Assembly of Shredded Pieces Depending on Distance and the Angle Change

Sameh Shawki Ebrahim
sameh_shk_2017@yahoo.com

Yasser M. Kamal
dr_yasser_omar@yahoo.com

Whaled Fakhr
waleedfakhr@yahoo.com

College of Computing and Information Technology
Arab Academy for Science
Technology and Maritime Transport
Cairo, Egypt.

***Abstract* - In several cases there are many documents or images that were shredded either by hand or by a machine. Moreover the images will be highly rich in color than a document. In shredding images by hand there are many changed points that required a proposed model which is able to match these pieces. On the other hand shredding by machine has only four changed points for each shredded piece (at corners). This paper introduces a method to assemble shredded pieces to form the desired image which is seriously vital in many applications such as multimedia and security applications. The proposed method gets the interest points which have a change in the angle and stay unchanged for a long distance, also it will use three features to get the matching between pieces to increase the matching accuracy. Three features are distance between interest points, angle of change at interest points, and the boundary color at the interest points. The proposed method starts by getting the outer contour of each shredded piece, then detects the interesting points, then it tries to match between different pieces. It is clear that the number of points on the outer contour is very large, so detecting the interest points will improve performance for getting the right value of matching between each two shredded pieces. Finally, a complete graph is constructed for all shredded pieces which form a matching graph whose edges represent matching success value and vertices represent shredded pieces. The method scored an overall accuracy of matching of 85% compared to previous work and improved the overall performance by 14% compared with previous related work.**

Keywords - assembly of shredded* pieces*; Computer aided analysis; Image processing.

I. INTRODUCTION

Recently in our life, there are a lot of important documents and photos which might be shredded either by machine or by hand. It is considered as a very hard job to arrange these fragments and try to reassemble them together manually, especially if we have a large number of shredded pieces.

Fig.1 shows this scenario. There are eight shredded pieces belongs to the same photo, and we need to assemble these eight pieces to construct the desired image. In the real applications, the main challenge of this problem mainly is that we haven't any previous knowledge about the shape or content of image, i.e. there is no solution in advance like the jigsaw puzzle solving problem [1],[2], [3],[4],[5],[6].

There are two unique characteristics that make our work different from other automated assembly problem solvers.

A. The targeted object of our method is different from the most previous works. In the first place, unlike the jigsaw puzzles, we do not have prior knowledge on the shape or size of shredded pieces.

B. The information incorporated and the way to use are different from those for other works. In many jigsaw solving and document reconstruction works only shape information is used to reconstruct, in our approach we used three feature related to shape and color for each point on the outer contour of the shredded pieces.

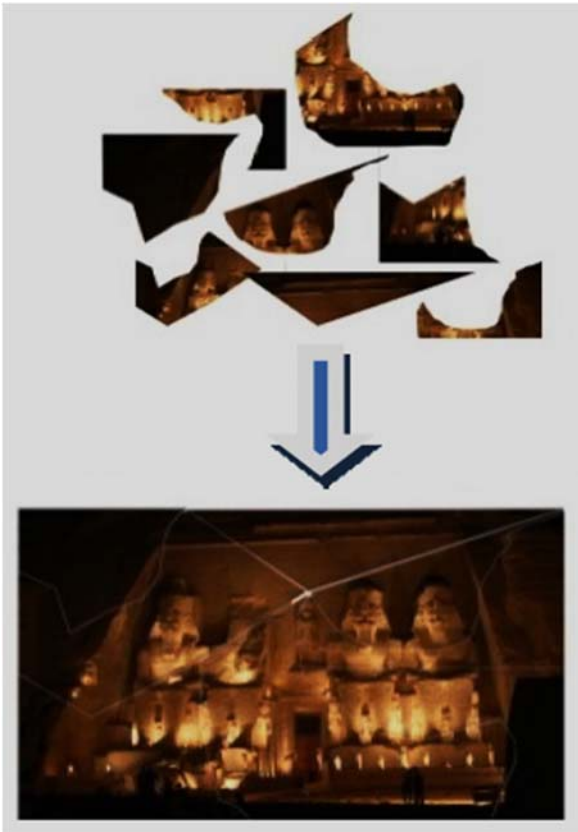

Fig. 1. 8 shredded pieces belong to similar ones

This paper is organized into five different sections; the related work is introduced in section II. Material and methods is proposed in section III. Section IV presents the evaluation of the proposed method by comparing it with the other related work methods in different situations. Finally, section V gives the conclusion then the discussion as well as the future work.

II. RELATED WORK

There are several techniques proposed in the previous researches that tried to assemble the shredded pieces for both images and documents. Hairong Liu and et al, presented an automatic method for assembly of shredded pieces from multiple photos [7]. They used Canny edge detector to get the outer contour, then used the visual curvature method to get the salient points on contour. Finally, they used only two features to get the matching between each two shredded pieces; color feature, and geometric feature at each salient point.

Abdullah M. Moussa presented a technique for puzzle solver [8]. It used an accelerated edge matching based on a greedy method to compare each two boundaries of all shredded squares then make rotation and translation of each shredded square to get the desired photo.

A Fast Shredded Document Reconstruction is presented by Mohamed El-Mahallawy and, Eid Emary[9]. It works on DARPA database documents, each document consists of 226 shredded pieces, uses ruled-lines and written letters that exist in documents to assemble shredded pieces but it doesn't work fit if the paper has no ruled-lines.

Using three features of curve matching will give our proposed method a good accurcy of matching than assembly of shredded pieces from multiple photos [7], also getting interest points for every shredded pieces which is less than the overall points on the outer contour of each shredded piece may give it a better performance than a puzzle solver [8] and A Fast Shredded Document Reconstruction [9].

III. MATERIAL AND METHODS

The Proposed system workflow has been shown in figure2. It is used to describe the most important steps. It first starts with digitizing each shredded piece and segment into separate files. Then, it extracts the boundary contour for each shredded piece. After that, getting maximum interest points based on the change of angle and distance on contour. These interest points may allow dividing each contour to a set of fragments. Furthermore, the proposed method compares each fragment with all other fragments of the shredded pieces depending on the angle, distance, and color then the matching graph is constructed. Furthermore, the proposed technique construct spanning tree that links all shredded pieces together in order of gathering related pieces. Finally, merging the pieces together for constructing the desired photo is done by using spanning tree.

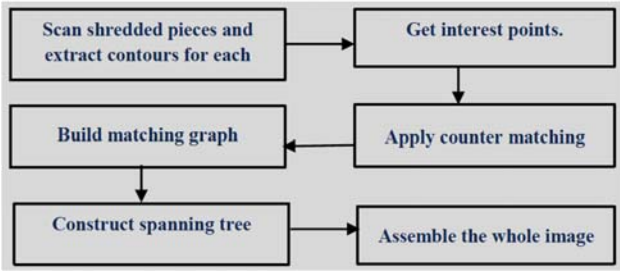

Fig. 2. System workflow

The following subsection will illustrate in more details the steps of the proposed methods.

A. Contour Extraction

The first step is to digitize each shredded piece. This can be done by a scanner or a camera. Digitization can be achieved by putting these pieces over a background with high contrast. In case of using the camera, the image plane of the camera should be parallel to the fragment and the environment light should be controlled.

Then, give each shredded piece a sequencing number and arrange them. After that changing each digitized shredded into gray scale image (single dimension array), then changing it to black and white (0 and 1) as shown in figure 3, 4 and 5.

Figure 5 illustrates how to get the outer contour of each shredded piece. This can be done by changing all pixels to zeros except the pixels that are originally ones and have one of their neighboring pixels is equal to zero.

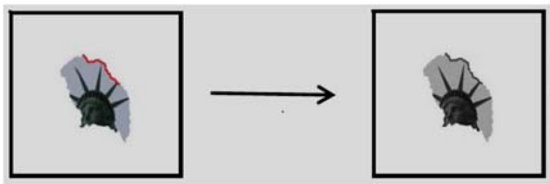
Fig. 3. Gray Scale Image

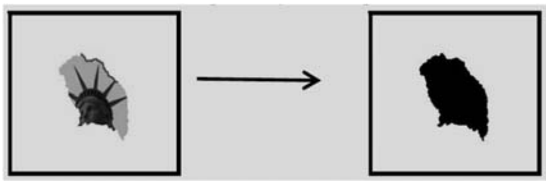
Fig. 4. White and black (binary) Image

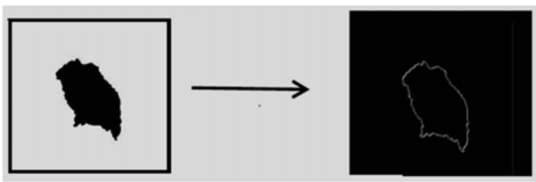
Fig. 5. Outer Contour

B. Getingt Interest points

Direct curve matching is very time consuming; because boundary contour has a large number of points on it [10], [11]. While, detecting interest points that act as an anchor points can enhance the performance of the pieces matching. These interest points are extracted based on satisfying two features:-

- There exists an abrupt change in the angle
- The distance between each consecutive interest points is greater than a certain threshold.

The algorithm1 shows how to get the interest points through two phases. The first phase is to sort the contour points. While, the second one, is to scan sorting points and extract points that satisfy interested criteria as described above.

Algorithm 1 Extracting Interest points

Phase 1: Order Contour Points

Input: *array of all boundary points.*

Output: *array of boundary points with it is order.*

1: *Initialize mask matrix **B** by the **A**(Binary Image)**;** **P**=1;*
2: *Initialize **CP**,**SP** by any pixel on contour;*
3: *Initialize **OrderList** by NULL;*
4: *Loop*
5: *if (**B**(**CP**)==1) Then //not visited before*
6: *Add **CP** to **OrderList**;.*
7: ***B**(**CP**)=2; // mark it as visited.*
8: *end if*
9: *check 8-neigbour of **CP** if there exists a pixel **P** such that **A**(**P**)=1 && **B**(**P**)=2 assign **CP**= **P**; **P** = **P** + **1**;*
10: *Until **CP**=**SP***

Phase 2: Extracting Interesting Points

Input: *array of boundary points with it is order*

Output: *interest points.*

1: *For each Pixel in the **OrderList** calculate absolute change in the X & Y directions with both its previous & next pixel as* (x_1, y_1, x_2, y_2) *;*

2: *At each Pixel get **CA**(change angle) as follows:*
if $((x_1, y_1, x_2, y_2)$ *in ((1,0,0,1),(0,1,1,0))) then*
$CA = 90;$
elsif $((x_1, y_1, x_2, y_2)$ *in ((0,1,1,1),(1,1,1,0), (1,0,1,1),(1,1,0,1))) then*
$CA = 135;$
else
$CA=0;$
End if;
3: *Mark all pixels with angles 90 or 135 as candidate interest points;*
4: *Get Euclidian distance at each interest points with its next interested one.*
5: *Ignore all points that have a distance less than a certain threshold.*

C. Contour Matching

After extraction of the interest points on the outer contour, we will divide each contour to a set of fragments, depending on the interest points the we will compare each two fragments of each two shredded pieces together. The first comparison will be in size of two fragments if they are equal then we continue otherwise these two pairs will be discarded. For each success pairs of fragments we will compare all interest points inside both fragments according to the angle of change, distance with the previous and next interest points and color. Let, for example, fragment number (i) on contour number 1 and fragment number (j) on contour number 2, first getting the length of each fragment, and we will set a low scale threshold λ for detecting the difference in length between two fragments, if the length of the fragments is equal to the length of ± λ then get success factor between all interest points in the fragment i and fragment j depending on distance , angle, and color of each interest point.

Fig 6. Interest points or two shredded pieces

Figure. 6 shows the interest points extracted on the two boundary contours. The interest points are densely distributed over all boundary and all essential interest points are detected. Thus, the interest points can work as "anchor" points and we use these points to find the matching parts. Since the interest points are densely distributed over the boundary, the accuracy of the matching will not be affected. Also, a number of interest points are significantly smaller than the number of total points over the boundary contours so the execution time will be reduced.

For each interest point on the outer contour its feature vector consists of 5 components as follows:

$$\{p_d, p_a, p_{cr}, p_{cg}, p_{cb}\}$$

Where:

P_d is the distance between each adjacent interested points.

P_a is the angle at each interest point.

P_{cr} is the red value of the interest point in the original image.

P_{cg} is the green value of the interest point in the original image.

P_{cb} is the blue value of the interest point in the original image.

According to the above features, we can calculate success value between each pair of fragments and then calculate the overall success value for matching between both shredded pieces and neglect the success values that less than a certain threshold.

D. Matching Graph Construction.

The matching graph is constructed based on candidate matching between shredded pieces. It consists of vertexes and edges. Each vertex represents a shredded piece and each edge means that there is a matching between two shredded pieces that edge links them and the success value is associated with the edge.

Fig.7 shows an example of the matching graph for Ramsis-II Image.

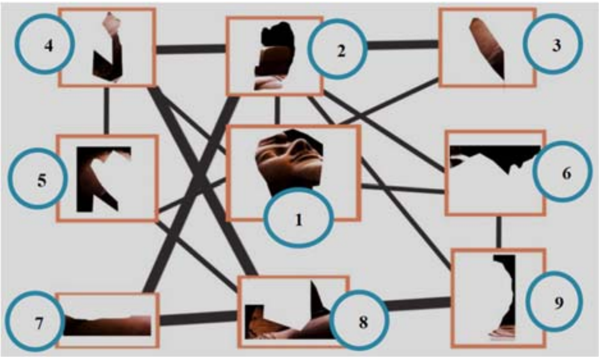

Fig. 7 Example for matching graph

E. Construction of Spanning Tree.

Building spanning tree from the matching graph allows constructing the whole image from the shredded pieces. First, it starts with a shredded piece that has a maximum number of links with the other shredded pieces and sets it as a root of the tree. Then, it appends all of the nodes that have direct edge with the root node as children. The appending process is done in a descending order based on how many edges they have. After that, the first child is checked and its connected nodes are replaced as its children. The iteration process is repeated until that all nodes are added in their proper level in the spanning tree. Figure8 shows the steps of construction spanning tree from the matching graph of fig 7.

Note that:-

- Each node appears once in the spanning tree because we will remove any node from higher levels when it appears in the lower levels of the spanning tree.
- An Image with (n) shredded pieces can be merged with (n-1) links so its spanning tree has (n-1) links.

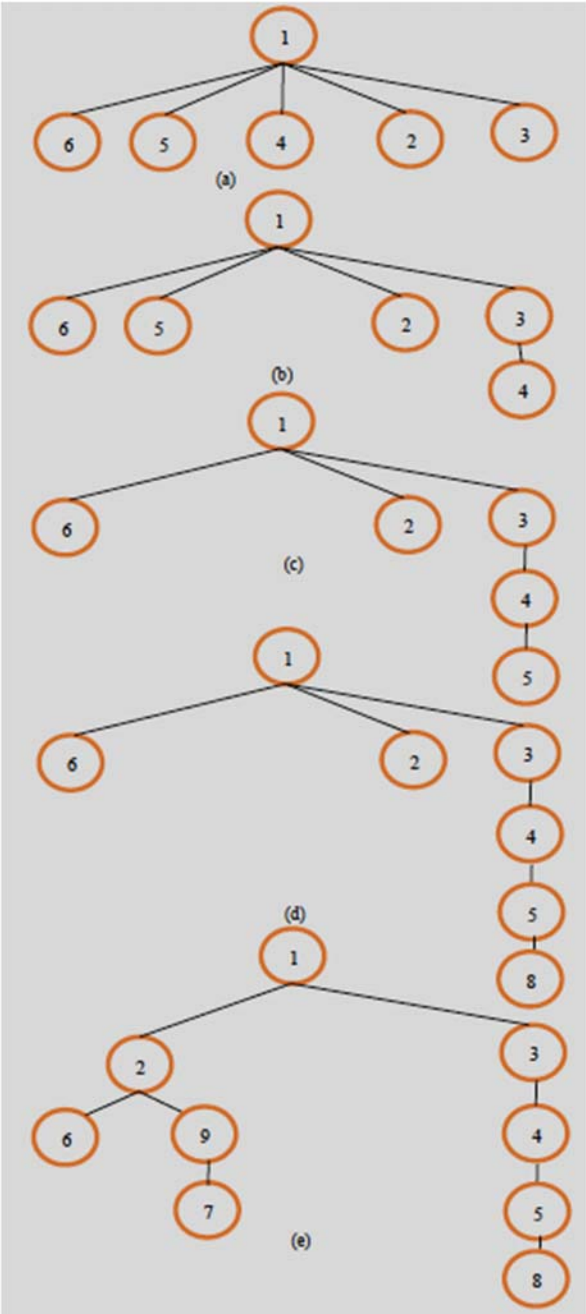

Fig 8. Spanning tree

DOI 10.5013/IJSSST.a.18.01.07 ISSN: 1473-804x online, 1473-8031 print

IV. EXPERIMENTS

In this section we will evaluate the effectiveness and the efficiency of the proposed method compared with the related work done in [7]. For comparing the performance of the two algorithms, we applied the two algorithms on four images as shown in figure 9, also all experiments have been run on a Core i5(2.3-GHz PC) with 8 GB of RAM.

Fig9. Experiment images.

The implementation details of the proposed method are listed as follows:

1) The interest point detection will get all consecutive interest points that are separated by at least 5 pixels.

2) 8-neighborhood is used to estimate the average colors.

3) The matched contour is taken into consideration if its length is greater than 20 pixels. This is the smallest matching length, which means if the length of a matched contour is too small, this matching is not reliable; thus, we discard it.

4) The maximum difference in length allowed during the linking process is 2 pixels which is small to ensure that the linked contours are truly matched.

Figures.10,11,12,13 are shown the results of contour matching of the shredded pieces of the four testing images. In this experiment, we notice that if we have a large portion of the photo has no big change in color, then distance and the angle information play the major roles in the matching procedure and the performance is noticeable improved (as images number 1 and 4 the performance improved with 15%,14% respectively), on the other hand if we have a large portion of the photo has a lot of colors, the color information plays a critical role in the matching procedure so the performance will improved with small value (as image number 2 the performance improved with only 7%), finally in cases of black and white images (image number 4), when we applied two algorithms on the same image, the proposed system gives successful results but the related work system doesn't give successful results.

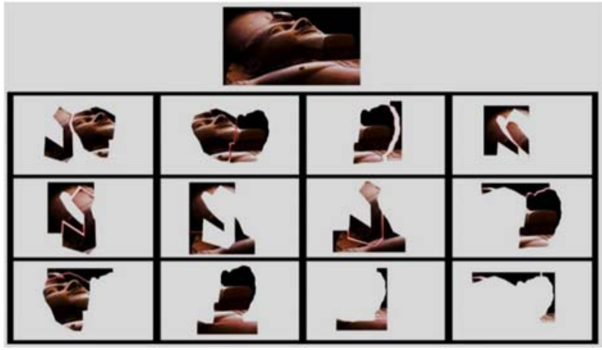

Fig10. Contour matching result on the shredded pieces of first image.

Fig11. Contour matching result on the second image (Egypt Flag).

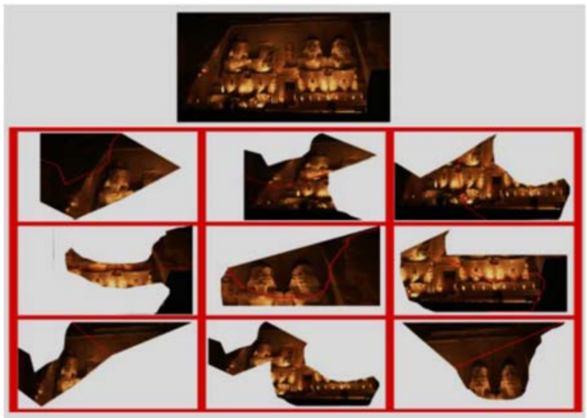

Fig12. Matching results of the third image.

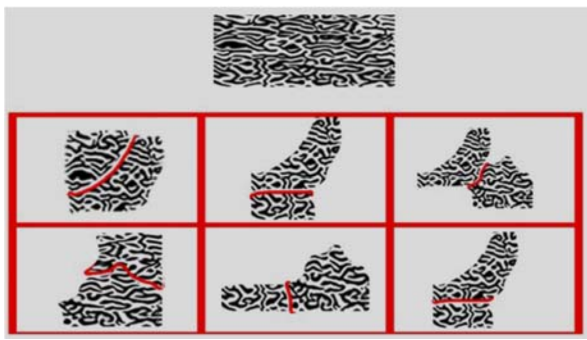

Fig13. Contour matching result on the shredded pieces of forth image.

System/image	1	2	3	4
1-Proposed System	123	156	134	124
2-Related Work	141.45	166.92	152.76	300(Fail)

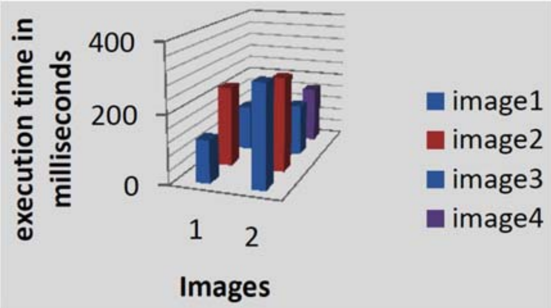

Fig14. Chart for matching time of two algorithms.

a- shredded piece (8)

b- (8) + (5)

c- (8) + (5) + (4)

d- (8) + (5) + (4)+ (3) – ***first branch***

e- (7) + (9)

f- (6) + (2)

g- (7) + (9)+ (6) + (2) – second branch of the spanning tree

h- (7) + (9)+ (6) + (2)+ (8) + (5)+ (4) + (3)+(1) – full image

Fig15. Steps of assemble the first image (Ramsis II).

Figure 14 shows chart for comparison between two systems with respect of execution time in milliseconds.

Now we will illustrate how we can assemble the first image to get the desired image, in the previous section we showed how we get the spanning tree of this image and now we will apply this tree (fig.8_e) to get the desired image, we will use depth-first technique [12], [13] to assemble the desired image, figure 15 shows the steps of the assembling the first image.

V. CONCLUSION AND FUTURE WORK

This paper presented a proposed method for automatic assembly of shredded pieces from a single photo. This method utilizes for getting the interest points that used to get success value for matching procedure between each two shredded pieces, Furthermore, these interest points have difference angle (90 or 135 degrees) and have a long distance between it and the previous interest point without change in angle. The proposed method scored an overall accuracy of matching of 85% and improved the overall performance with 14% comparing with the previous related work. Also we detect that it make good performance improved when we have an image has no big change in the color, on the other hand if we have an image that has a lot of colors, performance will be improved with a small value.

In future work, we will apply post processing techniques to complete the holes due to material loss for removing discontinuities long borders so as to further improve the result also we try to build a learning system for assembly of documents.

REFERENCES

[1] H. Freeman and L. Garder, “Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition,” IEEE Trans. Electron. Comput., vol. 13, no. 2, pp. 118–127, 1964.

[2] G. Burdea and H. Wolfson, “Solving jigsaw puzzles by a robot,” IEEE Transactions on Robot. Autom., vol. 5, no. 6, pp. 752–764, 1989.

[3] R. Webster, P. LaFollette, and R. Stafford, “Isthmus critical points for solving jigsaw puzzles in computer vision,” IEEE Trans. Syst., Man, Cybern., vol. 21, no. 5, pp. 1271–1278, 2002.

[4] D. Kosiba, P. Devaux, S. Balasubramanian, T. Gandhi, and R. Kasturi, “An automatic jigsaw puzzle solver,” in Proc. 12th Int. Conf. Pattern Recognition, 1994, vol. 1, pp. 616–618.

[5] D. Goldberg, C. Malon, and M. Bern, “A global approach to automatic solution of jigsaw puzzles,” in Proc. Conf. Computational Geometry, 2002, pp. 82–87.

[6] F. Yao and G. Shao, “A shape and image merging technique to solve jigsaw puzzles,” Pattern Recognit. Lett., vol. 24, no. 12, pp. 1819–1835, 2003.

[7] Hairong Liu,Shengjiao Cao,and Shuicheng Yan,Senior Automated, assembly of shredded pieces from multiple photos IEEE 2011,

[8] Abdullah M. Moussa, Jigsopu: Square Jigsaw Puzzle Solver with Pieces of Unknown Orientation IJACSA, 2015.

[9] Associate Prof. Mohamed El-Mahallawy,Dr. Eid Emary,A Fast Shredded Document Reconstruction Arab Academy for Science Technology and Maritime Transport, 2015.

[10] G. Barequet and M. Sharir, “Partial surface and volume matching in three dimensions,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 9, pp. 929–948, 2002.

[11] H. Wolfson, “On curve matching,” IEEE Trans. Pattern Anal. Mach.Intell., vol. 12, no. 5, pp. 483–489, 2002.
[12] D. Eppstein, “Spanning trees and spanners,” in Handbook of Compu- tational Geometry. New York: Elsevier,2000, pp.425–461.
[13] C. Menard and R. Sablatnig, “On finding archaeological fragment as- assemblies using a bottom-up design,” in Proc. 21st Workshop Austrian Association for Pattern Recognition, 1997, pp. 203–207.
[14] C. Papaodysseus, T. Panagopoulos, M. Exarhos, C. Triantafillou, D. Fragoulis, and C. Doumas, “Contour-shape based reconstruction of fragmented, 1600 bc wall paintings,” IEEE Trans. Signal Process., vol. 50, no. 6, pp. 1277–1288, Jun. 2002.
[15] J. Canny, “A computational approach to edge detection,” IEEE Trans.Pattern Anal. Mach. Intell., vol. 8, no. 6, pp. 679–698, 1986.
[16] M. Marques and C. Freitas, “Reconstructing strip-shredded documents using color as feature matching,” in Proc. ACM Symp. Applied Com- puting, 2009, pp. 893–894.