Regular Paper

Solving Jigsaw Puzzles by Computer (Part I) ——Jigsaw Pieces Extraction, Corner Point Detection, and Piece Classification and Recognition

Fenghui Yao,† Guifeng Shao,†† Hiroaki Yamada† and Kiyoshi Kato†

We proposed a method to solve jigsaw puzzles by computer. This method employs both the piece boundary shape information and the piece boundary image information. Firstly, the jigsaw pieces are extracted from the input image. Then, the corner points of jigsaw pieces are detected, and the piece classification and recognition are performed based on the jigsaw piece models. Next, the connection relationships are calculated according to piece boundary shape matching and image merging. Finally, the connection relationships among the pieces are recovered. This paper, as the first part, is limited to describe jigsaw piece extraction, jigsaw piece model construction, corner point detection, and piece classification and recognition.

1. Introduction

A match between the world champion of international chess and the IBM supercomputer Deep Blue gathered the attention of the chess fans, engineers, researchers and scientists from the whole world [10]. The win of Deep Blue became a big encouragement for computer engineers and scientists. In Japan, computer Shogi (the Japanese chess) has been researched actively since some years. However, the computer still can not beat a human champion of Shogi. Together with the international chess and Shogi, the jigsaw puzzle is also a very difficult problem for a computer to solve. The two-dimensional jigsaw puzzle is a one-person game in which one is given a picture cut into many irregularly shaped pieces; the objective of the game is to fit the pieces together to reconstruct the original picture. Different with the computer chess and Shogi in which the knowledge and reasoning are mainly employed, the jigsaw puzzles involve the application of concepts of pattern recognition to both the shapes of the pieces and the pictorial information on them. That is why this problem has rarely been done. To the author's knowledge, two studies of how to deal with this problem have been carried out by Freeman and Garder [2] and Yao, et al. [7].

As pointed by Freeman and Garder, the jigsaw puzzles have the following characteristics.

1) *Orientation*: Jigsaw puzzles pieces are usually given without information about their orientation.

2) *Connectedness*: An assembled puzzle may cover a simple-connected area, or it may have "holes" in it and thus be multiply-connected. Alternately, a given set of pieces may assemble into two or more disjoint areas, in which case the set of pieces is simply a mixture of two or more puzzles.

3) *Exterior boundary*: For some jigsaw puzzles the exterior boundary is known to be rectangular, and the length and width may or may not be known. In others the boundary may be irregular and either known or unknown.

4) *Uniqueness*: Most commercially available puzzles are unique, i.e., the pieces can be assembled properly in only one way.

5) *Radiality*: It refers to the kinds of interior and exterior junction in the assembled puzzle. A *triradial* junction is the junction of three boundary lines. A *quadradial* junction joins four boundary lines and a *quintradial* junction joins five. These characteristics are important when to solve the jigsaw puzzles by the computer.

The work of Freeman and Garder was done in 1964. It is a fundamental work about this problem. Because the language limitation of the computer, the resolution limitation of digitizer and the limitation of imaging device, they only showed how to solve this problem by using the piece boundary shape information.

The work of Yao, et al. was done recently. It is a preparatory work to solve a jigsaw puzzle

† Graduate School of Integrated Science & Arts, University of East Asia
†† Department of Commercial Science, Seinan Gakuin University

by computer. It pointed out the problems existing in the work done by Freeman and Garder, and proposed a new method to solve the jigsaw puzzle problem by also employing the piece boundary shape information.

Both researches did not show how to use the image information to solve jigsaw puzzle. Here, we propose a method to solve a jigsaw puzzle by using both the piece boundary shape information and the piece boundary image information. In our method, the jigsaw pieces are firstly extracted from the input image. Then, the corner points of jigsaw pieces are detected. Next, the piece classification and recognition are performed based on the jigsaw models. Connection relationships are calculated according to the piece boundary shape matching and image merging. Finally, the connection relationships among the pieces are recovered. This paper, as part I of our work, is limited to describe the jigsaw piece extraction, the jigsaw model construction, the corner point detection, and the piece classification and recognition [6]. Part II, another paper, will relate jigsaw piece boundary shape matching, the image merging, and the recovery of the connection relationships of the jigsaw puzzles [8].

The organization of the rest of this paper is as follows. In section 2, the motivation and the strategies to solve a jigsaw puzzle problem are related. In section 3, we briefly review the work done by Freeman and Garder, and that by Yao, et al., and point out problems with these algorithms. Section 4 introduces the symbols and notations employed in the description of our method. Section 5 describes the jigsaw models. Section 6 gives an overview of the methods employed in this paper. Section 7 relates the jigsaw piece edge recognition and piece type recognition. Experiment results with real-world images are given in section 8. The paper ends with some concluding remarks.

2. Motivations and Strategies

The motivation of this research is to find the scientific solution for the jigsaw puzzle problem in pattern recognition. As a scientific problem, the jigsaw puzzle problem can be defined as follows. *For $S_P = \{P_0, P_1, \cdots, P_{N-1}\}$, where P_i represents the i-th jigsaw piece ($i \in \{0, 1, \cdots, N-1\}$), and has attribute of closed boundary and solid texture, and for P_i, there exists $P_j (i \neq j, i, j \in \{0, 1, \cdots, N-1\})$ so that a part of the boundary of P_i is completely same with a part of that of P_j, and the textures nearby these two parts of boundaries are most similar, that is, P_i and P_j are neighborhood (or can be connected), then find the neighbor for P_k ($k \in \{0, 1, \cdots, N-1\}$) and make all pieces in S_P be connected into one large piece.* The strategies to solve this problem by computer can be top-down or bottom-up methodologies. The top-down methodology is like the process that the jigsaw puzzle was made. Usually, to generate a jigsaw puzzle, a large piece of picture with regular shape is cut into plenty of 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*. The bottom-up methodology intends to find the neighbor piece for a given piece in S_P by using the shape and texture information. Generally, the cutting-rules are not given to the player, and because the bottom-up methodology is similar to the process that the human people solve it by hand, we employed the bottom-up methodology to solve this problem. The following shows the details.

3. Brief Review of Previous Works

As mentioned in Section 1, two works have been carried out by Freeman and Garder, and Yao, et al., on how to solve jigsaw puzzles by computer.

Freeman and Garder's algorithm for solving a jigsaw puzzle may be briefly summarized as follows: 1) Express the jigsaw pieces by means of chain-encoding boundary curves, and find the slope-discontinuity points or curvature inflection points if very few or no slope-discontinuities exist. 2) Separate the boundary curves into a set of chainlets, each of that is likely to mate with one and only one chainlet from another set, and then compute their features. 3) Calculate the feature separation between a given chainlet and all chainlets in the set of pieces, to obtain the mating candidates.

This algorithm works well if the slope-discontinuity points can be obtained correctly. However, it is difficult to determine the slope-discontinuity points uniformly and steady because of boundary noise. This gives rise to the wrong segmentation of the boundary curves, with the result that the chainlet candidates and the mating chainlet cannot be determined correctly. This is because the segmentation of a boundary curve at slope-discontinuity points is a kind of coarse segmentation.

To solve this problem, Yao, et al. proposed to perform fine segmentation at points with local maxima of the absolute curvature, and to separate the boundary curve into boundary curve segments rather than chainlets. Owing to this fine segmentation, the features defined in Freeman and Garder's work can no longer be used. Yao, et al.'s algorithm can be briefly summarized as following: 1) Express the pieces by their boundary curves, and extract the dominant points of each curve; 2) Separate the curves into sets of curve segments by taking the dominant points as the separation points, perform matching between every two curve segments that belong to two different pieces, and calculate the minimal matching error between them; and 3) Detect the longest consecutive matched curve segments, and then recover the connection relationships among pieces.

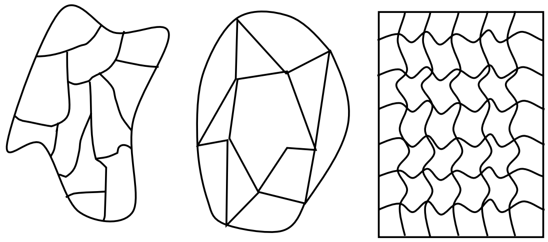
Yao, et al.'s algorithm works well. It also has the following problem. Because the boundary curve is separated into the curve segment at the point with local maxima of the absolute curvature, the total number of the curve segments in a given jigsaw puzzle becomes big, and the total number of matching among the curve segments becomes huge. This makes the detection of the longest consecutive matched curve segments be difficult. And also, the matching between two curve segments belonging to two different pieces is performed by rotating one curve segment from 0° to 360° with the step of 1°, and then calculating the matching error after each rotation, to obtain the best fitted orientation. This is very time-consuming.

To solve the problems in Yao, et al.'s algorithm, we propose to separate the piece boundary curve at the corner point. This paper describes the jigsaw piece extraction, corner point detection, and the jigsaw piece classification and recognition. For convenience in describing the details of our method, we give some definitions and introduce the some symbols and notations in the next section.

4. Definitions, Symbols, and Notations

Before explaining the definitions, symbols, and notations, let us make it clear that what kinds jigsaw puzzles we deal with in our work.

Figure 1 shows three types of jigsaw puzzle configuration. The puzzle of Fig. 1 (a) has only triradial junctions, and that of Fig. 1 (b) has only quadradial junctions. Figure 1 (c) shows the configuration of the most popular and most commercially available jigsaw puzzles (shorted as MPMCA jigsaw puzzles). It has triradial junctions on its exterior boundary and quadradial junctions in its interior. This is the target jigsaw puzzles of our research. In the following, the jigsaw puzzle means the MPMCA jigsaw puzzle that takes the structure as shown in Fig. 1 (c), if no specific explanation.



(a) (b) (c)

Fig. 1 Jigsaw puzzle configuration and junction radiality.

The objective of our work is to let the computer arrange a set of given jigsaw pieces into a single, well-fitting structure, with no gaps left between adjacent pieces.

Next, let us describe the definitions, symbols, and notations.

4.1 Jigsaw Pieces

We use P_i to denote the i-th piece, and the set S_P to denote all pieces, that is,

$$S_P = \{P_0, P_1, \cdots, P_{N-1}\}, \tag{1}$$

where N is the total number of the jigsaw pieces.

4.2 Jigsaw Piece Boundary Curves

For $P_i \in S_P (i = 0, 1, \cdots, N-1)$, its boundary curve is denoted by B_i, which is a closed curve. In Cartesian coordinates, the boundary curve B_i is expressed by its coordinate functions $x_i(s)$ and $y_i(s)$, where s is a path-length variable along the curve. All boundary curves of pieces in S_P are denoted by

$$S_B = \{B_0, B_1, \cdots, B_{N-1}\}. \tag{2}$$

In the following description, the boundary curve B_i is simply called curve B_i if this does not cause confusion.

4.3 Curvature

For the curve $B_i \in S_B (i = 0, 1, \cdots, N-1)$, the curvature at any point M is defined as the instantaneous rate of change of α, which is the angle subtended by the tangent at point M with the x axis, with respect to the arc-length s, given by

$$K(M) = \lim_{\triangle s \to 0} \frac{\triangle \alpha}{\triangle s}. \tag{3}$$

The curvature function can be defined in terms of the derivative of the coordinate functions $x_i(s)$ and $y_i(s)$ as

$$K(x_i, y_i) = \frac{\partial^2 y_i}{\partial x_i^2} \Big/ [1 + (\frac{\partial y_i}{\partial x_i})^2]^{\frac{3}{2}}. \qquad (4)$$

A simple form of the curvature, derived by Rattarangsi and Chin [5]), is given by

$$K(x_i, y_i) = \dot{x}_i \ddot{y}_i - \ddot{x}_i \dot{y}_i. \qquad (5)$$

All curvature functions of curves in S_B are denoted by

$$S_K = \{K_0, K_1, \cdot, K_{N-1}\}. \qquad (6)$$

4.4 Dominant Point

A *dominant point* is normally considered as a point on the curve in which the absolute curvature has a local maximum. For $B_i \in S_B$, its k-th dominant point is denoted by D_i^k, and all its dominant points are denoted by

$$S_{D_i} = \{D_i^0, D_i^1, \cdots, D_i^{N_i-1}\}, \qquad (7)$$

where N_i is the total number of dominant points of the curve B_i. The dominant points are numbered clockwise.

4.5 Interior Angle of the Dominant Point

As shown later in Section 6.2.1, to detect the dominant points, a searching procedure is applied to detect the local maximum of the absolute curvature within the *region of support* given by the sequence $\{|K_i^l|, \cdots, |K_i^{k-1}|, |K_i^k|, |K_i^{k+1}|, \cdots, |K_i^r|\}$, where K_i^k is the curvature of the point in question, and K_i^l and K_i^r are the curvature at the leftmost point M_l and rightmost point M_r of the local region of support, respectively. The region of support for each point M_k is the largest possible window containing M_k in which $|K_i|$ to both the left and right of M_k is strictly decreasing. Then, M_k is considered as a dominant point, and is denoted as D_i^k. The angle formed by the three points M_l, D_i^k, and M_r, i.e., $\angle M_l D_i^k M_r$, is defined as the interior angle of the dominant point D_i^k, and is simply denoted by θ_i^k. The interior angles of all dominant points in P_i are expressed as

$$S_{\theta_i} = \{\theta_i^0, \theta_i^1, \cdots, \theta_i^{D_i-1}\}. \qquad (8)$$

4.6 Corner Point and Its Interior Angle

The point, where the junction exists, is defined as the *corner point*. For an MPMCA jigsaw puzzle, each piece has four corner points.

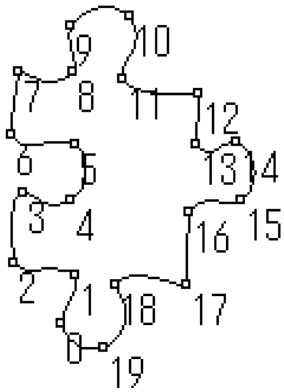


Fig. 2 Boundary curve of a Jigsaw piece included in a real-world image.

Corner points are included in dominant points, and they are special dominants. To differentiate from the dominant points, the corner points and their corresponding interior angle are, respectively, denoted by

$$S_{C_i} = \{C_i^0, C_i^1, C_i^2, C_i^3\} \qquad (9)$$

$$S_{\alpha_i} = \{\alpha_i^0, \alpha_i^1, \alpha_i^2, \alpha_i^3\}. \qquad (10)$$

Figure 2 shows an example of the boundary curve of a jigsaw piece in a real-world image. The points numbered from 0 to 19 clockwise are dominant points, and the points numbered as 2, 7, 12, 17 are the corner points which are renumbered from 0 to 3.

4.7 Jigsaw Piece Edge

For $B_i \in S_B$, the curve is segmented at the corner points in S_{C_i}. The curve between two consecutive corner points C_i^k and C_i^{k+1} is defined as a *Jigsaw piece edge*, or simply called *edge*, where $k = 0, 1, 2, 3$ (modulo 4). Every piece of MPMCA jigsaw puzzles has four edges. An edge of jigsaw piece P_i, when it is traced from its initium m to its terminus n clockwise, is denoted as E_i^{mn}, and is denoted as $\hat{E}_i^{nm}$ when traced from its terminus n to its initium m, counterclockwise. Clockwise edges and counterclockwise edges are respectively denoted by

$$S_{E_i} = \{E_i^{01}, E_i^{12}, E_i^{23}, E_i^{30}\} \qquad (11)$$

$$S_{\hat{E}_i} = \{\hat{E}_i^{10}, \hat{E}_i^{21}, \hat{E}_i^{32}, \hat{E}_i^{03}\}. \qquad (12)$$

4.8 Central Angle between Two Dominant Points

Let O_i express the centroid of P_i. The angle formed by the three points D_i^k, O_i, and D_i^l, i.e., $\angle D_i^k O_i D_i^l$, is called the central angle between D_i^k and D_i^l, and is denoted by γ_i^{kl}.

Moreover, to recognize jigsaw pieces, it is necessary to construct a jigsaw model. This is given in the next section.

5. Model Construction of Jigsaw Pieces

To reduce the computation time, jigsaw pieces should be roughly categorized by using the shape information because the number of the boundary shapes of the jigsaw pieces is very small. To do this way, firstly, it is necessary to recognize the boundary shape of jigsaw pieces. Therefore, the jigsaw piece models are needed.

When the people solve the MPMCA jigsaw puzzle by hand, they firstly focus on the boundary of a jigsaw piece and separate it into four edges at four corner points. And then they try to find their best-matched neighbor pieces by matching the shape and image texture of each edge with those of other pieces. Each of the four edges can only be one of the following three patterns, “—”, “‾⊔‾” and “_⊓_”. The first one is the straight line and denoted by “L”. The second is the curve that is concave to the centroid of the jigsaw piece and denoted by “C”. The third one is the curve that is convex to the centroid of the jigsaw piece and denoted by “V”. The edges of any MPMCA jigsaw pieces are comprised of these three kinds of patterns. These three kinds of patterns 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 C, is located at the corner of the whole jigsaw puzzle. There exist four corner pieces in a jigsaw puzzle no matter how big the number of the total pieces. Considering all combinations of three kinds of edge patterns, the corner pieces are further classified into four type as shown in the first row of **Table 1**, which are named as C_0, C_1, C_2 and C_3, respectively, and the edge configurations are “CCLL”, “VCLL”, “CVLL”, and “VVLL”, correspondingly, as given in **Table 2**. Here, it is necessary to notice that “CLLC” is the same with “CCLL” because “CCLL” can be obtained by rotating “CLLC” or by changing the start corner point. This is same to other pieces hereafter. In Table 1, the start corner point for all patterns is set at left-bottom corner, and the edges are traced clockwise. It is also necessary to notice that the piece type C means the “corner piece”, and the edge type C means “concave”.

Table 1 Jigsaw pieces classification and their model patterns.

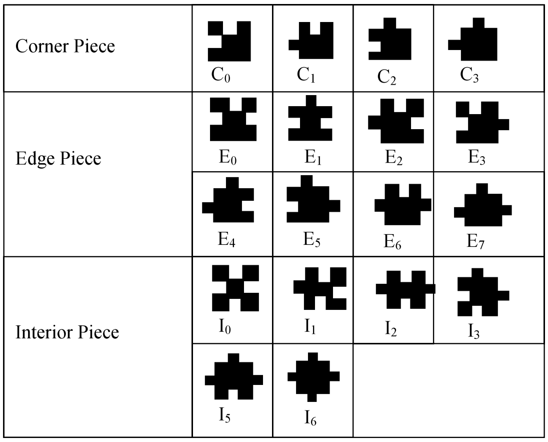


Table 2 Jigsaw piece boundary description and each edge's neighbor piece type and edge type.

C	Piece Type	Boundary	Required Neighbor Piece and Edge Type: Edge 0	Edge 1	Edge 2	Edge 3
C	C_0	CCLL	E.V	E.V	X	X
	C_1	VCLL	E.C	E.V	X	X
	C_2	CVLL	E.V	E.C	X	X
	C_3	VVLL	E.C	E.C	X	X
E	E_0	CCCL	E.V, C.V	I.V	E.V, C.V	X
	E_1	CVCL	E.V, C.V	I.C	E.V, C.V	X
	E_2	VCCL	E.C, C.C	I.V	E.V, C.V	X
	E_3	CCVL	E.V, C.V	I.V	E.C, C.C	X
	E_4	VVCL	E.C, C.C	I.C	E.V, C.V	X
	E_5	CVVL	E.V, C.V	I.C	E.C, C.C	X
	E_6	VCVL	E.C, C.C	I.V	E.C, C.C	X
	E_7	VVVL	E.C, C.C	I.C	E.C, C.C	X
I	I_0	CCCC	E.V, I.V	E.V, I.V	E.V, I.V	E.V, I.V
	I_1	VCCC	E.C, I.C	E.V, I.V	E.V, I.V	E.V, I.V
	I_2	VCVC	E.C, I.C	E.V, I.V	E.C, I.C	E.V, I.V
	I_3	CVVC	E.V, I.V	E.C, I.C	E.C, I.C	E.V, I.V
	I_4	VVVC	E.C, I.C	E.C, I.C	E.C, I.C	E.V, I.V
	I_5	VVVV	E.C, I.C	E.C, I.C	E.C, I.C	E.C, I.C

The edge piece, denoted by E, is located in the four edges of the whole jigsaw puzzle. The number of edge pieces change according with the size of the jigsaw puzzle. Taking account of all combinations of three kinds of edge patterns, the edge pieces are classified into eight types as shown in the second and third rows of Table 1, which are named as $E_0, E_1, \cdots, E_7$, and the edge configurations are “CCCL”, “CVCL”, $\cdots$, “VVVL”, sequentially, as shown in Table 2.

The interior piece, denoted by I, is located in the interior area of the whole jigsaw puzzle. The number of the interior pieces also changes with the size of a jigsaw puzzle, and is much bigger than that of edge pieces. Counting all combinations of three kinds of edge patterns, the interior pieces are classified into six types as shown in the fourth and fifth row of Table 1, which are named as $I_0, I_1, \cdots, I_5$, and the edge configuration are expressed as "CCCC","VCCC", $\cdots$, "VVVV", correspondingly, as shown in Table 2.

From above consideration, we can conclude that there are 18 kinds of pieces in an MPMCA jigsaw puzzle. Table 1 shows their pattern images, which are denoted by,

$$S_M = \{C_0, \cdots, C_3, E_0, \cdots, E_7, I_0, \cdots, I_5\}. \tag{13}$$

Let us employ the notation "$W.U$" to express the edge connection relationships among jigsaw pieces, where W means the neighbor piece type, and U the neighbor edge type. Then the connection relationships among these 18 kinds of jigsaw pieces are summarized in Table 2. The first column of Table 2 shows the jigsaw piece categories, the second the piece types in each category, and the third the piece boundary configuration starting from the left-bottom corner point of each piece pattern, clockwise. The 4-th, 5-th, 6-th, and 7-th column shows the required neighbor piece type and neighbor edge type of the four edges, i.e., E^{01}, E^{12}, E^{23}, and E^{30}, respectively. For the corner pieces, there exist two kinds of relationships — $E.V$ and $E.C$, as shown in the 4-th and 5-th column in Table 2. $E.V$ means that the required neighbor piece is the edge piece, and the required neighbor piece edge is "convex" edge. Similarly, for the edge pieces, there exist 6 kinds of relationships — $E.V, E.C, C.V, C.C, I.V$, and $I.C$. And for the interior pieces, there exist 4 kinds of relationships — $E.V, E.C, I.V$, and $I.C$. And X in the 6-th and 7-th column means there does not exist the neighbor piece.

Table 2 shows all possible connection relationships existing in a MPMCA jigsaw puzzle. This table takes very important role in solving a jigsaw puzzle by computer.

6. Algorithm Overview

Our algorithm to solve the jigsaw puzzle can be formulated in six steps:

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

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

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

Step 4 Perform the boundary shape matching to get the candidates of the neighbor pieces to reduce the computation time because the number of boundary shapes are very small.

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

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

Step 1, 2, and 3 are explained in greater detail below. Step 4, 5, and 6 are given in Part II, another paper [8].

6.1 Jigsaw Piece Extraction

Jigsaw piece extraction is a problem belonging to the field of object extraction. Roughly speaking, there is no difference between contour extraction and object extraction because the object extraction is mainly realized via the contour extraction and the goal of the contour extraction is to extract the objects. These two words will be considered as the same through this paper if no specific notification. The contour extraction is so important that it is impossible to perform the object classification and recognition if the contours of objects are not extracted correctly. There is an abundant literature on this topic. This problem can be further classified into the following four categories. They are I) contour extraction of objects with the monochromic texture from the uniform background, II) contour extraction of objects with the monochromic texture from the complicated background (e.g., the indoor natural scene), III) contour extraction of objects with the complicated texture from the uniform background; and IV) contour extraction of objects with the complicated texture from the complicated background. For the problems in category I, it can be solved by applying the existed binarization and contour-tracing algorithm. This method has already been used in the practical industry systems. The problems in category II are represented by the human face extraction from the natural scene [3]. The problems in category III and IV have rarely been discussed.

The problem of jigsaw piece extraction belongs to category III. This kind of problem is difficult since the textures are complicated and they may include the same texture as that of the background. 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 image. The last step is to perform the conditional contraction to the mask image.

This method requires that the jigsaw pieces be in the same position in the multiple input images with a different background color, and that the jigsaw pieces do not overlap each other but keep apart enough to be extracted independently. To satisfy this requirement, the following contrivances are utilized to make the input image. i) When to input the images by a camera, the jigsaw pieces are put on a hard and transparent plate (e.g., a piece of glass) which is fixed on the table and kept an opening with the table in order to change the background color sheet. The camera is set upward. After an image is taken, the background color sheet is changed into another one with a different color as shown in **Fig. 3** (a). ii) When the image is input by a scanner, the jigsaw pieces are put on the top of the scanner, and pressed by a hard and transparent plate. The background color sheet is put on the top of the plate as shown in Fig. 3 (b). After an image is scanned, the background color sheet is changed into another one with a different color. With these techniques, the jigsaw pieces are kept at the same position in the different input images with different background colors. Usually, the more the background colors are, the better this method works. At least, three components, such as red, green and blue are needed. To ensure that this method works well, another two background colors, such as black and white, are employed. Therefore, at present, red, green, blue, white and black are selected as the background colors. Details of this method are described below.

(1) Mask Image

The input image with red, green, blue, white, and black background is binarized at a predetermined threshold, respectively. Because the background colors are known, the threshold can be determined easily. These five images, after being binarized, are conjugated to make the mask image by using OR operation.

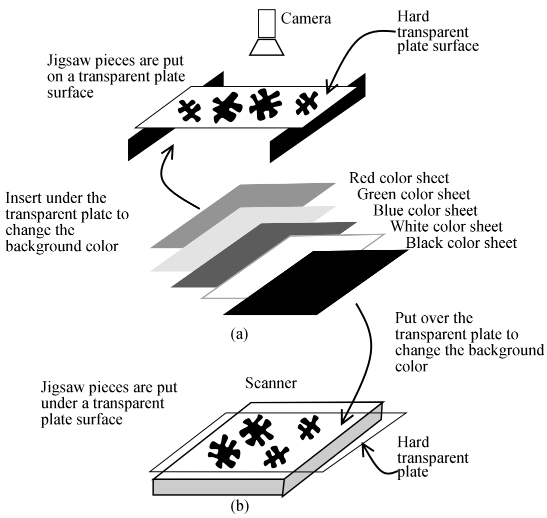


Fig. 3 (a) Input images with five different background colors from video camera; (b) input images from the scanner.

(2) Evaluation Image

The input image with a black background is binarized by the automatic threshold selection method [4]. The output image is used as the evaluation image. The evaluation image is used as the condition to stop the conditional contraction of the mask image. Because the mask image is made from multiple images with different background colors, by applying the automatic threshold selection method to each of them and then employing OR operation to all of them, the mask images contains the shades the of jigsaw pieces and the spread of the contours of the jigsaw pieces. To reason the contours of the jigsaw pieces, a conditional contraction is performed as described below.

(3) Conditional Contraction

The processing at this step is to make the mask image converge to the evaluation image by the conditional contraction. The conditional contraction of the mask image is defined as follows:

If the contraction processing of the mask image encounters a point where its corresponding point on the evaluation image is not 0*, this point will not be eliminated. That is,* $G(x, y)$ *is set to* 0*, if and only if* $F_M(x, y)$ *is equal to* 0 *or any of its* 4 *neighbors (or* 8 *neighbors) is equal to* 0*, and* $F_E(x, y)$ *is also equal to* 0*, otherwise,* $G(x, y)$ *is set to* 1*.*

Where $G(x, y)$ is the output of the contraction, F_M and F_E are the mask image and evaluation image, respectively. F_E is used as the condition during the contraction processing.

The processing at this step includes:

1) Perform the conditional contraction to the mask image F_M;
2) Apply the contour-tracing algorithm to detect the contours from the mask image;
3) Calculate the degree of accordance between the contour detected from the mask image F_M and the evaluation image F_E. Let ε denote the degree of accordance, ε is defined by
$$\varepsilon = \frac{N_E}{N_T}, \tag{14}$$
where N_T is the total number of the points on the contour, and N_E is the number of the contour points which lie on the evaluation image.

These three operations are repeated until ε exceeds the predetermined threshold ε_T. As soon as this repetition stops, the contours obtained at the step 2) are considered as the contours (jigsaw piece boundary) we are searching for.

Figure 4 (a)∼(e) show the five input images with black, blue, green, red, and white background, respectively, (f) the mask image, (g) the evaluation image, and (h) the extracted jigsaw pieces, respectively.

6.2 Corner Point Detection

Since the corner points are the special dominants, it needs to relate the dominant point detection before talking about the corner point detection. The corner point detection is the succeeding processing of dominant point detection. There is much literature on the dominant point detection [5),7),9)]. The method used in Yao, et al's work is cited here, and is briefly summarized in Section 6.2.1. The corner point detection will be described in Section 6.2.2.

6.2.1 Dominant Point Detection

The dominant point detection based on multi-scale curvature is employed here [5)]. Let us represent the coordinate functions $x_i(s)$ and $y_i(s)$ of the jigsaw boundary curve B_i digitally by a set of equally spaced Cartesian grid samples $\{x_i^k, y_i^k\}$ for $k = 0, 1, \cdots, S_i - 1$ (modulo S_i). The digital form of the curvature in equation (5) at point k on the curve B_i is computed by
$$K_i^k = \triangle x_i^k \triangle^2 y_i^k - \triangle^2 x_i^k \triangle y_i^k, \tag{15}$$
where $\triangle$ denotes the difference operator and $\triangle^2$ is the second-order difference operator.

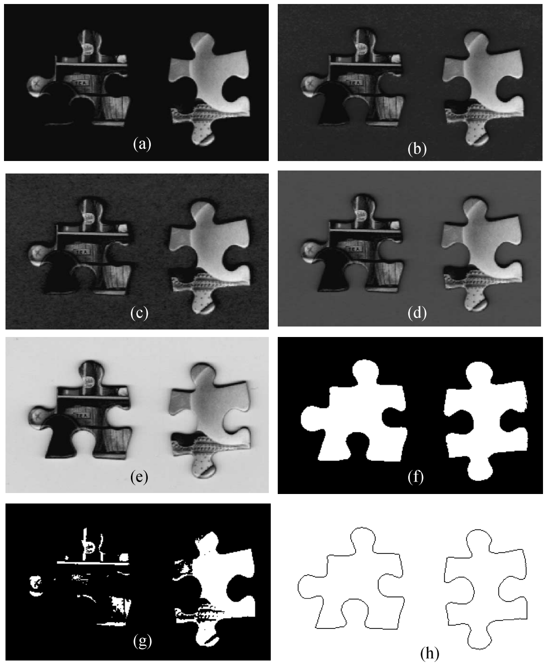


Fig. 4 (a)∼(e) are the five input images with black, blue, green, red, and white background, respectively; (f) is the mask image; (g) is the evaluation image; and (h) is the extracted jigsaw pieces.

The digital Gaussian function in Burt's work [1)] with a window size of $K = 3$ is used here to generate smoothing functions at various values of σ, and is given by
$$h[0] = 0.2261, h[1] = 0.5478, h[2] = 0.2261, \tag{16}$$
where $h[1]$ is the center value and $\sum h[k] = 1 (k = 0, 1, 2)$. This digital function has been mentioned as the best approximation of the Gaussian distribution [1)]. For the digital smoothing function with higher values of σ, the above $K = 3$ function is used in a repeating convolution process. For example, a $K = 5$ smoothing function is obtained by convolving Eq. (16) with itself once, and a $2(j + 1) + 1$ digital smoothing function is created by repeating the self-convolution process j times.

For the boundary curve B_i digitally represented by $\{x_i^k, y_i^k\}$ for $k = 0, 1, \cdots, S_i - 1$ (modulo S_i), which has a perimeter arc length of length S_i, the digital Gaussian smoothing function with a largest σ for the coarsest boundary representation must have a window size no larger than S_i in order to avoid aliasing.

A digital multiscale representation of the curve B_i from $\sigma = 0$ to $\sigma_{max}(= S_i/6)$ is constructed by the digital Gaussian function defined above. The multiscale digital curvature of the curve B_i can be obtained according to Eqs. (15) and (16). The next step is to determine the dominant points for a given $\sigma(0 \sim \sigma_{max})$. Details are as follows: For each point k of the curve B_i, a searching procedure is applied to detect the local maximum of the absolute curvature within the *region of support* given by the sequence $\{|K_i^l|, \cdots, |K_i^{k-1}|, |K_i^k|, |K_i^{k+1}|, \cdots, |K_i^r|\}$, where K_i^k is the curvature of the point in question, and K_i^l and K_i^r are the leftmost point M_l and rightmost point M_r of the local region of support, respectively. The region of support for each point k is the largest possible window containing k in which $|K_i|$ to both the left and right of k is strictly decreasing. A boundary point is not an absolute maximum if such a region of support cannot be determined for that point. This processing is applied for all σ from 0 to σ_{max}. When σ changes from small to big, those dominant points brought about the boundary noise will disappear. For the multiscale digital curvature map, the filtering processing is employed. After this processing, the left dominant points are considered as the final result. Details are related in literatures [5),7)].

For $B_i \in S_B(i = 0, 1, \cdots, N-1)$, the above procedure is applied, then the dominant point for $P_i \in S_P$ can be obtained.

6.2.2 Corner Point Detection

The corner points are the special dominant points. Suppose that quartet points $D_i^s, D_i^t, D_i^u, D_i^v \in S_{P_i}(i = 0, 1, \cdots, N-1, s \neq t \neq u \neq v)$ are the corner points, they have the following properties:

I Their interior angles are almost same, that is,

$$\alpha_i^s \approx \alpha_i^t \approx \alpha_i^u \approx \alpha_i^v \approx 90°. \qquad (17)$$

II The central angles between any succeeding two of them are nearly same, that is,

$$\gamma_i^{st} \approx \gamma_i^{tu} \approx \gamma_i^{uv} \approx \gamma_i^{vs} \approx 90°. \qquad (18)$$

III Their distances to the centroid of the jigsaw piece P_i are approximately same, that is,

$$||D_i^s - D_c|| \approx ||D_i^t - D_c|| \approx$$
$$||D_i^u - D_c|| \approx ||D_i^v - D_c||, \qquad (19)$$

where D_C is the centroid of P_i. This is the same in the following if without specific explanation.

IV Within the regions of support of them, the partial boundary curves are all convex from the centroid.

V The line decided by the leftmost and rightmost points of the region of support of D_i^s is nearly perpendicular to the line decided by D_i^s and D_C. This is also right to D_i^t, D_i^u, and D_i^v.

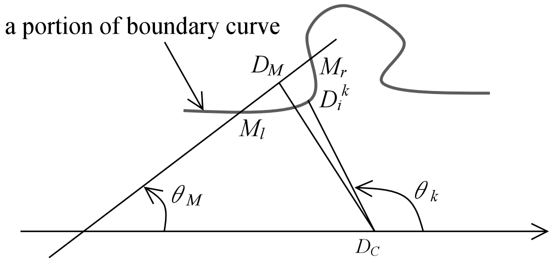


Fig. 5 Filtering processing of the dominant points.

The above properties are used to detect the corner points from the dominant points. The following gives the details.

(1) Filtering of Dominant Points

The filtering processing is based on property IV and V as shown above.

For $D_i^k \in S_{P_i}$, if $\| D_M - D_C\| > \| D_i^k - D_C \|$, the partial curve in the region of support of D_i^k is concave to the centroid of P_i, it is discarded (according to the property IV), where D_M is the intermediate point of the leftmost and rightmost points of the region of support of D_i^k (refer to **Fig. 5**).

And also, if $|\theta_k - \theta_M| < T_{\theta 1}$, or if $|\theta_k - \theta_M| > T_{\theta 2}$, D_i^k is discarded (according to the property V), where θ_k is the slope of the line passing D_i^k and D_C , θ_M is that of M_l and M_r (refer to Fig. 5), and $T_{\theta 1}$ and $T_{\theta 2}$ are the predetermined threshold values (at present they are set at 60° and 120°, respectively).

This processing is applied to all dominant points of P_i in $S_P(i = 0, 1, \cdots, N-1)$. The dominant points, after being filtered, are passed to the next step for searching the corner points.

(2) Searching the Corner Points from Dominant Points

The corner point searching procedure is as follows:

1) Suppose $D_i^j \in S_{D_i}$ is the first corner point (j is initialized with 0);

2) Search the second corner point. For D_i^k (initial value of k is $j+1$), if it satisfy

$$T_{\alpha 1} \leq \alpha_i^k \leq T_{\alpha 2}, \qquad (20)$$

$$T_{\gamma 1} \leq \gamma_i^{kj} \leq T_{\gamma 2}, \qquad (21)$$

$$\left|\frac{||D_i^j-D_c||-||D_i^k-D_c||}{||D_i^j-D_c||}\right|\leq T_{Dist}, \quad (22)$$

it is considered as the second corner point, then go to 3) to search the third corner point. Otherwise, move to the next dominant point and check whether it satisfies the above requirements or not. This is repeated till the second corner point is obtained. If all dominant points, except D_i^j in S_{D_i}, are checked, and the second corner point is not detected, j is increased by 1 and go to 1). It is necessary to notice that $T_{\alpha 1}$ and $T_{\alpha 2}$ are the predetermined threshold values for the interior angles, $T_{\gamma 1}$ and $T_{\gamma 2}$ are those of central angles, and T_{Dist} is that of relative distance of the Euclidean distance. At present, they are set at $70°, 120°, 60°, 120°$, and 0.3, respectively.

3) Search the third corner point. For D_i^l (initial value of l is $k+1$), if it satisfy the requirements in Eqs. (20), (21), and (22), it is thought of as the third corner point, then go to 4) to search the fourth corner point. Otherwise, move to the next dominant point and check whether it satisfies these three requirements or not. This is repeated till the third corner point is obtained. If all dominant points, except D_i^j and D_i^k, in S_{D_i}, are checked, and the third corner point is not detected, j is increased by 1 and go to 1).

4) Search the fourth corner point. For D_i^m (initial value of m is $l+1$), if it satisfy the requirements in Eqs. (20), (21), and (22), it is thought of as the fourth corner point, then go to 5). Otherwise, move to the next dominant point and check whether it satisfies these three requirements or not. This is repeated till the fourth corner point is obtained. If all dominant points, except D_i^j and D_i^k, and D_i^l, in S_{D_i}, are checked, and the fourth corner point is not detected, j is increased by 1 and go to 1).

5) The quartet points obtained from 1) to 4) are considered as one candidate of the quartet corner points. Repeat the processing from 1) to 4) to obtain all candidates.

6) All candidates are sorted according to the likelihood degree ξ of the rectangle. Suppose D_i^u, D_i^v, D_i^w, and D_i^x represent one quartet points of candidates corresponding to the $1st, 2nd, 3rd$ and 4-th corner point, ξ is defined as

$$\begin{aligned}\xi = &|\angle D_i^u D_i^v D_i^w - \angle D_i^w D_i^x D_i^u|+ \\ &+|\angle D_i^w D_i^u D_i^v - \angle D_i^u D_i^w D_i^x|+||D_i^u - D_i^v|| \\ &+||D_i^v - D_i^w||+||D_i^w - D_i^x||+||D_i^x - D_i^u|| \\ &+||D_i^v - D_i^x||+||D_i^w - D_i^u||. \qquad (23)\end{aligned}$$

The candidate with smallest likelihood degree of the rectangle is selected as the corner points.

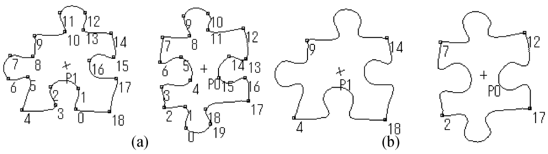


Fig. 6 (a) is the dominant points from the boundary curves in Fig. 4 (h), and (b) is the corner points detected from (a).

This processing is applied to all dominant points in $S_{D_i}(i=0,1,\cdots,N-1)$. Then we can obtain the corner points for all jigsaw pieces in S_P.

Figure 6 (a) shows the dominant points from the boundary curves in Fig. 4 (h), and (b) is the corner points detected from the dominant points in Fig. 6 (a).

7. Jigsaw Piece Classification and Recognition

The jigsaw piece classification is to determine which category the present piece belongs, and the jigsaw piece recognition is to determine which model it corresponds to, in S_M. The jigsaw piece recognition consists of the edge recognition and the piece type recognition.

7.1 Edge Recognition

For $B_i \in S_{B_i}(i=0,1,\cdots,N-1)$, it is separated into four edges at the corner points. The edge recognition is to determine which edge pattern the present edge corresponds to, the *line pattern* L, or the *concave pattern* C, or the *convex pattern* V. The following shows the details.

For the edge between the corner point C_i^k and C_i^{k+1}, that is, $E_i^{k,k+1}$ of $P_i(k=0,1,2,3,$ modulo 4), if it satisfies $h \leq ||C_i^k - C_i^{k+1}||/5$, it is thought of as a line edge L. If it satisfies $h > ||C_i^k - C_i^{k+1}||/5$, and D_M is on the left side when tracing from C_i^k to C_i^{k+1}, clockwise, it is considered as a convex edge V, otherwise it is a concave edge C. Where D_M is the most deviant point from the line l decided by C_i^k and C_i^{k+1}, and h is the distance from D_M to l, as shown in **Fig. 7**.

This processing is applied to $P_i \in S_P(i=$

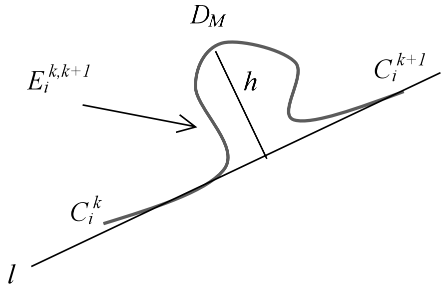


Fig. 7 Edge recognition.

$0, 1, \cdots, N-1$), and then the edge types of all pieces are obtained.

7.2 Jigsaw Piece Recognition

After the edge recognition, we obtain a four-letter string for every piece, which is comprised of "L", "C", and "V". 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 S_M. Suppose "$X_1X_2X_3X_4$" is the four-letter string of the jigsaw piece in question, the details of the matching are as follows.

Take the four-letter string of the first model in S_M, and perform the matching with "$X_1X_2X_3X_4$" , "$X_2X_3X_4X_1$" , "$X_3X_4X_1X_2$", and "$X_4X_1X_2X_3$". If the matched string is found, the piece in question is thought of as the same type with that one in S_M. Otherwise, move to the next model in S_M and repeat the above matching. This operation is repeated until the matched model is found.

This processing is applied to $P_i \in S_P (i = 0, 1, \cdots, N-1)$, and then the piece types of all pieces are obtained.

8. Experiment Results

The above algorithms are implemented on a Windows platform. The programming language is C++.

Figure 8 shows one of the experimental results. Figure 8 (a) is one of the input images with a white background, where 14 jigsaw pieces are included (Ignore red "×" marks at this time, this will be described later). Figure 8 (b) shows the detected jigsaw pieces and their corner points obtained from the dominant points. The jigsaw pieces are numbered from P_0 to P_{13} during the detection operation, automatically. The numbers on the corners of each piece are the corresponding numbers of the dominant points.

The edge recognition and piece recognition results are shown in **Fig. 9**, where corner points are renumbered from 0 to 3, in the order of searching, the small "●" shows the start corner points referring to jigsaw piece model in Table 1, and the piece type is plotted right after the piece number and "," at the centroids.

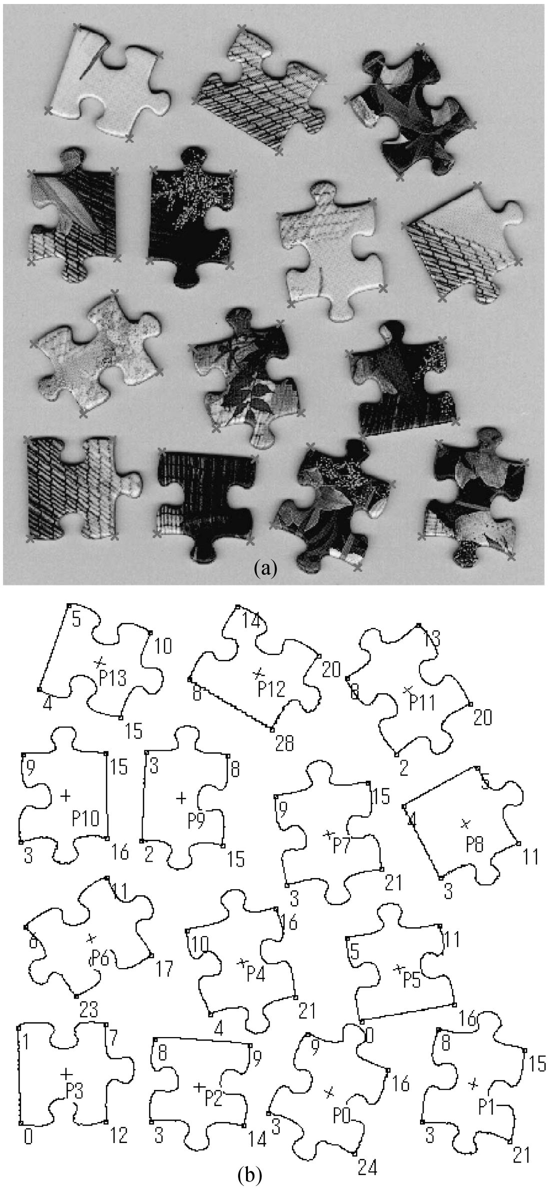


Fig. 8 (a) is one of the input image, and (b) shows the corner points obtained.

To evaluate this method quantitatively, the corner point detection error is determined as the average distance between the assigned corner points and the detected corner points, that is,

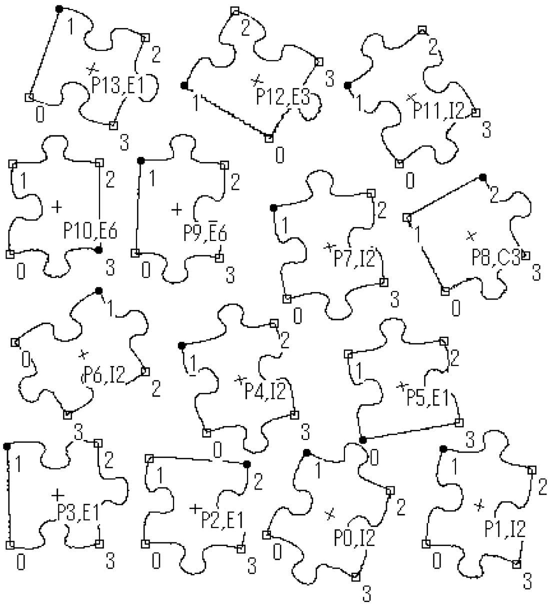


Fig. 9 Piece recognition result of the jigsaw pieces extracted from the input image in Fig. 8 (a).

$$err = \sum_{i=0}^{N-1}\sum_{j=0}^{3} ||\tilde{C}_i^j - K\times C_i^j||/(4\times N), \quad (24)$$

where $\tilde{C}_i^j$ is the assigned corner points as marked by "×" in the input image in Fig. 8 (a), C_i^j is the detected corner point, N is the total number of jigsaw pieces in the input image, and K is the shrinkage coefficient to show the whole input image on the computer screen. The position of $\tilde{C}_i^j$ is assigned by clicking the mouse button on the computer screen by the operator. In this method, the position of $\tilde{C}_i^j$ may change according to the operators. Therefore, the evaluation experiment is repeated for ten times by two operator (five times for each), the average value of err from the Eq. (24) is considered as the corner point detection error. For the input image in Fig. 8 (a), the value of K is set at 0.3 to show the image on the screen completely, and the corner point detection error is equal to 1.53 dots.

We also tested the other real-world images. The corner point detection errors are less than 2 dots. A part of these results can be seen in Part II, another paper [8].

9. Conclusions

This paper is Part I of the whole work and related jigsaw piece extraction, corner point detection, and piece classification and recognition. Real world images are used to test these algorithms, for all input images, the corner point detection errors are smaller than 2 dots. These results are satisfactory.

The algorithms for extracting the jigsaw pieces, detecting the dominant points, and searching the corner points can be directly used for the other applications.

Pieces of MPMCA jigsaw puzzles can be classified into 18 model patterns. The connection relationships among these 18 patterns are derived and summarized in Table 2. This greatly reduces the computation time and makes it possible for a computer to solve jigsaw puzzles.

The next step of our work is to recover the connection relationships among the jigsaw pieces based on the piece recognition results. This will be related in Part II, another paper.

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

References

1) Burt, P.J. and Adelson, E.H.: The Laplacian pyramid as a compact image code, *IEEE Trans. Commun.*, Vol.COM-31, No.4 (1983).

2) Freeman, H. and Garder, L.: A pictorial jigsaw puzzles: The computer solution of a problem in pattern recognition, *IEEE Trans. Electronic Comput.*, Vol.EC-13, pp.118–127, (Apr. 1964).

3) Fukui, K.: Contour Extraction Method Based on Separability of Image Features, *Trans. IEICE*, Vol.J80-D-II, No.6, pp.1406–1414 (1997).

4) Otsu, N.: An Automatic Threshold Selection Method Based on Discriminant and Least Squares Criteria, *Trans. IEICE*, Vol.J63-D, No.4, pp.349–356 (1980).

5) Rattarangsi, A. and Chin, R.T.: Scale-based detection of corners of planar curves, *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol.PAMI-14, No.4, pp.432–449 (1992).

6) Shao, G.F., Yao, F.H., Yamada, H. and Kato, K.: The Computer Solution of Jigsaw Puzzles (Part I) — Jigsaw Pieces Extraction, Corner Point Detection, and Piece Classification and Recognition, *IPSJ CVIM*, 125-1 (2001).

7) Yao, F.H., Shao, G.F., Tamaki, A. and Kato, K.: Recovery of Connection Relationships among Two-dimensional Objects, *Trans. IPSJ*, Vol.38, No.4, pp.746–761 (1997).

8) Yao, F.H., Shao, G.F., Yamada, H. and Kato, K.: The Computer Solution of Jigsaw Puzzles (Part II) — Jigsaw Piece Boundary Shape Matching, Image Merging, and Recovery of Connection Relationships, *IPSJ CVIM*, 125-2 (2001).

9) Zhu, P. and Chirlian, P.M.: On critical point detection of digital shapes, *IEEE Trans. Patt. Anal.Mach.Intell.*, Vol.PAMI-17, No.8, pp.737–748 (1995).

10) http://www.research.ibm.com/deepblue/.

(Received May 9, 2001)
(Accepted February 13, 2002)

Fenghui Yao received his Doctor of Engineering Degree from Kyushu Institute of Technology in 1992. From 1992 to 1994, he worked at Soliton Systems K.K. He worked as a research associate in Kyushu Institute of Technology, from 1994 to 1996. Currently, he is an associate professor at the Graduate School of Integrated Science and Arts, University of East Asia. His current research interests are image processing, multimedia information processing, computer architecture, and parallel processing. He is a member of IEEE, IPSJ, IEICE, and RSJ.

Guifeng Shao received her Doctor of Engineering Degree from Kyushu Institute of Technology in 1995. She is currently an associate professor at Department of Commercial Science, Seinan Gakuin University. Her research interests are natural language understanding, knowledge representation, image processing and multimedia information processing. She is a member of IPSJ.

Hiroaki Yamada is currently a professor at the Graduate School of Integrated Science and Arts, University of East Asia. Before joining in University of East Asia, he was a professor at Nihon University. He received his Ph.D. from the University of Tokyo. His research interests are image processing, ultrasonic sensor, and AI. He is a member of IEEE.

Kiyoshi Kato received his B.S. and M.S. degrees from the University of Tokyo in 1959 and 1961, respectively. He received his Doctor of Engineering Degree from the University of Tokyo in 1975. He worked in Faculty of Engineering, Kyushu Institute of Technology, from 1964 to 1997. He is currently a professor at University of East Asia. His research interests include image processing, computer application in music and education, and human interfaces. He is a member of IPSJ, IEICE, SICE, and RSJ.