

# Optimization for Automated Assembly of Puzzles

Mahmut Şamil Sağıroğlu, Aytül Erçil

*Sabanci University, Faculty of Engineering and Natural Sciences*

*Tuzla, Istanbul TURKEY*

[aytulercil@sabanciuniv.edu](mailto:aytulercil@sabanciuniv.edu)

Tel: +90 216 483 9543, Fax: +90 216 483 9550

## 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. An fft based image registration technique is used to speed up the alignment of the pieces. Experimental results are presented on real and artificial data sets.

**Keywords:** *archeological reconstruction, partial matching, puzzle solving, inpainting, image registration*

**MSC Codes:** *68T10, 68T45, 90C05*

# Optimization for Automated Assembly of Puzzles

Mahmut Şamil Sağıroğlu, Aytül Erçil

*Sabancı University, Faculty of Engineering and Natural Sciences*

*Tuzla, Istanbul TURKEY*

[aytulercil@sabanciuniv.edu](mailto: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. An fft based image registration technique is used to speed up the alignment of the pieces. Experimental results are presented on real and artificial data sets.

**Keywords:** *archeological reconstruction, partial matching, puzzle solving, inpainting, image registration*

## 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 Freeman, (1864) who successfully solved a 9-piece jigsaw puzzle. Other works (Chung 1998; Goldenberg 2002; Kosiba 2001) also use feature based matching approaches. These methods are respectively fast so that they manage to carry out the 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 (Chung 1998).

More general partial curve matching algorithms that solve the global 2D and 3D assembly problems based on geometrical properties were presented in (Kong 2001; Radack 1982; Willis 2003). The problem of 3D curves is addressed by (Ucoluk 1999). 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 (Stol 2002) 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 (Levoy 2000). 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” (Papaioannou 2001) 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 (Willis 2002, 2003). 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. (Fornasier 2005)

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 the 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 archeology, 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 neighboring 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. The fft technique to find the best transform are explained in Section 5. Experimental results are given in Section 6.

## 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

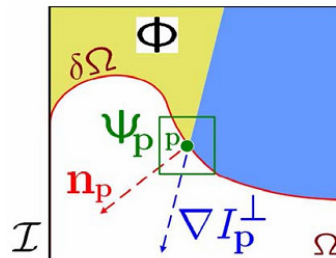
As mentioned in section 2, the first step in the assembly process is the expansion of each piece in a band around the border of the piece by predicting the pictorial information on the surface outwards. Inpainting and texture synthesis are two techniques that will be used to carry out this task (Criminisi 2003, Levin 2003; Oliveira 2001; Bertalmio 2001; Ballester 2001, 2001; Sapiro 2002). 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 (2003) 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  $m^{\text{th}}$  piece. A target band,  $I_m^+$ , outwards from the  $m^{\text{th}}$  piece is defined (so  $I_m = I_m^0 + I_m^+$ ). This target band represents the extension region of the  $m^{\text{th}}$  piece. The border between  $I_m^0$  and  $I_m^+$  is indicated by  $\delta I_m$ . 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 values are computed for the patches  $\Psi_p$  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,  $C$ :

$$P(p) = D(p).C(p) \quad (3.1)$$

$$C(p) = \frac{\sum_{q \in \Psi_p \cap I_m^0} C(q)}{|\Psi_p|}, \quad D(p) = \left| \nabla I_p^\perp \cdot n_p \right| \quad (3.2)$$

where  $|\Psi_p|$  is the area of  $\Psi_p$ ,  $n_p$  is unit vector orthogonal to the front  $\delta I_m$  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  $\delta I_m$ . 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.

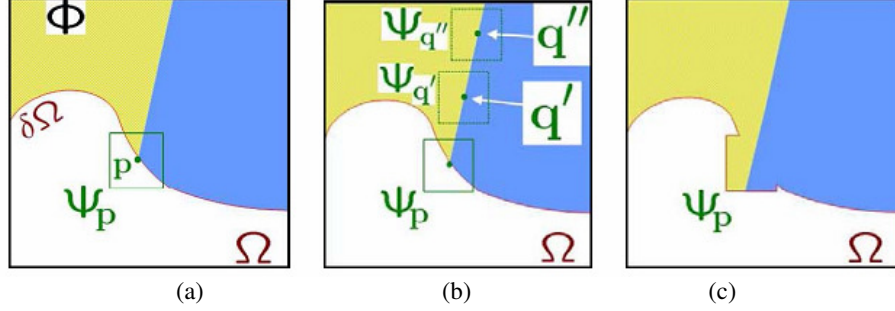


**Figure 1.** The notations: Original image, with the target region, its contour, and the source region

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:

$$\Psi_{q'} = \underset{\Psi_q \in I_m^0}{\operatorname{argmax}} d(\Psi_{p'}, \Psi_q) \quad (3.3)$$

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.

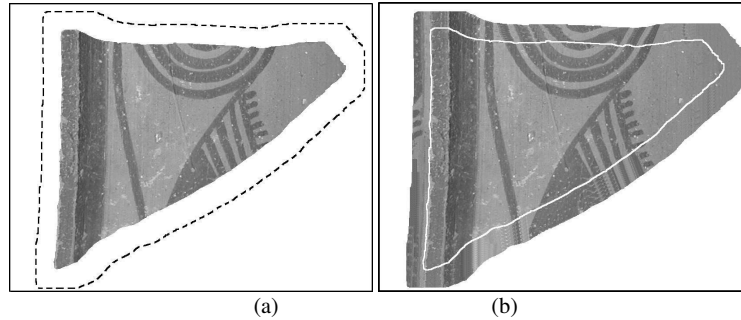


**Figure 2.** (a) We want to synthesize the area delimited by the patch  $\Psi_p$  centered on the point  $p \in \partial\Omega$ . (b) The most likely candidate patches for  $\Psi_p$  lie along the boundary between the two textures in the source region, e.g.  $\Psi_{q'}$  and  $\Psi_{q''}$ . (c) The best matching patch in the candidates set has been copied into the position occupied by  $\Psi_p$ , thus achieving partial filling of  $\Omega$ .

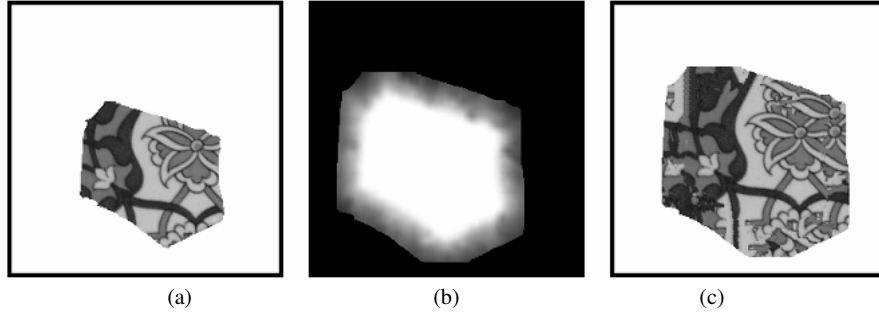
The last step for iterations is to update the confidence values. After the patch  $\Psi_{p'}$  has been filled with new values, the confidence values affected by the filling of the new patch are updated. This region is limited by the neighbors of the point  $p'$ .

$$C(p) = C(p') \quad \forall p \in \Psi_{p'} \cap I_m^+ \quad (3.4)$$

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.



**Figure 3.** (a) An archeological sherd to be expanded (b) The expanded piece



**Figure 4.** (a) the original image of a piece, (b) the corresponding confidence image, (c) the expanded images of the original piece

A pseudo-code description of the algorithm is shown in Table 3.1. The substring  $i$  indicates the current piece and superscript  $t$  indicates the current iteration.

<ol style="list-style-type: none"> <li>1. Acquire the image</li> <li>2. Extract the boundary of the <math>i^{\text{th}}</math> piece</li> <li>3. Morph the source region outwards to find the expanding band</li> <li>4. Set the confidence of the source region as 1, and the confidence of the expanding band or target region as 0.</li> <li>5. Repeat until there exist no pixel to be filled in the target region: <ol style="list-style-type: none"> <li>a. Compute new <math>D(p) \forall p \in \mathcal{R}_i^t</math></li> <li>b. Compute priorities <math>P(p) \forall p \in \mathcal{R}_i^t</math></li> <li>c. Find the patch <math>\Psi_p</math> with the maximum priority, <math>p' = \arg \max_{p \in \mathcal{R}_i^t} P(p)</math></li> <li>d. Find the exemplar <math>\Psi_{q'} \in \Phi</math> that minimize <math>d(\Psi_p, \Psi_{q'})</math></li> <li>e. Copy image data from <math>\Psi_{q'}</math> to <math>\Psi_p, \forall p \in \Psi_p \cap \Omega_i^t</math></li> <li>f. Update <math>C(p), \forall p \in \Psi_p \cap \Omega_i^t</math></li> </ol> </li> </ol>
--

Table 3.1 : The pseudo code of the expansion algorithm.

#### 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  $k^{\text{th}}$  feature values of the  $i^{\text{th}}$  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,  $C_k$ .

Let  $D_k(f_{ki}, f_{kj})$  be the distance function between the  $k^{\text{th}}$  feature values of  $i$  and  $j$  pieces.  $T_i = (\Delta x_i, \Delta y_i, \Delta \theta_i)$  denotes the transform of the  $i^{\text{th}}$  piece and  $T_i(f_{ki})$  denotes transform of the  $k^{\text{th}}$  feature extracted from the  $i^{\text{th}}$  piece. For the simplicity of expressions, the  $(\Delta x_i, \Delta y_i, \Delta \theta_i)$  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  $k^{\text{th}}$  feature distance and a similarity measure  $S_k$ :

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

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.

$$\sum_k S_k = \sum_k [D_k(f_{ki}, f_{kj}) - Th_k] \quad (4.2)$$

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  $Th_k$

$$\sum_k (S_k / Th_k) = \sum_k w_k [D_k(f_{ki}, f_{kj}) - 1] \quad (4.3)$$

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

$$\sum_{j \neq i}^{n_p} \left[ \sum_k w_k D_k(f_{ki}, f_{kj}) - 1 \right] C'_j \quad (4.4)$$

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:

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

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. If we insert the transforms to the above expression:

$$m_1(x, y) = \frac{\sum_i^{n_p} \sum_{j=i+1}^{n_p} [w_k D_k(T_i(f_{ki}), T_j(f_{kj})) - 1] T'_j(C'_j) T'_i(C'_i)}{\sum_i^{n_p} T'_i(C'_i)} \quad (4.6)$$

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.

$$m_2(x, y) = \sum_i^{n_p} \sum_{j=i+1}^{n_p} L(T_i(C_i)) L(T_j(C_j)) \quad (4.7)$$

$$L(x) = \begin{cases} 0 & \text{if } x \neq 0 \\ 1 & \text{if } x = 0 \end{cases} \quad (4.8)$$

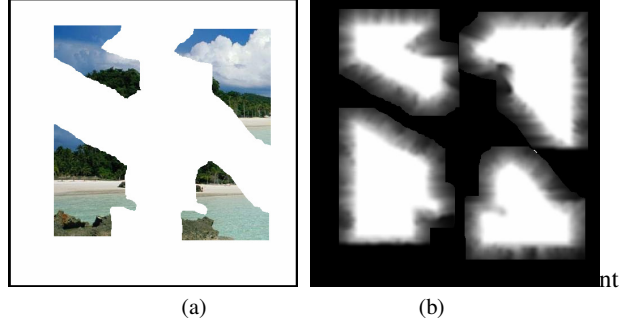
where

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.

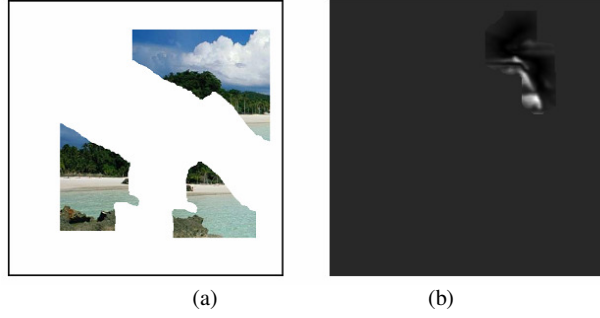
$$F_{cost} = \sum (m_1 + m_2) \quad (4.9)$$

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,  $T_i$

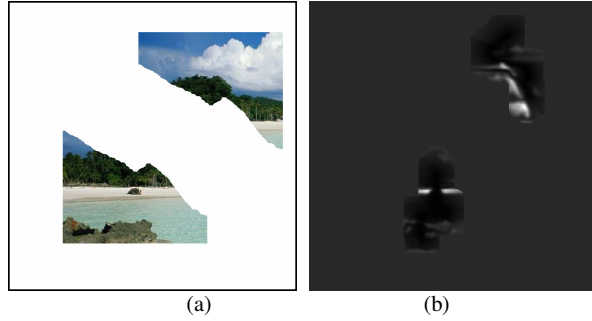
This value goes towards negative if there exists a good matching between the pictures on the candidate pieces. The fitness between the pieces is increasing while the Cost function is being optimized. Figures 5-8 demonstrate the responses of  $m_1$  and  $m_2$  functions (equations 4.6 and 4.7) for different assemblies.



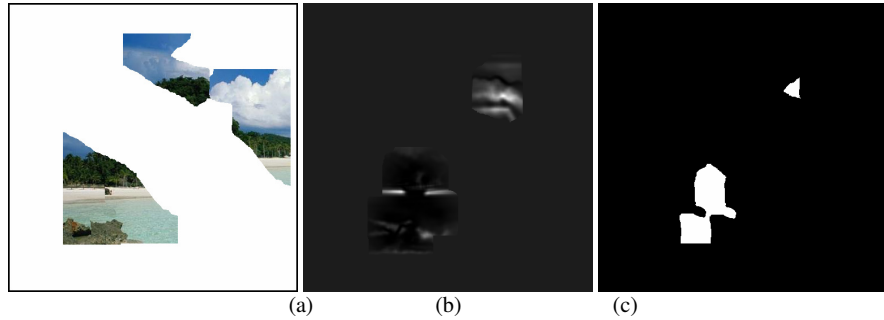
**Figure 5.** (a) Four pieces that do not intersect in any point in the space. (b) the corresponding confidence values.  $m_1=0$ ,  $m_2=0$  and  $F_{\text{cost}}=0$ .



**Figure 6.** (a) The first and second pieces are assembled. (b)  $m_1$  values for all  $(x,y)$ .  $m_2=0$  and  $F_{\text{cost}}=-988$ .



**Figure 7.** (a) Two groups of the assembled pieces. (b)  $m_1$  values for all  $(x,y)$ .  $m_2=0$  and  $F_{\text{cost}}=-3552$ .



**Figure 8.** (a) The original region of the pieces are overlapped. (b) Image represents  $m_1$  values for all  $(x,y)$ . (c) Image represents  $m_2$  values for all  $(x,y)$  and  $F_{\text{cost}}=+77564$ .

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 a unique piece. The algorithm is stopped when 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.

$$C'_{new} = 1 - \prod_{i \in M} (1 - C'_i) \quad (4.10)$$

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

$M$  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  $k^{\text{th}}$  feature values by calculating the weighted mean of pieces in the set  $M$ .

## 5. Texture Based Partial Matching Using FFT Techniques

Although, the puzzle assembly problem can be stated as the optimization of the above cost function, the optimization problem is too computationally costly. Note that minimizing the above Distance function,  $D$ , is equivalent to maximizing the correlation between the pieces. We will therefore use the fft shift theory to find a solution that will maximize the correlation between the predicted parts of a piece and other pieces. Let us first consider the solution to a 2 piece puzzle. The solution set consists of the piece  $I_0^0$  and the transformed version of the features of the piece  $I_1^0$ . The transformation consists of translation and rotation. (  $T_i = (\Delta x_i, \Delta y_i, \Delta \theta_i)$  ) The transformation that gives maximum correlation between  $I_1$  and  $I_0$  is the best match between the two pieces and hence is the solution defined as below:

$$S^2_{\text{general}} = \underset{T_1}{\operatorname{argmax}} \sum_k^{n_k} C(f_k I_0, T_1 \rightarrow f_k I_1) \quad (5.1)$$

$C$  denotes the correlation operator. The maximum correlation solution does not guarantee the real solution to the puzzle since it does not incorporate the physical constraint that two pieces cannot overlap. This constraint can be expressed in terms of correlations:

$$(I_0^0) \cap (T_1 \rightarrow I_1^0) = \emptyset \quad \text{or} \quad C(I_0^0, T_1 \rightarrow I_1^0) = 0 \quad (5.2)$$

Hence the real solution set is given by:

$$S^2 = \left\{ \underset{T_1}{\operatorname{argmax}} \sum_k^{n_k} C(f_k I_0, T_1 \rightarrow f_k I_1) \mid C(I_0^0, T_1 \rightarrow I_1^0) = 0 \right\} \quad (5.3)$$

The solution set lies where the original pieces have correlation 0, and the  $I_1^{\text{th}}$  piece has maximum correlation with the  $I_0^{\text{th}}$  piece.

These correlations can be carried out very fast using FFT operations as in image registration techniques (Wolberg 2000)

$$S^2_{\text{general}} \equiv \operatorname{imax} \sum_k^{n_k} F \left( \frac{F(f_k I_0) \cdot F^*(f_k I_1)}{|F(f_k I_0)| \cdot |F^*(f_k I_1)|} \right) \dots\dots\dots (5.4)$$

$$\text{and also } C(I_0^0, I_1^0) = \overline{F(F(I_0^0) \cdot F^*(I_1^0))} \quad (5.5)$$

where imax returns the indices of the max value. If we substitute (5.4) and (5.5) in (5.3);

$$S^2 = \text{imax} \left[ \sum_k^{n_k} \overline{F} \left( \frac{F(f_k I_0) \cdot F^*(f_k I_1)}{|F(f_k I_0)| \cdot |F^*(f_k I_1)|} \right) \cdot L \left[ \overline{F}(F(I_0^0) \cdot F^*(I_1^0)) \right] \right] \quad \text{where } L[x] = \begin{cases} 0 & \text{if } x \neq 0 \\ 1 & \text{if } x = 0 \end{cases} \quad (5.6)$$

$F$ ,  $F^*$  and  $\overline{F}$  denote the Fourier operator, its complex conjugate and the inverse Fourier operator, respectively. When the inverse Fourier value in the second part of the formula is zero, the maximum values of the first part of the formula gives the ideal transformation parameters.

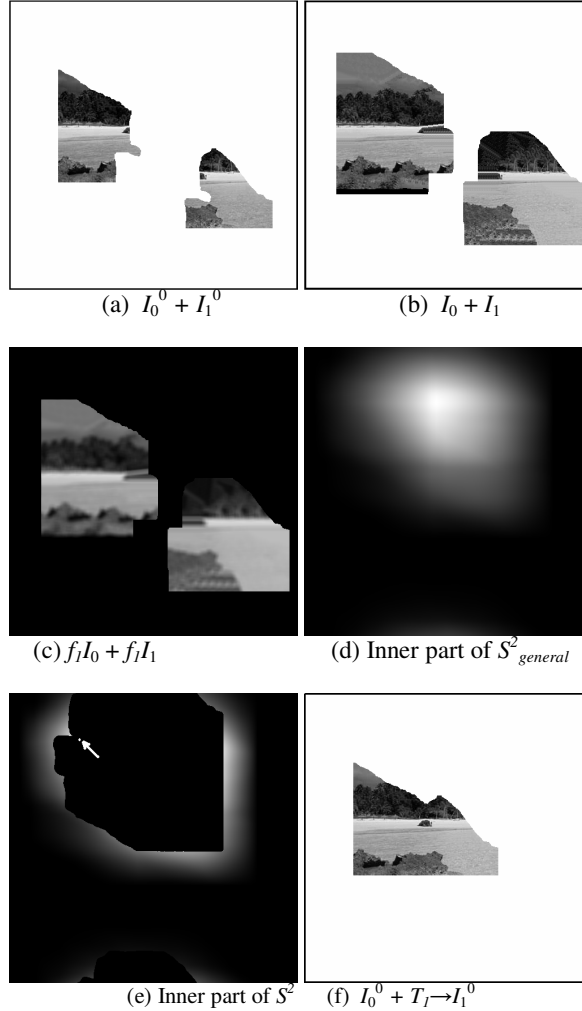
Figure 9 shows the steps of a 2 pieces puzzle assembly. In figure 9.a, there are two original pieces. In 9.b, the expanded pieces are shown. Then, the features are derived from the expanded pieces. In 9.c, only the mean feature is shown. The image in 9.d represents the correlation matrix between these two expanded pieces. This is the general solution for the problem. If the constraint expression is applied, however, the image in 9.e is found. This matrix is the output of the inner part of expression (5.6). The arrow in this image indicates the maximum point in this matrix. The indices of this maximum point give the translation coefficients. The last image in figure 2 shows the solution if this translation is applied to the second piece.

The FFT solution helps to solve only translation. The rotation can be solved using polar coordinates in an iterative scheme by using the same formulation. See (Wolberg 2000) for details.

The solution outlined above can be generalized to the solution of more realistic puzzles with larger number of pieces. Let  $n_p$  be the number of pieces involved. The solution is the set of appropriate transformations of each piece:

$$S^{n_p} = \{ 0, T_1, T_2, T_3, \dots, T_{n_p} \} \quad (5.7)$$

Let's assume all the pieces are randomly dispensed on a big enough board ( $B$ ). We randomly select a piece ( $I_i$ ). For this piece, the transformation giving maximum correlation is obtained using the above technique. (Sağıroğlu, 2006)



**Figure 9.** (a) Original pieces (b) Expanded pieces (c) First features of expanded pieces (d,e) Correlations (f) Solution of 2 pieces

$$\left\{ \operatorname{argmax}_{T_i} \sum_k^{n_k} C(f_k B - f_k I_i, T_i \rightarrow f_k I_i) \mid C(B - I_i^0, T_i \rightarrow I_i^0) = 0 \right\} \text{ where } B = \{ I_0^0, I_1^0, I_2^0, \dots, I_N^0 \} \quad (5.8)$$

The algorithm used in solving the puzzle is outlined below:

1. Place the pieces on a board ( $B$ )
2. Find the best transform for a randomly selected piece  $t$  using the presented method
3. If there exists a piece that can be moved, go to step 2
4. Select a piece and transform it
5. Go to step 2, until the puzzle is uniquely assembled even if all pieces are tried in step 4

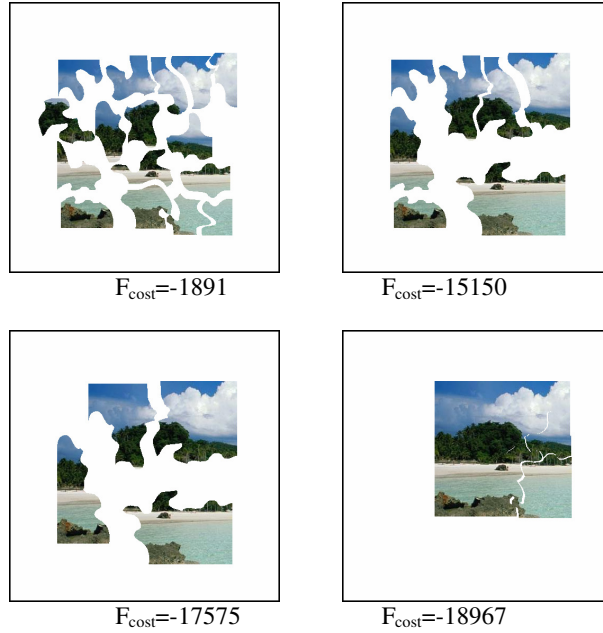
The main drawback of this operation is that there may be multiple solutions to the problem. These multiple solutions depend on the initial placement of pieces on  $B$  and the random selections of the  $i^{\text{th}}$  piece. In these situations, the affinity measure developed above is used. For a possible placement given by the proposed technique, the  $F_{\text{cost}}$  is calculated. If the algorithm reaches to a new solution that has a lower cost value than before, the new one becomes the best placement. These iterations continue until the last  $N$  possible placement cannot offer a better cost. This  $N$  value directly depends on the complexity of the puzzle. The main argument of the complexity for a puzzle is the number of the pieces. So the number  $N$  mainly depends on the  $n_p$  value. We used  $N \approx n_p^2$  for our experiments.

Hence the final version of the algorithm has the following two additional steps:

6. Find  $F_{\text{cost}}$  and if the transformed piece is better then before, save the transformation as a best solution and clear the counter  $N$ .
7. Go to step 1, until  $N$  reaches to  $n_p^2$

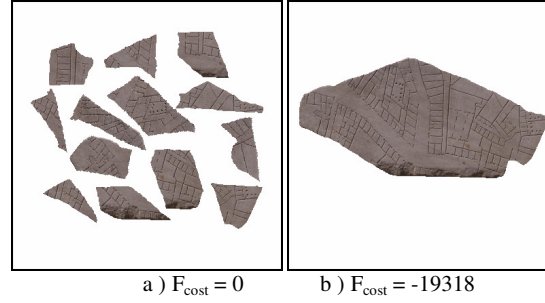
## 6. Experimental results

We will demonstrate the results of the proposed algorithm on three different datasets. The first data is from a simple jigsaw puzzle. Figure 10 shows the cost function at different stages of the solutions.



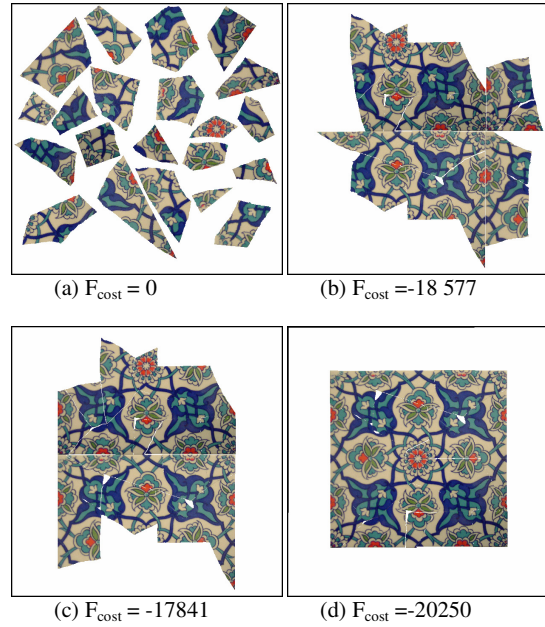
**Figure 10.** Different stages of the puzzle solution

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. Figure 11.a shows the pieces in the dataset and Figure 11.b shows the final assembly obtained.



**Figure 11.** (a) Initial layout of the pieces (b) Completed puzzle

The third dataset consists of 21 pieces of a ceramic tile. Figure 12.a shows the pieces to be assembled. Figures 12.b, 12.c and 12.d 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.



**Figure 12.** (a) Initial layout of the pieces (b,c,d)  $F_{\text{cost}}$  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 Figure 13.



**Figure 13.** Solution obtained for mixture of pieces from two different tiles.

## 7. Computational Complexity

The complexity of affinity calculation depends on the size of the image, number of features,  $n_k$ , and the number of pieces,  $n_p$ . The equation (4.6) contains the transformations, the distance calculations, the confidence multiplications and the summations. The operations for a transformation of any point are four real multiplications and two summations. The operations of the distance calculation are one summation and one multiplication. The weighting and the subtraction of the threshold are performed by one multiplication and one summation, respectively. Thus, the total operations for the calculation of the distance function of two transformed features are ten multiplications and six summations (Total of 16 operation). For all features in an  $(x_0, y_0)$  point, calculating the total distance has a complexity  $O(10n_k)$ . So, the complexity of the  $m_1$  for a point becomes  $O(n_p^2 \times (10n_k + 1) + 10n_p) \approx O(C_1 n_p^2 n_k)$ . Similarly, the equation (4.7) has a complexity  $O(C_2 n_p^2)$ . The summation of these two complexities gives the complexity of the  $F_{cost}(x_0, y_0)$  function at the point  $(x_0, y_0)$  as  $O(C_1 n_p^2 n_k + C_2 n_p^2)$ . Suppose that the images have a size of  $N \times N$ . Then, the complexity of the total  $F_{cost}$  is  $O(N^2(C_1 n_p^2 n_k + C_2 n_p^2))$ .

The complexity of the optimization of the  $F_{cost}$  function by searching exhaustively all possible translations and the rotations is important to decide the following operations. All possible translation set includes  $N \times N$  translation and suppose that there exists  $N$  possible discrete rotation angle. The complexity of the search operation becomes  $O(N^5(C_1 n_p^2 n_k + C_2 n_p^2))$ . If the number of pieces is not very big, the dominant variable in this complexity is the size of the board. The fifth degree of the size of a board, which can contain all pieces freely, is very high value to compute. Thus, this is not feasible for practical implementation.

We have introduced the Fourier method in order to reduce the time consumption while searching for the best place for a piece in the optimization stage. Image registration methods using 2D Fourier transform can find the image correlations very fast. If it is considered that the original and expanded regions of pieces are correlated, assembly algorithm became faster by implementing the mentioned method.

Here, the success of the algorithm is proportional to the size of the search buffer,  $n_{buf}$ . The time for the solution increases as the size of the buffer increases. In the first iteration (for the step 3), we compute the expression (5.8)  $(n_p - 1)$  times for each  $n_{buf}$ . Then, (as a coarse assumption) we compute  $(n_p - 2)$  times for each buffer element.

Thus, we use this expression  $n_{buf} \cdot n_p \cdot (n_p - 1) / 2 \approx n_{buf} \cdot n_p^2 / 2$  times. As we define the complexity of the mentioned expression, total complexity becomes  $O(20C_1 n_{iter} n_k n_{buf} n_p^2 N^2 \log(N))$  for this algorithm. Also, suppose that the size of the buffer should, at least, be linearly proportional to the number of pieces. Thus, the complexity may be  $O(20C_1 n_{iter} n_k n_p^3 N^2 \log(N))$ .

To improve the optimization, some other methods might also be implemented. The self-organizing map and genetic algorithm are reasonable methods for optimizing the above cost function.

## 8. Summary and conclusions

We presented a method for the automated puzzle assembly problem using surface texture and picture. In this research, the main idea used to solve the assembly problem can be explained in that the correlation between the features of the predicted region and the right neighbor is significantly higher than the alternative pairings. The approach is based on expanding the boundary of each piece using inpainting and texture synthesis methods and maximizing an affinity measure that represents the appropriateness of the assembly based on textural features and geometrical shape. We try to find the best transformations of pieces that maximize harmony of textures of fragments while the geometrical constraints are being satisfied. 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.

## 9. Acknowledgements

This research was partially supported by NSF- IIS-0205477, TUBITAK EEAG 104E155 and FP6-2004-ACC-SSA-2 SPICE projects.

## 10. References

- A. Ballester, M. Bertalmio, V. Caselles et al., "Filling in by joint interpolation of vector fields and gray levels", *IEEE Transactions on Image Processing*, 10(1), August 2001
- A. Ballester, M. Bertalmio, V. Caselles et al., "A variational model for filling-in gray level and color images", In *Proc. of ICCV*, 2001, p. 10
- M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier stokes fluid dynamics and image and video inpainting", In *Proc. of Conf. Computer Vision Pattern Recognition*, Hawaii, December 2001, pp. 1:355-365
- M.G. Chung, M.M. Fleck, and D.D. Forsyth, "Jigsaw Puzzle Solver Using Shape and Color", In *Proc. of ICSP*, 1998.
- A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting", In *CVPR*, 2003, pp. 721-728.
- H. Freeman and L. Gardner, "A pictorial Jigsaw puzzles: The computer solution of a problem in pattern recognition", *IEEE Trans. Electron. Computers*, 13(1964) , pp. 118-127.
- M. Fornasier, D. Toniolo, "Fast, robust and efficient 2D pattern recognition for re-assembling fragmented images", *Pattern Recognition*, 38(2005), pp. 2074-2087
- D. Goldenberg, C. Malon, and M. Bern, "A global approach to automatic solution of jigsaw puzzles", In *Proc. of Conf. on Computational Geometry*, 2002, pp. 82-87.
- M. Kampel, R. Sablatnig, "Profile based Pottery Reconstruction", in Martin D. (ed). *IEEE/CVPR Workshop on Applications of Computer Vision in Archaeology*, CD-ROM, Madison, USA, 2003.
- W. Kong and B.B.Kimia, "On solving 2D and 3D puzzles using curve matching", In *Proc. of CVPR*, Hawaii, USA, 2001.
- A.D. Kosiba, P.M. Devaux, S. Balasubramanian, T.L. Gandhi, and R. Kasturi, "An automatic jigsaw puzzle solver", In *Proc. Int. Conf. Pattern Recognition*, 2001.
- A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics", In *Proc. of ninth International Conference on Computer Vision*, vol. 2, 2003, pp. 305
- M. Levoy, "The digital michelangelo project: 3D scanning of large statues. Computer Graphics", In *Proc. SIGGRAPH*, New York, 2000, pp. 131-144.
- M. M. Oliveira, B. Bowen, R. McKenna, and Y. S. Chang, "Fast digital image inpainting", In *Proc. of Visualization, Imaging, and Image Processing*, Marbella, Spain, September 2001, pp. 261-266
- G. Papaioannou, E.A. Karabassi, and T. Theoharis, "Virtual archaeologist: assembling the past", *IEEE Computer Graphics and Applications*, 21(2)(2001), pp. 53- 59.
- G.M. Radack and N.I. Badler, "Jigsaw puzzle matching using a boundary-centered polar encoding", In *CGIP*, 19(1982), pp. 1- 17.
- G. Sapiro, "Image inpainting", from *SIAM News*, Vol. 35, No 4, 2002
- [
- M. Sağıroglu, A. Ercil, "A Texture Based Approach to Reconstruction of Archaeological Finds", In *Virtual Reality, Archaeology, and Cultural Heritage (VAST)*, 2005.
- M. Ş. Sağıroglu, A. Ercil, "A Texture Based Matching Approach for Automated Assembly of Puzzles," *Proceedings of ICPR2006*.
- M. Ş. Sağıroglu, "Computer Aided Puzzle Assembly based on Shape and Texture Information," *Ph.D Thesis*, 2006.
- Stol and H. Leitao, "A multiscale method for the reassembly of two-dimensional fragmented objects", *IEEE Trans. Pattern Anal. and Mach. Intel.* ,24(2002), pp. 1239-1251.

- G. Üçoluk and I.H. Toroslu, "Automatic reconstruction of broken 3-D surface objects", *Comp. and Graph.*, 23(1999), pp. 573-582.
- A. Willis and D. Cooper, "Bayesian pot-assembly from fragments as problems in perceptual grouping and geometric-learning", In *ICPR*, 3(2002), pp. 297-302.
- A. Willis and D. Cooper, "Accurately estimating sherd 3D surface geometry with application to pot reconstruction", In *CVPR Workshop: ACVA*, June 2003.
- G. Wolberg, S. Zokai, "Robust image registration using log-polar transform", In *Proc. of IEEE ICIP*, 2000.
- [19] Wolfson, 'On curve matching', *IEEE Trans. Pattern Anal. and Mach. Intel*, 12(1990), pp. 483-489.