Journal of Pattern Recognition Research 1 (2014) 1-12

Received Jul 02, 2013. Revised Jan 07, 2014. Accepted Jan 07, 2014.

Speed-up Template Matching through Integral Image based Weak Classifiers

Tirui Wu *twu32@ford.com*
Ford Motor Research & Engineering (Nanjing) Co., Ltd., Nanjing, China

Alexander Toet *lex.toet@tno.nl*
TNO, Kampweg 5, 3769DE, Soesterberg, The Netherlands

Abstract

Template matching is a widely used pattern recognition method, especially in industrial inspection. However, the computational costs of traditional template matching increase dramatically with both template-and scene imagesize. This makes traditional template matching less useful for many (e.g. real-time) applications. In this paper, we present a method to speed-up template matching. First, candidate match locations are determined using a cascaded blockwise computation of integral image based binary test patterns. Then, traditional template matching is applied at the candidate match locations to determine the best overall match. The results show that the proposed method is fast and robust.

Keywords: Template matching, integral images, binary test patterns.

1. Introduction

Matching a template sub-image into a given image is one of the most common techniques used in signal and image processing [1], and widely used in many fields related to computer vision and image processing, such as image retrieval [2], image recognition [3], image registration [4], object detection [5] and stereo matching [6].

Traditional template-matching consists in sliding the template over the search area and, at each position, calculating a correlation (ordistortion) measure estimating the degree of (dis-) similarity between the template and the image. Then, the maximum correlation (or minimum distortion) position is taken to represent the instance of the template into the image under examination, with a threshold on the (dis-)similarity measure allowing for rejection of poor matches. The typical distortion measures used in template matching algorithms are the sum of absolute differences (SAD) and the sum of squared differences (SSD), while normalized cross-correlation (NCC) is by far the most widely used correlation measure.

The NCC value ρ representing the similarity of a template image $T(i,j)$ of size $m \times n$ at the location (x,y) in a scene image $I(x,y)$ of size $M \times N$ is defined as

$$\rho(x,y) = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(x+i,y+j)-\mu_I)\cdot(T(i,j)-\mu_T)}{\sqrt{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(x+i,y+j)-\mu_I)^2\cdot(T(i,j)-\mu_T)^2}} \quad (1)$$

for all $(x, y) \in M \times N$, with

$$\begin{aligned} \mu_I &= \tfrac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(x+i, y+j), \\ \mu_T &= \tfrac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} T(i, j) \end{aligned} \tag{2}$$

Due to the complexity of the matching function and the large number of locations to check, traditional template matching is computationally expensive. To speed-up the basic approach, two categories of approaches have been proposed: (*i*) efficient representation of both the template and the image so that matching can be done quickly, and (*ii*) fast search (reduced precision) techniques to reduce the number of matching operations between the target and the image. A well-known example of the first approach is the use of the fast Fourier transform (FFT) to simultaneously calculate the correlation between the template and every part of the input image as the product of their Fourier transforms [7]. A typical example of the second approach is the use of multi-resolution schemes [8, 9] to locate a coarse-resolution template into a coarse-resolution representation of the image and then refining the search at higher resolution levels only at locations where the low-resolution similarity is high. However, these methods still have limited practical value, due algorithmic complexity, strict conditions for application, and sensitivity to local variations in luminance, scale, rotation and distortion.

Here we propose a new template matching scheme that combines computational efficiency with low sensitivity to local image variations, using blockwise computed binary test patterns based on integral images to determine candidate match locations over the image support [10]. Integral images have previously been applied to speed-up template matching, e.g. to accelerate the computation of the NCC [11, 12], and to efficiently compute polynomial approximations [13] or characteristic features [14, 15] of image patches. Also, it has been demonstrated that template matching can be performed fast and robust by computing and matching image features in a blockwise fashion from integral images [16]. The contribution of the proposed approach is that it enables fast and robust assessment of candidate match locations through a cascaded blockwise computation of weak integral image based binary test patterns (classification ability of an individual binary test pattern is weak). This approach speeds up conventional template matching by quickly discarding background image regions that are unlikely to contain the template. The use of integral images allows fast computation of the individual weak block binary test patterns, while their cascaded computation allows early termination of the computational process if the initial estimates suggest that the location corresponds to a poor match.

The rest of this paper is organized as follows. First we introduce the concept of integral images. Next we describe our new fast template matching scheme. Then, we present the results of some computational experiments that demonstrate the efficiency and robustness of our new method. Finally, we end with some conclusions.

2. Integral images

This section describes the concept of integral images that was introduced by Viola and Jones [10] in computer vision and which is based on prior work in computer graphics [17]. Integral images (also known as summed-area tables) allow fast computation of rectangular image features since they enable the summation of image values over any rectangle image region in constant time. In the next section we will show that integral images can also serve

to speed up template matching for rectangular shaped templates. Let an $i(x, y)$ be the original image value at location x, y. The integral image $I(x, y)$ is an intermediate image representation with a value at image location x, y that is equal to the sum of the pixel values above and to the left of x, y including the value at the location x, y itself:

$$I(x, y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} (x', y') \tag{3}$$

The integral image can be computed in a single pass over the original image, using the following pair of recursive formulas [10]:

$$\begin{aligned} S(x, y) &= S(x, y-1) + i(x, y), \\ I(x, y) &= I(x-1, y) + S(x, y) \end{aligned} \tag{4}$$

where $S(x, y)$ represents the cumulative column sum, with $S(x, -1) = 0$ and $I(-1, y) = 0$ Since each pixel involves only two additions, the computation of an integral image of $M \times N$ pixels requires only $2MN$ additions. A visualization of an integral image is shown in Fig.1(b). Once the integral image has been computed, the sum of values over any rectangular region of the image i can be computed from only four references to the integral image array (see Fig.2):

$$I(x, y) = \sum_{x=x_a}^{x_b} \sum_{y=y_a}^{y_b} i(x, y) = I(x_b, y_b) + I(x_a - 1, y_a - 1) - I(x_a - 1, y_b) - I(x_b, y_a - 1) \tag{5}$$

Integral images have been widely used to speed up the computation of region-based statistical measures, such as area sums [18], covariance [19, 20], and co-occurrence [21] and have successfully been applied to texture mapping [17], the detection of features [22], faces [10], humans [23], and objects [24], stereo correspondence [25], and adaptive thresholding [26].

(a) (b)

Fig. 1: (a) Original image with (b) its integral image representation.

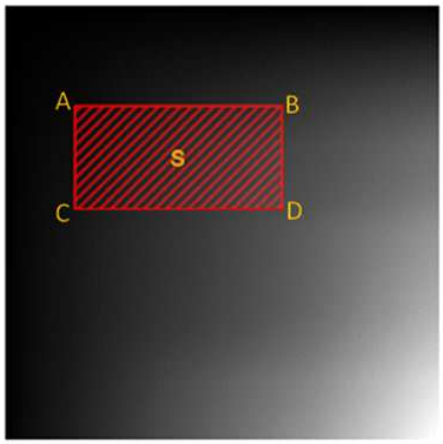


Fig. 2: Using integral images it takes only three additions and four memory accesses to calculate the sum of intensities over a rectangular image region of any size (S = A-B-C+D).

3. Fast template matching

The fast template matching method proposed here involves two phases. In the first phase, a set of candidate target locations (locations that match the template image to a certain degree) is determined by cascading a set of blockwise computed weak image binary test patterns. Since these weak image blockbinary test patterns are based on local averages over rectangular image regions, they can efficiently be computed using integral images. In the second phase, conventional template matching is applied to the set of candidate target locations. Restricting conventional template matching to a small set of image locations significantly reduces the overall computational effort. In this section we will introduce the use of cascaded weak image block binary test patterns to select candidate target locations, and we will show how we can speed-up their computation using integral images.

Let T represent a template (small grayscale image) of size $m \times n$ that is to be matched with a larger grayscale image I of size $M \times N$ ($m < M, n < N$). To identify candidate match locations we will use strong binary test patterns that are constructed in a cascaded fashion from a set of weak binary test patterns. The weak binary test patterns are computed as follows (see also Fig.3). First, the template image T is partitioned into $k \times l$ (k and l both positive integers) rectangular blocks of equal size. Next, both the mean of each template block and the mean of the entire template image are calculated. The sign of the difference between the mean of a block and the overall template mean is adopted a weak binary test pattern. Thus, the weak binary test pattern c_i^T of the i^{th} template block is given by

$$c_i^T = \begin{cases} 1, \text{if } \mu_i > \mu \\ -1, \text{if } \mu_i \leq \mu \end{cases} \tag{6}$$

where, $i \in \{l, k \times l\}, \mu_i$ is the mean value of the i^{th} template block, and μ represents the mean image value over the entire template. A strong binary test pattern is then obtained by computing consecutive weak binary test patterns in a cascaded fashion (Fig. 3).

The first phase of the template matching method proposed here involves sliding the template over the image to determine (candidate target) locations where the template pattern reflects the local image structure to a certain degree. Thereto, at each position of the template (i.e. at each location in the image) set of weak image block binary test patterns c_i

is computed over the current image window in the same order in which the weak binary test patterns c_i^T were computed for the template image (i.e. by partitioning the $M^T \times N^T$ local image window into $k \times l$ rectangular blocks). To reduce the overall computational costs of the matching process, classification is performed in a cascaded fashion: each local weak image block binary test pattern is first compared to the corresponding weak template block binary test pattern c_i^T, before computing the next one (see Fig. 3). If $c_i = c_i^T$ then the next local weak image block binary test pattern is computed, else the computation of local weak image block binary test patterns is terminated and the position of the current image window is shifted one step further. If all weak local image block binary test patterns and template block binary test patterns are the same, then the current window passes the overall (strong) classification and its position is added to a list of candidate target positions. In practice, the requirement of identity between the complete set of template and image weak block binary test patterns is too strict since there will typically be rotation, scale or distortion differences between the template and the image content. The result would be that no scan location would pass all binary test patterns and no candidate matches would be found. To include some tolerance for small image variations in the matching process we introduce a limit parameter L_{NE} representing the maximal number (the limit) of instances in which the corresponding weak image- and template binary test patterns are allowed to be different (*Not Equal*), i.e. the maximal number of weak binary test pattern rejections that is allowed before final overall rejection occurs. Thus, the current position of the sliding template window is added to the list of candidate matches if it is rejected by at most L_{NE} weak binary test patterns, or equivalently, if it passes at least $k \times l - L_{NE}$ weak binary test patterns. L_{NE} value range is [0, $k \times l$], when L_{NE} equals $k \times l$, the current scanned position will pass all the tests which reduces the proposed method to the traditional template matching.

In the second phase of the template matching method proposed here the overall best match is determined by applying conventional template matching to the set of candidate matches identified in the first phase.

In many practical situations like industrial and medical environments imagesareoften captured under stable illumination conditions. As a result, the difference between the mean grayscale value of the template and the mean of the corresponding target area in the scene will be small. This allows a further reduction of the computational effort: first compare the mean of the template image with the mean of current image window, and skip further inspection if this difference exceedsa prespecified intensity threshold T; else, test the current location further by computing the cascade of weak block binary test patterns.

The implementation of the template matching process proposed in this study involves the following steps:

1. Compute integral images for both the scene and template images.

2. Compute the set of weak image block binary test patterns $c_i^T, i \in \{1, k \times l\}$ for the template image.

3. Slide an inspection window (with the size of the template image) to the next position in the scene.

4. If the difference between the mean value of the local window and the mean value of the template exceeds a given threshold T go to step 3, else continue.

5. Calculate the weak image block binary test patterns $c_i, i \in \{1, k \times l\}$ over the window support and compare them to the corresponding weak template block binary test patterns, while the number of block rejections is less than or equal to a given limit L_{NE}.

6. If the total number of block rejections does not exceed L_{NE} assign a candidate status to the current image window, else reject the current window position as a candidate target location.

7. Continue at step 3 until all image pixels have been processed.

8. Determine the overall best match by applying conventional template matching to the set of candidate target locations.

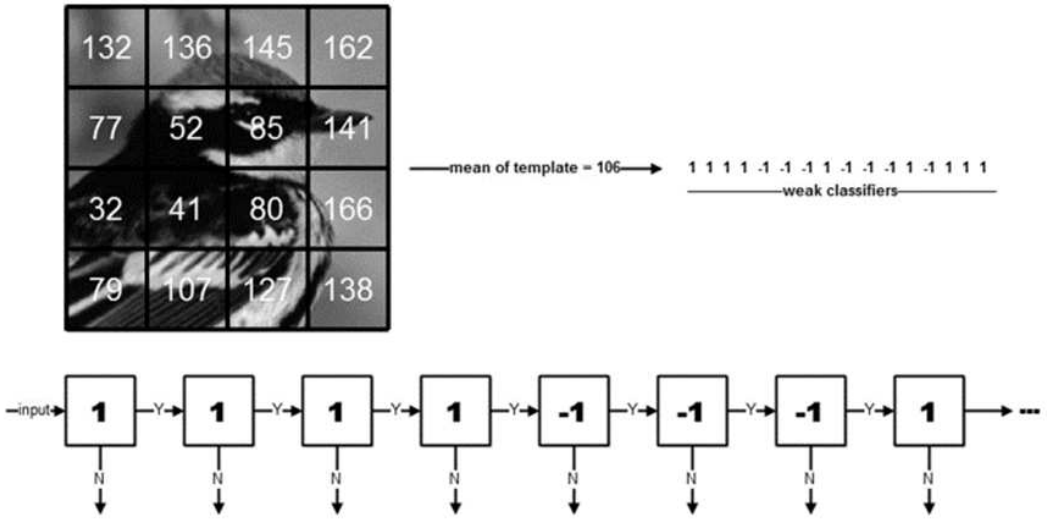


Fig. 3: The computation of a strong template binary test pattern as a cascade of weak block binary test patterns.

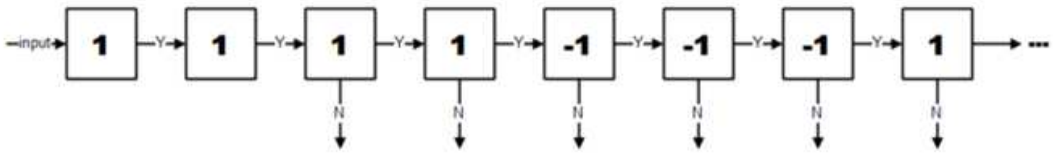


Fig. 4: Cascaded binary test pattern structure for $L_{NE} = 2$.

4. Experiments

In this section we investigate the sensitivity of the proposed template matching process to variations in the limit parameter L_{NE} representing the maximal allowed number of block rejections, and we investigate the efficiency of the new method by comparing its

performance relative to both a standard template matching technique and a state-of-the-art FFT-based method. The proposed method was implemented in C++ and not optimized. All computations were performed on a Pentium Dual-CoreE5800 3.2 Ghz computer with 2 GB RAM.

In all experiments, the template window was partitioned in 4×4 rectangular blocks. Note that very large block sizes would result in a large number of candidate matches, thereby increasing the computational effort in the second stage. For example, if the template window is partitioned in rectangular blocks of size 1×2, then there are only four possible patterns: [1, 1], [1, -1], [-1, 1], [-1, -1]. We did an experiment on the well-known Lena test image (512×512 pixels) to compute the fractionof candidate match locations for those four patterns. We set the window size to 64×32 pixels and block size to 32×32 pixels. Thefraction of candidate target locations for the four different patterns is respectively 0, 0.440481, 0.559505, and 1.39509×10^{-5} So, if the input template image has the pattern [1, -1] or [-1, 1], then only about 44% or 56% pixels areleft toevaluate in the second stage.In contrast, small block sizes make the classification process more sensitive to noise, resulting in the rejection of many good candidate matches. An extreme example is the case when the block size is 1 and $L_{NE} = 0$. In that case, all scanned positions must pass $k \times l$ tests. If a single pixel is corrupted by noise, the current scanned position will be rejected.

5. Sensitivity to the maximum allowed number of template rejections L_{NE}

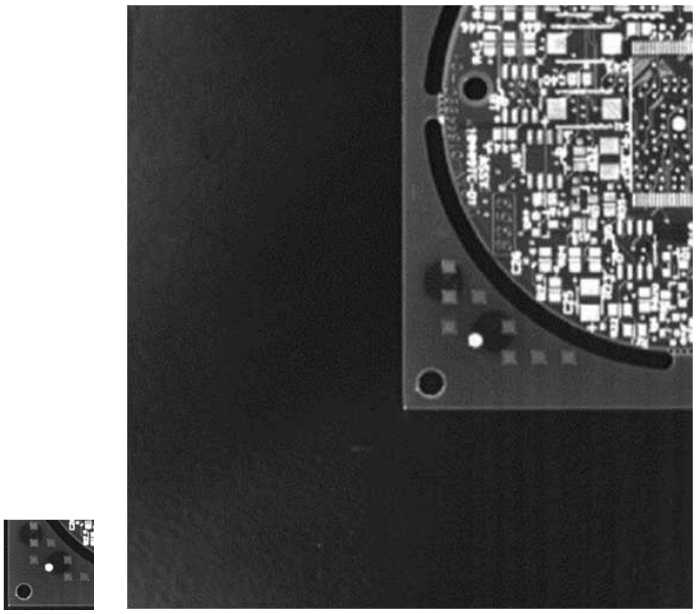
To investigate the selectivity of the strong template binary test patterns for different values of the parameter L_{NE} (the maximum number of allowed template block rejections) we computed the number of candidate target matches for a large number of different scenes and template images. We used 63 images varying in size from 2000×3000 to 3000×4000 pixels representing different natural scenes (e.g., landscapes, buildings etc.). From each image we cropped 100 patches of size 128×128 pixels centered on randomly selected corners (obtained with the Harris corner detector [27]), and we used these patches as templates in the computation of candidate matches. Centering the windows on corners avoids the selection of featureless regions, which would result in spurious false matches. The block size used in the computation of the weak binary test patterns was 4×4 (i.e. $k, l = 4$), and the intensity threshold was fixed at $T = 1$. Table 1 lists the results quantified as the fraction of the total number of image pixels that was classified as a candidate target location, for different values of the maximum allowed number of block rejections L_{NE}. Performance is quantified as the fraction of image pixels that are identified as candidate target locations. The results shown are the mean, standard deviation (SD), minimum, median, and maximum of these ratios over 6300 trials (i.e. for 63 different images and 100 different template regions each). These results show that the mean fraction of candidate target locations increases only slowly with the parameter , from 0.1% for $L_{NE} = 0$ (no block rejections allowed, the strongest binary test pattern) to about 6% for $L_{NE} = 7$ (7 rejections allowed on a total of $4 \times 4 = 16$ weak block binary test patterns). Hence, even when about half (7 out of 16) of the weak binary test patterns are rejected, the overall strong binary test patterns still filters out about 94% of non-target locations on average (i.e., only about 6% of the image pixels need to be processed in the second stage), thus yielding a significant overall reduction of the computational effort.The standard deviation of the fraction of candidate target locations also increases only slowly, indicating that the binary test patterns perform well in most cases.

Table 1: Performance of the weak block classification process.

L_{NE}	Mean	SD	Minimum	Median	Maximum
0	0.00101	0.00426	1.09E-07	1.15E-04	0.12162
1	0.0031	0.00857	4.16E-06	7.41E-04	0.17543
2	0.0069	0.013	1.93E-05	0.00288	0.23924
3	0.01271	0.01794	6.43E-05	0.00755	0.20386
4	0.02118	0.02303	2.08E-04	0.01517	0.19451
5	0.03356	0.03133	5.73E-04	0.02616	0.26578
6	0.04801	0.03905	6.59E-04	0.03998	0.30008
7	0.06402	0.04833	0.00142	0.05444	0.38331

6. Efficiency

To give an impression of computational efficiency of the proposed method, we compare its performance both with traditional template matching and with an FFT based template matching method (http://docs.opencv.org), implemented by the OpenCV function matchTemplate which has been optimized using Intel's SSE (Streaming SIMD Extensions: http://software.intel.com) instructions. Figure 5 shows a template image (Fig.5a) and the image from which it was cropped (Fig.5b).



(a) (b)

Fig. 5: (a) A template image and (b) the scene from which is was taken.

This template was matched both to the scene from which it was taken (Fig.6a) and to 9 other images created by cropping windows with different orientations and positions from the original scene (Fig.6b-j).

Table 2 lists the results of both template matching procedures for the 10 images shown in Figure 6.

Table 2: Performance of the proposed method relative to traditional and FFT based template matching.

	Position (x,y)			**Time (ms)**			**Speed-up relative to**	
Image	Actual	Traditional/ FFT	Proposed Method	Traditional	FFT	Proposed Method	Traditional	FFT
a	(223, 191)	(223, 191)	(223, 191)	1274.51	24.40	1.35	945.66	18.11
b	(204, 190)	(202, 203)	(201, 205)	1275.16	23.40	1.38	924.12	16.96
c	(147, 193)	(149, 182)	(147, 182)	1275.35	23.40	1.46	873.4	16.02
d	(84, 199)	**(12, 270)**	(97, 161)	1293.91	23.51	1.20	1074.54	19.53
e	(202, 241)	(203, 224)	(207, 221)	1293.09	23.50	1.44	895.29	16.27
f	(216, 244)	(211, 264)	(202, 259)	1287.68	23.42	1.51	854.75	15.55
g	(151, 243)	(153, 235)	(151, 234)	1292.35	23.46	1.59	810.47	14.71
h	(154, 297)	(150, 316)	(145, 317)	1281.57	23.36	1.57	815.91	14.87
i	(228, 293)	(227, 294)	(226, 293)	1278.37	23.60	1.59	801.51	14.8
j	(110, 320)	**(4, 288)**	(143, 265)	1293.53	23.65	0.94	1368.84	25.03

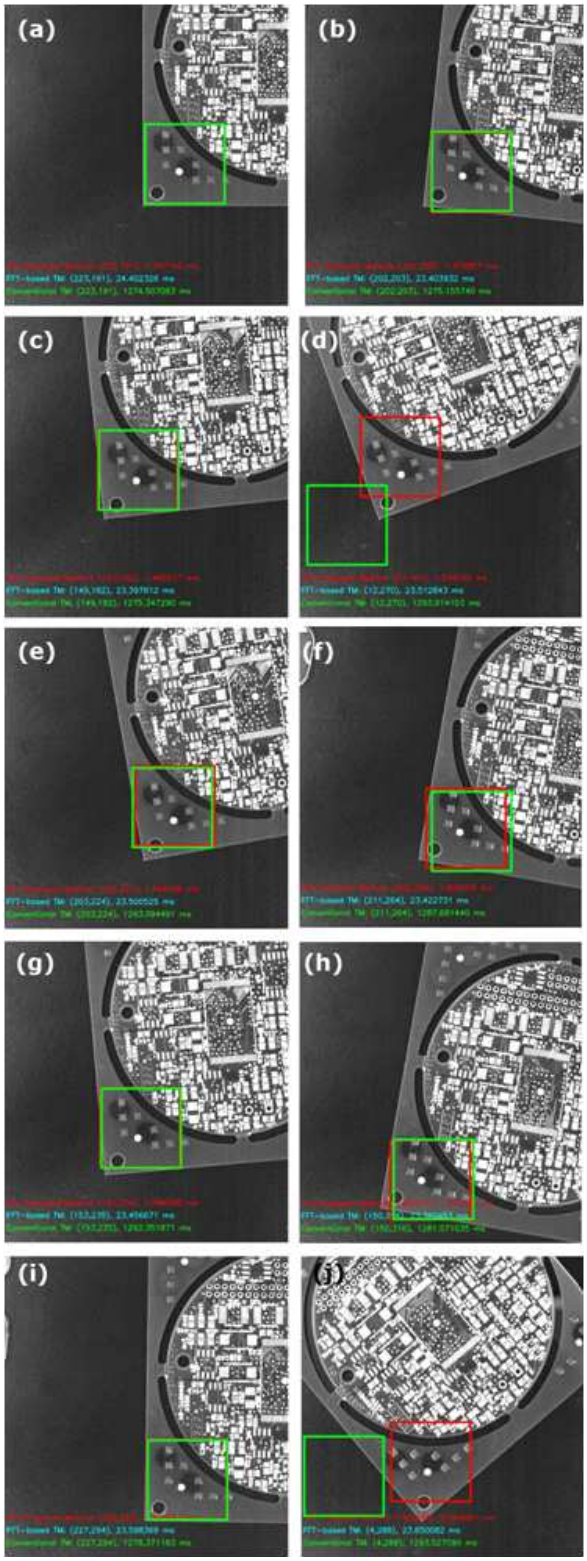


Fig. 6: Match results of the proposed method (red) compared to conventional (green) and FFT based (blue) template matching for the template from Fig.5a and for different orientations and sections of the original scene (Fig.5b). Note that conventional and FFT based template matching yield exactly the same template positions (i.e. the green and blue windows coincide).

The coordinates represent the position of the top-left corner of the template. Listed are the actual position coordinates, and the results of the three template matching methods investigated (traditional matching, FFT based matching, and the proposed method). Coordinates in bold indicate incorrect matches.These results show that restricting traditional template matching to a set of candidate target locations determined from integral image based weak block binary test patterns that are computed in a cascaded fashion can reduce computation time up to three orders of magnitude. Also, the proposed method yields a speed-up of at least one order of magnitude relative to the FFT based method. In some cases (Fig.6d and 6j) both traditional and FFT based template matching fail to find reasonable matches (when the scene image is rotated there is no longer an element wise correspondence to the template image that yields a global maximum of the NCC and local maxima at other locations will be taken as matches), while the new approach yieldsthe correct target position because it rejects these locations as false matches already in the first candidate target classification stage.

7. Conclusions

In this paper we proposed a new approach to speed up template matching. Integral images are used to efficiently compute local (image and template) binary test patterns in a cascaded and patchwise fashion, which allows early termination of the computational process if the initial estimates suggest that the local image region corresponds to a poor match. The present results show that the proposed method can provide up to three orders of magnitude speedup over the traditional computation of normalized cross correlation, and up to an order of magnitude improvement over a state-of-the-art FFT based method. Furthermore, it is relatively robust for local image distortions. A limitation of the proposed method is the fact that the parameter L_{NE} cannot be determined adaptively or based on the content of the input template image. We plan to address this issueina future study.

References

[1] S. Omachi and M. Omachi, Fast template matching with polynomials, *IEEE Transactions on Image Processing*, vol. 16(8), 2007, pp. 2139-2149.

[2] A. Del Bimbo and P. Pala, Visual image retrieval by elastic matching of user sketches, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(2), 1997, pp. 121-132.

[3] H. Peng, L. Fulmi, and C. Zheru, Document image recognition based on template matching of component block projections, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25(9), 2003, pp. 1188-1192.

[4] Y. Bentoutou, N. Taleb, K. Kpalma, and J. Ronsin, An Automatic Image Registration for Applications in Remote Sensing, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43(9), 2005, pp. 2127-2137.

[5] R.M. Dufour, E.L. Miller, and N.P. Galatsanos, Template matching based object recognition with unknown geometric parameters, *IEEE Transactions on Image Processing*, vol. 11(12), 2002, pp. 1385-1396.

[6] L.D. Stefano, M. Marchionni, and S. Mattoccia, A fast area-based stereo matching algorithm, *Image and Vision Computing*, vol. 22(12), 2004, pp. 983-1005.

[7] S.L. Kilthau, M.S. Drew, and T. Moller, Full search content independent block matching based on the fast Fourier transform, *Proceedings of the 2002 International Conference on Image Processing*, vol. I, IEEE, Piscataway, NJ, USA, 2002, pp. 669-672.

[8] A. Rosenfeld and G.J. Vanderburg, Coarse-fine template matching, *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-7(2), 1977, pp. 104-107.

[9] S.L. Tanimoto, Template matching in pyramids, *Computer Graphics and Image Processing*, vol. 16(4), 1981, pp. 356-369.

[10] P. Viola and M.J. Jones, Robust real-time face detection, *International Journal of Computer Vision*, vol. 57(2), 2004, pp. 137-154.
[11] J.P. Lewis, Fast template matching, *Vision Interface*, vol. 95(120123), 1995, pp. 15-19.
[12] K. Briechle and U.D. Hanebeck, Template matching using fast normalized cross correlation, *Optical Pattern Recognition XII*,vol. SPIE-4387, The International Society for Optical Engineering, Bellingham, WA, USA, 2001, pp. 95-102.
[13] H. Schweitzer, J.W. Bell, and F. Wu, Very fast template matching, *Computer Vision ECCV 2002*, Springer, Berlin Heidelberg, Germany, 2006, pp. 358-372.
[14] Y. Hel-Or and H. Hel-Or, Real-time pattern matching using projection kernels, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27(9), 2005, pp. 1430-1445.
[15] F. Tang and H. Tao, Fast multi-scale template matching using binary features, *Proceedings of the IEEE Workshop on Applications of Computer Vision WACV '07*, 2007, p. 36.
[16] G. Guo and C.R. Dyer, Patch-based image correlation with rapid filtering, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, IEEE, Piscataway, NJ, USA, 2007, pp. 1-6.
[17] F.C. Crow, Summed-area tables for texture mapping, *Proceedings of the 11th annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH '84)*,vol. 18, ACM, New York, NY, USA, 1984, pp. 207-212.
[18] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*,vol. I, IEEE, Piscataway, NJ, USA, 2001, pp. 511-518.
[19] O. Tuzel, F. Porikli, and P. Meer, Region covariance: A fast descriptor for detection and classification, *Computer Vision ECCV 2006*, vol. LNCS 3952, Part II, Springer-Verlag, Berlin Heidelberg, 2006, pp. 589-600.
[20] F. Porikli and O. Tuzel, Fast construction of covariance matrices for arbitrary size image windows, *Proceedings of the IEEE International Conference on Image Processing 2006*, IEEE, Piscataway, NJ, USA, 2006, pp. 1581-1584.
[21] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu, Shape and appearance context modeling, *Proceedings of the 11th International Conference on Computer Vision (ICCV 2007)*, IEEE, Piscataway, NJ, USA, 2007, pp. 1-8.
[22] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, Speeded-up robust features (SURF), *Computer Vision and Image Understanding*, vol. 110(3), 2008, pp. 346-359.
[23] C. Beleznai and H. Bischof, Fast human detection in crowded scenes by contour integration and local shape estimation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, IEEE, Piscataway, NJ, USA, 2009, pp. 2246-2253.
[24] C. Messom and A. Barczak, Stream processing of integral images for real-time object detection, *Proceedings of the Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2008)*, IEEE, Piscataway, NJ, 2008, pp. 405-412.
[25] O. Veksler, Fast variable window for stereo correspondence using integral images, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*,vol. I, IEEE, Piscataway, NJ, USA, 2003, pp. 556-561.
[26] F. Shafait, D. Keysers, and T.M. Breuel, Efficient implementation of local adaptive thresholding techniques using integral images, *Document Recognition and Retrieval XV*, vol. SPIE-6815, article id 681510, The International Society for Optical Engineering, Bellingham, WA, USA, 2008.
[27] C. Harris and M. Stephens, A combined corner and edge detector, *Proceedings of the Fourth Alvey Vision Conference*, 1988, pp. 147-151.