

基於物件標記與邊界矩之快速視覺定位技術

葉郡維

台北科技大學

自動化科技研究所

s4618007@ntut.edu.tw

陳金聖

台北科技大學

自動化科技研究所

saint@ntut.edu.tw

中文摘要

本文提出一種基於物件標記與邊界矩之快速視覺定位技術。此技術分為兩階段；影像教導與定位處理。在影像教導中，應用物件標記法、輪廓追蹤技術及邊界不變矩，記錄選取目標之邊界不變矩，以利於後續定位處理之用。接著，定位處理以儲存的目標邊界不變矩為基礎，應用最小距離分類器來辨識待測影像中與教導目標最相似的物件，且此演算法不受待測影像旋轉、位移及縮放之影響。本文所提出之物件標記法，經由實驗證實，其效能較另外三種演算法快速，可符合工業上的需求。最後，以 IC 圖樣及手機面板之影像定位為測試標的，來驗證此定位演算法對於待測影像旋轉、位移與縮放之實用性，由實驗結果可看出本文所提出的物件標記及快速視覺定位演算法，可有效應用在視覺檢測之前處理工具上。

關鍵字：物件標記法、輪廓追蹤技術、邊界不變矩、最小距離分類器。

Fast Image Alignment Based on Connected Component Labeling and Improved Moment Invariants

Chun-Wei Yeh

National Taipei University of Technology

Graduate Institute of Automation Technology

s4618007@ntut.edu.tw

Chin-Sheng Chen

National Taipei University of Technology

Graduate Institute of Automation Technology

saint@ntut.edu.tw

ABSTRACT

This paper presents a fast image alignment algorithm based on connected component labeling and improved moment invariants resulting in rotation, scale and translation invariant. This algorithm consists of two phases: 1) the training phase, and 2) the matching phase. In training phase, the algorithm firstly finds the moment invariants of selected shape in the reference image by using the proposed connected component labeling algorithm, contour tracing technique, and improved moment invariants technique. Subsequently, an efficient pattern matching, used the minimum distance classifier, between the selected shape in training phase and the candidate-shapes in the inspected image is applied in matching phase. Experimental results with LCD drive IC, print circuit board (PCB) and cellular phone images first show the proposed component-labeling algorithm is superior to the other three algorithms in computation loading. And the accurate and high speed alignment results reveal that the proposed alignment algorithms are computing efficient and invariant to rotation, scale and translation. Therefore, the proposed connected component labeling and image alignment algorithm could be the useful preprocessing tools in automated optical inspection (AOI).

Keywords: Connected component labeling, Contour tracing technique, Improved moment invariants, Minimum distance classifier.

1. INTRODUCTION

Image alignment is a fundamental technique for many applications of machine vision and image processing, including image search, object recognition, pose estimation, industrial inspection, target tracking, etc. Image alignment with a two-dimensional (2-D) rigid transformation is usually applied to industrial inspection under a well-controlled environment. The change in rotation of the pattern in the image is one of the most difficult conditions; therefore, the rotation-invariant matching schemes have been required in many industries of inspection. In addition, some vision applications need to be scale-invariant matched, so the image alignment algorithm should be able to accommodate variations in different rotation and scale conditions.

Image alignment technique generally is divided into two major categories, area-based (or intensity-based) and feature-based (or geometry-based) methods [1]. Area-based method sometimes is called correlation-like or template matching method; has been very popular from the past due to the feature of basic ideas. Firstly, the small reference template is applied in a large scene image by sliding the template window in a pixel-by-pixel basis and the normalized cross correlation (NCC) is computed between the template and the scene image. Then, the maximum values or peaks of the computed correlation values indicate the matches between a template and sub-images in the scene. The NCC is most often used to suit the registration of images which only differs by a translation partially. If images are deformed by more complex transformations, the template window can not cover the same parts of the scene in the reference and sensed images. Therefore, several researchers proposed to modify the NCC for the complex transformations of images. In order to make the match invariant to rotation, a ring-projection transformation was proposed [2]. It transforms a two-dimensional (2D) gray-level image into a rotation-invariant representation in the 1D ring-projection space. Tsai and Chiang [3] further proposed the ring-projection to represent a target pattern in the wavelet decomposed detail sub-image, used only the pixels with high wavelet coefficients at lower resolution level to compute the NCC between two compare patterns. In addition, Choi and Kim [4] presented a two-stage image alignment method that first finds candidates by comparing the vector sums of ring-projection, and then these candidates are further matched based on the rotationally invariant Zernike moments.

In the feature-based method, Huttenlocher et al. [5,6] applied the directed Hausdorff distance to develop efficient algorithms for image alignment. Kown et al. [7] proposed a robust hierarchical Hausdorff distance to compare edge maps in a multi-level pyramid structure. In addition, Chen et al. [8] used the Hausdorff distance for image alignment in PCB inspection system. Nevertheless, the above algorithms made the high computational cost when the image alignment involves rotation and scale compensation.

Connected component labeling is a very important preprocessing of image alignment, and many algorithms of connected component labeling have been proposed. Suzuki et al [9] used a one-dimensional table that is able to memorize label equivalences, is used for uniting equivalent labels successively during the operations in forward and backward raster directions. Chang et al [10] further presented the one-pass component-labeling algorithm which is to use a contour tracing technique to detect the external contour and possible internal contours of each component. A general approach to handle the label equivalence information is to use Union-Find algorithm with pointer based rooted trees [11], and then Wu et al. [12] has

used an array instead of the pointer based rooted trees to speed the connected component labeling algorithm. However, the execution time of the above conventional algorithms can not be applied for the real-time applications. The component contour is often a useful resource for identifying objects, therefore, we propose a new algorithm for connected component labeling on binary images, and identify objects from improved moment invariants [13] which are derived from object contours.

In this paper, a fast image alignment technique which combines the connected component labeling and improved moment invariants techniques is developed for industry inspection. This algorithm consists of two phases: 1) the training phase, and 2) the matching phase. In training phase, the algorithm firstly finds the moment invariants of selected shape in the reference image by using the proposed connected component labeling algorithm, contour tracing technique, and improved moment invariants technique. Subsequently, an efficient pattern matching, used the minimum distance classifier, between the selected shape in training phase and the candidate-shapes in the inspected image is applied in matching phase. Experimental results show the proposed algorithm under a wide variation of rotation and scale is robust and computing efficient. Therefore, this could be a useful preprocessing tool in AOI.

2. PROPOSED CONNECTED COMPONENT LABELING AND IMPROVED MOMENT INVARIANTS

The image alignment algorithm, proposed in this paper, includes three techniques: 1) proposed connected component labeling, 2) contour tracing technique, and 3) improved moment invariants.

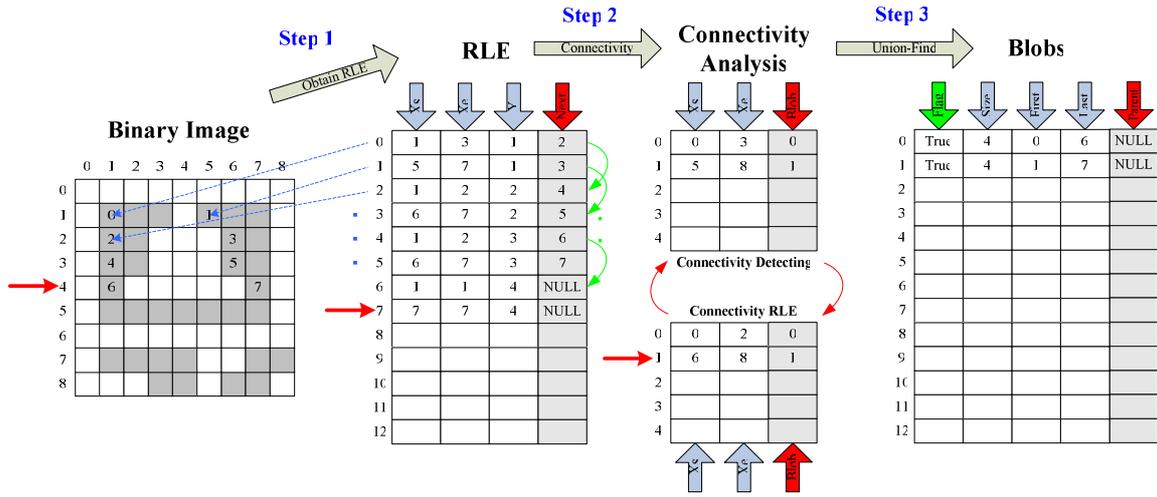
2.1 Proposed Connected Component Labeling

A set of pixels in which each pixel is connected to all other pixels is called a connected component. The proposed connected component labeling algorithm finds all connected components in a binary image and assigns a unique label in the same component. The connected components are determined to find the object properties and locations. In this algorithm, we scan a binary image from left to right and from top to bottom per each line. First of all, a typical binary image shown in Fig. 1 is demonstrated the overview of this algorithm. Conceptually, we can divide the operations into three major steps that are illustrated in Fig. 1. The details of the three major steps are given below.

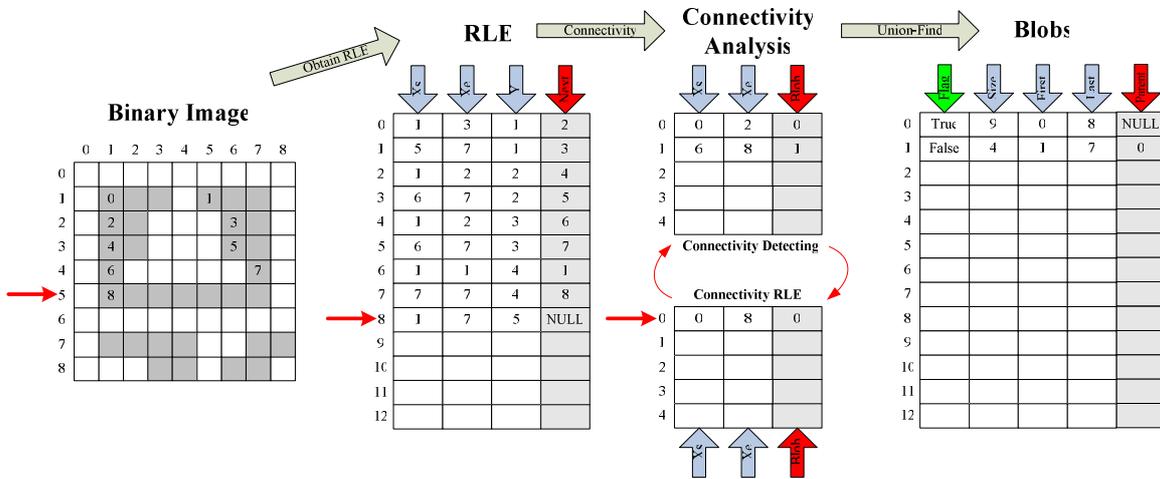
Step 1-Obtain RLE:

The RLE is made by the use of spatial coherence in binary images, and we use the start and end position of runs each row to represent the image. The RLE structure is listed as following:

- Xs : x-coordinate of the start position of a run
- Xe : x-coordinate of the end position of a run.
- Y : y-coordinate of a run.
- Next : index of the next run.



(a) Running in the fourth row of the binary image.



(b) Running in the fifth row of the binary image.

Fig. 1. The proposed connected component labeling for binary image.

Step 2-Connectivity:

From step 1, we use RLE to carry out segment-wise instead of pixel-wise, and only two rows of the image are required in the connectivity step. Note that connectivity is an equivalence relation. For instance, from Fig. 1(a) to 1(b), the connectivity detecting and the connectivity RLE table are changed each row. If the 8th run in the 5th row of the binary image is connected to the runs of the connectivity detecting table (Fig. 2), then union-find step must be executed.

Step 3-Union Find:

In this step, two connected blobs are successively merged into a whole blob tree, where the blob tree is a union of blobs. A typical blob tree is shown in Fig. 3. A function *Union* is used to merge the connected blobs. In Fig. 4(a), blob 1 is merged into blob 0, i.e. blob 0 becomes the parent of blob 1. Hence, the *flag* of blob 1 is set as false which means blob 1 is not a root of the blob tree. Following the above process, we set the *next* of the run 6 as 1 and the *next* of the run 7 as 8, which means blob 0 includes the runs of blob 1, and then the *last* of blob 0 is set as 8.

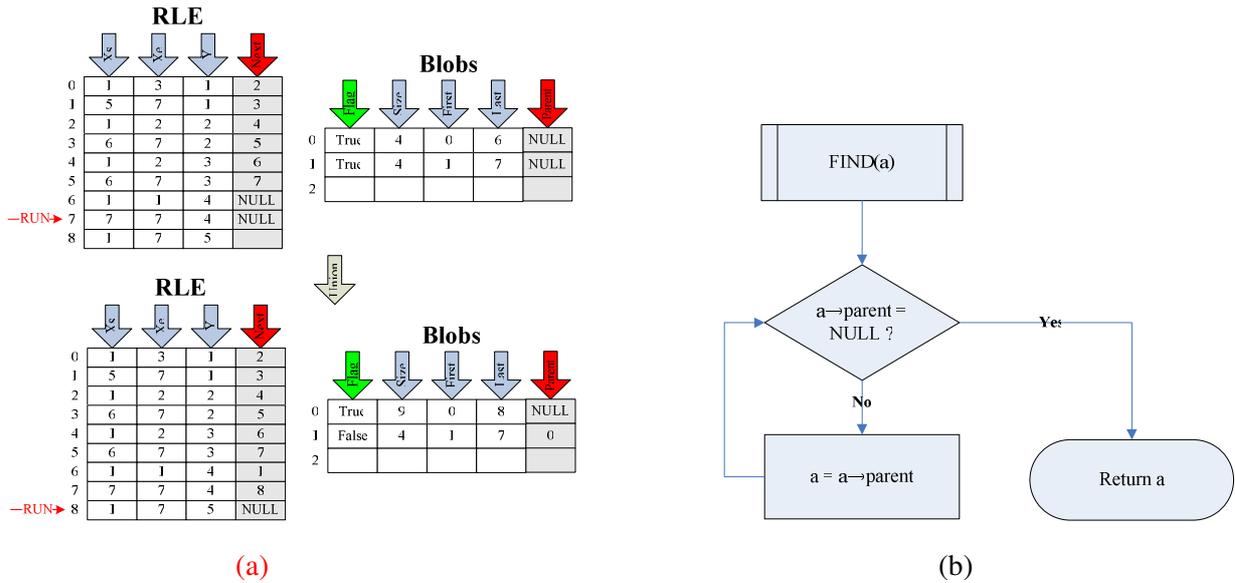


Fig. 4. (a) The *UNION* function, and (b) The *FIND* function in our algorithm.

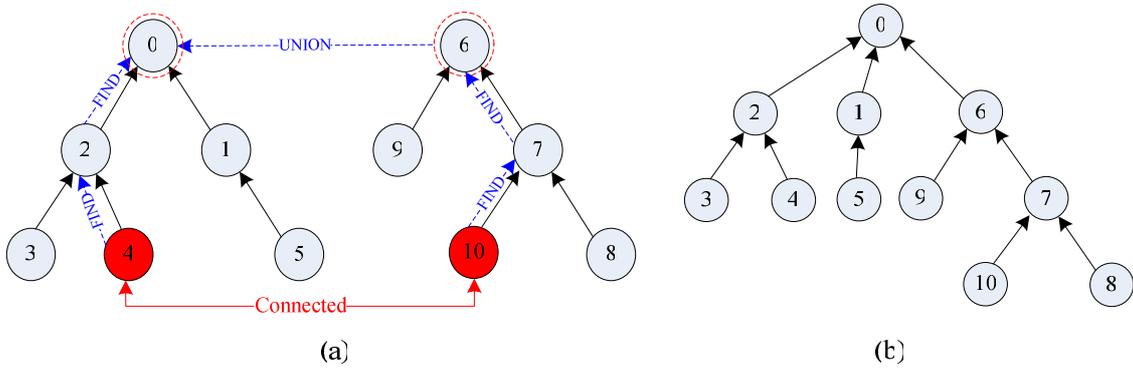


Fig. 5. The *Union Find* processes with two blob trees. (a) Before the *Union Find* process (b) After the *Union Find* process.

2.2 Contour Tracing Technique[10]

As shown in Fig. 6, the goal of the contour tracing technique applied in this paper is to find an external contour at a given point, say F . We first execute a procedure called *Tracer* at point F . *Tracer* will output the contour point following F . Let this point be S , and then we continue to execute *Tracer* to find the contour point following S and so on, until the following two conditions hold: 1) *Tracer* outputs F and 2) the contour point following F is S . The procedure stops only when both conditions hold. In Fig. 6, when F is the starting point and S is the next contour point, the path traced by *Tracer* is $FSTSFDEDF$.

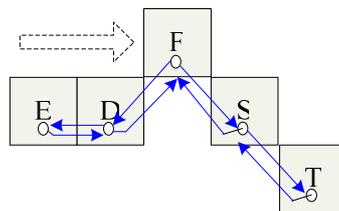


Fig. 6. Tracing the contour of a component.

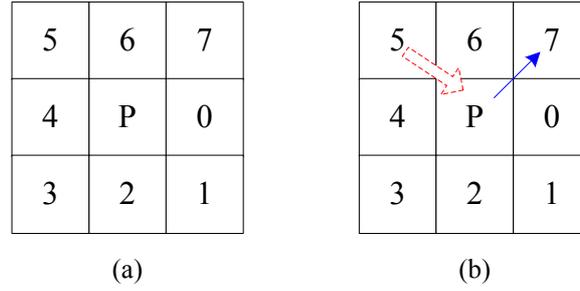


Fig. 7. (a) The neighboring points of P are indexed from 0 to 7. (b) If the previous contour point lies at 5, the next search direction is set on 7.

For a given contour point P , the goal of *Tracer* is to find among P 's eight neighboring points. In Fig. 7(a), the position of each neighboring point of P is assigned to an index. The search proceeds in a clockwise direction from the initial position that is determined in the following way.

In general, when P is not the starting point of a contour, its initial search position is set as $d+2(\text{mod } 8)$, where d is the position of the previous contour point. For example, if the path traced by *Tracer* is FS in Fig. 6, i.e., the previous contour point exists and lies at position 5(Fig. 7(b)), then the initial position is set as index 7, the target path FST is found after the tracing process, i.e., the next is index 1.

2.3 Improved Moment Invariants[13]

The traditional moment invariants, computed based on the information provided by both the shape boundary and its interior region, have been frequently used as feature for shape recognition. To reduce the computation of the traditional moment invariants, the improved moment invariants are computed using the shape boundary only.

Define the $(p, q)^{\text{th}}$ modified moment as

$$m_{pq} = \int_C x^p y^q ds, \quad \text{for } p, q = 0, 1, 2, 3, \dots \quad (1)$$

where \int_C is a line integral along the curve C , $ds = \sqrt{dx^2 + dy^2}$, the modified central moments can be similarly defined as

$$\mu_{pq} = \int_C (x - \bar{x})^p (y - \bar{y})^q ds, \quad \text{where } \bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}. \quad (2)$$

Eq. (2) can be represented as a discrete form

$$\mu_{pq} = \sum_{(x,y) \in C} (x - \bar{x})^p (y - \bar{y})^q. \quad (3)$$

It can be easily verified that the central moments up to the order $p + q \leq 3$ could be computed as

$$\begin{aligned}
\mu_{00} &= m_{00}, & \mu_{11} &= m_{11} - \bar{y}m_{10} \\
\mu_{10} &= 0, & \mu_{30} &= m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2 m_{10} \\
\mu_{01} &= 0, & \mu_{12} &= m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2 m_{10} \\
\mu_{20} &= m_{20} - \bar{x}m_{10}, & \mu_{21} &= m_{21} - 2\bar{x}m_{11} - \bar{y}m_{02} + 2\bar{x}^2 m_{01} \\
\mu_{20} &= m_{02} - \bar{y}m_{01}, & \mu_{03} &= m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01}.
\end{aligned} \tag{4}$$

The central moments are invariant to translation, and can also be normalized to a scaling change by the following formula.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \text{ where } \gamma = \frac{p+q}{2} + 1, \text{ for } p+q = 2, 3, \dots \tag{5}$$

The quantities in equation (5) are called normalized central moments. The following moment invariants were frequently used as features of shape recognition.

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02}, & \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2, & \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2
\end{aligned} \tag{6}$$

The quantities ϕ_i , $1 \leq i \leq 4$, were shown to invariant to scaling, translation, and rotation. The modified moment invariants are computed based on the shape boundary only, therefore, we call them improved moment invariants.

3. FAST IMAGE ALIGNMENT

In this section, a fast image alignment algorithm, integrated connected component labeling algorithm with improved moment invariants, is proposed. The connected component labeling combines run length encoding (RLE) and array-based *Union Find* algorithm to increase the computational efficiency. After the labeling process, the boundary of object is detected using a contour tracing technique. The improved moment invariants, based on the shape boundary, are presented to detect the specified object. The improved moment invariants technique reduces the computation loading because it only uses the boundary information of object and it is also invariant to scaling, translation, and rotation. The proposed fast image alignment algorithm can divided into training phase and matching phase. Figure 8 shows the procedures of these two phases.

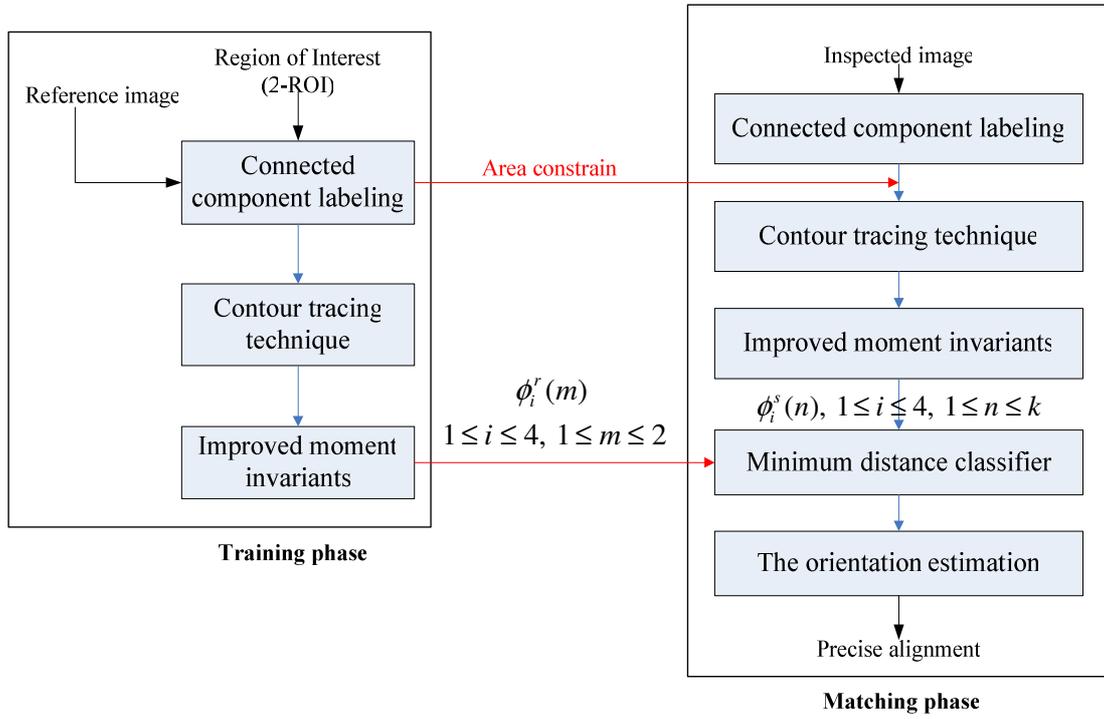


Fig. 8. The fast image alignment algorithm.

3.1 Training Phase

The training phase is based on the reference image, and procedures of training phase are as follows.

- Step1: Selecting two regions of interest (ROI) in reference image.
- Step2: Selecting an appropriate threshold for the different inspection conditions.
- Step3: The proposed connected component labeling is used to label the objects in above ROIs, and the labeled object areas are output for the matching phase.
- Step4: After labeling process, the contour of the specified object is traced by using the contour tracing technique.
- Step5: Here, the four improved moment invariants $\phi_i^r(m)$ ($i = 1, 2, 3, 4; m = 1, 2$) for the two training patterns detected in the reference image are calculated. Finally, the improved moment invariants of selected shapes are output for the matching phase.

3.2 Matching Phase

The procedures of the matching phase based on inspected image are as follows.

- Step1: The proposed connected component labeling is directly used to label the objects. A predetermined threshold is embedded to transfer the gray level image into binary image during the labeling process in order to save the time consumed for image binarization.
- Step2: The contours of the labeling objects are traced under the area limitation, and then the improved moment invariants of the contours are computed.

Step3: The minimum distance classification is evaluated for shape localization using the improved moment invariants. Let $\phi_i^s(n)$ ($i=1, 2, 3, 4; n=1, 2, \dots, k$) denote the four moment invariants of the k objects detected in the inspected image. Then, the normalized distance of invariant-moment between the referenced objects m and inspected objects n is depicted as

$$d_{mn} = \sqrt{\sum_{i=1}^4 \left[\frac{\phi_i^r(m) - \phi_i^s(n)}{\phi_i^r(m)} \times 100 \right]^2}. \quad (7)$$

The result of Eq. (7) is a $2k$ distance matrix in 4-D feature space, which represents the similarity relationship between the objects m and n . Using the principle of minimum distance classifier, the target object is selected when the smallest distance is met between the training patterns and the shapes in the inspected image. In this paper, two optimal objects are selected in the inspection image in order to estimate the orientation of image.

Step4: Finally, the orientation angle between the reference and inspected image is computed. Figure 9 represents the geometric relationship between reference image 1 and inspected image 2, where shape 1 and 2 are the corresponding left and right targets. The vectors between the geometric centers of left and right targets in reference and inspected images are depicted as

$$\vec{P}_1 = (x_{1R} - x_{1L}, y_{1R} - y_{1L}) = (\Delta x_1, \Delta y_1) \quad (8)$$

$$\vec{P}_2 = (x_{2R} - x_{2L}, y_{2R} - y_{2L}) = (\Delta x_2, \Delta y_2) \quad (9)$$

, respectively. And $x_{1R}, x_{1L}, y_{1R}, y_{1L}$ are the horizontal and vertical coordinates of reference image; $x_{2R}, x_{2L}, y_{2R}, y_{2L}$ are the horizontal and vertical coordinates of inspected image. Then, the orientation angle between the reference and inspected image is computed as

$$\begin{aligned} \theta &= \cos^{-1} \left(\frac{\vec{P}_1 \cdot \vec{P}_2}{|\vec{P}_1| \cdot |\vec{P}_2|} \right) \\ &= \cos^{-1} \left(\frac{\Delta x_1 \cdot \Delta x_2 + \Delta y_1 \cdot \Delta y_2}{\sqrt{(\Delta x_1)^2 + (\Delta y_1)^2} \cdot \sqrt{(\Delta x_2)^2 + (\Delta y_2)^2}} \right) \end{aligned} \quad (10)$$

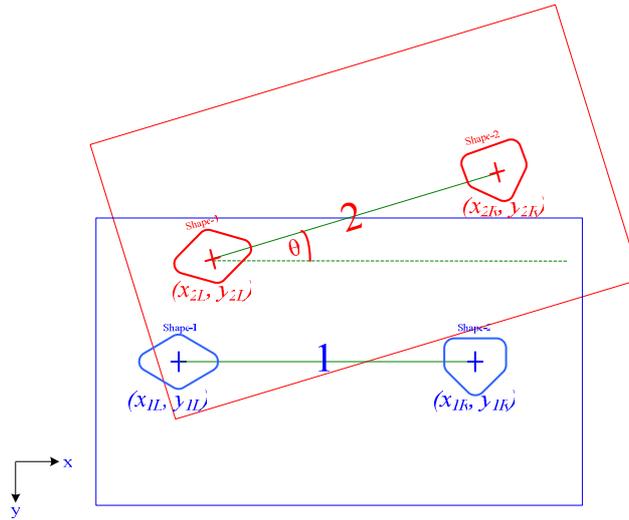


Fig. 9. The orientation estimation procedure.

4. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental results contain two parts. In the first part, the proposed component-labeling algorithm is compared with the other three algorithms which are a recently published algorithm [10] (Chang's algorithm) and two commercial software, *Mil* and *eVision*. In the second part, the proposed image alignment algorithm is compared with the *Easyfind* tool of *eVision*. The test platform is a PC equipped with P4-2.8G CPU and 1G MB RAM.

4.1 Testing on Connected Component Labeling Algorithm

In the experiments, the LCD driver IC images with 2 different resolutions (1000×1000 and 1360×1024) were used, shown in Fig. 10. Table 1 indicates the comparison datum in detail and Fig. 11 shows the performances between our proposed algorithm and the other algorithms. In the table and figure, the density means the ratio of labeled objects to the full image with number of pixel. No matter what image resolution is, our algorithm is more efficient than the other three algorithms. Moreover, Fig. 12 shows the execution time is almost same in both our proposed algorithm and *Mil* 5.12 when the image size increased.

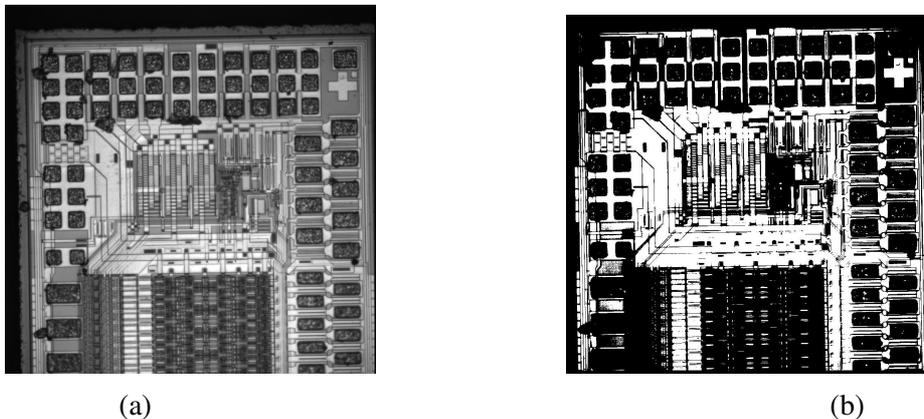
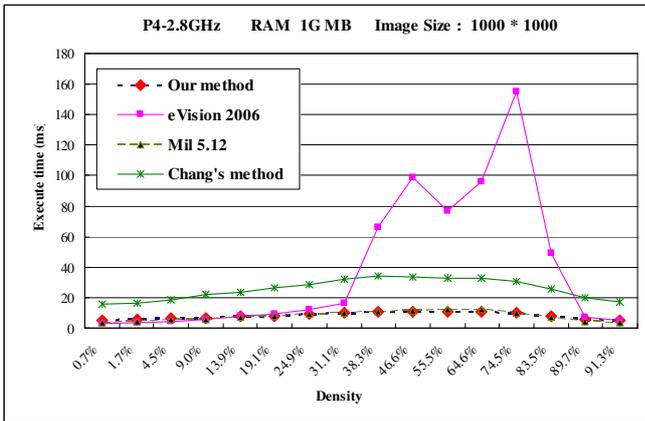


Fig. 10. (a) The LCD driver IC image (b) The driver IC image after image binarization.

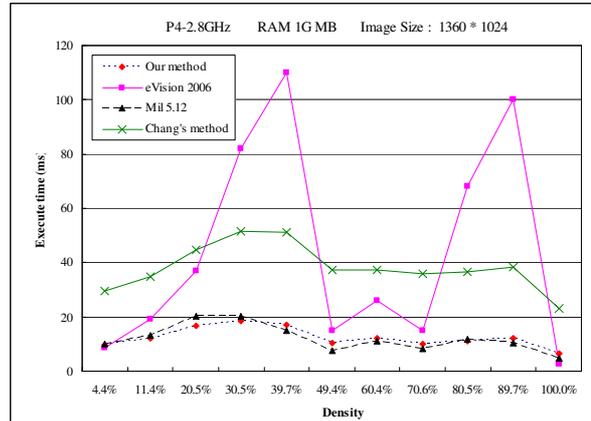
Table 1. Performance of the four algorithms.

P4-2.8GHz		RAM 1G	Image Siz 1k*1k		Execute Time (ms)
Density	New method	Evison 2006	Mil	Chang method	
0.7%	4.789	2.31	3.48	15.944	
1.7%	5.36	3.211	3.93	16.286	
4.5%	6.423	4.351	5.63	18.791	
9.0%	6.511	5.457	5.98	21.783	
13.9%	7.859	8.066	7.06	23.726	
19.1%	8.148	9.368	8.04	26.231	
25.0%	9.011	12.034	9.05	28.623	
31.0%	10.309	16.643	10.24	31.732	
38.3%	10.631	66	10.98	34.048	
46.6%	10.692	99	11.2	33.561	
55.5%	10.971	77	12.06	32.656	
64.6%	10.754	96	12	32.469	
74.5%	9.701	155	10.02	30.515	
83.5%	7.798	49	7.39	25.872	
89.7%	5.415	7.349	4.3	19.661	
91.3%	4.817	5.281	3.75	17.376	

P4-2.8GHz		RAM 1G	Image Siz 1360*1024		Execute Time (ms)
Density	New method	Evison 2006	Mil	Chang method	
4.4%	9.592	8.417	9.82	29.358	
11.4%	12.053	19	13	34.885	
20.5%	16.656	37	20.14	44.911	
30.5%	18.476	82	20.29	51.408	
39.7%	16.942	110	15.01	51.295	
49.4%	10.149	15	7.59	37.373	
60.4%	11.944	26	10.84	37.319	
70.6%	9.855	15	8.16	35.817	
80.5%	10.912	68	11.55	36.426	
89.7%	12.155	100	10.47	38.239	
95.5%	6.331	2.655	4.47	23.112	



(a)



(b)

Fig. 11. Performances of the four algorithms for the (a) 1000x1000 (b) 1360x1024 driver IC images .

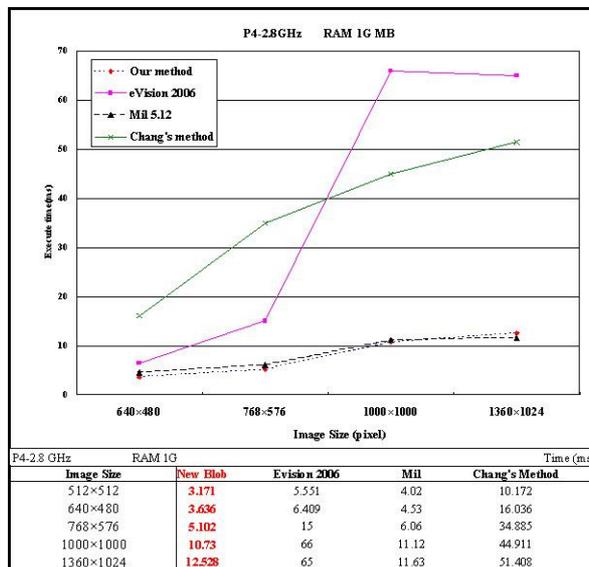


Fig. 12. Performances of the four algorithms for the different image sizes.

4.2 Fast Image Alignment Experiment

In fast image alignment experiment, the simulated and real images are prepared to verify the alignment algorithm. The simulated alignment experiments demonstrate the efficiency of the proposed image alignment algorithm under different simulated translation and rotation conditions. Then, the real images acquired from CCD camera with translation and rotation variations are prepared to validate the robustness of the proposed algorithm. The proposed image alignment algorithm is compared with the *Easyfind* tool of *eVision* in the above experiments.

4.2.1 The simulated alignment experiment

The synthesized images with different translation and rotation are used to verify the proposed image alignment algorithm. There are three original images, depicted in Fig. 13, with different image size 768×576, 1024×768 and 1360×1024, respectively. For each dataset, we simulate 13 images with different translation and rotation values from the original image. Table 2 shows that the proposed algorithm gets more accurate alignment errors and the faster computing time than *Easyfind* under different image size.

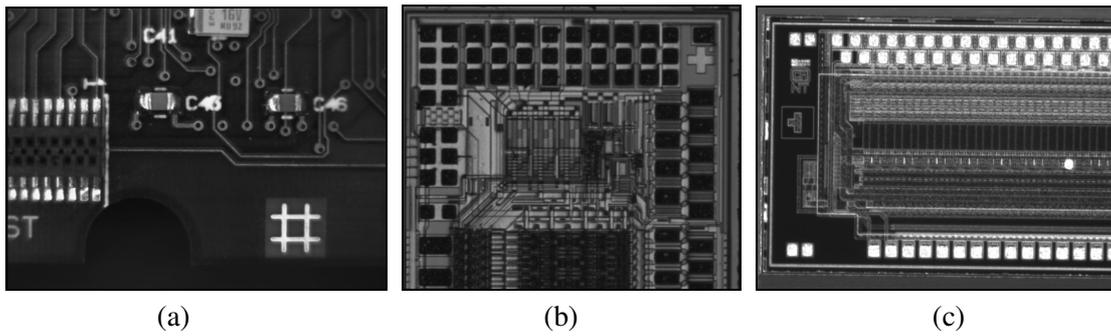


Fig. 13. The three images with different image size. (a) 768×576. (b) 1024×768. (c) 1360×1024.

Table 2. Performance of the simulated image alignment algorithms.

P4-2.8 GHz		RAM1G		Image Size 768*576	
Rotation angle	Proposed Method		Evision-Easyfind		
	Error angle(°)	Execute time(ms)	Error angle(°)	Execute time(ms)	
1°	0.005	3.81	0.221	7.77	
2°	0.011	3.65	0.061	8.538	
3°	0.012	3.42	0.008	8.53	
4°	0.015	3.88	0.061	8.53	
5°	0.023	3.65	0.042	8.2	
6°	0.002	3.68	0.041	8	
7°	0.01	3.62	0.051	8	
8°	0.011	3.58	0.046	8.2	
9°	0.015	3.68	0.039	8.5	
10°	0.03	3.63	0.01	8	
15°	0.017	3.69	0.009	7	
20°	0.033	3.52	0.049	7.24	
25°	0.019	3.6	0.05	6.5	
Average	0.01562	3.64692	0.05292	7.92369	

P4-2.8 GHz		RAM1G		Image Size 1024*768	
Rotation angle	Proposed Method		Evision-Easyfind		
	Error angle(°)	Execute time(ms)	Error angle(°)	Execute time(ms)	
1°	0.026	6.18	0.102	11	
2°	0.021	6.54	0.035	11	
3°	0.024	6.49	0.019	11	
4°	0.035	6.33	0.009	11	
5°	0.033	6.53	0.06	11	
6°	0.038	6.52	0.032	11	
7°	0.028	6.51	0.051	12	
8°	0.04	6.68	0.013	11	
9°	0.037	6.48	0.043	11	
10°	0.016	6.51	0.018	11	
15°	0.034	6.51	0.081	11	
20°	0.026	6.1	0.096	11	
25°	0.029	6.26	0.053	12	
Average	0.02977	6.43305	0.04708	11.1538	

P4-2.8 GHz		RAM1G		Image Size 1360*1024	
Rotation angle	Proposed Method		Evision-Easyfind		
	Error angle(°)	Execute time(ms)	Error angle(°)	Execute time(ms)	
1°	0.046	11.2	0.168	20	
2°	0.022	11.33	0.08	19	
3°	0.004	11.25	0.03	19	
4°	0.042	11.64	0.001	20	
5°	0	11.27	0.045	19	
6°	0.015	11.4	0.006	19	
7°	0.015	11.24	0.039	20	
8°	0.018	11.4	0.004	18	
9°	0.012	11.28	0.017	19	
10°	0.017	11.79	0.01	20	
15°	0.008	11.4	0.037	18	
20°	0.015	11.51	0.046	17	
25°	0.017	11.57	0.071	20	
Average	0.01777	11.4062	0.04262	19.0769	

4.2.2 The real image alignment experiment

Furthermore, the real images acquired from a CCD camera, shown in Fig. 14, are tested under different translation and rotation. The targets of the cell phone and the IC pattern were installed onto a positioning table that can be rotated with a resolution of 0.01° . Table 3 shows the average errors are smaller than 0.05° by using our alignment algorithm; it is superior to by *Easyfind* tool. The computation time in Table 3 also shows our algorithm is more efficient than *Easyfind* tool.



Fig. 14. The two real images for image alignment.

Table 3. Performance of the real image alignment algorithms.

P4-2.8 GHz		RAM 1G		Example-1		Image Size 1360*1024		P4-2.8 GHz		RAM 1G		Example-2		Image Size 1360*1024	
Rotation angle($^\circ$)	Error angle($^\circ$)	Execute time(ms)	Proposed Method	Error angle($^\circ$)	Execute time(ms)	Evision-Easyfind		Rotation angle($^\circ$)	Error angle($^\circ$)	Execute time(ms)	Proposed Method	Error angle($^\circ$)	Execute time(ms)	Evision-Easyfind	
4.5	0.09	11.78		0.071	22			4.5	-0.08	12.49		-0.034	17		
9	0.134	11.78		0.12	21			9	-0.063	12.74		-0.014	20		
13.5	0.099	12.01		0.087	26			13.5	-0.083	12.17		0.012	19		
18	0.038	11.55		0.026	21			18	-0.101	11.72		-0.078	23		
22.5	0.064	12.15		0.071	13			22.5	-0.038	11.92		-0.024	24		
27	0.101	12.05		0.106	20			27	-0.011	12.07		0.025	24		
31.5	0.064	11.74		0.075	24			31.5	-0.035	12.11		0.022	25		
36	0.004	11.9		0.009	21			36	-0.119	11.89		-0.064	26		
40.5	0.004	11.38		0.041	20			40.5	-0.018	12.45		-0.04	31		
45	0.057	11.87		0.067	19			45	-0.017	12.34		0.054	42		
49.5	0.015	12.02		0.034	26			49.5	-0.047	12.64		0.043	38		
54	-0.044	12.87		-0.019	24			54	0.157	12.06		-0.013	40		
58.5	-0.016	11.87		0.038	26			58.5	0	12.98		0.025	38		
63	0.047	11.94		0.053	26			63	0.018	12.85		0.041	39		
67.5	0.031	11.9		0.028	29			67.5	-0.008	11.96		0.066	48		
72	0.022	11.62		-0.024	28			72	-0.071	12.32		-0.005	53		
76.5	0.049	12.05		0.003	28			76.5	-0.023	12.08		0.04	43		
81	0.104	11.49		0.033	31			81	0.016	12.29		0.129	37		
85.5	0.079	11.88		0.015	25			85.5	0.062	12.74		0.094	42		
90	0.04	12.33		0.135	16			90	-0.01	12.79		0.084	42		
Average	0.0491	11.909		0.04845	23.3			Average	-0.02355	12.3905		0.01815	33.55		

5. CONCLUSION

In this paper, a fast image alignment algorithm based on connected component labeling and improved moment invariants resulting in rotation, scale and translation invariant is presented. The new connected component labeling algorithm combines a novel RLE and *Union Find* algorithm to increase the computational efficiency. After the labeling process, the boundary of object is detected using a contour tracing technique. To reduce the computation of the traditional moment invariants, the improved moment invariants are computed using the shape boundary only. The proposed fast image alignment algorithm consists of a training phase and a matching phase in this paper. In training phase, the algorithm firstly finds the moment invariants of selected shape in the reference image by using the proposed connected component labeling algorithm, contour tracing technique, and improved moment invariants technique. Subsequently, an efficient pattern matching which is used the normalized minimum distance classifier is applied in matching

phase. Finally, the orientation angle between the reference and inspected images is computed. Experimental results show that the proposed connected component labeling algorithm and fast image alignment algorithm is higher speed and accurate than the other algorithms and commercial software. Therefore, they could be preprocessing tools used for the real-time industrial inspections.

6. REFERENCES

- [1] B. Zitova, J. Flusser (2003), "Image registration methods: a survey," *Image Vision Computing*, Vol. 21, pp. 977-1000.
- [2] Y. Y. Tang, H. D. Cheng and C. Y. Suen (1991), "Transformation-Ring-Projection algorithm and its VLSI implementation," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 5, NO. 1&2, pp. 25-56.
- [3] D.M. Tsai and C.H. Chiang (2002), "Rotation-invariant pattern matching using wavelet decomposition," *Pattern Recognition Letters*, Vol. 23, pp. 191-201.
- [4] M.S. Choi and W.Y. Kim (2002), "A novel two stage template matching method for rotation and illumination invariance," *Pattern Recognition*, Vol. 35, pp. 119-129.
- [5] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge (1993), "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, NO. 9, pp. 850-863.
- [6] D. P. Huttenlocher, R. H. Lilien, and C. F. Olson (1999), "View-based recognition using an eigenspace approximation to the Hausdorff measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, NO. 9, pp. 951-955.
- [7] O. K. Kwon, D. G. Sim, and R. H. Park (2001), "Robust Hausdorff distance matching algorithms using pyramidal structures," *Pattern Recognition*, Vol. 34, NO. 10, pp. 2005-2013.
- [8] C. J. Chen, S. H. Lai, S. W. Liu, T. Ku, and S. Y. C. Yeh (2005), "Optical PCB inspection system based on Hausdorff distance," *Machine Vision Applications in Industrial Inspection*, Vol. 5679, pp. 53-61.
- [9] K. Suzuki, I. Horiba, and N. Sugie (2003), "Linear-time connected-component labeling based on sequential local operations," *Computer Vision and Image Understanding*, Vol. 89, pp. 1-23.
- [10] F. Chang, C. J. Chen, and C. J. Lu (2004), "A linear-time connected-component labeling using contour tracing technique," *Computer Vision and Image Understanding*, Vol. 93, pp. 206-220.
- [11] C. Fiorio, and J. Gustedt (1996), "Two linear time Union-Find strategies for image processing," *Theoretical Computer Science*, Vol. 154, pp. 165-181.

- [12] K. Wu, E. Otoo, and A. Shoshani (2005), "Optimizing connected component labeling algorithms," *Medical Imaging 2005: Image Processing*, Vol. 5747, pp. 1965-1976.
- [13] C. C. Chen (1993), "Improved moment invariants for shape discrimination," *Pattern Recognition*, Vol. 26, NO. 5, pp. 683-686.