New puzzle assembly

Chung, Min, Fleck, Margaret, Forsyth, David

Min Gyo Chung, Margaret M. Fleck, David A. Forsyth, "New puzzle assembly," Proc. SPIE 3545, International Symposium on Multispectral Image Processing (ISMIP'98), (25 September 1998); doi: 10.1117/12.323624

Event: International Symposium on Multispectral Image Processing, 1998, Wuhan, China

New Puzzle Assembly

Min Gyo Chung[a], *Margaret M. Fleck*[b], and *David A. Forsyth*[c]

[a]Multimedia Technology Research Lab., Korea Telecom, Seoul 137-792, S. Korea

[b]Dept. of Computer Science, Harvey Mudd College, Claremont, CA 91711, USA

[c]Dept. of EECS, University of California, Berkeley, CA 94720, USA

ABSTRACT

This paper describes a new jigsaw puzzle solver using chromatic information as well as geometric shape. Three new puzzle assembly algorithms are developed and experimental results on their performance are provided.

Keywords: Partial boundary matching, Color, Puzzle assembly, Shape

1. INTRODUCTION

Much work[1–6] has been done on solving the jigsaw puzzle problem since the early 1960's. The 2D puzzle problem, an instance of the general *partial boundary matching* problem,[7] has long been intriguing, since it contains many popular problems encountered in image processing applications and has a great potential for interdisciplinary research such as molecular docking[8,9] and computer assisted anthropology system.[10]

Earlier approaches are principally concerned with the geometric shape information of puzzle pieces, disregarding other useful information such as colors, surface markings, or textures. Motivated by this fact, this paper presents a new jigsaw puzzle solver, which is equipped with three new assembly algorithms and uses information on both color and shape.

2. NEW METHOD

It is crucial to accurately detect the outer boundary of a puzzle piece. Assuming that each image contains a puzzle piece on a uniformly colored background, the piece's outer boundary is detected in such a way that the image is divided into two disjoint regions(i.e., background and piece) using *histogram thresholding*, the binary segmented image is smoothed by a morphological operator, and it is lastly fed to the Fleck's[11] upgraded version of Canny edge finder.

2.1. Boundary Matching

Basically, our boundary matching scheme uses the notion of canonical frame.[6] Given the closed boundary of a puzzle piece represented in canonical frame, consider a boundary segment between two corners. For the segment, a mapping f called *canonical distance function* is defined to be $f(i) = d_i$, where d_i is the signed distance from the ith sampling point on the boundary segment to the horizontal axis connecting two corners

Suppose that $f_s(i)$ and $f_t(i)$ are canonical distance functions for two boundary segments s(slot curve, m sampling points) and t(tab curve, n sampling points). Assuming without loss of generality that $m \geq n$, for $e = 0, \ldots,$ m-n, we define

$$Cost^e_{boundary}(s,t) = \sqrt{\frac{\sum_{i=1}^{n} |f_s(i+e) - f_t(i)|^2}{n}}$$

The boundary matching cost between s and t is then expressed as

$$Cost_{boundary}(s,t) = \text{Average}_{e \in \{0,1,\ldots,m-n\}}[Cost^e_{boundary}(s,t)]$$

2.2. Color Matching

Assuming e is an integer in $[0, m-n]$, let $< W^s_{1+e}, \ldots, W^s_{n+e} >$ be a sequence of n cicular sampling window at regular intervals along s and $< W^t_1, \ldots, W^t_n >$ a sequence of n cicular sampling windows along t. For each pair of corresponding windows W^s_{i+e} and W^t_i, compute a quantity $u_{i,e}$

$$u_{i,e} = |q^s_{i,e} - q^t_i| + \sum_{j=1}^{p} |h^{s,i,e}_j - h^{t,i}_j|$$

where $q^s_{i,e}(q^t_i)$ is a median at window $W^s_{i+e}(W^t_i)$, $H^{s,i,e} = \{h^{s,i,e}_1, \ldots, h^{s,i,e}_p\}(H^{t,i} = \{h^{t,i}_1, \ldots, h^{t,i}_p\})$ is a feature histogram for window $W^s_{i+e}(W^t_i)$. The second term measures how much the two histograms $H^{s,i,e}$ and $H^{t,i}$ differ.

Assuming that T_{high} and T_{low} are two thresholds with $T_{high} > T_{low}$, $u_{i,e}$ is said to be abnormal if (a)$u_{i,e} > T_{high}$ and (b)$u_{i-1,e} < T_{low}$ and $u_{i+1,e} < T_{low}$. Let

$$Cost^e_{color}(s,t) = \sqrt{\frac{\sum_{i=1}^{n'} [u^{norm}_{i,e}]^2}{n'}}$$

where $u^{norm}_{i,e}$ denotes a normal point and n' is the total number of normal points. The color matching cost between s and t is then

$$Cost_{color}(s,t) = \text{Average}_{e \in \{0,1,\ldots,m-n\}} [Cost^e_{color}(s,t)]$$

2.3. Assembling Puzzle Pieces

2.3.1. Assignment Problem based Method

Let $C = (c_{i,j})$ be a $k \times k$ cost matrix, where $c_{i,j}$, total cost between ith slot and jth tab, derives from a weighted sum of shape and color costs. Assuming that $N = \{1, 2, \cdots, k\}$, assignment problem(AP)[12] refers to finding a permutation $\sigma \in \Phi_N$ with the smallest cost

$$\min_{\sigma \in \Phi_N} \sum_{i \in N} c_{i,\sigma(i)}$$

where Φ_N denotes the set of all permutations of the set N.

The first solution to puzzle assembly is referred to as *AP-based method* and its basic idea is as follows. Suppose assignment problem takes the cost matrix C as input and produces a permutation σ of $\{1,2, \ldots, k\}$ such that $\sum_{i=1}^{k} c_{i,\sigma(i)}$ is minimized. A pair $(i, \sigma(i))$ means that ith slot matches with $\sigma(i)$th tab. If the puzzle assembly with those pairs is verified to lead to a successful reconstruction, σ is then the solution to the underlying jigsaw puzzle problem and the whole process terminates. However, if the reconstruction ends up in a failure, another iteration is performed to find the next permutation σ' such that $\sum_{i=1}^{k} c_{i,\sigma'(i)}$ is the second best. This process is repeated until a true solution is found.

The crucial component of AP-based method is the way of systematically enumerating permutations by increasing order of cost. The algorithm by Murty[13] is adopted for this purpose. While in assembling pieces one by one according to a permutation, the verification phase in AP-based method checks whether an undesirable situation takes place. Figure 1 demonstrates some undesirable situations: (a) new puzzle piece P introduced by fitting together e and m has already been used in another place; (b) e_1 matches m_1, e_2 matches m_2, but m_1 and m_2 belong to different pieces; (c) the matching of e and m results in m_1 (straight edge) and m'_1 (straight edge) being joined together.

2.3.2. Traveling Salesman Problem and Assignment Problem based Method

A puzzle piece belongs to one of two types: frame piece and interior piece. Frame pieces, constituting the outer frame of the completely assembled picture, have at least one straight edge. Interior pieces have no straight edge and fill the inside of the picture. Taking advantage of this observation, the second puzzle assembly algorithm consists of two parts: one part is to assemble frame pieces and the other part is to assemble interior puzzle pieces. Frame assembly is performed using traveling salesman problem(TSP)[14] and interior assembly is done with assignment problem. Thus we call this approach *TSP&AP-based method.*

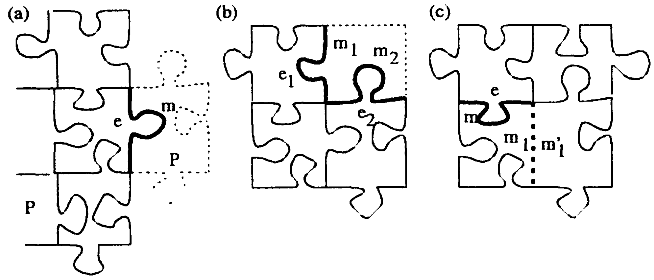

Figure 1. Some undesirable situations.

A Hamilton circuit of a graph is a circuit passing through every vertex of the graph exactly once. Traveling salesman problem is to find in the graph a Hamilton circuit having the least cost. The assembly of frame pieces can be formulated as follows with traveling salesman problem. Let $C^{TSP} = (c^{TSP}_{i,j})$ be the matrix taken as an input to the traveling salesman problem, where each entry $c^{TSP}_{i,j}$ specifies the matching cost between two frame pieces i and j. Note that C^{TSP} is not symmetric. A tour $\tau =< p_1, \ldots, p_{k'} >$ means that k' frame pieces$(p_1, \ldots, p_{k'})$ are arranged in this order. TSP&AP-based methods generates the best tour τ to traveling salesman problem with C^{TSP} and checks if the frame pieces arranged according to τ really form a close loop. If a closed loop is formed, then τ is the solution to the assembly of frame pieces. Otherwise, the second best tour τ' is produced and the similar procedure is repeated until the solution is found. The algorithm of ranking the tours in increasing order of cost is a slight modification to Murty's algorithm and its full description appears in Refs. 15.

After TSP&AP based method finishes assembling all frame pieces, it continues to assemble interior puzzle pieces. The assembly of inner pieces can be achieved exactly in the same way as AP-based method.

2.3.3. Traveling Salesman Problem and Kbest based Method

The last approach to puzzle assembly is *TSP&Kbest-based method.* Like TSP&AP-based method, the TSP&Kbest-based method is composed of frame assembly and interior assembly. The frame assembly of TSP&Kbest-based method is accomplished in the same way as frame pieces are assembled in TSP&AP-based method. However, the inner assembly adopts *Kbest* algorithm.[5]

Basically, Kbest assembles the first row of puzzle interior, starting with the first element of the row and advancing to the right. After finishing the first row, it continues to assemble the second row, third row, ... until the last row is assembled. It generates K^2 partial solutions at a stage of assembly, but passes only K best partial solutions to the next stage.

2.4. Displaying Pieces

In order to graphically display a completely assembled picture of puzzle pieces, it is necessary to provide a means to translate and rotate pieces. Translation and rotation can be carried out by a 3 × 3 homogeneous transformation matrix, which is easily computed if three pairs of source and destination points are known. However, we use more than 3 points to obtain the transformation matrix more accurately. The algorithm to compute the transformation matrix iteratively seeks the best affine parameters in the first and second rows of the matrix by a least-square method, and then the best bottom two parameters in the third row by Newton's method.

3. EXPERIMENTAL RESULTS

The NEC color camera equipped with a Computar 6.0 mm lens is right above a puzzle piece and fixed at a same height. The camera interface uses the camera calibration algorithm developed by Stevenson and Fleck[16] to remove perspective image distortions. Each image(213 × 293) contains one puzzle piece and is represented by the 8-bit log-opponent color model. Given $L(x) = k \log_{10}(x + 1)$, where x is a raw RGB value and k is a constant to scale the log values into the range of 8-bit values, three log-opponent values are $Intensity = L(G)$, $Rg = L(R) - L(G)$, and $By = L(B) - \frac{L(R)+L(G)}{2}$.

(a)

Puzzle *panther*						
Type	Feature	Assembly Algorithms				
		AP	TSP&AP		TSP&K	
			Frame	Interior	Interior	
Shape		1000^+	24	1000^+	3	
Color	Hue	1000^+	2	1000^+	12	
	Sat.	1000^+	1000^+	1000^+	500^+	
	Hue&Sat.	1000^+	2	1000^+	16	
	Rg&By	1000^+	13	1000^+	190	
	RGB	1000^+	948	1000^+	500^+	

(b)

Puzzle *panther*					
Common	Feature	Assembly Algorithms			
		AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape+	Hue	2	1	1	1
	Sat.	1000^+	4	567	2
	Hue&Sat.	1	1	1	1
	Rg&By	3	1	3	2
	RGB	1000^+	2	1000^+	2

Table 1. Assembly outcome of puzzle *panther*. (a) When using shape or color information separately; (b) When integrating color into shape information.

(a)

Puzzle *car*					
Type	Feature	Assembly Algorithms			
		AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape		1000^+	1000^+	1000^+	500^+
Color	Hue	1000^+	13	1000^+	96
	Sat.	1000^+	3	1000^+	35
	Hue&Sat.	1000^+	1	1000^+	4
	Rg&By	1000^+	1	1000^+	3
	RGB	1000^+	1	1000^+	3

(b)

Puzzle *car*					
Common	Feature	Assembly Algorithms			
		AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape+	Hue	1000^+	4	1000^+	1
	Sat.	1000^+	1	1000^+	1
	Hue&Sat.	396	1	396	1
	Rg&By	25	1	25	1
	RGB	38	1	36	1

Table 2. Assembly outcome of puzzle *car*. (a) When using shape or color information separately; (b) When integrating color into shape information.

We ran the proposed algorithms over the following six color puzzles available on the market: *panther* (54 pieces); *car* (54 pieces); *boy* (25 pieces); *barney* (25 pieces); *elephant* (20 pieces); *bear* (12 pieces). Table 1(a) shows the results of executing three assembly algorithms over puzzle *panther* when applying shape or color information separately. Table 1(b) supplies the results of executing three assembly algorithms when combining the boundary information and the color information. The values under TSP&Kbest's *Interior* column are the magnitude of the number K in Kbest algorithm. In other columns, the value in a cell means the number of iterations until the solution is obtained. The plus sign(+) on a number signifies that the algorithm had not found the solution even after the specified number of iterations was reached.

As Table 1(b) demonstrates, when chromatic information is incorporated, the results improve much better, compared with the results obtained through only the shape information. This observation implies that color information significantly aids in solving the puzzle assembly problem. In terms of how rapidly each assembly algorithm reaches a solution, the TSP&Kbest-based method is the best, followed by TSP&AP-based method, and followed by AP-based method.

Similar assembly outcomes from other puzzles are shown in Table 2, 3, 4, 5, and 6. Note that as we move down to the smaller size puzzles from large puzzles, the gap in performance among three assembly algorithms gets smaller. The completely reconstructed pictures for all 6 puzzles are provided in Figure 2.

The assembly program was written in Common Lisp and was tested on IBM RS/6000 workstation. Some sampling timing numbers will be provided to give an idea of how long things take. Table 7 shows the time (in unit of second) which three assembly algorithms take on the puzzle *panther*, *barney*, and *bear*, when the color information is added into the shape information. The unknown value in a table cell corresponds to 1000^+.

The comparison of the time for one iteration of TSP and the time for one iteration of AP is given in Table 8. As the number of pieces in a puzzle decreases, the time for one iteration of TSP or AP also decreases.

(a)

Puzzle *boy*					
		Assembly Algorithms			
Type	Feature	AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape		27	1	20	3
Color	Hue	159	1	159	26
	Sat.	1000^+	58	1000^+	2
	Hue&Sat.	105	1	97	4
	Rg&By	713	1	19	1
	RGB	1000^+	5	1000^+	4

(b)

Puzzle *boy*					
		Assembly Algorithms			
Common	Feature	AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape+	Hue	1	1	1	1
	Sat.	1	1	1	1
	Hue&Sat.	1	1	1	1
	Rg&By	1	1	1	1
	RGB	145	1	27	1

Table 3. Assembly outcome of puzzle *boy*. (a) When using shape or color information separately; (b) When integrating color into shape information.

(a)

Puzzle *barney*					
		Assembly Algorithms			
Type	Feature	AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape		1000^+	2	1000^+	49
Color	Hue	1	1	1	1
	Sat.	1000^+	3	1000^+	1
	Hue&Sat.	1	1	1	1
	Rg&By	3	1	3	1
	RGB	2	1	2	1

(b)

Puzzle *barney*					
		Assembly Algorithms			
Common	Feature	AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape+	Hue	1	1	1	1
	Sat.	44	1	26	1
	Hue&Sat.	1	1	1	1
	Rg&By	3	1	3	1
	RGB	1	1	1	1

Table 4. Assembly outcome of puzzle *barney*. (a) When using shape or color information separately; (b) When integrating color into shape information.

(a)

Puzzle *elephant*					
		Assembly Algorithms			
Type	Feature	AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape		1000^+	39	1000^+	500^+
Color	Hue	1	1	1	1
	Sat.	1	1	1	1
	Hue&Sat.	1	1	1	1
	Rg&By	1	1	1	1
	RGB	1	1	1	1

(b)

Puzzle *elephant*					
		Assembly Algorithms			
Common	Feature	AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape+	Hue	1	1	1	1
	Sat.	1	1	1	1
	Hue&Sat.	1	1	1	1
	Rg&By	1	1	1	1
	RGB	1	1	1	1

Table 5. Assembly outcome of puzzle *elephant*. (a) When using shape or color information separately; (b) When integrating color into shape information.

(a)

Puzzle *bear*						
Type	Feature	Assembly Algorithms				
		AP	TSP&AP		TSP&K	
			Frame	Interior	Interior	
Shape		1	1	1	1	
Color	Hue	1	1	1	1	
	Sat.	1	1	1	1	
	Hue&Sat.	1	1	1	1	
	Rg&By	1	1	1	1	
	RGB	1	1	1	1	

(b)

Puzzle *bear*					
Common	Feature	Assembly Algorithms			
		AP	TSP&AP		TSP&K
			Frame	Interior	Interior
Shape+	Hue	1	1	1	1
	Sat.	1	1	1	1
	Hue&Sat.	1	1	1	1
	Rg&By	1	1	1	1
	RGB	1	1	1	1

Table 6. Assembly outcome of puzzle *bear*. (a) When using shape or color information separately; (b) When integrating color into shape information.

Puzzle	Feature (Shape+)	Assembly Algorithms			
		AP	TSP&AP		TSP&Kbest
			Frame	Interior	Interior
Panther	Hue	55	3	2	1
	Saturation	(Unknown)	65	3580	2
	Hue&Saturation	5	3	2	1
	Rg&By	75	1	25	2
	RGB	(Unknown)	20	(Unknown)	2
Barney	Hue	1	1	1	1
	Saturation	115	1	25	1
	Hue&Saturation	1	1	1	1
	Rg&By	7	1	3	1
	RGB	1	1	1	1
Bear	Hue	0.5	0.5	0.5	0.5
	Saturation	0.5	0.5	0.5	0.5
	Hue&Saturation	0.5	0.5	0.5	0.5
	Rg&By	0.5	0.5	0.5	0.5
	RGB	0.5	0.5	0.5	0.5

Table 7. Times taken for puzzle assembly

Puzzles	Average Time per AP Iteration in AP Algorithm	Average Time per TSP Iteration in TSP&AP Algorithm	Average Time per AP Iteration in TSP&AP Algorithm
Panther	19	7	4
Barney	2	1	1
Bear	0.5	0.5	0.5

Table 8. Times taken for one iteration of TSP and AP

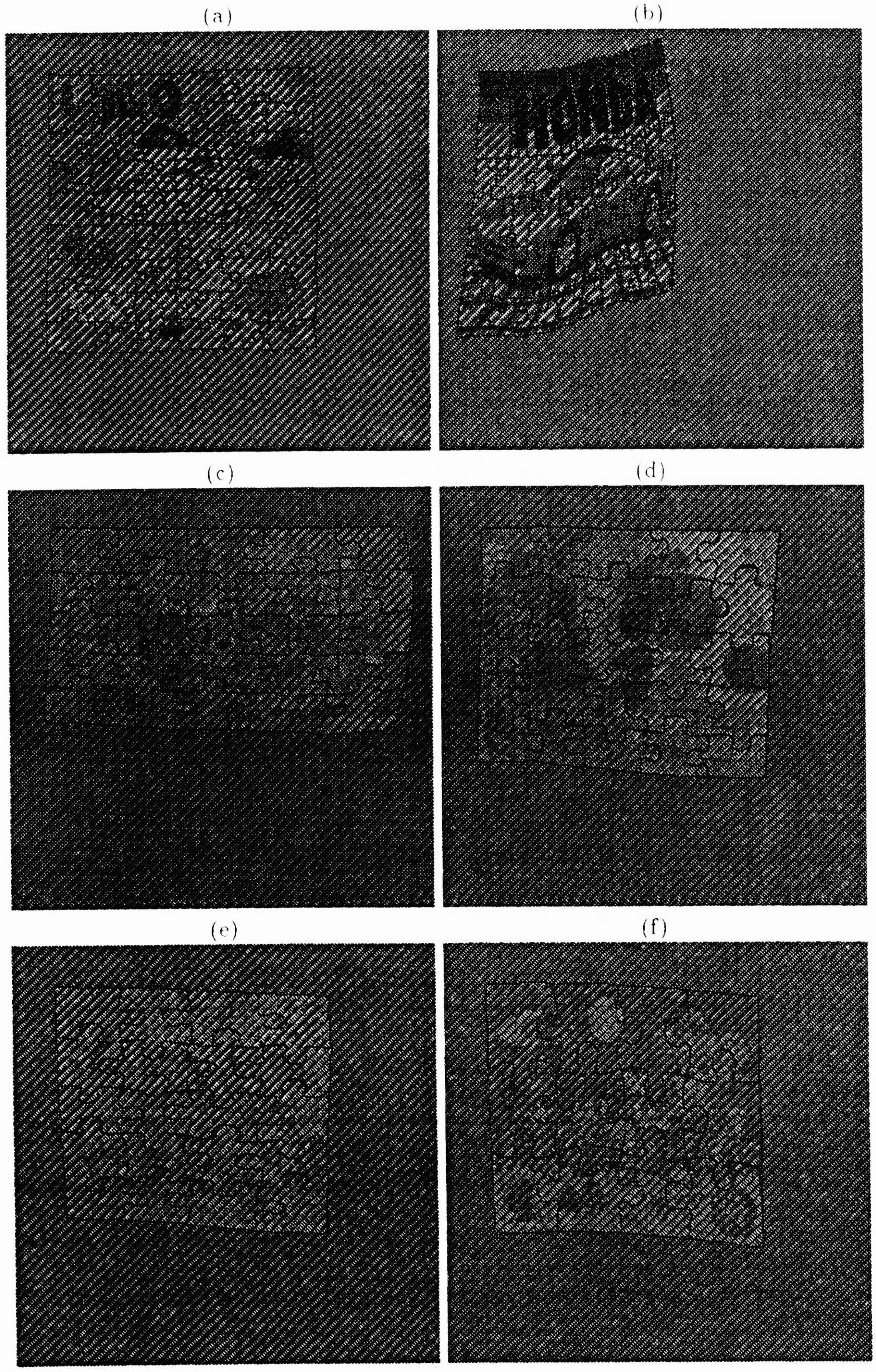

Figure 2. Assembled pictures. (a) puzzle *panther*; (b) puzzle *car*; (c) puzzle *boy*. (d) puzzle *barney*; (e) puzzle *elephant*; (f) puzzle *bear*.

4. CONCLUSION

This paper proposed a new puzzle solver which uses chromatic information as well as geometric shape and is powered by three new assembly algorithms. As the experimental results demonstrate, it is clear that many jigsaw puzzle problems which are hard to solve with only shape information can be easily solved when color information is integrated into the shape information. We have also observed significant performance difference among three assembly algorihtms. The gap in their performance gets wider as the number of pieces in a puzzle gets greater. In terms of performance, TSP&Kbest-based algorithm is the best, TSP&AP-based algorithm is the second, and the AP-based algorithm is the last.

This work might be extended in two directions. One would be to make the puzzle boundaries richer by including other valuable cues such as surface markings, edge segments, or textures. The richer structures could eliminate ambiguous multiple matches more effectively. The other direction would be to increase the dimensionality of the jigsaw puzzle problem, i.e., to solve 3D puzzles. The algorithms developed for 3D puzzle problem could be practically applied to such applications as repair of broken objects, restoration of archaeological findings, and molecular docking problem. Especially, molecular docking has been used in studies of structure-based drug design.

REFERENCES

1. H. Freeman and L. Gardner, "Apictorial jigsaw puzzles: the computer solution of a problem in pattern recognition," *IEEE Transactions on Electronic Computers* **13**, pp. 118–127, Apr. 1964.
2. G. M. Radack and N. I. Badler, "Jigsaw puzzle matching using a boundary-centered polar encoding," *Computer Graphics and Image Processing* **19**, pp. 1–17, 1982.
3. T. Altman, "Solving the jigsaw puzzle problem in linear time," *Applied Artificial Intelligence* **3**, pp. 453–462, 1989.
4. R. W. Webster, P. S. LaFollette, and R. L. Stafford, "Isthmus critical points for solving jigsaw puzzles in computer vision," *IEEE Transactions on Systems, Man and Cybernetics* **21**, pp. 1271–1278, 1991.
5. H. Wolfson, E. Schonberg, A. Kalvin, and Y. Lamdan, "Solving jigsaw puzzles by computer," *Annals of Operations Research* **12**, pp. 51–64, 1988.
6. C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy, "Canonical frames for planar object recognition," in *Proc. 2nd European Conference on Computer Vision*, pp. 757–772, 1992.
7. H. Freeman, "Shape description via the use of critical points," *Pattern Recognition* **10**, pp. 159–166, 1978.
8. F. S. Kuhl, G. M. Crippen, and D. K. Friesen, "A combinatorial algorithm for calculating ligand binding," *Journal of Computational Chemistry* **5**(1), pp. 24–34, 1984.
9. E. C. Meng and I. D. Kuntz, "Molecular docking: a tool for ligand discovery and design," *Drug Information Journal* **28**, pp. 735–749, 1994.
10. A. D. Kalvin, D. Dean, and J.-J. Hublin, "Reconstruction of human fossils," *IEEE Computer Graphics and Applications* **15**, pp. 12–15, Jan. 1995.
11. M. M. Fleck, "Some defects in finite difference edge finders," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, pp. 337–345, Mar. 1992.
12. H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly* **2**, pp. 83–97, 1955.
13. K. G. Murty, "An algorithm for ranking all the assignments in increasing order of cost," *Operations Research* **16**, pp. 682–687, 1968.
14. M. Bellmore and J. C. Malone, "Pathology of traveling salesman subtour-elimination algorithms," *Operations Research* **19**, pp. 278–307, 1971.
15. M. G. Chung, *Integration of color into partial boundary matching: via solving pictorial jigsaw puzzle.* PhD thesis, University of Iowa, Dec. 1996.
16. D. E. Stevenson and M. M. Fleck, "Robot aerobics: four easy steps to a more flexible calibration," in *Proc. International Conference of Computer Vision*, pp. 34–39, 1995.