



ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Pattern Recognition Letters 24 (2003) 1819–1835

Pattern Recognition
Letters

www.elsevier.com/locate/patrec

A shape and image merging technique to solve jigsaw puzzles

Feng-Hui Yao ^{a,*}, Gui-Feng Shao ^b

^a General Information Processing Center, Shimane University, 1060 Nishikawatsu-machi, Matsue-shi, Shimane-ken 6908504, Japan

^b Seinan Gakuin University, 6-2-92 Nishijin, Sawara-ku, Fukuoka 8140002, Japan

Received 12 March 2002; received in revised form 4 December 2002

Abstract

This paper proposes an algorithm for solving subsets of typical (canonical) jigsaw puzzles. This algorithm combines shape and image matching with a cyclic “growth” process that tries to place pieces in their correct positions. First, the jigsaw pieces are extracted from the input image. Then, the corner points of the jigsaw pieces are detected. Next, piece classification and recognition are performed based on jigsaw piece models. Connection relationships between pieces are calculated and finally recovered by using boundary shape matching and image merging. We tested this algorithm by employing real-world images containing dozens of jigsaw pieces. The experiment’s results show this algorithm is efficient and effective.

© 2003 Elsevier Science B.V. All rights reserved.

Keywords: Jigsaw puzzles; Boundary shape matching; Image merging; Recovery of connection relationships

1. Introduction

Radack and Badler (1982) originally summarized the two-dimensional jigsaw puzzle problem as: “Given a set of simply connected planar regions (silhouetted puzzle pieces), rotate and translate each piece so that the pieces fit together into one region, with no significant area gaps or overlapping pieces.” However, this summarization made no mention of texture and color information. Here, as a scientific problem, the jigsaw puzzle problem can

be defined as follows: For $S_p = \{P_0, P_1, \dots, P_{N-1}\}$, where P_i represents the i th jigsaw piece ($i \in \{0, 1, \dots, N-1\}$), and has the attributes of a closed boundary and a solid texture; and for P_i , there exists P_j ($i \neq j, i, j \in \{0, 1, \dots, N-1\}$) so that a boundary part of P_i is completely identical with a boundary part of P_j , and the textures and color near these two boundary parts are most similar, that is, P_i and P_j are neighbors (or can be connected), then find the neighbor for P_k ($k \in \{0, 1, \dots, N-1\}$) and connect all pieces in S_p into one large piece. Moreover, Radack and Badler did not take into account that jigsaw puzzle solving can be a group activity (e.g. for children), and not necessarily a solitary pastime.

Solving jigsaw puzzles with a computer involves a number of tasks related to various machine vision

* Corresponding author. Fax: +81-852-32-6178.

E-mail addresses: yao@po.cc.toua-u.ac.jp, yao@ipc.shimane-u.ac.jp (F.-H. Yao).

applications: shape description, partial boundary matching, pattern recognition, feature extraction, and heuristic matching. Consequently, many inspired researchers have taken up the challenge. Freeman and Garder first tackled the problem in 1964, and their work remains fundamental in the field. Because of limitation in computer language, digitizer resolution, and device imaging, they only managed to solve this problem by using the piece boundary shape information. Since then, many algorithms have been proposed. Radack and Badler (1982) proposed a shape representation algorithm capable of efficiently matching boundaries of two-dimensional jigsaw puzzle pieces. This process uses a polar coordinate system located at point with curvature maxima and minima. It is a method for general boundary curve representation and matching. Webster et al. (1991) proposed isthmus critical points for solving jigsaw puzzles. Their solution deals with the canonical (off-the-shelf) jigsaw puzzles. Isthmus and isthmus critical points are very efficient features for solving jigsaw puzzles. Kosiba et al. (1994) derived a new set of features based on the shape and color characteristics of off-the-shelf puzzles. The work of Yao et al. (1997) is a preparatory work which tries to solve the jigsaw puzzle with a computer. It pointed out the problems existing in the work done by Freeman and Garder, and proposes using the fine segmentation of boundary curve of off-the-shelf puzzle pieces. All works can be categorized by the types of information used in solving the problem and by whether the targets were canonical jigsaw puzzles. The summaries are shown in Table 1. The work of Freeman and Garder and the work of Radack and Badler did not deal with canonical jigsaw puzzles. However, all methods employed the piece boundary shape information. Kosiba et al.'s and Chung et al.'s (1998) methods both employed color information. Like Kosiba's method, Chung et al. also proposed solving this problem using shape and color information in which the distance from points on boundary curves to the line determined by two neighboring corner points are used for shape matching. There are also new attempts to solve this problem by using neural networks (e.g., Suganthan, 1999), but the results are less than satisfactory. There are two strategies when solving

Table 1
Categorization of methods to solve jigsaw puzzle problem

Method proposed by	Deal with the canonical jigsaw puzzles?	Use piece boundary shape information?	Use piece texture or color information?
Freeman and Garder (1964)	No	Yes	No
Radack and Badler (1982)	No	Yes	No
Webster et al. (1991)	Yes	Yes	No
Kosiba et al. (1994)	Yes	Yes	Yes
Yao et al. (1997)	Yes	Yes	No
Chung et al. (1998)	Yes	Yes	Yes

this problem by a computer: *top-down* or *bottom-up*. The top-down methodology is very similar to the way a jigsaw puzzle is fabricated. Usually, to generate a jigsaw puzzle, a large piece of picture is cut into numerous small irregular pieces by using cutting-rules. For a given piece in S_p , the shape of its neighbor piece can be determined by using the *cutting-rules*. Glassner (2002) employed this approach. He took a photograph of the entire picture of the jigsaw puzzle, and then drew jigsaw-shaped pieces in Adobe Photoshop. Then, the pieces were separated and scattered by a program with the orientation of each piece being fixed. An attempt was then made to reassemble the pieces. This is not a real-world canonical jigsaw puzzle, so it has not been put into Table 1. The bottom-up methodology intends to find the neighboring piece for a given piece in S_p by using shape and texture information. Works in Table 1 use this approach. Generally, the cutting-rules are not given to the player, and because the bottom-up methodology is similar to the process that people use in solving a puzzle by hand, we employed the bottom-up methodology to solve this problem. Here, we propose a method to solve the jigsaw puzzle problem by using both piece boundary shape information and piece boundary image information. Kosiba et al.'s and Chung et al.'s methods also employed these two types of information. The difference between their method and ours is as follows. In both methods, color information is employed, besides of

the piece boundary shape information. This distinguishes their method from others. The color characteristics are the mean and the variance of the hue, saturation, and intensity values for each piece or for the sampling windows situated at regular intervals along the contour of the piece. Kosiba et al. defined a feature set for shape matching, and Chung et al. used the distance from points on boundary curves to the line determined by two neighboring corner points for shape matching. There are three aspects to our method as described in Section 3: (i) we directly use the partial boundary curve (or jigsaw piece edge) matching to decide the candidates of the matched pieces. This method omits the problem of determining and calculating the feature set from the boundary curve; (ii) we use image merging for piece boundary images; (iii) we introduce the canonical jigsaw piece model in the proposed method. By using the jigsaw piece model, jigsaw pieces are classified into 18 categories according to the shape information. This reduces the computation time. Our algorithm to solve the jigsaw puzzle can be formulated in six steps:

- (1) Extract the jigsaw pieces from the color input image and express the jigsaw pieces according to their boundary curves.
- (2) Detect the dominant points of each piece, and then seek the corner points from them.
- (3) Separate the boundary curves into four edges by using the corner points as the separation points (note that the corner points are quartet points), and then perform the piece classification and recognition.
- (4) Perform the boundary shape matching to establish the candidates for the neighbor pieces.
- (5) Perform image merging between the present piece and all candidates.
- (6) Based on the results of steps 4 and 5, the connection relationships among the jigsaw pieces are recovered.

The of the rest of this paper is organized as follows: Section 2 describes the jigsaw piece model construction. Section 3 shows the whole algorithm to solve the jigsaw puzzles. Section 4 represents the experiment results of employing real-world im-

ages. The paper ends with conclusions and suggestions for future research.

2. Model construction of jigsaw pieces

For a piece of an N -piece canonical jigsaw puzzle, considering all possible matching with other pieces, the number reaches $4 \times 4 \times (N - 1)$ (each piece generally has four edges). So people can imagine how big the total number of matches among all jigsaw pieces. There exists a “combination explosion” phenomenon in the jigsaw puzzle problem. To alleviate the “combination explosion”, we introduced jigsaw piece models for canonical jigsaw puzzles. There exist four corner points for the canonical jigsaw puzzle piece, and its boundary curve can be separated into four edges at four corner points. Each of the four edges fits into one of the following three patterns: (1) a straight lined edge denoted by “ L ”, (2) a concaved and curved edge denoted by “ C ”, (3) a convexed and curved edge denoted by “ V ”. These three types of edges are typical of any canonical jigsaw piece and are used to classify the jigsaw pieces.

If we skeletonize the four edges of a jigsaw piece, all jigsaw pieces can be classified into three categories: corner piece, edge piece and interior piece.

The corner piece, denoted by R , is located at the corner of the whole jigsaw puzzle. There are always only four corner pieces in a canonical jigsaw puzzle no matter how big the number of the total pieces. Corner pieces are further classified into four types as shown in the second column, from row 2 to 5, of Table 2, which are named R_0 , R_1 , R_2 and R_3 , respectively. The edge configurations of R_i ($i = 0, 1, 2, 3$) are “ $CCLL$ ”, “ $VCLL$ ”, “ $CVLL$ ”, and “ VLL ”, correspondingly, as given in the third column, from row 2 to 5, of Table 2. Notice here that “ $CLLC$ ” is the same as “ $CCLL$ ” because “ $CCLL$ ” can be obtained by rotating “ $CLLC$ ” or by changing the starting corner point. This is true of other pieces. In Table 2, the starting corner point for all jigsaw piece patterns in the second row is set at the left-bottom corner, and the edges are traced clockwise.

The edge pieces, denoted by E , are found on the four sides of the whole jigsaw puzzle. The number

Table 2

Jigsaw piece boundary description and each edge's neighboring piece type and edge type

	Piece type	Boundary description	Edge 0	Edge 1	Edge 2	Edge 3
R	 R_0	<i>CCLL</i>	E.V	E.V	None	None
	 R_1	<i>VCLL</i>	E.C	E.V	None	None
	 R_2	<i>CVLL</i>	E.V	E.C	None	None
	 R_3	<i>WLL</i>	E.C	E.C	None	None
E	 E_0	<i>CCCL</i>	E.V, R.V	<i>I.V</i>	E.V, R.V	None
	 E_1	<i>CVCL</i>	E.V, R.V	<i>I.C</i>	E.V, R.V	None
	 E_2	<i>VCCCL</i>	E.C, R.C	<i>I.V</i>	E.V, R.V	None
	 E_3	<i>CCVL</i>	E.V, R.V	<i>I.V</i>	E.C, R.C	None
	 E_4	<i>WVCL</i>	E.C, R.C	<i>I.C</i>	E.V, R.V	None
	 E_5	<i>CVL</i>	E.V, R.V	<i>I.C</i>	E.C, R.C	None
	 E_6	<i>VCVL</i>	E.C, R.C	<i>I.V</i>	E.C, R.C	None
	 E_7	<i>WVL</i>	E.C, R.C	<i>I.C</i>	E.C, R.C	None
I	 I_0	<i>CCCC</i>	E.V, I.V	E.V, I.V	E.V, I.V	E.V, I.V
	 I_1	<i>VCCC</i>	E.C, I.C	E.V, I.V	E.V, I.V	E.V, I.V
	 I_2	<i>VCVC</i>	E.C, I.C	E.V, I.V	E.C, I.C	E.V, I.V
	 I_3	<i>CVWC</i>	E.V, I.V	E.C, I.C	E.C, I.C	E.V, I.V
	 I_4	<i>WVC</i>	E.C, I.C	E.C, I.C	E.C, I.C	E.V, I.V
	 I_5	<i>WWW</i>	E.C, I.C	E.C, I.C	E.C, I.C	E.C, I.C

of edge pieces varies with the size of the jigsaw puzzle. Taking into account all possible combinations of edge patterns, eight types of edge pieces are found, and are shown in rows 6 through 13 in the second column of Table 2, and are named E_0 through E_7 , and the edge configurations are “*CCCL*”, “*CVCL*”, ..., “*WVL*”, sequentially, as shown in the third column, from row 6 to 13.

The interior pieces, denoted by **I**, are found on the interior area of the jigsaw puzzle. The number of interior pieces also changes with the size of the jigsaw puzzle, and there are far more interior pieces than edge pieces. There are six types of interior pieces given all possible edge combination and they are listed in the second column, from row 14 to 19, of Table 2, and are named I_0, I_1, \dots, I_5 , and the edge configurations are expressed as “*CCCC*”, “*VCCC*”, ..., “*WWW*”, correspondingly, as given in the third column.

Thus, we can conclude that the canonical jigsaw pieces can be classified into 18 categories which are denoted by set S_M , i.e., $S_M = \{R_0, \dots, R_3\}$,

$\{E_0, \dots, E_7\}$, $\{I_0, \dots, I_5\}$. Their pattern images are shown in the second column of Table 2.

Let us employ the notation “**PT.ET**” to express the edge connection relationships among jigsaw pieces, where **PT** means the neighboring piece type (i.e., **R**, **E** or **I**), and **ET** the neighbor edge type (i.e., *L*, *C* or *V*). Then the connection relationships among these 18 kinds of jigsaw pieces are summarized in column 4 to 7 of Table 2. The 4th, 5th, 6th, and 7th columns shows the required neighboring piece type and neighbor edge type of the four edges respectively. For the corner pieces there exist two kinds of connection relationships—**E.V** and **E.C**, as shown in the 4th and 5th columns of Table 2. **E.V** means that the required neighboring piece is an edge piece, and the required neighbor piece has a “convex” edge. Similarly, for the edge pieces, there exist six kinds of connection relationships—**E.V**, **E.C**, **R.V**, **R.C**, **I.V**, and **I.C**. And for the interior pieces, there exist four kinds of connection relationships—**E.V**, **E.C**, **I.V**, and **I.C**. “None” in the 6th and 7th column means no

neighboring piece exists. This table's contents plays an important role in the proposed algorithm to solve the jigsaw puzzle problem.

3. Overview of the proposed algorithm

Our algorithm can be formulated in six steps. Once the process of image capture begins, there is no other human intervention while the puzzle is being solved. Steps 1–3 relate to the image processing techniques in the proposed algorithm. A brief summary is given in the following; details refer to the literature (Yao et al., 2002). The definitions, symbols, and notations, used in a previous work (Yao et al., 2002), are summarized in Table 3. They will be also employed here to describe the whole algorithm.

Step 1. Extract the jigsaw pieces from the color input image and express the jigsaw pieces according to their boundary curves.

Table 3
Nomenclature

Symbols	Meaning
P_i	i th jigsaw piece
S_p	Set of jigsaw pieces extracted from the input image
N	Number of jigsaw pieces in S_p
B_i	Jigsaw piece boundary curve of P_i
S_B	Set of piece boundary curves
K_i	Curvature function of B_i
D_i^k	k th dominant of B_i
S_{D_i}	Set of dominant points of B_i
N_i	Number of dominant points in S_{D_i}
θ_i^k	Interior angle at D_i^k of P_i
S_{θ_i}	Set of interior angles at D_i^k of P_i
C_i^m	m th corner point of P_i
S_{C_i}	Set of corner points of P_i
α_i^m	Interior angle at m th corner point of P_i
S_{α_i}	Set of interior angles at corner points of P_i
E_i^{mn}	Clockwise edge from start-point m to end-point n of P_i
\widehat{E}_i^{mn}	Counterclockwise edge from start-point n to end-point m of P_i
S_{E_i}	Set of clockwise edges of P_i
\widehat{S}_{E_i}	Set of counterclockwise edges of P_i
L_i^{mn}	Length of edge E_i^{mn}
γ_i^{kl}	Central angle by the dominant point D_i^k , the centroid O_i , and the dominant point D_i^l of P_i
PME	Possibly matched edge
TME	Truly matched edge

Jigsaw piece extraction is a problem belonging to the field of *object extraction*. We developed a new method to extract the jigsaw pieces. In this method, multiple color images with different background color are employed to make a mask image. Then, the evaluation image is created from one of the input images. The last step is to perform a *conditional contraction* to the mask image. This method requires that the jigsaw pieces be in the same position in the multiple input images with different background color, and that the jigsaw pieces do not overlap each other but remain far enough apart to be extracted independently.

Step 2. Detect the dominant points of each piece, and then seek the corner points from them.

The dominant points are detected based on multi-scale curvature. The corner points are the special dominant points, which have the following properties:

- (1) their interior angles are almost the same;
- (2) the central angles between any two succeeding corner points are nearly the same;
- (3) their distances to the centroid of the jigsaw piece P_i are approximately the same;
- (4) within the regions supporting them, the partial boundary curves are all convex from the centroid;
- (5) the line determined by the leftmost and rightmost points of the region of support of the dominant point, D_i^s , is nearly perpendicular to the line determined by D_i^s and D_C . Note that D_C is the middle point of the chord between the leftmost and rightmost points of the region of support of the dominant point, D_i^s .

The corner points are calculated from the dominant points based on above properties.

Step 3. Separate the boundary curves into four edges by using the corner points as the separation points (note that the corner points are quartet points), and then perform the piece classification and recognition.

The jigsaw piece classification determines which category the piece belongs to, i.e., **R**, **E** or **I**, and the jigsaw piece recognition determines which model in Table 2 it corresponds to. The jigsaw piece recognition consists of edge recognition and piece

type recognition. Edge recognition determines which edge pattern the current edge corresponds to, i.e., L , C or V . For the edge E_i^{mn} , if the distance from the point D_M to the line l is smaller than the predetermined threshold value, it is considered to be L (straight line). Otherwise, if D_M is on the left side when tracing the edge from the start-point to the end-point clockwise, it is considered to be V (convex edge), if on the right, it is C (concave edge). Note that l is the line determined by the start-point and end-point of the edge in question, and D_M is the most deviant boundary point from the line l . After the edge recognition, we obtain a four-letter string for every piece, which comprises “ L ”, “ C ”, and “ V ”. The object of the jigsaw piece recognition is to perform the matching between the four-letter string of the jigsaw piece in question and those of models in \mathbf{S}_M . If “ $S_1S_2S_3S_4$ ” is the four-letter string of the jigsaw piece in question, the details of the matching are as follows. We take the four-letter string of the first model in \mathbf{S}_M , and perform the matching with “ $S_1S_2S_3S_4$ ”, “ $S_2S_3S_4S_1$ ”, “ $S_3S_4S_1S_2$ ”, and “ $S_4S_1S_2S_3$ ”. If the matched string is found, the piece in question is thought of as the same type as that one in \mathbf{S}_M . Otherwise, move to the next model in \mathbf{S}_M and repeat the above matching. This operation is repeated until the matched model is found.

Step 4. Perform the boundary shape matching to identify candidates for neighboring pieces.

The details will be described in Section 3.1.

Step 5. Perform the image merging between the current piece and all candidates.

The details will be related in Section 3.2.

Step 6. Based on the results of step 4 and 5, the connection relationships among the jigsaw pieces are recovered.

Section 3.3 gives details.

3.1. Jigsaw piece boundary shape matching

The proposed algorithm combines boundary shape matching and image matching (image merging) with a cyclic “growth” process of trying to place pieces in their correct positions. This section relates to the partial boundary shape matching, the image merging will be explained in the next section, and the cyclic “growth” process

of trying to place pieces in their correct positions will be handled in Section 3.3.

A computer performs the partial boundary curve matching process by using a clockwise jigsaw piece edge and counterclockwise jigsaw piece edge. For $P_i, P_j \in S_P$ ($i \neq j, i, j = 0, 1, \dots, N-1$), the boundary shape matching between these two pieces is shown below (note that the boundary curve is given by discrete x - and y -coordinates in a Cartesian coordinate system). We take the clockwise edge $E_i^{k,k+1}$ ($k = 0, 1, 2, 3$, modulo 4) of P_i and the counterclockwise edge $\widehat{E}_j^{m,m+1}$ ($m = 0, 1, 2, 3$, modulo 4) of P_j ; if their edge types satisfy any connection relationships defined in jigsaw piece model table, the matching between them is performed. Otherwise, we try other edges of P_i that can be mated. The matching between $E_i^{k,k+1}$ and $\widehat{E}_j^{m,m+1}$ is described as follows. $\widehat{E}_j^{m,m+1}$ is translated along the X - and Y -axis so that its start-point overlaps with the start-point (x_i^k, y_i^k) of $E_i^{k,k+1}$. The displacements of X - and Y -axis are given by

$$D_x(E_i^{k,k+1}, \widehat{E}_j^{m,m+1}) = x_i^k - x_j^m \quad (1)$$

$$D_y(E_i^{k,k+1}, \widehat{E}_j^{m,m+1}) = y_i^k - y_j^m, \quad (2)$$

where (x_i^k, y_i^k) are the coordinates of the start-point of $E_i^{k,k+1}$, and (x_j^m, y_j^m) are the coordinates of the start-point of $\widehat{E}_j^{m,m+1}$. The edge $\widehat{E}_j^{m,m+1}$ is rotated counterclockwise so that the line determined by its start-point and end-point overlaps with that determined by the start-point and end-point of the clockwise edge $E_i^{k,k+1}$. The rotation angle is given by

$$\begin{aligned} \varphi(E_i^{k,k+1}, \widehat{E}_j^{m,m+1}) &= \tan^{-1} \frac{y_j^{m+1} - y_j^m}{x_j^{m+1} - x_j^m} \\ &\quad - \tan^{-1} \frac{y_i^{k+1} - y_i^k}{x_i^{k+1} - x_i^k} \end{aligned} \quad (3)$$

where (x_i^{k+1}, y_i^{k+1}) are the coordinates of the end-point of $E_i^{k,k+1}$, and (x_j^{m+1}, y_j^{m+1}) are the coordinates of the end-point of $\widehat{E}_j^{m,m+1}$. The matching error is defined as

$$\varepsilon(E_i^{k,k+1}, \widehat{E}_j^{m,m+1}) = \iint_{\Omega} dx dy \quad (4)$$

where Ω is the region circled by the clockwise edge $E_i^{k,k+1}$ and the counterclockwise edge $\widehat{E}_j^{m,m+1}$.

For the digital jigsaw boundary curve, Eq. (4) can be rewritten as the sum of area surrounded by the boundary point pair on $E_i^{k,k+1}$ and that on $\widehat{E}_j^{m,m+1}$, plus the difference between the lengths of these two edges, multiplied by a weighting coefficient. That is,

$$\varepsilon(E_i^{k,k+1}, \widehat{E}_j^{m,m+1}) = \sum_{p=0, q=0}^{\lambda} (S_{\Delta(p,p+1,q)} + S_{\Delta(q,q+1,p+1)}) + W \times |L_i^{k,k+1} - L_j^{m,m+1}| \quad (5)$$

where λ is the smaller of the two edge lengths, that is, $\lambda = \min\{L_i^{k,k+1}, L_j^{m,m+1}\}$, W is a weighting coefficient (at present it is set at 100), $S_{\Delta(p,p+1,q)}$ and $S_{\Delta(q,q+1,p+1)}$ are the areas of the triangles formed by the boundary points $p, p+1$, and $q; q, q+1$, and $p+1$; respectively, p is the boundary point on edge $E_i^{k,k+1}$ starting from D_i^k , and q is the boundary point on edge $\widehat{E}_j^{m,m+1}$ corresponding to p . The areas of the triangles are computed as follows (refer to Fig. 1),

$$S_{\Delta(p,p+1,q)} = (y_j^p - y_j^{p+1})x_i^q - (x_j^p - x_j^{p+1})y_i^q + (x_j^p - x_j^{p+1})y_j^p - (y_j^p - y_j^{p+1})x_j^p \quad (6)$$

$$S_{\Delta(q,q+1,p)} = (y_i^q - y_i^{q+1})x_j^{p+1} - (x_i^q - x_i^{q+1})y_j^{p+1} + (x_i^q - x_i^{q+1})y_i^q - (y_i^q - y_i^{q+1})x_i^q \quad (7)$$

$E_i^{k,k+1} \in S_{E_i}$ of P_i ($k = 0, 1, 2, 3$, modulo 4) is matched with $\forall \widehat{E}_j^{m,m+1} \in S_{\widehat{E}_j}$ of $P_j \in S_P$ ($m = 0, 1, 2, 3$, modulo 4), and the matching error is calculated according to Eq. (5). The edge corresponding to the smallest matching error is considered to

be the possibly matched edge (PME) between $E_i^{k,k+1}$ of P_i and those of P_j . This is performed for $\forall P_j \in S_P$ ($j \neq i, j = 0, 1, \dots, N-1$), then all PME between $E_i^{k,k+1}$ of P_i and those of $P_j \in S_P$ ($j \neq i$) are obtained. All PMEs of $E_i^{k,k+1}$ of P_i are sorted according to the matching error, from small to big. The top M of the sorted PMEs are thought of as the truly matched edge (TME) candidates.

The above process is applied to $\forall P_i \in S_P$ ($i = 0, 1, \dots, N-1$), and the top M TME candidates of the four edges of P_i are obtained. One of the TME candidates for each edge will be determined as the TME by employing the image merging method which will be described in the next section. At present, M is set at 6. Notice that “ L ” type edge does not have a TME.

3.2. Image merging

The purpose of image merging is to determine the TME from TME candidates. This is based on the integration of image features.

3.2.1. Integration degree of image features

The integration degree of image features is defined from the separability of image features. If we suppose a region Φ of an image can be classified into sub-region Φ_1 and Φ_2 , the separability of image features is defined as (Otsu, 1980),

$$\eta = \frac{\sigma_b^2}{\sigma_T^2} \quad (8)$$

$$\sigma_b^2 = n_1(\bar{P}_1 - \bar{P}_m)^2 + n_2(\bar{P}_2 - \bar{P}_m)^2 \quad (9)$$

$$\sigma_T^2 = \sum_{i=1}^n (P_i - \bar{P}_m)^2 \quad (10)$$

where n_1 and n_2 are the number of image pixels in Φ_1 and Φ_2 , respectively, $n = n_1 + n_2$, \bar{P}_1, \bar{P}_2 , and \bar{P}_m are the average of the luminance in Φ_1, Φ_2 , and Φ , correspondingly, P_i is the luminance at i th pixel, and σ_T is the variance in Φ . η lies in the range $0 < \eta \leq 1.0$. If Φ_1 and Φ_2 can be separated completely, $\eta = 1.0$. If Φ_1 and Φ_2 cannot be separated, then η is near to 0. P_i can be the saturation, hue,

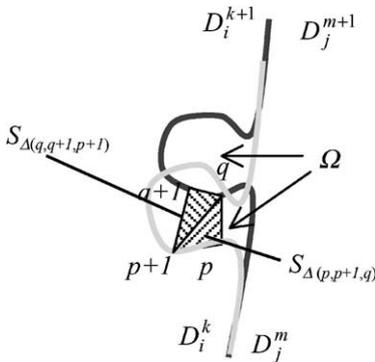


Fig. 1. Edge matching.

texture, and so on, instead of the luminance, at i th pixel.

Generally, for the ξ standardized image features, the separability is defined below (Fukui, 1997)

$$\eta = \frac{\sum_{f=1}^{\xi} \sigma_b^2}{\sum_{f=1}^{\xi} \sigma_{Tf}^2} \quad (11)$$

$$\sigma_f^2 = n_1(\overline{P_{1f}} - \overline{P_{mf}})^2 + n_2(\overline{P_{2f}} - \overline{P_{mf}})^2 \quad (12)$$

$$\sigma_{Tf}^2 = \sum_{i=1}^n (P_{if} - \overline{P_{mf}})^2 \quad (13)$$

where $\overline{P_{1f}}$, $\overline{P_{2f}}$, and $\overline{P_{mf}}$ are the average of the feature f in Φ_1 , Φ_2 , and Φ , respectively, and σ_{Tf} is the variance of image feature f ($f = 1, \dots, \xi$) in Φ . The value of η in Eq. (11) also lies in the range $0 < \eta \leq 1.0$.

From Eq. (11), the integration degree of image features is defined as

$$\rho = 1 - \eta. \quad (14)$$

If R_1 and R_2 can be merged completely, ρ takes a value near to 1.0. If Φ_1 and Φ_2 cannot be merged, ρ is equal to 0.

ρ is used to determine the TME from the TME candidates. Details are given in the next section.

3.2.2. Image merging

Let us employ Fig. 2 to explain the image merging between jigsaw pieces. According to the boundary shape matching, as described in Section 3.1, one TME candidate of E_0^{12} of P_0 is E_1^{23} of P_1 as shown in Fig. 2(a). However, we cannot say definitely say that E_0^{12} of P_0 can be connected to E_1^{23} of P_1 because the shape information is not sufficient. We must also check the images in the regions next to the edge E_0^{12} and E_1^{23} , that is, the regions R_0^{12} and R_1^{23} may or may not be merged according to the integration of image features defined in Eq. (14). Details are given below.

For the convenience of calculating the integration of image features between the image region near the edge E_0^{12} and near E_1^{23} , R_0^{12} and R_1^{23} are divided into rectangular regions \tilde{R}_0^{12} and \tilde{R}_1^{23} as shown in Fig. 2(b) respectively. \tilde{R}_0^{12} and \tilde{R}_1^{23} are

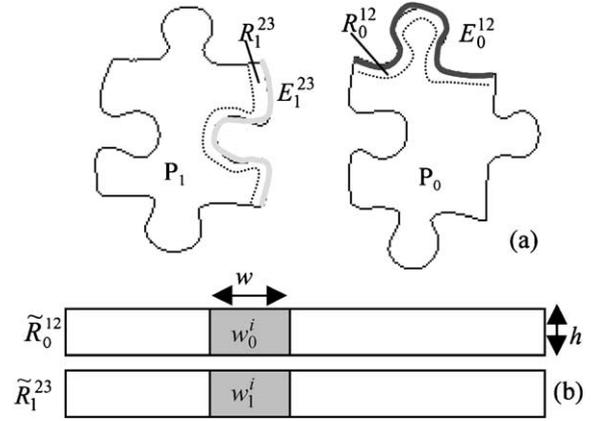


Fig. 2. Integration degree of image.

separated into small regions having the same width and height. The integration degree between \tilde{R}_0^{12} and \tilde{R}_1^{23} is defined as

$$\bar{\rho} = \frac{1}{\zeta} \sum_{i=0}^{\zeta-1} \rho_i \quad (15)$$

$$\zeta = \min \left\{ \left\lfloor \frac{L_0^{12}}{w} \right\rfloor, \left\lfloor \frac{L_1^{23}}{w} \right\rfloor \right\} \quad (16)$$

where ρ_i is the integration degree between i th small region w_0^i of \tilde{R}_0^{12} and w_1^i of \tilde{R}_1^{23} , which is given in Eq. (14); and ζ is the minimal number of small regions in \tilde{R}_0^{12} and \tilde{R}_1^{23} ; w is the width of the small region. The image features used in Eq. (14) are R -, G -, and B -components. Certainly, other features such as YIQ, Yuv, and so on, can also be employed.

Notice that when spreading the image data in R_0^{12} to the rectangular region \tilde{R}_0^{12} , the image data is copied along the normal line and the start-point is not from a point on the edge but from a point d_1 dots from the edge. This is shown in Fig. 3. Points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ are points d_2 dots from the point in question to the left and right along the edge, and $p_c(x_c, y_c)$ is the mediate point of p_1 and p_2 . The coordinates of the point $p_3(x_3, y_3)$ on the normal line l_2 are given by

$$x_3 = x_c + \frac{d}{\|P_2 - P_1\|} (y_2 - y_1) \quad (d_1 \leq d \leq d_1 + H) \quad (17)$$

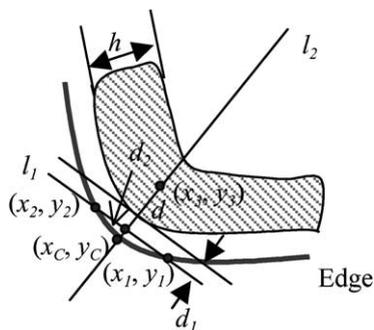


Fig. 3. Image region used to calculate the integration degree.

$$y_3 = y_c - \frac{d}{\|P_2 - P_1\|} (x_2 - x_1) \quad (d_1 \leq d \leq d_1 + H) \quad (18)$$

where d is the distance from p_c along the normal line l_2 . At present, w , d_1 , and d_2 are set at 10 dots, h

20 dots. Note that when performing image merging, because the boundary curve includes boundary noise and the boundary curve may include shades formed by the image input device, the sampling of the pixels starts a small distance, i.e., d_1 , into the puzzle piece.

The edges in Fig. 4(b)–(g) marked by two red lines are the TME candidates of the edge in Fig. 4(a) also marked by two red lines. These are obtained by piece boundary shape matching. The images near these edges in (a)–(g) are spread into rectangular regions as shown in (h)–(n), respectively. The integration degrees between the rectangle region in (h) and those in (i)–(n) are shown in (o)–(t), correspondingly.

3.3. Recovery of connection relationships

TME candidates obtained in the previous section are put into the jigsaw piece connectedness

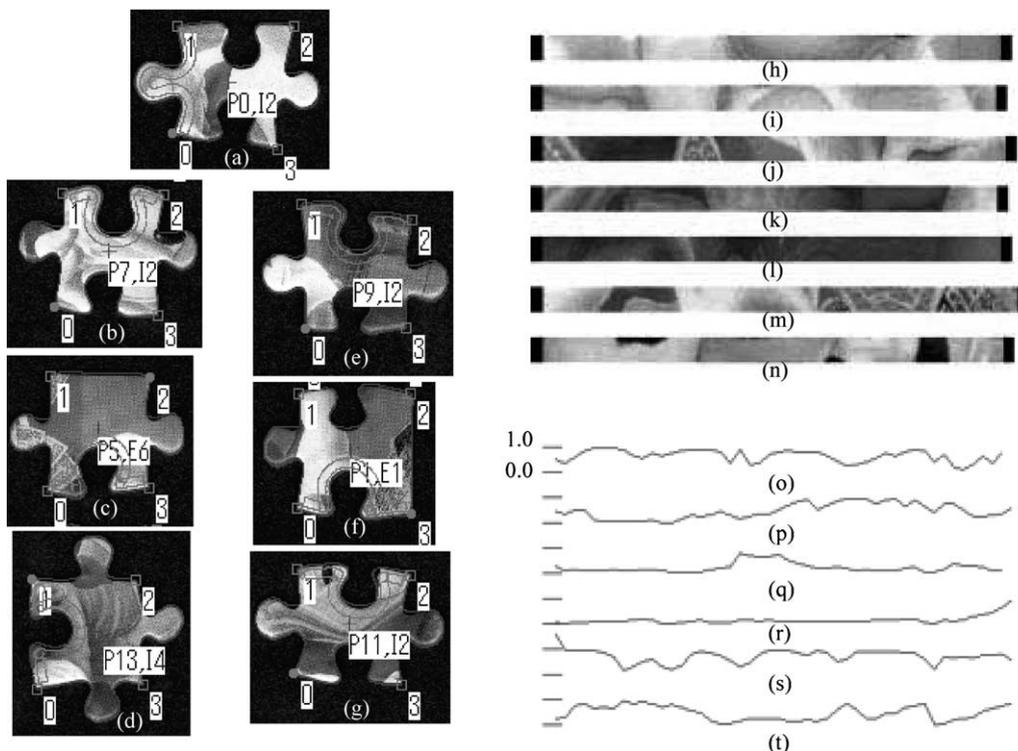


Fig. 4. Edges in (b)–(g) marked by double red curves are six TME candidates for the edge also marked by double red curves in (a). Image regions near the edges in (a)–(g) are converted to rectangular regions in (h)–(n). Degrees of image integration between (h) and (i)–(n) are shown in (o)–(t).

Table 4
Format of jigsaw piece connectedness table

	Cad	X_1	Y_1	X_2	Y_2	ω_k	ρ_k	τ_k
TME candi-	0							
dates of 1st	1							
edge of P_0	:							
	$M - 1$							
TME	0							
candidates	1							
of 2nd edge	:							
of P_0	:							
	$M - 1$							
:	0							
	1							
	:							
	$M - 1$							
	:							

table. The connection relationships are recovered according to this table. The format of the jigsaw piece interconnected table is given in Table 4. The first column, *Cad*, shows the candidate number which is in repetition with length M , that is, the first M shows the candidate number of the first edge of P_0 , the second of the second edge of P_0 , and so on. The second and the third columns, X_1 and Y_1 , are the piece and edge numbers of the piece in question respectively; and the fourth and fifth columns, X_2 and Y_2 , are the piece and edge numbers of the neighbor of the piece in question. The sixth column, ω_k , is the jigsaw piece boundary shape matching error, which is ranked from small to big. The seventh column, ρ_k , is the integration degree of the images between the regions near the edge X_1 of the piece Y_1 and X_2 of Y_2 . The eighth column, τ_k , is the ratio between the boundary shape matching error of k th candidate and that of the first, that is, $\tau_k = \omega_k/\omega_0$, where $k = 0, 1, \dots, M - 1$.

The recovery of the connection relationships is based on the following *decision rules*.

Rule 1: If the k th candidate ranks first, both in the boundary shape matching error field and in the integration degree field, the k th candidate is con-

sidered as the TME. Notice that first rank in the matching error field means its matching error is the smallest among the candidates, and the first rank in the integration degree field means that its integration degree is the largest.

Rule 2: If the m th candidate ranks first in the boundary shape matching error field, n th candidate ranks first in the integration degree field ($m \neq n$), and if τ_n is bigger than the threshold τ_{thres} , the m th candidate is thought of as the TME. At present τ_{thres} is determined experimentally and set at 1.5.

Rule 3: If the s th candidate ranks first in the boundary shape matching error field, the t th candidate ranks first in the integration degree field ($s \neq t$), and if τ_t is smaller than the threshold τ_{thres} , then the t th candidate is thought of as the TME.

Rule 4: In other cases TME candidates are left unsolved.

Moreover, because the jigsaw pieces are interconnected, the contents of jigsaw piece connectedness table can be corrected according to the following *correction rule*.

Correction rule: The jigsaw piece connectedness table is searched from top to bottom. If P_m 's k th edge connects to the s th edge of P_n ($n > m$) according to the decision rule 1, 2, or 3, then it can be said that the s th edge of P_n is connected to the k th edge of P_m . Therefore, for the first TME candidate of the s th edge of P_n , the piece and the edge numbers of the neighboring piece, that is, X_2 and Y_2 , can be determined as P_m and k , respectively. Its integration degree, that is, ρ_0 can be set to 1.0, and the matching error, i.e., ω_0 , can be set to a positive non-zero value as small as possible, e.g., 1.0. Next, the jigsaw piece connectedness table is searched from bottom to top, and above process is repeated. In this way, the TME can be rescued from the unsolved TME candidates.

After TMEs are determined, the connection relationships among the jigsaw pieces can be recovered as follows:

- (1) Search the jigsaw piece connectedness table and find the corner piece. Call the corner piece be P_i , and put it into the *buffer*, and set the index c of the buffer to 0, and the number of jigsaw pieces in the buffer n to 1. Then make the

output image I_{out} , and initialize I_{out} with 0. Copy the image of P_i to I_{out} .

- (2) Find the neighboring piece for the k th edge of P_i in the *buffer*, from the jigsaw piece connectedness table. If its neighboring piece, for example P_j , is found, then P_j is translated and rotated according to the displacements and rotation angle obtained from Eqs. (1)–(3); the image of P_j is pasted to I_{out} , and n is increased by 1. Where the edge index, k , changes from 0 to 3.
- (3) The operation in (2) is repeated for c from 0 to n .

In this way, the connection relationships among the jigsaw pieces can be recovered.

4. Experiment result

We employed real-world images to test the algorithm related above. The algorithms were implemented on a Windows platform, and the programming language was C++. An EPSON GT-9500 scanner was used as the image input device.

Fig. 5 shows one of the five input images, and is the input image with a white background. The resolution of the scanner is set at 300 dpi, and the size of each image is $1704 \times 1644 \times 3$ bytes. This image includes 16 canonical jigsaw pieces, one corner piece (R_2), six edge pieces (three E_1 and three E_6), and nine interior pieces (one I_0 , five I_2 , two I_3 and one I_4). Fig. 6 shows the piece recognition results, where the extracted jigsaw pieces are shown by the boundary curve, and are numbered from P_0 to P_{15} in the order of extraction, and the centroids by “+”. The detected corner points are shown by “□” or “•”, where “•” marks the start corner point in accordance with those of jigsaw piece models in Table 2. The recognized piece type is written right after the piece number and “;” at the centroids.

By applying the jigsaw piece boundary shape matching and image merging as described in Section 3, the connectedness table is obtained. The size of the table is $64 \times M \times 4$ (M is the number of TME candidates). A portion of the connectedness table for the input image in Fig. 5 is given in Table

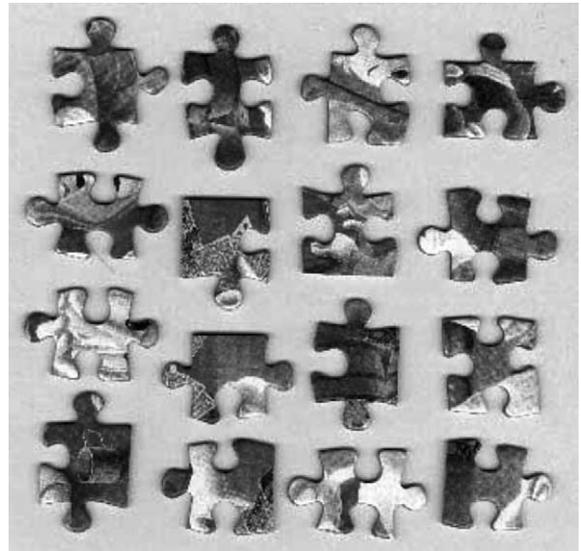


Fig. 5. One of the input images (white background).

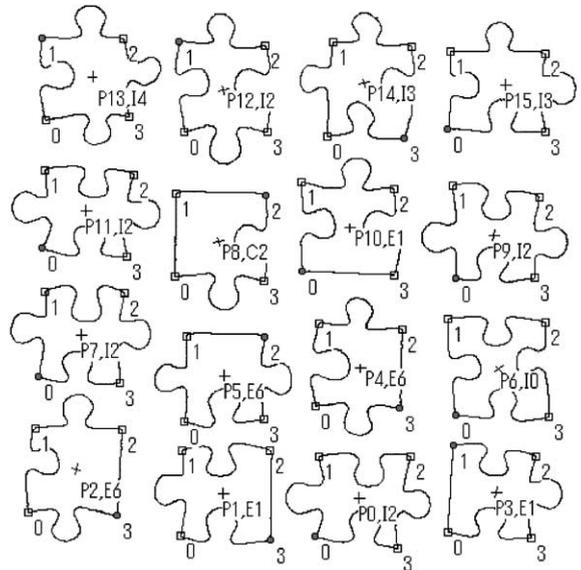


Fig. 6. Extracted jigsaw pieces; corner points, orientation, and piece type of each piece.

5. The ninth column of Table 5 shows the applied decision rule. There are 64 interconnected relationships in this table, among them there are eight

Table 5
Connectedness table for input image in Fig. 6

C	X_1	Y_1	X_2	Y_2	ω_k	ρ_k	τ_k	R
0	0	0	7	1	1989.3	0.66890	1.00	R2
1	0	0	5	3	2026.4	0.47689	1.02	
2	0	0	13	0	2332.6	0.25023	1.17	
3	0	0	9	1	2968.0	0.17238	1.49	
4	0	0	1	3	3443.9	0.72644	1.73	
5	0	0	11	1	3736.3	0.49098	1.88	
⋮								
0	0	3	9	2	1953.1	0.42560	1.00	
1	0	3	13	1	2180.7	0.44246	1.12	
2	0	3	1	0	4305.5	0.40391	2.20	
3	0	3	14	0	4310.3	0.46123	2.21	
4	0	3	7	0	4584.3	0.44262	2.35	
5	0	3	5	0	4999.5	0.39152	2.56	
⋮								
						Correction rule		
0	11	2	1	3	2139.2	0.33707	1.00	
1	11	2	13	0	2367.4	0.53802	1.11	
2	11	2	7	1	2558.9	0.21289	1.20	
3	11	2	0	1	2734.5	0.23092	1.28	
4	11	2	9	1	3137.8	0.47419	1.47	
5	11	2	5	3	3433.9	0.38821	1.61	
⋮								
						Correction rule		
0	13	0	12	1	1257.5	0.43268	1.00	
1	13	0	5	0	2052.7	0.54184	1.63	
2	13	0	4	3	2211.6	0.49541	1.76	
3	13	0	15	2	2304.1	0.54444	1.83	
4	13	0	0	0	2468.4	0.25630	1.96	
5	13	0	11	2	3065.8	0.57924	2.44	
0	13	1	0	3	1617.9	0.43169	1.00	
1	13	1	11	3	4238.5	0.45921	2.62	
⋮								

“L” edges that do not have neighboring pieces, and there are eight pseudoedges because the jigsaw pieces in Fig. 5 make up a portion of the whole puzzle. By applying the decision rules, 46 of the remaining 48 interconnected relationships can be determined, and two are left unsolved. Interconnected relationships determined by rules 1, 2, and 3 are 27, 10, and 9, respectively. By applying the correction rule, the two unsolved interconnected relations are rescued as shown in Table 5 marked by the two arrow-lines. Therefore, the connection relationships in input image Fig. 5 can be recovered completely. The recovery process at steps 0, 3, 6, 9, and 14 is shown in Fig. 7.

Another experiment used the input image shown in Fig. 8, and is one of the five input images. The resolution of the scanner was set at 200 dpi, the size of each image being $1268 \times 1642 \times 3$ bytes. This image includes 25 canonical jigsaw pieces, one corner piece (R_1), eight edge pieces (four E_1 and four E_6), and 16 interior pieces (one I_4 and 15 I_2). Fig. 9 shows the piece recognition results, where the extracted jigsaw pieces are shown by the boundary curve, and are numbered from P_0 to P_{24} in the order of extraction. Then a connectedness table is constructed by applying the boundary shape matching and image merging. The size of the table is $100 \times M \times 4$. There are 100

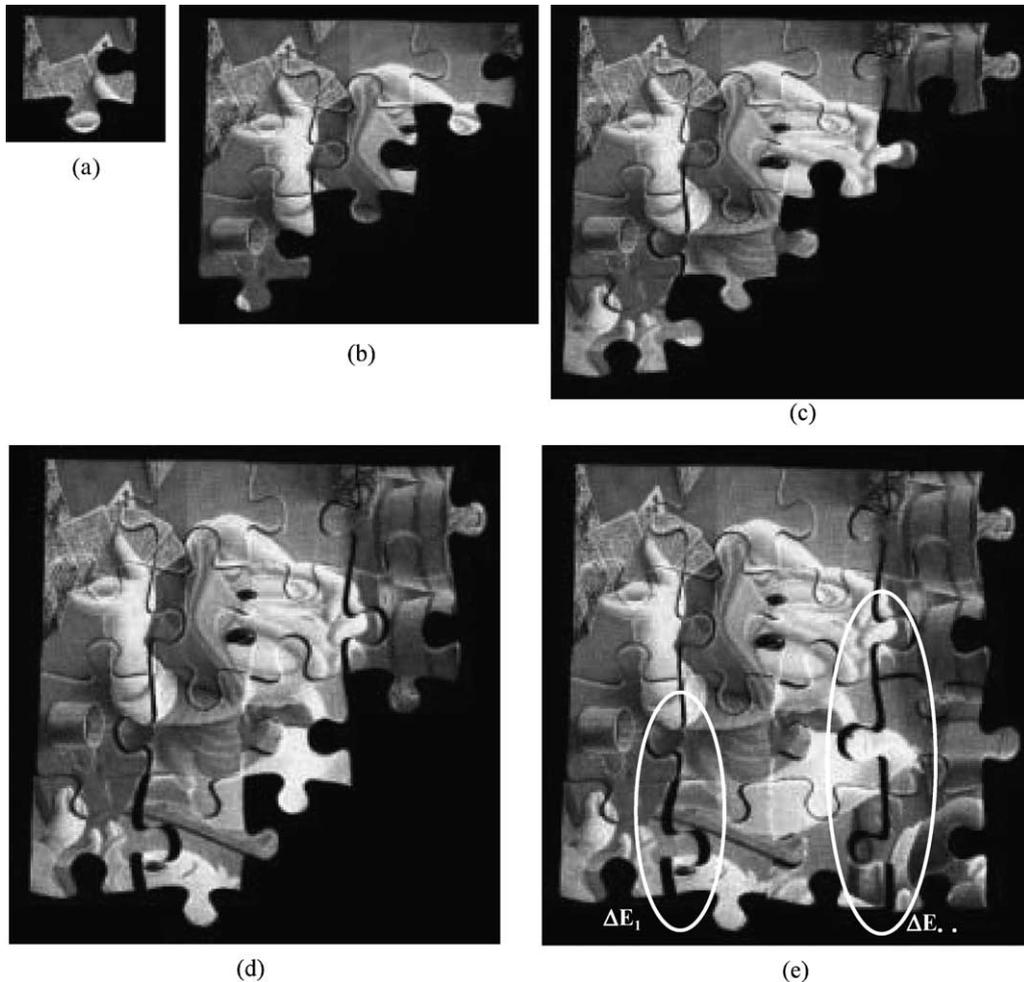


Fig. 7. Recovery process of the connection relationships in the input image in Fig. 5 (©Disney). Repetition times: (a) $c = 0$, (b) $c = 3$, (c) $c = 6$, (d) $c = 9$, (e) $c = 14$.

interconnected relationships in this table (omitted here because it become very large); among them are 10 “L” edges that do not have neighboring pieces, and are 10 pseudoedges because the jigsaw pieces in Fig. 8 are only a part of the entire puzzle. For the remaining 80 interconnected relationships, by applying the decision rules, 75 of them can be determined, and 5 are left unsolved. Interconnected relationships determined by rules 1, 2, and 3 are 41, 12, and 22, respectively. It is worth noting that there is one mistake that arises from rule 1, 4 from rule 2, and 3 from rule 3. However, by

applying the correction rule, these mistakes can be corrected. Two unsolved interconnected relationships are rescued by the correction rule, and another three are left undetermined. Therefore, for the input image in Fig. 8, 96% of the connection relationships can be recovered. The recovery process at step 0, 5, 10, 15, and 23 is shown in Fig. 10.

The experiments are performed on a Windows 2000 machine with a Pentium III CPU and 128 MB memory, working at 600 MHz. The computation time is given in Table 6. The computation

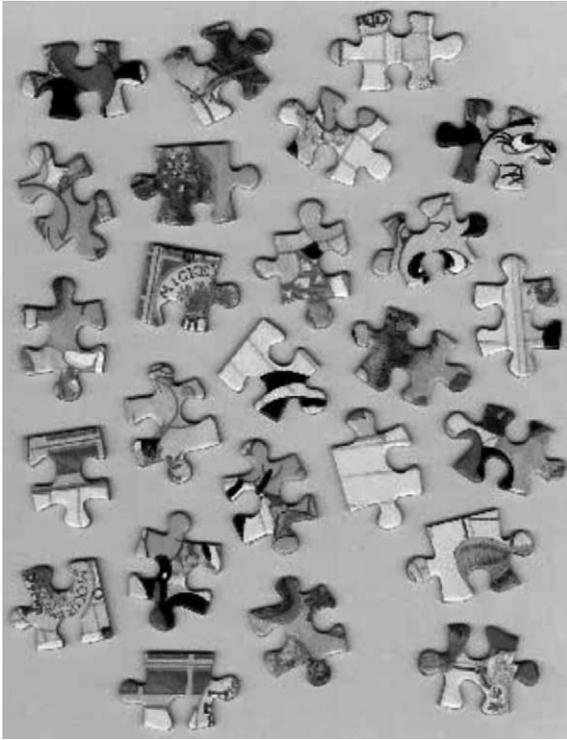


Fig. 8. One of the input images (white background).

time in experiment 1 and 2 cannot compare directly because the resolution of the scanner is set differently. The “combination explosion” is alleviated by employing the jigsaw puzzle models as given in Section 2. We also tested other jigsaw puzzles. If the whole picture of the jigsaw puzzle is a natural scene, the algorithm gets the mistake solution. This is because, (i) many pieces have similar textures and the textures are very delicate; (ii) the image integration degree in Eq. (15) is defined at the large-scale, i.e., along the whole edge; (iii) the decision rules are not sufficient. Moreover, if the resolution of image input device is low, the algorithm fails because this will lead to missing texture on each piece. This can be improved by introducing local image integration degree, employing higher resolution images (over 600 dpi) and utilizing more decision rules. All these will be done in a succeeding work.

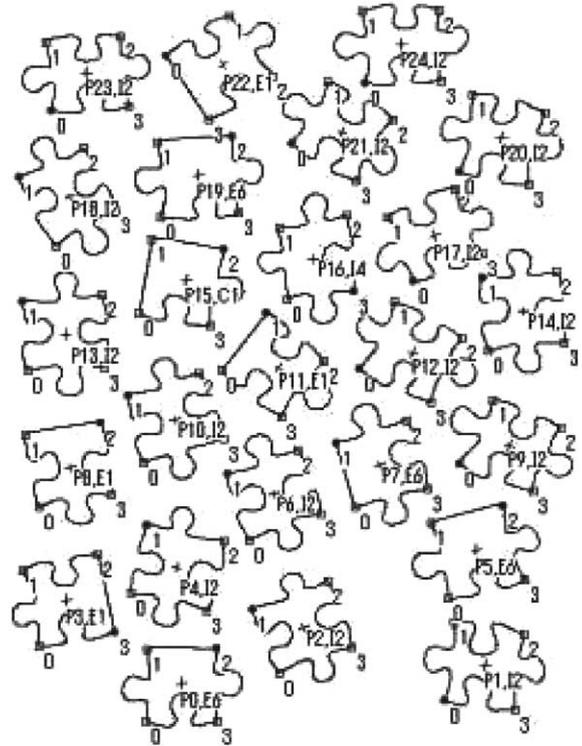


Fig. 9. Extracted jigsaw pieces; corner points, orientation, and piece type of each piece.

5. Conclusions

This paper discusses a computational solution of jigsaw puzzles. The experiments were performed with real-world jigsaw puzzle images. The experiment results show that our method was successful. This method of solving jigsaw puzzles can also be applied in intelligent robot assembly systems, and in map matching.

In this method, we categorized canonical jigsaw pieces into 18 patterns. This greatly reduced the computation time and simplified the process. This method includes jigsaw piece extraction, corner point detection, and piece type recognition in the first phase, and boundary shape matching, image merging, and connectedness recovery in the second phase.

The summary of the first phase is given in (Yao et al., 2002). The following is a summary of the

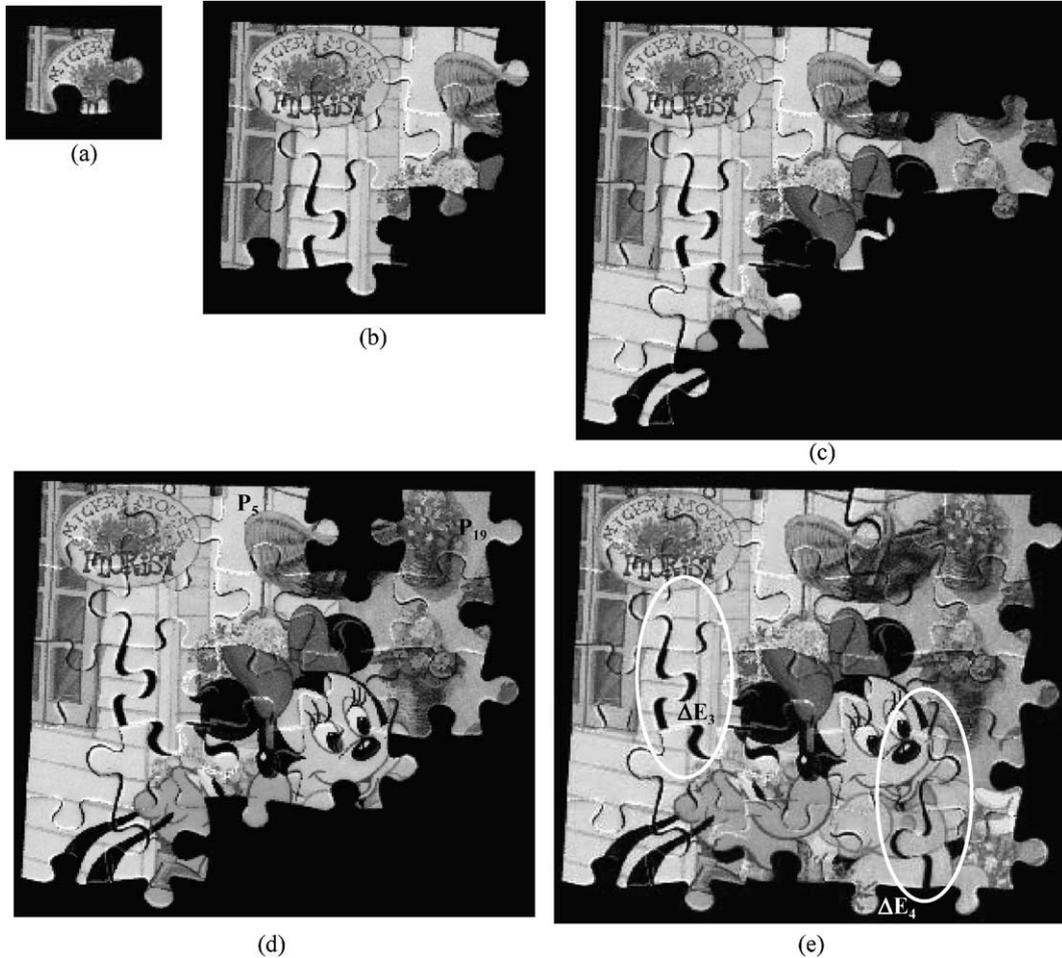


Fig. 10. Recovery process of the connection relationships in the input image in Fig. 8 (©Disney). Repetition times: (a) $c = 0$, (b) $c = 5$, (c) $c = 10$, (d) $c = 15$, (e) $c = 23$.

Table 6
Computation time to solve the jigsaw puzzles by computer and people

	Ex 1	Ex 2
Step 1 (ms)	3985	4066
Step 2 (ms)	11,928	10,765
Step 3 (ms)	120	1032
Step 4 (ms)	4647	7370
Step 5 (ms)	10,905	11,817
Step 6 (ms)	30,173	30,895
Total (ms)	61,758	65,945

second phase. The computation time of boundary shape matching is given by

$$T_{\text{shape-match}} = t_{\text{match}} \times O(16 \times N \times (N - 1)) \quad (19)$$

where N is the number of jigsaw pieces in the input image, and t_{match} is the matching time between two edges of two different pieces. t_{match} contains the time for copying the coordinates of the points on edges, for translating and rotating the coordinates of an edge in matching, and for calculating the matching error. It is directly proportional to the edge length, which depends on the size of the input image. All edges are matched to each other to get the TME candidates. The matching results are sorted by the matching error from small to big, and the top $M (= 6)$ of them are considered as the TME

candidates. If the N becomes large, the value of M should be also set larger. When calculating the matching error between the edges of two different jigsaw pieces, the weighting coefficient W (in Eq. (5)) is employed. Eq. (5) considers two factors—the area surrounded by two edges and the difference in lengths of the edges of two different pieces. The weighting coefficient W balances the affects of these two factors. Let us look at the following case where W is set at a very small value, $W = 0$. The edges of two pieces may have different lengths, but the concave portion of an edge matches the convex portion of another edge completely. In this case, these two edges may be selected as TME because of $W = 0$. This is obviously wrong. On the other hand, if W is set at very big value, the following case can be considered: For two edges of differing pieces, their lengths being almost the same (the difference between them is very small), and the concave portion of an edge matches the convex portion of another edge completely, the two edges might not be selected as TME because W is too big. This is also wrong. The setting of W is very important, and is determined experimentally. At present it is set at 100. We also tested algorithm with 150 and 200, the experiment results were the same.

The object of image merging is to calculate the integration degree of image in the region near the edge in question and those of the TME candidates. The computation time is given by

$$T_{\text{image-merge}} = t_{\text{merge}} \times O(M) \quad (20)$$

where t_{merge} is the time for calculating the integration degree of the images in the regions near the edge in question and those near its TME candidates. It includes the time for spreading the image in the region along the edge into rectangular regions, and for calculating the integration degree between the corresponding small regions of two rectangular regions. The small region's width is 10 dots, and its height is 20 dots. The experiment results show that they are appropriate. The values may be changed for the jigsaw puzzle images at different resolutions.

We made three decision rules and one correction rule to determine the TME from its candidates, based on the boundary shape matching error and the image integration degree. These two

parameters are criteria with very different characteristics. The image integration degree ranges from 0.1 to 1.0, however, the value of the boundary shape matching error is usually very big (for the best matched edges, its value is 0). The parameter τ_{thres} intends to balance these two parameters. If τ_{thres} is set too small, for example a value near to 1, it is difficult to give the priority to the boundary shape matching error when determining TME. Similarly, If τ_{thres} is set too big, for example a value larger than 2, it is hard to give the priority to the image integration degree when deciding TME. At present, it is determined experimentally, and set at 1.5. And if the TME cannot be determined by using rules 1–3 and a correction rule, then TME candidates are left unsolved. If an edge of a piece is best fit at two different locations, it is necessary to check other edges of that piece. For large size jigsaw puzzles, these rules need to be improved. This will be done in succeeding work. The computation time for recovering the connection relationships among jigsaw puzzles is given by

$$T_{\text{recovery}} = t_{\text{recovery}} \times O(N) \quad (21)$$

where t_{recovery} is the time for copying the image data of a jigsaw piece from the input image, rotating this image data and putting them into the output image. The iteration times are smaller than $N - 1$, where N is the number of puzzle piece, because multiple interconnection relationships may be recovered in each iteration.

Although the TMEs cannot be determined at a hundred percent, the jigsaw puzzle can still be solved completely. For example, in Fig. 10(d), the neighbor piece of the third edge of P_5 cannot be determined from its TME candidates; however, it can be determined from the TME candidates of the first edge of P_{19} , as shown in Fig. 10(e).

The gaps marked by the white circles in Fig. 7(e) (represented by ΔE_1 and ΔE_2) and Fig. 10(e) (ΔE_3 and ΔE_4) are the errors in the recovery of the connection relationships among jigsaw pieces. There are two factors causing these errors. Firstly, because the image is scanned from top to bottom, the brightness of the edges on the top-side and bottom-side are different, and shadows appear on the bottom-sides (see Figs. 5 and 8), and this gives rise to the deformation of boundary curves. Sec-

only, for an edge of a jigsaw piece, when it matches best with an edge of another piece, it may be that its other edges maybe are not in the best matched orientation with edges of other pieces. This may cause an error in piece orientation. The gaps become larger and larger as the recovery process continues. This is because the errors are integrated. The gaps can be minimized by making the current piece fit with its four neighboring pieces at best orientation. This is our future goal.

At present, we tested this method by using an image containing dozens of pieces. It is also necessary to test it by using images containing more pieces. As an application of the method to solve the jigsaw puzzles, it can be directly applied to intelligent robot assembly systems. This is left to do in the future.

Acknowledgements

This work is partially supported by the Kayamori Foundation of Informational Science Advancement.

References

- Chung, M.G., Fleck, M.M., Forsyth, D., 1998. Jigsaw puzzle solver using shape and color. *Proc. ICSP'98 Fourth International Conf. on Signal Processing*, Beijing, China, 12–16 October 1998.
- Freeman, H., Garder, L., 1964. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Trans. Electron. Comput.* EC-13, 118–127.
- Fukui, K., 1997. Contour extraction method based on separability of image features. *Trans. IEICE J80-D-II* (6), 1406–1414.
- Glassner, A., 2002. Putting the pieces together. *IEEE Comput. Graphics Appl.* (May/June), 76–86.
- Kosiba, D.A., Devaux, P.M., Balasubramanian, S., Gandhi, T.L., Kasturi, R., 1994. An automatic jigsaw puzzle solver. *Proc. 12th IAPR Internat. Conf. on Pattern Recognition*, Part I, October 9–13, Jerusalem, Israel.
- Otsu, N., 1980. An automatic threshold selection method based on discriminant and least squares criteria. *Trans. IEICE J63-D* (4), 349–356.
- Radack, G.M., Badler, N.I., 1982. Jigsaw puzzle matching using a boundary-centered polar encoding. *Comput. Graphics Image Process.* 19, 1–17.
- Suganthan, P.N., 1999. Solving jigsaw puzzles using Hopfield neural networks. *Proc. Internat. Conf. on Neural Networks*, Washington, DC, USA, 10–16 July 1999.
- Webster, R.W., LaFollette, P.S., Stafford, R.L., 1991. Isthmus critical points for solving jigsaw puzzles in computer vision. *IEEE Trans. Systems Man Cybernet.* 21 (5), 1271–1278.
- Yao, F.H., Shao, G.F., Tamaki, A., Kato, K., 1997. Recovery of connection relationships among two-dimensional objects. *IPSJ J.* 38 (4), 746–761.
- Yao, F.H., Shao, G.F., Yamada, H., Kato, K., 2002. Solving the jigsaw puzzles by the computer (Part I)—Jigsaw pieces extraction, corner point detection, and piece classification and recognition. *IPSJ J.* 43 (5), 1492–1504.