Research article

An image processing approach for jigsaw puzzle assembly

Ni Fei, Fu Zhuang, Liu Renqiang, Cao Qixin and Zhao Yanzheng
State key laboratory of Vibration, Shock and Noise at Shanghai JiaoTong University, Shanghai, People's Republic of China

Abstract
Purpose – To develop an image processing approach for jigsaw puzzle assembly.
Design/methodology/approach – First, pixels are extracted from the jigsaw puzzle blocks to calculate their rotation angles and centre coordinates. Second, a template matching method is employed to recognise each block and its orientation.
Findings – A robot-based jigsaw puzzle system is established; and an effective image processing approach for assembly is developed.
Practical implications – Automatic assembly lines that assemble parts with the same shape, but random position and angle, can employ the jigsaw puzzle assembly method.
Originality/value – An effective image processing method for jigsaw puzzle assembly is presented in this paper. The validity of the method is proved by analysis and experiment.

Keywords Assembly, Image processing, Robotics

Paper type Research paper

1. Introduction

Vision recognition techniques have been widely employed in assembly lines (Mario Pena-Cabrera, 2005). In many cases, the assembly system handles parts with the same shape, but at random positions and angles. For example, in clothing manufacture, the cloth must be automatically delivered to different sewing positions according to its textures and patterns. Generally, it is more difficult to recognise these parts with the same shape because of the lack of easily-distinguishable information. In order to assemble parts with same form, but random position and angle, the identity, position and orientation angle of every part must be detected. Once this information is obtained, the manipulator can pick up the parts and place them in the assigned positions. For this purpose, an effective image processing method must be developed. The assembly system is shown in Figure 1.

In Figure 1, the disordered parts are placed on the disordered parts pallet (1). Using image processing, the six degree of freedom robot (3) can pick up these parts one by one using a suction cup, and then place them (4) at assigned positions on the assembled parts pallet (2) according to their identities.

Various effective image-processing methods have been proposed for assembly systems. Qiang, *et al.* (2004) presented a centre-invariable moment technique. In the image, edges are detected by two-value processing, the profiles are connecting by closed curves, and then the centre of each curve is found in order to calculate the centre moment within the curves. Obviously, parts with different shapes have different centre moments. This method is appropriate for assembling parts with different shapes, but it does not work well for parts with same shape. Alternatively, a grey-scale histogram can be employed. (Fan-Di *et al.*, 2004). The method is described as follows: first, a grey-scale histogram of every part is established, called the template grey level histogram. Second, pixels of every part are extracted from the image, and their grey-scale histograms are established. Finally, a comparison of the grey level histogram of these parts with every template histogram identifies the parts. This method is a fast and accurate way to identify parts, but it does not recognise a part's orientation angle very well. A further method is the XY-Theta system presented by Barrett *et al.* (1996): during clothing manufacture, according to their texture and pattern, different pieces of cloth with the same shape must be automatically delivered to different sewing points by manipulators and transporters. Generally in this paper, template matching is based on monochrome imaging, and the system finds the same position repeatedly. It is not necessary to consider the position of every part in the image. Moreover, the cloth has been stacked in a consistent direction, simplifying the recognition of its orientation. Another method called the orientation code (Ullah and Kaneko, 2004) has been presented, and can be employed for template matching when the blocks have random orientation. Over 360° in either direction, it attempts to match the current grey level histogram with the

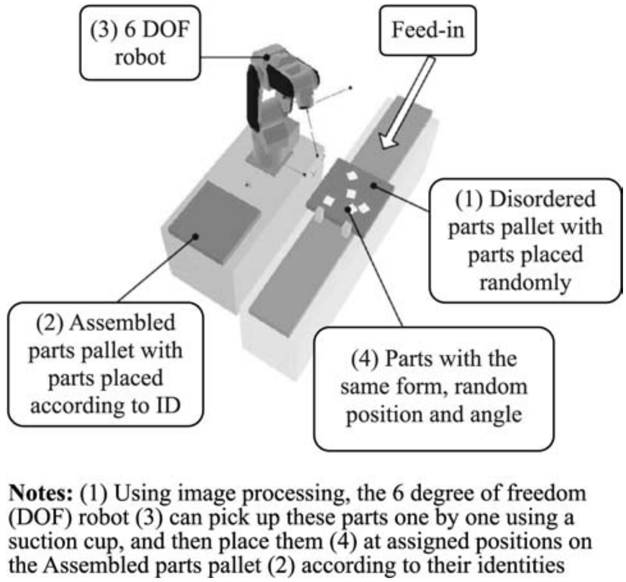
The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0144-5154.htm

Assembly Automation
27/1 (2007) 25–30
 [ISSN 0144-5154]
[DOI 10.1108/01445150710724676]

This work was partially supported by the National Natural Science Foundation of China under grant No. 60304010 and the Start-up Grant for Young Teachers from the Shanghai Jiao Tong University.

Assembly Automation
Volume 27 · Number 1 · 2007 · 25–30

Figure 1 The assembly system for parts with the same form, at random positions and angles



every template's grey level histogram, and this gives the identities and the orientation angles of parts in the image, but requires the parts to be in fixed positions. Choi and Kim (2002) presented a two-stage template matching method: in the first stage, the matching candidates are selected using a computational low cost feature. In the second stage, rotation-invariant template matching is performed only on the matching candidates using Zernike moments. The method can deal with the recognition of fixed position. A comparison of these above methods is shown in Table I.

Evidently, there is no effective image processing method for the assembly of parts with the same form, random position and angle. Looking at the assembly process, we find it is similar to jigsaw puzzle assembly. This paper has established a robot-based jigsaw puzzle assembly system, which includes up-to-date hardware and software techniques.

Table I Advantages and shortcomings of related method

Method	Advantages	Disadvantages
Centre invariable moment	Assembles parts with random position and different forms	Cannot assemble parts with the same form and different appearance
Grey level histogram	Rapidly calculates, and assembles parts with the same forms	Cannot assemble parts with random orientation angles
XY-Theta system method	Assembles parts with the same form and different appearance	Cannot assemble parts with random position and orientation.
Orientation code	Assembles parts with the same form and random orientation	Large computation cost. Cannot assemble parts with random position

2. Framework of system

The robot-based jigsaw puzzle system is shown in Figure 2. The robot picks up the block by means of the suction cup from the disordered parts pallet, and then rotates it to the appropriate angle, and places it at an assigned location in the assembled parts pallet. In our system, the jigsaw puzzle is made up of 25 square blocks. Following a sequence from right to left and top to bottom, the identities (IDs) of the blocks are denoted as 1,2, . . . ,25.

The flow chart of the software system is shown in Figure 3.

Figure 2 The robot-based jigsaw puzzle system

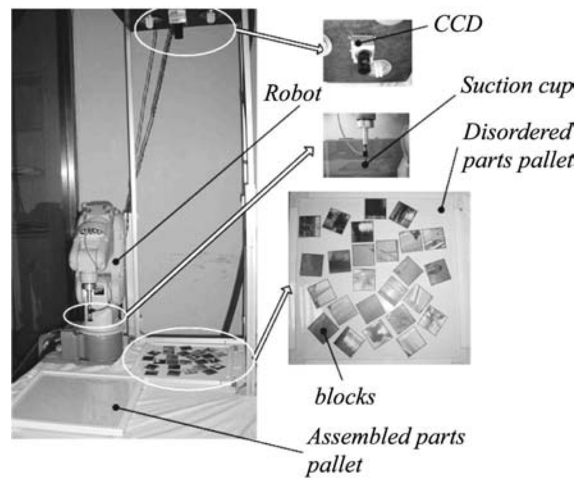
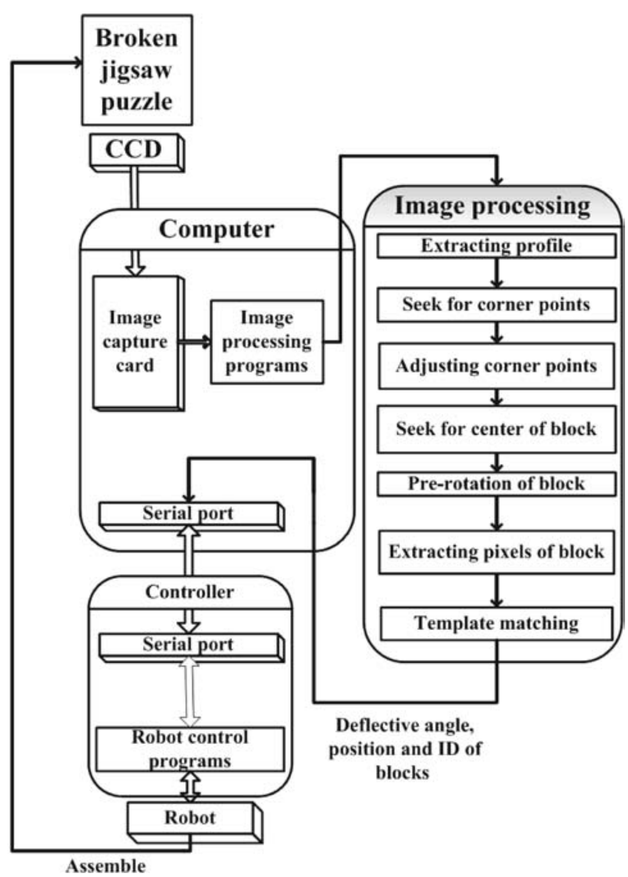


Figure 3 Flow chart of the jigsaw puzzle system



Assembly Automation
Volume 27 · Number 1 · 2007 · 25–30

3. Obtaining pixels and pre-rotation angle

First of all, the block template T_i (i = 1,2,...,25) must be created. The template T_i is a 71 × 71 matrix, in which every element denotes the grey-scale values of pixels in the *i*th block. To extract the profile of each block, grey transformation and threshold division algorithms are employed. Under 300 lumens, blocks are easily recognised when the threshold value is 250-320. In the image the block dealt with is defined as the current block, and when the current block recognition is completed the next block becomes the current block. In a two-value image, beginning with the first black point, the current block's edge points are extracted. The coordinates of these edge points are recorded as (X_e, Y_e), where, the *i*th edge point is denoted as $(x_e[i], y_e[i])$. Some noise is present in the form of small black blocks or dots, but it is easy to ignore these in the analysis of a number of points on the boundary: $(x_e[i], y_e[i])$.

3.1 The corner points and the centre coordinates

Using the corner points of the square block (Figure 4), the centre coordinates and the pre-rotation angle of the current block are calculated. If the pre-rotation angle is not too small, the top corner point (y_{ctop}, y_{ctop}) is the highest point (x_{top}, y_{top}) with minimum y coordinates in y_e:

$$y_{ctop} = y_{top} = \min(y_e) \tag{1}$$

$$x_{ctop} = x_{top} = x_e[p] \tag{2}$$

where, p is the position of y_{top} in y_e.

In the same way, we can find the bottom, left and right corner point coordinates $(x_{cbottom}, y_{cbottom})$, (x_{cleft}, y_{cleft}), and (x_{cright}, y_{cright}). Hence, the coordinates of the centre point (x_{mid}, y_{mid}) are computed by the following formula:

$$\left(\frac{x_{cright} + x_{cleft}}{2}, \frac{y_{ctop} + y_{cbottomt}}{2}\right) \tag{3}$$

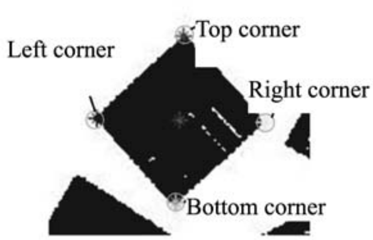
The centre coordinates can be transformed to the robot coordinates system, so that the robot will know the positions of the blocks. This transformation will be discussed in section 5. The pre-rotation angle α_b denotes the angle of clockwise rotation for the current block relative to the edge of the original image. It can be worked out by following equation:

$$\alpha_b = a\tan\frac{y_{cright} - y_{ctop}}{x_{ctop} - x_{cright}} \tag{4}$$

where $x_{ctop} \neq y_{cright}$.

Generally, the pre-rotation only rotates the blocks' edge parallel to the pallet edge. Later, a compensation angle is applied to rotate the blocks to their correct orientation (Figure 6). Section 4 will explain how to find the compensation angle.

Figure 4 The corner points of the square block



3.2 Correcting algorithm for the corner points

When the pre-rotation is small, the highest point is not usually the top corner point (Figure 5), so the identification of the top corner point is difficult. Obviously, we need an effective approach for adjusting the corner points. Assuming that the coordinates of the current point $(x_{current}, y_{current})$ are $(x_e[1], y_e[1])$, the following steps will find the correct top corner point;

1 Estimate whether it is necessary to adjust the corner points. First, calculate the distance in y coordinates between the current point and the preceding tenth point, $|y_{current} - y_{current+10}|$ and distance in y coordinates between the current point and its following tenth point $|y_{current} - y_{current-10}|$. If equations (5) and (6) are satisfied, the current point is not the corner point.

$$|y_{current} - y_{current+10}| < 5 \tag{5}$$

$$|y_{current} - y_{current-10}| < 5 \tag{6}$$

So a search for the top corner point is required.

2 In y_e, extract all $y_{current}$ values.

3 For these points, find the left point with minimum x coordinates and the right point with maximum x coordinates. Now, regard the left point as the current point. If the current point satisfies equation (5), the left point is not the top corner point, otherwise it is. Likewise, we can estimate whether the right point is the top corner point or not according to equation (6).

4 After completing the above steps, if the top corner point is still not found, move to the y coordinates of the current point plus 1: $y_{current} = y_{current} + 1$. Then, extract all points whose y coordinates are $y_{current}$. For these points, take an arbitrary point as the current point, and then repeat steps 1-4 until the top corner point is found. Figure 5 shows the top corner point (blue) after this process.

Once the top corner point has been found, the furthest point away from the top corner is the bottom corner point. Then, the point with the smallest x coordinates is the left corner point. In a similar way, we can find the right corner point. When all the corner points are calculated, equation (4) can be used to calculate the correct pre-rotation angle.

3.3 Pixel extraction

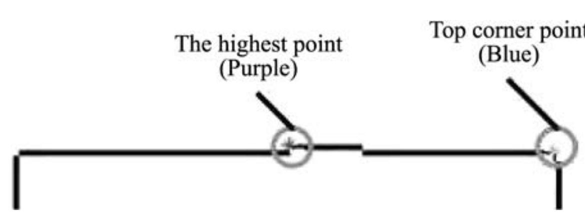
In order to match the current block with the template T_i, we employ the following approach to extract the pixels of the current block in the original image.

For the edge points, set (X_e, Y_e), to find all points p_j (x_j, y) with the same y coordinates, such as the vector:

$$Y_k = (p_1, p_2, \ldots, p_j \ldots, p_n)(k \in [\mathrm{Min}, \mathrm{Max}]) \tag{7}$$

where Min and Max are the minimum and maximum values in Y_e, respectively. In original image, extract and store all

Figure 5 The highest point is not the top corner point (blue) when pre-rotation is small



Assembly Automation
Volume 27 · Number 1 · 2007 · 25–30

pixels between the left point $p_1(x_1,y)$ with the minimum x and the right point p_n (x_n,y) with the maximum x. Then move to $y+1$, and repeat the above operation until k increases from Min to Max. Thus, all the pixels of the current block are identified. These pixels are rotated clockwise through the pre-rotation angle (Figure 6). Thus, 71 × 71 pixels are extracted, denoted by P_h ($h = 1,2,\ldots,25$), in which every element corresponds to a grey level of a pixel at the corresponding position of the current block. In the original image, the first current block is extracted as P_1, and then all pixels of the current block are set to white. Then the next block becomes the current block, defined as P_2, and so on until P_{25} is extracted.

4. The matching algorithm

After obtaining all P_h, we can test their match against every template $T_i(i = 1,2,\ldots,25)$. First, it is necessary to calculate the compensation angle. By rotating the P_h through one of four compensation angles α_a(0° 90° 180° 270°) as shown in Figure 6, the current block will be rotated to its correct position. The equation for matching P_h with T_i is as follow (Jen-Hui Chuang, 1996):

$$r_{\alpha_a} = \frac{\Sigma T_i(j,k) \times P_h(j,k)}{\sqrt{\Sigma T_i(j,k)^2}\sqrt{\Sigma P_h(j,k)^2}} \tag{8}$$

where, $j,k = 1,2,3,\ldots,71$, r_{α_a} denotes the distance between the hth block P_h and the ith template T_i. Obviously, to match the hth block with the ith template at four positions will result in four matching values: $r_0, r_{90}, r_{180}, r_{270}$. In the four values, choose the minimum one and denote it as r, then put r into a 25 × 25 matrix C called the minimal matching matrix at (h,i). At the same time, put the angle α_a corresponding to value r into a 25 × 25 matrix A called the angle matrix at (h, i). In other words, the hth block that has been rotated clockwise through the pre-rotation angle is closest to the ith template that has been rotated clockwise α_a. It is necessary to match the current block with every template, and put a minimum value and angle into C and A, respectively.

Having matched each $\mathrm{P}_h(i = 1,2,\ldots,25)$ with the respective template, matrixes A and C will be full. In matrix C, seek the minimum value. If, for instance, the position of the minimum value is at the hth row and the ith column, this means that in original image, the ID of the hth block is i. So the block should be placed at the ith position. The following equation calculates the compensation angle:

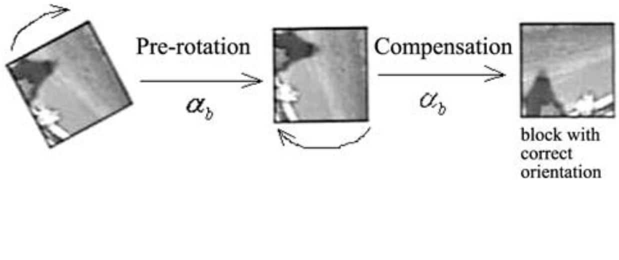
$$\alpha_a = \underbrace{(0,0,\ldots,1}_{h},0,0)\mathrm{A}\underbrace{(0,0,\ldots,1}_{i},0,0)^{\mathrm{T}} \tag{9}$$

So, the orientation angle α of the hth block is:

$$\alpha = \alpha_{\mathrm{a}} + \alpha_{\mathrm{b}} \tag{10}$$

Then, delete the row and the column in which the minimal value is located. Repeat this 25 times to obtain the ID and orientation angle of every block. Finally, on delivering the information to the robot controller, the robot will pick up the blocks by a suction cup, rotate the blocks by the orientation angles, and place them at the proper positions in the assembled parts pallet.

Figure 6 Rotation of block to its correct orientation



5. Experiments

The experimental system is shown in Figure 2. The camera, which is set at a resolution of 768 × 576, captures images of the disordered blocks on the disordered parts pallet and saves the picture in computer memory. The disordered pallet and the assembled parts pallet are both square, and their side lengths are 350 mm. Using the image processing method mentioned above, the robot picks up a block by a suction cup from the disordered parts pallet, and then rotates it through the appropriate angle, and places it at an assigned location on the assembled parts pallet.

Before the experiment, the system must be calibrated to relate the whole disordered parts pallet correctly to the camera's field of view. Then, in the image captured by the camera, take a corner of the pallet as reference, count the number of pixels N between the reference point and the neighbouring corner point. A transform coefficient E can be derived:

$$E = L/N \tag{11}$$

where L is the length of the pallet side. Thus, in the image, the distance: DE between a given point and the reference point can be calculated by:

$$\mathrm{DE} = E \times \mathrm{NE} \tag{12}$$

where NE is the number of pixels between the given point and the reference point. So the centre coordinates of the blocks in the image can be transformed to robot coordinates according to equation (12).

Once calibration is completed, the experiments are performed. During the robot assembly process, some errors in position and angle emerged. There are two reasons for these errors. Firstly, the calibration transform coefficient actually varies over the surface of the disordered parts pallet. Secondly, the coefficient is affected by the distance D between the disordered parts pallet and the CCD camera. Figure 7 shows the position and angle errors of the block located at the top right-hand corner of the disordered parts pallet as D varies. Obviously, a closer distance means more deformation at the edges of the image, and a farther distance means fewer pixels per block. According to Figure 7, to minimise the error the distance D must be about 950 mm. Blocks distant from the centre have larger position and angle of errors because of the physical characteristics of the CCD camera. Figure 8 plots the position and angle errors of blocks located at different positions on the disordered parts pallet.

To test our technique, simulations were performed. The broken jigsaw puzzle is shown in Figure 9(a), and the result of assembly is shown in Figure 9(b). If the blocks are placed at small pre-rotation angles, and they are assembled without correcting the corner points, the result is shown in Figure 9(c).

Figure 7 This plot shows the position and angle error of a block located at the top right corner of disordered parts pallet with varying camera distance

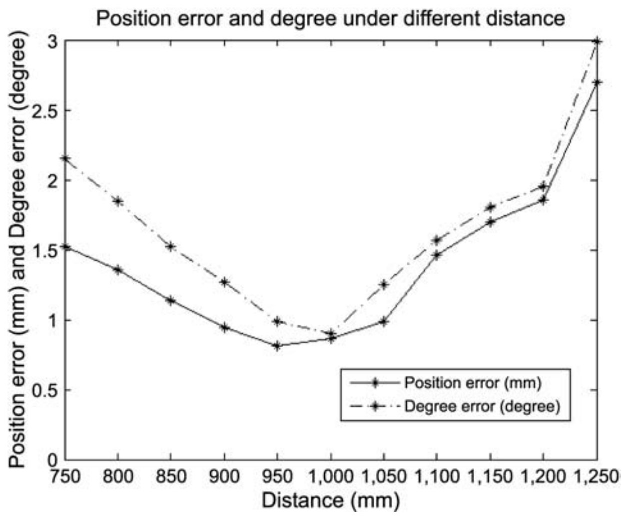
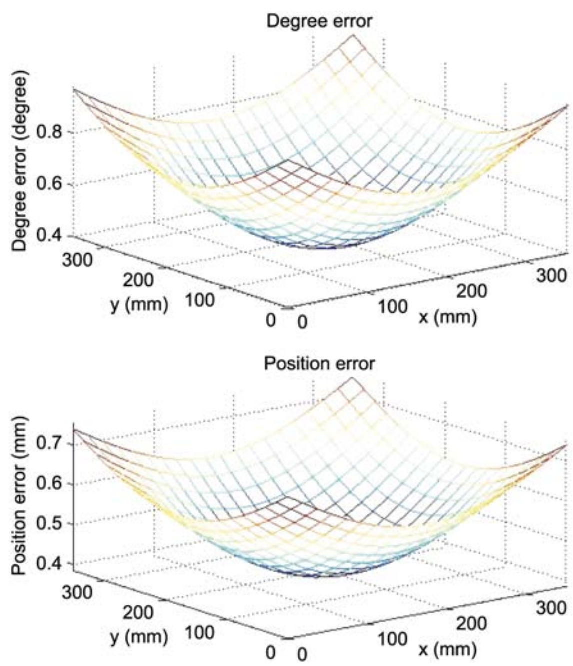


Figure 8 With a distance D of 950 mm, this plot shows the position and angle errors of blocks located at different positions in the disordered parts pallet



By correcting the corner points, the result of assembly is the same as Figure 9(b). Finally, practical experiments were carried out. Using the system shown in Figure 2, successful assembly under a camera distance of 950 mm within 3 min is shown in Figure 10.

6. Conclusion

In this paper, a robot-based jigsaw puzzle assembly system has been established. We developed an effective image processing

Figure 9 Simulation of assembly

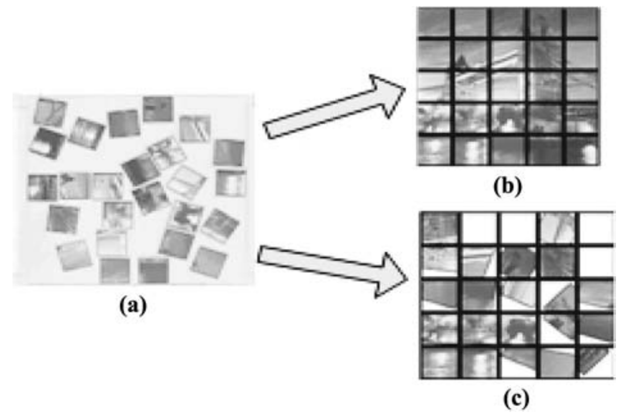
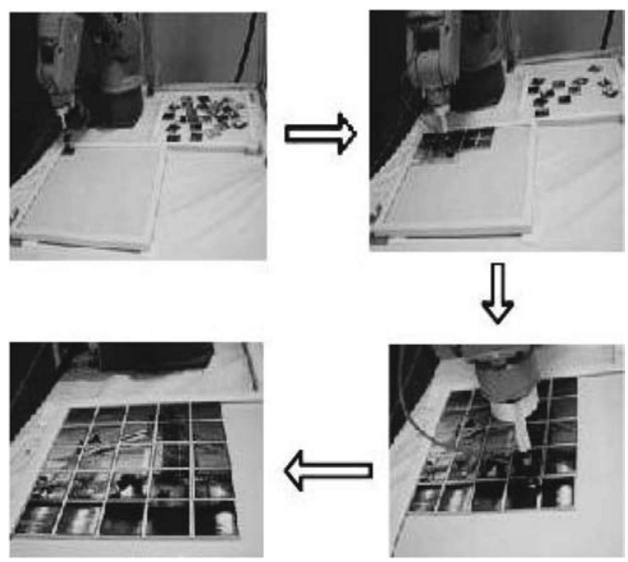


Figure 10 The experimental result of the robot-based jigsaw puzzle assembly



method to assemble parts with the same shape, but random position and angle. Simulation and experimentation showed (Figures 9 and 10) the validity and effectiveness of the method presented in this paper. Future work will concentrate on the recognition of occluded blocks, such as overlapping blocks.

References

Barrett, G.R., Farrington, C.E. and Clapp, T.G. (1996), "An XY manipulator for flexible fabric part positioning", *International Journal of Clothing Science & Technology*, Vol. 8 Nos 1/2, pp. 33-43.

Choi, M.-S. and Kim, W.-Y. (2002), "A novel two stage template matching method for rotation and illumination invariance", *Pattern Recognition*, No. 35, pp. 119-29.

Chuang, J.-H. (1996), "A potential-based approach for shape matching and recognition", *Pattern Recognition*, Vol. 29 No. 3, pp. 463-70.

Assembly Automation
Volume 27 · Number 1 · 2007 · 25–30

Fan-Di, J., Kuo-Chin, F. and Yang-Lang, C. (2004), "Efficient matching of large-size histograms", *Pattern Recognition Letters*, Vol. 25, pp. 277-86.
Pena-Cabrera, M. (2005), "Machine vision approach for robotic assembly", *Assembly Automation*, Vol. 25 No. 3, pp. 204-16.
Qiang, W., Ming, S.J., Ping, H.J. and Wen, Q.Z. (2004), "A rapid recognition approach for template matching", *Engineering and Application of Computer*, Vol. 6, pp. 42-6.
Ullah, F. and Kaneko, S. (2004), "Using orientation codes for rotation-invariant template", *Pattern Recognition*, Vol. 37, pp. 201-9.

Further reading

Caprari, R.S. (2000), "Duplicate document detection by template matching", *Image and Vision Computing*, No. 18, pp. 633-43.
Chen, Y.-S. *et al.*, (2001), "Three dimensional egomotion estimation from motion fields observed with multiple cameras", *Pattern Recognition*, Vol. 34, pp. 1573-83.
Chuang, J.-H., Sheu, J.-F., Lin, C.-C. and Yang, H.-K. (2001), "Shape matching and recognition using a physically based object model", *Computers & Graphics*, Vol. 25, pp. 211-22.
Jorg, S. *et al.* (2000), "Flexible robot-assembly using a multisensory approach", *Proceedings of the IEEE, International Conference on Robotics and Automation, San Francisco, CA*, pp. 282-302.
Saber, E. and Tekalp, A.M. (1997), "Region-based shape matching for automatic imageannotation and query-by-example", *Journal of Visual Communication and Image Representation*, Vol. 8 No. 1, pp. 3-20.
Super, B.J. (2002), "Fast retrieval of isolated visual shapes", *Computer Vision and Image Understanding*, Vol. 85, pp. 1-21.
Wang, A.-B. and Fan, K.-C. (2001), "Optical recognition of handwritten Chinese characters by hierarchical radical matching method", *Pattern Recognition*, Vol. 34, pp. 15-35.
Wu, C.-H. *et al.*, (2001), "A new computation of shape moments via quadtree decomposition", *Pattern Recognition*, Vol. 34, pp. 1319-30.

Corresponding author

Fu Zhuang can be contacted at: zhfu@sjtu.edu.cn