Fast Image Alignment with Fourier Moment Matching on GPU

Hong-Ren Su[1], Hao-Yuan Kuo[1], Shang-Hong Lai[1,2] and Chin-Chia Wu[3]
[1]Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan
[2]Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
[3]Mechanical and Systems Research Lab., Industrial Technology Research Institute, Chutung, Hsinchu, Taiwan
E-mail: suhongren@gmail.com, lai@cs.nthu.edu.tw

***Abstract*—In this paper, we develop a fast and accurate image alignment system which can be applied to image sequences in real time. The proposed image alignment system consists of two main components: the development of Fourier moment matching system and the implementation of the system in GPU. The Fourier moment matching is to efficiently find the location, orientation and size of the template from an input image. The GPU implementation speeds up the computation of the Fourier moment matching for the image alignment system to achieve real-time computation.**

I. INTRODUCTION

Automated visual inspection usually requires a precise and fast image alignment technique as the first step. Many image alignment methods have been proposed for this purpose in the past [1]. However, alignment methods with high precision usually have higher time complexity and take much time for alignment. On the contrary, alignment methods with low time complexity often have poor precision. It is a trade-off problem. High-speed computing platforms, like graphics processing unit (GPU), can significantly accelerate a complex and precise alignment technique [2], which is required for real-time inspection. Compared with other image registration techniques [3] in spatial domain, such as pixel-based methods, correlation methods and feature-based methods, Fourier-based methods [4], which estimate the transformation parameters in the frequency domain, usually have lower computational complexity and can handle large rigid transformation. Thus, they are quite suitable for industrial inspection applications.

Unfortunately, the accuracy of the traditional Fourier-based image registration approach is limited to the mapping of log-polar transform due to the spectrum interpolation error [5] from the Cartesian plane to the log-polar plane for estimating rotation and scaling. The interpolation directly on the 2D Cartesian Fourier domain will result in error in the image registration. Thus the log-polar transform is the primary challenge to improve the registration precision and the alignment range. Several modified Fourier-based image registration methods, like the log-polar method [6], phase-only-correlation (POC) [7, 8, 9] and multi-layer fractional Fourier transform (MLFFT) [5], have been proposed to evaluate the log-polar transform more efficiently and reliably. MLFFT uses the fractional Fourier transform to create several spectrums with different resolutions from an image and sums them into one for the log-polar transform. This strategy makes the Fourier-based image registration more accurate than the other Fourier-based methods. However, MLFFT cannot go beyond the framework of the log-polar transform, and it just selects multiple spectrums from the fractional FFT to achieve better approximation.

Fourier moment matching [10] was recently proposed to improve the accuracy of Fourier-based image registration without using the log-polar transform. The algorithm is based on the fact that the linear transform between two images corresponds to a related linear transform between their Fourier spectrums, whose energies are normally concentrated around the origin in the frequency domain. Thus, the moments for the corresponding Fourier spectrum distributions can be calculated. The resulting registration algorithm is based on minimizing the relationship between the moments for the Fourier spectrums of the two images. The method has higher accuracy and less time complexity. Based on the Fourier moment matching principle, we derive the Fourier moment relationship between two images related by a scaled rigid transformation.

Furthermore, the proposed image alignment method is implemented in a parallel computing platform. In this paper, we implement the proposed algorithm on a CPU+GPU computing platform and show significant speed-up through experiments.

II. FOURIER-BASED IMAGE REGISTRATION

There have been several Fourier-based methods proposed for image registration in the past [5, 11, 12]. The Fourier-based methods have the advantages that they are robust to noise with low computational complexity and they can be easily implemented on a parallel computing platform.

A. *Rigid Transform Relationship between Image and Fourier Domain*

Consider two images $f_1(x,y)$ and $f_2(x,y)$ which are related by a rigid transformation, i.e. $f_1(x,y) = f_2(r_0 cos\theta_0 x + r_0 sin\theta_0 y + c, -r_0 sin\theta_0 x + r_0 cos\theta_0 + f)$ with the rotation θ_0, scaling r_0 and translation vector $[\ c\ f\ ]^t$. Assume the Fourier transforms of the image functions $f_1(x,y)$ and $f_2(x,y)$ be denoted by $F_1(u,v)$ and $F_2(u,v)$, respectively. Then, we can derive the following rigid relationship between the Fourier transforms $f_1(x,y)$ and $f_2(x,y)$ given as follows [4]:

$$F_1(u,v) = \frac{1}{{r_0}^2} \times e^{i\frac{[(c cos\theta_0 - f sin\theta_0)u + (f cos\theta_0 + c sin\theta_0)v]}{r_0}} \times F_2\left(\frac{cos\theta_0 u + sin\theta_0 v}{r_0}, \frac{-sin\theta_0 u + cos\theta_0 v}{r_0}\right)$$

By using the transformation $u' = (cos\theta_0 u + sin\theta_0 v)/r_0$ and $v' = (-sin\theta_0 u + cos\theta_0 v)/r_0$, we have the relationship $u = r_0(cos\theta_0 u' - sin\theta_0 v')$ and $v = r_0(sin\theta_0 u' + cos\theta_0 v'$. Taking the absolute values on both sides of eq. (1), we have the rigid transformation relationship between the spectrums $|F_1(u,v)|$ and $|F_2(u,v)|$ as follows:

$$|F_1(u,v)| = \frac{1}{|r_0^2|}|F_2(u',v')| \tag{2}$$

where

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} r_0 cos\theta_0 & -r_0 sin\theta_0 \\ r_0 sin\theta_0 & r_0 cos\theta_0 \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix} \tag{3}$$

Eq. (2) and (3) show that the rigid transformation relationship in the amplitude of their Fourier spectrums only has two parameters, rotation θ_0 and scaling r_0. In order to evaluate them easily, Reddy and Chatterji [4] proposed the log-polar transform from the Cartesian plane $|F_1(u,v)|$ and $|F_2(u,v)|$ into the log-polar plane $M_1(log\ r, \theta)$ and $M_2(log\ r, \theta)$ in eq. (2) by letting $u = rcos\theta$ and $v = rsin\theta$. Eq. (2) can be re-written as follows:

$$M_1(log\,r, \theta) = \frac{1}{r_0^2} M_2(log\,r - log r_0, \theta - \theta_0) \tag{4}$$

Eq. (4) shows that the problem for estimating rotation and scaling is converted into a translation-like problem by taking the log-polar transform. Thus, the rotation and scaling can be determined as follows:

$$(log\,r_0, \theta_0) = arg \max_{(log\,r,\theta)} real\left(IFT\left\{\frac{FT(M_1)conj(FT(M_2))}{|FT(M_1)conj(FT(M_2))|}\right\}\right) \tag{5}$$

where FT and IFT denotes the forward and inverse Fourier transform operators, respectively, and conj denotes the complex conjugate operator.

The rotation θ_0 and scaling r_0 can be first estimated based on eq. (5) from the two Fourier spectrums $FT(M_1)$ and $FT(M_2)$ through the log-polar transform, and the remaining translation vector $[\ c\ f\ ]^t$ can also be computed by transforming image $f_1(x,y)$ with the transformation $f_1'(x,y) = f_1(r_0 cos\theta_0 x + r_0 sin\theta_0 y, -r_0 sin\theta_0 x + r_0 cos\theta_0)$ and determining the translation between f_1'(x,y) and f_2 (x,y) from their cross power spectrum. To be more specific, the translation vector is determined by

$$(c,f) = arg \max_{(x,y)} real\left(IFT\left\{\frac{F_1'(u,v){F_2}^*(u,v)}{|F_1'(u,v){F_2}^*(u,v)|}\right\}\right) \tag{6}$$

where $F_1'(u,v)$ is the Fourier transform of f_1'(x,y), and $F_2^*(u,v)$ is the complex conjugate of $F_2(u,v)$.

B. *Moment Matching Approach to Estimating Rigid Transform Matrix*

Due to the interpolation error in the log-polar transform, the Fourier moment matching approach [10] is to estimate the image transform in the spectrums of two images. We use the same idea to develop a fast image alignment for the rotation and scaling of the rigid transform in eq. (2) instead of using the log-polar transform.

Let the Fourier spectrums of the 2D image functions $f_1(x,y)$ and $f_2(x,y)$ be denoted by $F_1(u,v)$ and $F_2(u,v)$, respectively. If $f_1(x,y)$ and $f_2(x,y)$ are related by a rigid transformation, then their Fourier spectrums are also related by the corresponding rigid transform, i.e. $|F_1(u,v)| = |F_2(u',v')|/r_0^2$ with the relation between (u,v) and (u',v') given in eq. (3). We rewrite eq. (3) by $a = r_0 cos\theta_0$ and $b = r_0 sin\theta_0$ as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix} \tag{7}$$

To determine the rigid transform parameters, *a* and *b*, we employ the moment matching technique to the Fourier spectrums $F_1(u,v)$ and $F_2(u,v)$. The $(i+j)^{th}$-order moment for the Fourier spectrum $|F_k(u,v)|$, $k = 1$ or 2, is defined as

$$m_{i,j}^{(k)} = \iint u^i v^j |F_k(u,v)|\, dudv \tag{8}$$

By applying the coordinate substitution, we can derive the following equation:

$$m_{i,j}^{(1)} = \iint u^i v^j |F_1(u,v)|\, dudv \tag{9}$$

and

$$m_{i,j}^{(2)} = \iint (au' - bv')^i (bu' + av')^j |F_2(u',v')|\, du'dv' \tag{10}$$

Thus, we have the following relationship between the first-order moments of the two Fourier spectrums from eq. (9) and eq. (10).

$$\begin{bmatrix} m_{1,0}^{(1)} \\ m_{0,1}^{(1)} \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} m_{1,0}^{(2)} \\ m_{0,1}^{(2)} \end{bmatrix} \tag{11}$$

The 2D rigid transform parameters (a,b) can be estimated by minimizing the errors associated with the constraints in eq. (11) in a least-squares estimation framework. The least-square solution to these two parameters can be derived in a closed-form solution as follows:

$$a = \frac{m_{1,0}^{(1)} m_{1,0}^{(2)} + m_{0,1}^{(1)} m_{0,1}^{(2)}}{(m_{1,0}^{(2)})^2 + (m_{0,1}^{(2)})^2} \tag{12}$$

and

$$b = \frac{m_{0,1}^{(1)} m_{1,0}^{(2)} - m_{1,0}^{(1)} m_{0,1}^{(2)}}{\left(m_{1,0}^{(2)}\right)^2 + \left(m_{0,1}^{(2)}\right)^2} \tag{13}$$

III. IMPLEMENTATION OF THE PROPOSED ALGORITHM ON GPU COMPUTING PLATFORMS

In order to develop a real-time image alignment system, we need to speed up the proposed algorithm by the parallel computing of GPU. The parallel computing of GPU can share a lot of computation load in the CPU system to achieve real-time performance in the proposed algorithm. CUDA (Compute Unified Device Architecture) [13] is a parallel computing platform and programming model invented by NVIDIA. To facilitate scientific computing, CUDA also provides some libraries which are used for mathematical or scientific applications, such as cudaFFT, ArrayFile [14] and so on. We implement the proposed image alignment system with CUDA 4.1 and use the cudaFFT library for Fast Fourier transform (FFT) unit in the algorithm because cudaFFT library provides good performance in the parallel computing of GPU.

TABLE I
MAIN STEPS IN THE FOURIER MOMENT BASED IMAGE ALIGNMENT

FOURIER MOMENT BASED IMAGE ALIGNMENT
1. Compute the discrete Fourier transforms of two images $f_1(x,y)$ and $f_2(x,y)$ via FFT.
2. Compute the first order moments for the amplitude in Fourier spectrums $\|F_1(u,v)\|$ and $\|F_2(u,v)\|$ of $f_1(x,y)$ and $f_2(x,y)$.
3. Determine the rigid transform parameters (a,b) by Eq.(12) and (13).
4. Transform the $f_2(x,y)$ with rigid transform parameters (a,b) only without translation and the transformed data is denoted by $f_2^{'}(x,y)$.
5. Determine the translation vector by the cross-power spectrum of $f_1(x,y)$ and $f_2^{'}(x,y)$.

TABLE II
MAIN STEPS IN THE TRADITIONAL FOURIER BASED IMAGE ALIGNMENT

TRADITIONAL FOURIER BASED IMAGE ALIGNMENT
1. Compute the discrete Fourier transforms of two images $f_1(x,y)$ and $f_2(x,y)$ via FFT.
2. Compute the log-polar transform M_1 and M_2 for the amplitude in Fourier spectrums $\|F_1(u,v)\|$ and $\|F_2(u,v)\|$ of $f_1(x,y)$ and $f_2(x,y)$.
3. Determine the scale factor and the rotation angle by the cross-power spectrum of M_1 and M_2 in Eq. (5).
4. Transform the $f_2(x,y)$ with the scale factor and the rotation angle only without translation and the transformed data is denoted by $f_2^{'}(x,y)$.
5. Determine the translation vector by the cross-power spectrum of $f_1(x,y)$ and $f_2^{'}(x,y)$.

TABLE III
FREQUENCIES OF COMPUTING UNITS USED IN THE FOURIER MOMENT BASED IMAGE ALIGNMENT

Computation unit	Frequency
FFT	4
Moment	2
Cross power spectrum	1
Max value search	1
Image warp	2

TABLE IV
FREQUENCIES OF COMPUTING UNITS USED IN THE TRADITIONAL FOURIER BASED IMAGE ALIGNMENT

Computation unit	Frequency
FFT	7
Log-polar transform	2
Resample	2
Cross power spectrum	2
Max value search	2
Image warp	2

Table I and table II show the main steps of the Fourier moment matching and the traditional Fourier-based image alignment. These tables are very important to let us know the execution procedure for each step, and which step can be accelerated. We can summarize the algorithm in table I and II to obtain the computing units and their computing frequencies showed in table III and table IV. Table III shows the 5 computing units of Fourier moment matching; namely, FFT (table I .1 and 5), moment calculation (table I .2), Cross power spectrum (table I .5), max value search (table I .5) and Image warp (table I .4). Table IV shows the six computing units of traditional Fourier based image registration; namely,

TABLE V
THE IMPLEMENTATION OF THE PROPOSED FOURIER MOMENT BASED IMAGE ALIGNMENT

PRE-CALCULATION

Steps	Template
1	Image f_1 to Fourier spectrum $\|F_1\|$
2	Moment M_1 calculation of Fourier spectrum $\|F_1\|$

ALIGNMENT

Steps	Template	Target	Processing
	Read $\|F_1\|$ and M_1	Read image f_2	Read data
D1	X	Sent image f_2 to device	Data transfer
1	X	Image f_2 to Fourier spectrum $\|F_2\|$	GPU
2	X	Moment M_2 calculation of Fourier spectrum $\|F_2\|$	GPU
D2	Sent M_2 to host		Data transfer
3	Scaling and rotation estimation in M_1 and M_2		CPU
4	Target Image warp $f_1^{'}$		CPU
D3	Sent image $f_1^{'}$ to device		Data transfer
5	Image $f_1^{'}$ to Fourier spectrum $\|F_1^{'}\|$		GPU
6	Translation estimation in cross power spectrum (1 IFFT + max value search)		GPU
D4	Sent translation vector to host		Data transfer
7	Target Image warp $f_1^{''}$		CPU

FFT (table II .1, 3 and 5), log-polar transform (table II .2), resample (table II .2), Cross power spectrum (table I .3 and 5), max value search (table I .3 and 5) and image warp (table I .4). Compared with Table III and IV, the Fourier moment matching has low time complexity than the traditional Fourier-based image alignment. Therefore the subsequent description focuses on the GPU implement of the proposed Fourier moment matching algorithm.

We can observe that the main computational load of the proposed image alignment is the function calls to FFT. Thus, we take advantage of the current cudaFFT library to achieve GPU acceleration for the FFT function. Max value search, Cross power spectrum and moment calculation are programmed on GPU by ourselves. Image warp is programmed on the CPU to reduce the time consumed by the data transmission. Table V shows the implementation of the proposed Fourier moment matching algorithm.

(a) (b) (c)

Fig. 1. Image data sets (a), (b) and (c) with different rotation, scaling and translation for image alignment experiments.

IV. EXPERIMENTAL RESULTS

We have three sets with 100 images in each set for industrial inspection. Fig. 1 shows the images in set (a), (b) and (c). There are parameters of scaling, rotation and translation for ground truth in each image. Our computing platform for experiments shows in the table VI. Table VII shows the mean square errors in scaling, rotation and translation for the traditional Fourier-based registration [4] and Fourier moment matching method in each image set. Fourier moment matching provides higher accuracy from our experiments on simulated images. Table VIII shows the average running time of the traditional Fourier-based registration and the proposed Fourier moment matching method. The proposed Fourier moment matching method took less execution time than that of the traditional Fourier-based registration method for both CPU and GPU implementations. Thus, it is suitable for real-time industrial inspection. Fig. 2 and 3 are the true image for industrial inspection. The true rigid transform can be estimated by the Fourier moment matching and the detection results are showed in the red rectangle of Fig. 2 and 3.

V. CONCLUSIONS

Fast image alignment is very important for industrial visual inspection. In this paper, we developed an efficient Fourier-moment matching algorithm that is more accurate and more efficient than the traditional Fourier registration method. We also show how the proposed method is implemented on GPU to achieve significant speed-up with the parallel computing.

TABLE VI
COMPUTING PLATFORMS USED IN OUR EXPERIMENTS

Platform	System 1	System 2
CPU	Intel(R)Core(TM) i7 2760QM 2.2GHz	Intel(R)Core(TM) 2CPU 1.86GHz
Memory	DDR2-667 2G	DDR2-667 2G
GPU	GEFORCE GT 630M	GEFORCE GTX 650

TABLE VII
MEAN SQUARE ERROR (MSE) OF TRADITIONAL FOURIER-BASED REGISTRATION AND FOURIER MOMENT MATCHING IN SCALING, ROTATION AND TRANSLATION.

Algorithms	Rigid transform	(a)	(b)	(c)
Traditional Fourier-based registration	Scaling	0.02	0.03	0.05
	Rotation	0.83	0.94	0.75
	Translation	0.12	0.23	0.17
Fourier Moment matching	Scaling	0.02	0.04	0.04
	Rotation	0.78	0.87	0.71
	Translation	0.11	0.21	0.16

TABLE VIII
THE AVERAGE TIME OF TRADITIONAL FOURIER-BASED REGISTRATION AND FOURIER MOMENT MATCHING IN CPU AND GPU PROGRAM

Algorithms	System	Program	Average time (sec)
Traditional Fourier-based registration	System 1	CPU only	1.321
		CPU+GPU	0.134
	System2	CPU only	1.982
		CPU+GPU	0.203
Fourier Moment matching	System 1	CPU only	0.352
		CPU+GPU	0.038
	System 2	CPU only	0.781
		CPU+GPU	0.090

REFERENCES

[1] B. Zitova and J. Flusser. “Image Registration Methods: A Survey,” Image Vision Computing, 21(11), pp. 977-1000, 2003.

[2] R. Shams, P. Sadeghi, R. Kennedy, and R. Hartley. “A survey of medical image registration on multicore and the GPU,” IEEE Signal Processing Magazine, 27(2), pp. 50-60, 2010.

[3] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim. “Elastix: a toolbox for intensity-based medical image registration,” IEEE transactions on medical imaging, 29(1), pp. 196-205, 2010.

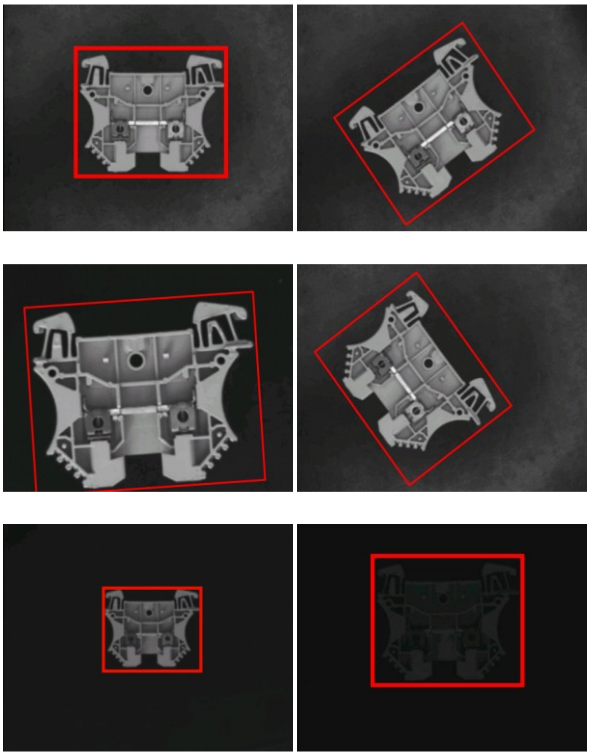

Fig. 2 True image data sets. Red rectangle means target detection location.

[4] B. S. Reddy and B. N. Chatterji. "An FFT-Based Technique for Translation, Rotation and Scale-Invariant Image Registration," IEEE Trans. on Image Processing, 5(8), pp.1266-1271, 1996.

[5] W. Pan, K. Qin and Y. Chen. "An Adaptable-Multilayer Fractional Fourier Transform Approach for Image Registration," IEEE Trans. on Pattern Analysis and Machine Intelligence, 31(3), pp.400-413, 2009.

[6] S. Zokai and G. Wolberg. "Image registration using Log-Polar mappings for recovery of large-scale similarity and projective transformations," IEEE Trans. Image Processing, 14(10), pp. 1422-1434, 2005.

[7] C. D. Kuglin and D. C. Hines. "The phase correlation image alignment method," Proc. Int. Conf. Cybernetics and Society, pp.163–165, 1975.

[8] K. Takita, T. Aoki, Y. Sasaki, T. Higuchi, and K. Kobayashi. "High-accuracy subpixel image registration based on phase-only correlation," IEICE Trans. Fundamentals, E86-A(8), pp.1925–1934, 2003.

[9] K. Ito, S. Litsuka, and T. Aoki. "A palmprint recognition algorithm using phase-based correspondence matching," Proc. IEEE Int. Conf. Image Processing (ICIP), pp. 1977-1980, 2009.

[10] H. R. Su and S. H. Lai. "CT-MR image registration in 3d k-space based on Fourier moment matching," Advances in Image and Video Technology (PSIVT), pp.299-310, 2012.

[11] H. R. Su, S. H. Lai and Y. H. Tsai: "Robust Fourier-Based Image Alignment with Gradient Complex Image," Proc. Int. Conf. on Pattern Recognition (ICPR), pp.2378-2381, 2010.

[12] Y. Keller, A. Averbuch, and M. Israeli, "Pseudopolar-Based Estimation of Large Translations, Rotations and Scalings in Images," IEEE Trans. Image Processing, 14(1), pp.12-22, 2005

[13] https://developer.nvidia.com/

[14] http://www.accelereyes.com/products/arrayfire

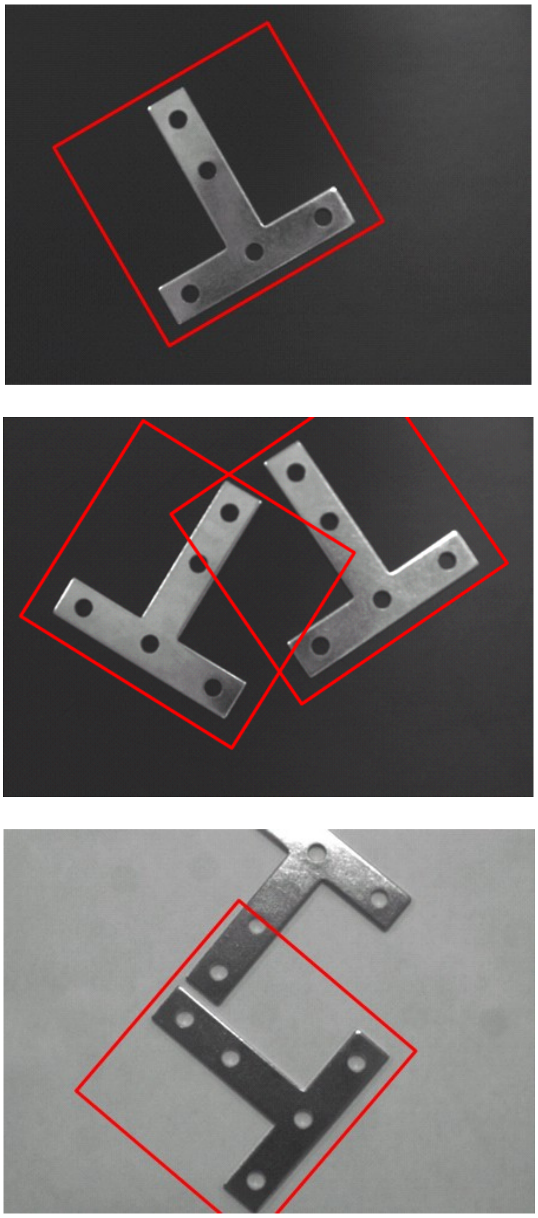

Fig. 3 True image data sets. Red rectangle means target detection location.