.5s and the computation of $\text{CisCorr}_{A,Q}(x, y)$ for all pixels of A took 4.5s. The remaining Cifi operations are almost instantaneous.

4 Radial Sampling Filter

The second filter is called radial sampling filter (Rafi) and uses the projections of images A and Q on a set of radial lines to upgrade some of the first grade candidate pixels to second grade. The pixels that are not upgraded are discarded. It also assigns a “probable rotation angle” to each second grade candidate pixel. Figure 4a marks in blue the radial lines and figure 4b marks with a red “x” each second grade candidate

pixel. The set of inclinations of the radial lines must be equal to the m chosen rotation angles (in our example, $\alpha_0=0, \alpha_1=10, \dots, \alpha_{35}=350$). As we show in subsection 7.2, the choice of m is not critical, provided that it is not too small.

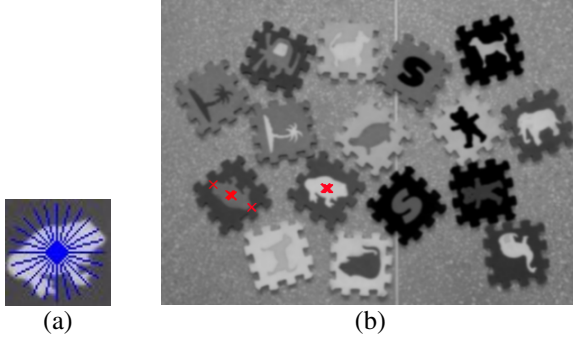


Fig. 4. Radial sampling filter Rafi. (a) The radial lines where the image is sampled. (b) The output of Rafi, where each second grade candidate pixel is marked with a red “x”.

Given an image B , let us define the radial sampling $\text{Ras}_B^\lambda(x, y, \alpha)$ as the average grayscale of the pixels of B located on the radial line with one vertex at (x, y) , length λ and inclination α :

$$\text{Ras}_B^\lambda(x, y, \alpha) = \int_0^\lambda B(x + t \cos \alpha, y + t \sin \alpha) dt \quad (10)$$

In practice, the integral must be replaced by a sum and a line drawing algorithm (as [22]) can be used to find efficiently all the pixels that belong to a line.

Given the template Q and the set of m angle inclinations $(\alpha_0, \alpha_1, \dots, \alpha_{m-1})$, Q is radially sampled using $\lambda=r_{l-1}$ (the largest sampling circle radius), yielding a vector R_Q with m features:

$$R_Q[j] = \text{Ras}_Q^{\eta-1}(x_0, y_0, \alpha_j), \quad 0 \leq j < m \quad (11)$$

where (x_0, y_0) is the central pixel of Q .

For each first grade candidate pixel (x, y) , A is radially sampled at its probable scale $i = \text{CisPS}_{A,Q}(x, y)$. The largest radius r_{l-1} resized to the probable scale s_i becomes $\lambda = s_i r_{l-1}$. Thus:

$$R_A[x, y, j] = \text{Ras}_R^{s_i \eta-1}(x, y, \alpha_j), \quad 0 \leq j < m \text{ and } (x, y) \in \text{f_gr_cand}(A) \quad (12)$$

At each first grade candidate pixel (x, y) , Rafi uses the vectors $R_A[x, y]$, R_Q and contrast and brightness thresholds t_β and t_γ to detect the radial sampling correlation RasCorr at the best matching angle:

$$\text{RasCorr}_{A,Q}(x,y) = \text{MAX}_{j=0}^{m-1} \left[\left| \text{Corr}(R_A[x,y], \text{cshift}_j(R_Q)) \right| \right], (x,y) \in \text{f_gr_cand}(A) \quad (13)$$

where “cshift_j” means circular shifting *j* positions of the argument vector. A first grade pixel (*x,y*) is upgraded to second grade if:

$$\text{RasCorr}_{A,Q}(x,y) \geq t_2 \quad (14)$$

for some threshold *t*₂ (in the example, *t*₂=0.9). As we show in subsection 7.2, the adequate choice of *t*₂ is not critical, provided that it is low enough to not discard the real matching pixels. The probable rotation angle RasAng at a second grade candidate pixel (*x,y*) is the best matching angle:

$$\text{RasAng}_{A,Q}(x,y) = \text{ARGMAX}_{j=0}^{m-1} \left[\left| \text{Corr}(R_A[x,y], \text{cshift}_j(R_Q)) \right| \right] \quad (15)$$

In the example, the computation of RasCorr_{A,Q}(*x,y*) in all pixels (*x,y*) of *A* took 13s. The remaining Rafi operations are almost instantaneous.

5 Template Matching Filter

The third filter is called Tefi and it is simply the BC-invariant template matching, applied only at the second grade candidate pixels, using the probable scale and angle determined respectively by Cifi and Rafi. Figure 5 depicts its output (that is also the final output of the Ciratefi algorithm).

Similarly to the RSTBC-invariant template matching, Tefi first resizes and rotates the template *Q* to all *m* angles and *n* scales. Let (*x,y*) be a second grade candidate pixel, with its probable scale *i* = CisPS_{A,Q}(*x,y*) and probable angle *j* = RasAng_{A,Q}(*x,y*). Then, Tefi computes the contrast/brightness-aware correlation Corr between the template image *Q* at scale *s*_{*i*} and angle α_{*j*}, and the image *A* at pixel (*x,y*). If the absolute value of the correlation is above some threshold *t*₃, the template is considered to be found at pixel (*x,y*).

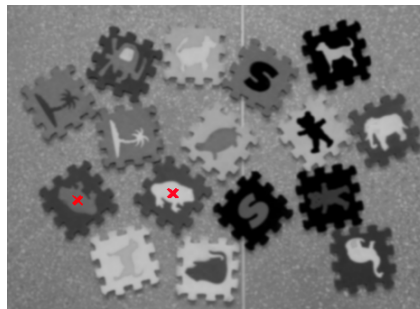


Fig. 5. The final output of Ciratefi. Each matching pixel is marked with a red “x”.

Adopting the same threshold used in the brute-force algorithm (that is $t_3=t_f$), the output is usually equal or very similar to the output of the brute-force algorithm. For even more robustness, it is possible to test the matchings at a set of scales around i (for example, $i-1, i, i+1$) and at a set of angles around j (for example, $j-1, j, j+1$, where the addition and subtraction must be computed modulus m). In our example, Tefi took 1s to be computed.

6 Complexity Analysis

The precise computational complexity of Ciratefi depends on many unforeseeable factors, like the number of the first and second grade candidate pixels. However, we will make some assumptions and approximations to analyze its complexity.

Let N be the number of pixels of the image to analyze A and M the number of pixels of the template image Q . To make our analysis, we will assume that the number of scales n , the number of angles m , and the number of sampling circles l are all $O(\sqrt{M})$. We will ignore all operations that does not depend on N , because usually N is much larger than M .

The brute force algorithm makes $n \times m$ template matchings for each pixel of A . Considering that each template matching makes $O(M)$ operations, this algorithm's complexity is $O(NnmM)$, or approximately $O(NM^2)$.

Ciratefi has four operations that depend on N :

- The generation of the 3-D image $C_A[x,y,k]$ takes $O(NM)$, considering that almost all pixels of the domain of Q must be scanned for each pixel (x,y) of A .
- The computation of CisCorr for all pixels of A takes $O(Nnl)$, or approximately $O(NM)$.
- The computation of $R_A[x,y,j]$ and RasCorr for all first grade candidate pixels takes $O(N_1m\sqrt{M})$ and $O(N_1m^2)$, respectively, where N_1 is the number of the first grade candidate pixels. $O(N_1m\sqrt{M} + N_1m^2)$ can be approximated by $O(NM)$.
- The computation of Tefi takes $O(N_2M)$, where N_2 is the number of the second grade candidate pixels, and $O(N_2M) \leq O(NM)$.

Consequently, the complexity of Ciratefi is $O(NM)$, while the complexity of the brute force algorithm is $O(NM^2)$. This makes a lot of difference! In our example, $M \approx 2500$, justifying why Ciratefi was 400 times faster than the brute force algorithm.

7 Experimental Results

7.1 Experiments

We made three experiments to evaluate Ciratefi, using a total of 145 images. In all images, shape instances appear in different rotations, scales, brightnesses and contrasts. We do not compare the accuracy of our algorithm with other techniques

because, as we considered in section 1, seemingly there is no rotation and scale-invariant grayscale template matching in the literature (except the brute-force algorithm). Choosing adequate parameters, Ciratefi and the brute-force algorithm yield exactly the same results, what makes foreseeable the behavior of our algorithm.

In the first experiment, we took 70 pictures of 16 toy figures randomly scattered on the floor. Then, we searched in the 70 images (each one with 512×384 pixels) for 5 templates: frog, dog, palm_tree, bear and letter_s (figure 6), extracted from one of the 70 images. 10 instances of the searched templates appear in each image. Figure 7 shows one final output where the matching positions are marked with “x”. All 700 Ciratefi matchings were perfect, without any false positive or false negative. Note that there are faintly visible shapes (dog and bear). These shapes also were successfully detected, in spite of their low contrast, using $t_{\beta}=0.1$.



Fig. 6. Template images (51×51 pixels)

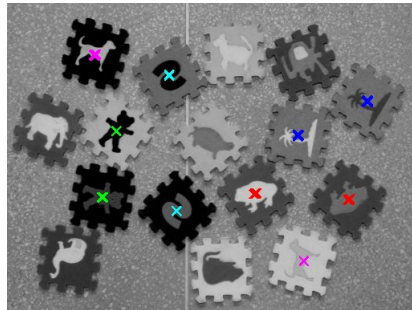


Fig. 7. Result of detection of the 5 templates



Fig. 8. Detection of McDonald’s symbols. (a) Template. (b) Perfect matching. (c) False negative case encircled in yellow.

In the second experiment, we searched for the McDonald's[®] symbol (figure 8a) in 60 images taken from different places and objects. The smallest image has 96×94 pixels, and the largest has 698×461 pixels. Figures 8b and 8c show two sample images matched against the template. Each matching is marked with a red "x". This experiment presented only one false positive and two false negatives. The bright background of the symbol probably caused one of the false negatives, marked with a yellow circle in figure 8c. Note that we did not use the color information. Probably, this task would become much easier using the color information.

Finally, in the third experiment we tested the detection of buildings with a specific shape in 15 remote sensing images with 515×412 pixels, provided by the Google Earth, using a grayscale template with 40×40 pixels (figure 9a). Figures 9b and 9c depict some examples of the experiment. In this experiment, the building appears 187 times in the 15 analyzed images. We detected 18 false positives and 16 false negatives cases, caused mainly by shadows, different illumination angles, occlusions and blurred images. The results of the three experiments are summarized in Table 1.

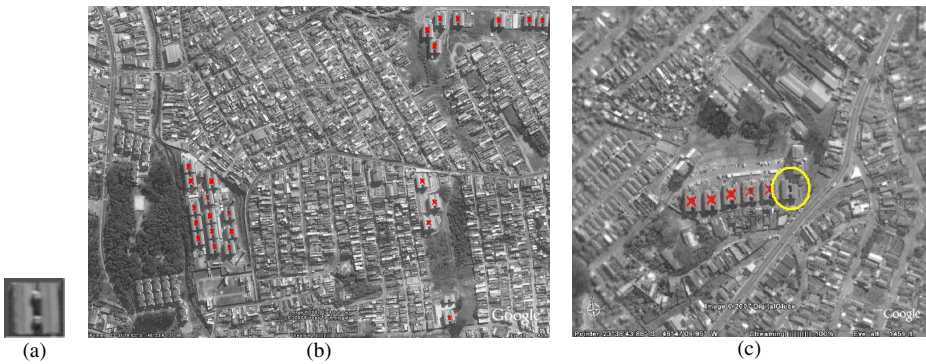


Fig. 9. Detection of a building in remote sensing images. (a) Template image. (b) Perfect matching. (c) False negative case encircled in yellow.

Table 1. Summary of the tests

Experiment	Instances of the shape	Correct detections	False positives	False negatives
Toys	700	700	0	0
McDonald's	116	114	1	2
Buildings	187	171	18	16

7.2 Parameters

We tested the sensitivity of Ciratefi to different choices of parameters, such as number of circles (l), number of radial lines (m) and thresholds t_1 , t_2 , t_3 , t_β and t_γ . We demonstrate below that the only really important parameter is t_3 . We searched for the frog template in one image of the toy experiment (figure 7). We used 10 scale factors ($s_0=0.4$, $s_1=0.5, \dots$, $s_9=1.3$). In each table, the fixed parameters appear in the first line.

Table 2. Sensitivity to the number of circles l

$m = 36, t_1 = 0.95, t_2 = 0.9, t_3 = 0.9, t_\beta = 0.1, t_\gamma = 1$			
Number of circles l	False positives	False negatives	First grade candidate pixels
05	0	0	83,310
10	0	0	69,389
15	0	0	50,519
20	0	0	74,970
25	0	0	77,375

Table 2 shows that the number of circles l does not have a strong influence on the final result, because no error was detected even varying its value. However, the suitable choice of l is important to minimize the number of the first grade candidate pixels and accelerate the processing.

Table 3. Sensitivity to the number of radial lines m

$l = 13, t_1 = 0.95, t_2 = 0.9, t_3 = 0.9, t_\beta = 0.1, t_\gamma = 1$			
Number of radial lines m	False positives	False negatives	Second grade candidate pixels
08	0	2	433
15	0	2	42
20	0	0	30
30	0	0	35
40	0	0	41

Table 3 shows that too small number of radial lines m can produce false negatives by eliminating the true matching pixels. In this experiment, no error was detected for $m \geq 20$. However, the algorithm becomes slower using large m .

Table 4. Sensitivity to the thresholds t_1, t_2 and t_3

$l = 13, m = 36, t_\beta = 0.1, t_\gamma = 1$					
Thresholds t_1, t_2, t_3	False Positives	False negatives	Thresholds t_1, t_2, t_3	False positives	False negatives
0.50, 0.50, 0.50	8376	0	0.95, 0.75, 0.75	104	0
0.50, 0.50, 0.75	286	0	0.95, 0.75, 0.95	0	0
0.50, 0.50, 0.95	0	0	0.95, 0.95, 0.95	0	0
0.75, 0.75, 0.50	1325	0	0.95, 0.95, 0.98	0	2
0.75, 0.75, 0.75	104	0	0.95, 0.98, 0.95	0	2
0.75, 0.95, 0.95	0	0	0.98, 0.95, 0.95	0	2

Table 4 shows that a incorrect choice of t_3 may produce false negatives or false positives. However, the choices of t_1 and t_2 are not critical to the detection of the shape, as long as their values are not too high to discard the true matchings. Indeed, the detection was errorless for $t_3=0.95$, for any $t_1 \leq 0.95$ and $t_2 \leq 0.95$. However, small values of t_1 and t_2 make the algorithm slower.

Table 5. Sensitivity to the thresholds t_β and t_γ

$l = 13, m = 36, t_1 = 0.95, t_2 = 0.9, t_3 = 0.9$					
Thresholds t_β, t_γ	False positives	False negatives	Thresholds t_β, t_γ	False positives	False negatives
0.10, 0.10	0	1	0.50, 0.50	0	1
0.10, 0.50	0	0	0.50, 1.00	0	1
0.10, 1.00	0	0	0.75, 0.10	0	1
0.25, 0.10	0	1	0.75, 0.50	0	1
0.25, 0.50	0	0	0.75, 1.00	0	1
0.25, 1.00	0	0	1.00, 0.10	0	2
0.50, 0.10	0	1	1.00, 1.00	0	2

As expected, table 5 shows that too large t_β or too small t_γ yields false negatives. However, there are large ranges of values that do not produce errors ($0.1 \leq t_\beta \leq 0.25$ and $0.5 \leq t_\gamma \leq 1.0$).

8 Conclusions and Future Works

In this paper, we have presented a new grayscale template matching algorithm, invariant to rotation, scale, translation, brightness and contrast, named Ciratefi. Differently from many other techniques, Ciratefi does not discard the rich grayscale information through operations like detection of edges, detection of interest points or segmentation/binarization of the images. The proposed algorithm was about 400 times faster the brute force algorithm in the experiments, while yielding practically the same output. Complexity analysis has shown that Ciratefi is indeed superior to the brute force algorithm. Experimental results demonstrate the efficiency and the robustness of the proposed technique. A straightforward generalization of this technique is to use the color information, together with the luminance. Another possible generalization is to use other features besides the mean grayscales on circles and radial lines, such as standard deviations, and maximum or minimum values.

References

1. Hutchinson, S., Hager, G.D., Corke, P.I.: A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation* 13(5), 651–670 (1996)
2. Brown, L.G.: A survey of image registration techniques. *ACM Computing Surveys* 24(4), 325–376 (1992)
3. Anandan, P.: A computational framework and an algorithm for the measurement of visual motion. *Int. J. Comput. Vision* 2(3), 283–310 (1989)
4. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition* 13(2), 111–122 (1981)
5. Lamdan, Y., Wolfson, H.J.: Geometric hashing: a general and efficient model-based recognition scheme. In: *Int. Conf. on Computer Vision*, pp. 238–249 (1988)
6. Wolfson, H.J., Rigoutsos, I.: Geometric hashing: an overview. *IEEE Computational Science & Engineering*, 10–21 (October-December 1997)

7. Leung, T.K., Burl, M.C., Perona, P.: Finding faces in cluttered scenes using random labeled graph matching. In: *Int. Conf. on Computer Vision*, pp. 637–644 (1995)
8. Mokhtarian, F., Mackworth, A.K.: A Theory of Multi-scale, Curvature Based Shape Representation for Planar Curves. *IEEE T. Pattern Analysis Machine Intelligence* 14(8), 789–805 (1992)
9. Kim, W.Y., Yuan, P.: A practical pattern recognition system for translation, scale and rotation invariance. In: *Computer Vision and Pattern Recognition*, pp. 391–396 (1994)
10. Torres-Méndez, L.A., Ruiz-Suárez, J.C., Sucar, L.E., Gómez, G.: Translation, rotation and scale-invariant object recognition. *IEEE Trans. Systems, Man, and Cybernetics - part C: Applications and Reviews* 30(1), 125–130 (2000)
11. Hu, M.K.: Visual Pattern Recognition by Moment Invariants. *IRE Trans. Inform. Theory* 1(8), 179–187 (1962)
12. Teh, C.H., Chin, R.T.: On image analysis by the methods of moments. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 10(4), 496–513 (1988)
13. Li, J.H., Pan, Q., Cui, P.L., Zhang, H.C., Cheng, Y.M.: Image recognition based on invariant moment in the projection space. In: *Int. Conf. Machine Learning and Cybernetics, Shanghai*, vol. 6, pp. 3606–3610 (August 2004)
14. Flusser, J., Suk, T.: Rotation moment invariants for recognition of symmetric objects. *IEEE T. Image Processing* 15(12), 3784–3790 (2006)
15. Dionisio, C.R.P., Kim, H.Y.: A supervised shape classification technique invariant under rotation and scaling. In: *Int. Telecommunications Symposium*, pp. 533–537 (2002)
16. Tao, Y., Ioerger, T.R., Tang, Y.Y.: Extraction of rotation invariant signature based on fractal geometry. *IEEE Int. Conf. Image Processing* 1, 1090–1093 (2001)
17. Ullah, F., Kaneko, S.: Using orientation codes for rotation-invariant template matching. *Pattern Recognition* 37, 201–209 (2004)
18. Tsai, D.M., Tsai, Y.H.: Rotation-invariant pattern matching with color ring-projection. *Pattern Recognition* 35, 131–141 (2002)
19. Chang, D.H., Hornak, J.P.: Fingerprint recognition through circular sampling. *The Journal of Imaging Science and Technology* 44(6), 560–564 (2000)
20. Tao, Y., Tang, Y.Y.: The feature extraction of chinese character based on contour information. In: *Int. Conf. Document Analysis Recognition (ICDAR)*, pp. 637–640 (September 1999)
21. Bresenham, J.E.: A linear algorithm for incremental digital display of circular arcs. *Comm. ACM* 20(2), 100–106 (1977)
22. Bresenham, J.E.: Algorithm for computer control of a digital plotter. *IBM Systems Journal* 4(1), 25–30 (1965)