International Conference
20th EURO Mini Conference
"Continuous Optimization and Knowledge-Based Technologies"
(EurOPT-2008)
May 20–23, 2008, Neringa, LITHUANIA

ISBN 978-9955-28-283-9
L. Sakalauskas, G.W. Weber and
E. K. Zavadskas (Eds.): EUROPT-2008
Selected papers. Vilnius, 2008, pp. 18–24

OPTIMIZATION FOR AUTOMATED ASSEMBLY OF PUZZLES

Mahmut Şamil Sağıroğlu, Aytül Erçil

Sabancı University, Faculty of Engineering and Natural Sciences,
Tuzla, Istanbul TURKEY. E-mail: aytulercil@sabanciuniv.edu

Abstract: The puzzle assembly problem has many application areas such as restoration and reconstruction of archeological findings, repairing of broken objects, solving jigsaw type puzzles, molecular docking problem, etc. The puzzle pieces usually include not only geometrical shape information but also visual information such as texture, color, and continuity of lines. This paper presents a new approach to the puzzle assembly problem that is based on using textural features and geometrical constraints. The texture of a band outside the border of pieces is predicted by inpainting and texture synthesis methods. Feature values are derived from these original and predicted images of pieces. An affinity measure of corresponding pieces is defined and alignment of the puzzle pieces is formulated as an optimization problem where the optimum assembly of the pieces is achieved by maximizing the total affinity measure. Experimental results are presented on real and artificial data sets.

Keywords: puzzle assembly, inpainting, texture synthesis, optimum assembly, reconstruction of broken pieces.

1. Introduction

The aim of this paper is to develop a method for the automated assembly of broken objects that have surface texture, from their pieces. The task of reassembling has great importance in the fields of anthropology, failure analysis, forensics, art restoration and reconstructive surgery. It also appears heavily in archaeology. The fact that performing reconstruction of archaeological objects from fragments manually is very time consuming motivates automatic techniques for reassembly of fragments. In general, reconstruction of objects can be regarded as a puzzle-solving problem, which contains many problems endemic to pattern recognition, computer vision, feature extraction, boundary matching, and optimization fields.

In classical jigsaw puzzles, the essentials of assembly depend on the alignments of object edges (e.g. picture of a house), the similarity of colors (e.g. cloud drawing) and continuity of textural properties (e.g. grass of a garden) for the adjacent pieces. The solution approach has to consider all these situations to match images of adjacent pieces.

Previous works on the assembly problem have focused mainly on geometrical properties of the pieces.

The puzzle pieces are represented by their boundary curves. Some approaches especially related to standard toy-store jigsaw puzzle solver use feature based matching methods. The problem of jigsaw puzzle solving is a reduced and restricted version of the general assembly problem. Its computerized solution was first introduced by Freman [3], who successfully solved a 9-piece jigsaw puzzle. Other works [1, 5, 8] also use feature based matching approaches. These methods are respectively fast so that they manage to assembly even if the number of puzzle pieces becomes large. The main drawback of this approach is that they cannot provide detailed matching of boundaries and overlapping regions. Research involving classical jigsaw puzzle has so far ignored texture or color information to the assembly problem. There are a few approaches, which use only the color values of pixels on the boundary contour [1].

More general partial curve matching algorithms that solve the global 2D and 3D assembly problems based on geometrical properties were presented in [7, 11, 17]. The problem of 3D curves is addressed by [15]. The accuracy of the matching technique depends on perfect extraction of the trace of a curve and the computation of curvature and torsion. It is potentially a non-robust process and has only been tested on artificial data. Another research [14] matches 2D and 3D break curves by combining a coarse-scale representation of curves and refine iteratively via a fine-scale elastic matching. The works that achieved global assembly of pieces based on curve matching have not attempted to combine the geometrical methods with textural information.

There is great scientific interest in the archaeological community in reconstructing objects from fragments. An automatic tool that assists archeologists in reconstructing monuments or smaller fragments would lead to avoiding unnecessary manual experimentation with fragile and often heavy fragments, and reduce the assembly time. Currently, the Digital Michelangelo team is tackling the problem of assembling the Forma Urbis Romae [9]. It is a marble map of ancient Rome that has more than a thousand fragments. Their investigation is based on broken surface border curves, possibly texture patterns, and additional features of the fragments. The University of Athens has developed "The Virtual Archaeologist" [10] system, relying on the broken surface morphology to determine correct matches between fragments. This method detects candidate fractured faces, matches fragments one by one and assembles fragments into complete or partially complete entities. The Shape Lab at Brown University presents an approach to automatic estimation of mathematical models of axially symmetric pots made on a wheel [16, 17]. This technique is based on matching break curves, estimated axis and profile curves, a number of features of groups of break-curves. Finally, the assembly problem is solved by maximum likelihood performance-based search. At the Technical University of Vienna, a fully automated approach to pottery reconstruction based on the fragments profile, is given [6]. Fornasier and Toniolo have developed a pattern matching algorithm for comparison of digital images by discrete Circular Harmonic expansions based on sampling theory. The assumption for this method is that the photographs of the original puzzle exist. [4]

Neglecting continuity of color and texture for adjacent fragments is a *waste* of valuable information for many cases. The pictorial information on a fragment consists of various components, and different specifications of surface image of pieces are dominant according to implementation field. In the archeological field, the pictorial features may include highly directional marble veining, the pattern of surface incisions, paintings on the outer and inner surfaces, carvings and horizontal circles due to finger smoothing while the pot is spinning on the wheel.

In archaeology, erosion, impact damages or undesired events cause fragments to vanish or deteriorate, such as in the case of Forma Urbis Romae. This reality increases the necessity of pictorial information to solve the reconstruction of all types of puzzles, because the geometrical approaches relying on exact matching of break curves are not applicable to the assembly of the pieces, if the border of fragments have disappeared. The texture prediction method can manage to estimate possible adjacent fragments, even if there is a gap caused by erosion between two neighbouring pieces.

In this paper, we design a texture prediction algorithm, which predicts the pixel values in a band outside the border of the pieces. Features obtained from the predicted texture outside a piece are correlated with original pictorial specifications of possible neighboring pairs. Also, a confidence measure depending on texture patterns is defined. Then, we define an affinity measure of corresponding pieces that utilizes all kinds of image information, such as continuity of edges, textural patterns, and color similarities. The puzzle solving problem is then reformulated as an optimization problem where the problem of finding matching pieces is stated as finding pieces that maximize the overall affinity function.

The rest of this paper is organized as follows: Section 2 outlines the method used in solving the assembly problem, Section 3 presents image inpainting and texture synthesis methods that are used in predicting the expanded part of the pieces. The affinity measure used in the assembly process is explained in Section 4. Experimental results are given in Section 5.

2. Automated puzzle assembly method

Our proposed approach is based on defining a fast and robust method that finds the best transformation of pieces that maximizes matching and continuity of textures of fragments while the geometrical constraints are satisfied. After the acquisition and preprocessing of the data, the first step is the prediction of the pixel values in a band around the border of the piece; this step is applied to all pieces separately. The prediction algorithm automatically fills in this extension region using information in the central part. The main idea in extending the picture/texture on the fragment outwards is that the correlation between the features of the predicted region and its true neighboring piece is significantly higher than alternative pairings. We use the mixture of inpainting and texture synthesis methods for prediction. Image inpainting is the process of filling in missing data in a designated part of an image or a video from the surrounding area, and texture synthesis is to create a new image with the same seed texture but of different shape to a sample region. While expanding the fragment image, we introduce the confidence of expansion as a new parameter in the prediction phase of the assembly problem. This parameter represents the reliability of expanded values and is used by later processes. The confidence depends on the structure of the texture such as the continuity of edges, the roughness of texture and the distance to the border of original fragment.

We then derive feature values in both the original fragment and the extended region. The proposed approach does not bound the number of features nor it restrict the type of image features. Any textural feature believed to improve the success of assembly can be easily inserted into the process. A combination of the feature and confidence values is used to generate an affinity measure of corresponding pieces. The matching of pieces and achievement of the assembly is established by optimizing this affinity measure.

3. Inpainting and texture synthesis for expanding the pieces

As mentioned in section 2, the first step in the assembly process is the expansion of each piece in a band around the border of the piece by predicting the pictorial information on the surface outwards. Inpainting and texture synthesis are two techniques that will be used to carry out this task. Image inpainting refers to the process of filling-in the missing areas or changing an image in a non-noticeable way by an observer. The problem of texture synthesis is to fill large image regions with a sample texture. In this paper, we use the approach used by Criminisi [16] to predict the pixel values in a band around the border of the piece.

The source region, I_m^0, is the acquired image of the mth piece. A target band, I_m^+, outwards from the mth piece is defined (so $I_m = I_m^0 + I_m^+$). This target band represents the extension region of the mth piece. The border between I_m^0 and I_m^+ is indicated by δI_m. This border evolves outward as the inpainting algorithm progresses. The inpainting algorithm consists of three main steps. These steps are iterated until the whole target region or band has been filled. The first step is to compute the priority, P, which determines the order in which they are filled. Priority value is computed for the patches Ψ_p centered at the point p for $p \in \delta I_m$. Conceptually, the priority depends on continuation of strong edges, D, and confidence of neighbor pixels, C:

$$P(p) = D(p).C(p) \quad (1) \qquad C(p) = \frac{\sum_{q \in \Psi_p \cap I_m^0} C(q)}{\left|\Psi_p\right|}, \quad D(p) = \left|\nabla I_p{}^{\perp}.n_p\right| \quad (2)$$

where $|\Psi_p|$ is the area of Ψ_p, n_p is unit vector orthogonal to the front δI_m at the point p and $\perp$ indicates the orthogonal operator. This confidence value reflects the reliability of a region or a pixel, and it affects the filling order during inpainting process. Initially, we set C=1 (%100 reliability) to pixels in the original piece, and assign C=0 to the pixels in the target region to be filled. The Data term D(p) is a function of the strength of isophotes hitting the front δI_m. This term increases the priority if an isophote flows into that patch which is important for the assembly process since it causes the linear structures to be synthesized or filled first. Therefore, the linear structures orthogonal to border of pieces are completed earlier and these points or patches get higher confidence values.

When all priorities have been computed, the highest priority, p', is determined. The second step of the Where $|\Psi_p|$ is the area of Ψ_p, n_p is unit vector orthogonal to the front δI_m at the point p and $\perp$ indicates the orthogonal operator. This confidence value reflects the reliability of a region or a pixel, and it affects the filling order during inpainting process. Initially, we set C=1 (%100 reliability) to pixels in the original piece, and assign C=0 to the pixels in the target region to be filled. The Data term D(p) is a function of the strength of isophotes hitting the front δI_m. This term increases the priority if an isophote flows into that patch which is important for the assembly process since it causes the linear structures to be synthesized or filled first. Therefore, the linear structures orthogonal to border of pieces are completed earlier and these points or patches get higher confidence values.

When all priorities have been computed, the highest priority, p', is determined. The second step of the prediction process is propagating the texture and structure information into the target band. The color information is propagated via diffusion in classical inpainting techniques. In our work, as in [16], propagation of the image texture occurs by direct sampling of source region. The most similar patch for sampling is given as:

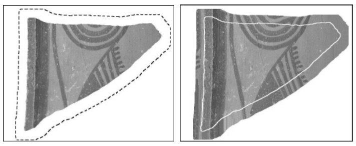
$$\Psi_{q'} = \underset{\Psi_q \in I_m^0}{\text{argmax}}\ d(\Psi_{p'}, \Psi_q) \qquad (3)$$

where $d(\Psi_{p'},\Psi_q)$ is the distance between the already filled pixels of patches at the points $p^{'}$ and q. The patch at the point $q^{'}$ is the most similar one and the values of each pixel to be filled in the $p^{'}$ patch are copied directly from the patch in the $q^{'}$ point.

The last step for iterations is to update the confidence values. After the patch $\Psi_{p'}$ has been filled with new values, the confidence values affected by the filling of the new patch are updated. This region is limited by the neighbors of the point $p^{'}$.

$$C(p) = C(p') \quad \forall p \in \psi_{p'} \cap I_m^{+} \tag{4}$$

As the filling proceeds, the confidence values decrease as the pixels in the predicted region get farther from the original boundary. This indicates that the color values of pixels far from border are less reliable than closer ones.



(a) (b)

Fig. 1. (a) An archeological sherd to be expanded (b) The expanded piece

4. An Affinity measure for compatibility of pieces

While matching or calculating similarity of possible two neighboring pieces, pixel-by-pixel comparison of two pieces is not meaningful. Thus, image features, f_{ki}, the k^{th} feature values of the i^{th} piece, are extracted from the source and target regions for each piece after predicting the target band. Selection of the features depends on the structure of the image. Currently, only first and second moments (mean and variance) are used in the experiments. In the case of using suitable texture features, serious improvements can be obtained. The features are calculated in a window whose size depends on the resolution of the pictures on the pieces. The next step is the computation of confidence values for the features. When a feature value is extracted by using the pixels in a window, the confidence of this feature for a point depends on the confidences of all pixels in the window. Mean of all confidence of pixels in the window is assigned as confidence of the feature, Ck'.

Let $D_k(f_{ki}, f_{kj})$ be the distance function between the k^{th} feature values of i and j pieces. $T_i=(\Delta x_i,\Delta y_i,\Delta \square_i)$ denotes the transform of the i^{th} piece and $T_i\ (f_{ki})$ denotes transform of the kth feature extracted from the i^{th} piece. For the simplicity of expressions, the $(\Delta x_i,\Delta y_i,\Delta \square_i)$ parameter for each variable will not be shown.

In the current experiments, the Euclidean distance is used for all features. Let's define a threshold value Th_k for the k^{th} feature distance and a similarity measure S_k:

$$S_k = D_k(f_{ki}, f_{kj}) - Th_k \tag{4.1}$$

We set the threshold, Th_k., so that the more similar the feature values are, the larger negative value the similarity measure, S_k will take or visa versa.

$$\sum_{k}^{n_k} S_k = \sum_{k}^{n_k} \left[D_k(f_{ki}, f_{kj}) - Th_k \right] \tag{4.2}$$

where n_k is the number of features. (4.2) gives the total similarity between i and j pieces. We can normalize the above equation by dividing all S_k into Th_k .

$$\sum_{k}^{n_k} (S_k / Th_k) = \sum_{k}^{n_k} w_k \left[D_k(f_{ki}, f_{kj}) - 1 \right] \tag{4.3}$$

where w_k are the weight values for the k^{th} feature and are inversely proportional to Th_k. If we add all the similarity measures for all the features, we obtain

$$\sum_{\substack{j \\ i \neq j}}^{n_p} \left[\sum_{k}^{n_k} w_k D_k(f_{ki}, f_{kj}) - 1 \right] C_j^{'} \tag{4.4}$$

where n_p is the number of pieces in the puzzle. Expression (4.4) denotes that the total similarity between the i^{th} and j^{th} pieces. The $\sum S_k$ are weighted according to j^{th} confidence values since the total similarity should be affected when confidence of a point is small (close to zero), even if two pieces are similar. It is also valid that the cost or affinity function should be more sensitive to the texture distance if the confidence is high. After weighting the similarities, summation for all j pieces where i is different than j shows how much the i^{th} piece fits the other pieces. If we sum the similarities for all possible pairs, we obtain:

$$m_1(x,y) = \frac{\sum_{i}^{n_p} \sum_{j=i+1}^{n_p} \left[w_k D_k(f_{ki}, f_{kj}) - 1 \right] C'_j C'_i}{\sum_{i}^{n_p} C'_i} \tag{4.5}$$

This is the first part of Cost or Affinity function and is derived from the weighted mean of (4.4). This value goes towards negative if there exists a good harmony between images of pieces.

The second part of general F_{cost} function is for embedding the geometrical constraints to Cost or Affinity. In reality, two pieces cannot overlap on any point. In order to prevent this situation, a sufficiently large, w_c, weight or constant is added to the Cost function for the overlapping points.

$$m_2(x,y) = \sum_{i}^{n_p} \sum_{j=i+1}^{n_p} L(T_i(C_i))L(T_j(C_j)) \quad (4.7) \quad \text{where} \quad L(x) = \begin{Bmatrix} 0 & \text{if x} \neq 0 \\ 1 & \text{if x} = 0 \end{Bmatrix} \quad (4.8)$$

The confidence values are used to formulize overlapping operation. The L function gives 1 when only the original part of image is input; otherwise it gives 0 for the predicted regions. Thus, the Cost increases when the original parts of i and j images overlap.

$$F_{cost} = \sum (m_1 + m_2) \tag{4.9}$$

Total cost is the summation of similarity and geometrical constraint terms for all points in the predefined board or space. The only parameter of this performance measure that represents the goodness of the assembly of the pieces based on textural features and geometrical shape is the transformation of pieces, T_i

This value goes towards negative if there exists a good matching between the pictures on the candidate pieces. The fitness between the pieces is increasing while the Cost function is being optimized.

Two types of optimization methods might be used in the experiments. The first one depends on the best replacement strategy. Initially, the transformations of pieces are randomly assigned. The algorithm progresses by finding best movement in each step. When the function is stuck into a local minimum, two randomly selected pieces are exchanged. All local minima are buffered to find the best assembly. The algorithm is stopped if the function reaches the best value in the local minima buffer more than n times.

The second method depends on pairing of pieces. Initially, the algorithm searches for the best pair that gives the minimum cost. Then, these paired pieces are merged to produce unique piece. The algorithm is stopped when the all the pieces in the puzzle are combined and become one piece. In this method, the algorithm backtracks when the pairing cannot improve the cost. To implement this method, the confidence and feature values of a new piece should be defined after the merging process.

$$C'_{new} = 1 - \prod_{i \in M} (1 - C'_i) \quad (4.10) \qquad f_{knew} = \frac{\sum_{i \in M} \left[\prod_{j \in M, i \neq j} (1 - C'_j).C'_i.f_{ki} \right]}{\sum_{i \in M} \left[\prod_{i \in M, i \neq j} (1 - C'_j).C'_i \right]} \quad (4.11)$$

M is the set of pieces that will be merged. (4.10) gives the new confidence value for overlapping points of pieces. The new confidence value is equal to 1 if one of piece has a confidence of 1, otherwise it is the geometrical mean of possible confidence values of that point. (4.11) gives the new k^{th} feature values by calculating the weighted mean of pieces in the set M.

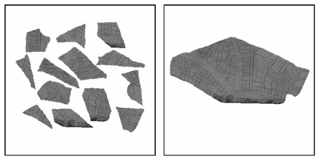
5. Experimental results

We will demonstrate the results of the proposed algorithm on three different datasets. The first data is from a simple jigsaw puzzle. Fig. 2 shows the cost function at different stages of the solutions.

F_{cost}=–1891 F_{cost}=–15150 F_{cost}=–17575 F_{cost}=–18967

Fig. 2. Different stages of the puzzle solution

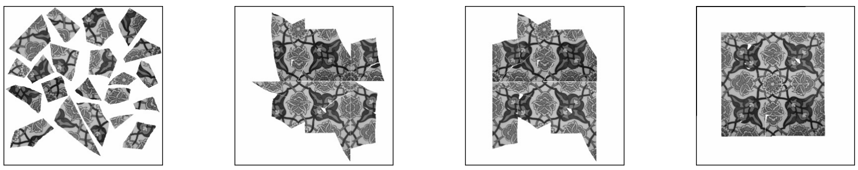
The second dataset (13 pieces) is from Stanford university website and is part of the Forma Urbis Romae dataset [10] which is a marble map of ancient Rome that has more than a thousand fragments. For this experiment, the image of a fragment from this dataset is broken artificially. Fig. 3a shows the pieces in the dataset and Fig. 3b shows the final assembly obtained.



a) F_{cost} = 0 b) F_{cost} = –19318

Fig. 3. (a) Initial layout of the pieces (b) Completed puzzle

The third dataset consists of 21 pieces of a ceramic tile. Fig. 4a shows the pieces to be assembled. Figs 4b, 4c and 4d give 3 possible solutions. The corresponding cost functions are also given. It is noted that all the solutions are visually feasible solutions in terms of the texture and geometry information and the correct solution has the minimum cost function.



(a) F_{cost} = 0 (b) F_{cost} =–18 577 (c) F_{cost} = –17841 (d) F_{cost} =–20250

Fig. 4. (a) Initial layout of the pieces (b, c, d) F_{cost} for 3 possible solutions

The last experiment has pieces from two different ceramic tiles. This experiment is important since in a real archaeological set-up, pieces may come from two or more objects. 10 pieces of the tile used in Experiment 2 are mixed with 9 pieces from another tile. Resulting assembly is given in Fig. 5.

Fig. 5. Solution obtained for mixture of pieces from two different tiles.

6. Summary and conclusions

We presented a method for the automated puzzle assembly problem using surface texture and picture. The approach is based on expanding the boundary of each piece using inpainting and texture synthesis methods and maximizing an affinity measure between the pieces. Experiments show that this approach is very promising for the automated puzzle assembly problem. Future work will concentrate on generalizing the presented algorithm to solving 3D puzzles.

References

1. Chung, M. G.; Fleck, M. M. and Forsyth, D. D. 1998. Jigsaw Puzzle Solver Using Shape and Color, in *Proc. of ICSP*.
2. Criminisi, A.; Perez, P. and Toyama, K. 2003. Object removal by exemplar-based inpainting, in *CVPR*, 721–728.
3. Freeman, H. and Gardner, L. 1964. A pictorial Jigsaw puzzles: The computer solution of a problem in pattern recognition, *IEEE Trans. Electron. Computers* 13: 118–127.
4. Fornasier, M.; Toniolo, D. 2005. Fast, robust and efficient 2D pattern recognition for re-assembling fragmented images, *Pattern Recognition* 38: 2074–2087
5. Goldenberg, D.; Malon, C.and Bern, M. 2002. A global approach to automatic solution of jigsaw puzzles, in *Proc. of Conf. on Computational Geometry*, 82–87.
6. Kampel, M.; Sablatnig, R. 2003. Profile based Pottery Reconstruction", in Martin D. (ed.). *IEEE/CVPR Workshop on Applications of Computer Vision in Archaeology*, CD-ROM, Madison, USA.
7. Kong, W. and Kimia, B. B. 2001. On solving 2D and 3D puzzles using curve matching, in *Proc. of CVPR*, Hawaii, USA.
8. Kosiba, A. D.; Devaux, P. M.; Balasubramanian, S.; Gandhi, T. L. and Kasturi, R. 2001. An automatic jigsaw puzzle solver, in *Proc. Int. Conf. Pattern Recognition.*
9. Levoy, M. 2000. The digital michelangelo project:3D scanning of large statues. Computer Graphics, in *Proc. SIGGRAPH*, New York, 131–144.
10. Papaioannou, G.; Karabassi, E. A. and Theoharis, T. 2001. Virtual archaeologist: assembling the past, *IEEE Computer Graphics and Applications* 21(2): 53– 59.
11. Radack, G. M. and Badler, N. I. 1982. Jigsaw puzzle matching using a boundary-centered polar encoding, in *CGIP*, 19: 1– 17.
12. Sagiroglu, M.; Ercil, A. 2005. A Texture Based Approach to Reconstruction of Archaeological Finds, in *Virtual Reality, Archaeology, and Cultural Heritage (VAST).*
13. Sagiroglu, M. Ş.; Ercil, A. 1990. A Texture Based Matching Approach for Automated Assembly of Puzzles, in *Proceedings of ICPR2006*.
14. Stol, J. and Leitao, H. 2002. A multiscale method for the reassembly of two-dimensional fragmented objects, *IEEE Trans. Pattern Anal. and Mach. Intel.*, 24: 1239–1251.
15. Üçoluk, G. and Toroslu, I. H. 1999. Automatic reconstruction of broken 3-D surface objects, *Comp. and Graph*. 23: 573–582.
16. Willis, A.and Cooper, D. 2002. Bayesian pot-assembly from fragments as problems in perceptualgrouping and geometric-learning, *ICPR*, 3: 297–302.
17. Willis, A. and Cooper, D. 2003. Accurately estimating sherd 3D surface geometry with application to pot reconstruction, In *CVPR Workshop: ACVA*, June 2003.
18. Wolberg, G.; Zokai, S. 2000. Robust image registration using log-polar transform, in *Proc. of IEEE ICIP*.
19. Wolfson. 1990. On curve matching, *IEEE Trans. Pattern Anal. and Mach. Intel*, 12: 483–489.