

Fast and Optimal Binary Template Matching Application to Manga Copyright Protection

Mathieu Delalandre¹

¹Laboratory of Computer Science
University of Tours, Tours city, France
mathieu.delalandre@univ-tours.fr

Motoi Iwata² and Koichi Kise²

²Graduate School of Engineering
Osaka Prefecture University, Osaka, Japan
lastname@cs.osakafu-u.ac.jp

Abstract—Template matching is a technique used in classifying an object by comparing portions of images with another image. Finding a given template in an image is typically performed by scanning the image and evaluating the similarity with the template. When the scanning is concerned with the entire image template matching is optimal. This paper considers a special case of template matching where the templates are binary. Although binary template matching has been studied extensively since the early days of pattern recognition, this technique seems not longer in use in Document Image Analysis (DIA). The major reasons are the time complexity, the no-invariance to scale and rotation and the lack of adaptability of similarity measures. However, different contributions have been investigated during the last years to improve these aspects: robustness and discrimination capability of similarity measures, their characterization, time-processing optimization with hardware support, etc. In this paper, we will review first some of the recent issues about binary template matching. We will present then a system exploiting bitwise operators and parallel processing supporting fast and accurate binary template matching for Manga copyright protection. This system is compared to a FFT-based template matching, and it outperforms both in processing-time and detection accuracy.

Keywords—binary template matching & similarity measures, optimal, FFT, bitwise operators, parallel processing

I. INTRODUCTION

Template matching is a technique used in classifying an object by comparing portions of images with another image. The traditional approach is interested with the distance-based comparison of an image with a template using the pixel intensity values. Finding a given template in an image is typically performed by scanning the image and evaluating the similarity between the template and an area. This approach is also known as area-based template matching in the literature or often called correlation-based template matching [1]. When the scanning is concerned with the entire image we're discussing about optimal or Full-Search (FS) template matching [2]. Template matching is concerned with different open problems that include the size, scale and position invariance, the selection of distance-based functions, time optimization, etc. These problems are linked to the method or algorithm selected, the type of template and the application use-case which is to be solved.

In this paper, we will discuss the application of template matching techniques to Document Image Analysis (DIA). An usual approach in DIA is to convert gray-level images to binary by thresholding. Here, binary template matching can be applied [3]. Although binary template matching has been

studied extensively since the early days of pattern recognition, this technique seems not longer in use in the DIA community. The major reasons are the no-invariance to scale and rotation, the lack of adaptability of similarity measures and the time-complexity. However, different contributions have been investigated during the last years to improve these aspects including the robustness and discrimination capability of similarity measures [4], [5], their characterization [5], [6], the time-processing optimization with hardware support [7], [8], etc.

This paper will present an extended discussion of [3] about the binary template matching problem. The paper in [3] is focussed on the generalization aspects of binary template matching. Here, binary template matching will be mainly discussed in terms of characterization of similarity measures, performance evaluation and optimization issues. In the rest of the paper we will introduce first in the section II the problem statement. The sections III and IV will provide a short review about binary similarity measures and fast optimal template matching methods. In the section V, we will present a system exploiting bitwise operators and parallel processing supporting fast and accurate binary template matching for Manga copyright protection. Conclusions and perspectives will be given in section VI.

II. PROBLEM STATEMENT

Template matching is conceptually a simple process. We need to match a template T to an image I , where the template is a sub-image that contains the shape we are trying to find. Formally the template matching can be defined as a method of parameter estimation. The template can be defined as discrete function $T_{x,y}$, taking values in a window W . That is, the coordinates of the points $(x, y) \in W$. The template matching problem is to choose the template position that minimizes a given similarity measure between the template T and the image I at a given position (i, j) in the image with $(i, j) \in I$. The Eq. (1) gives the general form of the minimization problem, using the sum of squared differences as similarity measure.

$$\min e = \sum_{(i,j) \in I} \sum_{(x,y) \in W} (I_{x+i,y+j} - T_{x,y})^2 \quad (1)$$

The template matching problem is concerned with different parameters

n the size of the template

$O(f(n))$ the computation cost of the similarity measure
 m the image size
 θ_k the orientation search parameter
 S_k the scale search parameter
 p the template number

The combination of the m, θ_k, S_k, p parameters is the search space, and total computation cost depends on the dimension of this search space and the complexity of the used similarity measure $O(f(n))$ with n the size of the template.

These parameters depend mainly on the kind of application use-case which is to be solved. When dealing with DIA-based applications, the m parameter is large due to the paper format and the high resolution used during the digitalization process. A typical size is 2480×3508 pixels when scanning a A4 page at 300 dpi. The size of template depends on the application use-case. Character recognition involves small templates [9] 32×32 pixels, whereas applications such as document recognition [10] look for main components in whole document images i.e. logos, tables, etc. Scalability is a critical issue in DIA, therefore the number of template p is usually very large e.g. 50 – 1000 component blocks in [10] or 1500 glyphs in [9]. Hopefully, severe search space reduction can be achieved by expediting the scale and rotation invariance aspects with suitable selections of DIA applications. Document recognition [10], OCR [10], etc. are not still concerned with the scale and rotation invariance. The templates to detect appear at fix orientation in the documents, and information about the scale parameters can be deducted from the document workflow (resolution parameters in file formats, paper format vs image size, etc.).

III. BINARY SIMILARITY MEASURES

In order to perform the comparison between a template and a candidate sub-image, a function measuring the degree of similarity (or dissimilarity) is computed. Estimation of image similarity is an important problem of image analysis. It can be used for image matching, noise reduction, image coding, etc. Many different similarity measures are documented in the literature. In the field of computer vision, correlation-based functions applied on gray-level images are often preferred for their performance and computational complexity [11]. When the images are obtained in a binary form, binary similarity functions can be applied [6].

Numerous binary similarity measures have been proposed in various fields (biology, ethnology, etc.). These binary measures require significantly less resource compared to the ones working in the gray domain [11]. The computation is performed in absence of square, multiplications, summing up operations and floating-point coding required while computing similarity between gray-level images. Let's define some general terms.

We consider $X = (x_1, \dots, x_m, \dots, x_n)$ and $Y = (y_1, \dots, y_m, \dots, y_n)$ as n -dimensional binary vectors where X is one of the template X_1, \dots, X_C and Y is the image to compare. These binary vectors are used within a $\delta_m(u, v)$ function Eq. (2) where $u, v = 0, 1$ and y_m, x_m are the m_{th} elements of Y and X respectively. We define then in Eq. (3) $n_{u,v}$ as the number of occurrences where $x_m = u$ and $y_m = v \forall m$. The term n_{11} denotes the positive matches, i.e. the number

Measure	$S(X, Y)$	Range
Inner Product (<i>IP</i>)	n_{11}	$[0, +\infty[$
Jaccard and Needham (<i>Jaccard</i>)	$\frac{n_{11}}{n_{11} + n_{10} + n_{01}}$	$[0, 1]$
Dice (<i>DICE</i>)	$\frac{n_{11}}{2n_{11} + n_{10} + n_{01}}$	$[0, 0.5]$
Russel-Rao (<i>RUSS</i>)	$\frac{n_{11}}{n}$	$[0, 1]$
Kulzinsky (<i>KUL</i>)	$\frac{n_{11}}{n_{10} + n_{01}}$	$[0, +\infty[$
Hamming (<i>Hamming</i>)	$n_{11} + n_{00}$	$[0, +\infty[$
Sokal and Michner (<i>SM</i>)	$\frac{n_{11} + n_{00}}{n}$	$[0, 1]$
Rogers and Tanimoto (<i>RT</i>)	$\frac{n_{11} + n_{00}}{n_{11} + n_{00} + 2(n_{10} + n_{01})}$	$[0, 1]$
Correlation (<i>CORR</i>)	$\frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_{1x}n_{0x}n_{x1}n_{x0}}}$	$[-1, 1]$
Yule and Kendall (<i>Yule</i>)	$\frac{n_{11}n_{00} - n_{10}n_{01}}{n_{11}n_{00} + n_{10}n_{01}}$	$[-1, 1]$

Table I. COMMON BINARY SIMILARITY MEASURES IN DIA

Where $n_{ix} = n_{i0} + n_{i1}$, $n_{xi} = n_{0i} + n_{1i}$, $n = n_{ix} + n_{xi}$

of 1 bits that match between y_m and x_m . The term n_{00} is the negative matches, i.e. the number of 0 matching bits. The terms n_{10}, n_{01} denote the number of bit mismatches - the first where pattern x_m has a 1 and pattern y_m has a 0, and vice-versa.

$$\delta_m(u, v) = \begin{cases} 1 & \text{if } x_m = u \text{ and } y_m = v \\ 0 & \text{if otherwise} \end{cases} \quad (2)$$

$$n_{u,v} = \sum_{m=1}^n \delta_m(u, v) \quad (3)$$

Based on $n_{00}, n_{11}, n_{10}, n_{01}$, numerous binary similarity measures can be defined to evaluate the similarity $S(X, Y)$ between the X and Y binary vectors. In [6], the authors present a comprehensive survey where 76 binary measures used over the last century in various fields are presented and analysed through hierarchical clustering. The authors conclude about some close relationships among several measures. In the DIA field, the common used measures [3], [4], [5] are the *IP*, *Jaccard*, *DICE*, *RUSS*, *KUL*, *Hamming*, *SM*, *RT*, *CORR* and the *Yule* ones (Table I). Taxonomy of the measures is discussed in these different papers. The first major division is between the Inner-Product (*IP*) based similarity measures that focusses mainly on positive matches n_{11} (*IP*, *Jaccard*, *DICE*, *RUSS*, *KUL*) and those that credit both positive and negative matches n_{11}, n_{00} (*Hamming*, *SM*, *RT*, *CORR*, *Yule*). Next, those that consider both positive and negative matches are further categorized into the additive or ‘‘Hamming look-like distance’’ (*Hamming*, *SM*, *RT*) and the multiplicative form (*CORR*, *Yule*). The hamming distance is the binary form of the L_p norm distance commonly used for template matching with gray-level images. The binary form of the correlation is (*CORR*), however Yule and Kendall suggested a similar measure (*Yule*) straightforward to compute.

To further improve discrimination capability of binary measures, weights can be applied [5]. Here, the goal is to improve performance by optimizing the similarity measure rather than distance. The weights can be obtained using an optimization method such as genetic algorithms, or assigned to matches according to their relative frequency of occurrence

[3]. The typical similarity measures with weighting are the weighted inner product S_{WIP} or the weighted Hamming D_{WH} . The weighted Hamming D_{WH} is given in Eq. (4), where w_m is a weight describing probability of being correct when the m feature is used, with $\sum_{m=1}^n w_m = 1$. An alternative is to distinguish the match cases, by applying different weights. The authors in [5] propose like this the S_{WAZZO} measure given in Eq. (5), where w_m^+ , w_m^- are the weights applied to the positive and negative matches respectively with $\sum_{m=1}^n w_m^+ + w_m^- = 1$.

$$D_{WH} = \sum_{m=1}^n w_m (\delta_m(1, 0) + \delta_m(0, 1)) \quad (4)$$

$$S_{WAZZO} = \sum_{m=1}^n w_m^+ \delta_m(1, 1) + w_m^- \delta_m(0, 0) \quad (5)$$

These weights can be obtained using an optimization method such as genetic algorithms or assigned to matches according to their relative frequency of occurrence [3]. In [3], a weight $w_m = p_m(k|u)$ Eq. (6) denotes the conditional probability of a template class k , given that $x_{km} = u$ for $u = 0, 1$, and $X_k = (x_{k1}, \dots, x_{km}, \dots, x_{kn})$. These probability can be computed from the \mathbb{C} known template, $X_1, \dots, X_{\mathbb{C}}$ as detailed in Eq. (7).

$$w_m = p_m(k|u) = Pr[class = k|m = u] \quad (6)$$

$$p_m(k|1) = x_{km} / \sum_{l=1, l \neq k}^{\mathbb{C}} x_{lm} \quad (7)$$

$$p_m(k|0) = (1 - x_{km}) / \left[\mathbb{C} - \sum_{l=1, l \neq k}^{\mathbb{C}} x_{lm} \right]$$

Another possibility proposed in [3], [4] is to apply a global weighting in the computation of match cases n_{uv} as detailed Eq. (8). This is straightforward to compute as a single multiplication is needed following the $\delta_m(u, v)$ summing. This approach has been investigated on several binary similarity measures in [4], to evaluate the unequal importance of zero matches n_{00} and one matches n_{11} . The proposition of weighting zero matches resulted from the observation that a bit 0 provides less information of separability.

$$n_{uv} = w \sum_{m=1}^n \delta_m(u, v) \quad (8)$$

Performance evaluation of binary similarity measures in DIA has been reported for OCR [5], [3] and handwriting recognition [4] mainly. The presented results depend on the considered applications and cannot be totally conclusive. In the case of OCR [5], [3], it tends to prove that better performances can be obtained with weighting. This conclusion is also reported in [4] for handwriting recognition, where weighting of the zero matches is presented as very effective for boosting classification performance. However, weighting can disturb in a template matching context, as the measure appears

Method	Distances	weight	Complexity	Coding	Hardware
Brute-force	all	yes	$O(MNmn)$	byte	register operations, pipeline & multi-core architectures
FFT	based on n_{00}, n_{11}	no	$O(M^2 \log_2 M)$	float	multi-core architectures

Table II. COMPARISON OF OPTIMAL TEMPLATE MATCHING METHODS

MN and mn are the image and template sizes respectively
 $M > N$ and $MN \gg mn$

more sensitive to the miss-alignments [3]. In the other side, the selection of a similarity measures appears as application-dependent. The Hamming look-like distances present better properties for OCR, as they outperform the multiplicative-based (i.e. correlation) similarity measures [5], [3]. The conclusion is different in [4], where close performances are observed between some *IP*, *Hamming* and *CORR* based similarity measures for handwriting recognition.

IV. FAST OPTIMAL BINARY TEMPLATE MATCHING

The typical process for template matching is performed by scanning the entire image and evaluating the similarity between the pattern and an area. This is known as the Full-Search (FS) or optimal template matching¹ [2]. The brute-force method for optimal template matching presents a quadratic complexity of $O(MNmn)$ with MN and mn the image and template sizes respectively (Table II). Due to the high complexity and time consuming requirements of this task, optimization must be employed. Two main approaches can be considered and combined, using fast search algorithms (e.g. FFT² based template matching) and/or hardware support (e.g. register operations, parallel processing with pipeline or multi-core architectures).

Optimization can be achieved by expediting the brute-force method using FFT based template matching [12]. Indeed, when dealing with the L_p norm based similarity measures the template matching can be achieved through a correlation operator. It is standard result to reformulate Eq. (1) into Eq. (9) with I the image, (i, j) a given position in I , $T_{x,y}$ the template taking values in a window W with $(x, y) \in W$. With binary images, the usual approach is to shift the binary values from $(0, 1)$ to $(-1, 1)$ to get the hamming distance, while no modification results into the inner product. That is, the FFT based binary template matching is restricted to distances employing positive and negative matches only, and cannot support weighting. In addition, the computation is shifted to floating-point coding. As multiplication is dual to convolution in the Fourier domain, we can express the convolution in term of correlation Eq. (10). Template correlation $I \otimes T$ can be then implemented by multiplying the Fourier Transforms \mathfrak{F} of the flipped template T' and the image to compare, with $T' = T_{-x, -y}$. The result needs to be inverse transformed \mathfrak{F}^{-1} to return the picture domain.

$$\max e = \sum_{(i,j) \in I} \sum_{(x,y) \in W} I_{x+i,y+j} \times T_{x,y} \quad (9)$$

$$I \otimes T = I \star T' = \mathfrak{F}^{-1}(\mathfrak{F}(I) \times \mathfrak{F}(T')) \quad (10)$$

¹ Approximated search methods can be also employed, we will not discussed these methods here out of the scope of this paper.

² Fast Fourier Transform

FFT algorithms can rapidly compute the Fourier Transform. However, their implementation require that I and F be extended with zeros to a common power of two. This constraint to use square matrixes of same dimension $M \times M$, even though the template's size m, n and image's width N are usually smaller than M . The complexity of the transform computation is then $12M^2 \log_2 M$ real multiplications and $12M^2 \log_2 M$ real additions [12]. Once the FFT computed, additional steps for the frequency domain multiplication Eq. (10) and the maximum value search on the picture domain, are needed. These require M^2 additional multiplications. The total amount of operations become $M^2 + 48M^2 \log_2 M$, usually approximated as $M^2 \log_2 M$. To gain time, the FFT of templates to compare can be computed off-line during a training stage. FFT of the image and the inverse transform must compute online. The FFT becomes more efficient when M, N, m, n are large. The computation of the FFT can be supported by parallel implementation that can achieve acceleration factors of 3 to 7 using GPU [8], but raising other constraints linked to application portability.

Another possibility to perform fast optimal binary template matching is to deal with hardware support for the brute-force method. Comparison of binary templates can be achieved first with array encoding to push into the CPUs' registers. Comparing two arrays can be performed extremely quickly on modern CPUs, with parallel bitwise operators being part of the SSE³ instruction set. Population count instructions to get number of bits set to 1 in an array are also standard on modern CPUs. For a 256-bit code, a 2.0 Ghz CPU is able to compare 120 millions of pairs per second. A reduction of the brute force complexity $O(MNmn)$ is then obtained in $O(MNr)$ with $r = m, n$. The array size is limited to 256-bits within the recent architectures, but processors are designed to support 512 or 1024 bits in the future.

The parallelism can also be trivially extended at the multiprocessor level, with each processor in charge of an image part. Indeed, the brute-force approach takes part of the BLAS⁴-like functions that can be supported easily by parallel processing compared to FFT computation [8]. Time processing scales them in a linear fashion according to the degree of parallelism. We could notice that the general trend in processor development is moving standard architecture to octo-cores to ones with tens or even hundreds of cores in a "near" future, raising strong acceleration capabilities for this approach on standard computers. Another alternative is the pipeline architectures. Such an approach has been investigated in [7] where a pipelined implementation for binary template matching, which fits into a finite state machine, is presented. The author reports a speed factor of 100 comparing to standard conventional pipelined architecture, making the system suitable for some real-time applications (e.g. to match a 27×35 template(s) at video rate). However, this approach raises strong portability problems, as the used hardware appears too specific.

V. A SYSTEM FOR MANGA COPYRIGHT PROTECTION

In this section we will discuss about an application of binary template matching for Manga copyright protection.

Manga, especially in Japan, covers a large part of the publishing market e.g. about 4 billions \$ in 2011. Manga has also gained a significant worldwide audience. However, sales of manga magazines (the traditional delivery medium for Manga in Japan) peaked in 1995, and have been falling ever since. Analysts blame the lack of hit titles, one of the major reason resulting of this situation is piracy.

Piracy is interested to getting copies of mangas, possibly before it goes on sale officially. The usual process is to purchase versions of mangas in a print form, to scan and to preprocess them (e.g. border or brightness correction). At last, camera-ready versions of illegal copies are put online on sites that can contain thousands of pages of copyrighted work. The finished products look like officially published volumes and serve to start business operations. Online Manga piracy reached the point where publishers could no longer turn a blind eye to other people profiting off their work. Therefore, there were great interest in the last recent years in protecting copyrights of Manga [13].

The problem of Manga copyright protection is completely related to template matching. It can be formulated as the comparison of templates coming from legal copies to pages coming from illegal copies. The major issues here are related to the digitalization noise (e.g. quality of paper, used scanners, image shifting and skew, etc.) and artifacts resulting of image pre-processing and compression. The Fig. 1 compares a legal (top) and an illegal (bottom) manga copy, where skew and brightness variations (a) and compression and low resolution artifacts (b) appear.

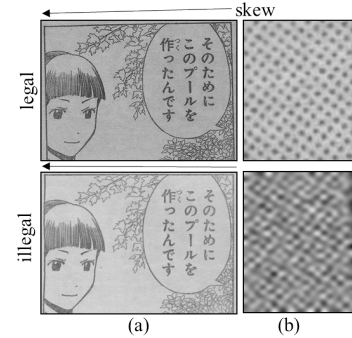


Figure 1. Comparison of a legal (top) and an illegal (bottom) manga copy (a) skew and brightness variations (b) compression and low resolution artifacts

To design our system, based on our state of the art we have noticed that the FFT based template matching seems not to be a good candidate. Indeed, it constraints the image size, bounds the selection of similarity measures, cannot support weighting, offers a limited gain in term of time processing when applied to document images and cannot be supported fully with hardware optimization. Therefore, we have decided to use a brute-force method with hardware support. Rather than a pipeline hardware support raising portability problems [7], we have taken benefit from recent advances of modern CPUs. Our approach exploits bitwise operators at array level to speed-up the matching process and multi-core processing.

The general architecture used for our application is presented in Fig. (2). A web crawler (1) collects manga images across the Web and stored them into an illegal copy database

³Streaming SIMD Extensions

⁴Basic Linear Algebra Subprograms

to be submitted to the copyright verification system. When stored, the images are converted to gray-level and re-sampled. We have fixed the standard resolution in the system to 130 dpi, as it appears to be common on the Web for the diffusion of illegal copies of manga magazines. At 130 dpi, the mean size of a manga magazine page is around 1300×900 pixels.

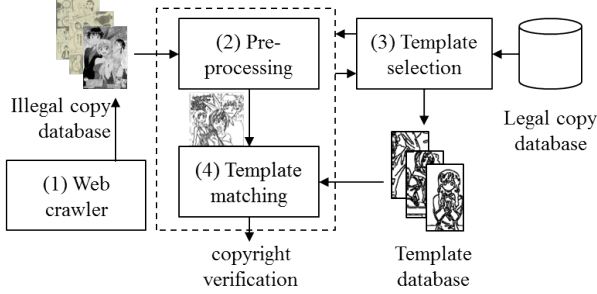


Figure 2. Copyright protection system of manga with Template Matching

The pre-processing stage (2) applies a Canny edge detection and a morphological dilatation. Mangas are pure line drawing images, but in their initial form they include textured and filled-shape regions. The Canny edge detector is a good candidate here to separate the line drawing layer from background. However, applying a canny edge detection results into thin edges (1-pixel width). That is, the morphological dilatation aims to re-equilibrate the background and foreground elements for matching. In addition, with a previous dilatation the template matching becomes less sensitive to small deviations between edges. The pre-processing serves both for illegal and legal manga images. On legal manga images, a template selection (3) is applied next to extract randomly templates of size 256×128 from complete images. These templates are stored in a template database and serve as page signature for copyright verification.

In a final step binary template matching (4) is applied. Before any matching, a 256-bit encoding makes the images ready for direct comparison with templates through bitwise operators. At every pixel location $(i, j) \in I$ the image, we encode a 256-bits length vector $Y_i = (y_{i,j-255}, \dots, y_{i,m}, \dots, y_{i,j})$ with $y_{i,m}$ the element of Y_i at position (i, m) and $m \in [j - 255, j]$. We are applying here a vertical encoding as it is more suitable to catch the content of mangas mainly composed of characters. This step is straightforward to compute, as it requires a single scan of the whole image where a 256-bits array is shifted at every position. The images of illegal copy database are encoded on the full pages whereas the 256×128 templates are encoded on the last line resulting in 128 vectors $Y_i = (y_{i,j-255}, \dots, y_{i,m}, \dots, y_{i,j})$ of 256-bits length each, with $i \in [0, 127]$. The templates are encoded off-line to gain processing time and stored in the template database.

The Fig. (3) gives evaluation results in term of processing time for matching using our approach. The tests have been performed on a Windows Seven Operating System based on an Intel CPU 2.1 GHz - quad-core - 8 threads. Comparing two templates requires some μs only and scales in a linear way depending of the templates' width (a). Indeed, comparison of 256-bit codes (i.e. the template columns) is done in $O(1)$ using intrinsic instructions for call of bitwise operators and population counting.

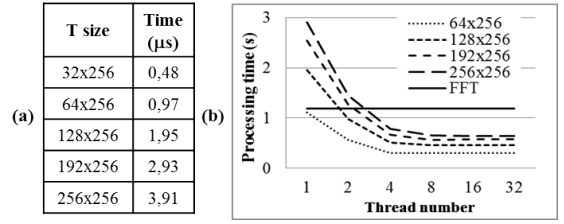


Figure 3. Processing time evaluation for template matching (a) template to template (b) template to image

When the matching is applied on a full page, the processing time shifts to seconds as the page is composed of more than a million of pixels. To support the full matching we have exploited parallel processing possibilities offered by the multicore architectures. To do so, the complete images to be submitted for copyright verification are divided into $N \times N$ tiles, each tile is then processed by a specific thread. The different thread results are synchronized by a main thread in charge CPU and core scheduling and memory sharing issues.

The Fig. (3) (b) gives the mean processing times for a complete matching against the thread number. The minimum is reached when meeting the hardware specification; 8 threads in our case. The matching is achieved in the range 295 to 640 ms for templates of size 256×64 to 256×256 respectively. This represents a speeding factor from 3.7 to 20.9 compared to the single thread processing. We have compared this approach to FFT computation based on the *Fast Hartley Transform* (FHT). This computation has been done as described in section IV, where the FHT of flipped templates has been done off-line. Using our approach, we have obtained an acceleration factor of 1.8 to 4 compared to FFT for templates of sizes 256×256 to 256×64 .

In a second step we have evaluated the discrimination capability of binary similarity measures for Manga copyright protection. Such an evaluation is usually conducted in the literature as an inter/intra-class discrimination problem [3], [5], [4]. We have such a protocol using a database composed of legal and an illegal copy of a 404 pages of a Manga magazine.

For the inter-class discrimination, we have followed the protocol defined in [3]. We have extracted randomly 5000 templates from the legal copy and matched all of them, resulting in 25 millions of comparisons. For all these comparisons, we have computed the $S_{ref}, S_{max}, S_{mean}, S_{min}$ values as detailed in Eq. (11) (12). We define in addition $S_{rm} = S_{ref} - S_{max}$ as the distance that template l is to its nearest false identification.

$$S(X_l, X_k) = d_{lk} \quad S_{ref} = d_{ll} \quad S_{max} = \max_{k \neq l} d_{lk} \quad (11)$$

$$S_{mean} = \frac{1}{\sum_{k \neq l} 1} \sum_{k \neq l} d_{lk} \quad S_{min} = \min_{k \neq l} d_{lk} \quad (12)$$

We have conducted this analysis on the main measures of the literature described in Table I. The results are reported in the Fig. (4) where we have normalized the graphics representation according to the ranges of measures⁵. For the

⁵Apart for the *KUL* measures as $S_{ref} \rightarrow \infty$

$SM, RT, CORR, Yule$ measures, as suggested in [4] we have weighted the zero matches n_{00} to obtain an equal credit with the one matches n_{11} . This has resulted in a strong improvement of the S_{rm} values mainly for the $RT, YULE, CORR$ measures (e.g. from 0.42 to 0.63 for the RT measures). Despite this improvement, the best S_{rm} is obtained by (i) the $JACCARD$ measure $0.73 = 1 - 0.27$ followed by (ii) the RT and $DICE$ measures presenting close performances. This tends to prove that the mismatches n_{10}, n_{01} and weighting are very effective for boosting the matching performances. As in [5], [3], we can notice that the additive forms of similarity measures $JACCARD, RT, DICE$ outperform the multiplicative forms $CORR, YULE$, despite the weighting. In addition, the $SM, RUSS$ measures corresponding to a normalized hamming distance and inner product that can be obtained with FFT offer poor discrimination results.

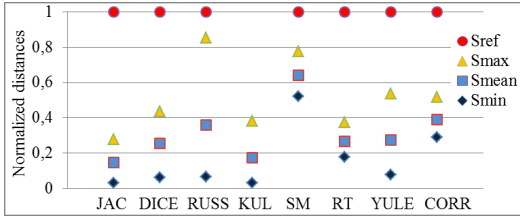


Figure 4. Inter-class evaluation for manga copyright protection

In a second step, we have investigated the intra-class discrimination. We have extracted randomly a template in each of the 404 pages of the legal copy and matched it with the corresponding whole page of the illegal copy, using the $JACCARD$ similarity measure. In addition, for each matching we have computed the localisation error ε (in pixels) between the coordinates of the template in the legal copy and the matching result in the illegal copy. The Fig. (5) gives the results of the $JACCARD$ similarity measure against the localisation error ε . We have obtained a min, mean and max distances respectively of 0.25, 0.61 and 0.80 with a mean localisation error ε of 21 pixels. Considering the border corrections done for digitalization, this localisation error enters a normal range. Based on the $S_{max} = 0.27$ value reported in the inter-class experiments Fig. 4, we can conclude that using the $JACCARD$ similarity measure the discrimination problem here is almost separable. On the complete magazine, we have noticed a single matching error; that is; a detection accuracy for copyright verification of 99.75%.

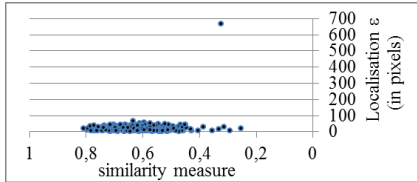


Figure 5. Intra-class evaluation for manga copyright protection

VI. CONCLUSION

In this paper we have proposed first a review about binary similarity measures and optimal fast template matching methods. We have presented then a system exploiting bitwise operators and parallel processing supporting fast and accurate op-

timal binary template matching, with an application to Manga copyright protection. Compared to the FFT based template matching, this system outperforms both in processing time and detection accuracy. Our experiments report acceleration factors from 1.8 to 4 compared to FFT. In addition the $SM, RUSS$ similarity measures, that can be computed through the FFT, offer poor discrimination results. Application of weighting in the similarity measures, that cannot be done with the FFT, results too in strong improvements in term of discrimination. Our experiments highlight that with a suitable selection of the similarity measure and proper weighting, the discrimination problem for Manga copyright protection is almost separable. The additive forms of similarity measures that take into account matches and mismatches ($JACCARD, RT, DICE$) obtain the better discrimination performances with equal credit weighting.

VII. ACKNOWLEDGEMENTS

This work has been supported by the JSPS short-term fellowships program FY2013 - ID NO. S13059. The authors wish to thank Atsushi Ito and Eiki Imazu (Graduate School of Engineering, Osaka Prefecture University, Osaka, Japan) for their participation to this work.

REFERENCES

- [1] M. Storrang and T. Moeslund, "An introduction to template matching," Computer Vision and Media Technology Laboratory (CVMT), Aalborg University, Denmark., Tech. Rep. 01-04, 2001.
- [2] M. Gharavi-Alkhansari, "A fast globally optimal algorithm for template matching using low-resolution pruning," *Transactions on Image Processing (TIP)*, vol. 10, no. 4, pp. 526–533, 2001.
- [3] J. Tubbs, "A note on binary template matching," *Pattern Recognition (PR)*, vol. 22, no. 4, pp. 359–365, 1989.
- [4] B. Zhang and S. Srihari, "Properties of binary vector dissimilarity measures," in *International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICCV)*, 2003.
- [5] S. Cha, C. Tappert, and S. Yoon, "Enhancing binary feature vector similarity measures," *Journal of Pattern Recognition Research (JPRR)*, vol. 1, pp. 63–77, 2006.
- [6] S. Choi, S. Cha, and C. Tappert, "A survey of binary similarity and distance measures," *Journal of Systemics, Cybernetics and Informatics (SCI)*, vol. 8, no. 1, pp. 43–48, 2010.
- [7] F. Waltz, "Application of skips to binary correlation," in *Conference on Machine Vision Applications, Architectures, and Systems Integration*, ser. SPIE Proceedings, vol. 2597, 1995.
- [8] A. Ali, L. Johnsson, and J. Subhlok, "Scheduling fft computation on smp and multicore systems," in *International Conference on Supercomputing (ICS)*. ACM, 2007, pp. 293–301.
- [9] A. Negi, K. Shanker, and C. Cherredhi, "Localization and extraction of text in telugu document images," in *International Conference on Document Image Analysis and Recognition (ICDAR)*, vol. 2, 2003, pp. 749–752.
- [10] H. Peng and F. L. and Z. Chi, "Document image recognition based on template matching of component block projections," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, no. 9, pp. 1188–1192, 2003.
- [11] N. Roma, J. Victor, and J. Tome, "A comparative analysis of cross-correlation matching algorithms using a pyramidal resolution approach," in *Workshop on Empirical Evaluation Methods in Computer Vision (ECCV)*, ser. Machine Perception and Artificial Intelligence, vol. 50. World Scientific, 2002, pp. 117–142.
- [12] J. Lewis, "Fast template matching," in *Vision Interface*, 1995, pp. 120–123.
- [13] S. Weihaan and K. Kise, "Detection of exact and similar partial copies for copyright protection of manga," *International Journal on Document Analysis and Recognition (IJDA)*, pp. 1–19, 2013.