

Fast Affine Template Matching over Galois Field

Chao Zhang
zhang@scv.cis.iwate-u.ac.jp

Graduate School of Engineering
Iwate University
Iwate, Japan

Takuya Akashi
akashi@iwate-u.ac.jp

Faculty of Engineering
Iwate University
Iwate, Japan

Abstract

In this paper, we address the problem of template matching under affine transformations with general images. Our approach is to search an approximate affine transformation over a binary Galois field. The benefit is that we can avoid matching with huge amount of potential transformations, because they are discretely sampled. However, a Galois field of affine transformation can still be impractical for exhaustive searching. To approach the optimum solution efficiently, we introduce a level-wise adaptive sampling (LAS) method under genetic algorithm framework. In LAS, individuals converge to the global optimum depending on a level-wise selection and crossover while the population number is decreased by a population bounding scheme. In the experiment section, we analyse our method systematically and compare it against the state-of-the-art method on an evaluation data set. The results show that our method has a higher accuracy performance with fewer matching tests.

1 Introduction

In this paper, we consider the problem of template matching under arbitrary 2D affine transformations. Template matching is a classical computer vision problem which aims to find a global optimum area in the target image (i.e. source image) according to the hint provided by a rectangular template. In affine template matching, each candidate affine transformation corresponds to a candidate area in the target image. Each candidate area is a parallelogram because of the properties of affine transformations. We only use gray scale information of images as the hint which is quantified by sum of absolute difference (SAD).

Recently, feature-based matching methods like SIFT and its variants are very efficient to estimate the 2D transformation matrix between template and target image. Only a few correctly matched key points are required for solving a system of linear equations. With matching results which contain outliers, we can also use method like RANSAC [1] to estimate the correct transformation matrix. Feature-based methods depend on the assumption that the key point matching results consist of inliers, there also exist images in which key points are hard to be detected like blur images, texture-less images, etc. Key points may

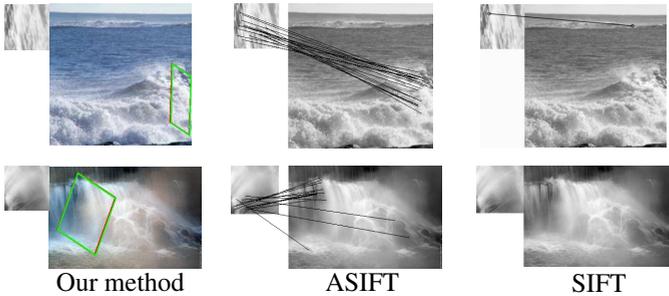


Figure 1: Our matching result (represented by green parallelogram) completely covers the ground truth area (represented by red parallelogram) in both cases. Affine-SIFT (ASIFT) can well handle affine transformation in the case when template has strong features (upper), while mismatches in the case when template has weak features (lower). Common SIFT can not handle affine transformation well.

also be mismatched heavily under the influence of noise, illumination changes, etc. A common template matching method is usually considered to be effective against such special situations. Figure 1 shows two matching examples respectively when a template has strong features and weak features.

As we all know, template matching potentially requires a huge number of samples in order to ensure the global optimum solution can be obtained. Especially for affine template matching, the number of candidate transformations increases exponentially when more accurate solution is required to be obtained, because scaling, rotation and shear are taken into account additionally compared with common template matching. Matching with numerous candidate solutions is ineffective and not practical. In fact, it is possible to estimate only a small fraction of candidate solutions in order to solve the optimum solution if the following assumption is made: a template is smooth. Under this assumption, SAD will not change much around the ground truth area of a target image. This assumption provides chances for developing more efficient matching methods. At the same time, such methods can not guarantee the accuracy with highly textured template.

The rest of this paper is structured as follows. In Section 2, we survey template matching methods with transformations and the efforts that have been done on solving affine template matching problem. In Section 3, we introduce our method from two perspectives: 1) construction of Galois field. 2) level-wise adaptive sampling method over Galois field. In Section 4, we investigate the effects of tunable parameters and compare our method against the state-of-the-art method [10]. Finally, we conclude this paper in Section 5.

2 Related Works

In this section, we mainly survey previous works on template matching involving geometric transformations. Despite the feature-based matching methods like SIFT [10], ASIFT [14], direct methods also have been widely studied. A common direct template method only involves the translation in x-axis and y-axis, thus the degree of freedom (DOF) is simply two. However, many applications require methods to be robust with varied transformations.

Rotation and translation: Same with common template matching, target area in target image is still rectangular. The difference is, it is rotated and repositioned by translation. The

DOF in this situation is three. Choi and Kim [10] proposed a method combining the projection method and Zernike moments in two stages. Candidates with low cost feature extracted are selected at first stage, and rotation invariant matching is performed at second stage. Fredriksson *et al.* [11] used string matching technique to deal with rotation.

Rotation, translation and scaling: In this situation, scaling is additionally involved in the matching problem, thus the DOF grows to five. The number of candidate areas becomes large and it is no longer practical for exhaustive searching. To accelerate matching procedure, Kim *et al.* [12] applied cascaded filters to exclude areas which have low probability to be selected as the final result. Akashi *et al.* [13] treated template matching as an optimization problem under genetic algorithm (GA) framework and applied their method into real-time eye detection by inheriting previous frame’s matching result to the next. GA can evolutionarily select “promising” candidate areas to evaluate, thus can avoid exhaustive searching.

Affine transformation: Despite basic Euclidean transformations, shear and scaling are involved additionally. The DOF then grows to six. To the best of our knowledge, few direct methods have been proposed under this situation compared with aforementioned two situations as a result of the broad search space. In [14], the state-of-the-art work is proposed which matches template in a very sparse way under the smooth assumption. In this paper, a discrete sampling net is constructed depending on an accuracy parameter, after that, a branch-and-bound scheme is employed to search an approximate solution over the net. The basic idea of this paper is to rule out a large portion of “unpromising” candidate transformations and focus on estimating the ones which are close to the ground truth. However, branch-and-bound scheme is still exhaustive to a certain extent, because the number of candidate transformations need to be estimated grows rapidly with the increase of expected accuracy. Insufficient samplings will lead to a totally different result. On the other hand, our method constructs a Galois field instead of a sampling net, and employs adaptive sampling to approach the ground truth from the perspective of optimization algorithm. Additionally, [15] provides a comprehensive comparison of affine covariant region detectors.

3 Methodology

3.1 Problem Description

Two grayscale images I_1 and I_2 are given as the input with each pixel’s gray value normalized to $[0, 1]$. I_1 is defined as a template image with size of $n_1 \times n_1$. I_2 is defined as a target image with size of $n_2 \times n_2$. For clarity, we assume I_1 and I_2 are square images. An arbitrary pixel in I_1 and its mapped pixel are denoted as p_1 and p_2 respectively. We have

$$p_2 = T(p_1). \quad (1)$$

T is a 3×3 matrix which denotes affine transformation between p_1 and p_2 . In the following formula, \mathbf{k} includes operations such as rotation, scaling, and shear. \mathbf{t} includes translation operations:

$$T = \begin{bmatrix} \mathbf{k} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \quad (2)$$

SAD is used to measure the similarity between I_1 and a candidate area in I_2 which is generated by a candidate transformation T . Normalized gray scale difference between each p_1

transformation	range	step amount	step size
rotation	$[0, 2\pi]$	2^n	$\frac{\pi}{2^{n-1}}$
translation	$[-n_2, n_2]$	2^n	$\frac{n_2}{2^{n-1}}$
scale	$[\frac{n_1}{n_2}, \frac{n_2}{n_1}]$	2^n	$\frac{n_2^2 - n_1^2}{n_1 n_2 2^n}$

Table 1: Value ranges of parameters for constructing a Galois field of affine transformation.

and corresponding p_2 is summed, which can be written as:

$$S(I_1, I_2, T) = \frac{\sum_{p_1 \in I_1} |I_1(p_1) - I_2(p_2)|^m}{n_1^2}, \quad m = \begin{cases} 0 & p_2 \notin I_2 \\ 1 & p_2 \in I_2 \end{cases}. \quad (3)$$

The purpose of our paper is to estimate an approximate affine transformation \hat{T} from a given candidate set. In the best case, \hat{T} equals to transformation \bar{T} . \bar{T} is the closest transformation to ground truth T' among all the candidate transformations. A natural way to estimate \hat{T} is to minimize SAD. Formally, our purpose can be denoted as:

$$\hat{T} = \arg \min_{T \in \mathbb{F}_{2^{6n}}} S(I_1, I_2, T). \quad (4)$$

The construction of candidate set will be introduced in the Section 3.2. From Equation 4, we can still not ensure that \hat{T} is close enough to T' , because SAD is related with not only transformation but also the variation of a template. Variation v of a template can be defined as the sum of maximal difference between each p_1 and its eight neighbors $N_8(p_1)$. Formally,

$$v = \sum_{p_1 \in I_1} \max_{q \in N_8(p_1)} |I_1(p_1) - I_1(q)|. \quad (5)$$

Large variation means that a template is not smooth. In this case, SAD values of two candidate transformations can differ a lot even the transformations are very close. Detail explanation will be discussed in the next section.

3.2 Galois Field of Affine Transformation

Matching with complete continuous affine transformation set which contains infinite candidates can be impractical. To simplify this problem, we build a discrete searching space in terms of binary Galois field. The smallest Galois field \mathbb{F}_2 can be extended to arbitrary fields \mathbb{F}_{2^n} as long as we allow definition of bitwise operations on strings of bits [10]. \mathbb{F}_{2^n} has been widely applied in coding theory. In our research, it performs as a sampling grid.

According to [10], a general affine transformation matrix can be decomposed into $T = TrR_2SR_1$, where R_1 represents matrix operation of first rotation, S is scale operation in x-axis and y-axis, R_2 is second rotation, Tr is translation operation in x-axis and y-axis. By this decomposition, we will have six DOFs given a certain affine transform. To construct a Galois field of affine, we summarize the range of each DOF in Table 1.

Transformations over each decomposed DOF can be modeled by a Galois field \mathbb{F}_{2^n} , n is a positive integer denoting the length of binary codes and 2^n is the field's size. Elements in \mathbb{F}_{2^n} are represented as binary codes. For clarity, we assume n of each decomposed DOF is

the same. Each DOF's range is then divided into 2^n discrete segments. $T \in \mathbb{F}_{2^{6n}}$ denotes a general affine transformation in six DOFs. Acceptable margin of error can then be guaranteed on this Galois field. The maximum error of rotation is within $[-\pi/2^{n-1}, \pi/2^{n-1}]$, the maximum error of translation is within $[-n_2/2^{n-1}, n_2/2^{n-1}]$, the maximum error of scaling is within $[(-n_2^2 - n_1^2)/(n_1 n_2 2^n), (n_2^2 - n_1^2)/(n_1 n_2 2^n)]$.

To quantify the error between two transformations, following formula is defined:

$$E(T_1, T_2) = |S(I_1, I_2, T_1) - S(I_1, I_2, T_2)|. \quad (6)$$

It has been proved in [10] (Theorem 3.1) that the upper limit of $E(\bar{T}, T')$ is associated with three factors in a discrete set of affine: step amount, variation of template, and template size. For Galois field of affine, we can rewrite:

$$E(\bar{T}, T') \leq O\left(\frac{\nu}{2^{6n} \times n_1}\right) \quad (7)$$

With loose upper limitation, which may be caused by small n_1 , small n , or large ν , there exists possibility that $E(\hat{T}, T') < E(\bar{T}, T')$. Note that \bar{T} is the closest transformation to T' in the Galois field, not the transformation which can minimize $E(T, T')$. In such situation, it is impossible to estimate the right affine transformation by minimizing SAD and will not be taken into account in this paper. In order to avoid such conditions, n_1 is limited in the experiments, because the influence of ν is inevitable and n should be reasonably small considering the time and space complexity.

An appropriate choice of n is required in order to limit the maximum error to an acceptable range. However, size of Galois field grows exponentially with the increase of n . Typically, when $n = 8$, the total size of entire Galois field can be nearly 2.8×10^{14} . Considering a personal computer can not afford such a large amount of calculation, we will introduce our sampling method over $\mathbb{F}_{2^{6n}}$ in the next section.

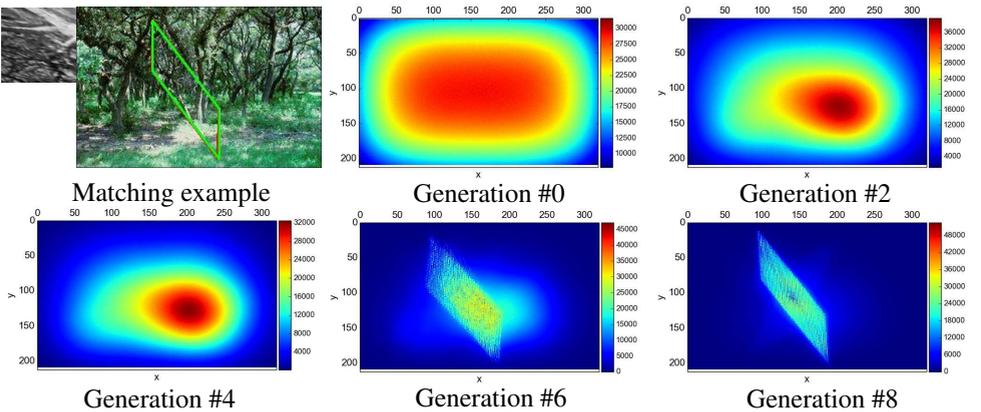


Figure 2: Heat map of matching frequency. This figure shows the frequency that each pixel has been used for calculating SAD. With the decrease of population number, the total matching frequency reduces while a more accurate candidate area can be localized.

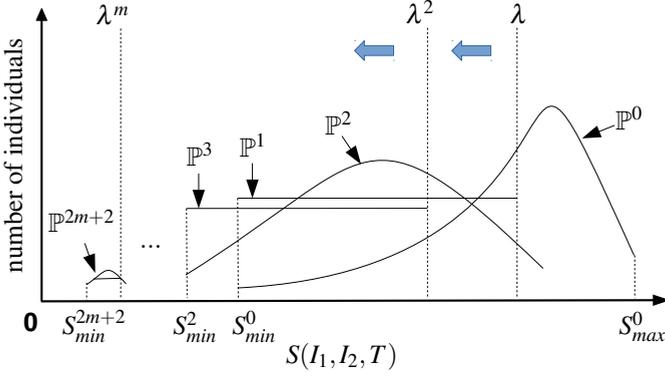


Figure 3: Illustration of SAD’s distribution in each generation \mathbb{P} . Selection, bounding scheme and crossover on the individuals make the distribution move to left gradually, which is the procedure of estimating \hat{T} .

3.3 Level-wise Adaptive Sampling (LAS)

In this section, we will introduce LAS which aims to achieve a satisfactory error rate instead of testing the complete $\mathbb{F}_{2^{6n}}$. Our method is based on genetic algorithm (GA) [4]. From the perspective of GA, our problem can be defined as a minimization problem of SAD. However, in order to optimize \hat{T} in such a broad search space, two major issues should be faced: 1) how to escape from local optimum. 2) how to control the optimization response time.

Preserving genetic variety: It has been argued in [8] that in order to prevent GA from falling into local optimum, genetic variety should be preserved somehow. Although mutation operation can surely increase the genetic variety randomly, it can also destroy individuals which are potentially to be close to \hat{T} . In a broad search space, the probability to create a “suitable” diversity is very low and a high mutation rate can contrarily slow down the speed of convergence. It is worth noting that in our problem, a large enough number of randomly initialized population keeps sufficient genetic variety for converging to \hat{T} . During the evolution, selection operation such as roulette wheel selection is likely to select individuals which hold larger fitness for crossover operation. With the combination of selection and crossover, genetic variety decreases and the whole population converges to an optimum solution. However, if an individual happened to hold small SAD (e.g. a flat candidate area) in the early stage of evolution, the whole population will easily fall into a local optimum especially when the search space is very broad. To preserve genetic variety, we select individuals from each SAD level uniformly. Each SAD level is a discrete interval which is occupied by a part of individuals. With maximum SAD in m th generation defined as S_{max}^m , minimum SAD in m th generation defined as S_{min}^m and the number of SAD level defined as σ , we can define i th SAD level as $[S_{min}^m + (i - 1)(S_{max}^m - S_{min}^m)/\sigma, S_{min}^m + i(S_{max}^m - S_{min}^m)/\sigma]$. Each individual which is assigned to i th SAD level should have a fitness within this range. Individuals of next generation are then randomly selected from each SAD level. The number of individuals selected from each SAD level is the same. With the increase of σ , distribution of SAD in $m + 1$ generation approximates to uniform distribution.

Fitness uniform selection scheme (FUSS) is proposed in [8], which selects a fitness value uniformly at first and then randomly select the nearest individual. The difference is, LAS can

Algorithm 1 Level-wise Adaptive Sampling.**Input:** Normalized template and target image, I_1, I_2 .**Input:** Population number δ of initial generation.**Input:** Population bounding parameter λ .**Input:** Population number c of last generation.**Output:** Estimated transformation \hat{T} .

```

1:  $\mathbb{P}^0 = \{T_0, \dots, T_\delta\}$ 
2:  $m = 0$ 
3: while  $|\mathbb{P}^{2m}| > c$  do
4:   Sample learning set  $\mathbb{L} \in \mathbb{P}^{2m}$ 
5:    $\mathbb{L}' = \{T_i | \exists T_i \in \mathbb{L} \text{ s.t. } S(I_1, I_2, T_i) < 0.1 \times \alpha + \beta\}$ , tuning  $\alpha$  and  $\beta$  s.t.  $|\mathbb{L}'|/|\mathbb{L}| \approx \lambda$ 
6:    $\mathbb{P}^{2m+1} = \{T_i | T_i \in \mathbb{P}^{2m}, S(I_1, I_2, T_i) < 0.1 \times \alpha + \beta, S(I_1, I_2, T_i) \sim U(S_{min}^{2m}, S_{max}^{2m})\}$ 
7:    $\mathbb{P}^{2m+2} = \text{crossover}(\mathbb{P}^{2m+1})$ 
8:    $m = m + 1$ 
9: end while
10: return  $\hat{T} \in \mathbb{P}^{2m+2}$  s.t.  $S(I_1, I_2, \hat{T}) = S_{min}^{2m+2}$ 

```

control the degree of uniform approximation by σ , which can directly affect the convergence speed. FUSS will take a longer time to converge, because the individuals with high fitness in FUSS make up only a small percentage of overall individuals.

Bounding population number: Evaluating a large number of population at initial generation is very important to avoid falling into local optimum. However, evaluating entire generations with same population number is time consuming and not practical. To accelerate the evolution procedure, we wish to rule out the candidate individuals which hold high SAD score. Instead of determining a fixed threshold, we learn a threshold at each generation which can rule out a certain fraction (λ percent) of individuals. Learning procedure is to tune two constants α and β such that $S(I_1, I_2, T) < 0.1 \times \alpha + \beta$ holds for λ percent of the individuals. To make the algorithm find the approximate threshold more effectively, α varies as an integer and β varies as a float number. Figure 2 illustrates the matching frequency of each pixel in each generation. With the decrease of population number, matching frequency around each local optimum reduces, and finally the area with respect to global optimum transformation is approached. Considering the bounding scheme, the number of matching tests that LAS requires can be represented as $\sum_{i=0}^m \delta \lambda^i$, where $\delta \lambda^m > c$. δ is the population number of initial generation. c is a small constant which denotes the population number of the last generation. The time complexity can then be ensured as long as the parameters are predetermined.

Approximation of SAD: Each matching test with respect to a single transformation has a time complexity of $O(n_1^2)$. To speed up each matching test, we wish to inspect only a small fraction of pixels instead of the entire pixels in template. We sample pixels at an equal interval on both width and height of template by a parameter ε to reduce the time complexity to $O(n_1^2/\varepsilon^2)$. The Equation 3 can then be rewrote as following if the number of sampling pixels is enough.

$$S(I_1, I_2, T) \approx S(I'_1, I_2, T), |I'_1| = n_1^2/\varepsilon^2. \quad (8)$$

According to Chernoff bound, the number of sampling pixels should be $\log(1/\eta)n_1^2/\varepsilon^2$ if we

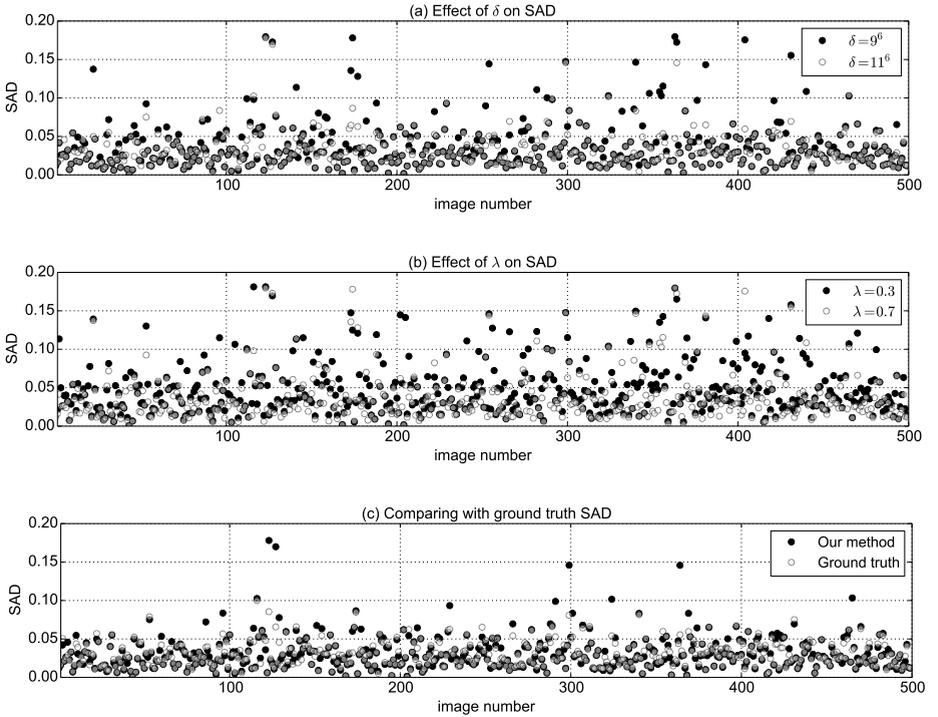


Figure 4: Parameter analysis on evaluation data set. (a) Effect of δ . Other parameters are: $\varepsilon = 3$ and $\lambda = 0.7$. (b) Effect of λ . Other parameters are: $\varepsilon = 3$ and $\delta = 9^6$. (c) Comparing tuned parameters with ground truth. Tuned parameters are: $\varepsilon = 3$, $\delta = 11^6$, and $\lambda = 0.7$.

wish $|S(I_1, I_2, T) - S(I'_1, I_2, T)| < \varepsilon/n_1$ holds with probability $1 - \eta$. In our setting, $\eta = 1/e$. This also has been pointed out in [10].

The entire procedure of LAS is described in Algorithm 1. All the transformations T are represented as binary Gray codes in Galois field. LAS runs in multiple generations, with each generation i generates a population \mathbb{P}^i . In initial population \mathbb{P}^0 , individuals are sampled randomly from \mathbb{F}_{2^n} . Figure 3 illustrates the relation between SAD and the number of corresponding individuals throughout the convergence process. With the generation number grows, the overall distribution translates from right to left as a result of selection and crossover. The amplitude decreases as a result of the population bounding scheme. Note that S_{min}^{2m+1} equals to S_{min}^{2m+2} and S_{max}^{2m+1} equals to S_{max}^{2m+2} , because the level-wise selection will not generate new solutions.

4 Experiments

To evaluate our algorithm, we use images from the famous SUN database [15], which has been used in evaluating many vision problems. We select 500 images as tests from category “waiting room” to “zoo”. We randomly generate a ground truth affine transformation matrix for each test image, and make sure that the four corners of parallelogram generated by corre-

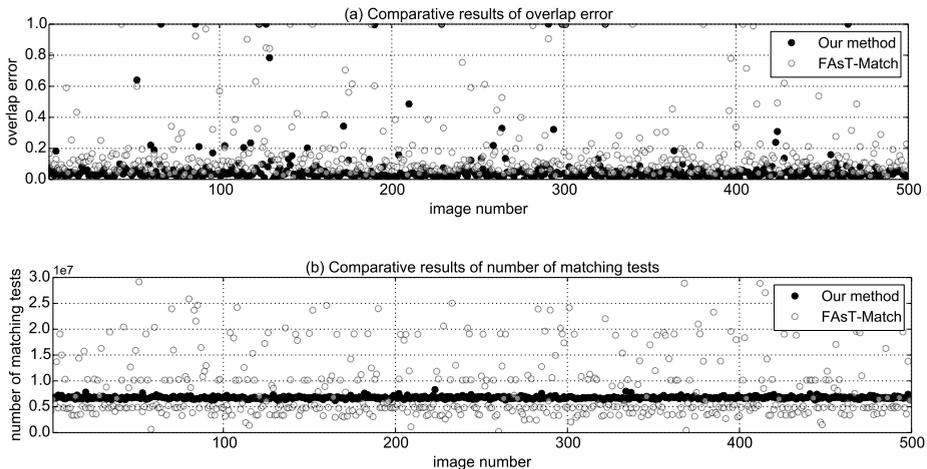


Figure 5: Comparative results with 500 images. Gray colour represents the overlapping. Parameters are set as: $\delta = 11^6$, $\varepsilon = 3$, $\lambda = 0.7$. (a) Overlap rate error on each test image. (b) Number of matching tests on each test image.

sponding matrix are all in the image. Pixels in the parallelogram are then warped to generate the square template. In our experiment, each template has a size of 100×100 pixels and the origin of each transformation is the origin of each target image.

Effect of parameters: We observe the change of SAD while changing the parameter δ and λ . Figure 4(a) shows that larger δ can improve the performance of SAD on the images which are not matched well using smaller δ . For the matching results which are close enough to the ground truth, it is hard to improve the performance by increasing δ . Figure 4(b) shows that small λ will only achieve rough results, because the algorithm converges too fast before a global optimum is localized. It is worth pointing out that even the ground truth transformations can have SAD larger than 0, because interpolation operations (bilinear interpolation) are involved during the creation of templates (warping). From Figure 4(c), we can find out that our result can well fit the SAD of ground truth in most cases.

Comparative results: We compare our algorithm with the state-of-the-art method FAsT-Match [14]. We use the overlap error to compare the accuracy which is defined as $1 - (\text{area}(\hat{T}) \cap \text{area}(T')) / (\text{area}(\hat{T}) \cup \text{area}(T'))$ by PASCAL measure [9]. We use number of matching tests to compare the efficiency which does not depend on type of programming languages and hard devices. In order to ensure the comparative results to be fair and accurate, the experiment is carried out under the following conditions: 1) No preprocessing like Gaussian blur. Although smoothing images will surely improve the accuracy, it will also bring complexities when analysing the results. 2) Set the approximation method of SAD as the same, number of sub-sampled pixels should be n_1^2 / ε^2 . 3) Set the number of matching tests as the same. It is difficult to control the number of matching tests of FAsT-Match, because it is dynamically determined. We only set its upper limit to avoid memory overflow. 4) To keep the simplicity of algorithm, restarting an algorithm or other similar tricks for improving the accuracy are not allowed. From Figure 5(a), we can see that with respect to different images, our method has a significant reduction on overlap error. From Figure 5(b),



Figure 6: Examples of successful matching. Red parallelogram represents the ground truth, green parallelogram represents the matching result.

we can see that our method is more stable in algorithm’s complexity. By changing the criterion of overlap error, we report accuracy in Table 2. We present examples of our matching results in Figure 6.

method	error < 50%	error < 10%	error < 5%	average matching tests
FASt-Match	92.2%	48.4%	11.8%	8.6×10^6
Our method	97.4%	91.0%	68.0%	6.7×10^6

Table 2: Accuracy of different overlap error criterion and average number of matching tests.

5 Conclusion and limitations

In this paper, we presented a method to solve affine template matching problem in Galois field. For efficiency, we proposed level-wise adaptive sampling (LAS) method under genetic algorithm framework to estimate only a small fraction of candidate transformations. Experiments have shown that our algorithm is more accurate and faster than the state-of-the-art affine template matching method. The drawbacks of our algorithm can be concluded as: 1) The smooth assumption limits the application of our algorithm. For template with large variation, we have to increase δ . 2) Since GA brings about heuristics, there is no absolute assurance that our algorithm can find the global optimum by the limited matching tests. As the future work, we plan to extend our algorithm to projective template matching problem.

References

- [1] Takuya Akashi, Yuji Wakasa, Kanya Tanaka, Stephen Karungaru, and Minoru Fukumi. Using genetic algorithm for eye detection and tracking in video sequence. *Journal of*

- Systemics, Cybernetics and Informatics (JSCI)*, 5(2):72–78, 2007.
- [2] Min-Seok Choi and Whoi-Yul Kim. A novel two stage template matching method for rotation and illumination invariance. *Pattern Recognition (PR)*, 35(1):119–129, 2002.
- [3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision (IJCV)*, 88(2):303–338, 2010.
- [4] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [5] Kimmo Fredriksson, Veli Mäkinen, and Gonzalo Navarro. Rotation and lighting invariant template matching. *Information and Computation*, 205(7):1096–1113, 2007.
- [6] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [7] John H Holland. Adaptation in natural and artificial systems. an introductory analysis with applications to biology, control and artificial intelligence. *Ann Arbor: University of Michigan Press*, 1, 1975.
- [8] Marcus Hutter and Shane Legg. Fitness uniform optimization. *IEEE Transactions on Evolutionary Computation (TEVC)*, 10(5):568–589, 2006.
- [9] Hae Yong Kim and Sidnei Alves de Araújo. Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast. In *Pacific Rim Conference on Advances in Image and Video Technology (PSIVT)*, pages 100–113. Springer-Verlag, 2007.
- [10] Simon Korman, Daniel Reichman, Gilad Tsur, and Shai Avidan. Fast-match: Fast affine template matching. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2331–2338. IEEE, 2013.
- [11] Rudolf Lidl. *Finite fields*, volume 20. Cambridge University Press, 1997.
- [12] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.
- [13] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1-2):43–72, 2005.
- [14] Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences (SIIMS)*, 2(2):438–469, 2009.
- [15] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR)*, pages 3485–3492. IEEE, 2010.